FORTINET

Administration Guide

FortiOS 8.0.0

FORTINET DOCUMENT LIBRARY
https://docs.fortinet.com

FORTINET VIDEO LIBRARY
https://video.fortinet.com

FORTINET BLOG
https://blog.fortinet.com

CUSTOMER SERVICE & SUPPORT
https://support.fortinet.com

FORTINET TRAINING & CERTIFICATION PROGRAM
https://www.fortinet.com/training-certification

FORTINET TRAINING INSTITUTE
https://training.fortinet.com

FORTIGUARD LABS
https://www.fortiguard.com

END USER LICENSE AGREEMENT
https://www.fortinet.com/doc/legal/EULA.pdf

FEEDBACK
Email: techdoc@fortinet.com

June 15, 2026
FortiOS 8.0.0 Administration Guide
01-800-1247743-20260615

TABLE OF CONTENTS

Change Log **29**
Getting started **30**
Summary of steps 31
Setting up FortiGate for management access 32
Logging in to FortiOS GUI 33
Registering FortiGate 39
Completing the FortiGate Setup wizard 45
Configuring basic settings 46
Configuring a firewall policy 51
Backing up the configuration 51
Troubleshooting your installation 52
Using the GUI 54
Connecting using a web browser 54
Menus 55
Tables 56
Entering values 60
GUI-based global search 62
Loading artifacts from a CDN 64
Accessing additional support resources 64
Command palette 65
Recovering missing graphical components 71
Using the CLI 73
Connecting to the CLI 73
CLI basics 76
Command syntax 82
Subcommands 85
Permissions 87
FortiAI assistant and CLI Code Lab NEW 88
Requirements and tokens 88
Enabling FortiAI and accessing CLI Code Lab 90
FortiAI settings 93
FortiAI examples 97
Configuration and management 100
FortiExplorer Go 101
Migrating a configuration with FortiConverter 101
Accessing Fortinet Developer Network 107
Terraform: FortiOS as a provider 110
Product registration with FortiCare 115
FortiCare and FortiGate Cloud login 115
FortiCare Register button 118
Transfer a device to another FortiCloud account 119
Deregistering a FortiGate 121
FortiGate models 122
Differences between models 122
Low encryption models 123

LEDs 123
Proxy-related features not supported on FortiGate 2 GB RAM models 126
GUI features not supported on FortiGate 2 GB RAM models 128
FGR-70G/70F/70F-3G4G GPIO/DIO module 129
Default SD-WAN configuration on FortiGate models with two WAN ports 131
FortiData 133
Connecting FortiData to FortiGate 134
FortiSASE-Sovereign 136
Enabling FortiSASE-Sovereign 137
Disabling FortiSASE-Sovereign 138
FortiTelemetry 139
Configuring a FortiTelemetry Controller 140
Dashboard 142
Status dashboard 143
Updating system information 144
Resource widgets 144
Viewing Fabric devices 145
Viewing administrators 146
Viewing logs sent for remote logging source 146
Assets and Identities dashboard 147
Assets 147
DHCP monitor 154
Firewall Users monitor 155
Network Monitor dashboard 157
Routing monitor 157
Security 161
VPN monitor 162
FortiView dashboard 166
Using FortiView 167
Enabling FortiView from devices 170
FortiView AI widgets 173
FortiView sources 173
FortiView Sessions 175
FortiView Top Source and Top Destination Firewall Objects monitors 177
WiFi dashboard 179
FortiAP Status monitor 180
Clients by FortiAP monitor 181
Monitors and widgets 184
FortiView monitors and widgets 184
Security Fabric widgets and monitors 186
Network widgets and monitors 186
System widgets and monitors 187
Resource Usage widgets 187
Security widgets and monitors 188
User & Authentication widgets and monitors 188
WiFi widgets and monitors 188
WAN Opt. & Cache widgets and monitors 189
Widgets in VDOM mode 189

Using dashboards of widgets and monitors 192
Adding dashboards of widgets and monitors 192
Editing dashboards of widgets and monitors 194
Using widgets 195
Resetting all dashboards 197
Viewing device dashboards in the Security Fabric 197
Creating a fabric system and license dashboard 198
Adding an Agentless VPN dashboard 200
Adding a FortiTelemetry dashboard 201
Examples 203
Top websites and sources by category 203
Cloud application view 206
Application risk levels 215
Application performance monitoring analytics NEW 216
Network 222
Interfaces 222
Interface settings 225
Physical interface 249
VLAN 252
Aggregation and redundancy 267
Loopback interface 276
Software switch 278
Hardware switch 280
Zone 289
Virtual wire pair 291
Enhanced MAC VLAN 301
VXLAN 304
DNS 358
Important DNS CLI commands 358
DNS domain list 363
FortiGate DNS server 365
DDNS 385
DNS latency information 390
DNS over TLS and HTTPS 392
Transparent conditional DNS forwarder 396
Interfaces in non-management VDOMs as the source IP address of the DNS conditional forwarding server 400
DNS session helpers 402
DNS troubleshooting 404
Explicit and transparent proxies 405
Explicit web proxy 406
FTP proxy 410
Transparent proxy 414
Proxy policy addresses 416
IPv6 proxy address and address group objects New 423
Proxy policy security profiles 424
Explicit proxy authentication 429
Transparent web proxy forwarding 435

Transparent web proxy forwarding over IPv6 439
Upstream proxy authentication in transparent proxy mode 443
Multiple dynamic header count 444
Restricted SaaS access 447
Explicit proxy and FortiGate Cloud Sandbox 456
Proxy chaining 459
WAN optimization SSL proxy chaining 464
Agentless NTLM authentication for web proxy 472
Multiple LDAP servers in Kerberos keytabs and agentless NTLM domain controllers 476
Learn client IP addresses 477
Explicit proxy authentication over HTTPS 478
mTLS client certificate authentication 480
CORS protocol in explicit web proxy when using session-based, cookie-enabled, and captive portal-enabled SAML authentication 486
Display CORS content in an explicit proxy environment 489
HTTP connection coalescing and concurrent multiplexing for explicit proxy 491
Secure explicit proxy 493
Secure explicit proxy with client certificates 496
Explicit proxy logging 498
Configuring fast fallback for explicit proxy 508
Forward HTTPS requests to a web server without the need for an HTTP CONNECT message 512
Specifying outgoing interface and VRF for a web proxy forward server or isolator server 514
Isolator servers in proxy policies 516
Secure explicit proxy with client certificate blocklist enforcement NEW 520
Session and user limits 523
IPAM 524
IPAM introduction 525
Advanced configurations 530
Configuration examples 540
Diagnostics 549
DHCP servers and relays 550
Default DHCP server for entry-level FortiGates 551
Basic configuration 551
DHCP options 555
DHCP addressing mode on an interface 563
VCI pattern matching for DHCP assignment 566
DHCP shared subnet 568
Multiple DHCP relay servers 570
DHCP smart relay on interfaces with a secondary IP 571
FortiGate DHCP works with DDNS to allow FQDN connectivity to leased IP addresses 574
Static routing 574
Routing concepts 575
Policy routes 590
Equal cost multi-path 595
Dual internet connections 599

Dynamic routing 605
RIP 607
OSPF 628
BGP 646
BFD 720
Routing objects 731
Multicast 742
Multicast routing and PIM support 742
Configuring multicast forwarding 748
Using IPS inspection for multicast UDP traffic 754
Including denied multicast sessions in the session table 757
IPv6 multicast routing BSR support New 759
FortiExtender 766
WAN extension mode 767
LAN extension mode 767
Maximum FortiExtender devices supported per mode 767
Adding a FortiExtender 767
Cellular modems 769
Automatic LTE connection establishment 770
Direct IP support for LTE/4G 770
Cellular interface support for IPv6 773
Active SIM card switching 776
Airplane mode and LTE/BLE 784
Upgrade LTE modem firmware directly from FortiGuard 786
LLDP reception 788
Virtual routing and forwarding 791
Implementing VRF 791
VRF routing support 792
Route leaking between VRFs with BGP 802
Route leaking between multiple VRFs 805
VRF with IPv6 816
IBGP and EBGP support in VRF 820
Support cross-VRF local-in and local-out traffic for local services 822
Support specific VRF ID for local-out traffic 825
NetFlow 832
Verification and troubleshooting 834
NetFlow templates 835
NetFlow on FortiExtender and tunnel interfaces 850
Allow multiple NetFlow collectors 854
Netflow sampling 860
Filter NetFlow sampling 862
sFlow 867
Configuring sFlow 867
Link monitor 873
Link monitor with route updates 874
Enable or disable updating policy routes when link health monitor fails 875
Add weight setting on each link health monitor server 878
SLA link monitoring for dynamic IPsec and SSL VPN tunnels 881

HA remote link monitoring ... 884
IPv6 ... 887
IPv6 overview ... 888
IPv6 quick start ... 889
Neighbor discovery proxy ... 892
IPv6 address assignment ... 894
NAT66, NAT46, NAT64, and DNS64 ... 909
DHCPv6 relay ... 922
IPv6 tunneling ... 923
IPv6 Simple Network Management Protocol ... 935
Dynamic routing in IPv6 ... 937
IPv6 configuration examples ... 940
FortiGate LAN extension ... 995
Example CLI configuration ... 995
Example GUI configuration ... 1002
DHCP client mode for inter-VDOM links ... 1007
FortiGate secure edge to FortiSASE ... 1008
WiFi access point with internet connectivity ... 1012
SCTP packets with zero checksum on the NP7 platform ... 1020
Industrial Connectivity ... 1021
Sample configuration to convert IEC 60870-5-101 serial to IEC 60870-5-104 TCP/IP transport ... 1023
Sample configuration to convert Modbus serial to Modbus TCP ... 1024
Diagnostics ... 1025
Using the packet capture tool ... 1025
Using the debug flow tool ... 1032
SD-WAN ... 1037
SD-WAN overview ... 1037
SD-WAN components and design principles ... 1038
SD-WAN designs and architectures ... 1040
SD-WAN quick start ... 1041
Underlay ... 1042
Overlay ... 1043
Routing ... 1043
Security ... 1044
SD-WAN ... 1044
Results ... 1047
Configuring SD-WAN in the CLI ... 1049
SD-WAN members and zones ... 1052
Topology ... 1052
Configuring SD-WAN member interfaces ... 1053
Configuring SD-WAN zones ... 1055
Using SD-WAN zones ... 1056
Specify an SD-WAN zone in static routes and SD-WAN rules ... 1058
Defining a preferred source IP for local-out egress interfaces on SD-WAN members ... 1063
Specify SD-WAN zones in some policies ... 1065
SD-WAN performance SLA ... 1068

Performance SLA overview ... 1068
Link health monitor ... 1073
Monitoring performance SLA ... 1076
Passive WAN health measurement ... 1081
Passive health-check measurement by internet service and application ... 1087
Mean opinion score calculation and logging in performance SLA health checks ... 1092
SD-WAN application monitor using FortiMonitor ... 1095
Classifying SLA probes for traffic prioritization ... 1099
SD-WAN rules ... 1105
SD-WAN rules overview ... 1105
Implicit rule ... 1114
Strategy ... 1118
SDN dynamic connector addresses in SD-WAN rules ... 1143
Application steering using SD-WAN rules ... 1145
DSCP tag-based traffic steering in SD-WAN ... 1158
ECMP support for the longest match in SD-WAN rule matching ... 1165
Override quality comparisons in SD-WAN longest match rule matching ... 1168
Internet service and application control steering ... 1171
Use maximize bandwidth to load balance traffic between ADVPN shortcuts ... 1181
Use SD-WAN rules to steer multicast traffic ... 1189
Use SD-WAN rules for WAN link selection with load balancing ... 1209
Ensuring SD-WAN prioritizes healthy SD-WAN members ... 1217
SD-WAN Underlay Bandwidth and Quality Monitoring service ... 1218
Application performance and configuration ... 1218
Speed tests ... 1236
Overlay ... 1277
VPN ... 1277
ADVPN ... 1283
ADVPN 2.0 ... 1342
Self-healing and application protection ... 1382
Routing ... 1426
Routing overview ... 1426
Advanced BGP settings ... 1434
Local out traffic ... 1468
Security ... 1475
Advanced configuration ... 1475
HA ... 1476
SD-WAN configuration portability ... 1490
SD-WAN segmentation over a single overlay ... 1495
SD-WAN segmentation over a single overlay using IPv6 ... 1512
Copying the DSCP value from the session original direction to its reply direction ... 1520
Matching BGP extended community route targets in route maps ... 1524
SD-WAN multi-PoP multi-hub large scale design and failover ... 1530
SD-WAN traffic shaping and QoS ... 1549
Troubleshooting SD-WAN ... 1554
Tracking SD-WAN sessions ... 1555

Understanding SD-WAN related logs 1555
SD-WAN related diagnose commands 1558
Using SNMP to monitor health check 1563
SD-WAN cloud on-ramp 1567
Configuring the VPN overlay between the HQ FortiGate and cloud FortiGate-VM 1568
Configuring the VPN overlay between the HQ FortiGate and AWS native VPN gateway 1572
Configuring the VIP to access the remote servers 1576
Configuring the SD-WAN to steer traffic between the overlays 1579
Verifying the traffic 1582
Zero Trust Network Access 1590
Zero Trust Network Access introduction 1590
ZTNA application gateway and IP/MAC based access control 1590
ZTNA telemetry, tags, and policy enforcement 1591
Application gateway 1591
Basic ZTNA configuration components 1592
ZTNA service connector 1593
Basic ZTNA configuration 1593
Establish device identity and trust context with FortiClient EMS 1608
SSL certificate based authentication 1614
Full versus simple ZTNA policies 1616
Types of security posture tags 1623
ZTNA tag caching and direct tag synchronization 1628
Certificate verification for real servers 1629
ZTNA advanced configurations 1635
Access control of unmanageable and unknown devices 1635
HTTP2 connection coalescing and concurrent multiplexing for ZTNA 1641
Fabric integration with FortiGSLB 1644
ZTNA agentless web-based application access 1649
ZTNA domain matching 1660
ZTNA service connector NEW 1667
ZTNA connector - reverse gateway and forwarder New 1674
ZTNA configuration examples 1676
ZTNA web proxy example 1677
ZTNA web proxy with load balancing example 1691
ZTNA web proxy with basic authentication example 1696
ZTNA traffic forwarding proxy example 1708
ZTNA traffic forwarding proxy with FQDN example 1719
ZTNA SSH traffic forwarding proxy example 1726
ZTNA application gateway with SAML authentication example 1735
ZTNA application gateway with SAML and MFA using FortiAuthenticator example 1750
Secure LDAP connection from FortiAuthenticator with zero trust tunnel example 1776
ZTNA IP MAC based access control example 1776
ZTNA IPv6 examples 1788
ZTNA Zero Trust application gateway example 1795

ZTNA SaaS application access control with Inline CASB 1795
ZTNA application gateway with KDC to access shared drives 1801
Custom replacement message for ZTNA virtual hosts 1808
ZTNA single sign-on with Entra ID 1811
ZTNA troubleshooting and debugging commands 1820
Troubleshooting usage and output 1822
ZTNA troubleshooting scenarios 1827
ZTNA access control 1827
IP/MAC based access control 1828
Other useful CLI commands 1831
ZTNA troubleshooting with error messages 1831
Correlating with FortiGate ZTNA event logs 1833
Additional Examples 1834
Policy and Objects 1841
Policies 1841
Firewall policy 1842
NGFW policy 1868
Local-in policy 1886
DoS policy 1901
Telemetry policy 1909
Access control lists 1911
Interface policies 1912
Source NAT 1913
Destination NAT 1939
Examples and policy actions 1965
Address objects 2031
Address types 2032
Address group 2034
Subnet 2034
Dynamic policy — Fabric devices 2035
IP range 2038
FQDN addresses 2038
Using wildcard FQDN addresses in firewall policies 2040
Geography based addresses 2042
IPv6 geography-based addresses 2046
Wildcard addressing 2048
Interface subnet 2050
IPAM firewall address New 2051
Address group 2057
Address folders 2057
Allow empty address groups 2059
Address group exclusions 2060
FSSO dynamic address subtype 2061
ClearPass integration for dynamic address objects 2065
FortiNAC tag dynamic address 2068
FortiVoice tag dynamic address 2068
MAC address-based policies 2072
ISDB well-known MAC address list 2074

IPv6 MAC addresses and usage in firewall policies 2076
RSSO dynamic address subtype 2078
ISDB record for SOCaaS 2080
Telemetry addresses and address groups 2083
Custom tags NEW 2089
CLI syntax 2090
GUI 2091
Example 2095
Protocol options 2101
Log oversized files 2101
RPC over HTTP 2102
Protocol port mapping 2102
Common options 2102
Web options 2103
Email options 2103
Configuring handling of unknown content types 2104
Stripping the X-Forwarded-For value in the HTTP header 2105
Traffic shaping 2108
Configuration methods 2108
Traffic shaping policy 2110
Traffic shaping policies 2110
Traffic shaping profiles 2120
Traffic shapers 2131
Global traffic prioritization 2148
DSCP matching and DSCP marking 2152
Examples 2159
Internet Services 2178
Using Internet Service in a policy 2179
Using custom Internet Service in policy 2183
Using extension Internet Service in policy 2185
Global IP address information database 2188
IP reputation filtering 2190
Internet service groups in policies 2191
Allow creation of ISDB objects with regional information 2198
Internet service customization 2200
Look up IP address information from the Internet Service Database page 2201
Internet Service Database on-demand mode 2202
Enabling the ISDB cache in the FortiOS kernel 2205
ISDB FQDN groups 2206
Security Profiles 2212
Inspection modes 2212
Flow mode inspection (default mode) 2213
Proxy mode inspection 2213
Inspection mode feature comparison 2214
Proxy-based inspection for email protocols supported on models with 2 GB RAM 2218
AI protection 2224
Protecting GenAI access using web filter 2225

Protecting GenAI access using application control 2227
Protecting GenAI Access using DLP 2232
Antivirus 2240
Antivirus introduction 2240
Advanced configurations 2267
Configuration examples 2295
Web filter 2311
Web filter introduction 2311
Advanced CLI configuration 2342
Configuration examples 2362
Video filter 2372
Configuring a video filter profile 2372
YouTube API key 2373
Filtering based on FortiGuard categories 2373
Filtering based on YouTube channel 2378
Filtering based on title 2380
Filtering based on description 2381
Configuring a video filter keyword list 2382
Replacement messages displayed in blocked videos 2384
DNS filter 2386
DNS filter behavior in proxy mode 2387
FortiGuard DNS rating service 2387
Configuring a DNS filter profile 2388
FortiGuard category-based DNS domain filtering 2393
Botnet C&C domain blocking 2396
DNS safe search 2400
Local domain filter 2402
DNS translation 2406
Applying DNS filter to FortiGate DNS server 2411
DNS inspection with DoT and DoH 2412
DNS over QUIC and DNS over HTTP3 for transparent and local-in DNS modes 2416
Troubleshooting for DNS filter 2421
Telemetry profile 2424
Predefined applications 2426
Creating telemetry profiles 2426
Application control 2428
Configuring an application sensor 2429
Application matching signature priority 2430
Basic category filters and overrides 2432
Excluding signatures in application control profiles 2437
Port enforcement check 2439
Protocol enforcement 2440
SSL-based application detection over decrypted traffic in a sandwich topology 2442
Matching multiple parameters on application control signatures 2443
Application signature dissector for DNP3 2446
Domain fronting protection 2446
Inline CASB 2448

Inline CASB examples 2449
Intrusion prevention 2478
Signature-based defense 2480
Configuring an IPS sensor 2483
IPS configuration options 2486
SCTP filtering capabilities 2492
Diameter protocol inspection 2494
IPS signature filter options 2498
IPS with botnet C&C IP blocking 2502
IPS signatures for the operational technology security service 2506
IPS sensor for IEC 61850 MMS protocol 2508
IPS Modbus TCP decoder 2509
AI and ML-based IPS detection 2511
Support Ethernet layer protocols in the IPS engine 2514
File filter 2518
Supported file types 2520
Configuring a file filter profile 2523
Configuring a file filter profile with the warning action NEW 2527
Email filter 2531
Protocol comparison between email filter inspection modes 2531
Configuring an email filter profile 2532
Local-based filters 2532
FortiGuard-based filters 2540
Third-party-based filters 2542
Filtering order 2543
Protocols and actions 2545
Configuring webmail filtering 2546
Spam email header 2547
VoIP solutions 2548
General use cases 2548
NAT46 and NAT64 for SIP ALG 2552
SIP message inspection and filtering 2560
SIP ALG and SIP session helper 2566
SIP pinholes 2573
SIP over TLS 2575
Voice VLAN auto-assignment 2576
Scanning MSRP traffic 2578
ICAP 2582
ICAP configuration example 2583
ICAP response filtering 2586
Secure ICAP clients 2588
ICAP scanning with SCP and FTP 2589
Domain name in XFF with ICAP 2592
Protocol transparency for ICAP 2596
Selective forwarding to ICAP server 2597
Web application firewall 2599
Protecting a server running web applications 2600
Data loss prevention 2602

Protocol comparison between DLP inspection modes 2603
Archiving 2604
Logging and blocking files by file name 2604
DLP techniques 2605
Basic DLP settings 2606
Advanced DLP configurations 2612
DLP fingerprinting 2615
FortiGuard DLP service 2619
Sensitivity labels 2623
Exact data matching 2631
DLP examples 2640
Virtual patching 2675
OT and IoT device detection 2676
Virtual patching profiles 2676
Virtual patching signatures 2678
License and entitlement information 2680
OT virtual patching basic examples 2681
OT and IoT virtual patching on NAC policies 2687
SSL & SSH Inspection 2690
Configuring an SSL/SSH inspection profile 2691
Certificate inspection 2694
Deep inspection 2697
Protecting an SSL server 2700
Handling SSL offloaded traffic from an external decryption device 2701
SSH traffic file scanning 2703
Redirect to WAD after handshake completion 2706
HTTP/2 support in proxy mode SSL inspection 2706
Define multiple certificates in an SSL profile in replace mode 2708
Disabling the FortiGuard IP address rating 2710
Block or allow ECH TLS connections 2711
Configuring certificate probe failure option 2720
Hybrid PQC support in proxy mode SSL deep inspection 2721
Custom signatures 2724
Configuring custom signatures 2725
Blocking applications with custom signatures 2726
Filters for application control groups 2730
Application groups in traffic shaping policies 2733
Overrides 2736
Web rating override 2736
Using local and remote categories 2745
Web profile override 2747
IP ban 2751
IP ban using the CLI 2752
IP ban using security profiles 2753
Configuring the persistency for a banned IP list 2755
Profile groups 2757
IPsec VPN 2760
General IPsec VPN configuration 2760

Network topologies 2761
Phase 1 configuration 2761
Phase 2 configuration 2795
Tunneling protocol and encapsulation 2800
Post-Quantum readiness 2820
VPN security policies 2840
Encryption algorithms 2843
Blocking unwanted IKE negotiations and ESP packets with a local-in policy 2852
Renaming IPsec tunnels 2854
Site-to-site VPN 2857
FortiGate-to-FortiGate 2857
FortiGate-to-third-party 2935
Remote access 2961
FortiGate as dialup client 2962
FortiClient as dialup client 2973
Add FortiToken multi-factor authentication 2985
Add LDAP user authentication 2988
Configure FortiClient SIA for IPsec VPN tunnels 2993
Remote access VPNs with IPAM addressing New 3003
LDAP authentication with TCP as transport 3007
VPN 2FA with EAP and certificate authentication 3019
Using single or multiple user groups for user authentication 3024
VPN with FortiClient Standalone and FortiIdentity Cloud New 3026
Native VPN as dialup client - Windows New 3043
Native VPN as dialup client - iOS 3063
Windows IKEv2 native VPN with user certificate 3070
Dialup VPN with certificate authentication 3094
Dialup VPN with certificate and LDAP authentication 3114
SAML-based authentication for FortiClient remote access VPNs 3124
Enhancing IPsec security using EMS SN verification 3146
IKE Mode Config clients 3146
IPsec DNS suffix 3153
Defining gateway IP addresses in IPsec with mode-config and DHCP 3158
Dialup IPsec VPN using custom TCP port 3159
Matching IPsec tunnel gateway based on address parameters 3172
Enforcing security posture verification before dial-up VPN connection 3174
Aggregate and redundant VPN 3186
Manual redundant VPN configuration 3187
OSPF with IPsec VPN for network redundancy 3190
IPsec VPN in an HA environment 3198
Packet distribution and redundancy for aggregate IPsec tunnels 3205
Packet distribution for aggregate dial-up IPsec tunnels using location ID 3217
Packet distribution for aggregate static IPsec tunnels in SD-WAN 3222
Packet distribution for aggregate IPsec tunnels using weighted round robin 3227
Redundant hub and spoke VPN 3228
ADVPN 3235
IPsec VPN wizard hub-and-spoke ADVPN support 3235
ADVPN with BGP as the routing protocol 3244
ADVPN with OSPF as the routing protocol 3255

ADVPN with RIP as the routing protocol 3264
UDP hole punching for spokes behind NAT 3274
Fabric Overlay Orchestrator 3277
Prerequisites 3278
Network topology 3278
Using the Fabric Overlay Orchestrator 3279
SPA easy configuration key for FortiSASE 3297
Fabric Overlay Orchestrator Topology dashboard widget for hub FortiGates 3299
Other VPN topics 3302
VPN and ASIC offload 3302
Configure DSCP for IPsec tunnels 3313
IPsec IKE load balancing based on FortiSASE account information 3315
IPsec SA key retrieval from a KMS server using KMIP 3318
Multiple interface monitoring for IPsec 3329
VPN IPsec troubleshooting 3335
Understanding VPN related logs 3336
IPsec related diagnose commands 3338
Agentless VPN 3349
Agentless VPN portal configurations 3351
Quick Connection tool 3352
RDP sessions 3353
Agentless VPN bookmarks 3354
Configuring bookmarks 3354
Configuring group-based Agentless VPN bookmarks 3356
Agentless VPN for remote users 3357
Sample topology 3357
Sample configuration 3357
Customizing the RDP display size 3361
Example 3362
Showing the Agentless VPN web portal login page in the browser's language 3365
Agentless VPN custom landing page 3366
Example 3366
Agentless VPN protocols 3371
TLS 1.3 support 3371
SMBv2 support 3372
Agentless VPN security best practices 3372
Agentless VPN settings 3373
Authentication 3375
Authorization 3376
Agentless VPN authentication 3379
Agentless VPN with LDAP user authentication 3380
Agentless VPN with LDAP user password renew 3386
Agentless VPN with certificate authentication 3392
Agentless VPN with LDAP-integrated certificate authentication 3398
Agentless VPN for remote users with MFA and user sensitivity 3405
Agentless VPN with FortiToken mobile push authentication 3413
Agentless VPN with RADIUS on FortiAuthenticator 3419

Agentless VPN with RADIUS and FortiToken mobile push on FortiAuthenticator 3425
Agentless VPN with RADIUS password renew on FortiAuthenticator 3430
Agentless VPN with RADIUS on Windows NPS 3435
Agentless VPN with multiple RADIUS servers 3441
Agentless VPN with local user password policy 3452
Agentless VPN multi-realm 3458
NAS-IP support per Agentless VPN realm 3464
Agentless VPN with Okta as SAML IdP 3466
Agentless VPN with Microsoft Entra SSO integration 3472
Disable the clipboard for Agentless VPN RDP connections 3472
Example 3472
Post-Quantum Cryptography for Agentless VPN 3477
Use Case 1: FIPS-CC compliance customer 3478
Use Case 2: Normal mode customer testing PQC 3480
Agentless VPN troubleshooting 3481
Debug commands 3481
Troubleshooting common issues 3482
User & Authentication 3484
User definition, groups, and settings 3485
Users 3485
User groups 3487
Authentication settings 3495
Retail environment guest access 3500
Customizing complexity options for the local user password policy 3502
Basic authentication with cached client certificates 3507
LDAP servers 3509
Configuring an LDAP server 3509
Enabling Active Directory recursive search 3513
Configuring LDAP dial-in using a member attribute 3514
Configuring wildcard admin accounts 3515
Configuring least privileges for LDAP admin account authentication in Active Directory 3517
Tracking users in each Active Directory LDAP group 3518
Tracking rolling historical records of LDAP user logins 3521
Configuring client certificate authentication on the LDAP server 3524
RADIUS servers 3527
Configuring a RADIUS server 3528
Using multiple RADIUS servers 3531
RADIUS AVPs and VSAs 3534
RADIUS VSAs for captive portal redirects 3536
Restricting RADIUS user groups to match selective users on the RADIUS server 3538
Configuring RADIUS SSO authentication 3539
Trigger RADIUS authentication with DNS and ICMP queries 3545
RSA ACE (SecurID) servers 3548
Support for Okta RADIUS attributes filter-Id and class 3552
Sending multiple RADIUS attribute values in a single RADIUS Access-Request 3554

Traffic shaping based on dynamic RADIUS VSAs 3555
RADIUS Termination-Action AVP in wired and wireless scenarios 3562
Configuring a RADSEC client 3566
RADIUS integrated certificate authentication for Agentless VPN 3571
SAML 3573
Usage 3574
Identity providers 3574
Response and assertion signing 3575
Example 3575
Configuring SAML SSO 3576
Agentless VPN with FortiAuthenticator as a SAML IdP 3581
IPsec VPN with SAML IdP 3586
Outbound firewall authentication with Microsoft Entra ID as a SAML IdP 3586
SAML authentication in a proxy policy 3597
SAML authentication in a proxy policy using SCIM 3601
Local entries for SAML users 3606
Per-Session SAML authentication logging and logout support for ZTNA and explicit proxy users 3610
SSO integration with FortiIdentity Cloud 3615
TACACS+ servers 3620
SCIM servers 3622
FortiGate as SCIM server and FAC as SCIM client 3624
Using SAML with a SCIM group in a firewall policy NEW 3627
Bearer token authentication for SCIM servers 3630
Example 3630
FortiTokens 3633
FortiToken Mobile quick start 3635
FortiIdentity Cloud 3642
Registering hard tokens 3642
Managing FortiTokens 3645
FortiToken Mobile Push 3647
Synchronizing LDAP Active Directory users to FortiIdentity Cloud using the two-factor filter 3649
Enable FortiIdentity Cloud seats directly from FortiGate 3653
FortiGuard distribution of updated Apple certificates for push notifications 3655
Configure FTM push with dynamic IP handling in the GUI 3656
Troubleshooting and diagnosis 3658
PKI 3661
Configuring a PKI user 3662
Using the SAN field for LDAP-integrated certificate authentication 3666
FSSO 3669
FSSO polling connector agent installation 3672
FSSO using Syslog as source 3675
Managing collector agent connection failures 3678
Configuring FSSO firewall authentication 3680
Include usernames in logs 3687
Install and configure FSSO Agent 3687
Configure the FortiGate 3690

Log, monitor, and report examples ... 3692
Wireless configuration ... 3695
Switch Controller ... 3696
System ... 3697
Basic system settings ... 3697
Advanced system settings ... 3697
Operating modes ... 3698
Administrators ... 3700
Local authentication ... 3701
Remote authentication for administrators ... 3703
Administrator account options ... 3706
REST API administrator ... 3712
SSO administrators ... 3714
FortiCloud SSO ... 3715
Allowing the FortiGate to override FortiCloud SSO administrator user permissions ... 3716
Password policy ... 3721
Public key SSH access ... 3725
Separating the SSHD host key from the administration server certificate ... 3728
Restricting SSH and Telnet jump host capabilities ... 3729
Restricting local administrator logins through the console ... 3730
Remote administrators with TACACS+ VSA attributes ... 3732
Administrator profiles ... 3736
Access to FortiAI ... 3736
super_admin profile ... 3737
Creating customized profiles ... 3737
Controlling CLI system permissions ... 3738
Displaying execute commands for custom system permissions ... 3739
Editing profiles ... 3740
Deleting profiles ... 3740
Firmware & Registration ... 3741
About firmware installations ... 3742
Firmware labels ... 3743
Upgrading individual devices ... 3749
Upgrading all devices ... 3752
Registering devices ... 3755
Enabling automatic firmware upgrades ... 3757
Upgrade prompt when a critical vulnerability is detected upon login ... 3767
Viewing firmware upgrade reports ... 3769
Authorizing devices ... 3773
Firmware upgrade notifications ... 3775
Downloading a firmware image ... 3775
Testing a firmware version ... 3776
Installing firmware from system reboot ... 3777
Restoring from a USB drive ... 3780
Using controlled upgrades ... 3781
Downgrading individual device firmware ... 3781
Downloading the EOS support package for supported Fabric devices ... 3784

How the FortiGate firmware license works 3787
Enhanced firmware upgrade management for extension devices 3791
Settings 3795
General tab 3795
Access tab 3810
Device tab 3823
Preference tab 3823
Trusted platform module support 3830
Configure TCP NPU session delay globally 3833
Virtual Domains 3835
VDOM overview 3835
General configurations 3840
Configuring global profiles 3848
Backing up and restoring configurations in multi-VDOM mode 3849
Inter-VDOM routing configuration example: Internet access 3853
Inter-VDOM routing configuration example: Partial-mesh VDOMs 3863
SNMP 3877
Basic configuration 3877
MIB files 3881
Access control for SNMP 3882
Important SNMP traps 3884
SNMP traps and automation-stitch notifications for DIO module 3887
SNMP examples 3890
Replacement messages 3899
Modifying replacement messages 3900
Replacement message images 3901
Replacement message groups 3903
FortiGuard 3906
Subscriptions 3907
FortiGuard settings 3925
Anycast 3949
Anycast and unicast services 3951
Online security tools 3952
Feature visibility 3952
Certificates 3953
Automatically provision a certificate 3955
Generate a new certificate 3960
Regenerate default certificates 3961
Import a certificate 3962
Generate a CSR 3964
CA certificate 3967
Remote certificate 3968
Certificate revocation list 3968
Export a certificate 3969
Uploading certificates using an API 3970
Procuring and importing a signed SSL certificate 3974
Microsoft CA deep packet inspection 3977
Administrative access using certificates 3982

Creating certificates with XCA 3982
Enrollment over Secure Transport for automatic certificate management 3990
ACME External Account Binding support 4001
Security 4005
BIOS-level signature and file integrity checking 4005
Real-time file system integrity checking 4009
Running a file system check automatically 4013
Built-in entropy source 4013
FortiGate VM unique certificate 4015
Closed network VM license security 4016
Encrypt configuration files in the eCryptfs file system 4018
FIPS-CC mode and OpenSSL FIPS provider 4018
Enhanced administrator password security 4019
Black box for FortiGate 4022
DoS protection 4025
Configuration scripts 4028
Workspace mode 4029
RAID 4031
FortiGate encryption algorithm cipher suites 4034
HTTPS access 4034
SSH access 4035
Agentless VPN 4037
Additional features 4040
Other Products 4042
Conserve mode 4044
Proxy inspection in conserve mode 4045
Flow inspection in conserve mode 4045
Diagnostics 4046
Using APIs 4046
Token-based authentication 4047
Best Practices 4047
Making an API call to retrieve information from the FortiGate 4048
Configuration backups and reset 4053
Backing up and restoring configurations from the GUI 4053
Backing up and restoring configurations from the CLI 4056
Configuration revision 4060
Restore factory defaults 4061
Secure file copy 4062
Fortinet Support Tool for capturing incidents 4063
High Availability 4070
FortiGate Clustering Protocol (FGCP) 4070
FortiGate Session Life Support Protocol (FGSP) 4070
VRRP 4071
FGCP 4071
Critical cluster components 4071
General operation 4072
Failover 4072

Synchronizing the configuration 4072
Topics 4073
Cluster operation 4074
HA heartbeat 4097
Session synchronization and failover 4109
Management access 4131
Maintenance 4141
Advanced operations 4147
Troubleshooting 4197
Best practices 4209
FGSP 4211
Session pickup 4212
Session synchronization link 4212
Expectation sessions 4212
NAT session synchronization 4213
FGSP basic peer setup 4214
Synchronizing sessions between FGCP clusters 4219
Session synchronization interfaces in FGSP 4220
UTM inspection on asymmetric traffic in FGSP 4223
UTM inspection on asymmetric traffic on L3 4226
FGSP support for failover with asymmetric traffic and UTM 4228
Monitor routing prefix for FGSP session failover 4230
FGSP synchronizes firewall-authenticated users 4233
Encryption for L3 on asymmetric traffic in FGSP 4237
Optimizing FGSP session synchronization and redundancy 4238
Firmware upgrades in FGSP 4243
FGSP session synchronization between different FortiGate models or firmware versions 4244
Applying the session synchronization filter only between FGSP peers in an FGCP over FGSP topology 4245
FGSP static site-to-site IPsec VPN setup 4252
FGSP per-tunnel failover for IPsec 4254
FGCP over FGSP per-tunnel failover for IPsec 4260
Allow IPsec DPD in FGSP members to support failovers 4270
Standalone configuration synchronization 4276
Layer 3 unicast standalone configuration synchronization 4278
VRRP 4281
Basic VRRP configuration 4281
Adding IPv4 and IPv6 virtual routers to an interface 4282
VRRP failover 4283
VRRP groups 4287
VRRP virtual MACs 4288
Preempt mode 4289
Single-domain VRRP example 4290
Multi-domain VRRP example 4291
VRRP on EMAC-VLAN interfaces 4292
Ignore VRRP default route 4294

Fortinet Security Fabric **4297**
Components 4297
Security Fabric connectors 4301
Configuring the root FortiGate and downstream FortiGates 4302
Configuring logging and analytics 4310
Configuring FortiClient EMS 4322
Synchronizing FortiClient ZTNA tags 4344
Configuring LAN edge devices 4347
Configuring central management 4349
Configuring sandboxing 4354
Connecting to FortiTelemetry agents 4364
Connecting to FortiData 4369
Configuring supported connectors 4373
Allowing FortiDLP Agent communication through the FortiGate 4392
Using the Security Fabric 4392
Dashboard widgets 4393
Topology 4394
Asset Identity Center page 4400
OT asset visibility and network topology 4405
KEV information in IoT/OT vulnerabilities 4412
KEV information from FortiClient NEW 4416
WebSocket for Security Fabric events 4422
Deploying the Security Fabric 4423
Deploying the Security Fabric in a multi-VDOM environment 4431
Other Security Fabric topics 4437
Configuring the Security Fabric with SAML 4456
Configuring single-sign-on in the Security Fabric 4456
CLI commands for SAML SSO 4462
SAML SSO with pre-authorized FortiGates 4463
Navigating between Security Fabric members with SSO 4464
Integrating FortiAnalyzer management using SAML SSO 4466
Integrating FortiManager management using SAML SSO 4468
Advanced option - FortiGate SP changes 4469
Security rating 4470
Security Controls 4470
Vulnerabilities 4475
Security Fabric score 4477
Security rating insights 4478
Logging the security rating 4480
Security rating visibility 4480
Security rating tooltip controls 4484
Automation stitches 4487
Creating automation stitches 4488
Triggers 4506
Actions 4530
Public and private SDN connectors 4597
Getting started with public and private SDN connectors 4599
AliCloud SDN connector using access key 4603

AWS SDN connector using access keys 4606
Azure SDN connector using service principal 4611
Cisco ACI SDN connector using a standalone connector 4614
Import IPv6 addresses from an APIC controller 4616
Retrieve IPv6 dynamic addresses from Cisco ACI SDN connector 4620
ClearPass endpoint connector via FortiManager 4622
GCP SDN connector using service account 4625
IBM Cloud SDN connector using API keys 4627
Kubernetes (K8s) SDN connectors 4630
Nuage SDN connector using server credentials 4651
Nutanix SDN connector using server credentials 4654
OCI SDN connector using certificates 4656
OpenStack SDN connector using node credentials 4659
SAP SDN connector 4662
VMware ESXi SDN connector using server credentials 4665
VMware NSX-T Manager SDN connector using NSX-T Manager credentials 4667
Multiple concurrent SDN connectors 4671
Filter lookup in SDN connectors 4675
Support for wildcard SDN connectors in filter configurations 4677
Endpoint/Identity connectors 4679
Fortinet single sign-on agent 4679
Poll Active Directory server 4680
Symantec endpoint connector 4681
RADIUS single sign-on agent 4688
Exchange Server connector 4691
External feeds 4695
External resources file format 4696
External resource entry limit 4697
Configuring an external feed 4699
FortiGuard category external feed 4709
IP address external feed 4712
Domain name external feed 4716
MAC address external feed 4718
Malware hash external feed 4720
External feed connectors per VDOM 4722
STIX format for external feeds 4726
Using the AusCERT malicious URL feed with an API key 4728
Apply external feed connectors as source addresses in central SNAT 4732
Generic connector for importing addresses 4736
External feed update method example 4736
Push API update method example 4739
Troubleshooting 4743
Viewing a summary of all connected FortiGates in a Security Fabric 4743
Diagnosing automation stitches 4746
Log and Report 4750
Forward Traffic log page NEW 4750
Viewing event logs 4753
System Events log page 4755

Security Events log page 4760
Reports page 4764
FortiAnalyzer 4764
FortiGate Cloud 4766
Local 4767
Log settings and targets 4768
Global Settings 4768
Local Logs 4769
Threat Weight 4770
Configuring logs in the CLI 4771
Logging to FortiAnalyzer 4774
FortiAnalyzer log caching 4774
Configuring multiple FortiAnalyzers (or syslog servers) per VDOM 4777
Configuring multiple FortiAnalyzers on a FortiGate in multi-VDOM mode 4779
Switching to an alternate FortiAnalyzer if the main FortiAnalyzer is unavailable 4782
Advanced and specialized logging 4786
Logs for the execution of CLI commands 4786
Log buffer on FortiGates with an SSD disk 4788
Source and destination UUID logging 4790
Configuring and debugging the free-style filter 4792
Logging the signal-to-noise ratio and signal strength per client 4794
RSSO information for authenticated destination users in logs 4796
Destination user information in UTM logs 4799
Message ID in UTM logs 4803
Log fields for long-live sessions 4804
Generate unique user name for anonymized logs 4806
Configuring syslog overrides for VDOMs 4809
Customizing log formats for syslog servers NEW 4812
Logging MAC address flapping events 4818
Incorporating endpoint device data in the web filter UTM logs 4818
Logging detection of duplicate IPv4 addresses 4819
Including zone information fields in logs 4824
Local in and local out logging 4827
TACACS+ accounting details in audit logs NEW 4828
Sample logs by log type 4830
Troubleshooting 4854
Log-related diagnostic commands 4854
Backing up log files or dumping log messages 4860
SNMP OID for logs that failed to send 4862
WAN optimization 4866
Features 4866
Protocol optimization 4866
Byte caching 4866
Compression 4866
SSL offloading 4867
WAN optimization and HA 4867
Secure tunneling 4867

Prerequisites 4867
Disk usage 4868
Overview 4869
Client/server architecture 4869
Profiles 4870
Peers and authentication groups 4871
Tunnels 4872
Transparent mode 4873
Protocol optimization 4874
Cache service and video caching 4876
Manual and active-passive 4877
Monitoring performance 4878
System and feature operation with WAN optimization 4879
Best practices 4881
Example topologies 4881
In-path WAN optimization topology 4882
Out-of-path WAN optimization topology 4882
Topology for multiple networks 4882
Configuration examples 4883
Manual (peer-to-peer) WAN optimization configuration example 4884
Active-passive WAN optimization configuration example 4889
Secure tunneling configuration example 4894
Testing and troubleshooting the configuration 4900
VM 4903
Amazon Web Services 4903
Microsoft Azure 4903
Google Cloud Platform 4903
OCI 4904
AliCloud 4904
Private cloud 4904
VM license 4904
Uploading a license file 4905
License activation and validation 4906
VM license types 4907
Applying a FortiFlex token 4908
Consuming a new vCPU 4909
CLI troubleshooting 4909
Customizing the FortiFlex license token activation retry parameters 4911
Permanent trial mode for FortiGate-VM 4913
PF and VF SR-IOV driver and virtual SPU support 4916
Using OCI IMDSv2 4918
FIPS cipher mode for AWS, Azure, OCI, and GCP FortiGate-VMs 4921
Cloud-init 4923
TPM support for FortiGate-VM 4925

Hyperscale firewall ... **4934**
Troubleshooting ... **4935**
Troubleshooting methodologies ... 4936
Verify user permissions ... 4936
Establish a baseline ... 4936
Create a troubleshooting plan ... 4938
Connectivity Fault Management ... 4939
Example ... 4941
Troubleshooting scenarios ... 4942
System date and time settings ... 4944
Checking the hardware connections ... 4945
Checking FortiOS network settings ... 4946
Checking CPU and memory resources ... 4948
Troubleshooting high CPU usage ... 4951
Checking the modem status ... 4955
Running ping and traceroute ... 4956
Checking the logs ... 4961
Verifying routing table contents in NAT mode ... 4962
Verifying the correct route is being used ... 4962
Verifying the correct firewall policy is being used ... 4963
Checking the bridging information in transparent mode ... 4963
Checking wireless information ... 4965
Performing a sniffer trace or packet capture ... 4966
Debugging the packet flow ... 4968
Testing a proxy operation ... 4972
Displaying detail Hardware NIC information ... 4973
Performing a traffic trace ... 4978
Using a session table ... 4979
Finding object dependencies ... 4987
Diagnosing NPU-based interfaces ... 4988
Identifying the XAUI link used for a specific traffic stream ... 4990
Running the TAC report ... 4991
Using the process monitor ... 4991
Computing file hashes ... 4993
Other commands ... 4995
FortiGuard troubleshooting ... 4999
View open and in use ports ... 5004
IPS and AV engine version ... 5005
print tablesize ... 5006
CLI troubleshooting cheat sheet ... 5007
CLI error codes ... 5007
Additional resources ... 5007
Fortinet Document Library ... 5008
Release notes ... 5008
Fortinet Video Library ... 5008
Fortinet Community ... 5008
Fortinet Training Institute ... 5008
Fortinet Support ... 5009

Change Log

Date	Change Description
2026-04-21	Initial release.
2026-04-27	Added IPv6 proxy address and address group objects New on page 423 and BGP next hop self-support for VPNv4 and VPNv6 route reflectors New on page 688.
2026-05-08	Added Ensuring SD-WAN prioritizes healthy SD-WAN members on page 1217, BGP neighbor naming support New on page 690, and IPv6 multicast routing BSR support New on page 759. Updated Configuring PCP port mapping with SNAT and DNAT on page 2014.
2026-05-11	Added IPAM firewall address New on page 2051.
2026-05-14	Updated Post-Quantum readiness on page 2820.
2026-05-22	Added Certificate verification for real servers on page 1629 and ZTNA web proxy with load balancing example on page 1691. Updated ZTNA troubleshooting and debugging commands on page 1820.
2026-05-27	Added Session and user limits on page 523 and Configuring handling of unknown content types on page 2104. Updated IPsec DNS suffix on page 3153.
2026-05-29	Added Enhanced multicast session key for virtual wire pair New on page 298, EVPN support for VXLAN Anycast gateway and type-5 routing New on page 336, IPAM DHCP templates New on page 533, Dynamic BGP learning of ISDB reputation IPs New on page 705, Remote access VPNs with IPAM addressing New on page 3003, VPN with FortiClient Standalone and FortiIdentity Cloud New on page 3026, and SSO integration with FortiIdentity Cloud on page 3615.
2026-06-02	Added ZTNA tag caching and direct tag synchronization on page 1628.
2026-06-15	Updated PTPv2 on page 3800.

Getting started

FortiOS is the operating system that runs on Fortinet's FortiGate Next-Generation Firewall (NGFW). It supports different platforms, including:

- Physical appliances
- Hypervisors
- Cloud computing platforms

FortiOS delivers security as a hybrid mesh firewall that spans a meshed topology of on-prem and cloud environments. With FortiGuard's AI-powered security services, FortiOS provides protection across the attack surface with IPS, advanced malware protection, web security, inline malware prevention, data loss prevention, and more.

In addition, FortiOS is central to the SD-WAN solution by providing SD-WAN functionality and intelligence in a single FortiGate, a mesh of FortiGates, or integrated into a SASE environment. It is also central to the Zero Trust Network Access (ZTNA) solution by making policy decisions and applying policy enforcement based on security posture input.

Use the following resources to get started with FortiOS:

Task	Documentation links
Follow the steps to set up a new FortiGate	See Summary of steps on page 31. If you are migrating a configuration from another vendor to FortiGate, see the Migration section of the Best Practices guide or use the FortiConverter service.
Learn about best practices for FortiOS	Review Basic configuration in the Best Practices guide.
Learn about new FortiOS features	See FortiOS New Features and FortiOS Release Notes > New Features section.
Learn about standard practices for deploying a solution or an architecture	Go to Best Practices | 4-D Resources and review the document categories.
Review information about FortiOS releases, including resolved and known issues	See FortiOS Release Notes.

Summary of steps

Recent FortiGate models include a new BIOS version with a security enhancement that requires you to register your FortiGate and download the registration information to your FortiGate before you have complete access to the GUI or CLI. You must also register your FortiGate before you can upgrade its firmware.

These steps summarize how to get your FortiGate up and running by using the GUI. For information about the Command Line Interface (CLI), see Using the CLI on page 73.

1. Set up your FortiGate for initial management access through the GUI. See Setting up FortiGate for management access on page 32.

For more information	Go to
Physical appliances, such as FortiGate	Go to FortiGate/FortiOS Hardware Guides to view QuickStart Guides for all supported FortiGate models.
Hypervisors, such as FortiGate-VM on ESXi, KVM, Hyper-V, and so on.	Go to FortiGate Public Cloud or FortiGate Private Cloud and follow the deployment section of the administration guide for your hypervisor, for example, Microsoft Hyper-V Administration Guide > Deployment.

   Depending on the topology and FortiGate model, internet access may not yet be configured for the FortiGate.
2. Log in to the GUI and change your password to meet the password policy requirements. See Logging in to FortiOS GUI on page 33.

   FortiOS detects whether FortiGate has internet access to prepare for device registration to FortiCare. If no internet access is detected on recent FortiGate models, a wizard is displayed to help. See Configuring internet access for recent FortiGate models on page 35.
3. Register the FortiGate to FortiCare:
   - For recent FortiGate models, see Registering recent FortiGate models on page 39.
   - For older FortiGate models, see Registering FortiGate models on page 42. If no internet access, FortiGate cannot be registered until later in the setup.
4. Complete the *FortiGate Setup* wizard. See Completing the FortiGate Setup wizard on page 45.
5. Complete the basic configuration steps for FortiOS if needed. After this step, all FortiGate models should have internet access. See Configuring basic settings on page 46.
6. Configure a policy for the FortiGate to give clients behind FortiGate access to the internet. See Configuring a firewall policy on page 51.
7. Back up the configuration. See Backing up the configuration on page 51.
8. If necessary, troubleshoot the installation. See Troubleshooting your installation on page 52.

After completing the Getting started section, next steps can include:

- Getting familiar with the FortiOS GUI and CLI:
  - See Using the GUI on page 54.
  - See Using the CLI on page 73.
- Using the FortiAI assistant and CLI Code Lab to understand and configure FortiOS features. See FortiAI assistant and CLI Code Lab NEW on page 88.

- Configuring FortiOS features. The following table lists a few of the features available with FortiOS. Many additional features are available:

For	Go to
Security profiles	See antivirus, IPS, web filter, and application control.
VPN	See IPsec VPN on page 2760.
Fortinet Security Fabric	See Fortinet Security Fabric on page 4297.
User & Authentication	See User & Authentication on page 3484.
Software-defined wide area network (SD-WAN)	See SD-WAN on page 1037.
Zero Trust Network Access (ZTNA)	See Zero Trust Network Access on page 1590.

Setting up FortiGate for management access

After you receive your FortiGate, open the box, connect the cables for management and internet access, and use a management computer to access the FortiOS GUI.

For information about setting up FortiGate on hypervisors, such as FortiGate-VM on ESXi, KVM, Hyper-V, and so on, go to FortiGate Public Cloud or FortiGate Private Cloud and follow the deployment section of the administration guide for your hypervisor and cloud computing platform, for example, Microsoft Hyper-V Administration Guide > Deployment.

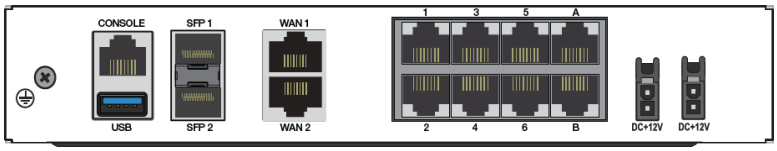
To set up FortiGate for initial management access:

1. Unpack the FortiGate box, and locate the following items:
   - FortiGate device
   - Power cable
   - Ethernet cable

   You will also need to provide the following items:
   - Second Ethernet cable. Only one Ethernet cable is provided to connect the FortiGate to a management computer. Locate a second Ethernet cable to connect the FortiGate to a port for internet access.
   - Management computer to access the FortiOS GUI
2. Use the power cable to connect the FortiGate to a power source.
3. Use one Ethernet cable to connect the management port on the FortiGate to a management computer.

   The default interface used for management differs from model to model. On most units with a single dedicated management port, the port is named MGMT. On units with multiple management ports, the names MGMT1 and MGMT2 are used. On units without dedicated management ports, port1 is used for initial management access, and the port can be part of a virtual switch group.

   The following example is for a FortiGate 80F, which uses port1 for initial management access. For information about your FortiGate hardware model, go to FortiGate/FortiOS Hardware Guides.



4. Use a second Ethernet cable to connect the WAN on the FortiGate to an upstream router, switch, or modem with access to the internet.

   On some FortiGate models, dedicated WAN interface(s) labeled WAN1, WAN2, and so on are available. If no dedicated WAN interfaces are present, select an interface of your choice for the WAN connection.

   Internet access is available when the FortiGate model has addressing mode set to DHCP by default on the WAN interface, and the WAN interface is connected to a network with a DHCP server assigning the correct IP and gateway for internet access. If these conditions are not met, then internet access is not available after connecting your WAN interface.

   - For recent FortiGate models, a setup wizard is displayed after you log in to FortiOS to help configure internet access. See Configuring internet access for recent FortiGate models on page 35.
   - For older FortiGate models, see Configuring basic settings on page 46 to configure internet access.
5. On the management computer, assign an address in the 192.168.1.0/24 network.
6. Log in to the FortiOS GUI. See Logging in to FortiOS GUI on page 33 for next steps.

Logging in to FortiOS GUI

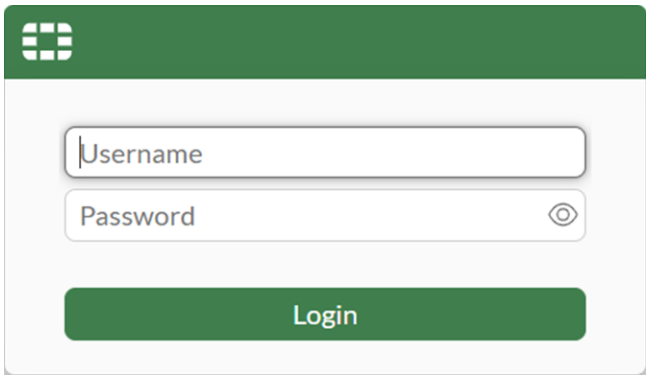
You can log in to the FortiOS GUI after completing FortiGate management access setup. See Setting up FortiGate for management access on page 32 for details.

To log in to FortiOS GUI:

1. In a web browser, go to https://192.168.1.99.

   By default, the management interface or the internal interface is configured to allow HTTPS access with the IP address 192.168.1.99.

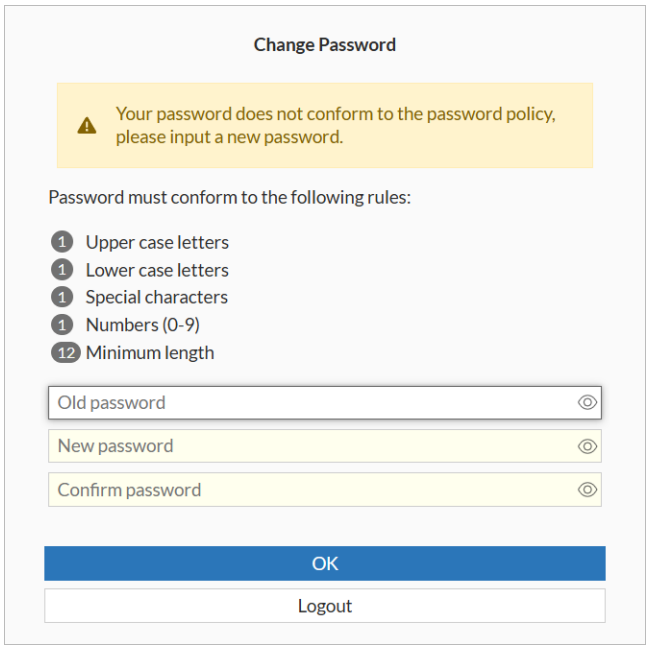
   The login page is displayed in your browser.



2. Log in:
   - In the *Username* field, enter `admin`.
   - leave the *Password* field blank.
   - Click *Login*.

   The *Change Password* dialog box is displayed.

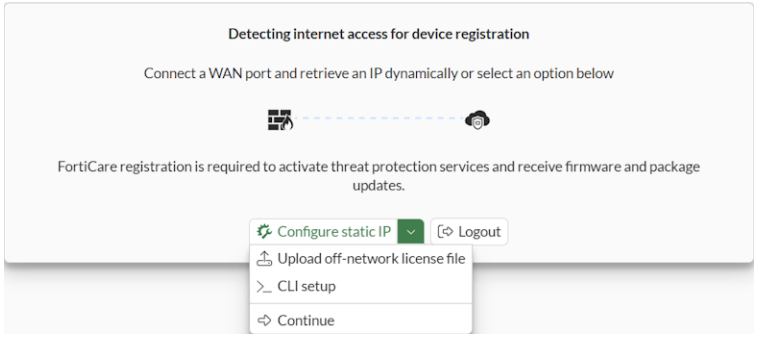
Starting in FortiOS 7.6.5, the password policy is enabled and enforced. You must change the `admin` password to meet the policy requirements. See also Password policy on page 3721.



3. Change the password and click *OK*. The login page is displayed again.
4. Log in again:
   - In the *Username* field, enter `admin`.
   - In the *Password* field, enter your new password.
   - Click *Login*.

After you log in, FortiOS checks for an internet connection, and next steps depend on your FortiGate model:

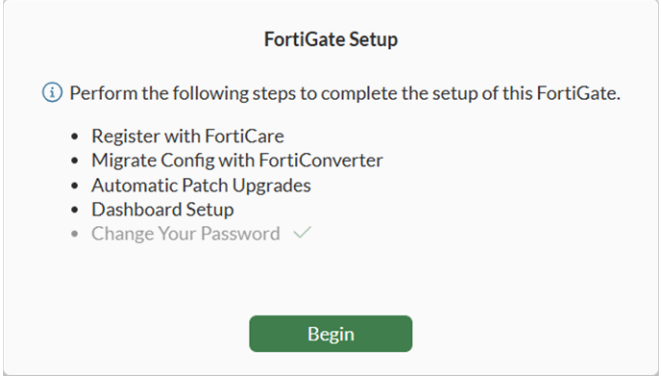
- For recent FortiGate models, the *Detecting internet access for device registration* dialog box is displayed.



When FortiOS detects internet access, the wizard moves automatically to the next step, and you proceed to device registration. See Registering recent FortiGate models on page 39.

When FortiOS does not detect internet access, it remains on the *Detecting internet access for device registration* dialog box, and you can use it to:

    - Configure a WAN interface for internet access. See Configuring internet access for recent FortiGate models on page 35.
    - Upload an off-network license file to FortiGate in an air-gapped environment. See Registering recent FortiGate models in an air-gapped environment on page 41.
    - Postpone device registration to access the seven-day setup period to configure the FortiGate before device registration. See Postponing registration to access the setup period on page 36.
- For older FortiGate models, the *FortiGate Setup* wizard is displayed:
    - When internet access is detected, the *FortiGate Setup* wizard starts on the *Register with FortiCare* step. See Registering FortiGate models on page 42 for next steps.
    - When no internet access is detected, the *FortiGate Setup* wizard starts on the *Migrate Config with FortiConverter* step. See Completing the FortiGate Setup wizard on page 45 for next steps.



Configuring internet access for recent FortiGate models

This section applies to FortiGate models with the latest BIOS version. For older FortiGate models, skip this step and go to Registering FortiGate on page 39.

The *Detecting internet access for device registration* dialog box helps you configure internet access for the FortiGate to prepare for device registration to FortiCare.

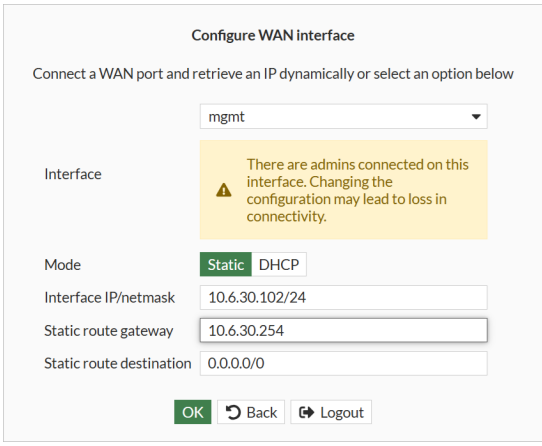
To configure internet access:

1. After logging in to FortiOS for the first time and the while the *Detecting internet access for device registration* dialog box is displayed, use an Ethernet cable to connect the WAN port on the FortiGate to an upstream router, switch, or modem with access to the internet.

   If you have already connected an Ethernet cable, unplug the cable, and plug it in again.

   Internet access is available when the FortiGate model has addressing mode set to DHCP by default on the WAN interface, and the WAN interface is connected to a network with a DHCP server assigning the correct IP and gateway for internet access.
2. If FortiGate does not connect to the internet, configure the WAN interface using the GUI or CLI:

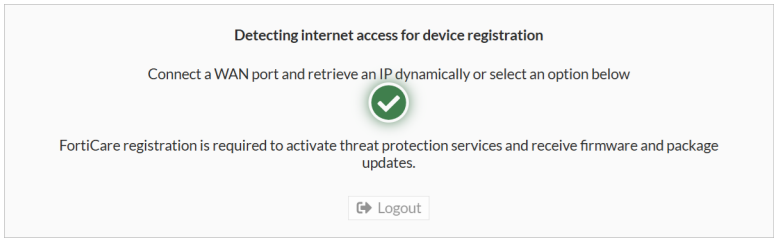
- For the GUI:
  i. Select *Configure WAN interface*. The *Configure WAN interface* dialog box is displayed.



  ii. Configure the options, and click *OK*.
- For the CLI, select *CLI setup* to configure the interface IP address and gateway:

```
config system interface
    edit <interface>
        set mode static
        set ip <address> <netmask>
    next
end
config router static
    edit 0
        set gateway <gateway IP>
        set device <interface>
    next
end
```

When FortiGate connects to the internet, the GUI displays the following page:



Your next step is to register the FortiGate to FortiCare. Go to Registering recent FortiGate models on page 39.

On the Register with FortiCare dialog box, you have the option of postponing registration to access the seven-day setup period. See Postponing registration to access the setup period on page 36.

Postponing registration to access the setup period

This section applies to FortiGate models with the latest BIOS version running FortiOS 8.0.0 or later.

Newer FortiGate models are allowed seven days to configure the FortiGate before device registration to FortiCare is enforced. During the seven-day setup period, administrators can also manually upgrade or downgrade the device using the GUI and CLI. When the seven-day setup period expires, you must register the FortiGate to FortiCare to continue accessing the GUI and CLI.

The seven-day setup period starts when the FortiGate is first powered up and counts down while the device is powered up. Log out of FortiOS and power down FortiGate to pause the countdown. The setup period reflects the cumulative device uptime and ends after seven days or when the FortiGate is registered to FortiCare, whichever occurs first.

The setup period allows, for example, distributors to configure FortiGates before sending the devices to end users for low-touch or zero-touch provisioning. End users can receive FortiGates after the seven-day setup period has expired. In this case, registration to FortiCare will be required at first login.

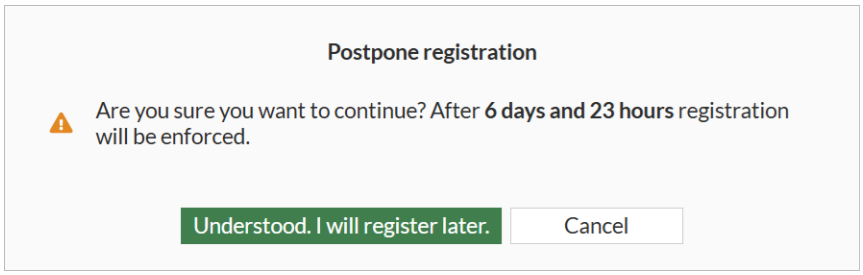
Click the *Postpone registration* button on the following dialog boxes to access the seven-day setup period:

- The *Detecting internet access* for device registration dialog box when FortiOS detects no internet access for FortiGate
- The *Register with FortiCare* dialog box when FortiOS detects internet access for FortiGate and automatically proceeds to this screen

To postpone registration and access the setup period:

1. After logging in to FortiOS for the first time and the *Detecting internet access for device registration* dialog box is displayed, click *Postpone registration*.

   The *Postpone registration* dialog box is displayed. It shows 6 days and 23 hours of FortiGate uptime remain until registration is required.

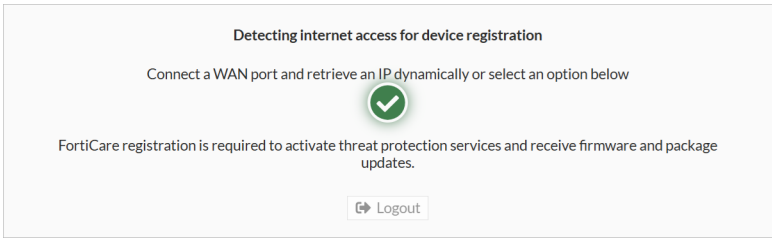
2. Click *Understood, I will register later*.

   The *FortiGate Setup* wizard is displayed.
3. Click *Later* to proceed to the GUI and configure the device.
4. When done configuring the FortiGate, send the device to the end user to configure internet access and register the FortiGate with FortiCare.

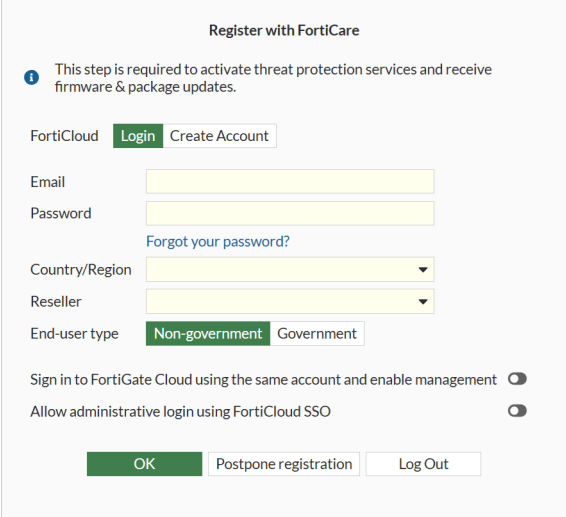
   End users must use their account to register the FortiGate to FortiCare.
5. When the device is not registered or configured for internet after seven days of uptime, the seven-day setup period expires, and the *Postpone registration* button is grayed out.

To postpone registration from the Register with FortiCare dialog box:

1. After logging in to FortiOS for the first time and FortiOS detects an internet connection, a confirmation dialog box is displayed. :



Then the *Register with FortiCare* dialog box is displayed.



2. Postpone registration:
   a. Click *Postpone registration*.
      The *Postpone registration* dialog box is displayed with the remaining setup time.
   b. Click *Understood, I will register later*.
      The *FortiGate Setup* wizard is displayed.
   c. Click *Later* to proceed to the GUI and configure the device.
3. After the seven-day setup period ends, the *Postpone registration* button is no longer available, and you must register with FortiCare or log out.

You can reset the seven-day setup period by performing a factory reset on the FortiGate or restoring the firmware. The seven-day setup period starts again when the FortiGate is powered up.

Registering FortiGate

The FortiGate, and then its service contract, must be registered to have full access to Fortinet Customer Service and Support, and FortiGuard services. The FortiGate can be registered in either the FortiGate GUI or the FortiCloud support portal. The service contract can be registered from the FortiCloud support portal.

The service contract number is needed to complete registrations on the FortiCloud support portal. You can find this 12-digit number in the email that contains your service registration document (sent from do-not-reply-contract@fortinet.com) in the service entitlement summary.

Recent FortiGate models include a new BIOS version with a security enhancement that requires you to register your FortiGate and download the registration information to your FortiGate before you have complete access to the GUI or CLI. You must also register your FortiGate before you can upgrade its firmware.

Although it is recommended to register all FortiGate models to FortiCare, it is not required for older FortiGate models without the latest BIOS version.

This topic contains the following sections:

- Registering recent FortiGate models on page 39

  You can temporarily postpone registration to access the seven-day setup period. See Postponing registration to access the setup period on page 36.
- Registering recent FortiGate models in an air-gapped environment on page 41
- Registering FortiGate models on page 42
- Registering on the FortiCloud support portal on page 43
- Bypassing FortiCare registration on page 44

Registering recent FortiGate models

For recent FortiGate models with the new BIOS version, FortiCare registration is required after new GUI login. When logging in to the FortiOS GUI for the first time on a new FortiGate installation, you are directed to the FortiCare registration dialog box to register your device with FortiCare. You must register the FortiGate to FortiCare before you can access the GUI for further device configuration.

For the list of affected devices, see Registration before device configuration FAQ.

Until the device is registered with FortiCare, the FortiOS CLI is read-only, except for the following commands:

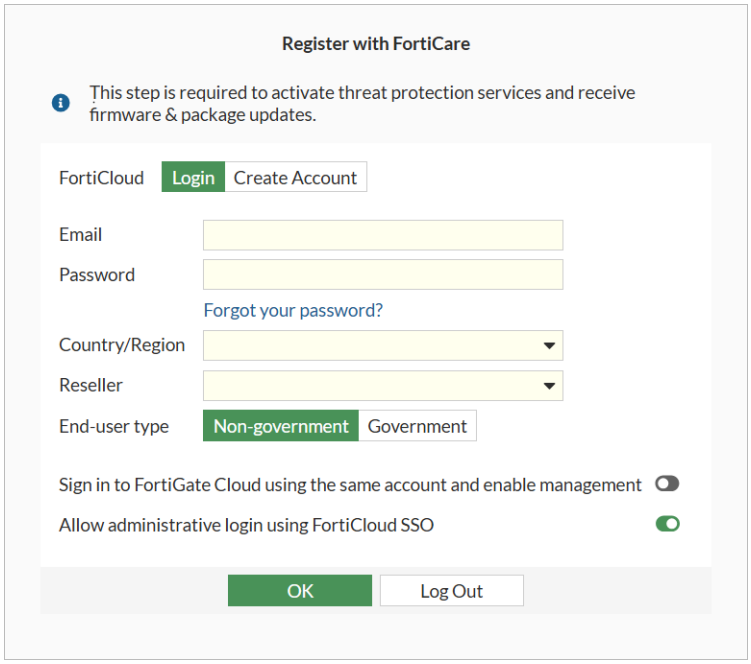
- `config firewall`
  - `policy`: Configure IPv4/IPv6 policies.
- `config ftp-proxy`
- `config router`
  - `setting`: Configure router settings.
  - `static`: Configure IPv4 static routing tables.

    - `static6`: Configure IPv6 static routing tables.
- `config system`
    - `admin`: Configure admin users.
    - `central-management`: Configure central management.
    - `dns`: Configure DNS.
    - `interface`: Configure interfaces.
    - `pppoe-interface`: Configure the PPPoE interfaces.
    - `settings`: Configure VDOM settings.
- `config web-proxy`

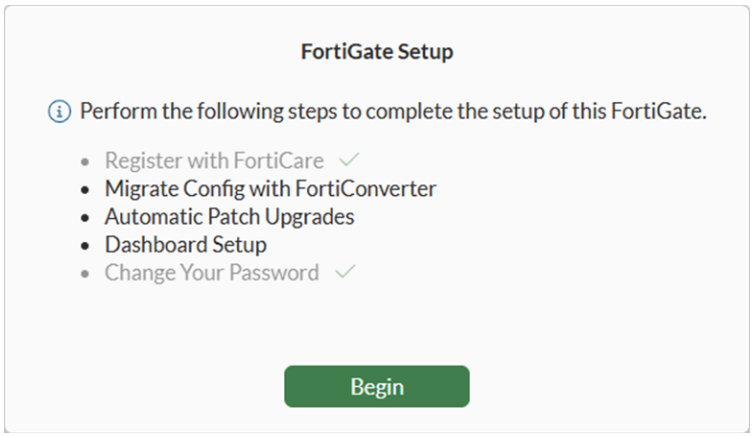
Only a subset of settings related to network configurations are supported.

To register recent FortiGate models with FortiCare:

1. Log in to the FortiOS GUI. The *Register with FortiCare* dialog box is displayed:



2. With *FortiCloud* set to *Login*, complete the fields, and click *OK* to register with FortiCare.
   If you do not want to register the device now, click *Log Out*, and register the device another time. However, you will be unable to access the GUI for further device configuration.
   The *FortiGate Setup* wizard is displayed.



3. Click *Begin* to continue. See Completing the FortiGate Setup wizard on page 45.

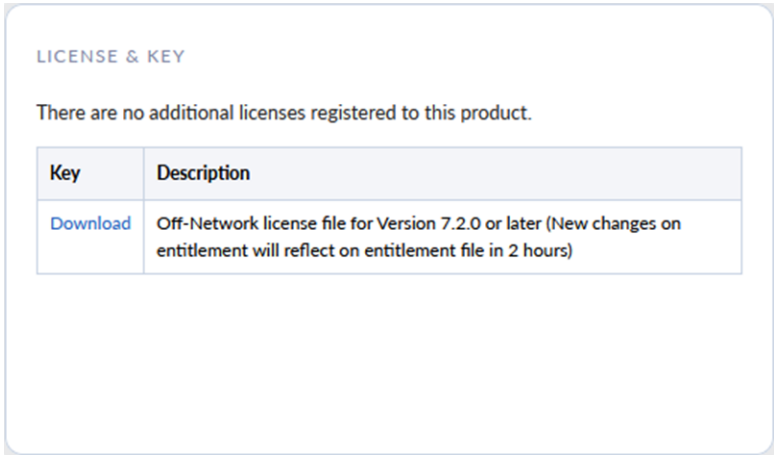
Registering recent FortiGate models in an air-gapped environment

This section applies to FortiGate models with the latest BIOS version.

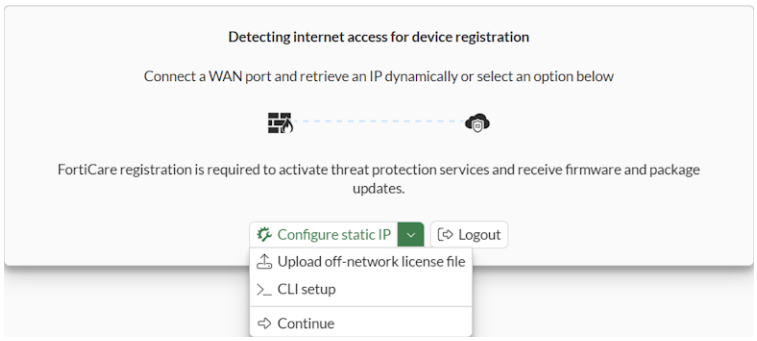
The FortiGate setup wizard can be used to manually upload a previously downloaded license file, allowing successful registration even when the device is unable to reach FortiCare. On FortiCare, register the FortiGate, download the license file, and use the wizard to upload the offline license file to the off-network FortiGate. As long as the FortiGate serial number in the license file matches the serial number of the off-network FortiGate, the device is considered registered. This is particularly useful for air-gapped environments where the lack of internet connectivity is by design.

To register an off-network FortiGate to FortiCare:

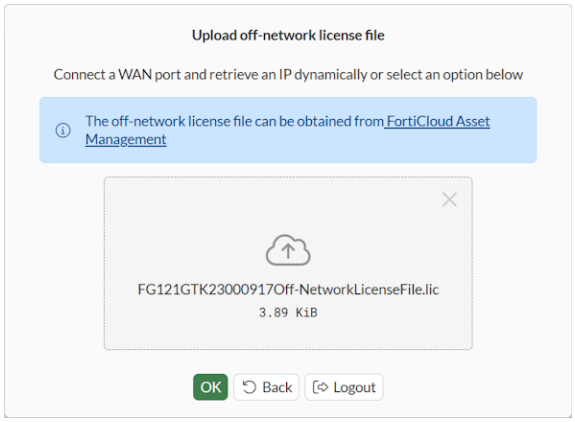
1. Go to FortiCare, log in, and register the FortiGate using the Asset Management portal. See Registering assets.
2. From FortiCare, download the license file. See Viewing licenses and keys.



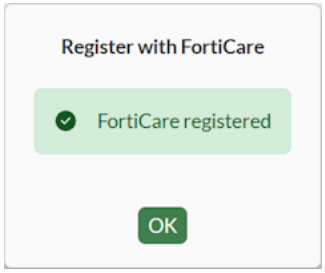
3. On the off-network FortiGate, log in to FortiOS and select *Upload off-network license file*.



4. Upload the off-network license file and click *OK*.



Once the upload is completed and the serial number in the license file matches the serial number of the FortiGate, the following screen is displayed:



5. Click *OK* to continue to proceed to the *FortiGate Setup* wizard. See Completing the FortiGate Setup wizard on page 45.

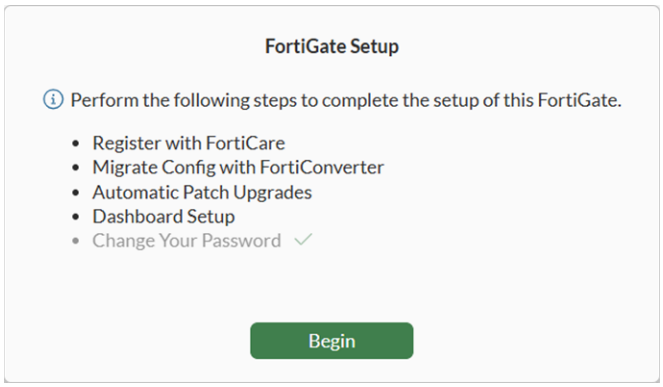
Registering FortiGate models

This section applies to older FortiGate models without the latest BIOS version.

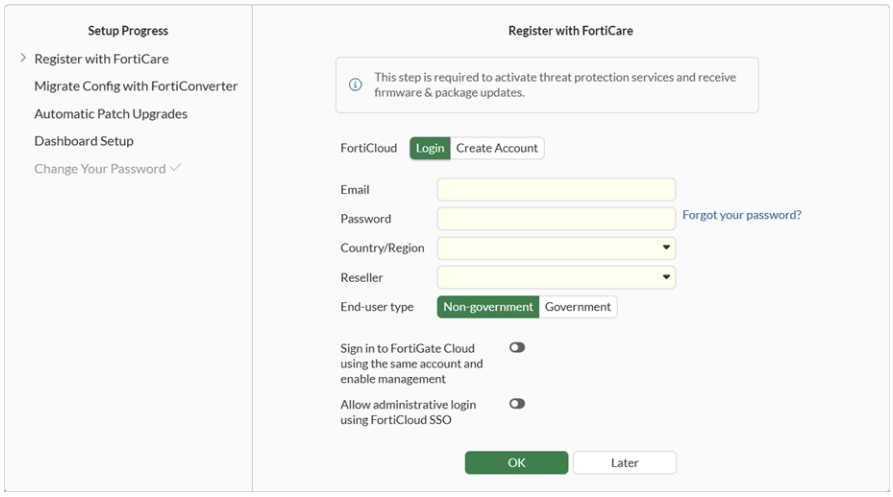
For FortiGate models connected to the internet, you have the option of registering the device with FortiCare using the Setup wizard.

To register FortiGate models in the GUI:

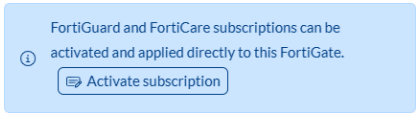
1. Connect to the FortiGate GUI. The FortiGate Setup wizard appears.



2. Click *Begin*. The *Register with FortiCare* step is displayed.



3. Register the FortiGate to FortiCare, and click *OK*. The next step in the wizard is displayed.
   Complete the wizard, or click *Later* to complete the wizard later. See Completing the FortiGate Setup wizard on page 45 for details.
4. Go to *System* > *FortiGuard* and click *Activate subscription*.



   In the CLI use `exec fortiguard-log certificate-activation <code>`.
5. Enter the contract registration code from your service registration document.
6. Click *OK*.

Registering on the FortiCloud support portal

To register the FortiGate on the FortiCloud support portal:

FortiGates can be registered with the Register More button in the Products views. For details, see Registering assets in the FortiCloud Account Services Asset Management guide.

Bypassing FortiCare registration

By default, FortiGates with new BIOS are set to FortiCare registration level 2 in the BIOS, and you must register the device to FortiCare before you can access the GUI. However, you can use the BIOS menu to manually set the FortiCare registration level to 1, and then you can bypass FortiCare registration in the GUI.

To change the FortiCare registration level:

1. Using the console cable, connect the FortiGate unit's console port to the serial communications (COM) port on your management computer. See Connecting to the CLI on page 73.
2. Hard reboot the FortiGate.
3. Press a key to interrupt the boot process when prompted:

```
Boot up, boot device capacity: 28626MB.
Press any key to display configuration menu...
```

4. Enter `I` to access `System configuration and information`.

```
[C]:  Configure TFTP parameters.
[R]:  Review TFTP parameters.
[T]:  Initiate TFTP firmware transfer.
[F]:  Format boot device.
[B]:  Boot with backup firmware and set as default.
[I]:  System configuration and information.
[Q]:  Quit menu and continue to boot.
[H]:  Display this list of options.
```

5. Enter `C` to access `Set FortiCare registration`.

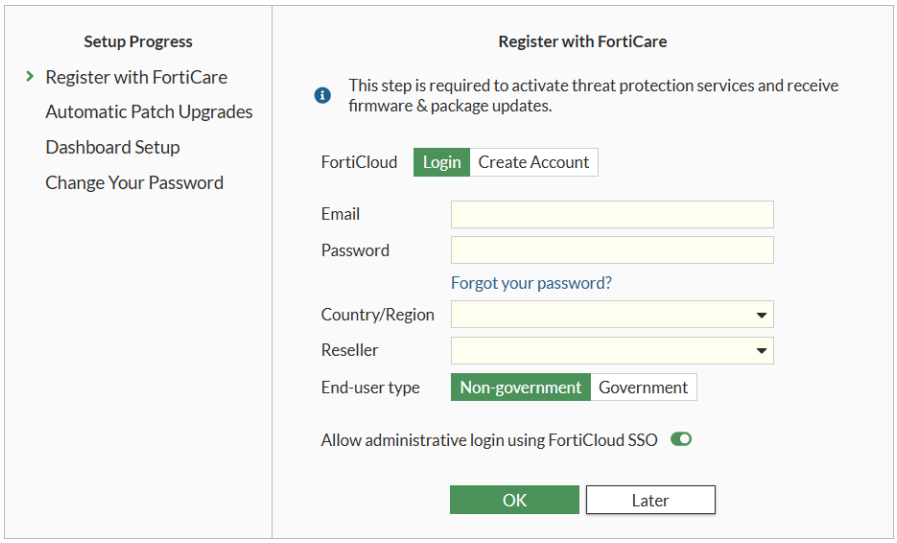
```
Enter C,R,T,F,B,I,Q,or H:
[S]:  Set serial port baudrate (will take effect on next boot).
[R]:  Set restricted mode.
[T]:  Set menu timeout.
[U]:  Set security level.
[C]:  Set FortiCare registration.
[I]:  Display system information.
[E]:  Reset system configuration.
[M]:  Display SPD information.
[Q]:  Quit menu and continue to boot.
[H]:  Display this list of options.
```

6. Enter `1` to set `Not Enforce`.

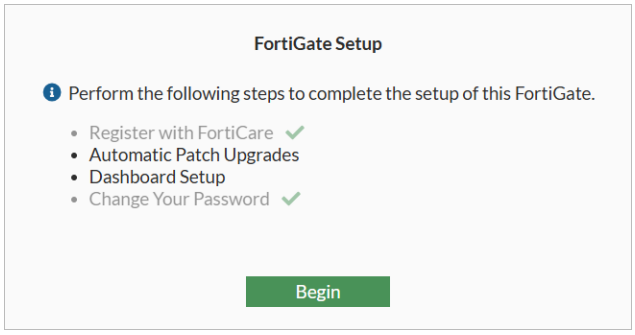
```
Enter S,R,T,U,C,I,E,M,Q,or H:
[1]:  Not Enforce
[2]:  Enforce
Enter FortiCare registration setting [2]: 1
```

To bypass FortiCare registration in the GUI after setting FortiCare registration Level to 1 in the BIOS:

1. Ensure the FortiGate has the correct networking configuration to reach the internet, such as receiving IP and gateway addresses from DHCP. See Configuring basic settings on page 46.
2. Log in to the FortiOS GUI. The *Register with FortiCare* dialog box is displayed:



3. Click *Later* to proceed without registering the FortiGate to FortiCare.
   The *FortiGate Setup* wizard is displayed.



4. Click *Begin* to continue. See also Completing the FortiGate Setup wizard on page 45.

Completing the FortiGate Setup wizard

After logging in to FortiOS, meeting the password policy requirements, and registering the FortiGate to FortiCare (on some FortiGate models this is required), the *FortiGate Setup* wizard displays.

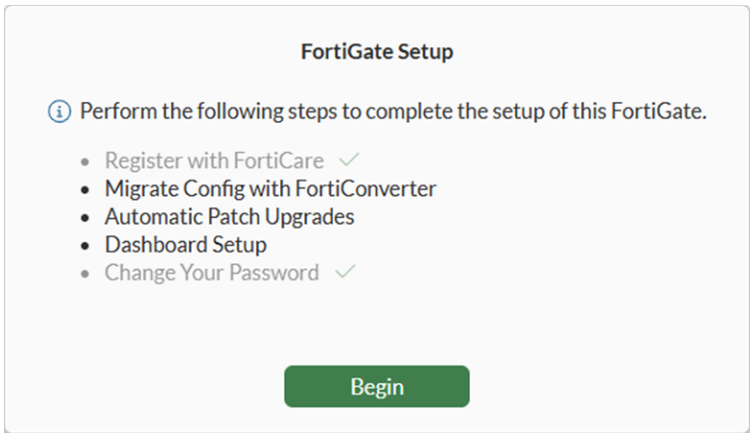
After the *FortiGate Setup* wizard, a FortiOS video is displayed to help you get familiar with the product.

To complete the FortiGate Setup wizard:

1. Connect to the FortiGate GUI. The FortiGate Setup wizard appears and includes some or all of the following:
   a. *Register with FortiCare*
   b. *Migrate Config with FortiConverter*

c. *Automatic Patch Upgrades*
d. *Dashboard Setup*
e. *Change Your Password*

Completed steps are grayed out. For example, the *Register with FortiCare* step and *Change your Password* steps are grayed out when complete. Starting in FortiOS 7.6.5, the password policy is enabled and enforced after first login, and your password must meet the policy requirements. See Password policy on page 3721 for more information.



2. Click *Begin* to start the *FortiGate Setup* wizard.
3. If applicable, register the FortiGate to FortiCare, and click *OK*. See Registering FortiGate on page 39.
   Click *Later* to skip this step and proceed to the next step.
   The *Migrate Config with FortiConverter* page is displayed.
4. Click *Convert* to migrate a configuration from another device to FortiGate using FortiConverter.
   Click *Later* to skip this step and proceed to the next step.
   Enable *Don't show again* to disable this option.
   The *Automatic Patch Upgrades* page is displayed.
5. Decide whether to enable automatic patch upgrades, and click *OK*.
   The *Dashboard Setup* page is displayed.
6. Choose what dashboards to display by default in FortiOS, and click *OK*.
   The *FortiGate Setup* is complete, and the FortiOS GUI is displayed.
7. A FortiOS 8.0 What's new video is presented. Watch the video, and then click *OK* to proceed.

Configuring basic settings

Complete the following basic settings on the FortiGate to get the device up and running, if not completed already:

1. Plan interface usage for MGMT, WAN, and LAN access, and configure the interfaces. See Planning and configuring the MGMT, WAN, and LAN interfaces on page 47.
   Starting in FortiOS 7.6.5, some FortiGate models with two WAN interfaces have a default SD-WAN configuration. See Default SD-WAN configuration on FortiGate models with two WAN ports on page 131 for more information.
2. Configure the default route. See Configuring the default route on page 49.

3. Configure the hostname if not done when completing the *FortiGate Setup* wizard. See Configuring the hostname on page 50.
4. Ensure internet and FortiGuard connectivity. See Ensuring internet and FortiGuard connectivity on page 50.
5. Use the default certificate for HTTPs administrative access. See Using the default certificate for HTTPS administrative access on page 50.

After configuring the basic settings, the FortiGate can access the internet and communicate with FortiGuard. Next, you can register the FortiGate with Fortinet. See Registering FortiGate on page 39. Firewall policies are also ready to be configured using the WAN and LAN interfaces.

Planning and configuring the MGMT, WAN, and LAN interfaces

On a typical deployment where the FortiGate NGFW is configured as an edge firewall, the administrator typically sets up access control between the LAN and WAN interface, and permanent management access either through in-band management or out-of-band management. The following sections outline steps to plan and configure your management, WAN, and LAN interfaces

Management access

So far the new FortiGate setup has been completed over a management interface, which is either a dedicated MGMT port named MGMT or MGMT1 or a port on the internal switch interface.

What interface to use for FortiGate management can depend on the FortiGate model. Some FortiGate models have a dedicated MGMT interface and some do not:

- Mid-size and high-end FortiGate models typically have a dedicated MGMT interface, and you can use the MGMT interface for FortiGate management. There is also a separate management network for accessing the FortiGate and other devices on the network. This is called out-of-band management.
- Desktop FortiGate models typically do not have a dedicated MGMT interface. In this case, you might be using the Internal or LAN interface for FortiGate management. There is no dedicated management network, and the management traffic is shared with internal traffic. This is called in-band management.

Following is a summary of what FortiGate models typically support in-band and out-of-band management:

FortiGate model	MGMT interface	In-band management	Out-of-band management
Desktop models	No	Recommended	Not supported*
Mid-size models	Yes	Supported	Recommended
High-end models	Yes	Supported	Recommended

*Although natively the FortiGate does not support out-of-band management, you can pick an unused interface and configure it as a dedicated interface for out-of-band management.

WAN interface

Similar to the management interface, some models have an interface labelled WAN, WAN1, or WAN2, and other models do not. On models with dedicated WAN interface(s), the interfaces are also configured as DHCP clients.

Therefore, if a DHCP server is present in the WAN network that points to the correct internet gateway, then internet access is available without further configuration.

On models without dedicated WAN interfaces, or in situations where you choose to configure the WAN interface statically, select an interface for WAN access. Connect the interface to your upstream router, L3 switch, or modem. Then use the following steps to configure your WAN interface.

Starting in FortiOS 7.6.5, some FortiGate models with two WAN interfaces have a default SD-WAN configuration. See Default SD-WAN configuration on FortiGate models with two WAN ports on page 131 for more information.

To configure a WAN interface in the GUI:

1. Go to *Network* > *Interfaces*. Select an interface and click *Edit*.
2. (Optional) Enter an *Alias*, such as WAN.
3. In the *Address* section, enter the *IP/Netmask*.
4. In *Administrative Access* section, select the access options as needed. For a WAN interface, it is recommended to only allow *PING*.
5. Click *OK*.

To configure a WAN interface in the CLI:

```
config system interface
    edit "port2"
        set ip 203.0.113.99 255.255.255.0
        set allowaccess ping
        set alias "WAN"
    next
end
```

LAN interface

On desktop and some mid-range models, a set of ports are grouped together by default in virtual switch mode for LAN access. The virtual switch interface may be called internal or lan, and it helps facilitate connecting endpoints directly to the FortiGate on the same L2 switching network.

Endpoints connected this way will also share the same access control configured for the internal or lan interface.

On models that lack a default LAN interface, or when you choose to configure a LAN interface manually, select an interface for LAN access. Connect this interface to an internal switch that connects to your LAN network. Then use the following steps to configure your LAN interface.

To configure a LAN interface in the GUI:

1. Go to *Network* > *Interfaces*. Select an interface and click *Edit*.
2. (Optional) Enter an *Alias*, such as LAN.
3. In the *Address* section, enter the *IP/Netmask*.
4. In *Administrative Access* section, select the access options as needed, such as *PING*. For in-band management, you may also want to allow administrative access for HTTPS and SSH.

5. Optionally, enable *DHCP Server* and configure as needed.
6. Click *OK*.

To configure a LAN interface in the CLI:

```
config system interface
  edit "port1"
    set ip 192.168.10.99 255.255.255.0
    set allowaccess ping https ssh
    set alias "LAN"
  next
end
config system dhcp server
    edit 1
        set dns-service default
        set default-gateway 192.168.10.99
        set netmask 255.255.255.0
        set interface "port1"
        config ip-range
            edit 1
                set start-ip 192.168.10.2
                set end-ip 192.168.10.254
            next
        end
    next
end
```

Configuring the default route

Setting the default route enables the FortiGate to route traffic through this interface and default gateway when no specific routes are found for a particular destination. The gateway address should be your upstream router or L3 switch that the FortiGate is connected to. Set the interface to be the WAN interface that the gateway is connected to.

If the WAN interface uses DHCP for address assignment, the default route may already be learned from the DHCP server, and this step is not needed.

To configure the default route in the GUI:

1. Go to *Network* > *Static Routes* and click *Create New*.
2. Leave the destination subnet as *0.0.0.0/0.0.0.0*. This is known as a default route, since it would match any IPv4 address.
3. Enter the *Gateway Address*.
4. Select an *Interface*.
5. Click *OK*.

To configure the default route in the CLI:

```
config router static
    edit 0
        set gateway 203.0.113.1
        set device port2
    next
end
```

Configuring the hostname

Setting the FortiGate's hostname assists with identifying the device, and it is especially useful when managing multiple FortiGates. Choose a meaningful hostname as it is used in the CLI console, SNMP system name, device name for FortiGate Cloud, and to identify a member of an HA cluster.

To configure the hostname in the GUI:

1. Go to *System* > *Settings*.
2. Enter a name in the *Host name* field.
3. Click *Apply*.

To configure the hostname in the CLI:

```
config system global
    set hostname 200F_YVR
end
```

Ensuring internet and FortiGuard connectivity

This step is not necessary for the configuration; however, it is necessary in order to keep your FortiGate up to date against the latest threats. Updates are provided to FortiGates that are registered and make a request to the FortiGuard network to verify if there are any more recent definitions.

Use `execute ping <domain.tld>` to ensure the DNS resolution is able to resolve the following FortiGuard servers:

- fds1.fortinet.com
- service.fortiguard.net
- update.fortiguard.net

You also need to ensure the necessary ports are permitted outbound in the event your FortiGate is behind a filtering device. Refer to the Ports and Protocols document for more information.

Using the default certificate for HTTPS administrative access

By default, the FortiGate uses the Fortinet_GUI_Server certificate for HTTPS administrative access. Administrators should download the CA certificate and install it on their PC to avoid warnings in their browser.

See Using the default certificate for HTTPS administrative access on page 3819 for more information.

Configuring a firewall policy

When devices are behind FortiGate, you must configure a firewall policy on FortiGate to grant the devices access to the internet. In other words, a firewall policy must be in place for any traffic that passes through a FortiGate.

To create a firewall policy in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Click *Create New*. The *New Policy* pane is displayed.
3. Enter a *Name* and configure the following necessary settings:

Field	Value
Schedule	always
Action	ACCEPT
Incoming Interface	LAN (port1)
Outgoing Interface	WAN (port2)
Source	Source IPv4 address name and address group names
Destination	Destination IPv4 address name and address group names
Service	All

4. Click *Save*.

Backing up the configuration

Once you successfully configure the FortiGate, it is extremely important that you back up the configuration. In some cases, you may need to reset the FortiGate to factory defaults or perform a TFTP upload of the firmware, which will erase the existing configuration. In these instances, the configuration on the device must be recreated, unless a backup can be used to restore it.

You can back up the configuration in FortiOS or YAML format. You have the option to save the configuration file in FortiOS format to various locations including the local PC and USB key.

To back up the configuration in FortiOS format using the GUI:

1. Click on the user name in the upper right-hand corner of the screen and select *Configuration* > *Backup*.
2. Direct the backup to your *Local PC* or to a *USB Disk*.
   The *USB Disk* option will not be available if no USB drive is inserted in the USB port. You can also backup to the FortiManager using the CLI.

3. If VDOMs are enabled, indicate whether the scope of the backup is the entire FortiGate configuration (*Global*) or only a specific VDOM configuration (*VDOM*).

   If backing up a VDOM configuration, select the VDOM name from the list.
4. Enable *Encryption*.

   This is recommended to secure your backup configurations and prevent unauthorized parties from reloading your configuration.
5. Enter a password, and enter it again to confirm it. This password will be required to restore the configuration.
6. Click *OK*.
7. When prompted, select a location on the PC or USB disk to save the configuration file. The configuration file will have a .conf extension.

To back up the configuration in YAML format using the GUI:

1. Click on the user name in the upper right-hand corner of the screen and select *Configuration* > *Backup*.
2. Direct the backup to your *Local PC* or to a *USB Disk*.
3. Select *YAML* for the *File format*.
4. Click *OK*.

Troubleshooting your installation

If your FortiGate does not function as desired after installation, try the following troubleshooting tips:

1. **Check for equipment issues**

   Verify that all network equipment is powered on and operating as expected. Refer to the QuickStart Guide for information about connecting your FortiGate to the network.
2. **Check the physical network connections**

   Check the cables used for all physical connections to ensure that they are fully connected and do not appear damaged, and make sure that each cable connects to the correct device and the correct Ethernet port on that device.
3. **Verify that you can connect to the internal IP address of the FortiGate**

   Connect to the GUI from the FortiGate's internal interface by browsing to its IP address. From the PC, try to ping the internal interface IP address; for example, `ping 192.168.1.99`. If you cannot connect to the internal interface, verify the IP configuration of the PC. If you can ping the interface but can't connect to the GUI, check the settings for administrative access on that interface. Alternatively, use SSH to connect to the CLI, and then confirm that HTTPS has been enabled for Administrative Access on the interface.
4. **Check the FortiGate interface configurations**

   Check the configuration of the FortiGate interface connected to the internal network (under *Network* > *Interfaces*) and check that *Addressing mode* is set to the correct mode.
5. **Verify the security policy configuration**

   Go to *Policy & Objects* > *Firewall Policy* and verify that the internal interface to Internet-facing interface security policy has been added and is located near the top of the policy list. Check the *Active Sessions* column to ensure that traffic has been processed (if this column does not appear, right-click on the table

header and select *Active Sessions*). If you are using NAT mode, check the configuration of the policy to make sure that *NAT* is enabled and that *Use Outgoing Interface Address* is selected.

6. **Verify the static routing configuration**

   Go to *Network* > *Static Routes* and verify that the default route is correct. Go to *Monitor* > *Routing Monitor* and verify that the default route appears in the list as a static route. Along with the default route, you should see two routes shown as *Connected*, one for each connected FortiGate interface.

7. **Verify that you can connect to the Internet-facing interface's IP address**

   Ping the IP address of the Internet-facing interface of your FortiGate. If you cannot connect to the interface, the FortiGate is not allowing sessions from the internal interface to Internet-facing interface. Verify that PING has been enabled for *Administrative Access* on the interface.

8. **Verify that you can connect to the gateway provided by your ISP**

   Ping the default gateway IP address from a PC on the internal network. If you cannot reach the gateway, contact your ISP to verify that you are using the correct gateway.

9. **Verify that you can communicate from the FortiGate to the Internet**

   Access the FortiGate CLI and use the command `execute ping 8.8.8.8`. You can also use the `execute traceroute 8.8.8.8` command to troubleshoot connectivity to the Internet.

10. **Verify the DNS configurations of the FortiGate and the PCs**

    Check for DNS errors by pinging or using traceroute to connect to a domain name; for example: `ping www.fortinet.com`.

    If the name cannot be resolved, the FortiGate or PC cannot connect to a DNS server and you should confirm that the DNS server IP addresses are present and correct.

11. **Confirm that the FortiGate can connect to the FortiGuard network**

    Once the FortiGate is on your network, you should confirm that it can reach the FortiGuard network. First, check the *Licenses* widget to make sure that the status of all FortiGuard services matches the services that you have purchased. Go to *System* > *FortiGuard*, and, in the Filtering section, click *Test Connectivity*. After a few minutes, the GUI should indicate a successful connection. Verify that your FortiGate can resolve and reach FortiGuard at `service.fortiguard.net` by pinging the domain name. If you can reach this service, you can then verify the connection to FortiGuard servers by running the command `diagnose debug rating`. This displays a list of FortiGuard IP gateways you can connect to, as well as the following information:

    - `Weight`: Based on the difference in time zone between the FortiGate and this server
    - `RTT`: Return trip time
    - `Flags`: D (IP returned from DNS), I (Contract server contacted), T (being timed), F (failed)
    - `TZ`: Server time zone
    - `Curr Lost`: Current number of consecutive lost packets
    - `Total Lost`: Total number of lost packets

12. **Use FortiExplorer Go if you cannot connect to the FortiGate over Ethernet**

    If you cannot connect to the FortiGate GUI or CLI, you may be able to connect using FortiExplorer Go. Refer to the FortiExplorer Go User Guide for more details.

13. **Contact Fortinet Support for assistance**

If you require further assistance, visit the Fortinet Support website.

Using the GUI

This section presents an introduction to the graphical user interface (GUI) on your FortiGate.

The following topics are included in this section:

- Connecting using a web browser
- Menus
- Tables
- Entering values
- GUI-based global search
- Loading artifacts from a CDN on page 64
- Accessing additional support resources on page 64
- Command palette on page 65
- Recovering missing graphical components on page 71

For information about using the dashboards, see Dashboard on page 142.

Connecting using a web browser

In order to connect to the GUI using a web browser, an interface must be configured to allow administrative access over HTTPS or over both HTTPS and HTTP. By default, an interface has already been set up that allows HTTPS access with the IP address 192.168.1.99.

Browse to https://192.168.1.99 and enter your username and password. If you have not changed the admin account's password, use the default user name, `admin`, and leave the password field blank.

The GUI will now display in your browser, and you will be required to provide a password for the administrator account.

To use a different interface to access the GUI:

1. Go to *Network* > *Interfaces* and edit the interface you wish to use for access. Take note of its assigned IP address.
2. In *Administrative Access*, select *HTTPS*, and any other protocol you require. You can also select *HTTP*, although this is not recommended as the connection is insecure.
3. Click *OK*.
4. Browse to the IP address using your chosen protocol.
   The GUI will now be displayed in your browser.

Menus

If you believe your FortiGate model supports a menu that does not appear in the GUI, go to *System > Feature Visibility* and ensure the feature is enabled. For more information, see Feature visibility on page 3952.

The GUI contains the following main menus, which provide access to configuration options for most FortiOS features:

Dashboard	The dashboard displays various widgets and monitors that display important system information and allow you to configure some system options. For more information, see Dashboard on page 142.
Network	Options for networking, including configuring system interfaces and routing options. For more information, see Network on page 222.
Policy & Objects	Configure firewall policies, protocol options, and supporting content for policies, including schedules, firewall addresses, and traffic shapers. For more information, see Policy and Objects on page 1841.
Security Profiles	Configure your FortiGate's security features, including Antivirus, Web Filter, and Application Control. For more information, see Security Profiles on page 2212.
VPN	Configure options for IPsec and Agentless virtual private networks (VPNs). For more information, see IPsec VPN on page 2760 and Agentless VPN on page 3349.
User & Authentication	Configure user accounts, groups, and authentication methods, including external authentication and single sign-on (SSO).
WiFi & Switch Controller	Configure the unit to act as a wireless network controller, managing the wireless Access Point (AP) functionality of FortiWiFi and FortiAP units. On certain FortiGate models, this menu has additional features allowing for FortiSwitch units to be managed by the FortiGate. For more information, see Wireless configuration on page 3695 and Switch Controller on page 3696.
System	Configure system settings, such as administrators, HA, FortiGuard, and certificates. For more information, see System on page 3697.
Security Fabric	Access the physical topology, logical topology, automation, and settings of the Fortinet Security Fabric. For more information, see Fortinet Security Fabric on page 4297.
Log & Report	Configure logging and alert email as well as reports. For more information, see Log and Report on page 4750.

Tables

Many GUI pages contain tables of information that can be filtered and customized to display specific information in a specific way. Some tables allow content to be edited directly on that table, or rows to be copied and pasted.

Filters

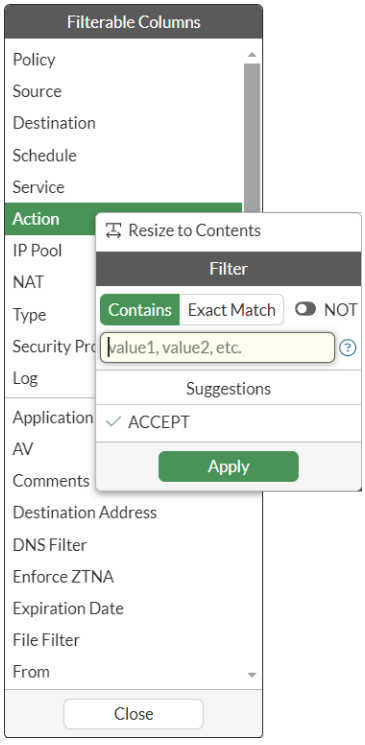
Filters are used to locate a specific set of information or content in a table. They can be particularly useful for locating specific log entries. The filtering options vary, depending on the type of information in the table.

Depending on the table content, you can apply filters using the filter bar, a column filter, or based on a cell's content. Some tables allow filtering based on regular expressions.

Administrators with read and write access can define filters. Multiple filters can be applied at one time.

To manually create a filter:

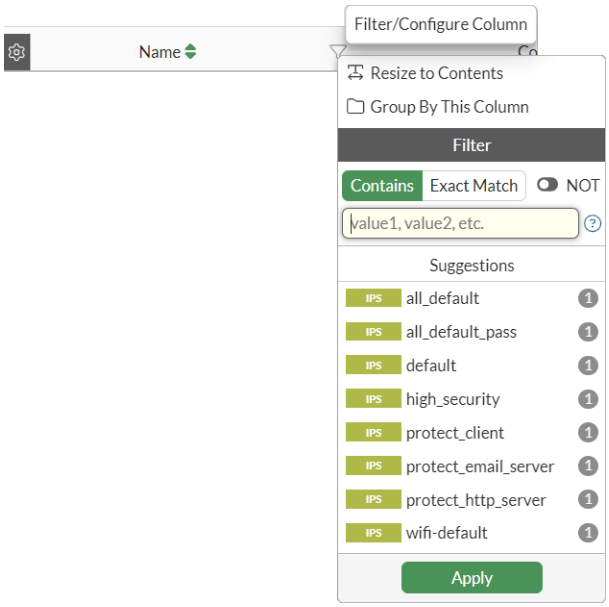
1. Click the add filter button, , in the table search bar. A list of the fields available for filtering is shown.
2. Select the field to filter by.
3. Enter the value to filter by, adding modifiers as needed.



4. Click *Apply*.

To create a column filter:

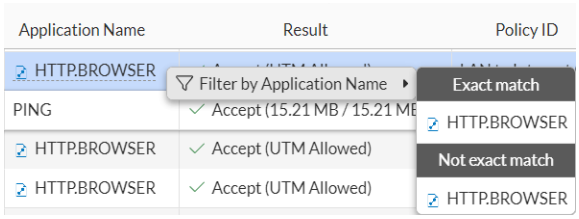
1. Click the filter icon on the right side of the column header.



2. Choose a filter type from the available options.
3. Enter the filter text, or select from the available values.
4. Click *Apply*.

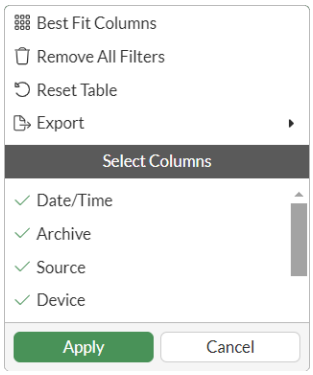
To create a filter based on a cell's content:

1. Right-click a cell in the table.
2. Select *Filter by [column name]* and configure a filtering option from the menu.



To remove all filters:

1. Right-click a column header, or click the gear icon on the left side of the header row that appears when hovering the cursor over the headers.
2. Select *Remove All Filters*.

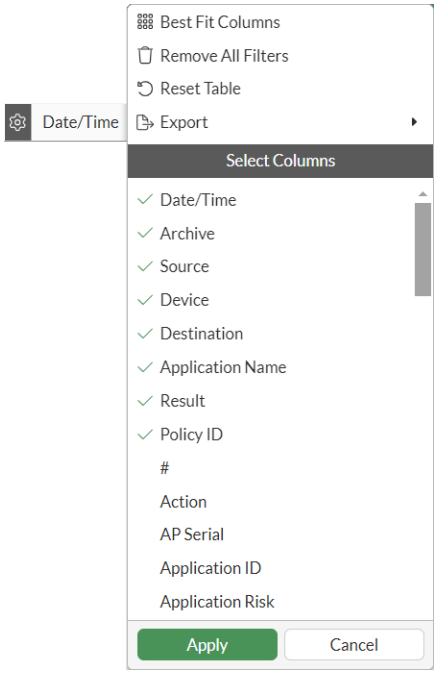


Column settings

Columns can be rearranged, resized, and added or removed from tables.

To add or remove columns:

1. Right-click a column header, or click the gear icon on the left side of the header row that appears when hovering the cursor over the headers.



2. Select columns to add or remove.
3. Click *Apply*.

To rearrange the columns in a table:

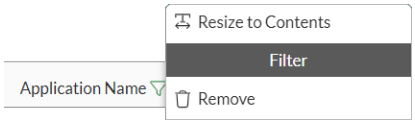
1. Click and drag the column header.

To resize a column:

1. Click and drag the right border of the column header.

To resize a column to fit its contents:

1. Click filter icon on the right side of the column header and select *Resize to Contents*.



To resize all of the columns in a table to fit their content:

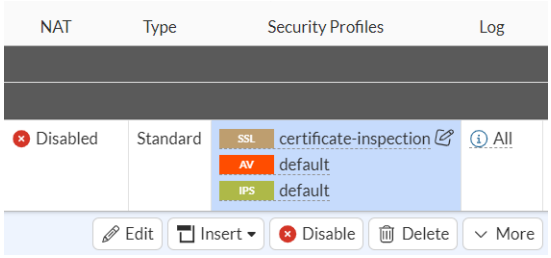
1. Right-click a column header, or click the gear icon on the left side of the header row that appears when hovering the cursor over the headers.
2. Click *Best Fit Columns*.

To reset a table to its default view:

1. Right-click a column header, or click the gear icon on the left side of the header row that appears when hovering the cursor over the headers.
2. Click *Reset Table*.

Editing objects

In some tables, parts of a configuration can be edited directly in the table. For example, you can add security profiles to an existing firewall policy. For the firewall policy, click the *Security Profiles* cell to display, and then click the edit icon.

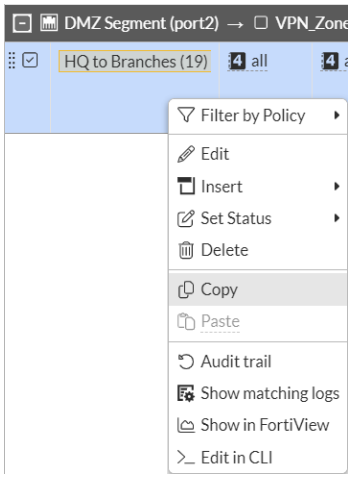


Copying rows

In some tables, rows can be copied and pasted using the right-click menu. For example, a policy can be duplicated.

To copy a row:

1. Right-click the policy, and select *Copy*.



2. Right-click the policy, and select *Paste* > *Above* or *Paste* > *Below*.

Downloading tables

For applicable tables, the content can be downloaded as a CSV or JSON file.

To download the table contents:

1. Click the gear icon on the left side of the header row that appears when hovering the cursor over the headers.
2. Select *Export* > *CSV* or *Export* > *JSON*.
3. Choose the program to open the file with, or save the file to the management computer.

Entering values

Numerous fields in the GUI and CLI require text strings or numbers to be entered when configuring the FortiGate. When entering values in the GUI, you will be prevented from entering invalid characters, and a warning message will be shown explaining what values are not allowed. If invalid values are entered in a CLI command, the setting will be rejected when you apply it.

- Text strings on page 60
- Numbers on page 62

Text strings

Text strings are used to name entities in the FortiGate configuration. For example, the name of a firewall address, administrator, or interface are all text strings.

The following characters cannot be used in text strings, as they present cross-site scripting (XSS) vulnerabilities:

- " - double quotes
- ' - single quote
- > - greater than
- < - less than

Most GUI text fields prevent XSS vulnerable characters from being added.

VDOM names and hostnames can only use numbers (0-9), letters (a-z and A-Z), dashes, and underscores.

The `tree` CLI command can be used to view the number of characters allowed in a name field. For example, entering the following commands show that a firewall address name can contain up to 79 characters, while its FQDN can contain 255 characters:

```
# tree firewall address
-- [address] --*name    (79)
             |- uuid
             |- subnet
             |- type
             |- route-tag       (0,4294967295)
             |- sub-type
             |- clearpass-spt
             |- [macaddr] --*macaddr    (127)
             |- start-ip
             |- end-ip
             |- fqdn    (255)
             |- country         (2)
             |- wildcard-fqdn   (255)
             |- cache-ttl       (0,86400)
             |- wildcard
             |- sdn     (35)
             |- [fsso-group] --*name    (511)
             |- interface       (35)
             |- tenant  (35)
             |- organization    (35)
             |- epg-name        (255)
             |- subnet-name     (255)
             |- sdn-tag         (15)
             |- policy-group    (15)
             |- obj-tag         (255)
             |- obj-type
             |- tag-detection-level     (15)
             |- tag-type        (63)
             |- dirty
             |- hw-vendor       (35)
             |- hw-model        (35)
             |- os      (35)
             |- sw-version      (35)
             |- comment
```

```
            |- associated-interface    (35)
            |- color   (0,32)
            |- filter
            |- sdn-addr-type
            |- node-ip-only
            |- obj-id
            |- [list] --*ip    (35)
                      |- obj-id        (127)
                      +- net-id        (127)
            |- [tagging] --*name       (63)
                         |- category   (63)
                         +- [tags] --*name     (79)
            |- allow-routing
            +- fabric-object
```

Numbers

Numbers are used to set sizes, rates, addresses, port numbers, priorities, and other such numeric values. They can be entered as a series of digits (without commas or spaces), in a dotted decimal format (such as IP addresses), or separated by colons (such as MAC addresses). Most numeric values use base 10 numbers, while some use hexadecimal values.

Most GUI and CLI fields prevent invalid numbers from being entered. The CLI help text includes information about the range of values allowed for applicable settings.

GUI-based global search

The global search option in the GUI allows users to search for keywords appearing in objects and navigation menus to quickly access the object and configuration page. Click the magnifying glass icon in the top-left corner of the banner to access the global search.

The global search includes the following features:

- Keep a history of frequent and recent searches
- Sort results by relevance (by search weight), or alphabetically in increasing or decreasing order
- Search by category
- Search in Security Fabric members (accessed by the Security Fabric members dropdown menu in the banner)
- Search for dashboard widgets and monitors, and preview the widget or go directly to the monitor dashboard if it exists.

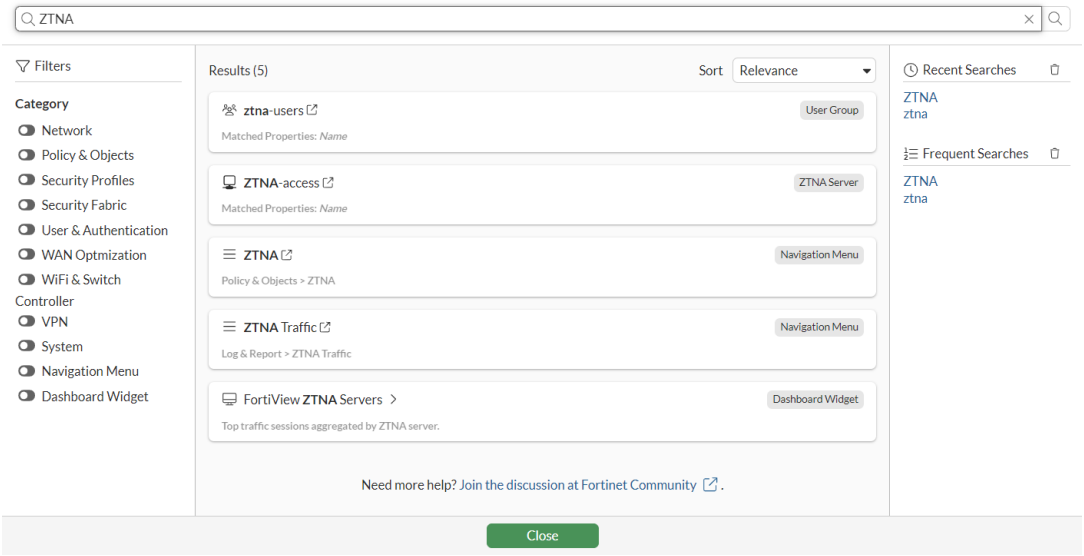
Examples

In this example, searching for the word *ZTNA* yields the following results:

- A ZTNA server, user group, and SAML SSO server that have the *ZTNA* in the *Name* field.
- Various ZTNA tag.

- ZTNA navigation tree items: *Policy & Objects* > *ZTNA* and *Log & Report* > *ZTNA Traffic*.
- The FortiView ZTNA Servers dashboard widget.

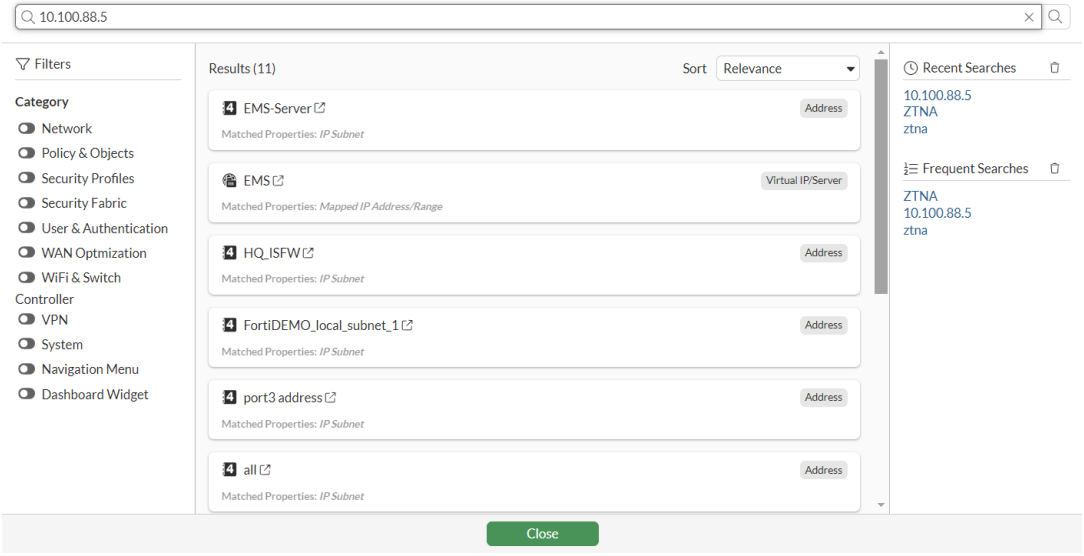
CMDB objects have a higher search weight (50) than navigation objects (20), so the navigation menus and widgets appears at the bottom of the results when sorting by relevance.



In this example, searching for the address *10.100.88.5* yields the following results:

- Various address objects that have a subnet of 10.100.88.5.
- A Virtual IP/Server object, *EMS*, that has a mapped IP address/range with 10.100.88.5.
- Address objects that have IP subnets of 0.0.0.0/0, which the search term falls into.
- Address group objects that contains members addresses that have IP subnets of 0.0.0.0/0.

Sorting by *Relevance* displays address objects that are more closely matched at the top (10.100.88.5), and more loosely matched at the bottom (0.0.0.0).



Loading artifacts from a CDN

To improve GUI performance, loading static GUI artifacts cached in CDN (content delivery network) servers closer to the user instead of the FortiGate can be enabled. This allows the GUI to load more quickly with less latency for administrators who are accessing the FortiGate remotely. Upon failure, the files fall back to loading from the FortiGate. The CDN is only used after successful administrator logins.

To configure loading static GUI files from a CDN:

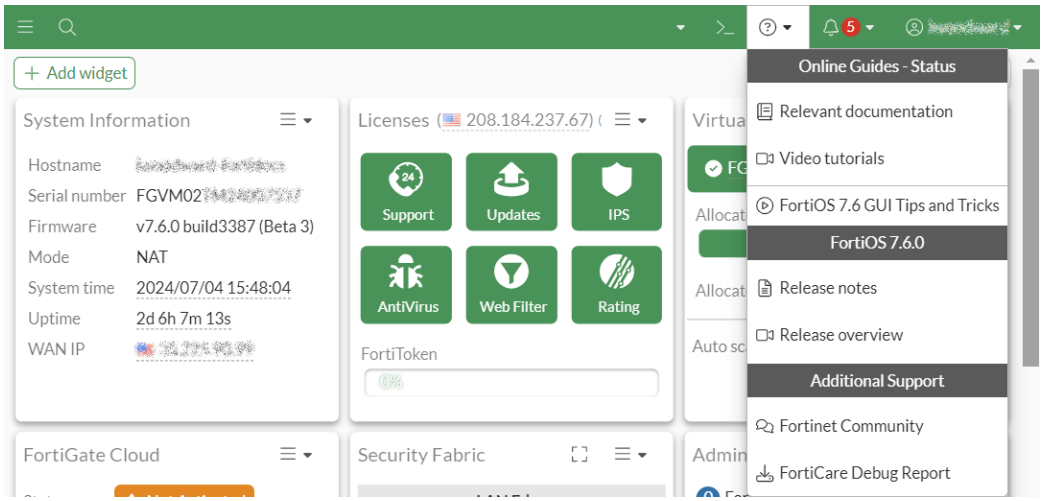
```
config system global
    set gui-cdn-usage {enable | disable}
end
```

Accessing additional support resources

Additional support resources can be accessed from the GUI to troubleshoot issues and get the most out of FortiOS. Online guides, FortiOS documentation, and additional support can now be accessed straight from the help menu.

To access support resources:

1. Click *Help* in the top menu. A dropdown menu is displayed.

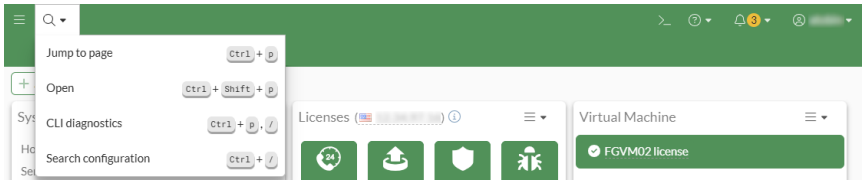


2. Select the support resource you are looking for:
   - *Online Guides* lists resources for help documentation and videos.
   - *FortiOS <version>* contains release information.
   - *Additional Support* contains a link to Fortinet Community and access to download the *FortiCare Debug Report*.

Command palette

The command palette is a keyboard shortcut menu that can be used to quickly navigate to GUI pages, run specific actions like opening the CLI console and restoring a system configuration, and running troubleshooting or debug commands like `get system status` and `show arp table`.

The enhanced global search in the top header menu provides quick command palette access from the GUI and additional keyboard shortcuts.

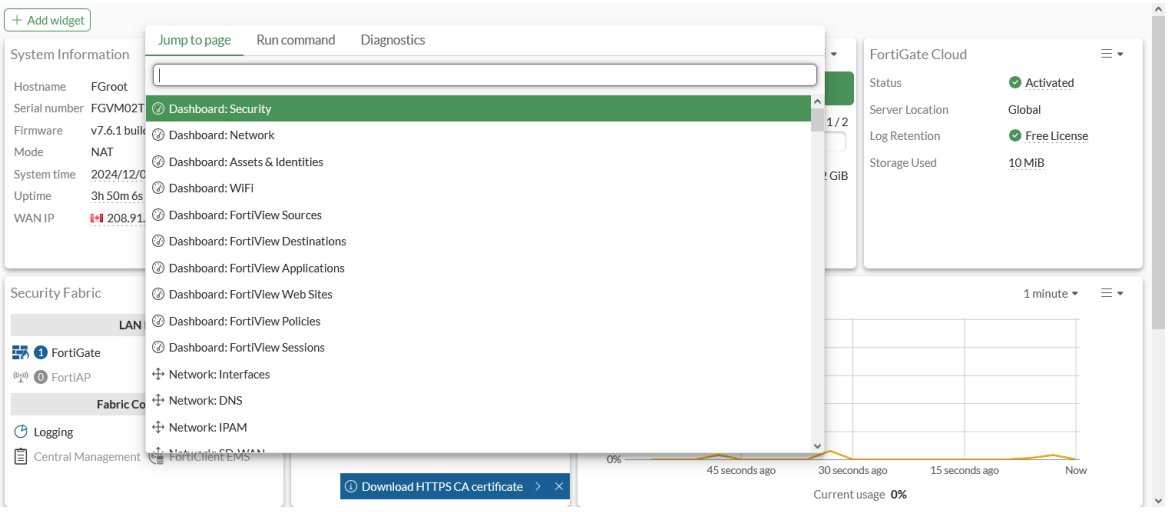


The following global search functions can be selected from the top header menu or by using keyboard shortcuts:

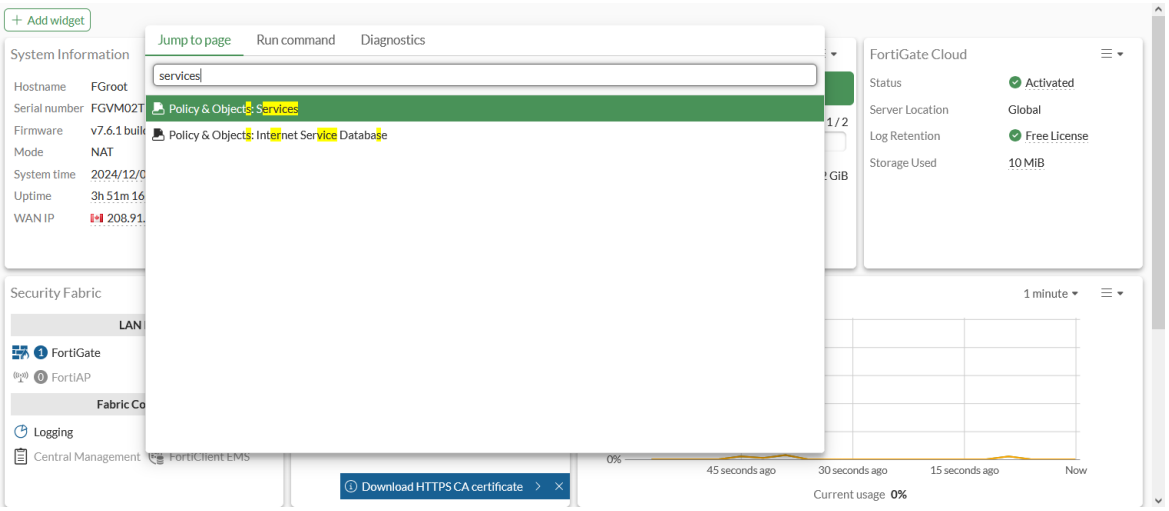
Search function	Keyboard shortcut	Definition
Jump to page	`ctrl + p`	Quickly navigate to a new GUI page.
Open	`ctrl + shift + p`	Open a monitor in a prompt without changing the current GUI page, such as packet capture and the CLI console.
CLI diagnostics	`ctrl + p` followed by `/`	Run a diagnostic CLI command without the CLI console. Recently used commands are listed first. Once a command has been selected, you can search within the output, copy the output, download an output file, and run the command again.
Search configuration	`ctrl + /`	Search for and open a specific configuration in the GUI.

To navigate to a new GUI page using the command palette:

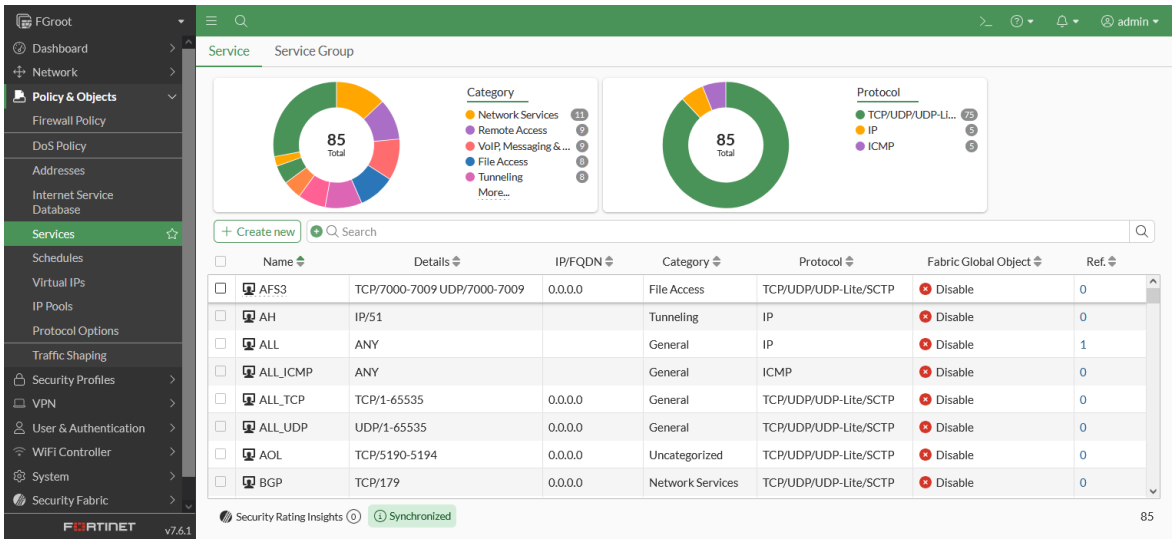
1. Press `ctrl+p` (or `cmd+p` for Mac). The command palette is displayed with available navigation links.



2. Enter the required destination, or browse the list for it.



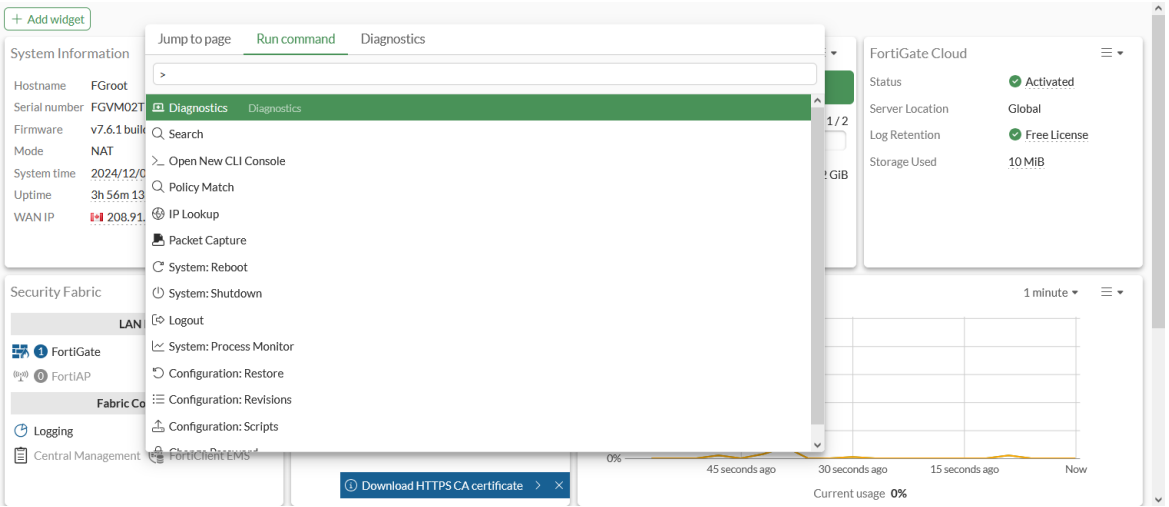
3. Click on the location to jump to the select GUI page.



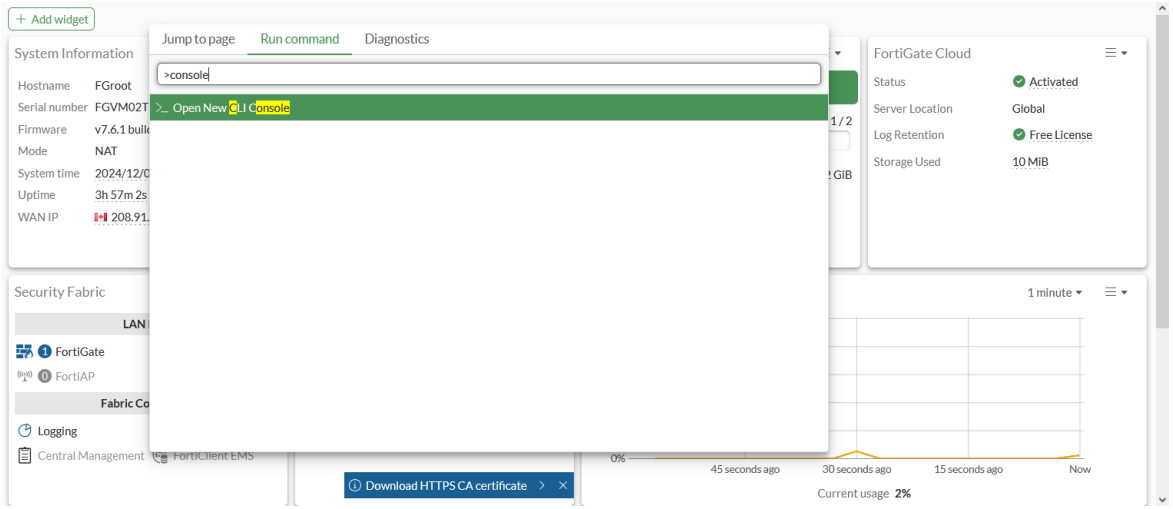
To activate an action using the command palette:

1. Press `ctrl+p` (or `cmd+p` for Mac) and then enter a `>`, or select the *Run command* tab. On supported browsers, `ctrl+shift+p` (or `cmd+shift+p` for Mac) can be used.

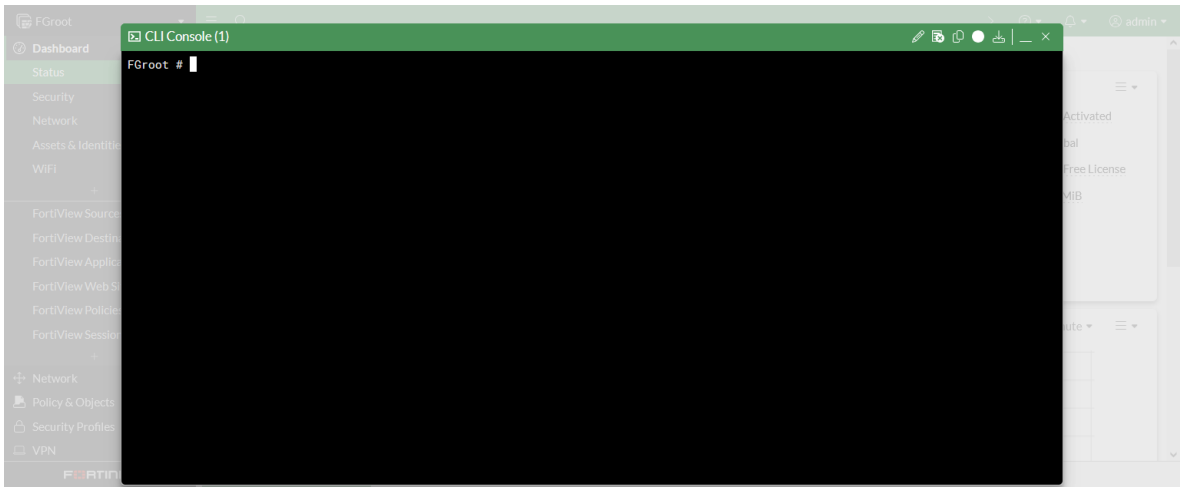
   The command palette is displayed with a runnable command list.



2. Enter the command key word, or browse the list for the command.

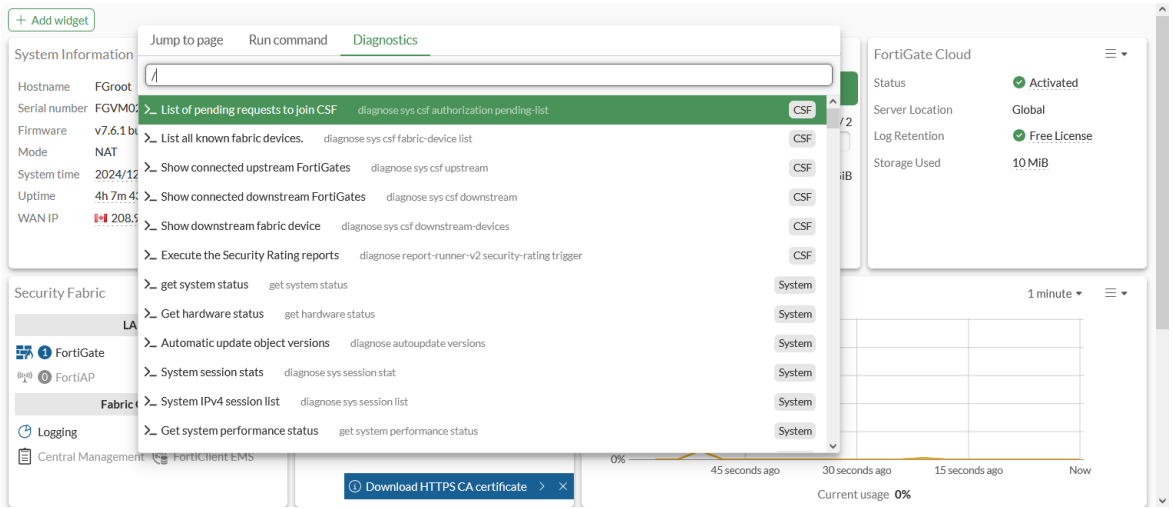


3. Click the result, or press `Enter`, to run the action.

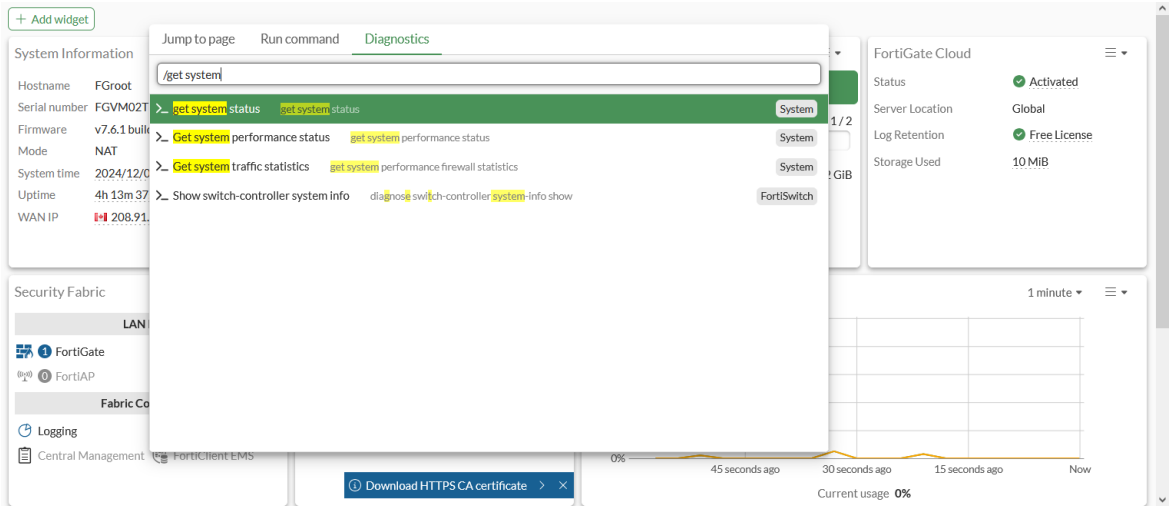


To use the Diagnostics command palette:

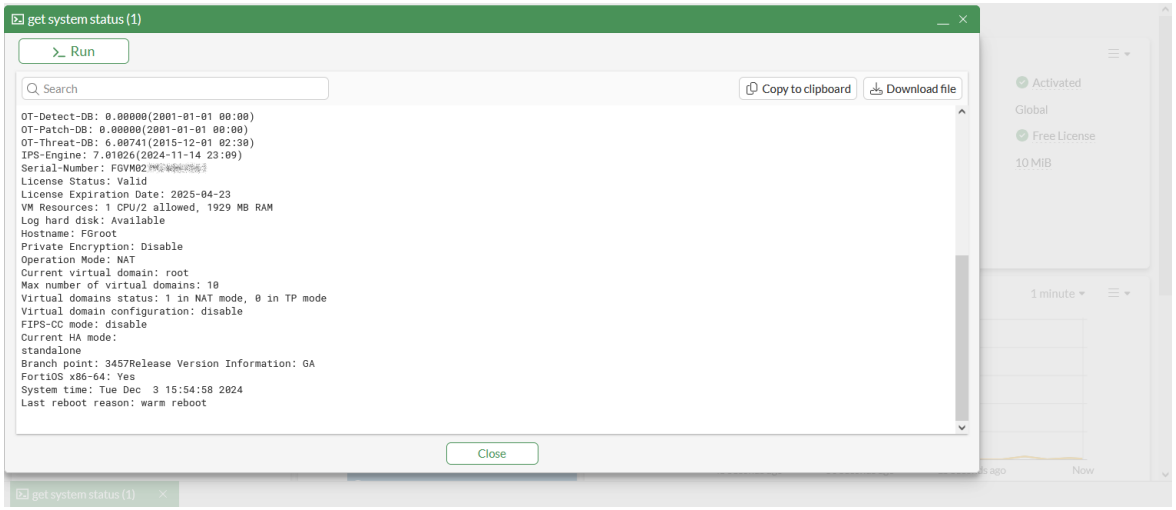
1. Press `ctrl+p` (or `cmd+p` for Mac) and then enter a `/`, or select the *Diagnostics* tab.



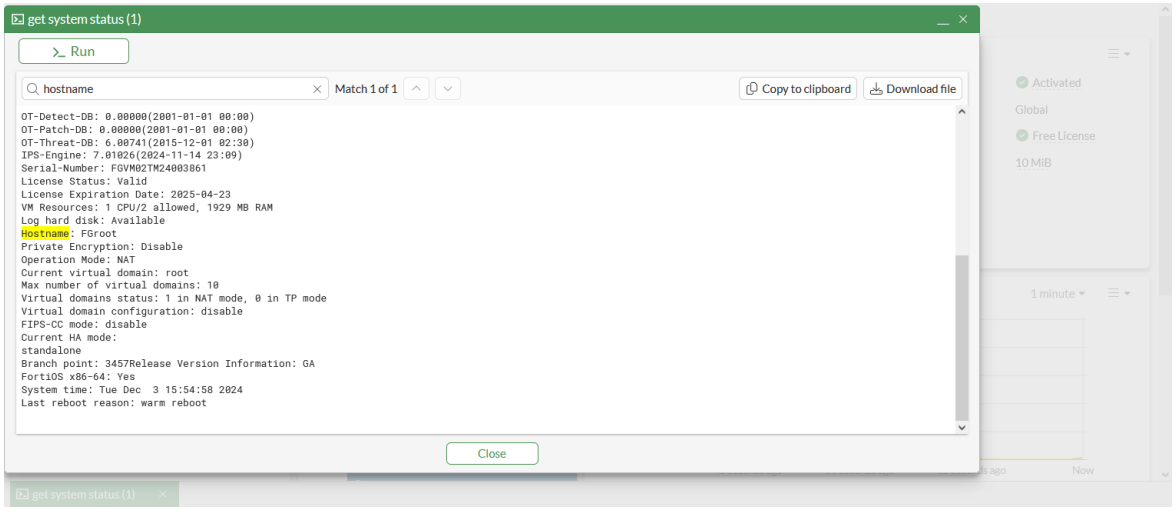
2. Enter the command name, or browse the list for the command.



3. Click the result, or press `Enter`, to run the command.



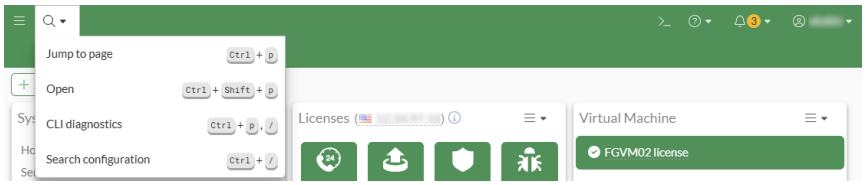
4. Enter a search string in the search field to search the command output.



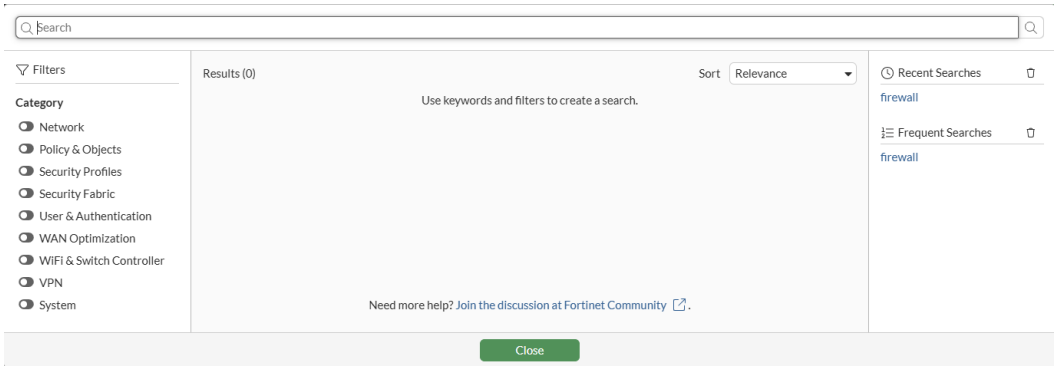
5. Click *Copy to clipboard* to copy the output to the clipboard, or *Download the file* to download the output as a text file.
6. Click *Run* to run the command again.

To search for a specific configuration:

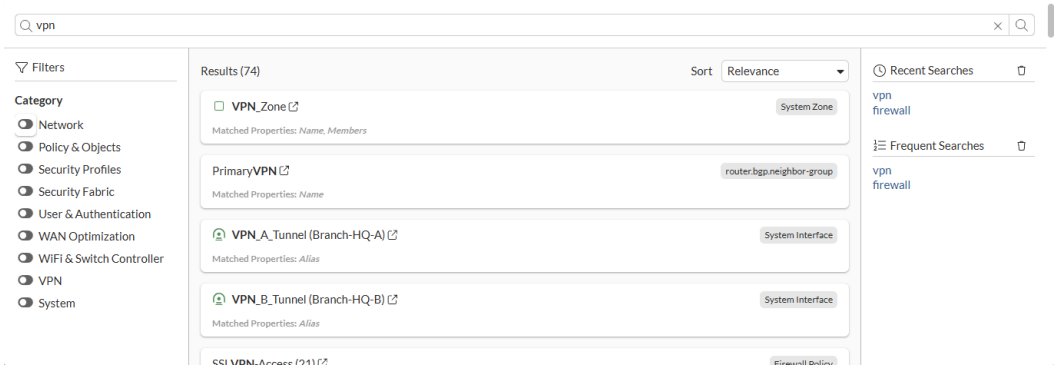
1. Select the global search menu.



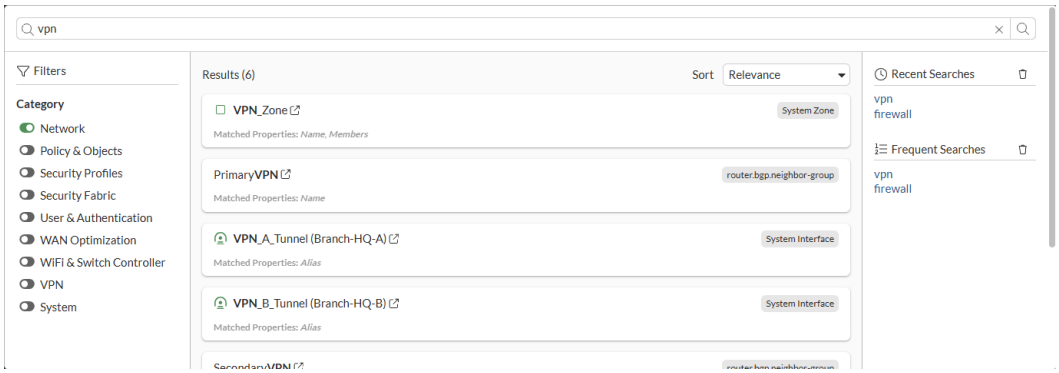
2. Select *Search configuration*.



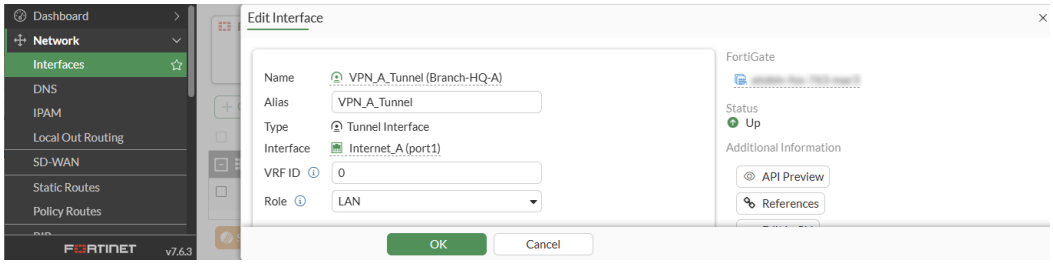
3. Enter the configuration you want and press `Enter`. Suggested results are displayed.



4. Enable categories to filter the results.



5. Select a configuration. The configuration GUI page is displayed.

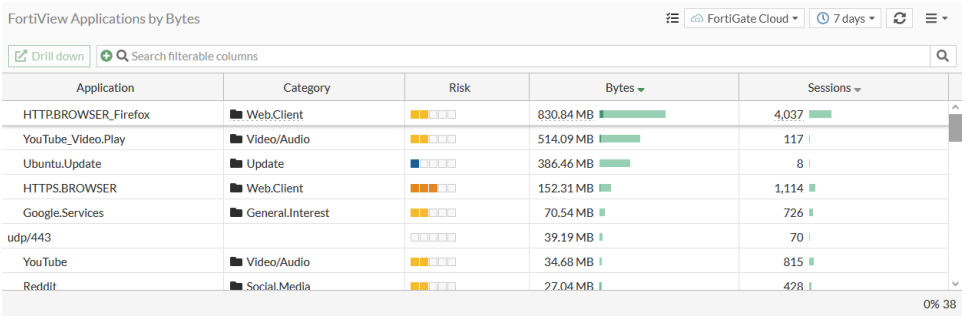


6. Review and edit the configuration as needed.

Recovering missing graphical components

Errors can sometimes cause the application icons, or other minor graphical components, to no longer show up in the GUI.

For example, in the *FortiView Applications* monitor, the icons could be missing from the *Application* column.



The `diagnose fortiguard-resources update` command can be used to delete cached files and force downloads of the FortiGuard resource, including icons.

Command	Downloaded resource
`diagnose fortiguard-resources update sprite-map`	Application icon sprite map (CSS & PNG).
`diagnose fortiguard-resources update sprite-isdb`	Application icon sprite ISDB (CSS & PNG).
`diagnose fortiguard-resources update sprite-casb`	Application icon sprite CASB (CSS & PNG).
`diagnose fortiguard-resources update app-info <id>`	Application info for a given ID.
`diagnose fortiguard-resources update ips-information <id>`	IPS information for a given ID.
`diagnose fortiguard-resources update wf-categories`	Web filter categories.
`diagnose fortiguard-resources update app-categories`	Application categories.
`diagnose fortiguard-resources update prefix-links`	Prefix links.
`diagnose fortiguard-resources update static-links`	Static links.
`diagnose fortiguard-resources update fortigate-end-of-support`	FortiGate product life cycle information.

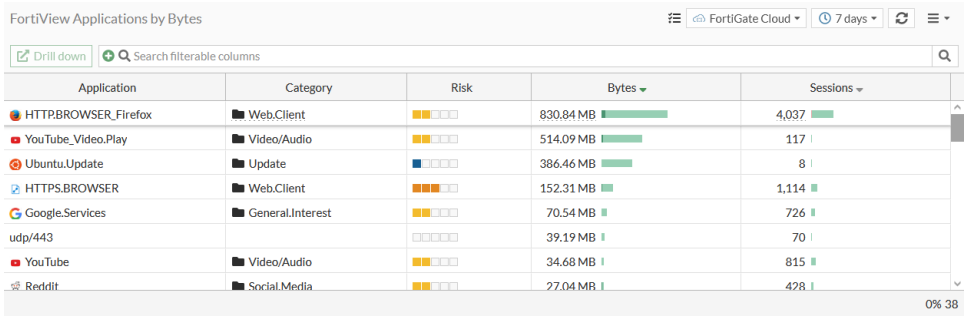
Command	Downloaded resource
`diagnose fortiguard-resources update fortiswitch-end-of-support`	FortiSwitch product life cycle information.
`diagnose fortiguard-resources update fortiap-end-of-support`	FortiAP product life cycle information.
`diagnose fortiguard-resources update fortiextender-end-of-support`	FortiExtender product life cycle information.

To recover missing application icons:

1. Run the update command:

```
# diagnose fortiguard-resources update sprite-isdb
Deleted cached resource file: sprite-isdb.css
Deleted cached resource file: sprite-isdb.png
Deleted cached resource file: sprite_map_front.css
Request URL: "https://globalproductapi.fortinet.net/v1/ref?key=spritemap&f=fos&v=2"
Host "globalproductapi.fortinet.net" resolved to "209.52.38.140"

Performing HTTP request...

Response identified resource location as "https://filestore.fortinet.com/fortiguard/isdb_
logos96/sprite.tar.gz"
Host "filestore.fortinet.com" resolved to "209.52.38.129"

Performing HTTP request...

Successfully downloaded sprite-isdb.css:
  Size: 18142 bytes
  ETag: "5d9814eb50b0a9f8c7b0271e8c5baf39"
  MD5:  a0474459f96edabbc61bfae9b40a9aec
Successfully downloaded sprite-isdb.png:
  Size: 294937 bytes
  ETag: "5d9814eb50b0a9f8c7b0271e8c5baf39"
  MD5:  c98ce9dc8d3c2ae174233798f7124937
```

2. Refresh the browser window. You might also need to clear your browser cache.



Using the CLI

The Command Line Interface (CLI) can be used in lieu of the GUI to configure the FortiGate. Some settings are not available in the GUI, and can only be accessed using the CLI.

This section briefly explains basic CLI usage. For information about the CLI config commands, see the FortiOS CLI Reference.

- Connecting to the CLI on page 73
- CLI basics on page 76
- Command syntax on page 82
- Subcommands on page 85
- Permissions on page 87

Connecting to the CLI

You can connect to the CLI using a direct console connection, SSH, the FortiExplorer Go app, or the CLI console in the GUI.

You can access the CLI outside of the GUI in three ways:

- Console connection: Connect your computer directly to the console port of your FortiGate.
- SSH access: Connect your computer through any network interface attached to one of the network ports on your FortiGate.
- FortiExplorer Go: Connect your device to the FortiExplorer Go app on your device to configure, manage, and monitor your FortiGate. See FortiExplorer Go on page 101 for details.

To open a CLI console, click the _> icon in the top right corner of the GUI. The console opens on top of the GUI. It can be minimized and multiple consoles can be opened. On many GUI pages, the CLI console can be opened with that pages specific commands already shown by clicking *Edit in CLI* in the right-side gutter.

To edit policies and objects directly in the CLI, right-click on the element and select *Edit in CLI*.

Console connection

A direct console connection to the CLI is created by directly connecting your management computer or console to the FortiGate using its DB-9 or RJ-45 console port.

Direct console access to the FortiGate may be required if:

- You are installing the FortiGate for the first time and it is not configured to connect to your network.
- You are restoring the firmware using a boot interrupt. Network access to the CLI will not be available until after the boot process has completed, making direct console access the only option.

To connect to the FortiGate console, you need:

- A console cable to connect the console port on the FortiGate to a communications port on the computer. Depending on your device, this is one of:

  - null modem cable (DB-9 to DB-9)
  - DB-9 to RJ-45 cable (a DB-9-to-USB adapter can be used)
  - USB to RJ-45 cable
- A computer with an available communications port
- Terminal emulation software

To connect to the CLI using a direct console connection:

1. Using the console cable, connect the FortiGate unit's console port to the serial communications (COM) port on your management computer.
2. Start a terminal emulation program on the management computer, select the COM port, and use the following settings:

Bits per second	9600
Data bits	8
Parity	None
Stop bits	1
Flow control	None

3. Press *Enter* on the keyboard to connect to the CLI.
4. Log in to the CLI using your username and password (default: *admin* and no password).
   You can now enter CLI commands, including configuring access to the CLI through SSH.

SSH access

SSH access to the CLI is accomplished by connecting your computer to the FortiGate using one of its network ports. You can either connect directly, using a peer connection between the two, or through any intermediary network.

If you do not want to use an SSH client and you have access to the GUI, you can access the CLI through the network using the CLI console in the GUI.

SSH must be enabled on the network interface that is associated with the physical network port that is used.

If your computer is not connected either directly or through a switch to the FortiGate, you must also configure the FortiGate with a static route to a router that can forward packets from the FortiGate to the computer. This can be done using a local console connection, or in the GUI.

To connect to the FortiGate CLI using SSH, you need:

- A computer with an available serial communications (COM) port and RJ-45 port
- An appropriate console cable
- Terminal emulation software
- A network cable
- Prior configuration of the operating mode, network interface, and static route.

To enable SSH access to the CLI using a local console connection:

1. Using the network cable, connect the FortiGate unit's port either directly to your computer's network port, or to a network through which your computer can reach the FortiGate.
2. Note the number of the physical network port.
3. Using direct console connection, connect and log into the CLI.
4. Enter the following command:

```
config system interface
    edit <interface_str>
        append allowaccess ssh
    next
end
```

   Where `<interface_str>` is the name of the network interface associated with the physical network port, such as `port1`.
5. Confirm the configuration using the following command to show the interface's settings:

```
show system interface <interface_str>
```

   For example:

```
show system interface port1
    config system interface
        edit "port1"
            set vdom "root"
            set ip 192.168.1.99 255.255.255.0
            set allowaccess ping https ssh
            set type hard-switch
            set stp enable
            set role lan
            set snmp-index 6
        next
    end
```

Connecting using SSH

Once the FortiGate is configured to accept SSH connections, use an SSH client on your management computer to connect to the CLI.

The following instructions use PuTTy. The steps may vary in other terminal emulators.

To connect to the CLI using SSH:

1. On your management computer, start PuTTy.
2. In the *Host Name (or IP address)* field, enter the IP address of the network interface that you are connected to and that has SSH access enabled.
3. Set the port number to 22, if it is not set automatically.
4. Select *SSH* for the *Connection type*.
5. Click *Open*. The SSH client connect to the FortiGate.

The SSH client may display a warning if this is the first time that you are connecting to the FortiGate and its SSH key is not yet recognized by the SSH client, or if you previously connected to the FortiGate using a different IP address or SSH key. This is normal if the management computer is connected directly to the FortiGate with no network hosts in between.

6. Click *Yes* to accept the FortiGate's SSH key.
   The CLI displays the log in prompt.
7. Enter a valid administrator account name, such as `admin`, then press *Enter*.
8. Enter the administrator account password, then press *Enter*.
   The CLI console shows the command prompt (FortiGate hostname followed by a #). You can now enter CLI commands.

⚠ If three incorrect log in or password attempts occur in a row, you will be disconnected. If this occurs, wait for one minute, then reconnect and attempt to log in again.

CLI basics

Basic features and characteristics of the CLI environment provide support and ease of use for many CLI tasks.

Help

Press the question mark (?) key to display command help and complete commands.

- Press the question mark (?) key at the command prompt to display a list of the commands available and a description of each command.
- Enter a command followed by a space and press the question mark (?) key to display a list of the options available for that command and a description of each option.
- Enter a command followed by an option and press the question mark (?) key to display a list of additional options available for that command option combination and a description of each option.
- Enter a question mark after entering a portion of a command to see a list of valid complete commands and their descriptions. If there is only one valid command, it will be automatically filled in.

Shortcuts and key commands

Shortcut key	Action
?	List valid complete or subsequent commands. If multiple commands can complete the command, they are listed with their descriptions.
Tab	Complete the word with the next available match. Press multiple times to cycle through available matches.
Up arrow or Ctrl + P	Recall the previous command. Command memory is limited to the current session.

Shortcut key	Action
Down arrow, or Ctrl + N	Recall the next command.
Left or Right arrow	Move the cursor left or right within the command line.
Ctrl + A	Move the cursor to the beginning of the command line.
Ctrl + E	Move the cursor to the end of the command line.
Ctrl + B	Move the cursor backwards one word.
Ctrl + F	Move the cursor forwards one word.
Ctrl + D	Delete the current character.
Ctrl + C	Abort current interactive commands, such as when entering multiple lines. If you are not currently within an interactive command such as `config` or `edit`, this closes the CLI connection.
\ then Enter	Continue typing a command on the next line for a multiline command. For each line that you want to continue, terminate it with a backslash (\ ). To complete the command, enter a space instead of a backslash, and then press *Enter*.

Command tree

Enter `tree` to display the CLI command tree. To capture the full output, connect to your device using a terminal emulation program and capture the output to a log file. For some commands, use the `tree` command to view all available variables and subcommands.

Command abbreviation

You can abbreviate words in the command line to their smallest number of non-ambiguous characters.

For example, the command `get system status` could be abbreviated to `g sy stat`.

Adding and removing options from lists

When configuring a list, the `set` command will remove the previous configuration.

For example, if a user group currently includes members A, B, and C, the command `set member D` will remove members A, B, and C. To avoid removing the existing members from the group, the command `set members A B C D` must be used.

To avoid this issue, the following commands are available:

append	Add an option to an existing list. For example, `append member D` adds user D to the user group without removing any of the existing members.

select	Clear all of the options except for those specified. For example, `select member B` removes all member from the group except for member B.
unselect	Remove an option from an existing list. For example, `unselect member C` removes only member C from the group, without affecting the other members.

Environment variables

The following environment variables are support by the CLI. Variable names are case-sensitive.

$USERFROM	The management access type (`ssh`, `jsconsole`, and so on) and the IPv4 address of the administrator that configured the item.
$USERNAME	The account name of the administrator that configured the item.
$SerialNum	The serial number of the FortiGate.

For example, to set a FortiGate device's host name to its serial number, use the following CLI command:

```
config system global
    set hostname $SerialNum
end
```

Special characters

The following characters cannot be used in most CLI commands: <, >, (,), #, ', and "

If one of those characters, or a space, needs to be entered as part of a string, it can be entered by using a special command, enclosing the entire string in quotes, or preceding it with an escape character (backslash, \).

To enter a question mark (?) or a tab, *Ctrl + V* or *Ctrl + Shift + -* (depending on the method being used to access the CLI) must be entered first.

Question marks and tabs cannot be copied into the CLI Console or some SSH clients. They must be typed in.

Character	Keys
?	Ctrl + V or Ctrl + Shift + - then ?
Tab	Ctrl + V then Tab
Space (as part of a string value, not to end the string)	Enclose the string in single or double quotation marks: `"Security Administrator"` or `'Security Administrator'`. Precede the space with a backslash: `Security\ Administrator`.
'	\'

Character	Keys
(as part of a string value, not to begin or end the string)	
" (as part of a string value, not to begin or end the string)	\"
\	\\

Using grep to filter command output

The `get`, `show`, and `diagnose` commands can produce large amounts of output. The `grep` command can be used to filter the output so that it only shows the required information.

The grep command is based on the standard UNIX grep, used for searching text output based on regular expressions.

For example, the following command displays the MAC address of the internal interface:

```
get hardware nic internal | grep Current_HWaddr
    Current_HWaddr        00:09:0f:cb:c2:75
```

The following command will display all TCP sessions that are in the session list, including the session list line number in the output:

```
get system session list | grep -n tcp
```

The following command will display all of the lines in the HTTP replacement message that contain `URL` or `url`:

```
show system replacemsg http | grep -i url
```

The following options can also be used:

```
-A <num> After
```

```
-B <num> Before
```

```
-C <num> Context
```

The `-f` option is available to support contextual output, in order to show the complete configuration. The following example shows the difference in the output when `-f` is used versus when it is not used:

Without `-f`:

```
show | grep ldap-group1
    edit "ldap-group1"
        set groups "ldap-group1"
```

With `-f`:

```
show | grep -f ldap-group1
    config user group
        edit "ldap-group1"
            set member "pc40-LDAP"
        next
```

```
end
config firewall policy
    edit 2
        set srcintf "port31"
        set dstintf "port32"
        set srcaddr "all"
        set action accept
        set identity-based enable
        set nat enable
        config identity-based-policy
            edit 1
                set schedule "always"
                set groups "ldap-group1"
                set dstaddr "all"
                set service "ALL"
            next
        end
    next
end
```

Language support and regular expressions

Characters such as ñ and é, symbols, and ideographs are sometimes acceptable input. Support varies depending on the type of item that is being configured. CLI commands, objects, field names, and options must use their exact ASCII characters, but some items with arbitrary names or values can be input using your language of choice. To use other languages in those cases, the correct encoding must be used.

Input is stored using Unicode UTF-8 encoding, but is not normalized from other encodings into UTF-8 before it is stored. If your input method encodes some characters differently than in UTF-8, configured items may not display or operate as expected.

Regular expressions are especially impacted. Matching uses the UTF-8 character values. If you enter a regular expression using a different encoding, or if an HTTP client sends a request in a different encoding, matches may not be what is expected.

For example, with Shift-JIS, backslashes could be inadvertently interpreted as the symbol for the Japanese yen (¥), and vice versa. A regular expression intended to match HTTP requests containing monetary values with a yen symbol may not work it if the symbol is entered using the wrong encoding.

For best results:

- use UTF-8 encoding, or
- use only characters whose numerically encoded values are the same in UTF-8, such as the US-ASCII characters that are encoded using the same values in ISO 8859-1, Windows code page 1252, Shift-JIS, and other encoding methods, or
- for regular expressions that must match HTTP requests, use the same encoding as your HTTP clients.

HTTP clients may send requests in encodings other than UTF-8. Encodings usually vary based on the client's operating system or input language. If the client's encoding method cannot be predicted, you might only be able to match the parts of the request that are in English, as the values for English characters tend to be encoded identically, regardless of the encoding method.

If the FortiGate is configured to use an encoding method other than UTF-8, the management computer's language may need to be changed, including the web browse and terminal emulator. If the FortiGate is configured using non-ASCII characters, all the systems that interact with the FortiGate must also support the same encoding method. If possible, the same encoding method should be used throughout the configuration to avoid needing to change the language settings on the management computer.

The GUI and CLI client normally interpret output as encoded using UTF-8. If they do not, configured items may not display correctly. Exceptions include items such as regular expression that may be configured using other encodings to match the encoding of HTTP requests that the FortiGate receives.

To enter non-ASCII characters in a terminal emulator:

1. On the management computer, start the terminal client.
2. Configure the client to send and receive characters using UTF-8 encoding.
   Support for sending and receiving international characters varies by terminal client.
3. Log in to the FortiGate.
4. At the command prompt, type your command and press *Enter*.
   Words that use encoded characters may need to be enclosed in single quotes (`'`).
   Depending on your terminal client's language support, you may need to interpret the characters into character codes before pressing *Enter*. For example, you might need to enter: `edit '\743\601\613\743\601\652'`
5. The CLI displays the command and its output.

Screen paging

By default, the CLI will pause after displaying each page worth of text when a command has multiple pages of output. this can be useful when viewing lengthy outputs that might exceed the buffer of terminal emulator.

When the display pauses and shows `--More--`, you can:

- Press *Enter* to show the next line,
- Press *Q* to stop showing results and return to the command prompt,
- Press an arrow key, *Insert*, *Home*, *Delete*, *End*, *Page Up*, or *Page Down* to show the next few pages,
- Press any other key to show the next page, or
- Wait for about 30 seconds for the console to truncate the output and return to the command prompt.

When pausing the screen is disabled, press *Ctrl + C* to stop the output and log out of the FortiGate.

To disable pausing the CLI output:

```
config system console
    set output standard
end
```

To enable pausing the CLI output:

```
config system console
    set output more
end
```

Editing the configuration file

The FortiGate configuration file can be edited on an external host by backing up the configuration, editing the configuration file, and then restoring the configuration to the FortiGate.

Editing the configuration file can save time is many changes need to be made, particularly if the plain text editor that you are using provides features such as batch changes.

To edit the configuration file:

1. Backup the configuration. See Configuration backups and reset on page 4053 for details.
2. Open the configuration file in a plain text editor that supports UNIX-style line endings.
3. Edit the file as needed.

   > ⚠ Do not edit the first line of the configuration file.
   > This line contains information about the firmware version and FortiGate model. If you change the model number, the FortiGate will reject the configuration when you attempt to restore it.

4. Restore the modified configuration to the FortiGate. See Configuration backups and reset on page 4053 for details.

   The FortiGate downloads the configuration file and checks that the model information is correct. If it is correct, the configuration file is loaded and each line is checked for errors. If a command is invalid, that command is ignored. If the configuration file is valid, the FortiGate restarts and loads the downloaded configuration.

Command syntax

When entering a command, the CLI console requires that you use valid syntax and conform to expected input constraints. It rejects invalid commands. Indentation is used to indicate the levels of nested commands.

Each command line consists of a command word, usually followed by configuration data or a specific item that the command uses or affects.

Notation

Brackets, vertical bars, and spaces are used to denote valid syntax. Constraint notations, such as `<address_ipv4>`, indicate which data types or string patterns are acceptable value input.

All syntax uses the following conventions:

Angle brackets < >	Indicate a variable of the specified data type.
Curly brackets { }	Indicate that a variable or variables are mandatory.
Square brackets [ ]	Indicate that the variable or variables are optional. For example: `show system interface [<name_str>]` To show the settings for all interfaces, you can enter `show system interface` To show the settings for the Port1 interface, you can enter `show system interface port1`.
Vertical bar |	A vertical bar separates alternative, mutually exclusive options. For example: `set protocol {ftp | sftp}` You can enter either `set protocol ftp` or `set protocol sftp`.

Space	A space separates non-mutually exclusive options.
	For example:
	`set allowaccess {ping https ssh snmp http fgfm radius-acct probe-response capwap ftm}`
	You can enter any of the following:
	`set allowaccess ping`
	`set allowaccess https ping ssh`
	`set allowaccess http https snmp ssh ping`
	In most cases, to make changes to lists that contain options separated by spaces, you need to retype the entire list, including all the options that you want to apply and excluding all the options that you want to remove.

Optional values and ranges

Any field that is optional will use square-brackets. The overall config command will still be valid whether or not the option is configured.

Square-brackets can be used is to show that multiple options can be set, even intermixed with ranges. The following example shows a field that can be set to either a specific value or range, or multiple instances:

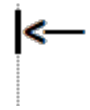
```
config firewall service custom
    set iprange <range1> [<range2> <range3> ...]
end
```

next

The `next` command is used to maintain a hierarchy and flow to CLI commands. It is at the same indentation level as the preceding `edit` command, to mark where a table entry finishes.

The following example shows the next command used in the subcommand `entries`:

```
config dlp filepattern
    edit <1>
        set name <name>
        set comment [comment]
        config entries
            edit <2>
                set filter-type {pattern | type}
            next
```



After configuring table entry <2> then entering next, the <2> table entry is saved and the console returns to the entries prompt:

```
FGT60E1Q23456789 (entries) #
```

You can now create more table entries as needed, or enter end to save the table and return to the filepattern table element prompt.

end

The end command is used to maintain a hierarchy and flow to CLI commands.

The following example shows the same command and subcommand as the next command example, except end has been entered instead of next after the subcommand:

```
config dlp filepattern
    edit <1>
        set name <name>
        set comment [comment]
        config entries
            edit <2>
                set filter-type {pattern | type}
    end
```

Entering end will save the <2> table entry and the table, and exit the entries subcommand entirely. The console returns to the filepattern table element prompt:

```
FGT60E1Q23456789 (1) #
```

Subcommands

Subcommands are available from within the scope of some commands. When you enter a subcommand level, the command prompt changes to indicate the name of the current command scope. For example, after entering:

```
config system admin
```

the command prompt becomes:

```
(admin)#
```

Applicable subcommands are available until you exit the command, or descend an additional level into another subcommand. Subcommand scope is indicated by indentation.

For example, the `edit` subcommand is only available in commands that affects tables, and the `next` subcommand is available only in the `edit` subcommand:

```
config system interface
    edit port1
        set status up
    next
end
```

The available subcommands vary by command. From a command prompt under the `config` command, subcommands that affect tables and fields could be available.

Table subcommands

edit <table_row>	Create or edit a table value.
	In objects such as security policies, `<table_row>` is a sequence number. To create a new table entry without accidentally editing an existing entry, enter `edit 0`. The CLI will confirm that creation of entry 0, but will assign the next unused number when the entry is saved after entering `end` or `next`.
	For example, to create a new firewall policy, enter the following commands:
	`config firewall policy`
	`edit 0`
	`...`
	`next`
	`end`
	To edit an existing policy, enter the following commands:
	`config firewall policy`
	`edit 27`
	`...`
	`next`
	`end`

	The `edit` subcommand changes the command prompt to the name of the table value that is being edited, such as `(27) #`.

delete <table_row>	Delete a table value.
	For example, to delete firewall policy 27, enter the following commands:
	`config firewall policy`
	`  delete 27`
	`end`

purge	Clear all table values. The `purge` command cannot be undone. To restore purged table values, the configuration must be restored from a backup.

move	Move an ordered table value.
	In the firewall policy table, this is equivalent to dragging a policy into a new position. It does not change the policy's ID number.
	For example, to move policy 27 to policy 30, enter the following commands:
	`config firewall policy`
	`  move 27 to 30`
	`end`
	The `move` subcommand is only available in tables where the order of the table entries matters.

clone <table_row> to <table_row>	Make a clone of a table entry.
	For example, to create firewall policy 30 as a clone of policy 27, enter the following commands:
	`config firewall policy`
	`  clone 27 to 30`
	`end`
	The `clone` subcommand may not be available for all tables.

rename <table_row> to <table_row>	Rename a table entry.
	For example to rename an administrator from Fry to Leela, enter the following commands:
	`config system admin`
	`  rename Fry to Leela`
	`end`
	The `rename` subcommand is only available in tables where the entries can be renamed.

get	List the current table entries. For example, to view the existing firewall policy table entries, enter the following commands:

	`config firewall policy` `get`
show	Show the configuration. Only table entries that are not set to default values are shown.
end	Save the configuration and exit the current `config` command.

⚠ Purging the `system interface` or `system admin` tables does not reset default table values. This can result in being unable to connect to or log in to the FortiGate, requiring the FortiGate to be formatted and restored.

Field subcommands

set <field> <value>	Modify the value of a field. For example, the command `set fsso enable` sets the `fsso` field to the value `enable`.
unset	Set the field to its default value.
select	Clear all of the options except for those specified. For example, if a group contains members A, B, C, and D, to remove all members except for B, use the command `select member B`.
unselect	Remove an option from an existing list. For example, if a group contains members A, B, C, and D, to remove only member B, use the command `unselect member B`.
append	Add an option to an existing multi-option table value.
clear	Clear all the options from a multi-option table value.
get	List the configuration of the current table entry, including default and customized values.
show	Show the configuration. Only values that are not set to default values are shown.
next	Save changes to the table entry and exit the `edit` command so that you can configure the next table entry.
abort	Exit the command without saving.
end	Save the configuration and exit the current `config` command.

Permissions

Administrator (or access) profiles control what CLI commands an administrator can access by assigning read, write, or no access to each area of FortiOS. For information, see Administrator profiles on page 3736.

Read access is required to view configurations. Write access is required to make configuration changes. Depending on your account's profile, you may not have access to all CLI commands. To have access to all CLI commands, an administrator account with the *super_admin* profile must be used, such as the *admin* account.

Accounts assigned the *super_admin* profile are similar to the root administrator account. They have full permission to view and change all FortiGate configuration options, including viewing and changing other administrator accounts.

To increase account security, set strong passwords for all administrator accounts and change the passwords regularly. See Default administrator password on page 3810 and Password policy on page 3721 for more information.

FortiAI assistant and CLI Code Lab NEW

FortiAI-Assist is embedded in FortiOS, and FortiOS includes the FortiAI assistant and CLI Code Lab tool to provide RAG-enhanced documentation support, automated diagnostic analysis, and CLI script execution.

Administrators can choose between FortiAI and OpenAI as the AI provider for FortiAI and CLI Code Lab. See Enabling FortiAI and accessing CLI Code Lab on page 90 for details.

With FortiAI and the required subscription, a monthly allotment of tokens is available for use, and additional tokens can be purchased as needed. With OpenAI, administrators can configure FortiAI to work directly with OpenAI for billing and tokens. See Requirements and tokens on page 88 for more information.

Administrator profiles control administrator access to FortiAI. See Access to FortiAI on page 3736.

This section includes the following topics:

- Requirements and tokens on page 88
- Enabling FortiAI and accessing CLI Code Lab on page 90
- FortiAI settings on page 93
- FortiAI examples on page 97

Requirements and tokens

FortiAI-Assist is available on the following FortiGate models with internet access:

FortiGate model	Required subscription	Available tokens
FortiGate hardware with more than 2 GB RAM	FortiCare Premium support	Includes 2,000,000 starter tokens each month per device Additional tokens can be purchased and are shared among all FortiGate devices registered under the same FortiCare account.
FortiGate-VM S-series	FortiCare Premium support or one of the following: • Enterprise bundle • UTP bundle • ATP protection bundle	

FortiGate models with 2 GB RAM or less do not support FortiAI-Assist.

The available tokens and billing depend on the AI provider for FortiAI:

- With FortiAI as the AI provider, a monthly number of tokens is provided per device. Additional tokens can be purchased from Fortinet, and purchased tokens are shared among all FortiGate devices registered under the same FortiCare account. You can view token usage in FortiAI. See Checking token usage for FortiAI on page 94.
- With OpenAI as the AI provider, tokens are purchased from and billed to OpenAI. FortiAI does not provide any information about OpenAI token usage.

How token usage is calculated

Tokens are used in large language models (LLMs) to process text and quantify usage. Token usage is calculated using the following guidelines:

- When you use the FortiAI assistant, the text in both the prompt (input) and the response (output) is processed as tokens.
- While there is not a one-to-one relationship between words or characters and tokens, in general, more text in the query and response means using more tokens.
- Because the FortiAI assistant uses session history to inform its responses, queries that are a part of a long session will use more tokens than new conversations.

Best practices for managing token usage

To ensure you are using your monthly allocation of tokens effectively, consider implementing best practices for FortiAI users. For example:

- Make your prompts concise and specific. In terms of token usage, the prompt `Can you please help me create a firewall address for 10.0.0.1 and another one for the domain awesome-domain.com?` is less effective than `Create firewall addresses for 10.0.0.1 and awesome-domain.com`.
- Use filters in your prompts to receive concise and specific responses. For example, say that you want to create a site-to-site VPN based on an uploaded topology image.
- Use words that relate to functions existing in FortiOS. For example, using `quarantine device` concisely tells the FortiAI assistant what action is required.

- Reference details in the existing thread when possible. This reduces redundancy and allows you to be concise and specific as you build upon previous prompts. However, note that the FortiAI assistant will not remember previous threads.

Enabling FortiAI and accessing CLI Code Lab

FortiAI-Assist supports the following AI providers:

- FortiAI
- OpenAI

Administrators can choose the AI provider for FortiAI-Assist.

The following table summarizes the different options:

AI provider	Details

FortiAI	• Optimized and tested by Fortinet
	• Uses tokens provided by Fortinet
	• Uses the following Fortinet documents for Retrieval-Augmented Generation (RAG) documentation support:
	  • FortiOS Administration Guide
	  • FortiOS CLI Reference
	• Answers technical questions by:
	  • Reading logs directly from FortiGate
	  • Analyzing debug output provided by administrators
	See Enabling FortiAI with FortiAI as AI provider on page 90 and Accessing CLI Code Lab on page 93.

OpenAI	Uses tokens and billing provided by OpenAI. See Enabling FortiAI with OpenAI as AI provider on page 91 and Accessing CLI Code Lab on page 93.

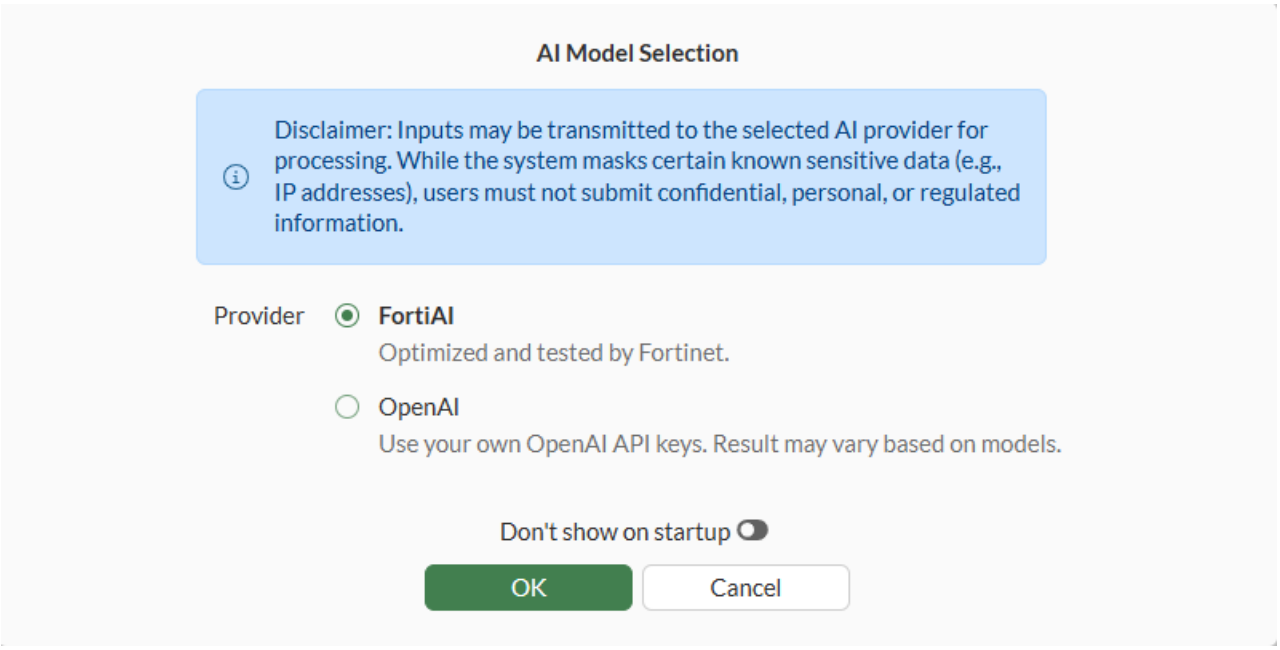
You can also change the AI provider in FortiAI settings. See Selecting an AI model on page 95.

Enabling FortiAI with FortiAI as AI provider

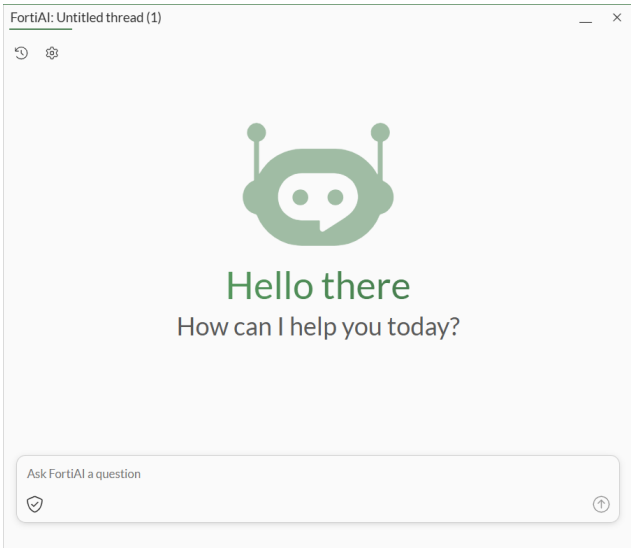
FortiAI can be enabled with FortiAI as the AI provider. This configuration includes a monthly allotment of tokens.

To enable FortiAI in the GUI:

1. In the banner, click the *FortiAI* icon. The *AI Model Selection* dialog box appears.



2. Select *FortiAI* and click *OK*. FortiAI is displayed on the right side of the GUI.

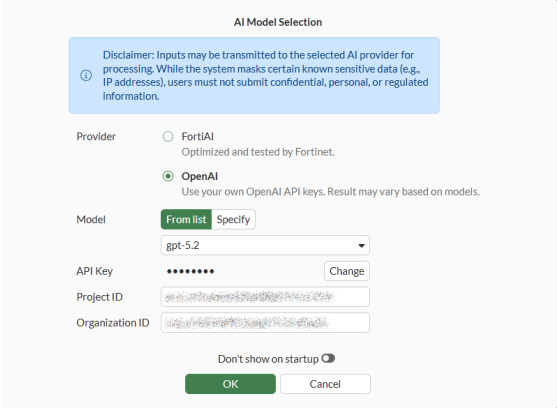


Enabling FortiAI with OpenAI as AI provider

FortiAI can be enabled with OpenAI as the AI provider. This option configures billing with OpenAI for tokens.

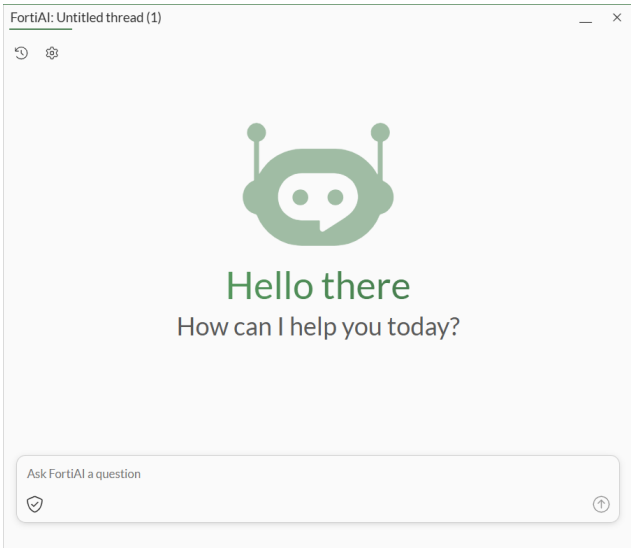
To configure OpenAI as AI provider in the GUI:

1. In the banner, click *FortiAI*. The *AI Model Selection* dialog box appears.
2. Select *OpenAI* and complete the options.



3. Click *OK*. FortiAI is displayed on the right side of the GUI powered by OpenAI instead of FortiAI.

> The GUI always displays FortiAI regardless of the selected LLM.



To configure OpenAI as AI provider in the CLI:

```
config system admin
    edit "admin"
        set vdom "root"
        set accprofile "super_admin"
        set gui-llm-provider openai
        set openai-api-key xxx
        set openai-api-key-part2 yyy
        set openai-model "gpt-5.2"
        set openai-project-id "<project ID>"
        set openai-org-id "<organization ID>"
    next
end
```

Option	Description
`gui-llm-provider {fortiai | openai}`	Select the LLM provider: • `fortiai`: Use FortiAI. • `openai`: Use OpenAI with your own OpenAI key.
`openai-api-key <string>`	Input the OpenAI password value.
`openai-api-key-part2 <string>`	OpenAI API key part 2 for excess password value length.
`openai-model <string>`	OpenAI model name.
`openai-project-id <string>`	OpenAI project ID.
`openai-org-id <string>`	OpenAI organization ID.

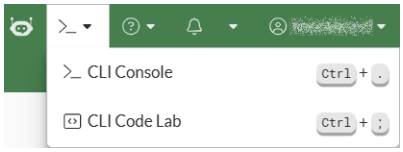
Accessing CLI Code Lab

The CLI Code Lab is available from the FortiOS GUI banner and from FortiAI, and it can be used to generate, edit, and execute CLI commands for configuring FortiGate.

When FortiAI uses FortiAI as the AI provider, the following FortiOS documents are used to provide answers: FortiOS Administration Guide, FortiOS CLI Reference. FortiAI also checks its answers against the FortiGate to ensure the results are suitable for the specific model.

To access CLI Code Lab:

1. Use one of the following methods:
   - From the FortiOS GUI banner, click *CLI Console* > *CLI Code Lab*.

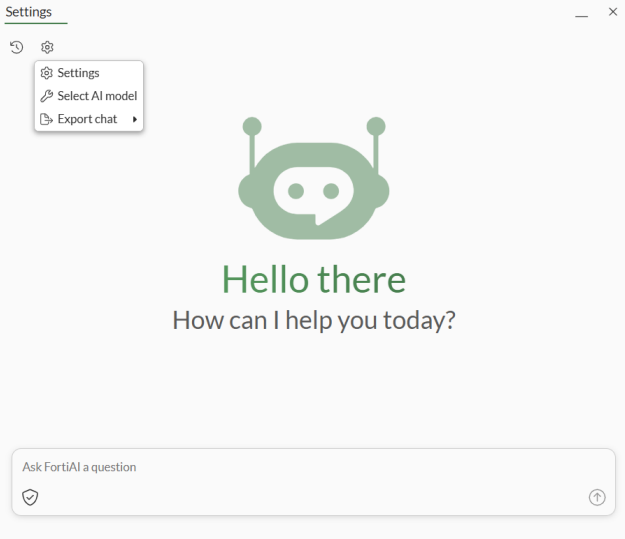
   - When FortiAI returns commands, an *Edit* button is available to open the commands in CLI Code Lab for editing.

FortiAI settings

In FortiAI, the following options are available under *Settings*:

- *Chat history* icon
- *Settings* icon

Option		Description
Chat history		Use to display a list of chat history. Click an item in the list to access it. See Accessing FortiAI chat history on page 96for details.
Settings	Settings	Use to check token usage for FortiAI and set the chat message history length. Available when FortiAI is the AI provider. See Checking token usage for FortiAI on page 94.
	Select AI model	Use to select one of the following AI providers for FortiAI-Assist: • FortiAI • OpenAI See Selecting an AI model on page 95.
	Export chat	Use to export FortiAI chats and threads to a JSON file. See Exporting FortiAI chats and threads on page 96.



You can also enable/disable data masking in chat queries and responses. See Masking sensitive data in chat on page 96.

Checking token usage for FortiAI

When FortiAI is the AI provider, you can check token usage in FortiAI settings. When monthly tokens are fully consumed, FortiAI access is temporarily suspended until the next monthly cycle begins. See also How token usage is calculated on page 89.

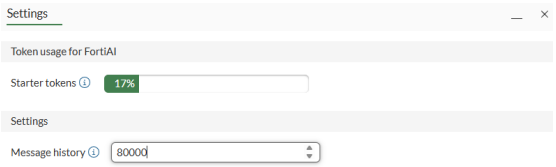
How much message history FortiAI sends with each new query in the same thread to the AI provider affects token usage. In FortiAI settings, the *Message history* field specifies the maximum number of words to include in message history. (The AI provider converts words to tokens.)

When you enter a query in FortiAI, the entire thread history is sent to the AI provider as context. When the thread history reaches the *Message history* limit, the history is compressed. Compressed history loses some information and is therefore less accurate.

A large history uses more allocated capacity for each query, and a small history may reduce the accuracy of the response because it provides too little context. It is recommended to start with the default setting.

To check FortiAI token usage:

1. In FortiAI, click *Settings* > *Settings*. The *Settings* are displayed.



2. The *Starter tokens* box displays the percentage of used tokens.
3. The *Message history* field displays how much message history FortiAI sends with each new query in the same thread to the AI provider.
4. Click *OK*.

Selecting an AI model

FortiAI-Assist supports the following AI providers:

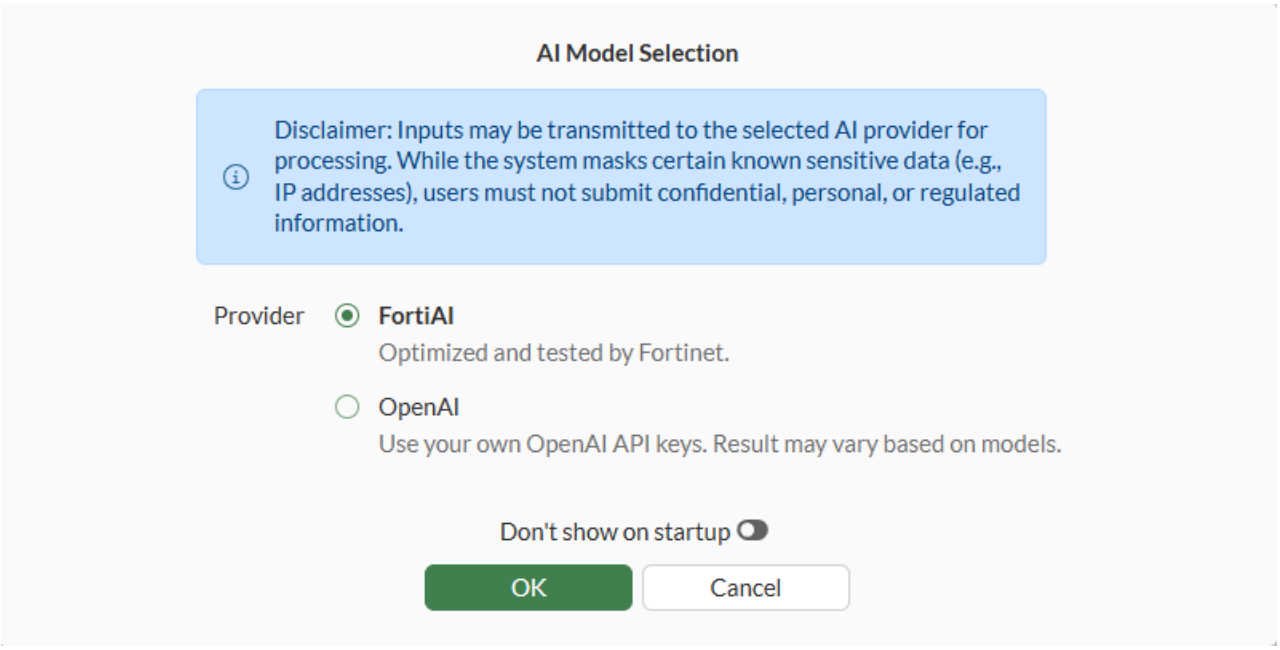
- FortiAI
- OpenAI

Administrators can choose the AI provider for FortiAI-Assist.

See Enabling FortiAI and accessing CLI Code Lab on page 90 for more information about each AI provider.

To select an AI model:

1. In FortiAI, click *Settings* > *Select AI model*. The *AI Model Selection* dialog box is displayed.



2. Set *Provider* to *FortiAI* or *OpenAI*.
3. Click *OK*.

Exporting FortiAI chats and threads

You can export the following history from FortiAI to a JSON file:

- Chat: the current conversation
- Thread: the current conversation plus details, such as timestamps, message IDs, and so on

To export chats and threads:

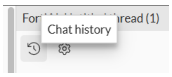
1. In FortiAI, click the *Settings* > *Export chat*, and select either *Export chat* or *Export thread*.
   A JSON file is downloaded to your local device.

Accessing FortiAI chat history

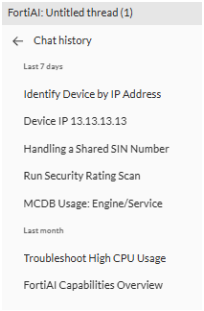
The *Chat history* button lets you view a list of chat history. You can click any entry in the list to view the thread.

To use chat history:

1. In FortiAI, click the *Chat history* button.



   A list of chat history is displayed.



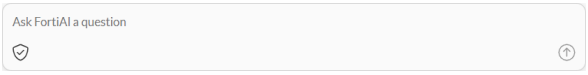
2. Click any item in the list to view the thread.

Masking sensitive data in chat

Sensitive data can be masked in FortiAI before it is sent to the AI provider. When data masking is enabled, a placeholder replaces sensitive data, such as IP addresses, MAC addresses, email addresses, phone numbers, URLs, and so on, before the data is sent to the AI provider.

Disclaimer: Inputs may be transmitted to the selected AI provider for processing. While the system masks certain known sensitive data (for example, IP addresses), administrators must not submit confidential, personal, or regulated information.

A checkmark displays in the FortiAI chat window when data masking is enabled:



An X displays in the FortiAI chat window when data masking is disabled:

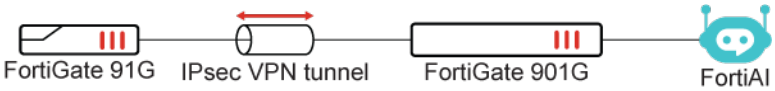
Click the icon to enable/disable data masking.

FortiAI examples

This section includes the following examples:

- Example 1: use FortiAI to troubleshoot FortiGate on page 97
- Example 2: use OpenAI to configure FortiGate on page 98

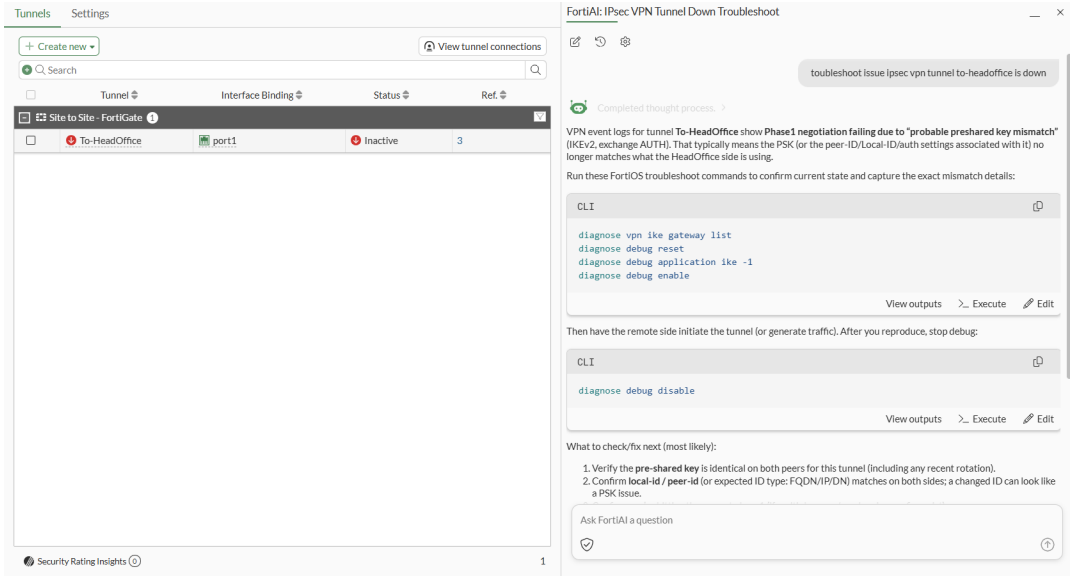
Example 1: use FortiAI to troubleshoot FortiGate



In this example, IPsec site-to-site VPN is configured between two FortiGates, but the tunnel is down. You can use FortiAI to help troubleshoot the issue.

To use FortiAI to troubleshoot:

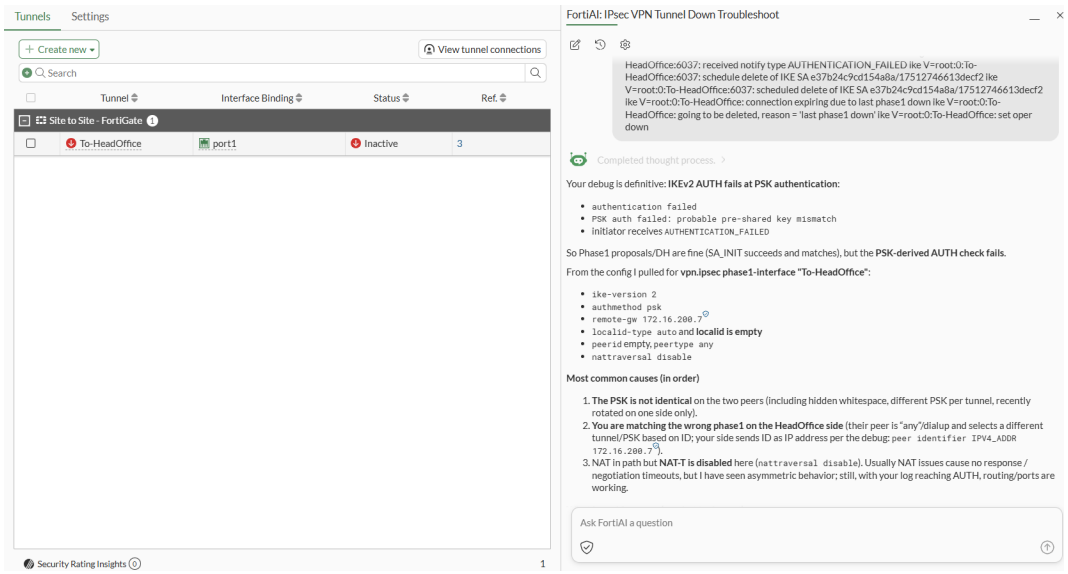
1. Enable FortiAI as the AI provider. See Enabling FortiAI with FortiAI as AI provider on page 90.
2. In the FortiAI chat window, enter the request: `toubleshoot issue ipsec vpn tunnel to-headoffice is down`.
3. FortiAI returns CLI commands to collect IKE debug logs as well as some suggested solutions.



4. In a separate SSH session, run the commands, collect the log output, and paste the output in the chat window.
5. FortiAI analyzes the log and returns the results in order with the most common cause listed first:

   Your debug is definitive: IKEv2 AUTH fails at PSK authentication:

   - authentication failed
   - PSK auth failed: probable pre-shared key mismatch
   - initiator receives AUTHENTICATION_FAILED

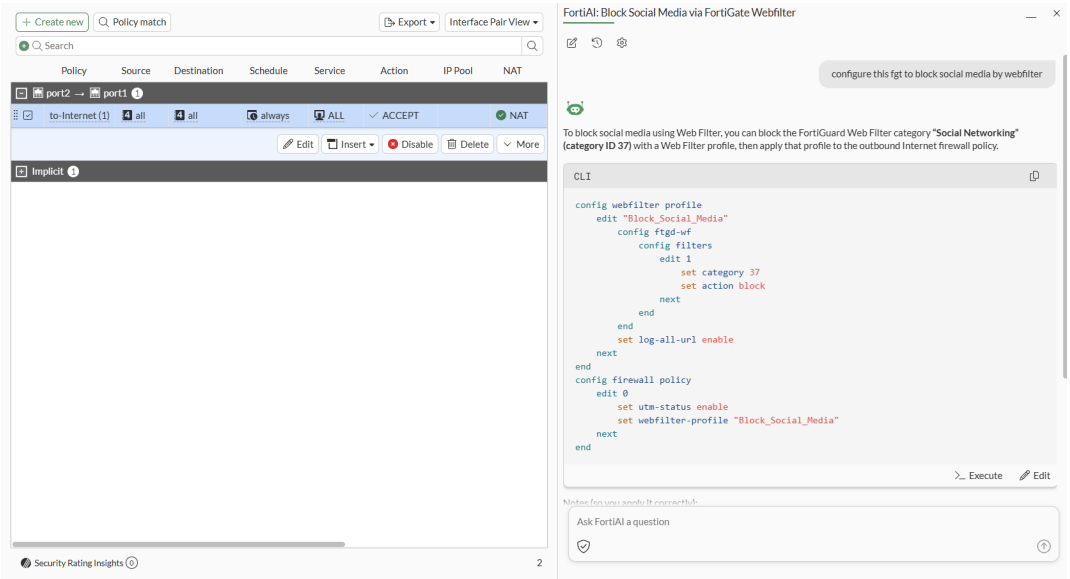


Example 2: use OpenAI to configure FortiGate

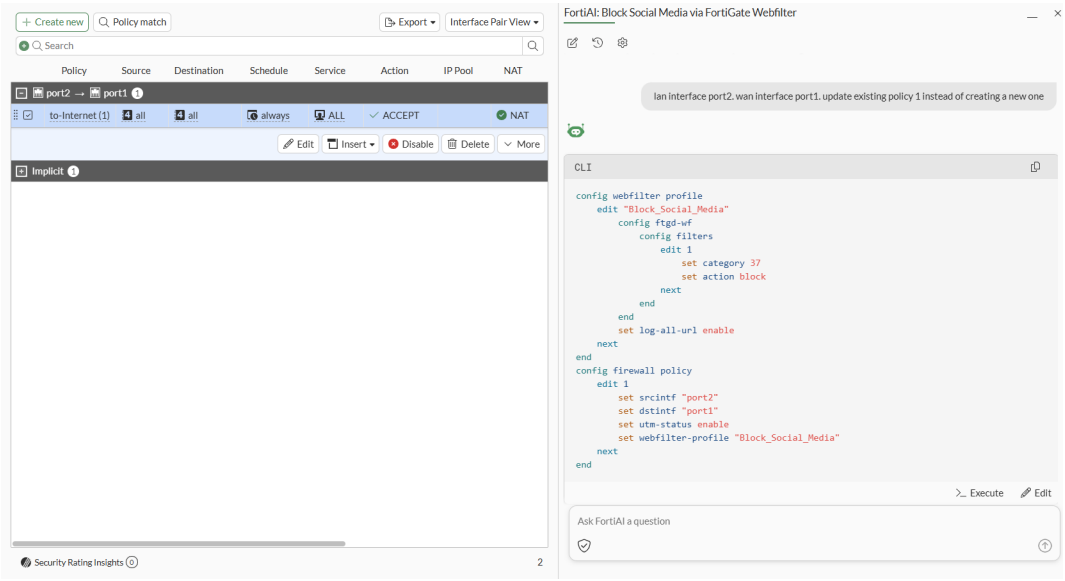
In this example, FortiAI is configured to use OpenAI as the AI provider. FortiAI is used to configure FortiGate to block social media by creating a web filter profile and updating a firewall policy to use the web filter profile.

To use OpenAI to configure FortiGate:

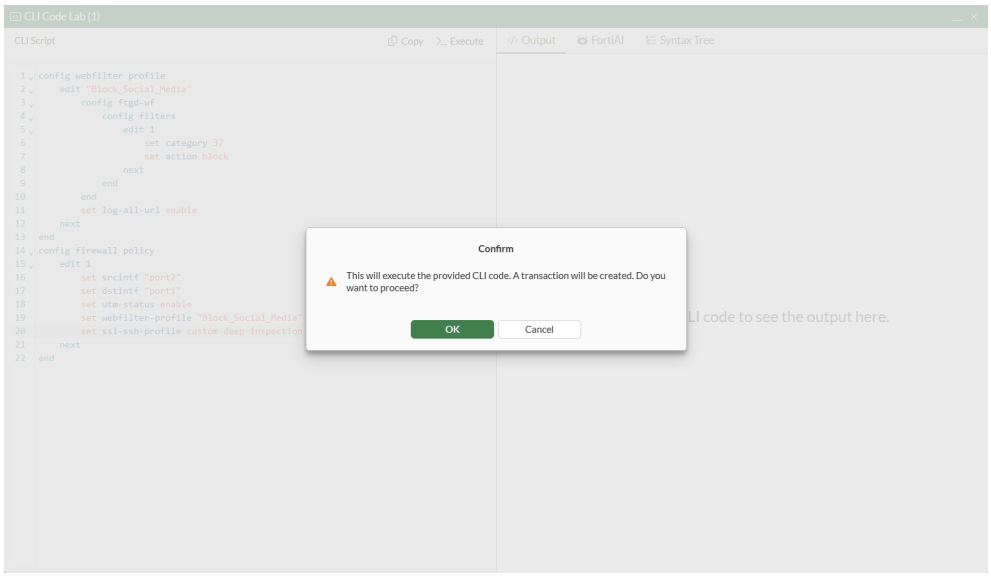
1. Enable OpenAI as the AI provider. See Enabling FortiAI with OpenAI as AI provider on page 91.
2. In the FortiAI chat window, enter the request: `configure this fgt to block social media by webfilter`.
3. FortiAI returns CLI commands for a web filter profile and firewall policy configuration.



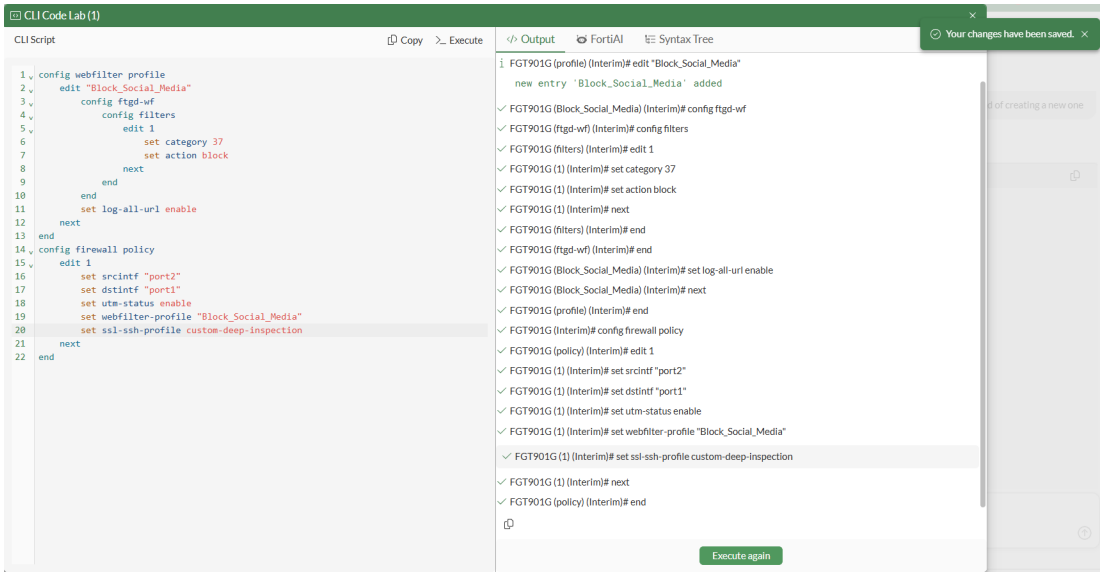
4. Enter more information, such as the LAN interface name or existing policy ID.
   For example, enter: `lan interface port2. wan interface port1. update existing policy 1 instead of creating a new one.`
   FortiAI uses the additional information to return tuned CLI commands.



5. On the bottom-right, click *Edit* to open the *CLI Code Lab*.
   In the left pane of the CLI Code Lab, edit the CLI commands as needed.
6. Click *Execute*. A *Confirm* dialog box is displayed.



7. Click *OK* to execute the commands. The output is printed in the right pane of the CLI Code Lab.
8. Click *Commit changes* to save the configuration changes.



Configuration and management

FortiOS can be managed through the graphical user interface (GUI) or the Command Line Interface (CLI) as well as other tools.

For	Use
Direct or individual configuration	FortiOS GUI and CLI: see Using the GUI on page 54 and Using the CLI on page 73.

For	Use
	FortiExplorer Go: see FortiExplorer Go on page 101.
Mass provisioning, management, and orchestration	FortiManager and FortiGate Cloud: see the FortiManager page and the FortiGate Cloud page on the Fortinet Document Library.
Automation	REST API accessible through Fortinet Developer Network (FNDN): see Accessing Fortinet Developer Network on page 107 and REST API administrator on page 3712. Automation tools, such as Terraform and Ansible: see Terraform: FortiOS as a provider on page 110.
Other tools and FortiConverter	The FortiConverter service helps you migrate a configuration from one FortiGate to another FortiGate, or from a third-party firewall to a FortiGate: see Migrating a configuration with FortiConverter on page 101.

FortiExplorer Go

FortiExplorer Go is a free mobile application that provisions and deploys BLE capable FortiGates with the BLE Autodiscovery feature. You can also use FortiExplorer Go to remotely manage FortiGates registered to your FortiCare account and deployed in FortiGate Cloud.

For more information, refer to the FortiExplorer Go User Guide for your respective device OS.

Migrating a configuration with FortiConverter

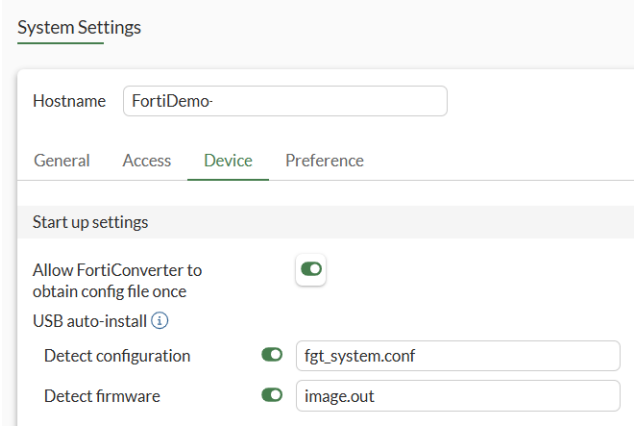
A configuration can be migrated from an older FortiGate device to a new FortiGate device directly from the FortiGate GUI, without having to access the FortiConverter portal.

Both the source and target FortiGates must be registered under the same FortiCare account and have internet connectivity to reach the FortiConverter server. The target FortiGate must also have a valid FortiConverter license.

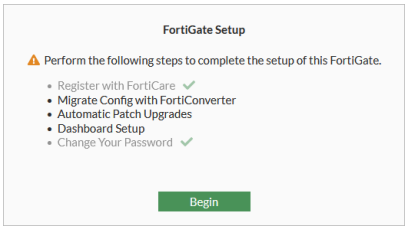
In this example, FortiGate A (FGTA) is replacing FortiGate B (FGTB). The configuration is migrated using FortiConverter, but without accessing the FortiConverter portal.

To migrate the configuration from FGTB to FGTA in the GUI:

1. On FGTB, go to *System > Settings > Device*, enable *Allow FortiConverter to obtain config file once*, then click *Apply*.

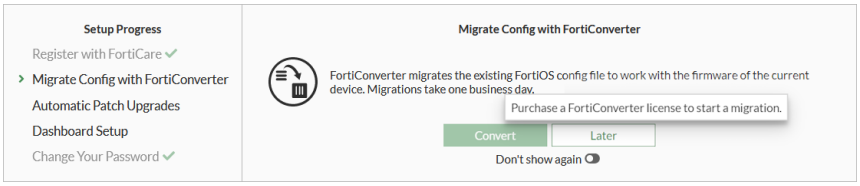


2. Log in to FGTA and on the GUI startup menu click *Begin* to start *Migrate Config with FortiConverter*.



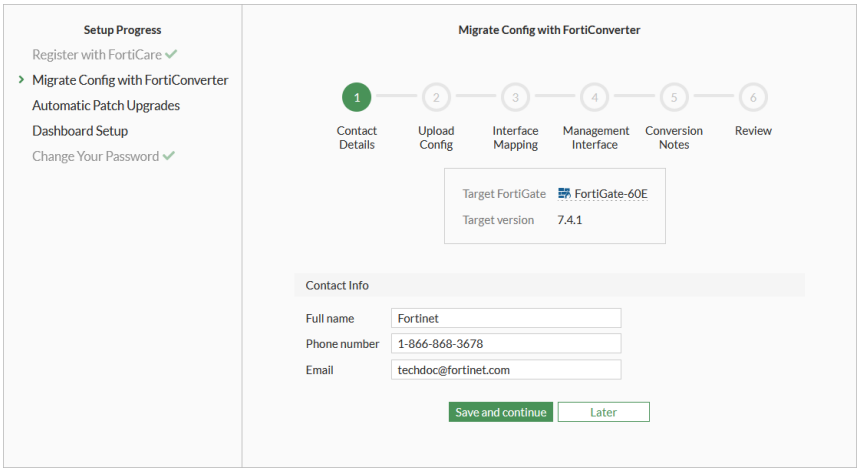
3. Click *Convert* to start the conversion process.

   If the device does not have a FortiConverter license, a warning will be shown and the *Convert* button will be unclickable. The license status is shown in the GUI on the *System* > *FortiGuard* page in the *License Information* table.



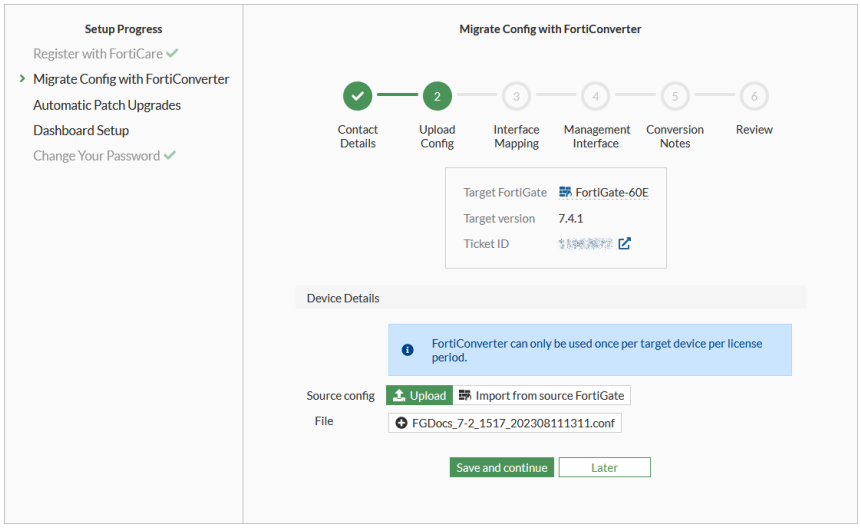
   You can toggle the *Don't show again* option and click *Later* to turn off reminders about the migration process.

4. Enter the user contact information, then click *Save and continue*.

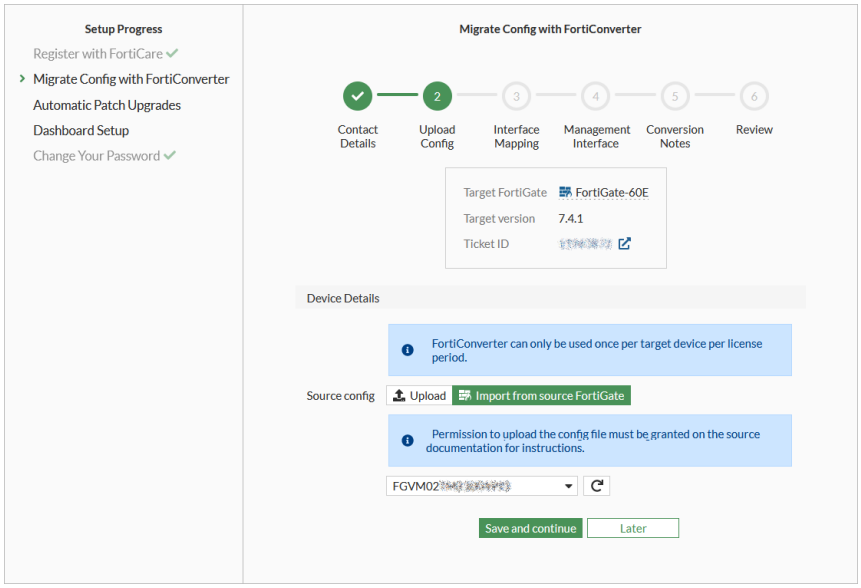


   The FortiConverter ticket is created.

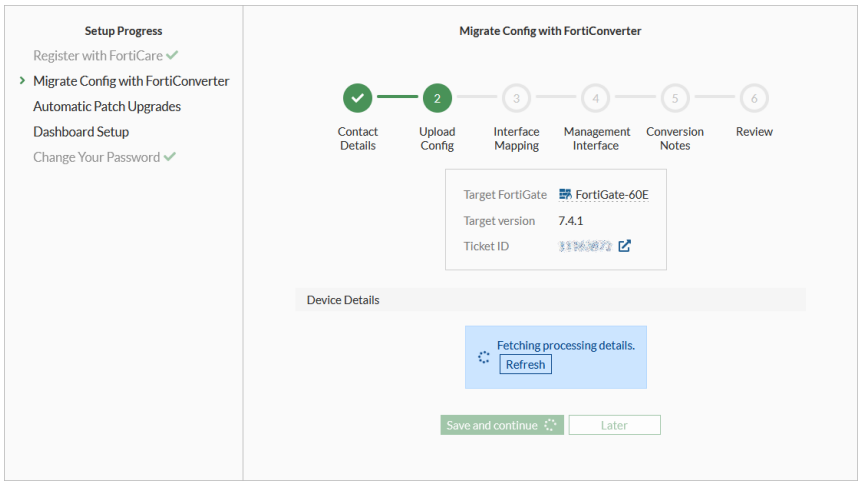
5. The source configuration can be uploaded from a file, or from another FortiGate. In this example, the configuration is uploaded from FGTB.
   - To upload from a file, set *Source config* to *Upload* then click *Browse* to locate the file.



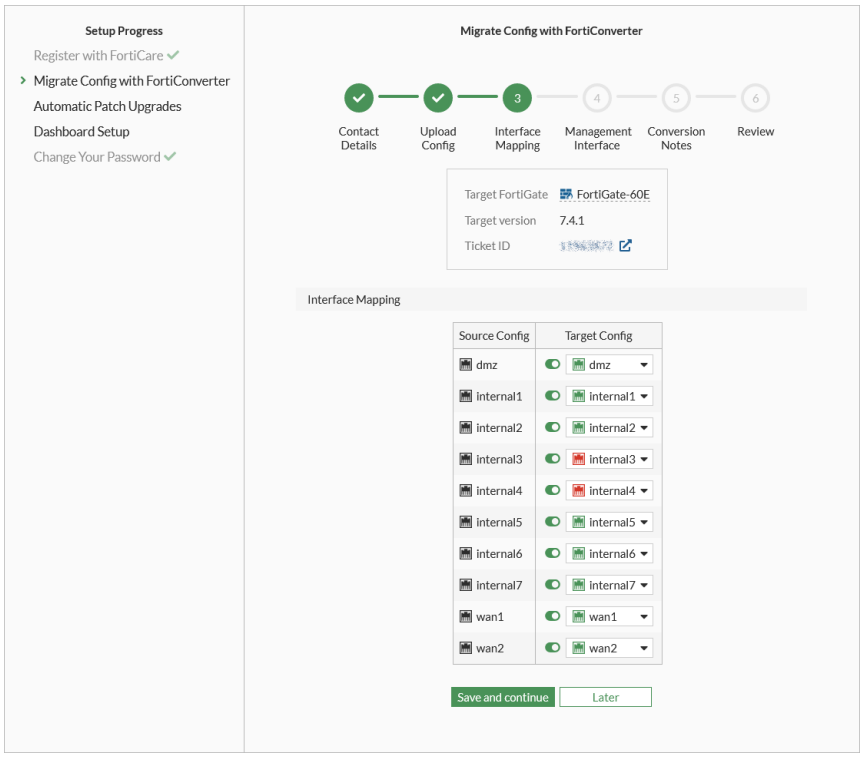
   - To import from FGTB, set *Source config* to *Import from source FortiGate* then select the FGTB. *Allow FortiConverter to obtain config file once* must be enabled in *System > Settings > Device* on FGTB.



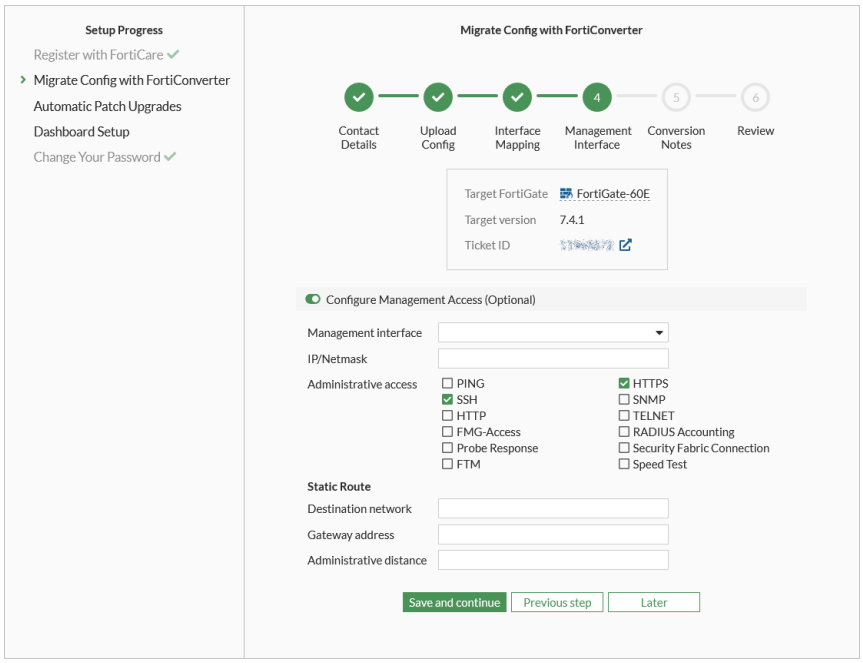
6. Click *Save and continue*, then wait for the FGTB configuration file to be uploaded to FortiConverter and processed. After the configuration is uploaded, the *Allow FortiConverter to obtain config file once* is automatically disabled on FGTB.



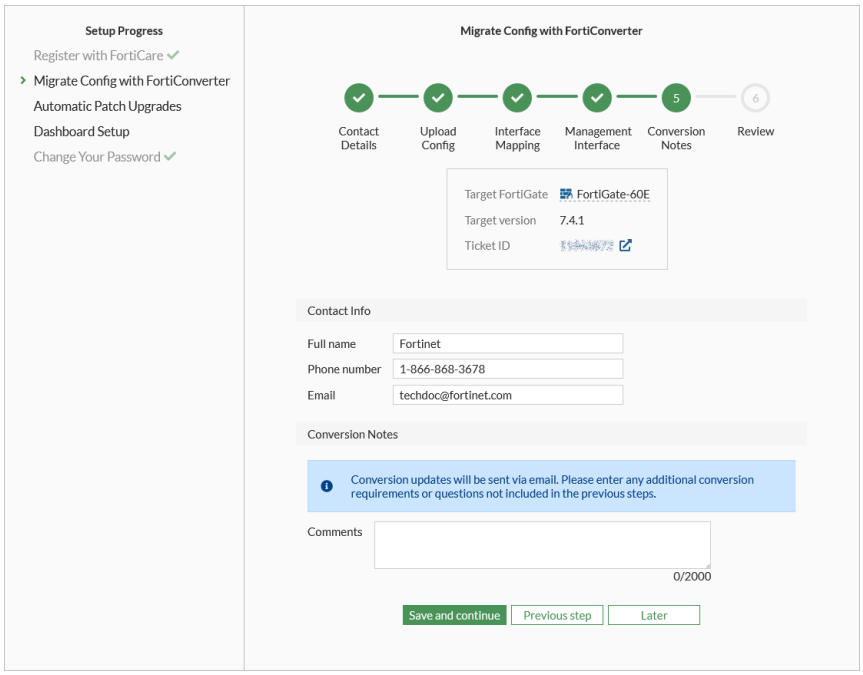
7. Define the interface mapping between the source and target configuration, then click *Save and continue*. The target interfaces are prepopulated.



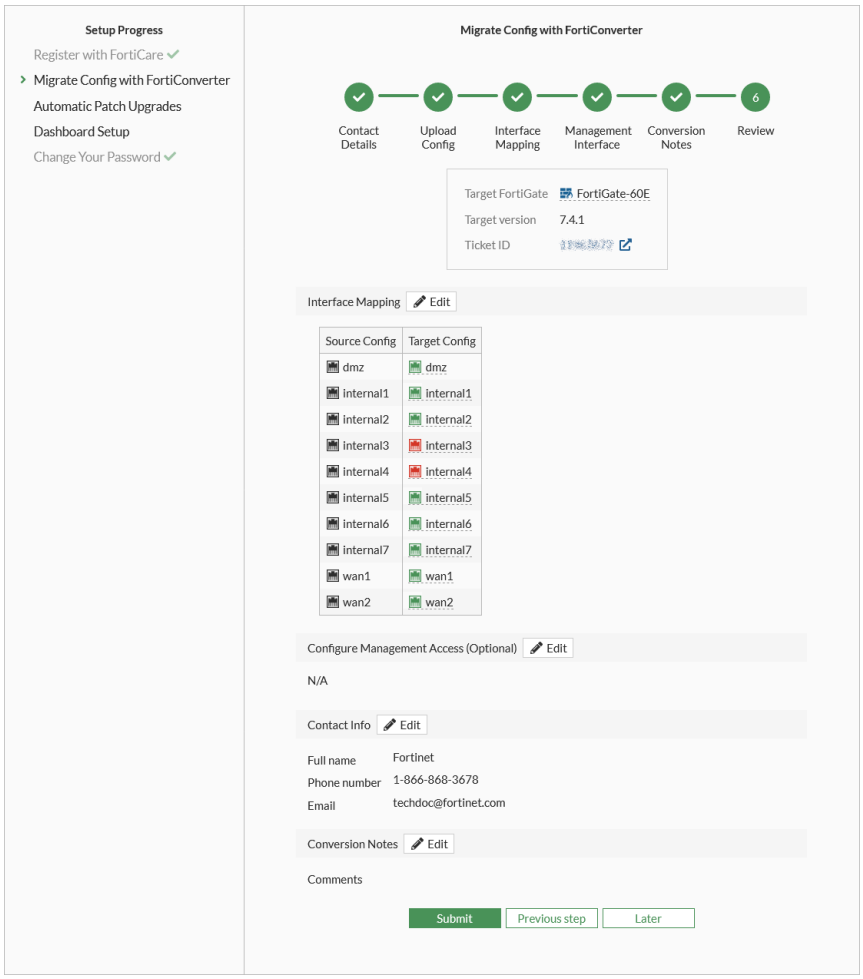
8. Optionally, configure management access on the target FortiGate (FGTA), then click *Save and continue*.



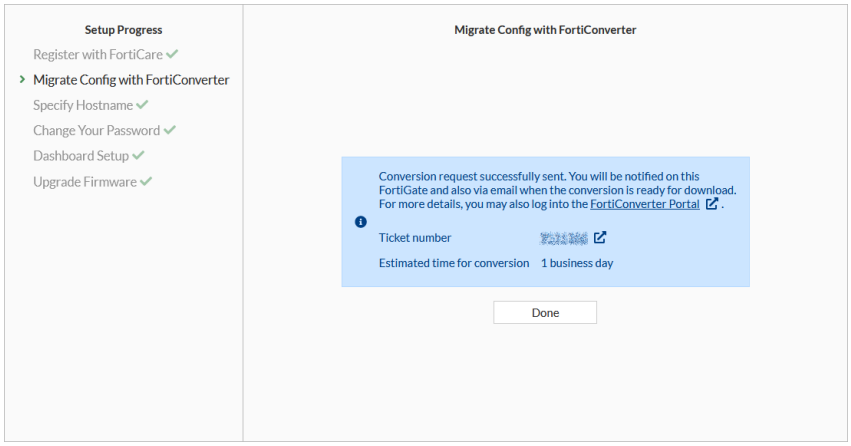
9. Enter conversion notes in the *Comments* field, then click *Save and continue*.



10. Review the content, then click *Submit*.

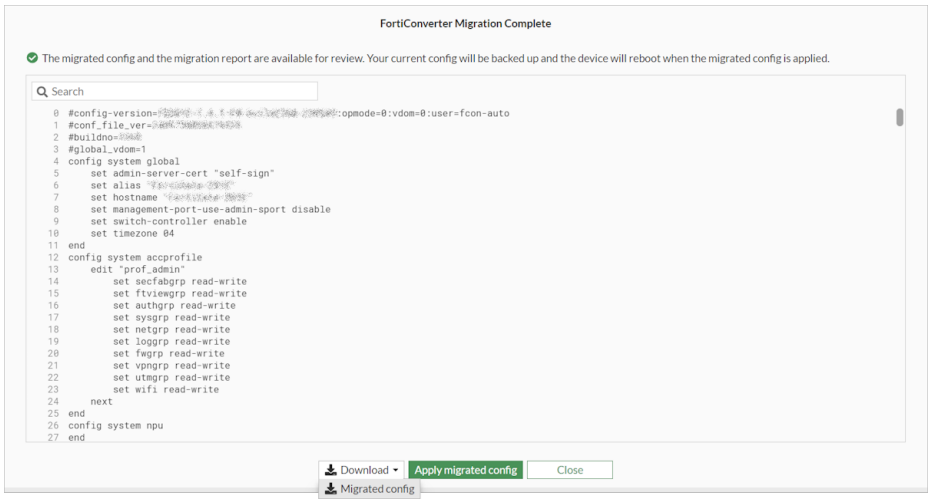


The conversion request is sent, an email is sent to confirm that the conversion process has started in FortiConverter, and the ticket status is shown. The estimated conversion time is one business day.

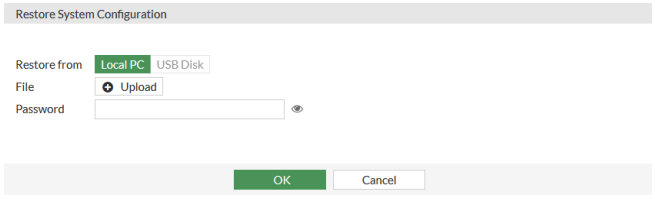


11. Click *Done*.

    When the conversion process completes, you will receive an email and a notifications in the FortiGate GUI.
12. In the GUI, click your administrator name and select *Configurations> FortiConverter*. The migrated configuration is shown for review, and can be downloaded.



13. Click *Apply migrated config* to apply the converted configuration to the FortiGate. This will cause the FortiGate to reboot. The existing configuration will be backed up before the converted configuration is applied.
14. To manually load to configuration file:
    a. Click your administrator name and select *Configuration* > *Restore*.



    b. Upload the converted configuration file, then click *OK*. This will cause the FortiGate to reboot.

To see the visibility status of the FortiConverter wizard:

```
diagnose sys forticonverter get-prompt-visibility
```

To set the visibility status of the FortiConverter wizard:

```
diagnose sys forticonverter set-prompt-visibility {visible | hidden}
```

Accessing Fortinet Developer Network

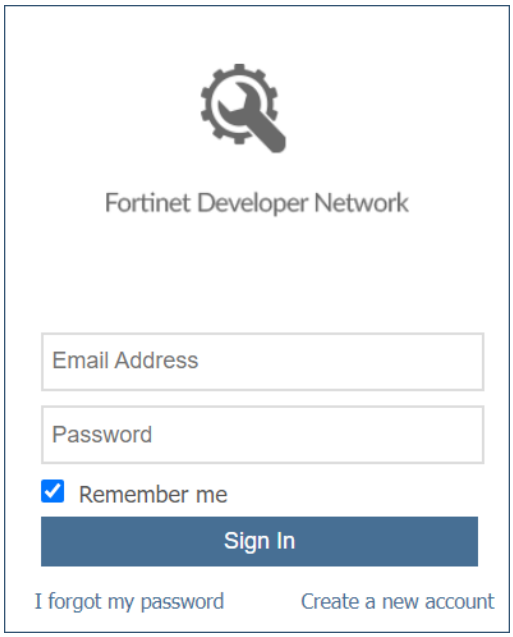
The Fortinet Developer Network (FNDN) is a subscription-based community that helps administrators enhance and increase the effectiveness of Fortinet products. Administrators can access the FortiAPI forum in FNDN to help create applications that interact with Fortinet products, such as custom web portals, automated deployment and provisioning systems, and scripted tasks. FNDN makes it easy for administrators and Fortinet professionals to interact, share sample code, and upload their own tools. The FortiOS REST API documentation is available within the FortiAPI forum.

All FNDN users must be sponsored by two Fortinet employees. The sponsors must be able to confirm the user's identity and need for access. Approvals from both sponsors are required before access is granted to new users. The sponsors' email addresses are required to create a new FNDN account.

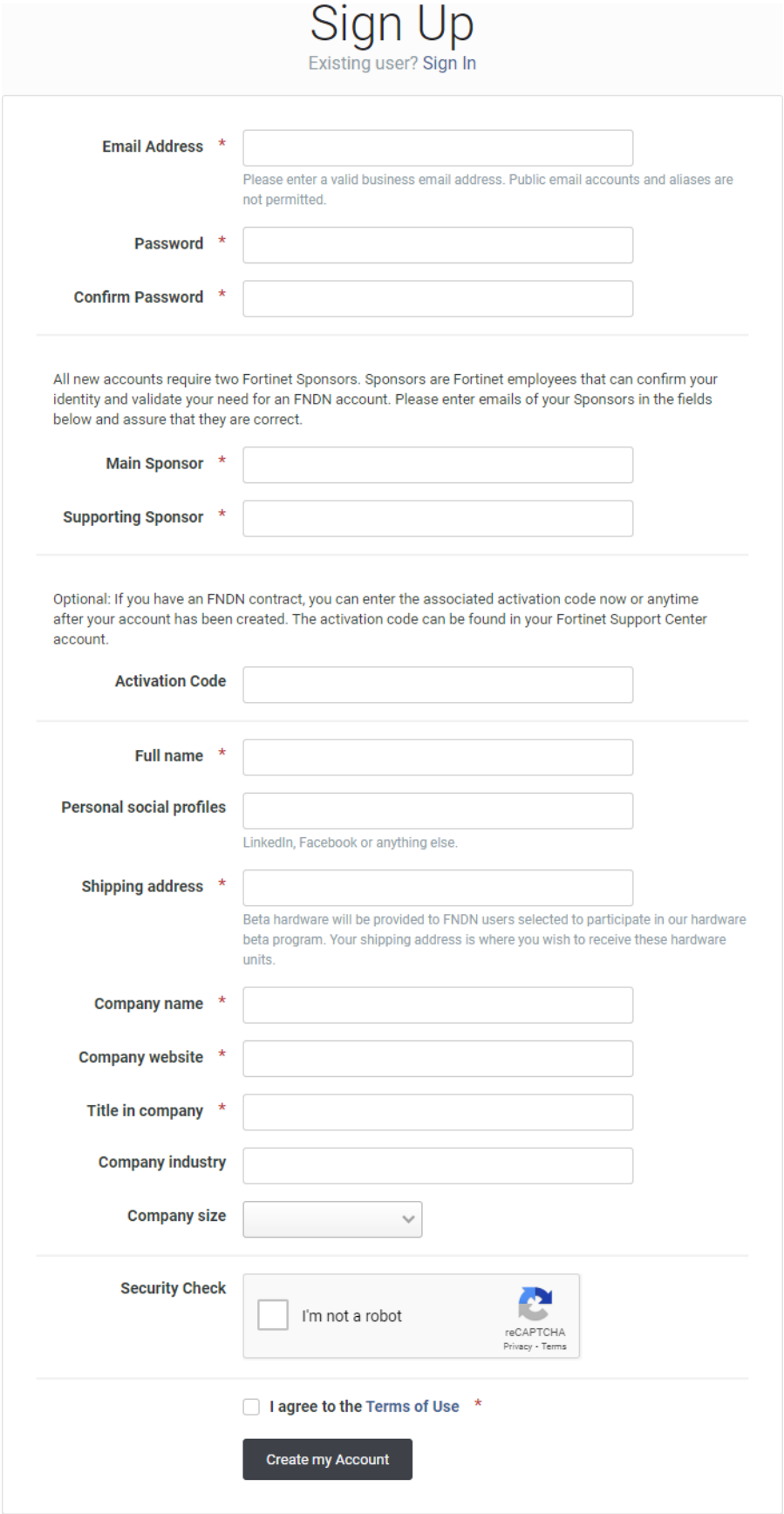
Basic and licensed access options are available. Refer to the Fortinet Developer Network data sheet for more information.

To create an FNDN account:

1. Obtain sponsorship from two Fortinet employees.
2. Go to the FNDN website, https://fndn.fortinet.net/. The log in page appears.



3. Click *Create a new account*. The *Sign Up* page appears.
4. Enter the information in the form fields and agree to the *Terms of Use*.


Sign Up
Existing user? Sign In
Email Address *
Please enter a valid business email address. Public email accounts and aliases are not permitted.
Password *
Confirm Password *
All new accounts require two Fortinet Sponsors. Sponsors are Fortinet employees that can confirm your identity and validate your need for an FNDN account. Please enter emails of your Sponsors in the fields below and assure that they are correct.
Main Sponsor *
Supporting Sponsor *
Optional: If you have an FNDN contract, you can enter the associated activation code now or anytime after your account has been created. The activation code can be found in your Fortinet Support Center account.
Activation Code
Full name *
Personal social profiles
LinkedIn, Facebook or anything else.
Shipping address *
Beta hardware will be provided to FNDN users selected to participate in our hardware beta program. Your shipping address is where you wish to receive these hardware units.
Company name *
Company website *
Title in company *
Company industry
Company size
Security Check
I'm not a robot
reCAPTCHA
Privacy - Terms
I agree to the Terms of Use *
Create my Account

5. Click *Create my Account*.
   New accounts are reviewed and approved by an FNDN administrator. After both sponsors approve the request, an FNDN administrator reviews the request and approves account access in around one business day if all requirements are met.

Terraform: FortiOS as a provider

Fortinet's Terraform support provides customers with more ways to efficiently deploy, manage, and automate security across physical FortiGate appliances and virtual environments. You can use Terraform to automate various IT infrastructure needs, thereby diminishing mistakes from repetitive manual configurations.

For example, if Fortinet is releasing a new FortiOS version, your organization may require you to test a new functionality to determine how it may impact the environment before globally deploying the new version. In this case, the ability to rapidly stand up environments and test these functions prior to production environment integration provides a resource-efficient and fault-tolerant approach.

The following example demonstrates how to use the Terraform FortiOS provider to perform simple configuration changes on a FortiGate unit. It requires the following:

- FortiOS 6.0 or later
- FortiOS Provider: This example uses terraform-provider-fortios 1.0.0.
- Terraform: This example uses Terraform 0.11.14.
- REST API administrator created on the FortiGate with the API key

For more information, see the Terraform FortiOS Provider at https://www.terraform.io/docs/providers/fortios/index.html.

To create a REST API administrator:

1. On the FortiGate, go to *System > Administrators* and click *Create New > REST API Admin*.
2. Enter the *Username* and, optionally, enter *Comments*.
3. Select an *Administrator Profile*.
4. We recommend that you create a new profile with minimal privileges for this terraform script:
   a. In the *Administrator Profile* drop down click *Create New*.
   b. Enter a name for the profile.
   c. Configure the *Access Permissions*:
      - *None*: The REST API is not permitted access to the resource.
      - *Read*: The REST API can send read requests (`HTTP GET`) to the resource.
      - *Read/Write*: The REST API can send read and write requests (`HTTP GET/POST/PUT/DELETE`) to the resource.
   d. Click *OK*.
5. Enter *Trusted Hosts* to specify the devices that are allowed to access this FortiGate.
6. Click *OK*.
   An API key is displayed. This key is only shown once, so you must copy and store it securely.

To configure FortiGate with Terraform Provider module support:

1. Download the terraform-provider-fortios file to a directory on the management computer.
2. Create a new file with the .tf extension for configuring your FortiGate:

```
root@mail:/home/terraform# ls
terraform-provider-fortios_v1.0.0_x4  test.tf
```

3. Edit the test.tf Terraform configuration file:
   In this example, the FortiGate's IP address is 10.6.30.5, and the API user token is 17b**********63ck. Your provider information must also be changed.

```
# Configure the FortiOS Provider
provider "fortios" {
hostname = "10.6.30.5"
token = "17b**********63ck"
}
```

4. Create the resources for configuring your DNS object and adding a static route:

```
resource "fortios_system_setting_dns" "test1" {
primary = "172.16.95.16"
secondary = "8.8.8.8"
}
resource "fortios_networking_route_static" "test1" {
dst = "110.2.2.122/32"
gateway = "2.2.2.2"
blackhole = "disable"
distance = "22"
weight = "3"
priority = "3"
device = "port2"
comment = "Terraform test"
}
```

5. Save your Terraform configuration file.
6. In the terminal, enter `terraform init` to initialize the working directory.
   It reads the provider if the name follows the convention `terraform-provider-[name]`:

```
root@mail:/home/terraform# terraform init
Initializing the backend...
Terraform has been successfully initialized!
You may now begin working with Terraform. Try running "terraform plan" to see
any changes that are required for your infrastructure. All Terraform commands
should now work.
If you ever set or change modules or backend configuration for Terraform,
rerun this command to reinitialize your working directory. If you forget, other
commands will detect it and remind you to do so if necessary.
```

7. Run `terraform -v` to verify the version of loaded provider module:

```
root@mail:/home/terraform# terraform -v
Terraform v0. 11.14
+ provider.fortios v1.0.0
```

8. Enter `terraform plan` to parse the configuration file and read from the FortiGate configuration to see what Terraform changes:

   This example create a static route and updates the DNS address. You can see that Terraform reads the DNS addresses from the FortiGate and then lists them.

```
root@mail:/home/terraform# terraform plan
Refreshing Terraform state in-memory prior to plan...
The refreshed state will be used to calculate this plan, but will not be
persisted to local or remote state storage.
fortios_networking_route_static.test1: Refreshing state... (ID: 2)
fortios_system_setting_dns.test1: Refreshing state... (ID: 96.45.45.45)
------------------------------------------------------------------------
An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
+ create
~ update in-place
Terraform will perform the following actions:
+ fortios_networking_route_static.test1
id: <computed>
blackhole: "disable"
comment: "Terraform test"
device: "port2"
distance: "22"
dst: "110.2.2.122/32"
gateway: "2.2.2.2"
priority: "3"
weight: "3"
~ fortios_system_setting_dns.test1
primary: "96.45.45.45" => "172.16.95.16"
secondary: "208.91.112.22" => "8.8.8.8"
Plan: 1 to add, 1 to change, 0 to destroy.
------------------------------------------------------------------------
Note: You didn't specify an "-out" parameter to save this plan, so Terraform
can't guarantee that exactly these actions will be performed if
"terraform apply" is subsequently run.
```

If you are running terraform-provider-fortios 1.1.0, you may see the following error:

```
Error: Error getting CA Bundle, CA Bundle should be set when insecure is false.
```

In this case, add the following line to the FortiOS provider configuration in the test.tf file:

```
insecure = "true"
```

9. Enter `terraform apply` to continue the configuration:

```
root@mail:/home/terraform# terraform apply
fortios_system_setting_dns.test1: Refreshing state... (ID: 96.45.45.45)
fortios_networking_route_static.test1: Refreshing state... (ID: 2)
An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
+ create
~ update in-place
Terraform will perform the following actions:
+ fortios_networking_route_static.test1
id: <computed>
blackhole: "disable"
comment: "Terraform test"
device: "port2"
distance: "22"
dst: "110.2.2.122/32"
gateway: "2.2.2.2"
priority: "3"
weight: "3"
~ fortios_system_setting_dns.test1
primary: "96.45.45.45" => "172.16.95.16"
secondary: "208.91.112.22" => "8.8.8.8"
Plan: 1 to add, 1 to change, 0 to destroy.
Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.
Enter a value: yes
fortios_networking_route_static.test1: Creating...
blackhole: "" => "disable"
comment: "" => "Terraform test"
device: "" => "port2"
distance: "" => "22"
dst: "" => "110.2.2.122/32"
gateway: "" => "2.2.2.2"
priority: "" => "3"
weight: "" => "3"
fortios_system_setting_dns.test1: Modifying... (ID: 96.45.45.45)
primary: "96.45.45.45" => "172.16.95.16"
secondary: "208.91.112.22" => "8.8.8.8"
fortios_networking_route_static.test1: Creation complete after 0s (ID: 2)
fortios_system_setting_dns.test1: Modifications complete after 0s (ID: 172.16.95.16)
Apply complete! Resources: 1 added, 1 changed, 0 destroyed.
```

The FortiGate is now configured according to the configuration file.

10. To change or delete something in the future, edit the configuration file and then apply it again. In supported cases, it deletes, adds, or updates new entries as configured. For instance, in this example you can remove the static route and revert the DNS address to its original configuration by changing the .tf file:
    a. Edit the configuration file:

```
# Configure the FortiOS Provider
provider "fortios" {
hostname = "10.6.30.5"
```

```
token = "17b********************63ck"
}
resource "fortios_system_setting_dns" "test1" {
primary = "96.45.45.45"
secondary = "208.91.112.22"
}
#resource "fortios_networking_route_static" "test1" {
# dst = "110.2.2.122/32"
# gateway = "2.2.2.2"
# blackhole = "disable"
# distance = "22"
# weight = "3"
# priority = "3"
# device = "port2"
# comment = "Terraform test"
#}
```

b. Entering `terraform apply` deletes the static route that is commented out of the configuration file, and reverts the DNS address to the old address:

```
root@mail:/home/terraform# terraform apply
fortios_system_setting_dns.test1: Refreshing state... (ID: 172.16.95.16)
fortios_networking_route_static.test1: Refreshing state... (ID: 2)
An execution plan has been generated and is shown below.
Resource actions are indicated with the following symbols:
~ update in-place
- destroy
Terraform will perform the following actions:
- fortios_networking_route_static.test1
~ fortios_system_setting_dns.test1
primary: "172.16.95.16" => "96.45.45.45"
secondary: "8.8.8.8" => "208.91.112.22"
Plan: 0 to add, 1 to change, 1 to destroy.
Do you want to perform these actions?
Terraform will perform the actions described above.
Only 'yes' will be accepted to approve.
Enter a value: yes
fortios_networking_route_static.test1: Destroying... (ID: 2)
fortios_system_setting_dns.test1: Modifying... (ID: 172.16.95.16)
primary: "172.16.95.16" => "96.45.45.45"
secondary: "8.8.8.8" => "208.91.112.22"
fortios_networking_route_static.test1: Destruction complete after 0s
fortios_system_setting_dns.test1: Modifications complete after 0s (ID: 96.45.45.45)
Apply complete! Resources: 0 added, 1 changed, 1 destroyed.
```

Troubleshooting

Use the HTTPS daemon debug to begin troubleshooting why a configuration was not accepted:

```
# diagnose debug enable
# diagnose debug application httpsd -1
```

The REST API 403 error means that your administrator profile does not have sufficient permissions.
The REST API 401 error means that you do not have the correct token or trusted host.

Product registration with FortiCare

It is recommended to register your product with Fortinet. A FortiCare/FortiCloud account with Fortinet Technical Support (https://support.fortinet.com) is required to register products. This section describes how to register the product and includes information about other tasks performed with a FortiCare/FortiCloud account.

- FortiCare and FortiGate Cloud login on page 115
- FortiCare Register button on page 118
- Transfer a device to another FortiCloud account on page 119
- Deregistering a FortiGate on page 121

FortiCare and FortiGate Cloud login

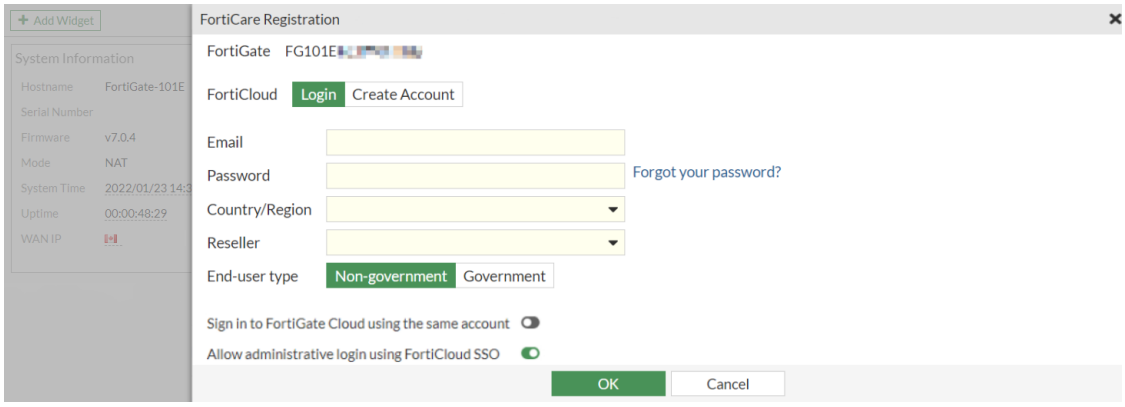
With FortiCloud, FortiOS supports a unified login to FortiCare and FortiGate Cloud. The FortiGate Cloud setup is a subset of the FortiCare setup.

- If the FortiGate is not registered, activating FortiGate Cloud will force you to register with FortiCare.
- If a FortiGate is registered in FortiCare using a FortiCloud account, then only that FortiCloud account can be used to activate FortiGate Cloud.
- If a different FortiCloud account was already used to activate FortiGate Cloud, then a notification asking you to migrate to FortiCloud is shown in the GUI after upgrading FortiOS.

The CLI can be used to activate FortiGate Cloud without registration, or with a different FortiCloud account.

To activate FortiGate Cloud and register with FortiCare at the same time:

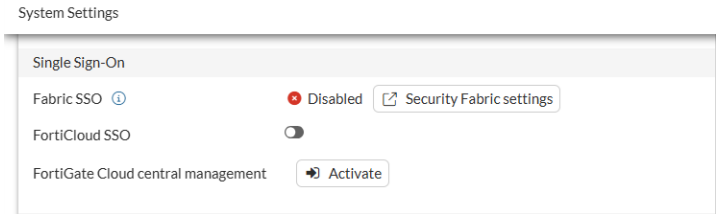
1. Go to *Dashboard* > *Status*.
2. In the FortiGate Cloud widget, click *Not Activated* > *Activate*.
   You must register with FortiCare before activating FortiGate Cloud.



3. Enter your FortiCare *Email* address and *Password*.
4. Select your *Country/Region*, *Reseller*, and *End-user type*.
5. Enable *Sign in to FortiGate Cloud using the same account*.
6. Click *OK*.

To activate FortiGate Cloud from System Settings:

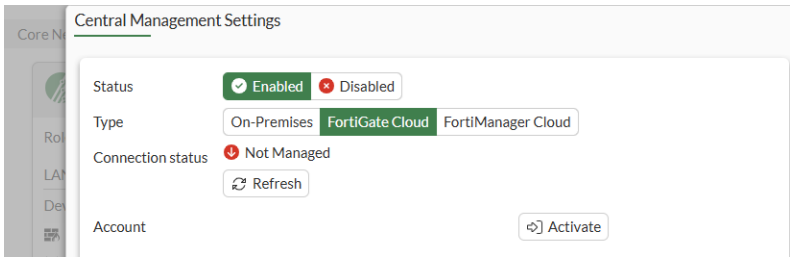
1. Go to *System* > *Settings* > *Access*.
2. Under *Single Sign-On* and beside *FortiGate Cloud central management*, click *Activate*.



3. Click *Apply*.

To activate FortiGate Cloud from Fabric connectors:

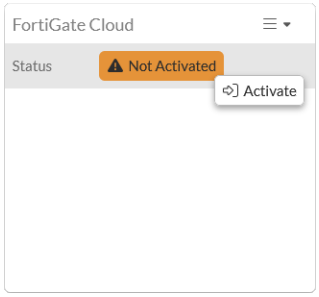
1. Go to *Security Fabric* > *Fabric Connectors*.
2. Click *Central Management*.
3. Set *Status* to *Enabled*, and set *Type* to FortiGate Cloud. *FortiGate Cloud*.



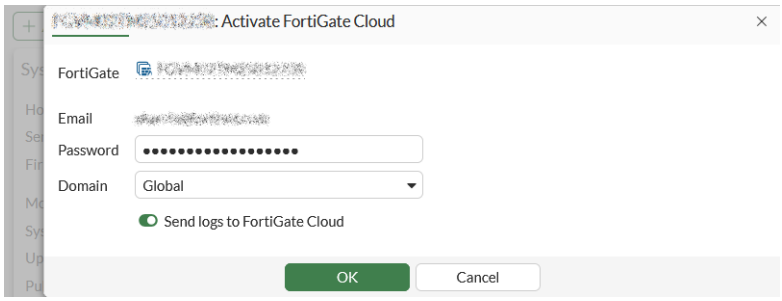
4. Click *OK*.

To activate FortiGate Cloud on an already registered FortiGate:

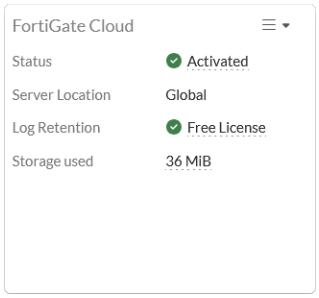
1. Go to *Dashboard* > *Status*.
2. In the *FortiGate Cloud* widget, click *Not Activated* > *Activate*.



3. Enter the password for the account that was used to register the FortiGate.

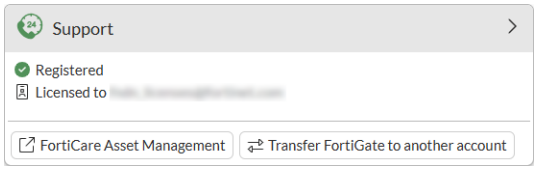


4. Click *OK*.
   The *FortiGate Cloud* widget now shows the activated FortiCloud account.

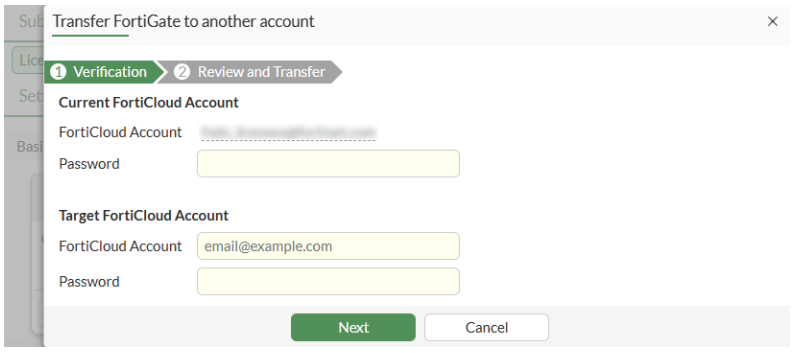


To migrate from the activated FortiGate Cloud account to the registered FortiCloud account:

1. Go to *System* > *FortiGuard*.
2. Go to the *Subscriptions* > *Licensed* tab.
3. In the *Support* widget, click *Transfer FortiGate to another account*.



4. Enter the *Password* of the current FortiCloud account.



5. Enter the target *FortiCloud Account* name and *Password*, then click *Next*.
6. Review the information in the *From* and *To* fields, then click *Transfer*.

To activate FortiGate Cloud using an account that is not used for registration:

1. Enter the following with the credentials for the account being used to activate FortiGate Cloud:

```
# execute fortiguard-log login <account_id> <password>
```

2. Check the account type:

```
# diagnose fdsm contract-controller-update
Protocol=2.0|Response=202|Firmware=FAZ-4K-FW-2.50-
100|SerialNumber=FAMS000000000000|Persistent=false|ResponseItem=HomeServer:172.16.95.151:443*A
lterServer:172.16.95.151:443*Contract:20200408*NextRequest:86400*UploadConfig:False*Management
Mode:Local*ManagementID:737941253*AccountType:multitenancy

Result=Success
```

A FortiCloud account that is not used for the support portal account cannot be used to register FortiGate. Attempting to activate FortiGate Cloud with this type of account will fail.

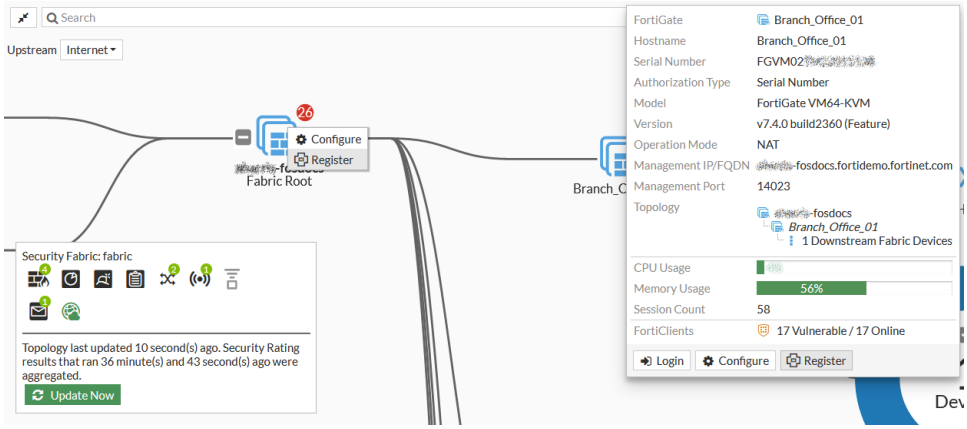
For FortiGates managed by FortiGate Cloud, automatic firmware patch may be enabled depending on the FortiGate Cloud version and portal in use. See the Administration Guide for the applicable FortiGate Cloud version and portal:

- Standard Portal Administration Guide
- 25.1.a Portal (Beta) Administration Guide
- Premium Portal Administration Guide

FortiCare Register button

The FortiCare *Register* button is displayed in the GUI on various Fabric and device related pages and widgets.

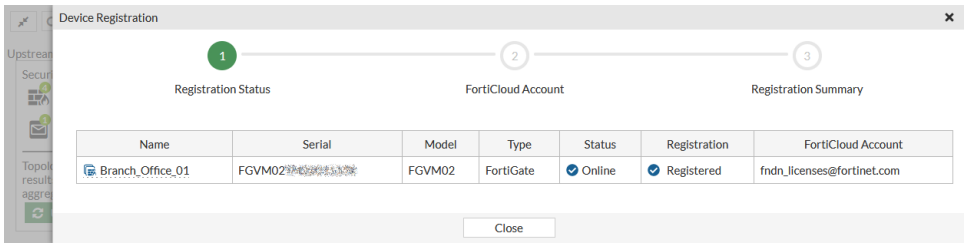
- To access the *Register* button on a topology page, click on or hover over the FortiGate device:



- To access the button from the *System > Firmware & Registration* or *System > HA* page, right-click on the device name.

The *Register* button is also accessible from tooltips for devices on the *Managed FortiAPs* and *Managed FortiSwitches pages*.

Clicking *Register* opens the *Device Registration* pane. If a device is already registered, the pane still opens and displays the device information.



Primary and secondary HA members can be registered to FortiCare at the same time from the primary unit by using the *Register* button. The secondary unit will register through the HA proxy.

In this example, a HA member is registered from the *Physical Topology* page.

To register a HA member to FortiCare:

1. On the primary unit, go to *Security Fabric > Physical Topology*, or expand the *Security Fabric* widget on the *Status* dashboard.
2. Hover over the HA member and click *Register*. The *Device Registration* pane opens.
3. Select the device and click *Register*.
4. Enter the required FortiCloud account information (password, country or region, reseller) and click *Submit*.
5. Once the registration is complete, click *Close*.

Transfer a device to another FortiCloud account

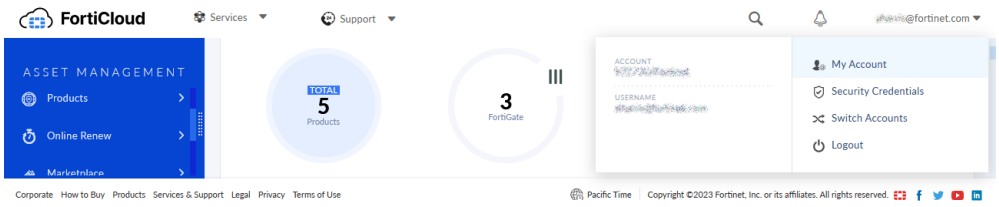
Master account users can transfer a device from one FortiCloud/FortiCare account to another. Users can transfer a device up to three times within a twelve-month time period. If more transfers are required within the twelve-month time period, contact Technical Support to request the transfer.

Requirements:

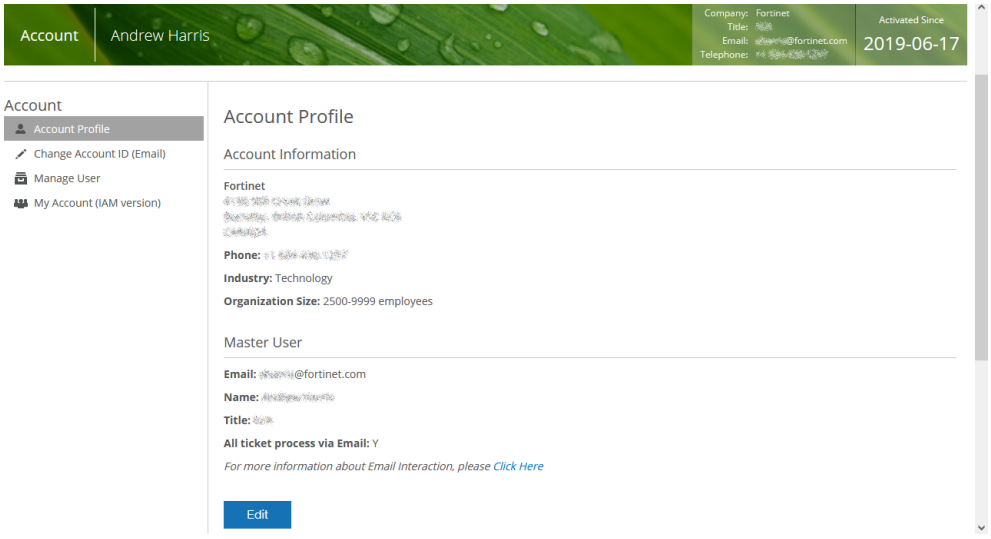
To transfer an account, you must:

- Have access to the FortiGate, as well as both the FortiCloud and FortiCare accounts.
- Be a master account user.

  To verify if you are the master account user, log in to support.fortinet.com. Click the username, then select *My Account*.



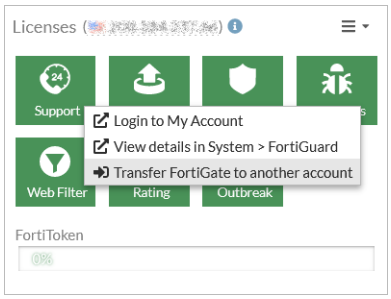
The *Account Profile* page opens.



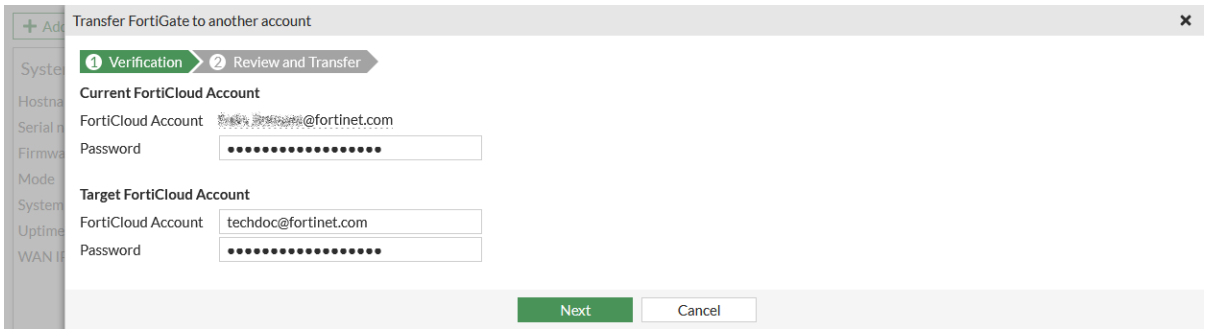
To transfer an account in the GUI:

1. Go to *Dashboard* > *Status*.
2. In the *Licenses* widget, click the *Support* link, then click *Transfer FortiGate to Another Account*.

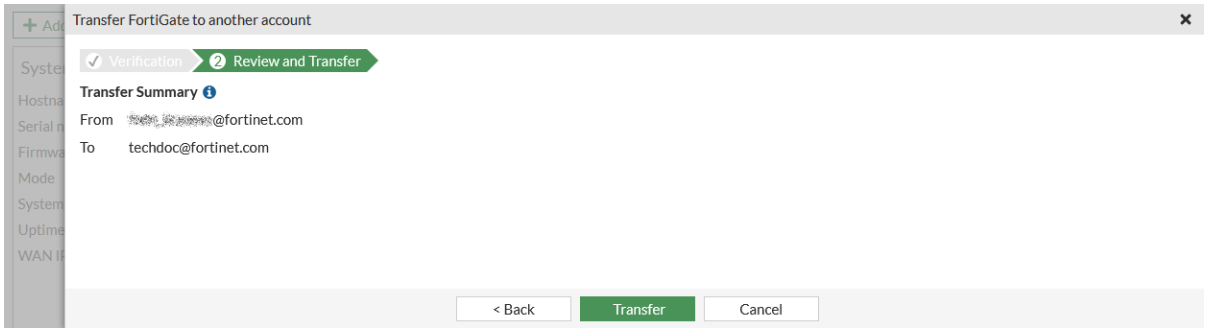
   You can also transfer an account from *System* > *FortiGuard*.



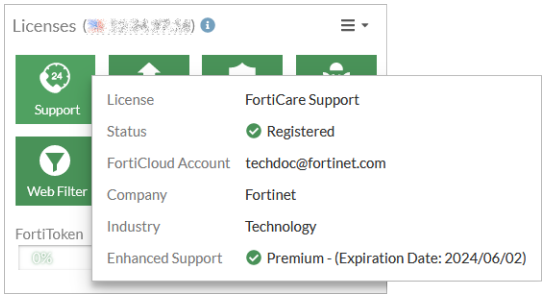
3. In the *Current FortiCloud Account* fields, enter the username and password for the current account. In the *Target FortiCloud Account* fields, enter the new username and password.
4. Click *Next*.



5. Review the information, then click *Transfer*.



After the transfer is complete, the new the FortiCloud account is displayed in the *Licenses* widget.



Deregistering a FortiGate

An administrator can deregister a FortiGate if the device has been registered for three or more years, using the GUI or CLI, without having to contact FortiCare administration. After the device is deregistered, all associated contracts are also deregistered, and all of the administrator's information is wiped.

To deregister the FortiGate in the GUI:

1. Go to *System* > *FortiGuard* and in the *FortiCare Support* row select *Actions* > *Deregister FortiGate*.
   The FortiCare Deregistration pane opens.
2. Enter your password then click *Next*.
3. Confirm the FortiGate deregistration then click *Submit*.
   If the FortiGate has been registered for less then three years, the deregistration will fail.

To deregister the FortiGate in the CLI:

```
# diagnose forticare direct-registration product-deregister <accountID> <password>
```

If the FortiGate has been registered for less then three years, the deregistration will fail:

```
forticare_product_deregister:1335: Failed to get response (rc = 0, http_code = 403)
Unit deregistration unsuccessful.
```

FortiGate models

Not all FortiGates have the same features, and some models support low encryption. This section also describes typical LEDs found on FortiGate models.

- Differences between models on page 122
- Low encryption models on page 123
- LEDs on page 123
- Proxy-related features not supported on FortiGate 2 GB RAM models on page 126
- GUI features not supported on FortiGate 2 GB RAM models on page 128
- FGR-70G/70F/70F-3G4G GPIO/DIO module on page 129

Differences between models

Not all FortiGates have the same features, particularly entry-level models (models 30 to 90). A number of features on these models are only available in the CLI.

Consult your model's QuickStart Guide, hardware manual, or the Feature / Platform Matrix for further information about features that vary by model.

FortiGate models differ principally by the names used and the features available:

- Naming conventions may vary between FortiGate models. For example, on some models the hardware switch interface used for the local area network is called *lan*, while on other units it is called *internal*.
- Certain features are not available on all models. Additionally, a particular feature may be available only through the CLI on some models, while that same feature may be viewed in the GUI on other models. If you believe your FortiGate model supports a feature that does not appear in the GUI, go to *System* > *Feature Visibility* and confirm that the feature is enabled. For more information, see Feature visibility on page 3952.

Low encryption models

Some FortiGate models support a low encryption (LENC) license. With an LENC license, FortiGate devices are considered low encryption models and are identified by *LENC*, for example FG-100E-LENC.

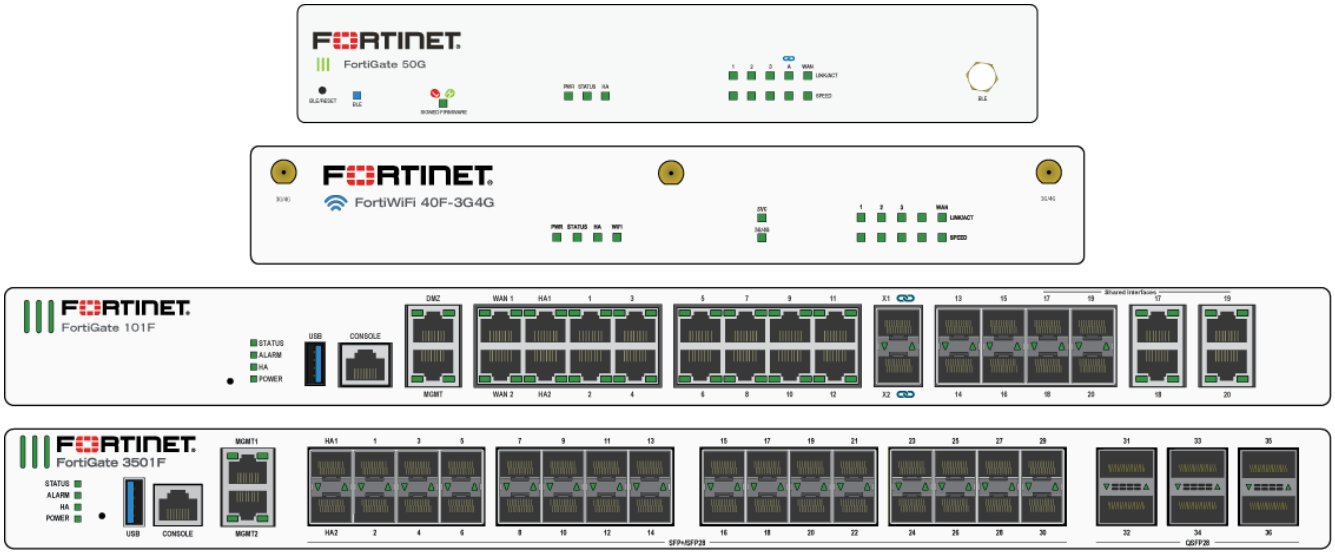
LENC models cannot use or inspect high encryption protocols, such as 3DES and AES. LENC models only use 56-bit DES encryption to work with Agentless VPN and IPsec VPN, and they are unable to perform SSL inspection.

For a list of FortiGate models that support an LENC license, see FortiGate LENC Models.

LEDs

Check your device's QuickStart guide for specific LED information: FortiGate QuickStart Guides.

The following faceplates show examples of where the LEDs are typically found on FortiGate models:



The following tables list the possible LED states and their meanings. Not all devices include every LED or support every state.

LED	State	Description
Logo	Green or Blue	The unit is on
	Off	The unit is off

LED	State	Description
Power (PWR)	Green	The unit is on and/or both power supplies are functioning
	Amber or Red	Only one power supply is functional
	Flashing Amber or Red	Power failure
	Off	The unit is off
Status (STA)	Green	Normal
	Flashing Green	Booting up
	Amber	Major or minor alarm
	Red	Major alarm
	Flashing Amber or Red	BLE is on
	Off	The unit is off
Bypass (BYP)	Amber	Bypass Port Pair is active
	Off	Bypass Port Pair is off
Alarm	Red	Major alarm
	Amber	Minor alarm
	Off	No alarms
HA	Green	Operating in an HA cluster
	Amber or Red	HA failover
	Off	HA disabled
Max PoE	Green, Amber, or Red	Maximum PoE power allocated
	Off	PoE power available or normal
PoE	Green	Power delivered
	Flashing Green	Error or PoE device requesting power
	Off	No PoE device connected or no power delivered
SVC	Green	SVC is on
	Flashing Green	SVC activity
	Off	SVC is off

LED	State	Description
3G4G, 3G / 4G	Green	3G / 4G service is on
	Flashing Green	3G / 4G activity
	Off	3G / 4G service is off
WiFi	Green	WiFi connected
	Flashing Green	WiFi activity
	Off	WiFi is off
DSL LINK/ACT	Green	DSL connected
	Flashing Green	DSL activity
	Off	DSL not connected
xDSL	Green	VDSL connected
	Amber	ADSL connected
	Off	DSL not connected
BLE	Blue	BLE on
	Flashing Blue	BLE in discovery
	Off	BLE off
Signed Firmware[1]	Green	High: Unsigned firmware blocked (default)
	Red	Low: Unsigned firmware allowed with a warning
Power supplies and fans	See your device's QuickStart guide for power supply and fan LED information: FortiGate QuickStart Guides.	

[1] See BIOS-level signature and file integrity checking on page 4005 for more information.

Port LEDs

LED	State	Description
Ethernet and SFP	Solid color	Connected
	Flashing color	Transmitting and receiving data
	Off	No link established
Speed	Green	Connected at maximum speed
	Amber	Connected at a midrange speed
	Off	Not connected or connected at minimum speed

LED	State	Description
PoE	Green	PoE power on or PoE device receiving power
	Amber	Providing power
	Red	Connected but not powered
	Off	PoE power off or no device receiving power

Alarm levels

Minor alarm

Also called an IPMI non-critical (NC) alarm, it indicates a temperature or power level outside of the normal operating range that is not considered a problem. For a minor temperature alarm, the system could respond by increasing the fan speed. A non-critical threshold can be an upper non-critical (UNC) threshold (for example, a high temperature or a high power level) or a lower non-critical (LNC) threshold (for example, a low power level).

Major alarm

Also called an IPMI critical or critical recoverable (CR) alarm, it indicates that the system is unable to correct the cause of the alarm, and that intervention is required. For example, the cooling system cannot provide enough cooling to reduce the temperature. It can also mean that the conditions are approaching the outside limit of the allowed operating range. A critical threshold can also be an upper critical (UC) threshold (such as a high temperature or high power level) or a lower critical (LC) threshold (such as a low power level).

Critical alarm

Also called an IPMI non-recoverable (NR) alarm, it indicates that the system has detected a temperature or power level that is outside of the allowed operating range and physical damage is possible.

Proxy-related features not supported on FortiGate 2 GB RAM models

As part of improvements to enhance performance and optimize memory usage on FortiGate models with 2 GB RAM or less, FortiOS 7.6.5 and later no longer supports most proxy-related features. For a list of affected models, see the *FortiOS 8.0.0 Release Notes*.

FortiGate VMs are not affected by the size of the memory and will continue to support proxy-related features after upgrading to FortiOS 7.4.4 or later. However, it is recommended to have at least 4 GB of RAM for proper operation.

After upgrade to FortiOS 7.6.5 or later, the following proxy features are no longer supported on impacted devices:

- Zero Trust Network Access (ZTNA)

  This includes all ZTNA objects and functionality. However, starting with FortiOS 7.6.3, ZTNA tags in IP/MAC-based access control are now supported. For example, users can enable `ztna-status`, and use `ztna-ems-tag` and `ztna-geo-tag`.
- UTM profile with proxy-based inspection mode, except for email protocols
- Firewall policy with proxy-based inspection mode, except for email protocols
- Explicit and transparent proxies
- Layer 7 Virtual server load balance protocols (HTTP, HTTPS, IMAPS, POP3S, SMTPS and SSL)
- Proxy-only UTM profiles:
  - Video Filter
  - Inline CASB
  - ICAP
  - Web application firewall (WAF)
  - SSH Filter
- WAN optimization

In FortiOS 7.6.0 or later, the following proxy features are no longer supported on impacted devices:

- Within `firewall profile-protocol-options`, SSH and MAPI are not supported.
- Within `firewall profile-protocol-options`, NNTP is still supported for flow mode, but the `splice` option is no longer supported.
- Within `firewall ssl-ssh-profile`, SSH is not supported.

FortiOS 7.6.5 and later brings back proxy-based inspection for email protocols on FortiGate models with 2 GB RAM. This covers the following services:

- SMTP(s)
- POP3(s)
- IMAP(s)
- NNTP

Firewall policies can once again support proxy-based inspection mode when users select one or more of the above services in the firewall policy. See Proxy-based inspection for email protocols supported on models with 2 GB RAM on page 2218 for details.

To confirm whether your FortiGate model has 2 GB RAM or less, enter `diagnose hardware sysinfo conserve` in the CLI. If the total RAM value is below 2000 MB (1000 MB = 1 GB), then your device has 2 GB RAM or less.

Upgrading from previous firmware versions

Before starting the upgrade from a firmware version that supports proxy-related features to FortiOS 7.6.5 or later that no longer supports most proxy-related features on FortiGate 2 GB RAM models, it is crucial that you carefully review the following upgrade scenarios. The scenarios provide important information about the upgrade process and its potential impacts. Please proceed with the upgrade only after you fully understand and are comfortable with the conditions and potential outcomes outlined in these upgrade scenarios.

Previous version	Upon upgrade to FortiOS 7.6.5 or later
Proxy-based inspection mode is enabled on a firewall policy.	Proxy-based inspection mode for email protocols is retained. However inspection mode for unsupported protocols is converted to flow mode.
Proxy-based inspection mode is enabled on a firewall policy with proxy-only UTM profiles, such as WAF applied.	Inspection mode is converted to flow mode, and the proxy-only UTM profiles are removed. Proxy-only UTM profiles are no longer supported.
Proxy-related settings are configured on a security profile, such as Content Disarm on an AntiVirus Profile.	The security profile is converted to flow-based, and the proxy-related setting is no longer available.
A proxy-only feature, such as ZTNA, explicit proxy or WAN optimization, is enabled.	The proxy-only configuration is removed.

Before initiating the firmware upgrade process, it is crucial to create a backup of the current working configuration. This step ensures that you have a fallback option in case of any unforeseen issues during the upgrade.

Once you have secured a backup, you can proceed with the upgrade process. After the upgrade has been successfully completed, it is highly recommended to thoroughly review all your policies.

This review process lets you confirm that all the policies that you expect to be in place are present and will function as intended. Ensure any settings that are removed do not impact the security of your firewall policy. See the Best Practices guide for more information.

GUI features not supported on FortiGate 2 GB RAM models

Fortinet has optimized memory usage on physical FortiGate devices with 2 GB of RAM to ensure smooth performance and reliability by adjusting memory used by some GUI features. This change prioritizes device stability and reduces the risk of performance issues.

Changes:

- **Removed CLI Command**: The `gui-rest-api-cache` command (under `config system global`) is no longer available. This command was used to enable/disable REST API result caching on FortiGate.
- **Streamlined GUI Pages**: The following pages in the *Security Fabric* section have been removed:
  - Physical Topology
  - Logical Topology
  - Security Rating
- **Security Rating**: With *Security Rating* page removed, security insights no longer show on pages, such as firewall policy, interface, or address object. Security rating event logs are no longer generated. However, PSIRT advisory warnings and critical vulnerability warnings are still visible in the GUI.
- **Visibility in Security Fabric topologies**: Since physical FortiGate devices with 2 GB RAM do not support topology features, any downstream physical FortiGate with 2 GB RAM will not appear in the *Physical Topology* or *Logical Topology* pages of the upstream root device's GUI, even if the upstream device has

more than 2 GB of RAM.

To verify the downstream device's connection, use the following CLI on the upstream FortiGate instead:

```
# diagnose sys csf downstream
```

- **Retired REST API Endpoints**: The `/api/v2/monitor/system/security-rating/*` endpoints are no longer supported.

These adjustments free up critical memory resources to help your FortiGate device operate more efficiently and reliably and to remain stable under typical workloads.

These changes apply exclusively to physical FortiGate devices with 2 GB of RAM, leaving devices with higher memory unaffected. Specifically, the impacted models are:

- FortiGate/FortiWiFi 40F and 60F series and their variants
- FortiGate-Rugged 60F (only the 2 GB versions)

To confirm whether your FortiGate model has 2 GB RAM, enter `diagnose hardware sysinfo conserve` in the CLI. If the total RAM value is around 2000 MB (1000 MB = 1 GB), then your device has 2 GB RAM.

FGR-70G/70F/70F-3G4G GPIO/DIO module

FortiGate Rugged 70G, 70F, 70F-3G4G include a general purpose input output (GPIO) module, also known as a digital I/O (DIO) module. This module activates a digital output when triggered by a change in any digital input. For example, the digital input can be connected to a cabinet door to monitor the open/close status or low/high voltage status, and the output can be connected to a buzzer. When the DIO module detects a change from open to closed or a voltage change from low to high, it triggers the buzzer.

CLI for configuring DIO module alarms is available only on FortiGate Rugged 70F and FortiGate Rugged 70F-3G4G devices.

Use the `config system digital-io` command to configure the input status for the DIO module to monitor:

```
config system digital-io
    set input1-detection-mode {default | voltage}
    set input2-detection-mode {default | voltage}
    set output-keep-last-state {enable | disable}
end
```

`input1-detection-mode {default | voltage}`	Configure the input mode: • `default`: Detect change from open to closed or closed to open. • `voltage`: Detect change from low to high voltage or high to low voltage.
`output-keep-last-state {enable | disable}`	Enable/disable FortiGate to keep the alarm status after a reboot.

Use the `execute digital-io set-output` command to configure the output mode when an alarm is triggered, namely, the state of the normally closed to common (`NC_COM`) output and the normally open to common (`NO_COM`) output:

```
# execute digital-io set-output
alternating     Alternates between default and opposite.
default         NC_COM=closed and NO_COM=open.
opposite        NC_COM=open and NO_COM=closed.
```

Use the `diagnose sys digital-io state` command to check the input and output status reported by the DIO module:

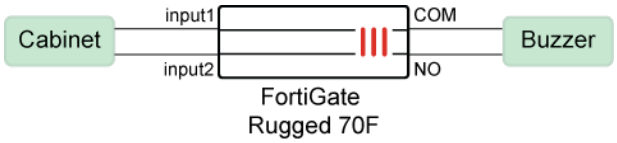
```
# diagnose sys digital-io state
Input1:    mode=default(open/closed) and state=open.
Input2:    mode=voltage(low/high) and state=low.
Output:    state=default, NO_COM=open, and NC_COM=closed.
output-keep-last-state: enable
```

New commands are also available to trigger SNMP traps and automation stitches for the DIO module. See SNMP traps and automation-stitch notifications for DIO module on page 3887 for more information.

For more information about the DIO module, see the FortiGate Rugged 70F Series QuickStart Guide and the Technical Tip: Overview of the Digital Input/Output (DIO) Module in FortiGate Rugged 70F Series community article.

Example

In this example, a FortiGate Rugged 70F is configured to monitor the open/close and low/high voltage status of a cabinet door, and the output is connected to a buzzer. When the status of the cabinet door changes, FortiGate triggers the buzzer



To configure the DIO module alarm:

1. Configure the input-detection mode:

   In this example, the input-detection mode for input1 is set to `default` and input2 is set to `voltage`, and the last output state is retained if FortiGate reboots.

   ```
   config system digital-io
       set input1-detection-mode default
       set input2-detection-mode voltage
       set output-keep-last-state enable
   end
   ```

2. Configure the output mode:

   In this example, the output is set to `default`. When triggered, the output is in a default state with the normally open to common (`NO_COM`) output being open, and the normally closed to common (`NC_COM`) output being closed. The output is triggered when the DIO module detects a change in the status of the inputs or detects an alarm event.

   ```
   # execute digital-io set-output default
   ```

3. View the input/output status being reported by the DIO module:
   In this example, the default state of `Input1` is `open`, and the default state of `Input2` is `low` voltage, which means the cabinet door is open, and the voltage is low.

```
# diagnose sys digital-io state
Input1: mode=default(open/closed), state=open
Input2: mode=voltage(low/high), state=low
Output: state=default, NO_COM=open, NC_COM=closed
output-keep-last-state: enable
```

4. Close the cabinet door.
5. View the input/output status being reported by the DIO module:
   The state of `Input1` has changed to `closed`, and the state of `Input2` has changed to `high` voltage.

```
# diagnose sys digital-io state
Input1: mode=default(open/closed), state=closed
Input2: mode=voltage(low/high), state=high
Output: state=default, NO_COM=open, NC_COM=closed
output-keep-last-state: enable
```

   The change in state triggers the buzzer.

Default SD-WAN configuration on FortiGate models with two WAN ports

FortiGate models with two WAN ports have the following added to their default configuration:

- Both WAN ports are set to DHCP mode.
- An SD-WAN (sd-wan) zone is created, and both WAN ports are added as members.
- Default firewall policy utilizes the SD-WAN zone.
- An SLA is created, utilizing IP addresses 1.1.1.1 and 9.9.9.9 for internet connectivity evaluation.

To view the default configuration:

1. On a supported device, view the SD-WAN configuration:
   - An SD-WAN (`sd-wan`) zone is created.
   - The zone members include `wan1` and `wan2`.
   - An SLA (`Default_Ping`) is created.

```
show system sdwan
config system sdwan
    set status enable
    config zone
        edit "sd-wan"
        next
    end
    config members
        edit 1
```

```
            set interface "wan1"
            set zone "sd-wan"
        next
        edit 2
            set interface "wan2"
            set zone "sd-wan"
        next
    end
    config health-check
        edit "Default_Ping"
            set server "1.1.1.1" "9.9.9.9"
            set members 0
        next
    end
end
```

2. View the interface settings for wan1 and wan2 to see the mode is set to DHCP:
   - View settings for wan1:

```
show system interface  wan1
config system interface
    edit "wan1"
        set vdom "root"
        set mode dhcp
        set allowaccess ping fgfm
        set type physical
        set role wan
        set snmp-index 3
        config ipv6
            set ip6-mode dhcp
            set ip6-allowaccess ping
        end
    next
end
```

   - View settings for wan2:

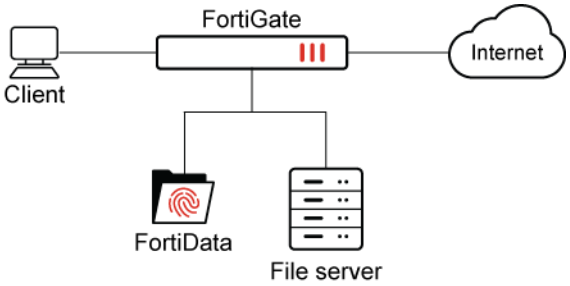
```
show system interface  wan2
config system interface
    edit "wan2"
        set vdom "root"
        set mode dhcp
        set allowaccess ping
        set type physical
        set role wan
        set snmp-index 4
        config ipv6
            set ip6-mode dhcp
            set ip6-allowaccess ping
        end
    next
end
```

3. View the default firewall policy that utilizes the SD-WAN zone.

```
show firewall policy
config firewall policy
    edit 1
        set uuid 119c2598-b5ab-51f0-2303-861480a28741
        set srcintf "lan"
        set dstintf "sd-wan"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

FortiData

FortiData is a data security product for discovering, classifying, and labeling files with sensitive data within your file storage system. Integration of FortiData with FortiGate lets you configure FortiGate to pass the fingerprint of transferred files to FortiData to check the label of the file. The labeling result is then returned and used for DLP processing in FortiGate policies



Integration requires FortiData 7.6.1 or later and FortiOS 7.6.4 or later. See the *FortiData Administration Guide* for more information about FortiData.

Following is a summary of information available to integrate FortiData with FortiOS:

1. On FortiData, configure a fabric connector to FortiGate, and add FortiGate as an allowed device. See Connecting FortiData to FortiGate on page 134.
2. On FortiGate, enable FortiData and establish a connection with FortiData. See Connecting to FortiData on page 4369.
3. On FortiData, discover, classify, and label files with sensitive data in a file storage system. See the *FortiData Administration Guide* for information.
4. Configure FortiGate to use the FortiData labels. See FortiData labels on page 2662.

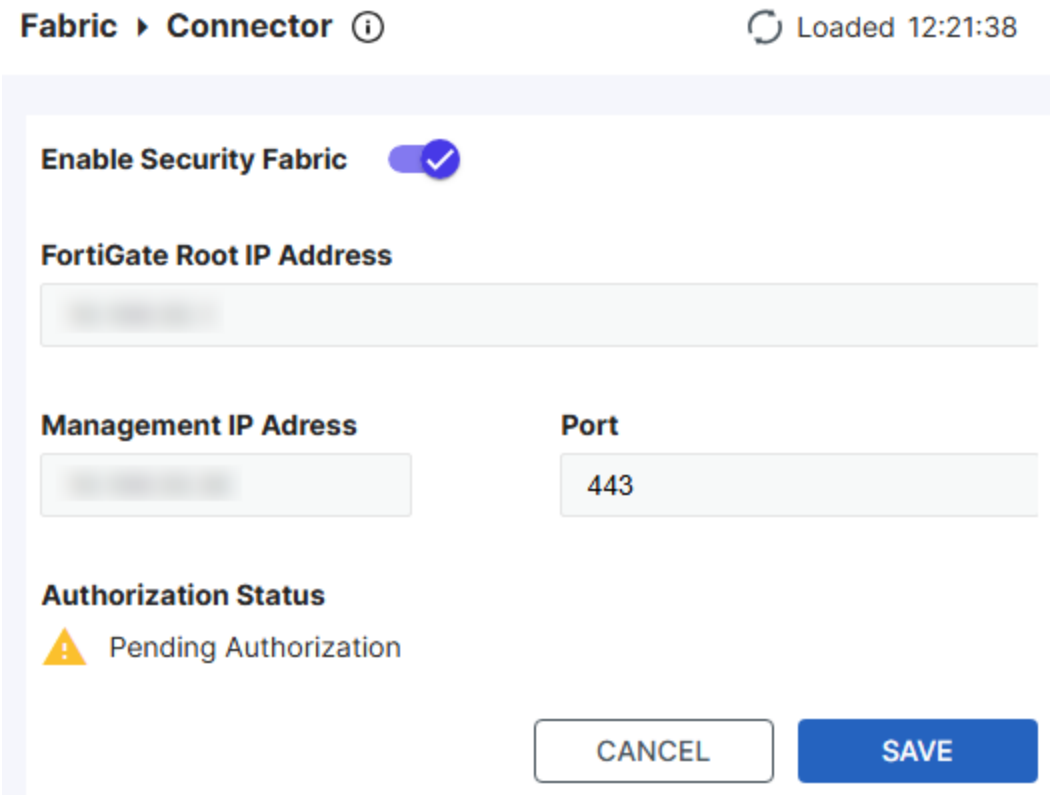
Connecting FortiData to FortiGate

On FortiData 7.6.1 or later, configure a fabric connector for FortiGate, identify a TLS certificate, and configure the HTTP2 service. Add FortiGate as an allowed device in FortiData.

After configuring the fabric connector in FortiData and adding FortiGate as an allowed device, go to FortiGate and complete the connection. See Connecting to FortiData on page 4369.

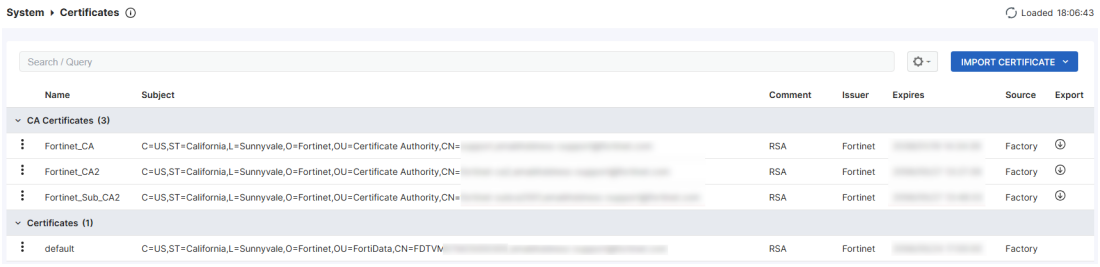
To configure FortiData in the GUI:

1. In FortiData on the *Fabric* > *Connector* page, enable security fabric, specify the details of the FortiGate, and click *SAVE*.

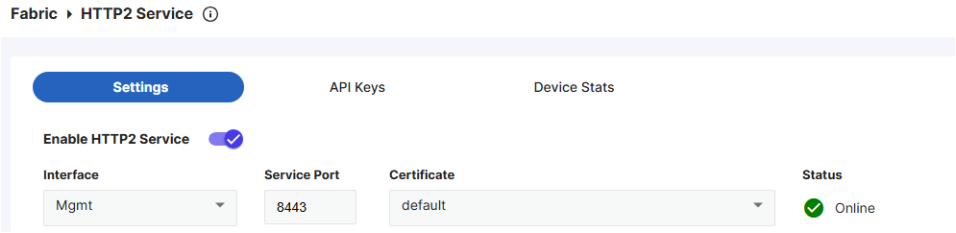


   The authorization status is shown as *Pending Authorized* at this point as the FortiGate configuration is not complete yet.

2. On the *System* > *Certificates* page, verify that at least one TLS certificate is listed. FortiData includes a default certificate that you can use.

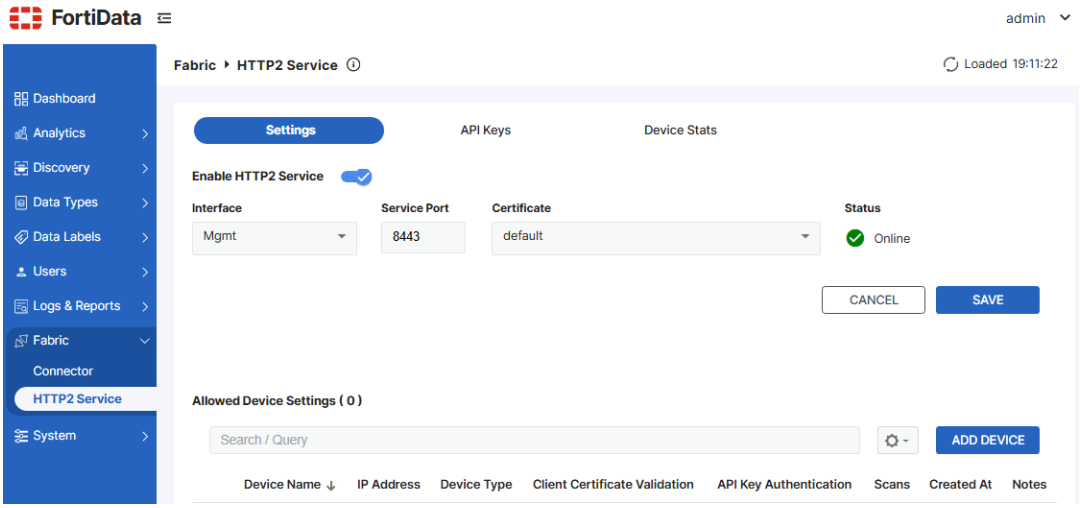


3. On the *Fabric > HTTP2 Service* page, ensure you have selected a valid certificate, which is required for the HTTPS connection between the FortiGate and FortiData. The default certificate will be used if you do not have any custom ones imported.

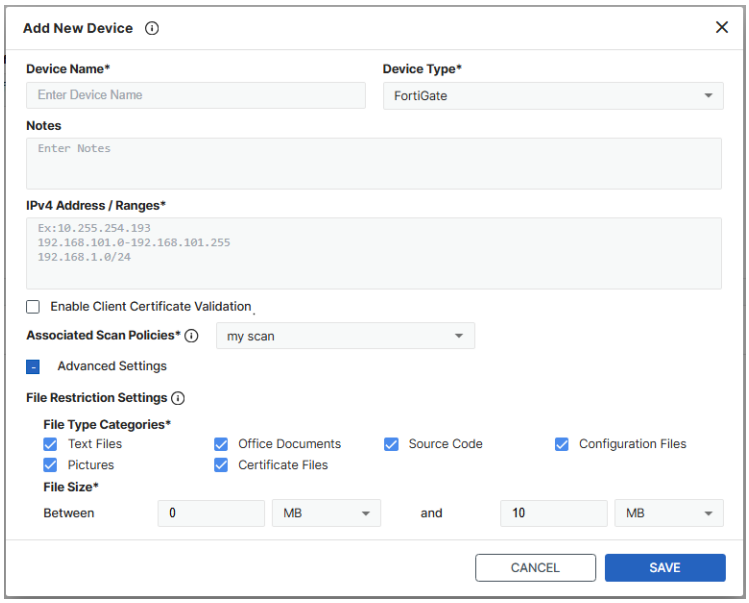


To add FortiGate as an allowed device in the GUI:

1. Go to *Fabric > HTTP2 Service*.
2. Under *Settings*, click *ADD DEVICE*.



3. From the *Device Type* dropdown list, select *FortiGate*.
4. In the *IPv4 Address / Ranges* field, enter `0.0.0.0/0` to allow all IP addresses, or enter a specific IP or IP range for the FortiGate to restrict access for security reasons.



5. From the *Associated Scan Policies* dropdown list, select the desired scan policy, or create a new one.
6. Configure any advanced settings as needed to narrow down the file types and size to scan.
7. Click *Save*.

FortiSASE-Sovereign

FortiSASE-Sovereign is Fortinet's sovereign SASE solution that offers a turnkey, private SASE service that ensures data remains within designated jurisdictions, while giving its customers complete control over their deployment locations, features, and infrastructure. See the *FortiSASE-Sovereign Architecture Guide* for more information.

FortiOS supports FortiSASE-Sovereign licensing bundles for the FortiGate 91G and 901G devices.

FortiSASE-Sovereign dictates the FortiOS version requirement for the devices as outlined in the following documents:

- MSSP Quick Start Guide (for multi-tenants)
- Dedicated Instance Deployment Guide (for dedicated tenants)

After applying the FortiSASE-Sovereign license in FortiOS, the following restrictions apply:

- The *Policy & Objects* pane is restricted in the GUI, and the `config firewall` command is restricted in the CLI.
- The restore option is restricted in the GUI, and the `exec restore` command is restricted in the CLI.

With the FortiSASE-Sovereign license installed and activated, go to the FortiSASE-Sovereign portal to complete the onboarding process. See Log on to FortiSASE-Sovereign for more information. Completion of the FortiSASE-Sovereign onboarding process enables sovereign SASE in FortiOS and further restricts the FortiOS GUI and CLI:

- All administrator sessions are logged out of FortiOS.
- The *Network Interfaces* pane becomes read-only in the GUI.
- Configurations can no longer be restored to FortiOS using the GUI and CLI.

After applying the license and completing the FortiSASE-Sovereign onboarding process, use FortiSASE-Sovereign to manage FortiGate configurations.

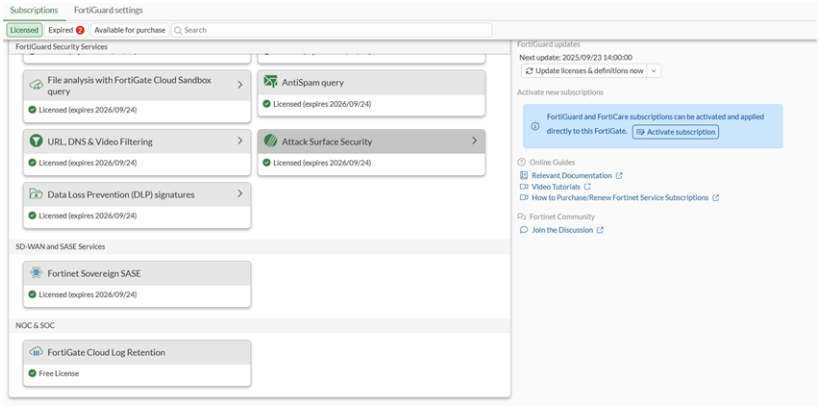
Do not manually enable sovereign SASE in FortiOS because it will cause the FortiSASE-Sovereign onboarding process to fail.

```
config system sov-sase
    set status enable
end
```

Enabling FortiSASE-Sovereign

To enable FortiSASE-Sovereign in the GUI:

1. In FortiOS, apply the FortiSASE-Sovereign license.
2. Verify the license is activated by going to *System* > *FortiGuard* to see the *Fortinet Sovereign SASE* license status.



3. In the FortiSASE-Sovereign portal, complete the onboarding process. See Log on to FortiSASE-Sovereign for more information.

   The onboarding process enables sovereign SASE on FortiOS.

   All admin sessions are logged out of FortiOS, and a confirmation dialog box is displayed:

4. In FortiOS, click *Confirm*.
   The FortiOS GUI and CLI become restricted.
5. Use FortiSASE-Sovereign portal for FortiGate configuration.

To enable FortiSASE-Sovereign in the CLI:

1. In FortiOS, apply the FortiSASE-Sovereign license.
2. In the FortiSASE-Sovereign portal, complete the onboarding process. See Log on to FortiSASE-Sovereign for more information.
   The onboarding process enables sovereign SASE on FortiOS.

```
config system sov-sase
    set status enable
end
```

All admin sessions are logged out of FortiOS, and the CLI becomes read-only, except for the following commands:

```
config system cloud-service
config system sdn-vpn
config system sov-sase
```

Disabling FortiSASE-Sovereign

Disabling sovereign SASE forces a factory reset of the FortiGate.

To disable FortiSASE-Sovereign:

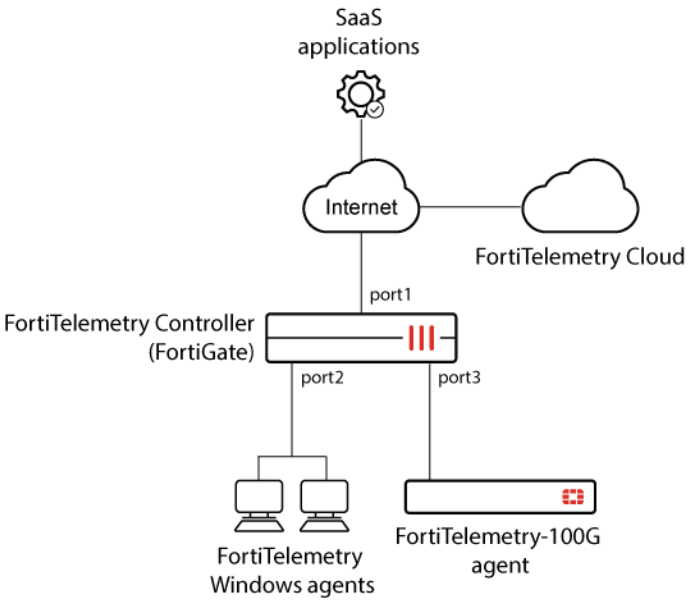
```
config system sov-sase
    set status disable
end
Sovereign SASE status will be disabled and the system will be reset to factory default state.
Do you want to continue?
```

FortiTelemetry

FortiTelemetry provides information about the SaaS user experience based on application and network performance. FortiTelemetry agents send raw application-level and network-level metrics to FortiTelemetry Cloud for data analysis. Based on its analytics, FortiTelemetry Cloud returns to the FortiGate acting as a FortiTelemetry Controller two additional metrics: application experience score and application failure rate. FortiTelemetry Controller presents the metrics on FortiTelemetry monitor pages.

A FortiTelemetry deployment consists of one FortiGate hardware model acting as FortiTelemetry Controller and one or more FortiTelemetry agents:

- The FortiTelemetry Controller manages the agents by onboarding them to the Security Fabric, managing telemetry profiles and policies, and providing monitoring tools to give administrators a view of application metrics and statistics.
- FortiTelemetry agents are on-premise FortiGate-integrated telemetry agents that continuously emulate, monitor, and detect performance metrics across SaaS applications without any user intervention or involvement.



Following is a summary of information available to deploy and configure FortiGate acting as a FortiTelemetry Controller:

1. Configure a FortiGate hardware model to act as a FortiTelemetry Controller. See Configuring a FortiTelemetry Controller on page 140.
2. Deploy FortiTelemetry agents. See the FortiTelemetry Administration Guide for details.
3. On the FortiTelemetry Controller:
   a. Authorize automatically discovered FortiTelemetry agents. See Connecting to FortiTelemetry agents on page 4364.
      A telemetry address object is automatically created for each authorized FortiTelemetry agent.
   b. View telemetry addresses, and optionally create telemetry address groups. See Telemetry addresses and address groups on page 2083.

c. Create a telemetry profile. See Creating telemetry profiles on page 2426.
d. Create a Telemetry firewall policy, select telemetry addresses or address groups, and select the telemetry profile. See Telemetry policy on page 1909.
e. Add a FortiTelemetry monitor to view the collected information. See Adding a FortiTelemetry dashboard on page 201.
f. Enable system event logs for FortiTelemetry. See FortiTelemetry events on page 4774.

For details about FortiOS CLI commands that can be used to configure the FortiTelemetry Controller, see Config CLI commands.

For details about FortiOS CLI commands that can be used to view different properties related to FortiTelemetry, see Get and Diagnose commands.

Configuring a FortiTelemetry Controller

An on-premise FortiGate hardware model running FortiOS 7.6.3 or later can be configured to act as a FortiTelemetry Controller. For information about requirements, see the *FortiTelemetry Administration Guide*.

You must supply and maintain the certificates used by FortiTelemetry Windows agents because they are not included by default.

To prepare a certificate:

1. Create a certificate authority (CA) certificate using your preferred certificate authority.
   - For details about creating a local CA certificate and user certificate using FortiAuthenticator, see Local CAs and End entities, respectively.
   - For details about creating a local CA certificate and user certificate using XCA, see Creating certificates with XCA on page 3982.
   - For details about creating a local CA certificate and user certificate using Windows Server with Active Directory Certificate Services (AD CS), see Implement and manage Active Directory Certificate Services.
2. Upload the CA certificate on the FortiGate acting as FortiTelemetry Controller:
   a. Go to *System* > *Certificates*.
   b. Click *Create/Import* > *CA Certificate*.
   c. Select *Type* as *File*, and then upload the CA certificate.
   d. Click *OK*. Once uploaded, you will see the certificate under *Remote CA Certificate*.
3. Using your certificate authority, create a user certificate, so you can import the certificate file to the Windows OS local machine that is hosting the FortiTelemetry Windows agent. The private key information is required, and the user certificate should be of the PFX or p12 file type. See the *FortiTelemetry Administration Guide* for details.

To configure the FortiTelemetry Controller:

1. Enable FortiTelemetry, and specify a data port:
   Although specified, the data port is not currently used but reserved for a future use.

```
config system global
    set telemetry-controller enable
    set telemetry-data-port 35246
end
```

2. Configure the retry interval for FortiTelemetry Cloud and the CA certificate used to validate the identity of the FortiTelemetry Windows agent:

```
config telemetry-controller global
    set region global
    set retry-interval 60
    set telemetry-ca-certificate "CA_Cert_1"
end
```

3. Enable telemetry feature visibility in the GUI:

```
config system settings
    set gui-fortitelemetry enable
end
```

4. Include `fabric` in the FortiGate interface `allowaccess`, and enable automatic discovery of FortiTelemetry agents:
   This example is for port2.

```
config system interface
    edit "port2"
        set ip 192.168.253.99 255.255.255.0
        set allowaccess ping https ssh snmp http telnet fabric
        set telemetry-discover enable
    next
end
```

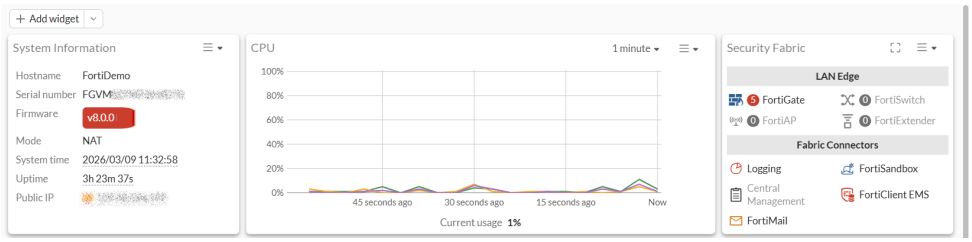
Ensure that the interface has an IP address configured when including `fabric` in the interface `allowaccess` setting.

Dashboard

A dashboard is a collection of widgets and monitors that display information about your FortiGate and network:

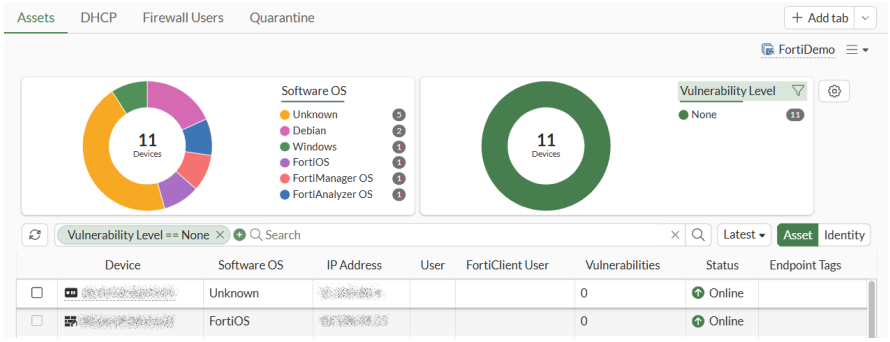
- Widgets are condensed monitors, and multiple widgets can be displayed on one tab.

  For example, the *Status* dashboard contains several widgets, such as the *System Information*, *CPU*, *Security Fabric*, and more:



- Monitors focus on the details of one area and fill an entire tab.

  For example the *Assets and Identities* dashboard contains the *Assets*, *DHCP*, *Firewall Users*, and *Quarantine* tabs of monitors:



FortiOS includes the following predefined dashboards:

Dashboard	Default display:
Status	The *Status* dashboard is one tab that displays the following widgets: • *System Information*: View the hostname, serial number, FortiOS version, mode, system time, uptime, and public IP. • *CPU*: View CPU usage over time. • *Security Fabric*: View the number and type of devices in the Security Fabric. • *Memory*: View memory usage over time. • *Sessions*: Monitor IPv4 and IPv6 sessions. • *Licenses*: View license status and expiration date. • *FortiGate Cloud*: Activate FortiGate Cloud. • *Virtual Machine*: View VM licenses and allocated vCPU and RAM.
Assets and Identities	The *Assets and Identities* dashboard contains the following tabs of monitors: • *Assets*: View users and devices connected to the network. • *DHCP*: Monitor DHCP clients.

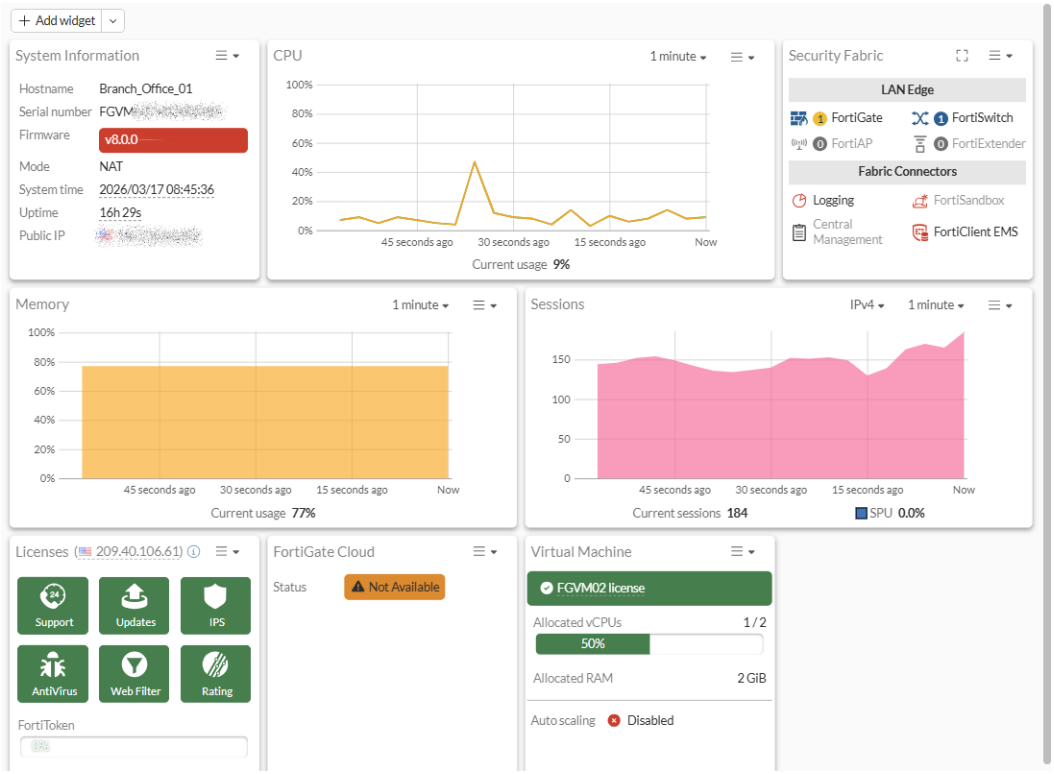
Dashboard	Default display:
	• *Firewall Users*: Monitor logged in firewall and proxy users.
Network Monitor	The *Network Monitor* dashboard contains the following tabs of monitors: • *Routing*: Monitor current routing table. • *Security*: Identify threats from individual users and devices. • *SD-WAN*: Monitor SD-WAN status. • *VPN*: Monitor IPsec VPN connections.
FortiView	The *FortiView* dashboard contains the following tabs of monitors: • *Sources*: Monitor top sources of traffic volume. • *Destinations*: Monitor top destinations by traffic volume. • *Applications*: Monitor top applications by traffic volume. • *Websites*: Monitor top websites by traffic volume. • *Policies*: Monitor top policies by traffic volume. • *Sessions*: Monitor top sessions by traffic source.
WiFi	The *WiFi* dashboard is one tab that contains the following widgets: • *FortiAP Status*: View FortiAP connections and 2.4 GHz channel utilization. • *Channel Utilization*: View channel utilization and clients. • *Clients by FortiAP*: View the number of WiFi clients

See also Monitors and widgets on page 184 and Widgets in VDOM mode on page 189.

You can edit predefined dashboards and add new dashboards of monitors and widgets. See Using dashboards of widgets and monitors on page 192.

Status dashboard

The *Status* dashboard provides an overview of your FortiGate device and the devices in your Security Fabric. If your FortiGate is a virtual machine, information about the virtual machine is also displayed in the dashboard.

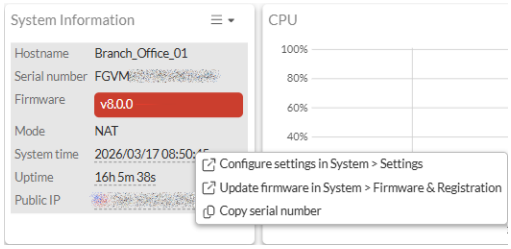


Updating system information

The *System Information* widget contains links to the *Settings* module where you can update the *System Time*, *Uptime*, and *Public IP*.

A notification will appear in the *Firmware* field when a new version of FortiOS is released. Click *Update firmware in System > Firmware & Registration* to view the available versions and update FortiOS.

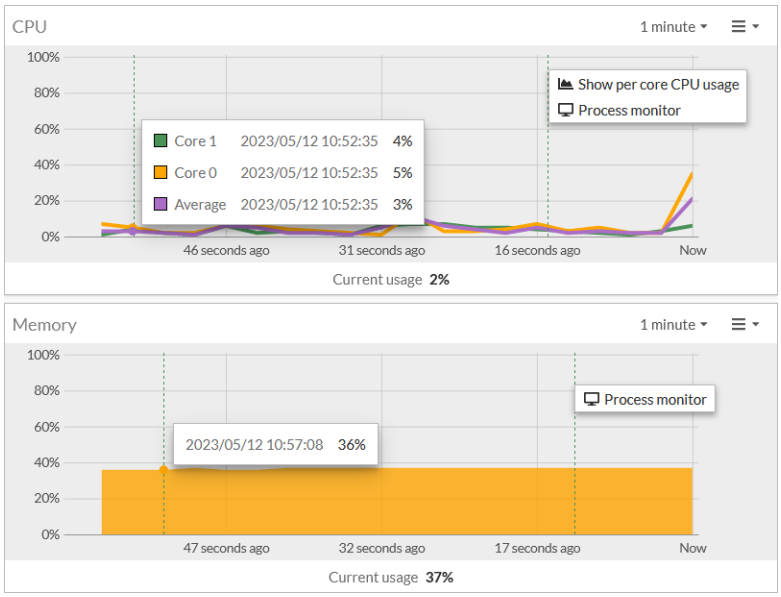
You can also click *Copy serial number* to copy the FortiGate serial number to the clipboard and paste in other applications.



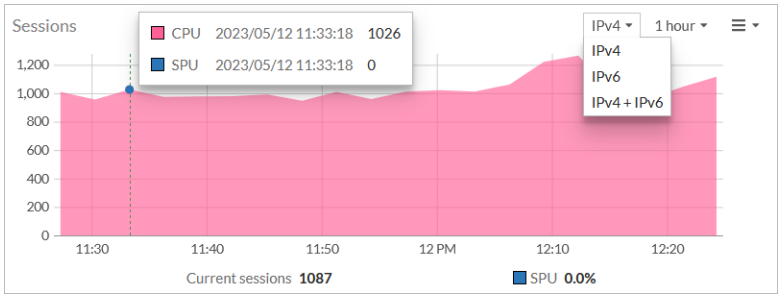
Resource widgets

The resource widgets show the current usage statistics for *CPU*, *Memory*, and *Sessions*.

Click the *CPU* widget to show the per core CPU usage or access the Process monitor.



You can switch between *IPv4*, *IPv6*, or *IPv4+IPv6* in the *Sessions* monitor.



Viewing Fabric devices

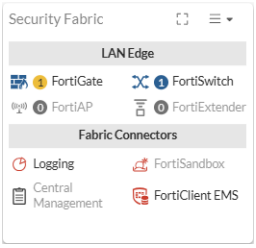
The *Security Fabric* widget provides a visual overview of the devices connected to the Fabric and their connection status. Hover of a device icon to view more information about the device.

Click a device in the Fabric to:

- View the device in the physical topology
- Register, configure, deauthorize, or log in to the device
- Open *Diagnostics and Tools*
- View the *FortiClient Monitor*

These options will vary depending on the device.

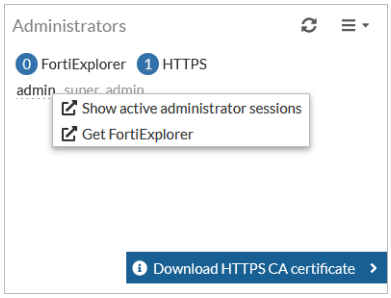
Click *Expand* to view all the devices in the Fabric at once.



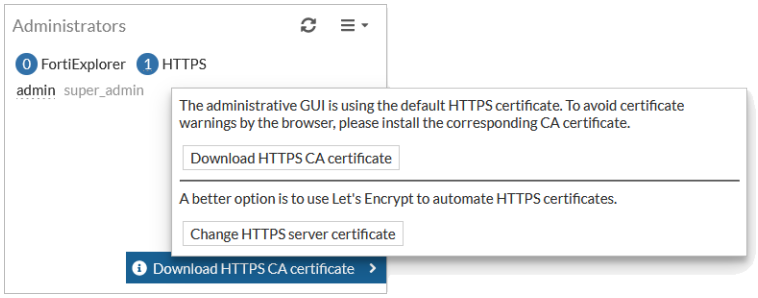
Viewing administrators

This widget is hidden by default. See Editing dashboards of widgets and monitors on page 194 for information about adding widgets.

The *Administrators* widget displays the active administrators and their access interface. Click the username to view the *Active Administrator Sessions* monitor. You can use the monitor to end an administrator's session.



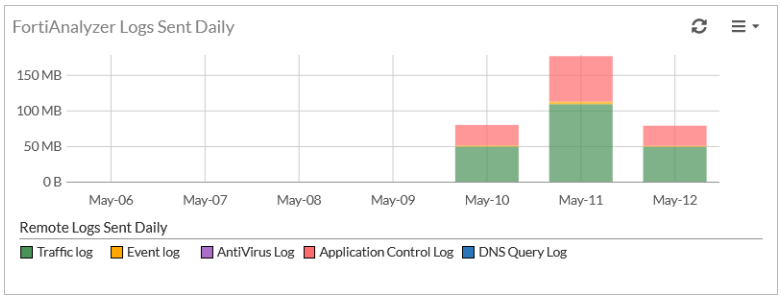
If the GUI is using the default HTTPS certificate, a warning is shown where you can download the HTTPS CA certificate or change the HTTPS server certificate.



Viewing logs sent for remote logging source

This widget is hidden by default. See Editing dashboards of widgets and monitors on page 194 for information about adding widgets.

The *Logs Sent* widget displays chart for remote logging sources (FortiAnalyzer, FortiGate Cloud, and FortiAnalyzer Cloud) sent daily.



Assets and Identities dashboard

The *Assets and Identities* dashboard includes several tabs of monitors that show the current status of users and devices connected to your network. You can view asset details, create firewall addresses, deauthenticate users, add IP addresses, ban IP addresses, quarantine hosts, and other such tasks.

The *Assets and Identities* dashboard includes the following tabs of monitors:

Tab	Description
Assets	Contains the following tabs: • Asset: Shows information from detected addresses, devices, and users on a single page. Information is grouped by device. For more information see Assets on page 147 and Asset details on page 149. • Identities: Shows information from detected addresses, devices, and users on a single page. Information is grouped by user.
DHCP	Shows the addresses leased out by FortiGate's DHCP servers. See DHCP monitor on page 154.
Firewall Users	Monitor users that are logged into the network.
Quarantine	Monitor quarantined devices.

See also Asset Identity Center page on page 4400.

Assets

You can enable device detection to allow FortiOS to monitor your networks and gather information about devices operating on those networks, including:

- MAC address
- Operating system
- IP address
- Username
- Vulnerabilities
- Status

- Endpoint tags
- Hostname
- When FortiOS detected the device and on which interface

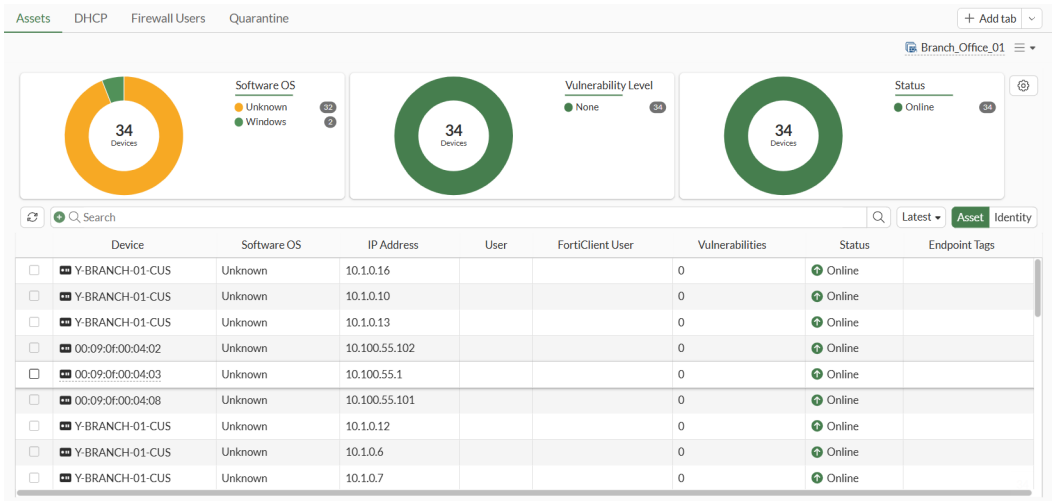
You can enable device detection separately on each interface in *Network* > *Interfaces*.

Device detection is intended for devices directly connected to your LAN and DMZ ports. The monitor is only available when your *Interface Role* is *LAN*, *DMZ* or *Undefined*. It is not available when the role is WAN.

To view the assets monitor:

1. Go to *Dashboard* > *Assets and Identities*. The *Assets* monitor opens.
2. Hover over a device to display a tooltip with information and buttons.
3. Click the *Asset Details* button to view the details. See Asset details on page 149.

To filter or configure a column in the table, hover over the column heading, and click *Filter/Configure Column*. See Assets and filtering on page 148.



Assets and filtering

The *Assets* monitor contains a series of summary charts that provide an overview of the operating system, vulnerability level, status, and interfaces. You can use these clickable charts to simplify filtering among your devices.

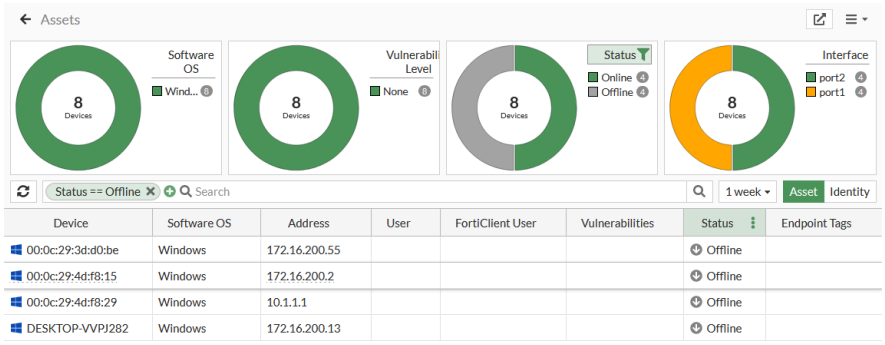
To view the device inventory and apply a filter:

1. Go to *Dashboard* > *Assets and Identities* > *Assets*. The *Assets* monitor opens.
   You can also go directly to *Security Fabric* > *Asset Identity Center* to view the *Assets*.
2. To filter a chart, click an item in the legend or chart area. The table displays the filter results.
3. To combine filters, hover over a column heading and click *Filter/Configure Column*.

4. Click the filter icon in the top-right corner of the chart to remove the filter.

To filter all offline devices:

1. In the *Status* chart, click *Offline* in the legend or on the chart itself.



To filter all devices discovered on port2:

1. Go to *Dashboard* > *Network Monitor*.
2. In the *Interfaces* chart, click *port2*.

To perform a free-style keyword search:

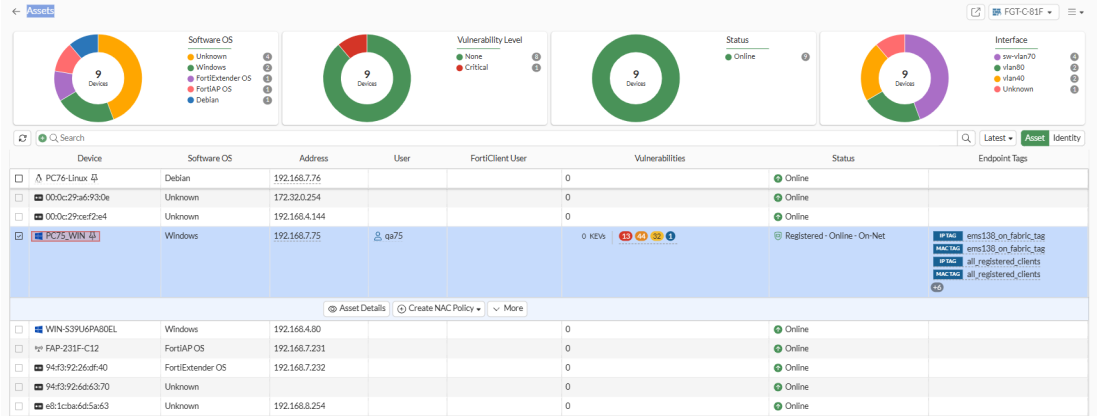
1. In the search bar, type in your search string. For example, *Windows*.
   A search is performed on all fields, and returns devices where *Software OS* contains *Windows*.
   Note that if multiple strings are entered, they are searched as a single keyword.

Asset details

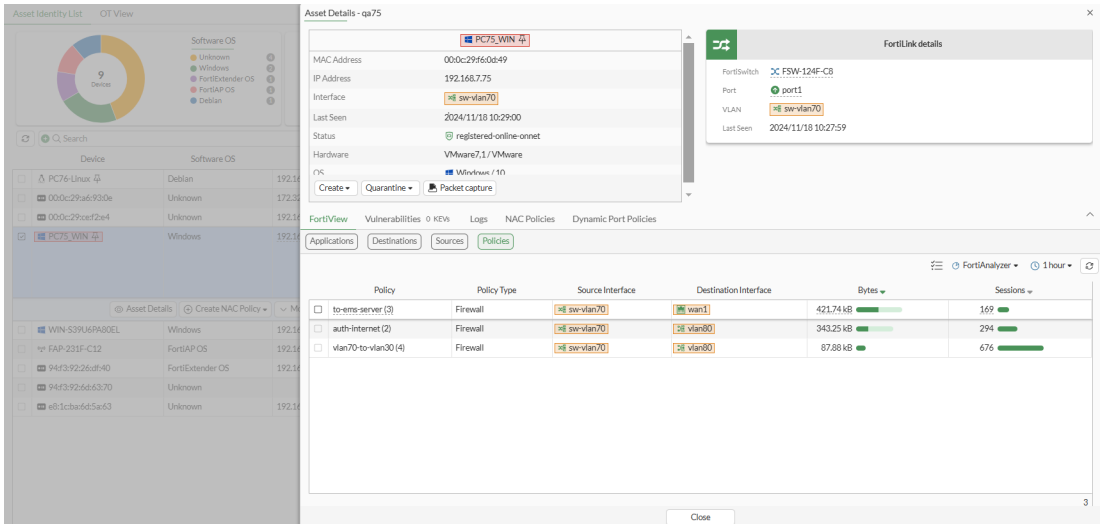
The *Asset Details* pane provides comprehensive endpoint information to streamline the diagnostic process and reduce reliance on CLI commands.

To view the Asset Details pane:

1. Go to *Dashboard* > *Assets and Identities* > *Assets*. The *Assets* monitor open
2. Select a row to display a toolbar that includes the *Asset Details* button:
   You can also hover over the device to display a tooltip with the *Asset Details* button.



3. Click *Asset Details* to display the details.



On the top-left of the *Asset Details* pane is information about the endpoint, such as MAC address, IP address, interface, and so on. The following buttons are also available for the endpoint:

- *Create > Firewall Address*
- *Create > Firewall Policy*
- *Quarantine > Quarantine Host*
- *Quarantine > Ban IP*
- *Disassociate* is available for endpoints using a WiFi interface
- *Packet capture*

On the top-right are details about the WiFi or FortiLink interface when used by the endpoint.

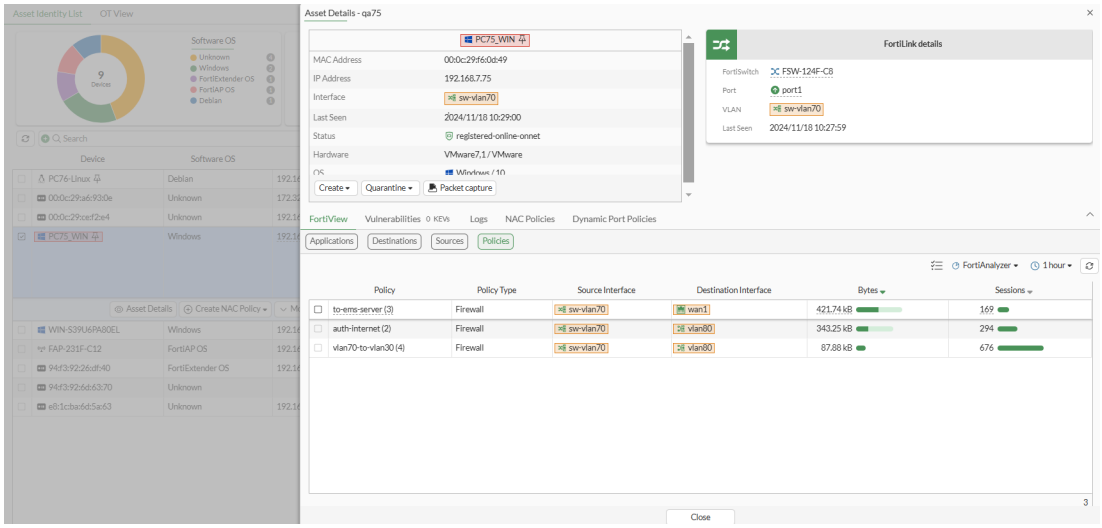
Along the bottom are tabs of information, depending on the endpoint and interface used:

Tab	Subtabs
FortiView	Applications Destinations Sources Policies
Vulnerabilities/Vulnerabilities <number> KEVs	FortiClient

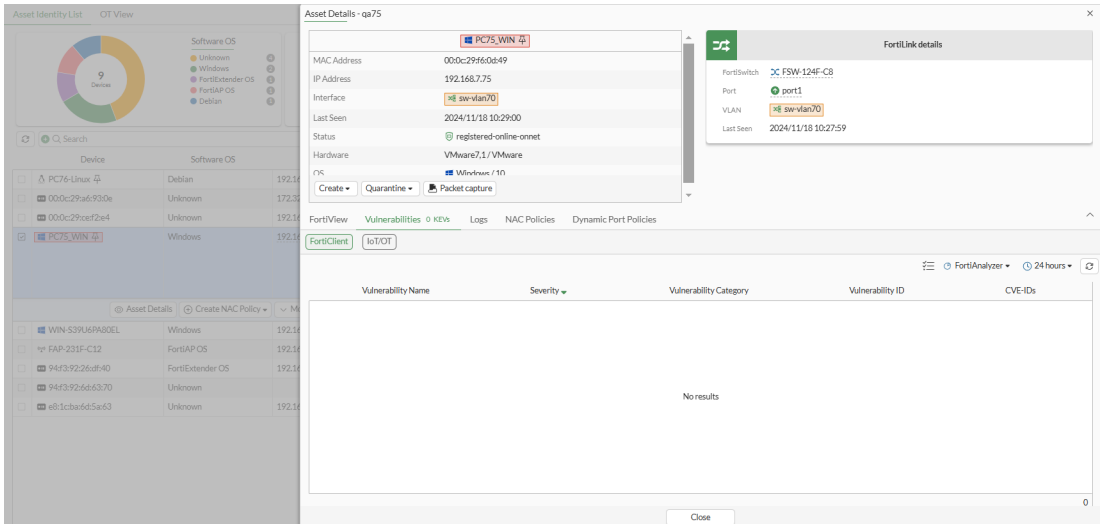
Tab	Subtabs
	IoT/OT
Logs	Forward Traffic ZTNA Traffic WiFi Events
WiFi Performance	N/A
NAC Policies	N/A
Dynamic Port Policies	N/A

Following are examples of the *Asset Details* pane for an endpoint named *PC75_WIN* using a VLAN interface:

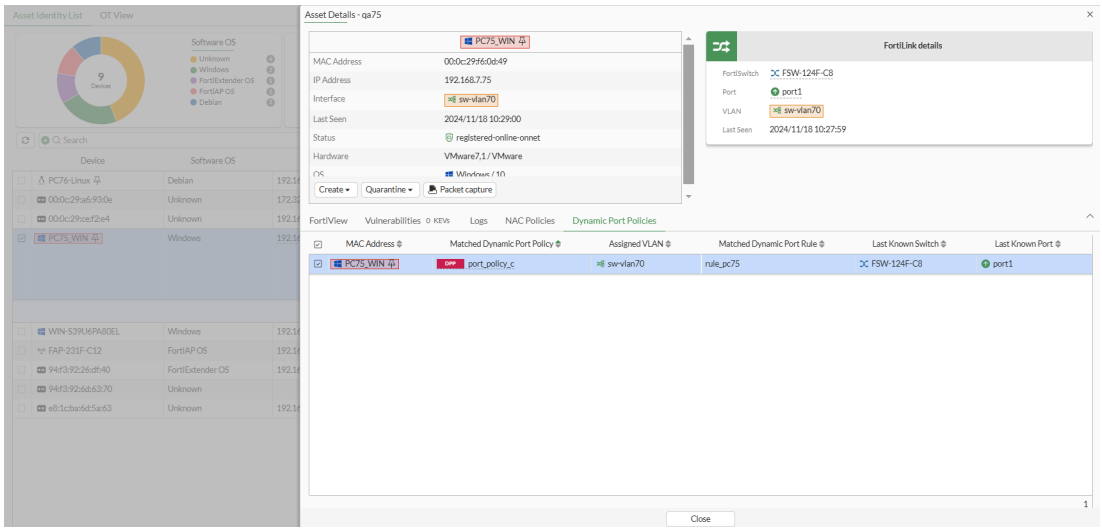
- The *FortiView > Policies* tab:



- The *Vulnerabilities > FortiClient* tab:

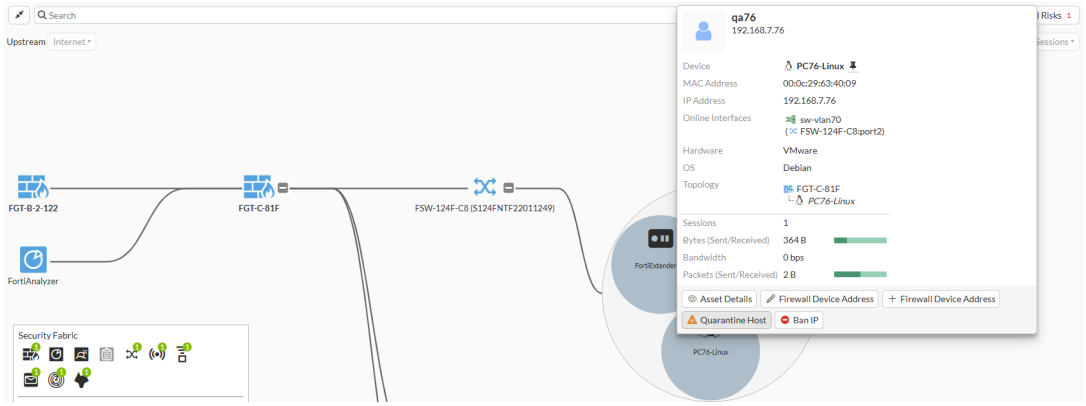


- The *Dynamic Port Policies* tab:

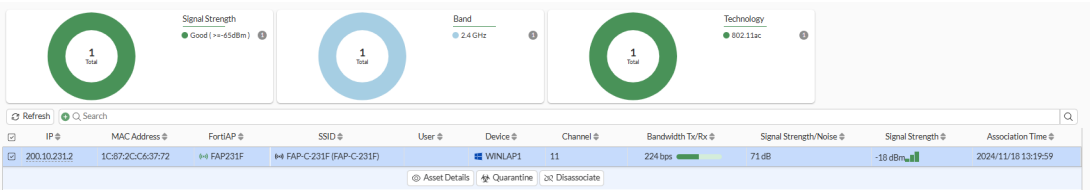


You can also access the *Asset Details* pane by clicking the *Asset Details* button from the following GUI locations:

- On the *Dashboard* > *Assets and Identities* page:
  - *Assets - Vulnerabilities*
  - *Assets - FortiClient*
- On the *Security Fabric* page:
  - Go to *Topology*, and hover over an endpoint to display a tooltip that includes the *Asset Details* button.

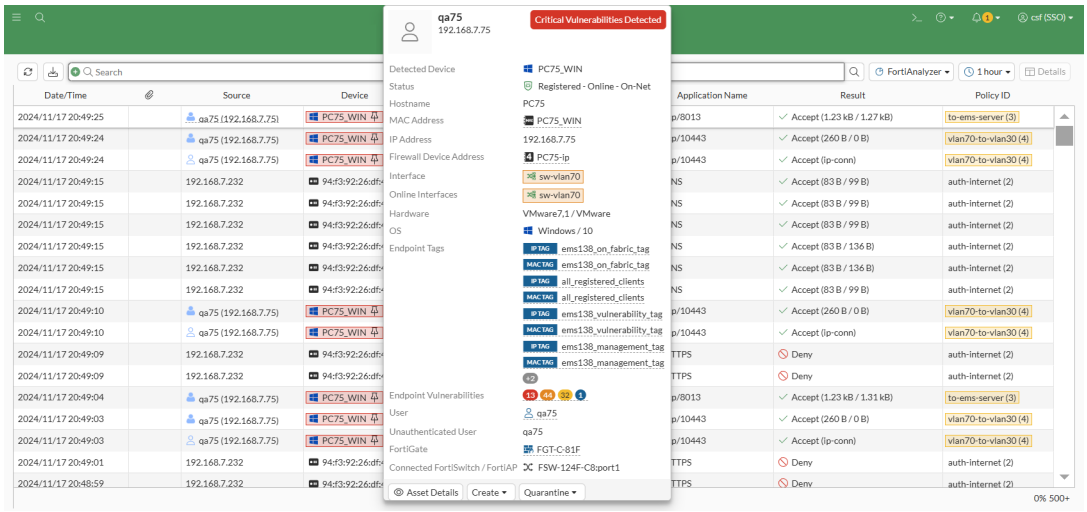


  - Go to *Asset Identity Center* page. Hover over a device to display a tooltip that includes the *Asset Details* button, or select a row in the table to display a toolbar that includes the *Asset Details* button.
- In the *WiFi & Switch Controller* menu:
  - Go to *WiFi Clients*. Hover over a device to display a tooltip that includes the *Asset Details* button, or select a row in the table to display a toolbar that includes the *Asset Details* button.



  - Go to *FortiSwitch Clients*. Hover over a device to display a tooltip that includes the *Asset Details* button, or select a row in the table to display a toolbar that includes the *Asset Details* button.

- On the *Log & Report* > *Forward Traffic* page, hover over a device to display a tooltip that includes the *Asset Details* button:

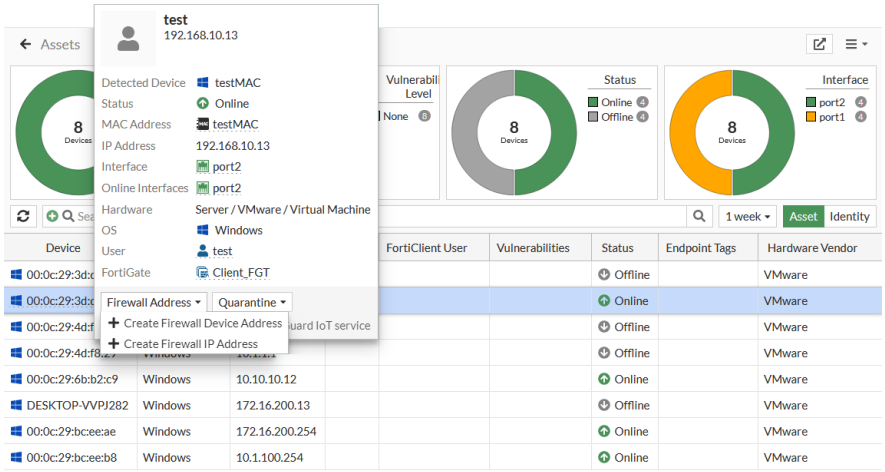


Adding MAC-based addresses to devices

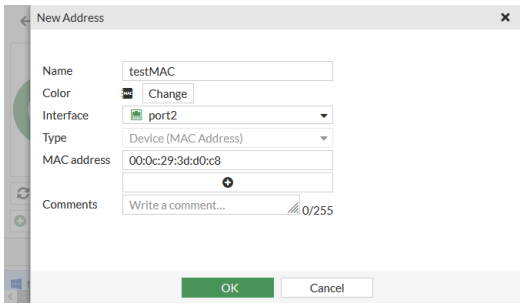
Assets detected by device detection appear in the *Assets* monitor. You can manage policies around devices by adding a new device object (MAC-based address) to a device. Once you add the MAC-based address, the device can be used in address groups or directly in policies.

To add a MAC-based address to a device:

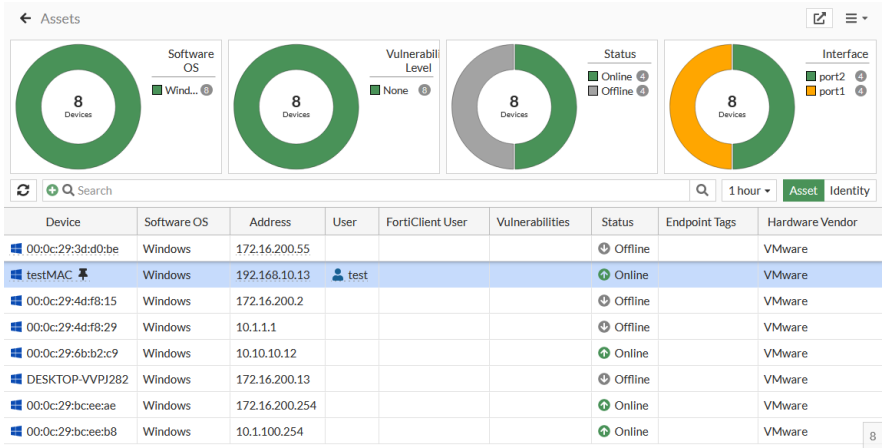
1. Go to *Dashboard* > *Assets & Identities* > *Assets*. The *Assets* monitor opens.
2. Click a device, then click *Create* > *Firewall Address* > *Create Firewall IP Address*. The *New Address* pane opens.



3. In the *Name* field, give the device a descriptive name so that it is easy to find it in the *Device* column.
4. Configure the *MAC Address*.

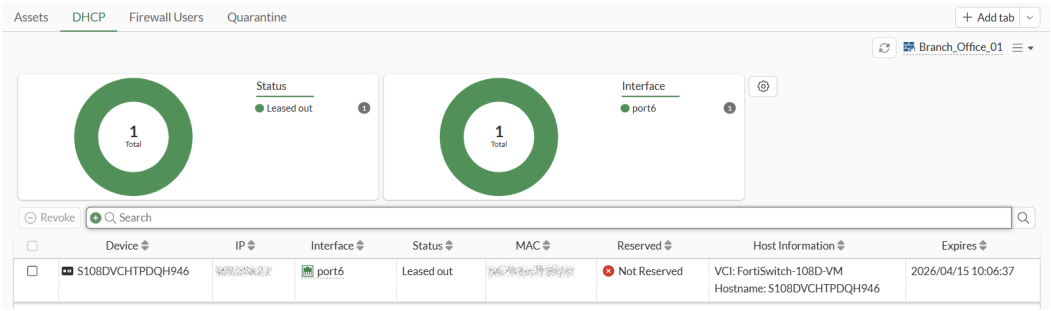


5. Click *OK*, then refresh the page. The MAC address icon appears in the *Address* column next to the device name.



DHCP monitor

The DHCP monitor shows all the addresses leased out by FortiGate's DHCP servers. You can use the monitor to revoke an address for a device, or create, edit, and delete address reservations.



To view the DHCP monitor:

1. Go to *Dashboard* > *Asset and Identities* > *DHCP*.

To filter or configure a column in the table, hover over the column heading and click the *Filter/Configure Column* button.

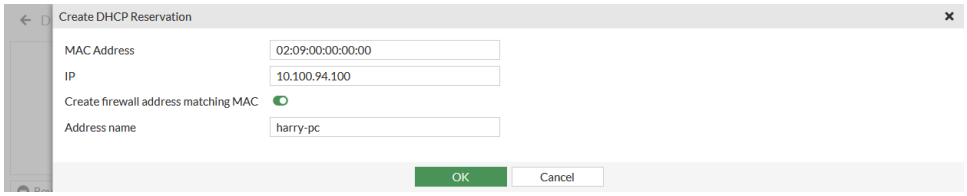
To revoke a lease:

1. Select a device in the table.
2. In the toolbar, click *Revoke*, or right-click the device, and click *Revoke Lease(s)*. The *Confirm* page is displayed.
3. Click *OK*.

> A confirmation window opens only if there is an associated address reservation. If there is no address, the lease will be removed immediately upon clicking *Revoke*.

To create a DHCP reservation:

1. Select a server in the table.
2. In the toolbar, click *Reservation* > *Create DHCP Reservation*, or right-click the device and click *Create DHCP Reservation*. The *Create New DHCP Reservation* page is displayed.
3. Configure the DHCP reservation settings.



4. Click *OK*.

To view top sources by bytes:

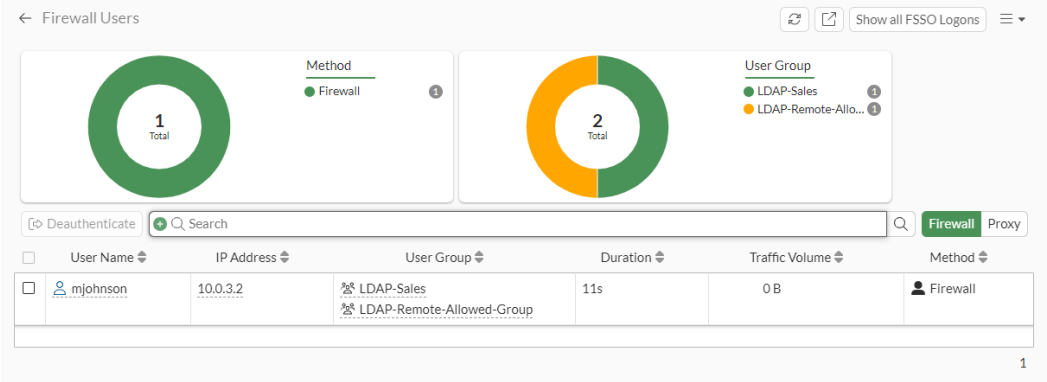
1. Right-click a device in the table and click *Show in FortiView*. The *FortiView Sources by Bytes* widget is displayed.

To view the DHCP lease list in the CLI:

```
# execute dhcp lease-list
```

Firewall Users monitor

The *Firewall Users* monitor displays all currently logged in firewall and proxy users. You can use the monitor to diagnose user-related logons or to highlight and deauthenticate a user.



To view the firewall monitor:

1. Go to *Dashboard* > *Assets and Identities* > *Firewall Users*.
2. On the *Firewall* tab, click *Show all FSSO Logons* at the top-right of the page to show FSSO logons.
3. To switch to the proxy user view, click *Proxy* (next to the search bar). Proxy user view shows used that authenticated over ZTNA and explicit proxy.

To filter or configure a column in the table, hover over the column heading and click the *Filter/Configure Column* button.

When the number of users exceeds 2000, the page no longer displays the search bar, filter columns, or graphs because lazily-loaded results do not support these GUI features.

To deauthenticate a user in the GUI:

1. Go to *Dashboard* > *Assets and Identities* > *Firewall Users*.
2. (Optional) Use the *Search* field to search for and select a specific user.
3. In the toolbar, click *Deauthenticate*, or right-click the user, and click *Deauthenticate*. The *Confirm* dialog is displayed.
4. Click *OK*.

To view and deauthenticate firewall users in the CLI:

```
# diagnose firewall auth list
# diagnose firewall auth filter <parameters>
# diagnose firewall auth clear
```

To view and deauthenticate proxy users in the CLI:

```
# diagnose wad user list
# diagnose wad user clear <ID> <IP|IPv6> <VDOM>
```

or

```
# diagnose wad user clear
```

Network Monitor dashboard

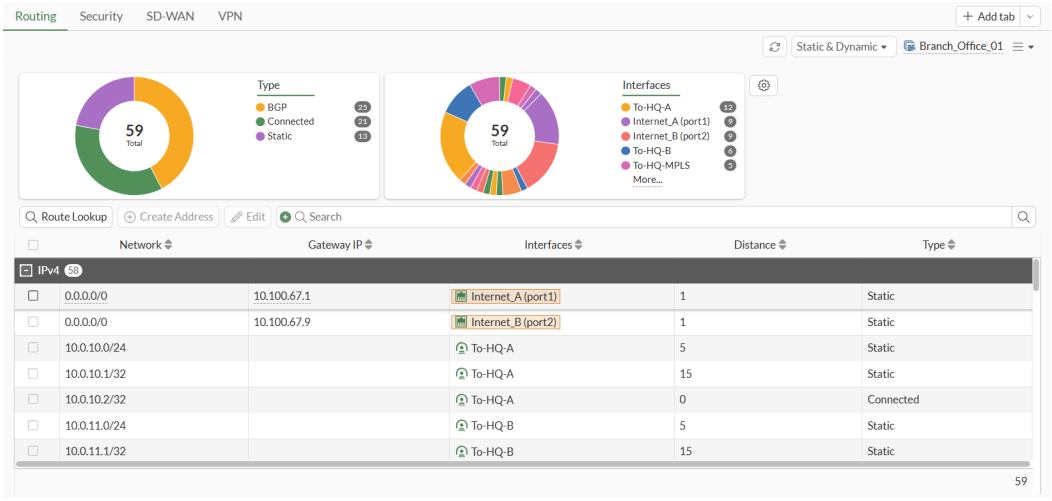
The *Network Monitor* dashboard contains several tabs of monitors that show information related to networking for this FortiGate and other devices connected to your Security Fabric. Use this dashboard to monitor the status of Routing, SD-WAN, and VPN tunnels. All of the widgets in the *Network* dashboard can be expanded to full screen and saved as a monitor.

The *Network Monitor* dashboard contains the following tabs of monitors:

Tab	Description
Routing	Shows the static and dynamic routes currently active in your routing table. The widget also includes policy routes, BGP neighbors and paths, and OSPF neighbors. See Routing monitor on page 157.
Security	Shows compromised hosts, top threats, and asset vulnerabilities. See Security on page 161.
SD-WAN	Shows a summary of the SD-WAN status, including ADVPN shortcut information.
VPN	Shows the connection statuses of your IPsec VPN site to site and dial-up tunnels. See VPN monitor on page 162.

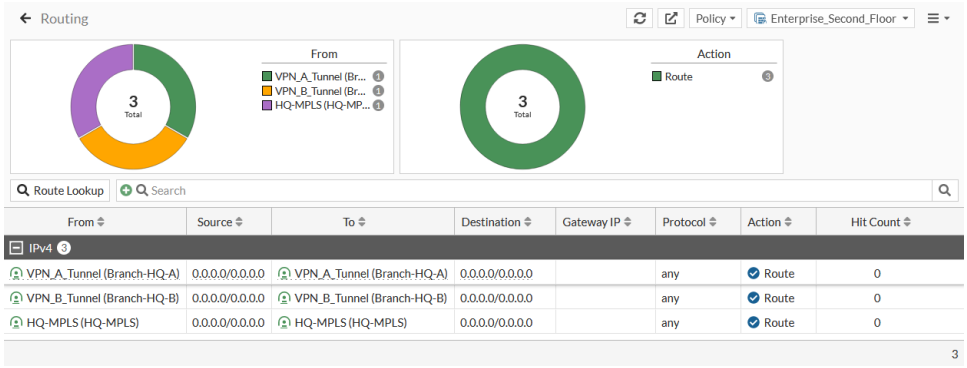
Routing monitor

The *Routing* monitor displays the routing table on the FortiGate, including all static and dynamic routing protocols in IPv4 and IPv6. You can also use this monitor to view policy routes, BGP neighbors and paths, and OSPF neighbors.

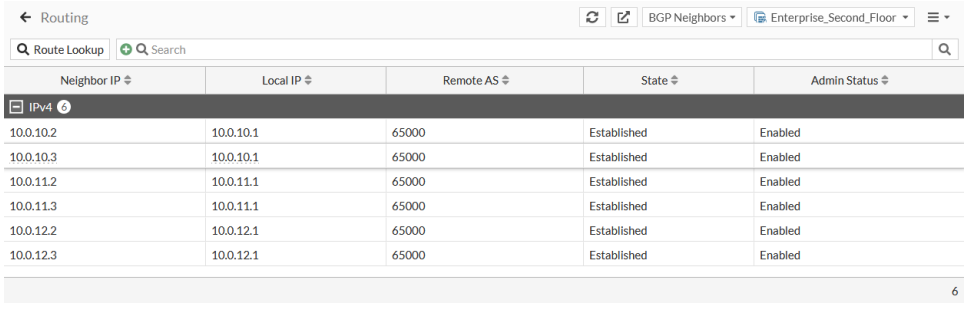


To view the routing monitor in the GUI:

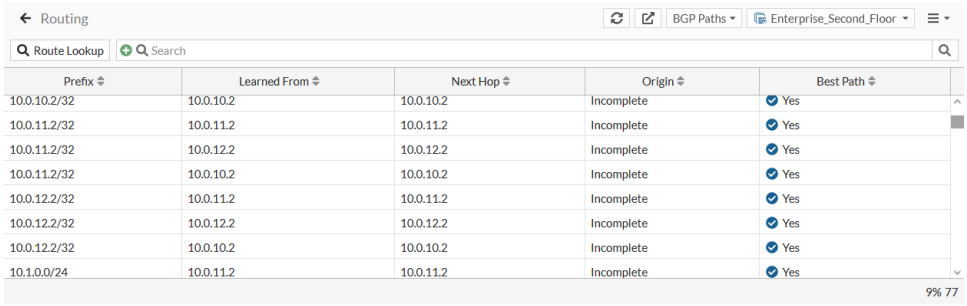
1. Go to *Dashboard* > *Network Monitor* > *Routing*. The *Routing* monitor is displayed.
2. Click the dropdown at the top of the page and select *Policy* to view policy routes.



3. Click the dropdown and select the required neighbor or path type to view neighbors and paths. For example:
   - *BGP Neighbors*

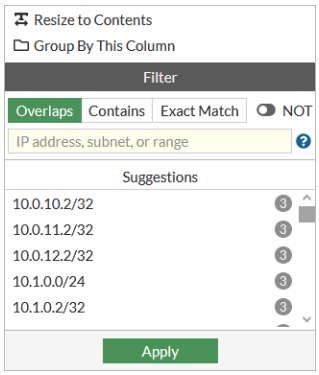


   - *BGP Paths*



4. To filter a column:
   a. Hover over the column heading, and click the *Filter/Configure Column* icon.

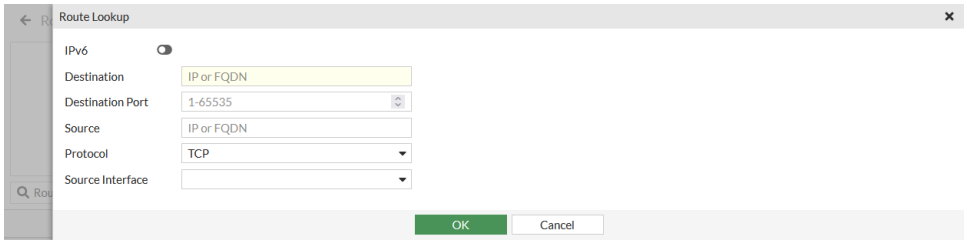
   b. Configure the filter, then click *Apply*.



5. (Optional) Click the *Save as Monitor* button to save the widget as monitor.

To look up a route in the GUI:

1. Click *Route Lookup*.



2. Enter an IP address in the *Destination* field.
3. Configure the remaining options as needed, then click *OK*.
   The matching route is highlighted on the *Routing* monitor.

To view the routing table in the CLI:

```
# get route info routing-table all
```

Sample output:

```
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
O - OSPF, IA - OSPF inter area
N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
E1 - OSPF external type 1, E2 - OSPF external type 2
i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
* - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [1/0] via 10.0.10.1, To-HQ-A
[1/0] via 10.0.12.1, To-HQ-MPLS
[1/0] via 10.10.11.1, To-HQ-B
[1/0] via 10.100.67.1, port1
[1/0] via 10.100.67.9, port2
C       10.0.10.0/24 is directly connected, To-HQ-A
C       10.0.10.2/32 is directly connected, To-HQ-A
C       10.0.11.0/24 is directly connected, To-HQ-B
C       10.0.11.2/32 is directly connected, To-HQ-B
C       10.0.12.0/24 is directly connected, To-HQ-MPLS
C       10.0.12.2/32 is directly connected, To-HQ-MPLS
C       10.1.0.0/24 is directly connected, port3
C       10.1.0.2/32 is directly connected, port3
C       10.1.0.3/32 is directly connected, port3
C       10.1.100.0/24 is directly connected, vsw.port6
```

To look up a firewall route in the CLI:

```
# diagnose firewall proute list
```

Sample output:

```
list route policy info(vf=root):

id=1(0x01) dscp_tag=0xfc 0xfc flags=0x0 tos=0x00 tos_mask=0x00 protocol=0 sport=0-0 iif=15(Branch-
HQ-A) dport=0-65535 path(1) oif=15(Branch-HQ-A)
source wildcard(1): 0.0.0.0/0.0.0.0
destination wildcard(1): 0.0.0.0/0.0.0.0
hit_count=0 last_used=2023-05-10 13:04:05

id=2(0x02) dscp_tag=0xfc 0xfc flags=0x0 tos=0x00 tos_mask=0x00 protocol=0 sport=0-0 iif=16(Branch-
HQ-B) dport=0-65535 path(1) oif=16(Branch-HQ-B)
source wildcard(1): 0.0.0.0/0.0.0.0
destination wildcard(1): 0.0.0.0/0.0.0.0
hit_count=0 last_used=2023-05-10 13:04:05

id=3(0x03) dscp_tag=0xfc 0xfc flags=0x0 tos=0x00 tos_mask=0x00 protocol=0 sport=0-0 iif=17(HQ-
MPLS) dport=0-65535 path(1) oif=17(HQ-MPLS)
source wildcard(1): 0.0.0.0/0.0.0.0
destination wildcard(1): 0.0.0.0/0.0.0.0
hit_count=0 last_used=2023-05-10 13:04:05
```

Security

The *Security* monitor provides a snapshot of the current threats and vulnerabilities targeting your Security Fabric.

The *Security* monitor contains the following widgets:

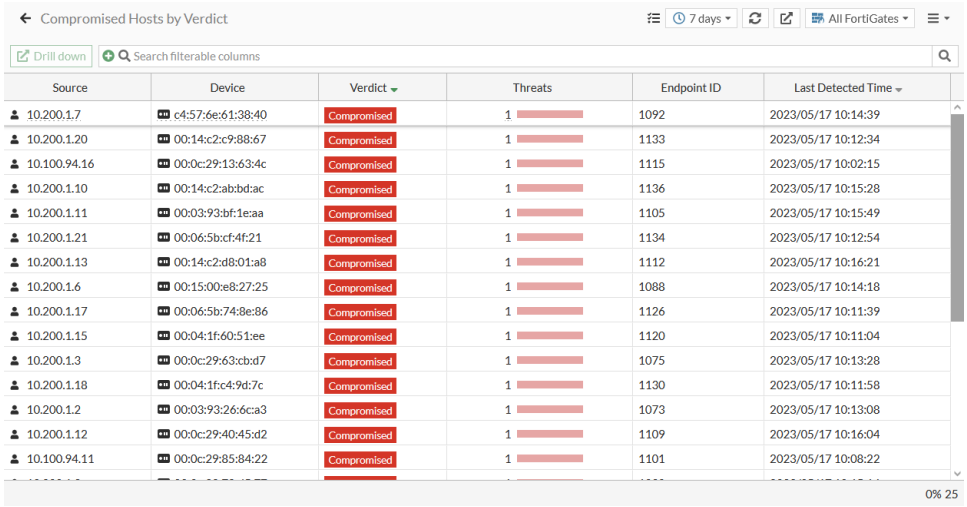
Widget	Description
Compromised Hosts by Verdict	Shows the session information for a compromised host. Click to expand the widget and view details. See Viewing session information for a compromised host on page 161.
Top Threats by Threat Level	Shows the top traffic sessions aggregated by threat. Click to expand the widget and view details about the *Threat*, *Threat Category*, *Threat Level*, *Threat Score* and *Sessions*.
Assets - Vulnerabilities	Shows the number of devices and the associated vulnerability level. Click to expand the widget and view details.

Viewing session information for a compromised host

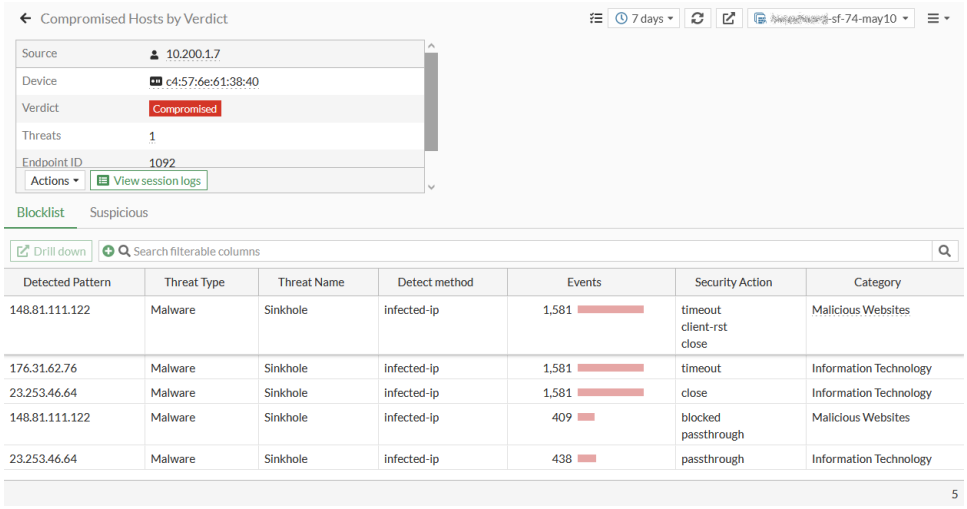
You can use the *Compromised Hosts by Verdict* widget to view the session information for a compromised host.

To view session information for a compromised host in the GUI:

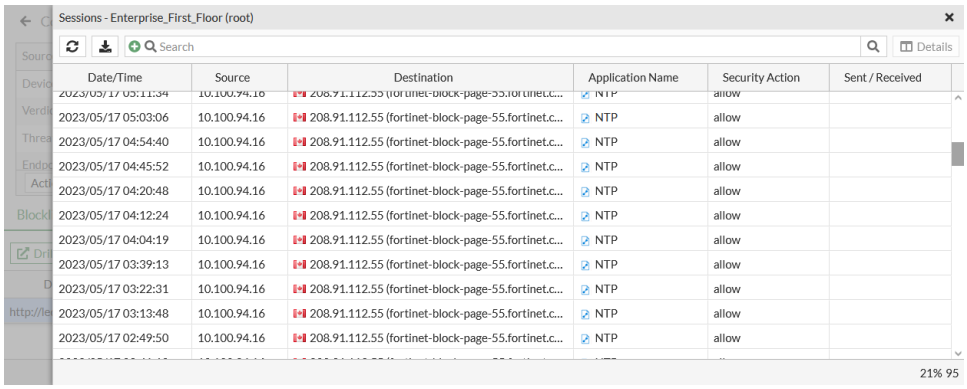
1. Go to *Dashboard* > *Network Monitor* > *Security*, and click the *Compromised Hosts by Verdict* widget to expand it.



2. Double-click a compromised host to view the session information.

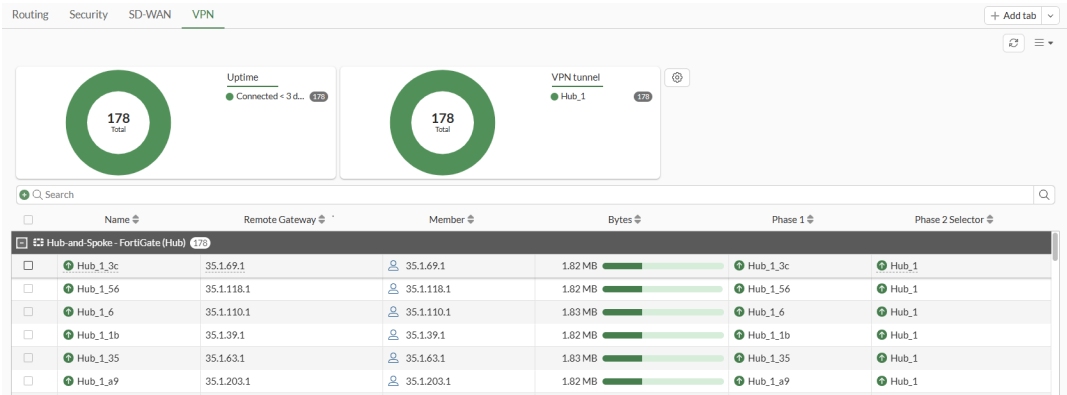


3. Select a session then click *View session logs* to view the session logs.



VPN monitor

The *VPN* monitor displays all connected Site to Site VPN, Dial-up VPNs, and ADVPN shortcut tunnel information. It also includes pie charts for tunnel status and uptime, filters, and quick access to several tools. You can use the monitor to bring a phase 2 tunnel up or down or disconnect dial-up users. A notification appears in the monitor when users have not enabled two-factor authentication.

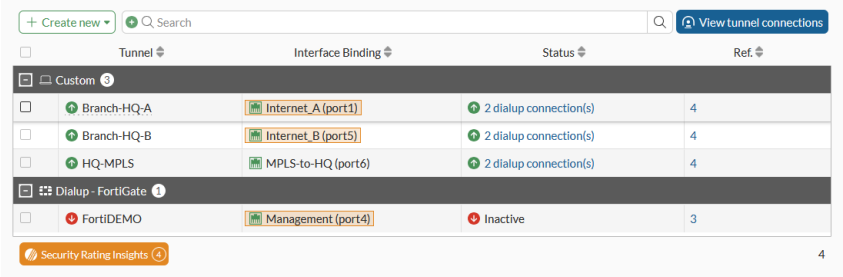


To access the VPN monitor in the GUI:

1. Use one of the following methods:
   a. Go to *Dashboard* > *Network Monitor* > *VPN*. A warning appears when an unauthenticated user is detected.

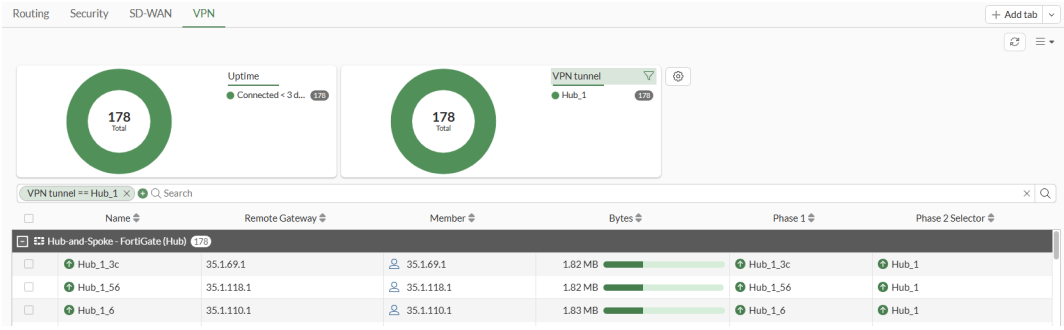
   > To filter or configure a column in the table, hover over the column heading and click the *Filter/Configure Column* button.

   b. Go to *VPN* > *VPN Tunnels* and click *View tunnel connections*.



2. Hover over a record in the table. A tooltip displays the *Phase 1* and *Phase 2 Tunnel* interfaces. A warning appears next to a user who has not enabled two-factor authentication.
3. Click each pie chart to filter its information.

   For example, in the *VPN tunnel* pie chart, click the green pie or *Hub-1*. A filter icon displays beside *VPN tunnel*, the filter text displays in the *Search* bar, and the table displays the filtered results too.



   Clear the filter by clicking the filter icon beside *VPN tunnel* or clicking an *X* in the *Search* bar.
4. Configure and filter the table as required, see Tables on page 56 for more information.

To reset statistics:

1. Select a tunnel in the table.
2. In the toolbar, click *Reset Statistics* or right-click the tunnel, and click *Reset Statistics*. The *Confirm* dialog is displayed.
3. Click *OK*.

To bring a tunnel up:

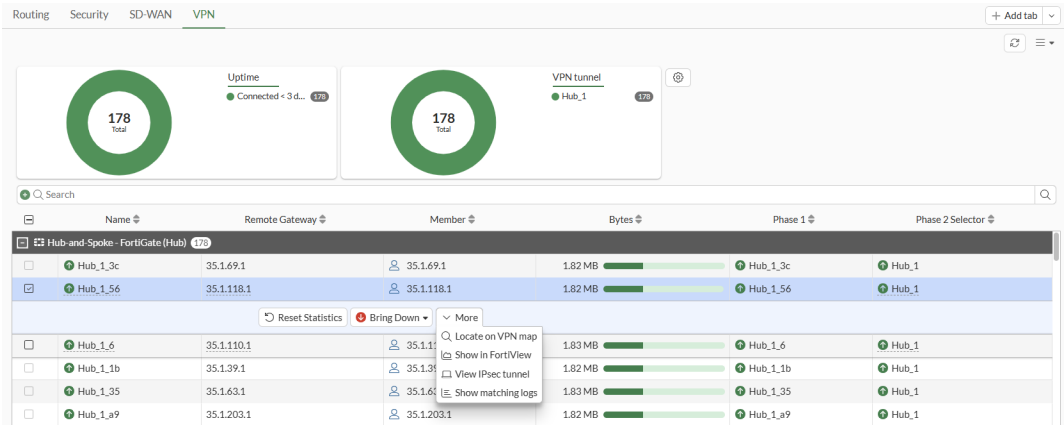
1. Select a tunnel in the table.
2. Click *Bring Up*, or right-click the tunnel, and click *Bring Up*. The *Confirm* dialog is displayed.
3. Click *OK*.

To bring a tunnel down:

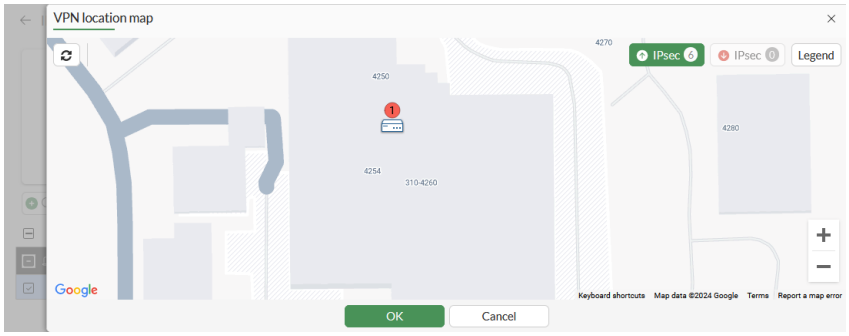
1. Select a tunnel in the table.
2. Click *Bring Down*, or right-click the tunnel, and click *Bring Down*.
   The entire tunnel, a phase 2 selector, or all phase 2 selectors can be brought down.
   The *Confirm* dialog is displayed.
3. Click *OK*.

To locate a tunnel on the VPN Map:

1. Select a tunnel in the table.
2. Click *More* > *Locate on VPN Map*, or right-click the tunnel, and click *Locate on VPN Map*.



The *VPN Location Map* is displayed.



3. Click *OK* to close the map.

To view tunnel connection in FortiView:

1. In the table, select a tunnel.
2. Click *More* > *Show in FortiView*, or right-click the tunnel and click *Show in FortiView*. The *FortiView VPN* pane is displayed.
3. Click *X* to close the pane.

To view the IPsec tunnel:

1. In the table, select a tunnel.
2. Click *More > View IPsec tunnel*, or right-click the tunnel and click *View IPsec tunnel*. The *VPN Tunnels* pane, filtered to the selected tunnel, is displayed.
3. Click *OK* or *Cancel* to close the pane.

To view matching logs for the tunnel:

1. In the table, select a tunnel.
2. Click *More > Show matching Logs*, or right-click the tunnel and click *Show matching logs*. The *Logs* pane is displayed.
3. Click *OK* to close the pane.

To view the IPsec monitor in the CLI:

```
# diagnose vpn tunnel list
```

Sample output:

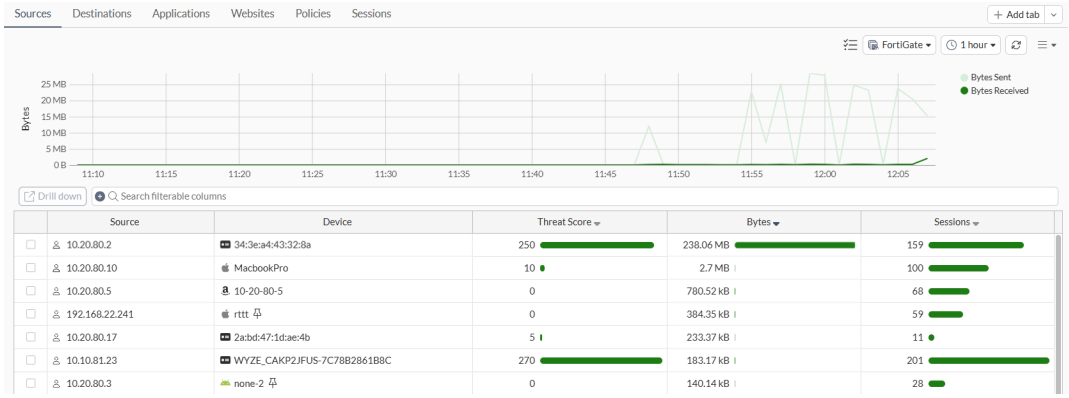
```
list all ipsec tunnel in vd 0
------------------------------------------------------
name=Branch-HQ-B_1 ver=2 serial=8 10.100.65.101:0->10.100.67.13:0 tun_id=10.0.11.2 tun_
id6=::10.0.0.8 dst_mtu=1500 dpd-link=on weight=1
bound_if=7 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/74408 options[122a8]=npu rgwy-chg
frag-rfc  run_state=0 role=primary accept_traffic=1 overlay_id=0

parent=Branch-HQ-B index=1
proxyid_num=1 child_num=0 refcnt=5 ilast=0 olast=0 ad=s/1
stat: rxp=1000472 txp=869913 rxb=184682116 txb=40548952
dpd: mode=on-idle on=1 idle=60000ms retry=3 count=0 seqno=1
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=Branch-HQ-B proto=0 sa=1 ref=6 serial=1 ads
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=20a03 type=00 soft=0 mtu=1438 expire=414/0B replaywin=2048
       seqno=1bcc esn=0 replaywin_lastseq=0000201a qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=1790/1800
  dec: spi=b4d54183 esp=aes key=16 6735d235de02f37d26809c0e8be44bbf
       ah=sha1 key=20 17261a0387d9c9a33a00a47bcf260fc59150535e
  enc: spi=28572715 esp=aes key=16 48b8a72ae69eee58699b43692ce1ccf1
       ah=sha1 key=20 3e7a219f4da33c785302ae7b935a6c15c4cc2a2a
  dec:pkts/bytes=16434/3317744, enc:pkts/bytes=14230/1299224
  npu_flag=00 npu_rgwy=10.100.67.13 npu_lgwy=10.100.65.101 npu_selid=3 dec_npuid=0 enc_npuid=0
------------------------------------------------------
name=Branch-HQ-A ver=2 serial=1 10.100.64.101:0->0.0.0.0:0 tun_id=10.0.0.1 tun_id6=::10.0.0.1 dst_
mtu=0 dpd-link=on weight=1
bound_if=3 lgwy=static/1 tun=intf mode=dialup/2 encap=none/552 options[0228]=npu frag-rfc
role=primary accept_traffic=1 overlay_id=0
```

```
proxyid_num=0 child_num=2 refcnt=4 ilast=43124593 olast=43124593 ad=/0
stat: rxp=1860386 txp=1598633 rxb=215561858 txb=71724716
dpd: mode=on-idle on=0 idle=60000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
run_tally=0
------------------------------------------------------
name=Branch-HQ-B ver=2 serial=2 10.100.65.101:0->0.0.0.0:0 tun_id=10.0.0.2 tun_id6=::10.0.0.2 dst_
mtu=0 dpd-link=on weight=1
...
```

FortiView dashboard

The *Dashboard* > *FortiView* dashboard includes several tabs of monitors that display information about top traffic volume by sources, destinations, applications, web sites, policies, and sessions.

Additional FortiView monitors and widgets can be added to the dashboard. You can also add FortiView monitors directly to the tree menu with the *+ Add dashboard/monitor*button.



The following tabs of monitors are available by default:

Dashboard	Usage
Sources	Displays Top Sources by traffic volume and drilldown by Source.
Destinations	Displays Top Destinations by traffic volume and drilldown by Destination.
Applications	Displays Top Applications by traffic volume and drilldown by Application.
Websites	Displays Top Websites by session count and drilldown by Domain.
Policies	Displays Top Policies by traffic volume and drilldown by Policy number
Sessions	Displays Top Sessions by traffic source and can be used to end sessions.

Usage is based on default settings. The pages may be customized further and sorted by other fields.

You can quarantine a host and ban an IP from all of the core FortiView monitors.

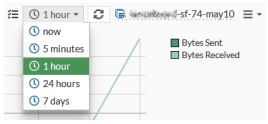
See Editing dashboards of widgets and monitors on page 194 for information about adding monitors and widgets.

Using FortiView

Use the *Dashboard* > *FortiView* dashboard to customize the view and visualizations within a monitor to find the information you are looking for. The tools in the top menu bar allow you to change the time display, refresh or customize the data source, and filter the results. You can also right-click a table in the monitor to view drilldown information for an item.

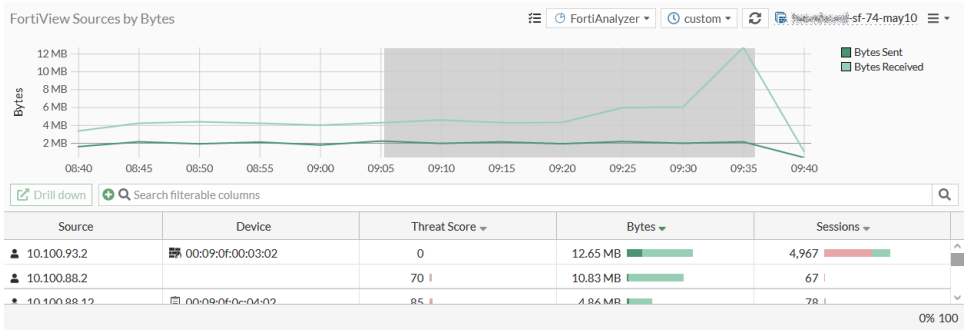
Real-time and historical charts

Use the *Time Display* dropdown to select the time period to display on the current monitor. Time display options vary depending on the monitor and can include real-time information (*now*) and historical information (*1 hour*, *24 hours*, and *7 days*).

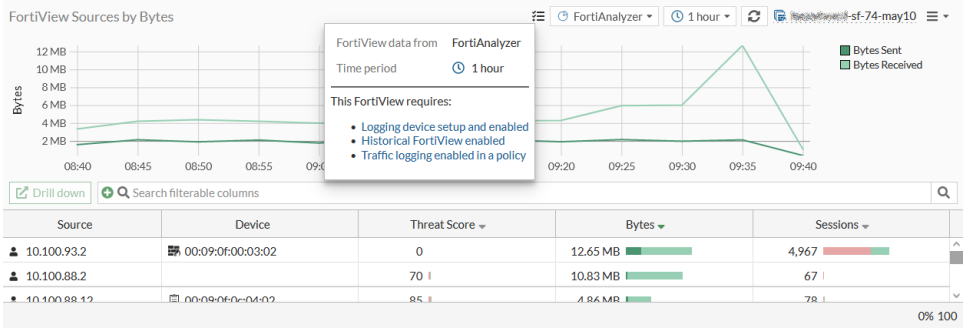


Disk logging or remote logging must be enabled to view historical information.

You can create a custom time range by selecting an area in table with your cursor.

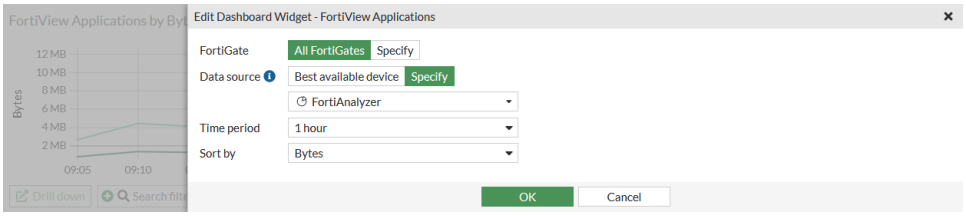


The icon next to the time period identifies the data source (FortiGate, FortiAnalyzer, or FortiGate Cloud). Hover over its icon to see a description of the chart, as well as links to the requirements.



Data source

FortiView gathers information from a variety of data sources. If neither log disk nor remote logging is configured, the data will be drawn from the FortiGate's session table, and the *Time Period* is set to *Now*.



Other data sources that can be configured are:

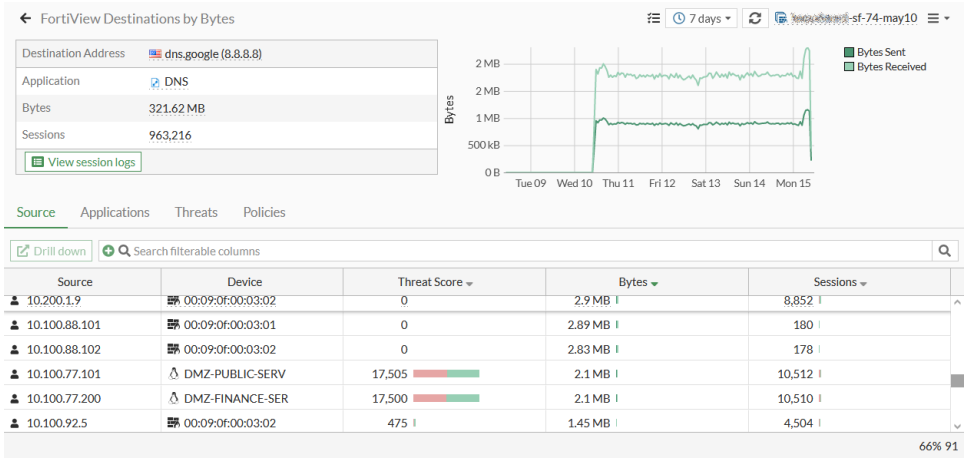
- FortiGate
- FortiAnalyzer
- FortiGate Cloud

When *Data Source* is set to *Best Available Device*, FortiAnalyzer is selected when available, then FortiGate Cloud, and then FortiGate.

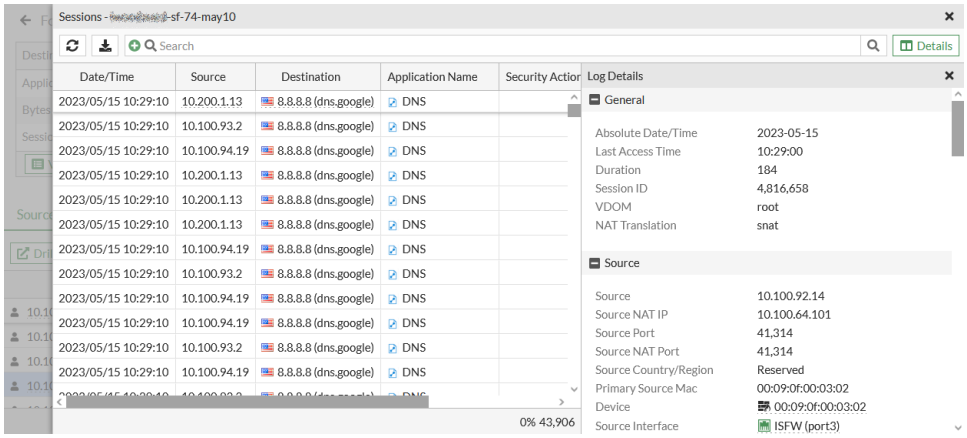
Drilldown information

Double-click or right-click an entry in a FortiView monitor and select *Drill Down to Details* to view additional details about the selected traffic activity. Click the *Back* icon in the toolbar to return to the previous view.

You can group drilldown information into different drilldown views. For example, you can group the drilldown information in the *FortiView Destinations* monitor by *Sources*, *Applications*, *Threats*, and *Policies*.



Select an entry, then click *View session logs* to view the session logs.

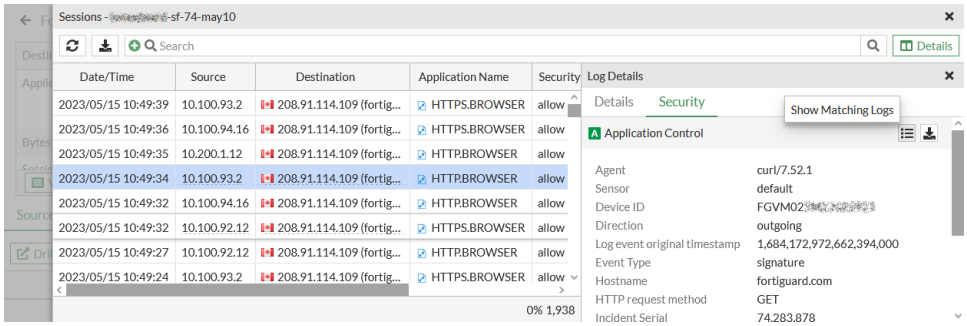


Graph	• The graph shows the bytes sent/received in the time frame. real time does not include a chart. • Users can customize the time frame by selecting a time period within the graph.
Summary of	• Shows information such as the user/avatar, avatar/source IP, bytes, and sessions total for the time period. • Can quarantine host (access layer quarantine) if they are behind a FortiSwitch or FortiAP. • Can ban IP addresses, adds the source IP address into the quarantine list.

Tabs	• Drilling down entries in any of these tabs (except sessions tab) will take you to the underlying traffic log in the sessions tab.
	• *Applications* shows a list of the applications attributed to the source IP. This can include scanned applications using Application Control in a firewall policy or unscanned applications.
	`config log gui-display`
	`set fortiview-unscanned-apps enable`
	`end`

- *Destinations* shows destinations grouped by IP address/FQDN.
- *Threats* lists the threats caught by UTM profiles. This can be from antivirus, IPS, Web Filter, Application Control, etc.
- *Web Sites* contains the websites which were detected either with webfilter, or through FQDN in traffic logs.
- *Web Categories* groups entries into their categories as dictated by the Web Filter Database.
- *Policies* groups the entries into which polices they passed through or were blocked by.
- *View session logs* shows the underlying logs (historical) or sessions (real time). Drilldowns from other tabs end up showing the underlying log located in this tab.
- *Search Phrases* shows entries of search phrases on search engines captured by a Web Filter UTM profile, with deep inspection enabled in firewall policy.
- More information can be shown in a tooltip while hovering over these entries.

To view matching logs or download a log, click the *Security* tab in the *Log Details* .



Enabling FortiView from devices

You can enable FortiView from SSD disk, FortiAnalyzer, and FortiGate Cloud.

FortiView from disk

FortiView from disk is available on all FortiGates with an SSD disk.

Restrictions

Model	Supported view
Entry-level models with SSD	Five minutes and one hour
Mid-range models with SSD	Up to 24 hours
High-end models with SSD	Up to seven days To enable seven days view:

Model	Supported view
	config log setting set fortiview-weekly-data enable end

Configuration

A firewall policy needs to be in place with traffic logging enabled. For optimal operation with FortiView, internal interface roles should be clearly defined as LAN. DMZ and internet facing or external interface roles should be defined as WAN.

To configure logging to disk:

```
config log disk setting
    set status enable
end
```

To include sniffer traffic and local-deny traffic when FortiView from Disk:

```
config report setting
    set report-source forward-traffic sniffer-traffic local-deny-traffic
end
```

This feature is only supported through the CLI.

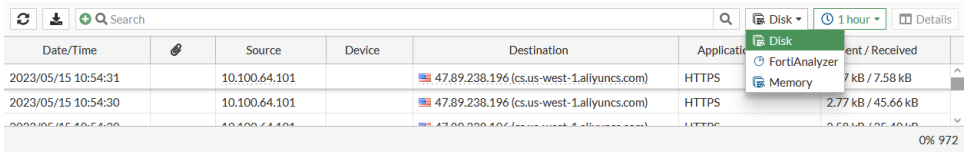
Troubleshooting

Use `execute report flush-cache` and `execute report recreate-db` to clear up any irregularities that may be caused by upgrading or cache issues.

Traffic logs

To view traffic logs from disk:

1. Go to *Log & Report*, and select either the *Forward Traffic*, *Local Traffic*, *Sniffer Traffic*, or *ZTNA Traffic* views.
2. In the toolbar, select *Disk* for the log location dropdown.



FortiView from FortiAnalyzer

Connect FortiGate to a FortiAnalyzer to increase the functionality of FortiView. Adding a FortiAnalyzer is useful when adding monitors such as the *Compromised Hosts*. FortiAnalyzer also allows you to view historical information for up to seven days.

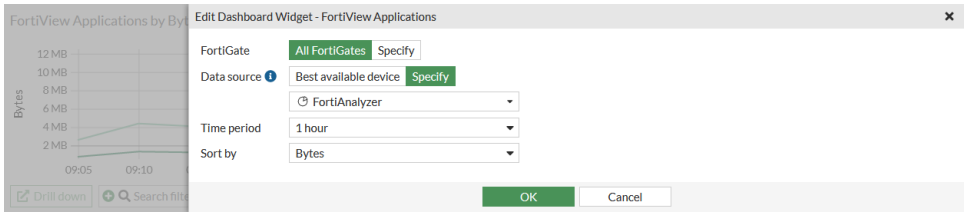
Requirements

- A FortiGate or FortiOS
- A compatible FortiAnalyzer (see Compatibility with FortiOS)

To configure logging to the FortiAnalyzer, see Configuring FortiAnalyzer on page 4311

To enable FortiView from FortiAnalyzer:

1. Go to *Dashboard* > *FortiView* > *Sources*.
2. Select a time range other than *Now* from the dropdown list to view historical data.
3. In top menu, click the dropdown, and select *Settings*. The *Edit Dashboard Widget* dialog is displayed.
   a. In the *Data Source* area, click *Specify*.
   b. From the dropdown, select *FortiAnalyzer*, and click *OK*.



All the historical information now comes from the FortiAnalyzer.

When *Data Source* is set to *Best Available Device*, FortiAnalyzer is selected when available, then FortiGate Cloud, and then FortiGate.

FortiView from FortiGate Cloud

This function requires a FortiGate that is registered and logged into a compatible FortiGate Cloud. When using FortiGate Cloud, the *Time Period* can be set to up to 24 hours.

To configure logging to FortiGate Cloud, see Configuring cloud logging on page 4314.

To enable FortiView with log source as FortiGate Cloud:

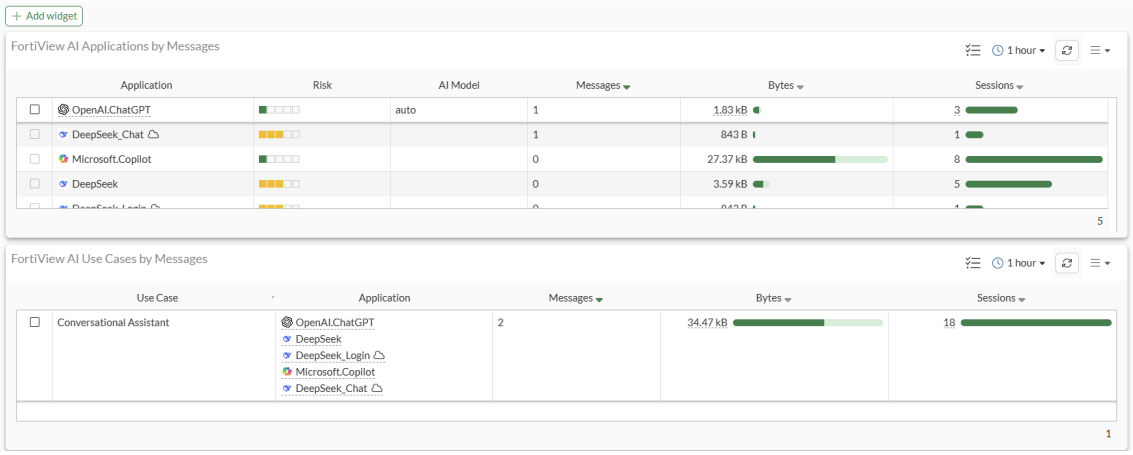
1. Go to *Dashboard* > *FortiView* > *Sources*.
2. In the top menu, click the dropdown, and select *Settings*. The *Edit Dashboard Widget* window opens.
   a. In the *Data Source* area, click *Specify*.
   b. From the dropdown, select *FortiGate Cloud*, then click *OK*.

You can select FortiGate Cloud as the data source for all available FortiView pages and widgets.

FortiView AI widgets

Aggregated FortiView AI application information can be monitored in the *FortiView AI Applications* and *FortiView AI Use Cases* widgets:

- The *FortiView AI Applications* widget displays top traffic sessions aggregated by AI application and includes information on *Risk*, *AI model*, *Sessions*, and so on. Selecting *Drill Down* on an *Application* will display specific information, such as *Users*, *Use Cases*, and *Data Center Locations*.
- The FortiView AI Use Cases widget displays top traffic sessions aggregated by AI use case and includes information on *Use Case*, *Application*, *Sessions*, and so on. Selecting *Drill Down* on an *Application* will display specific information, such as *Users*, *Applications*, and *Data Center Locations*.



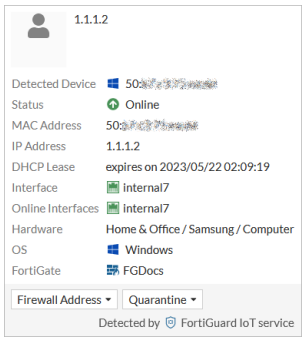
See Editing dashboards of widgets and monitors on page 194 for information about adding monitors and widgets.

FortiView sources

The *Dashboard* > *FortiView* > *Sources* monitor displays top sources sorted by Bytes, Sessions or Threat Score. The information can be displayed in real time or historical views. You can use the monitor to create or edit a firewall device address or IP address definitions, quarantine hosts, and temporarily or permanently ban IPs.

To add a firewall device or IP address:

1. In the table, hover over the source or device MAC address. An information window opens.

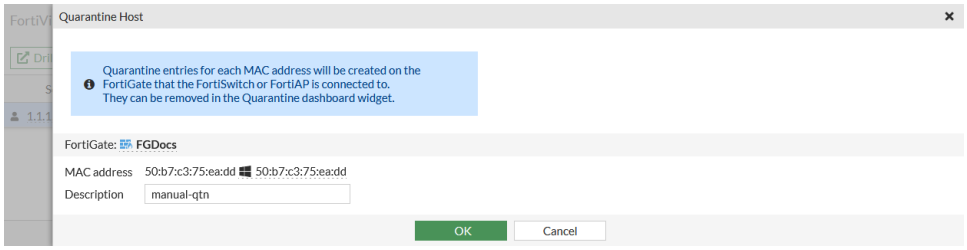


2. Click *Firewall Address* > *Create Firewall Device Address* or *Firewall Address* > *Create Firewall IP Address*. The *New Address* pane opens.
3. Configure the address settings as needed, then click *OK*.

> Use the *Name* field to assign a descriptive name to a device so it is easier to find it in the *Device* column. After you finish configuring the device, refresh the page to see the new name in the monitor.

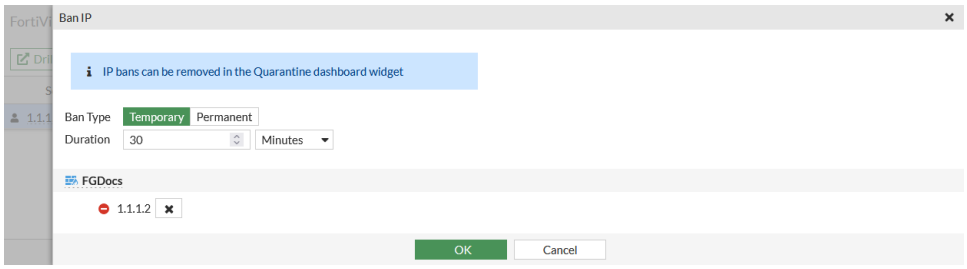
To quarantine a host:

1. In the table, hover over the source or device MAC address. An information window opens.
2. Click *Quarantine* > *Quarantine Host*. The *Quarantine Host* dialog is displayed.
3. Configure the quarantine settings, then click *OK*.



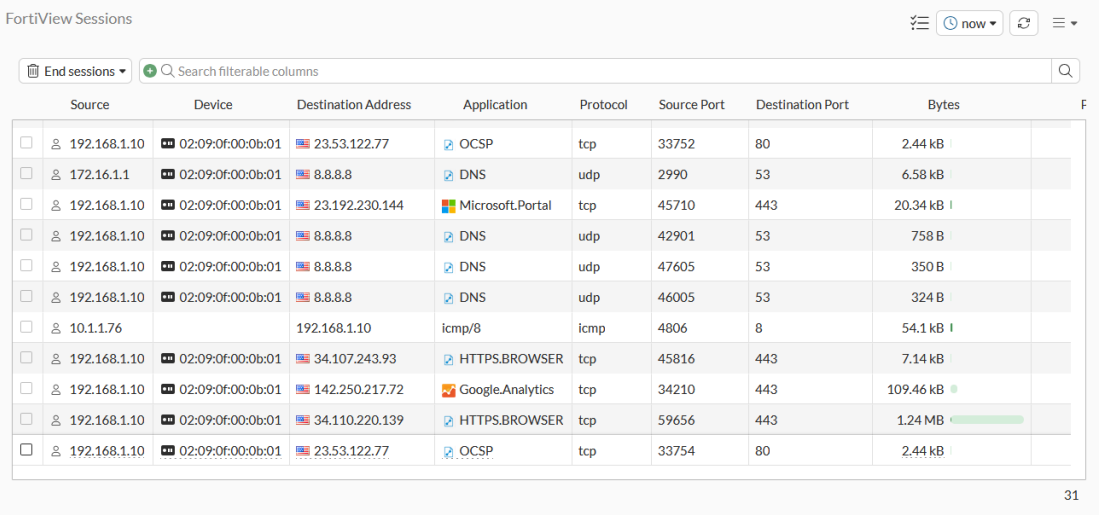
To ban an IP address:

1. In the table, hover over the source or device MAC address. An information window opens.
2. Click *Quarantine* > *Ban IP* . The *Ban IP* dialog is displayed.
3. Configure the ban IP settings, then click *OK*.



FortiView Sessions

The *Dashboard* > *FortiView* > *Sessions* monitor displays *Top Sessions* by traffic source and can be used to end sessions.

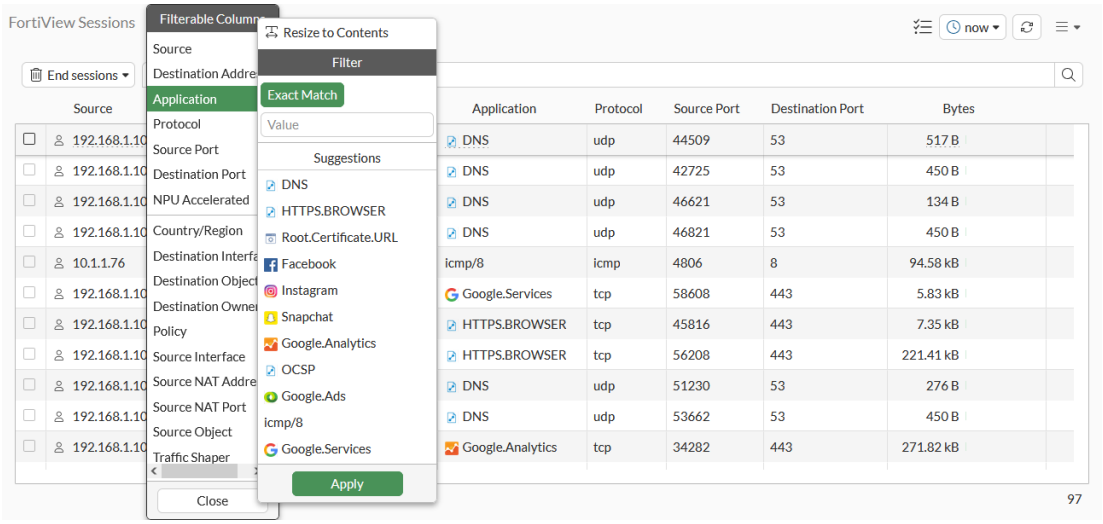


The session table displayed on the *Dashboard* > *FortiView* > *Sessions* monitor is useful when verifying open connections. For example, if you have a web browser open to browse the Fortinet website, you would expect a session entry from your computer on port 80 to the IP address for the Fortinet website. You can also use a session table to investigate why there are too many sessions for FortiOS to process.

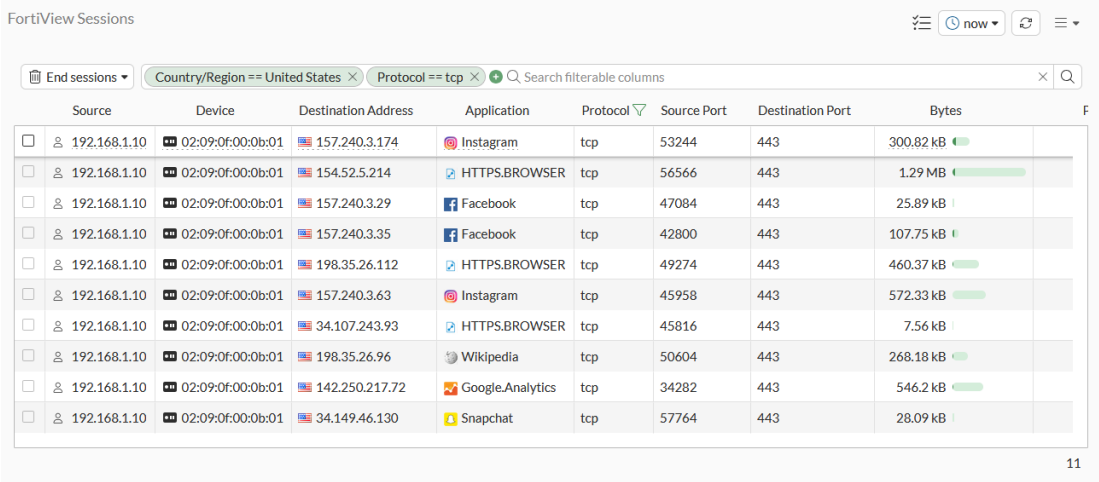
You can filter the sessions displayed in the session table by setting up the available filtering options.

To filter sessions in the session table:

1. Click on the *Add Filter* button at the top of the session table.
2. Select the required filtering option. The session table updates to the filter selection.



3. You may add one or more filters depending upon your requirements. To add more filters, repeat the above steps for a different set of filters.



You can be very specific with how you use filters and target sessions based on different filter combinations. For example, you may want to view all sessions from a device with a particular IP by adding the *Source IP* filter. Similarly, you may need to target all the sessions having a particular *Destination IP* and *Destination Port*, and so on.

You may also view the session data in the CLI.

To view session data using the CLI:

```
# diagnose sys session list
```

The session table output in the CLI is very large. You can use the supported filters in the CLI to show only the data you need.

To view session data with filters using the CLI:

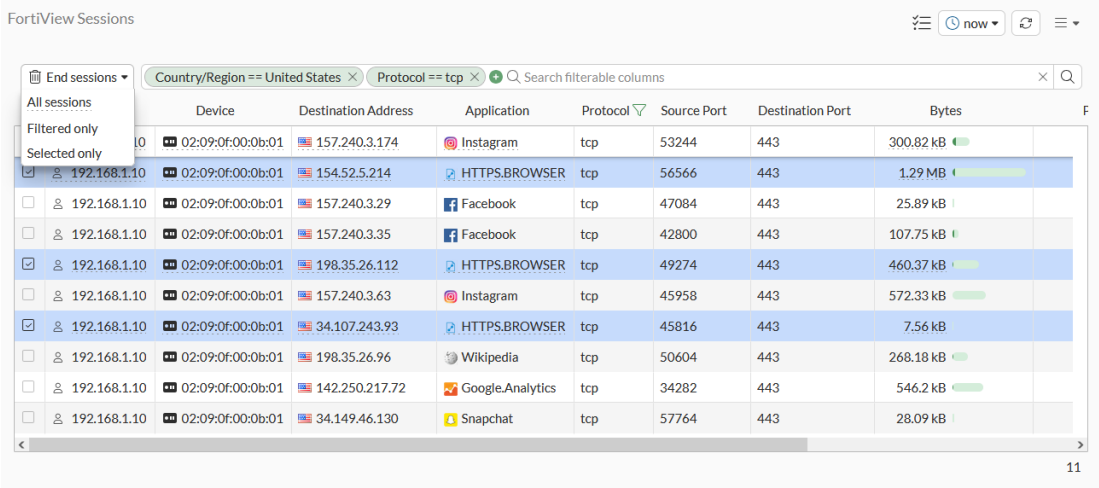
```
# diagnose sys session filter <option>
```

See to learn more about using the supported filters in the CLI.

You may also decide to end a particular session or all sessions for administrative purposes.

To end sessions from the GUI:

1. Select the session you want to end. To select multiple sessions, hold the *Ctrl* or *Shift* key on your keyboard while clicking the sessions.



2. Click on *End Sessions* > *Selected only* to end the selected sessions.
   You can also click *End Sessions* > *All Sessions* to end all active sessions, or *End Sessions* > *Filtered only* to end only the filtered sessions.
3. Click *OK* in the confirmation dialog.

FortiView Top Source and Top Destination Firewall Objects monitors

The *FortiView Source Firewall Objects* and *FortiView Destination Firewall Objects* monitors leverage UUID to resolve firewall object address names for improved usability.

Requirements

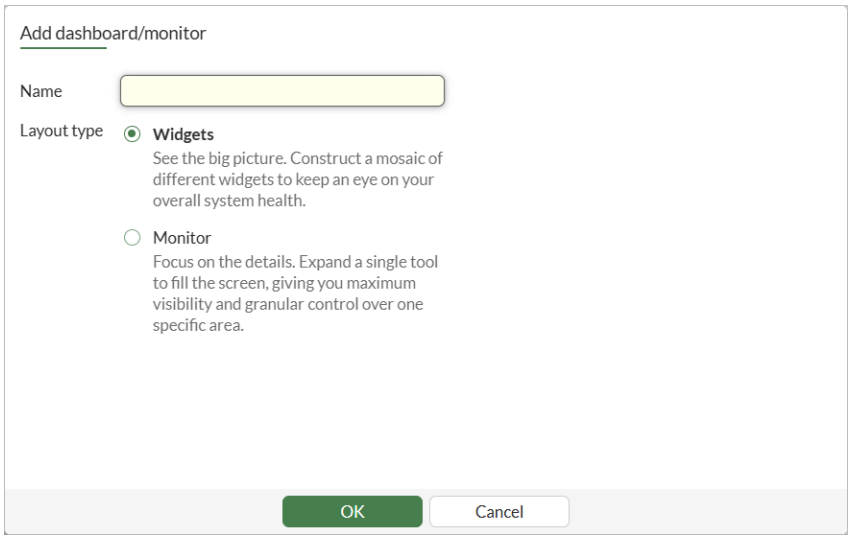
To have a historical *Firewall Objects*-based view, address objects' UUIDs need to be logged.

To enable address object UUID logging in the CLI:

```
config system global
    set log-uuid-address enable
end
```

To add a firewall object monitor in the GUI:

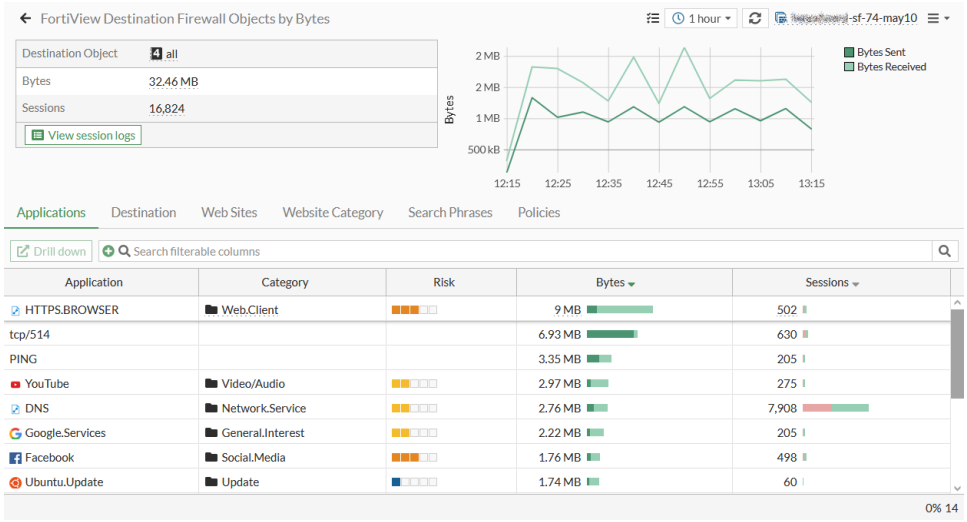
1. Go to *Dashboard* > *FortiView*.
2. Click *Add tab*. The *Add dashboard tab* window opens.



3. Enter a name for the tab.
4. Select *Monitor* and click *Choose monitor*. The *Add monitor* pane is displayed.
5. In the *Search* field, type *FortiView Destination Firewall Objects* and click the *Add* button next to the dashboard name.
6. Beside *FortiGate*, click *Specify* and select the FortiGate(s) from the dropdown.
7. In the *Data Source* area, select *Best Available Device* or *Specify*. For information, see Using FortiView on page 167.
8. From the *Time Period* dropdown, select the time period. Select *now* for real-time information, or (*1 hour*, *24 hours*, and *7 days*) for historical information.
9. From the *Sort By* dropdown, select *Bytes*, *Sessions*, *Bandwidth*, or *Packets*.
10. Click *OK*.
11. Click *OK*. The monitor tab is added to the dashboard.

To drill down Firewall Objects:

1. Go to *Dashboard* > *FortiView* > *FortiView Source Firewall Objects* or *FortiView Destination Firewall Objects* to open the monitor.
2. Select any source or destination object and click *Drill down*.
3. Click the tabs to sort the sessions.

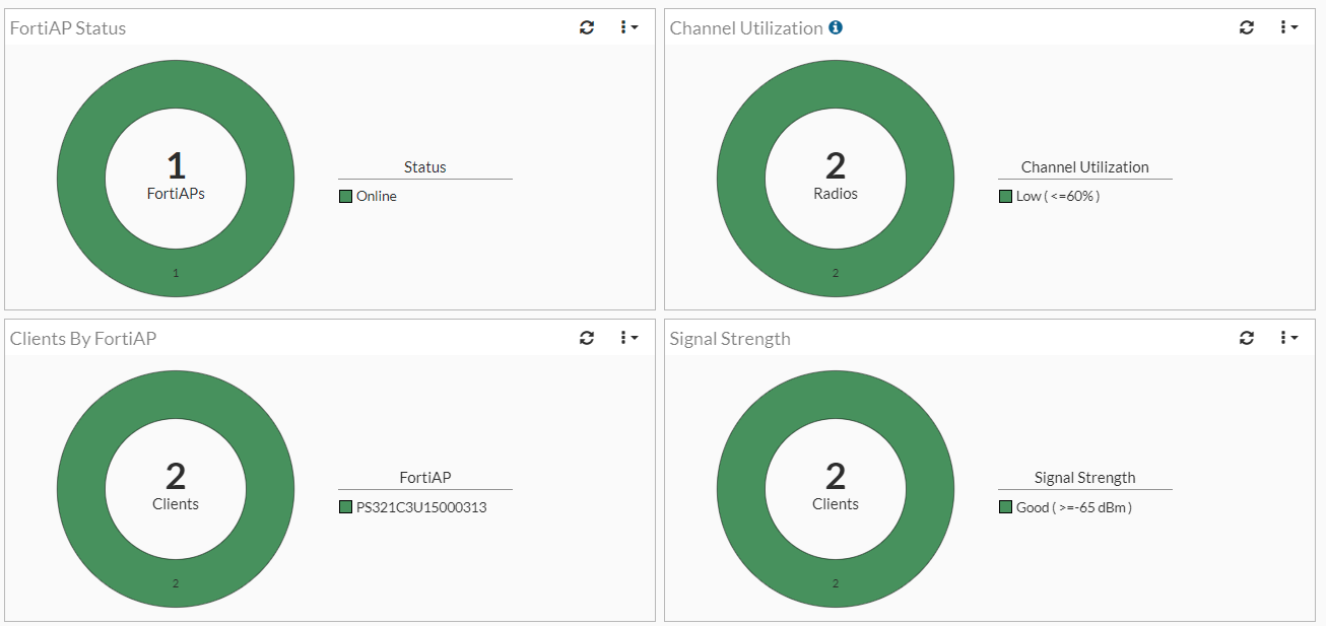


4. Select an entry, then click *View session logs* to view the session logs.

WiFi dashboard

The *WiFi* dashboard provides an overview of your WiFi network's performance, including FortiAP status, channel utilization, WiFi clients and associated information, login failures, and signal strength.

To access the WiFi dashboard, go to *Dashboard* > *WiFi*.



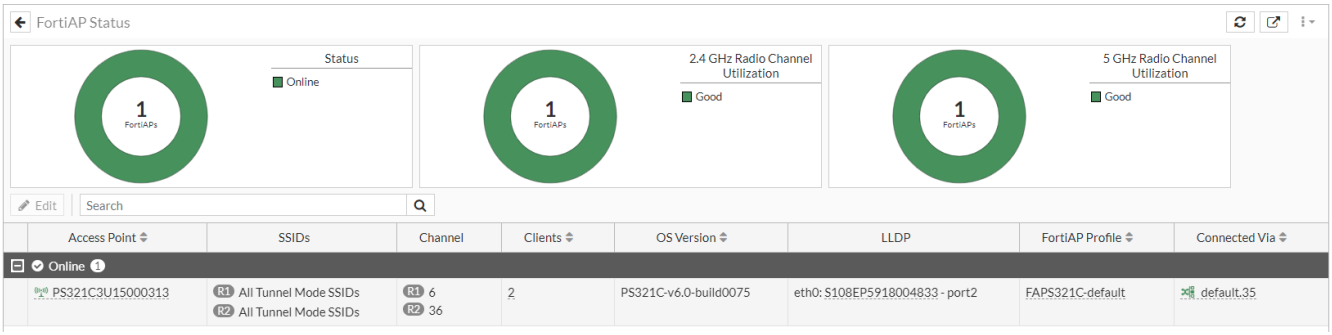
The WiFi dashboard can be customized for your requirements. To learn more about using and modifying dashboards and widgets, see Dashboard on page 142.

This section describes the following monitors available for the WiFi Dashboard:

- FortiAP Status monitor on page 180
- Clients by FortiAP monitor on page 181

FortiAP Status monitor

The *FortiAP Status* monitor displays the status and the channel utilization of the radios of FortiAP devices connected to a FortiGate. It also provides access to tools to diagnose and analyze connected APs.

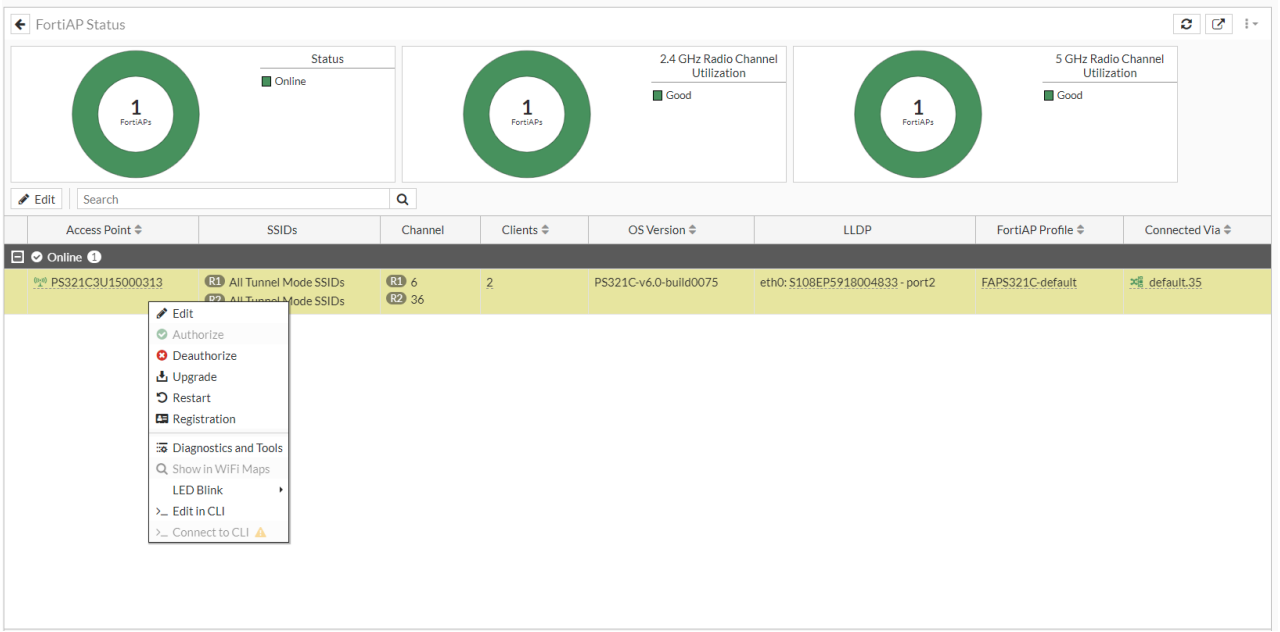


To view the *FortiAP Status* monitor:

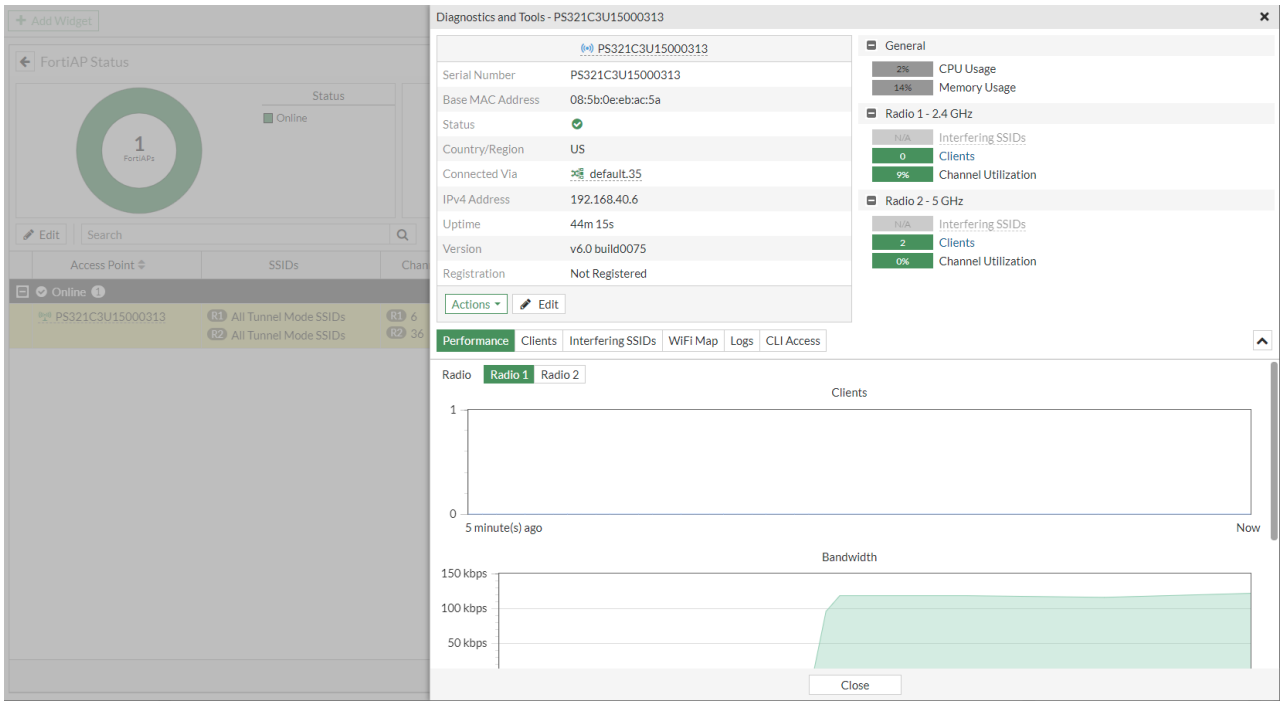
1. Go to *Dashboard* > *WiFi*.
2. Hover over the *FortiAP Status* widget, and click *Expand to Full Screen*. The *FortiAP Status* monitor opens.

To view the Diagnostics and Tools menu:

1. Right-click an *Access Point* in the table, and click *Diagnostics and Tools*. The *Diagnostics and Tools* dialog opens.



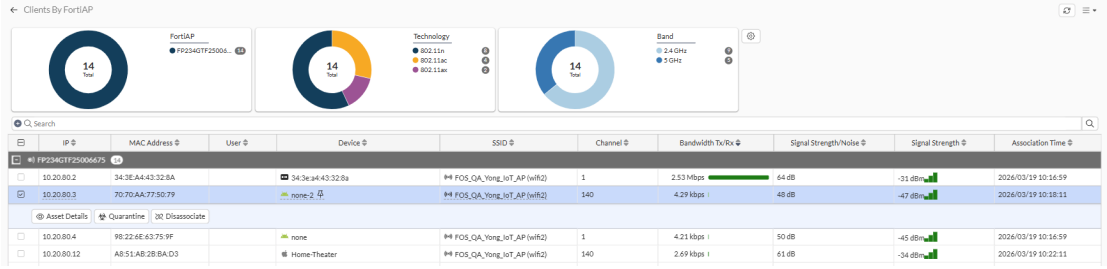
2. To monitor and analyze the FortiAP device, click on the tabs in the *Diagnostics and Tools* dialog, such as *Clients*, *Spectrum Analysis*, *VLAN Probe*, and so on.



The *Diagnostics and Tools* dialog is similar to the device dialog from *WiFi & Switch Controller > Managed FortiAPs*. To learn more about the various tabs and their functions, see Spectrum analysis of FortiAP E models, VLAN probe report, and Standardize wireless health metrics.

Clients by FortiAP monitor

The *Clients by FortiAP* monitor allows you to view detailed information about the health of individual WiFi connections in the network. It also provides access to tools to diagnose and analyze connected wireless devices.

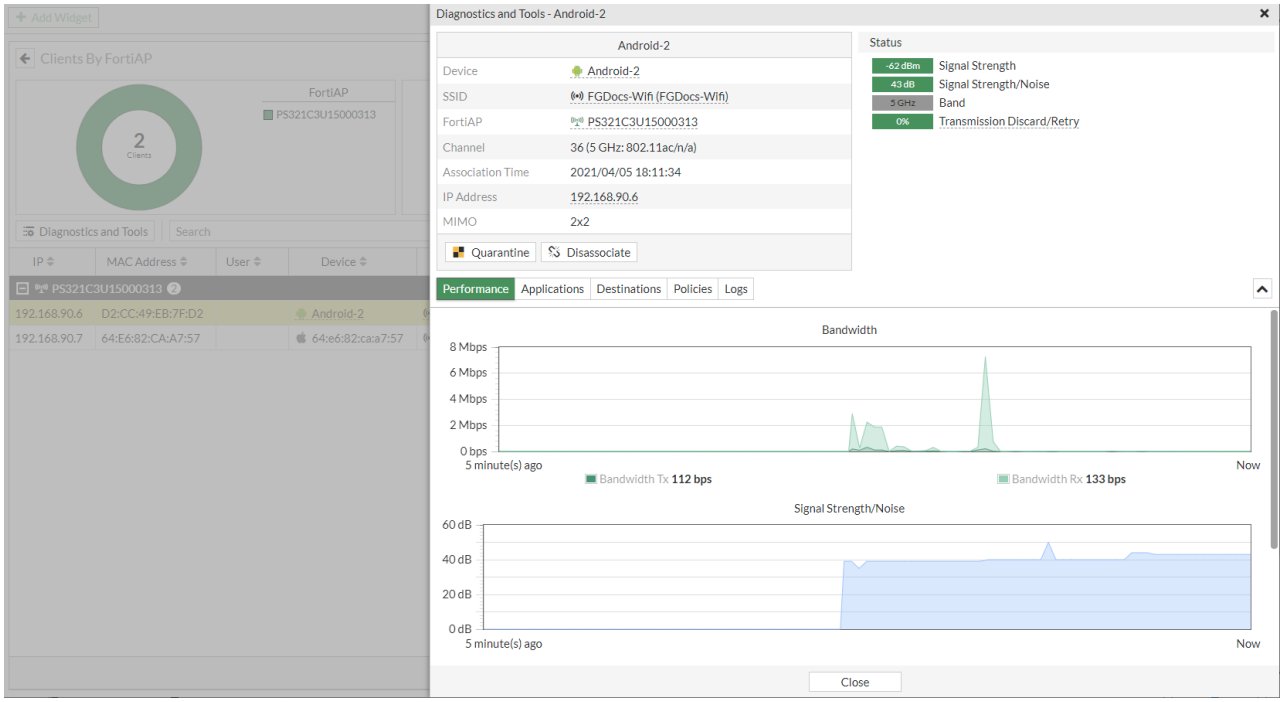


To view the *Clients by FortiAP* monitor:

1. Go to *Dashboard > WiFi*.
2. Hover over the *Clients by FortiAP* widget, and click *Expand to Full Screen*. The *Clients by FortiAP* monitor opens.

To view the summary page for a wireless client:

1. Right-click a client in the table and select *Diagnostics and Tools*. The *Diagnostics and Tools - <device>* page is displayed.



2. (Optional) Click *Quarantine* to quarantine the client,
3. (Optional) Click *Disassociate* to disassociate the client.

Health status

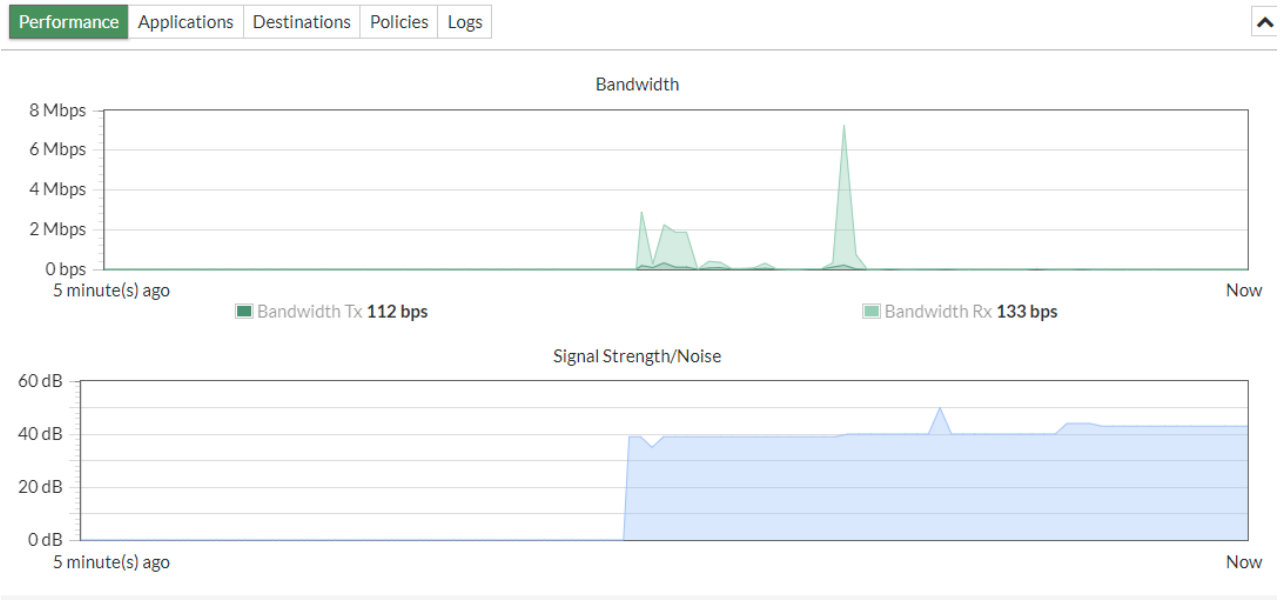
The *Status* section displays the overall health for the wireless connection. The overall health of the connection is:

- Good if the value range for all three conditions are *Good*
- Fair or poor if one of the three conditions is *Fair* or *Poor* respectively.

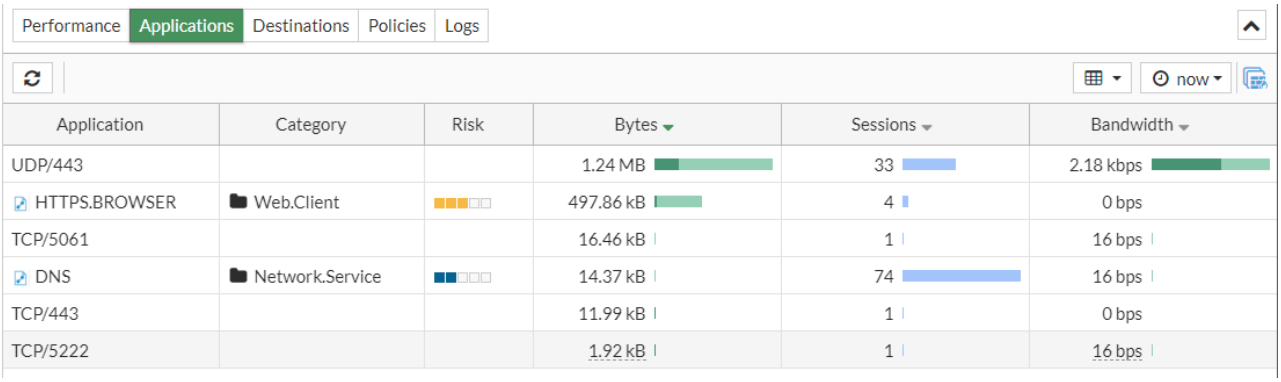
Condition	Value Range
Signal Strength	• *Good > -56dBm* • *-56dBm > Fair > -75dBm* • *Poor < -75dBm*
Signal Strength/Noise	• *Good > 39dBm* • *20dBm < Fair < 39dBm* • *Poor < 20dBm*
Band	• *Good = 5G band* • *Fair = 2.4G band*

The summary page also has the following FortiView tabs:

- *Performance*

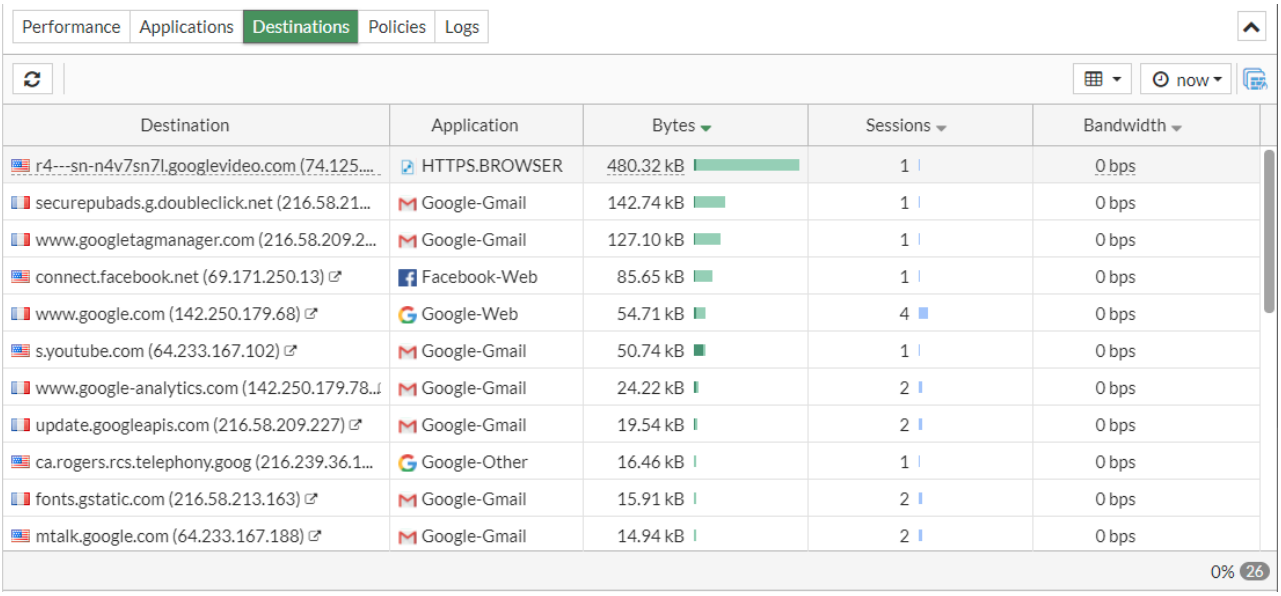


- *Applications*



Application	Category	Risk	Bytes	Sessions	Bandwidth
UDP/443			1.24 MB	33	2.18 kbps
HTTPS.BROWSER	Web.Client		497.86 kB	4	0 bps
TCP/5061			16.46 kB	1	16 bps
DNS	Network.Service		14.37 kB	74	16 bps
TCP/443			11.99 kB	1	0 bps
TCP/5222			1.92 kB	1	16 bps

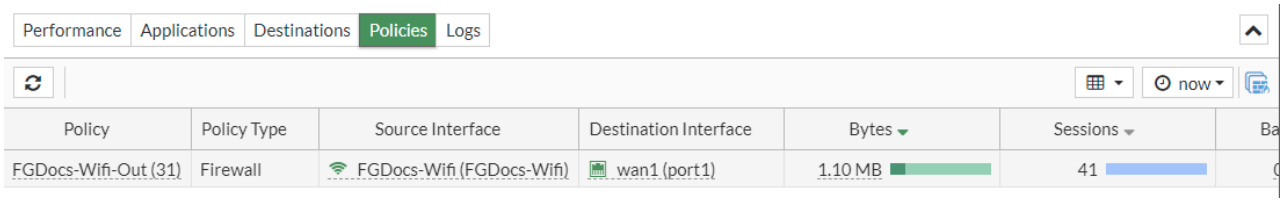
- *Destinations*



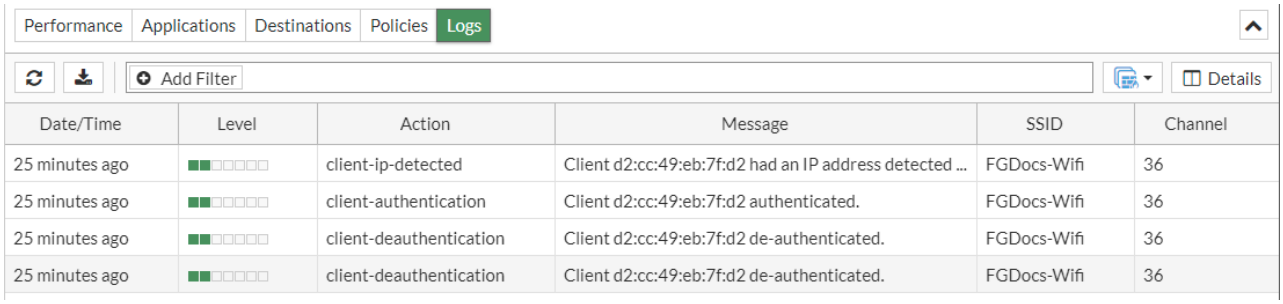
Destination	Application	Bytes	Sessions	Bandwidth
r4---sn-n4v7sn7l.googlevideo.com (74.125....	HTTPS.BROWSER	480.32 kB	1	0 bps
securepubads.g.doubleclick.net (216.58.21...	Google-Gmail	142.74 kB	1	0 bps
www.googletagmanager.com (216.58.209.2...	Google-Gmail	127.10 kB	1	0 bps
connect.facebook.net (69.171.250.13)	Facebook-Web	85.65 kB	1	0 bps
www.google.com (142.250.179.68)	Google-Web	54.71 kB	4	0 bps
s.youtube.com (64.233.167.102)	Google-Gmail	50.74 kB	1	0 bps
www.google-analytics.com (142.250.179.78...	Google-Gmail	24.22 kB	2	0 bps
update.googleapis.com (216.58.209.227)	Google-Gmail	19.54 kB	2	0 bps
ca.rogers.rcs.telephony.goog (216.239.36.1...	Google-Other	16.46 kB	1	0 bps
fonts.gstatic.com (216.58.213.163)	Google-Gmail	15.91 kB	2	0 bps
mtalk.google.com (64.233.167.188)	Google-Gmail	14.94 kB	2	0 bps

0% 26

- *Policies*



- *Logs*



Monitors and widgets

FortiGate supports both FortiView and non-FortiView monitors and widgets.

FortiView monitors and widgets are driven by traffic information captured from logs and real-time data. See FortiView monitors and widgets on page 184.

Non-FortiView monitors and widgets capture information on various state tables, such as the routes in the routing table, devices in the device inventory, DHCP leases in the DHCP lease table, connected VPNs, clients logged into the wireless network, and much more. These monitors are useful when troubleshooting the current state of the FortiGate and when identifying whether certain objects are in the state table. See:

- Security Fabric widgets and monitors on page 186
- Network widgets and monitors on page 186
- System widgets and monitors on page 187
- Resource Usage widgets on page 187
- Security widgets and monitors on page 188
- User & Authentication widgets and monitors on page 188
- WiFi widgets and monitors on page 188
- WAN Opt. & Cache widgets and monitors on page 189

See also Widgets in VDOM mode on page 189.

FortiView monitors and widgets

FortiView is the FortiOS log view tool and comprehensive monitoring system for your network. FortiView integrates real-time and historical data into a single view on your FortiGate. It can log and monitor network

threats, keep track of administration activities, and more.

Use FortiView monitors and widgets to investigate traffic activity such as user uploads and downloads, or videos watched on YouTube. You can view the traffic on the whole network by user group or by individual. FortiView displays the information in both text and visual format, giving you an overall picture of your network traffic activity so that you can quickly decide on actionable items.

FortiView is integrated with many UTM functions. For example, you can quarantine an IP address directly in FortiView or create custom devices and addresses from a FortiView entry.

The logging range and depth will depend on the FortiGate model.

See Dashboard on page 142.

The following FortiView resources are available as widgets and monitors:

Resource	Widget	Monitor
FortiView AI application	Yes	Yes
FortiView AI Use Cases	Yes	Yes
FortiView Application Bandwidth	Yes	No
FortiView Applications	Yes	Yes
FortiView Cloud Applications	Yes	Yes
FortiView Countries/Regions	Yes	Yes
FortiView Destination Firewall Objects	Yes	Yes
FortiView Destination Interfaces	Yes	Yes
FortiView Destination Owner	Yes	Yes
FortiView Destinations	Yes	Yes
FortiView Interface Pairs	Yes	Yes
FortiView Internal Hubs	No	Yes
FortiView Policies	Yes	Yes
FortiView Proxy Applications	Yes	Yes
FortiView Proxy Destinations	Yes	Yes
FortiView Proxy Policies	Yes	Yes
FortiView Proxy Sessions	Yes	Yes
FortiView Proxy Sources	Yes	Yes
FortiView Proxy Web Sites	Yes	Yes
FortiView SD-WAN Application Performance For an example, see Application performance monitoring analytics NEW on page 216.	Yes	Yes

Resource	Widget	Monitor
FortiView Search Phrases	Yes	Yes
FortiView Servers	Yes	Yes
FortiView Sessions	Yes	Yes
FortiView Source Firewall Objects	Yes	Yes
FortiView Source Interfaces	Yes	Yes
FortiView Sources	Yes	Yes
FortiView Sources - WAN	Yes	Yes
FortiView Traffic Shaping	Yes	Yes
FortiView VPN	Yes	Yes
FortiView Web Categories	Yes	Yes
FortiView Web Sites	Yes	Yes
FortiView ZTNA Servers	Yes	Yes

Security Fabric widgets and monitors

The following Security Fabric resources are available as widgets and monitors:

Resource	Widget	Monitor
Fabric Connector	Yes	No
FortiGate Cloud	Yes	No
FortiTelemetry	Yes	Yes
Security Fabric Status	No	Yes

Network widgets and monitors

The following Network resources are available as widgets and monitors:

Resource	Widget	Monitor
DHCP	Yes	Yes
DNS	Yes	No
Interface Bandwidth	Yes	No
IP Pool Utilization	Yes	Yes

Resource	Widget	Monitor
IPsec	Yes	Yes
Routing	Yes	Yes
SD-WAN	Yes	Yes
Top IP Pools by Assigned IPs	Yes	Yes

System widgets and monitors

The following FortiGate System resources are available as widgets and monitors:

Widget	Widget	Monitor
Administrators	Yes	No
Botnet Activity	Yes	No
Cluster Status	Yes	Yes
HA Status	Yes	No
License Status	Yes	No
System Information	Yes	No
Top System Events	Yes	Yes
Virtual Machine	Yes	No

Resource Usage widgets

The following Resource information is only available as widgets:

Resource	Widget	Monitor
CPU Usage	Yes	No
Disk Usage	Yes	No
Log Rate	Yes	No
Logs Sent	Yes	No
Memory Usage	Yes	No
Session Rate	Yes	No
Sessions	Yes	No

Security widgets and monitors

The following Security resources are available as widgets and monitors:

Resource	Widget	Monitor
Advanced Threat Protection Statistics	Yes	Yes
Assets - Vulnerabilities	Yes	Yes
Compromised Hosts	Yes	Yes
FortiSandbox Files	Yes	Yes
Quarantine	Yes	Yes
Top Endpoint Vulnerabilities	Yes	Yes
Top Failed Authentication	Yes	Yes
Top Threats	Yes	Yes
Top Threats - WAN	Yes	Yes

User & Authentication widgets and monitors

The following User & Authentication resources are available as widgets and monitors:

Widget	Widget	Monitor
Assets	Yes	Yes
Assets - FortiClient	Yes	Yes
Firewall Users	Yes	Yes
FortiGuard Quota	Yes	Yes
FortiSwitch 802.1X Clients	Yes	Yes
Identities	Yes	Yes
Matched Devices	Yes	Yes
Top Admin Logins	Yes	Yes
Top Cloud Users	Yes	Yes

WiFi widgets and monitors

The following WiFi resources are available as widgets and monitors:

Resource	Widget	Monitor
Channel Utilization	Yes	Yes
Clients by FortiAP	Yes	Yes
FortiAP Status	Yes	Yes
Historical Clients	Yes	Yes
Interfering SSIDs	Yes	Yes
Login Failures	Yes	Yes
Rogue APs	Yes	Yes
Signal Strength	Yes	Yes
Top WiFi Clients	Yes	Yes

WAN Opt. & Cache widgets and monitors

The following WAN Optimization and Cache resources are available as widgets and monitors:

Resource	Widget	Monitor
Cache Monitor	Yes	Yes
Peer Monitor	Yes	Yes
WAN Opt. Monitor	Yes	No

Widgets in VDOM mode

Dashboards are created per VDOM when VDOM mode is enabled. For information about VDOM mode, see Virtual Domains on page 3835.

Some dashboards and widgets are not available in Multi-VDOM mode.

The following table lists the available widgets in VDOM mode:

Category	Widgets
FortiView	• FortiView AI Applications • FortiView AI Use Cases • FortiView Application Bandwidth • FortiView Applications • FortiView Cloud Applications • FortiView Countries/Regions

Category	Widgets
	• FortiView Destination Firewall Objects • FortiView Destination Interfaces • FortiView Destination Owner • FortiView Destinations • FortiView Interface Pairs • FortiView Policies • FortiView Proxy Applications • FortiView Proxy Destinations • FortiView Proxy Policies • FortiView Proxy Sessions • FortiView Proxy Sources • FortiView Proxy Web Sites • FortiView SD-WAN Application Performance • FortiView Search Phrases • FortiView Servers • FortiView Sessions • FortiView Source Firewall Objects • FortiView Source Interfaces • FortiView Sources • FortiView Sources - WAN • FortiView Traffic Shaping • FortiView VPN • FortiView Web Categories • FortiView Web Sites • FortiView ZTNA Servers
Security Fabric	• Fabric Connector • FortiGate Cloud • FortiTelemetry • Security Fabric Status
Network	• DHCP • DNS • Interface Bandwidth • IP Pool Utilization • IPsec • Routing • SD-WAN • Top IP Pools by Assigned IPs The *Interface Bandwidth* widget can monitor a maximum of 25 interfaces.

Category	Widgets
System	• Administrators • Botnet Activity • Cluster Status • HA Status • License Status • System Information • Top System Events • Virtual Machine
Resource Usage	• CPU Usage • Disk Usage • Log Rate • Logs Sent • Memory Usage • Session Rate • Sessions
Security	• Advanced Threat Protection Statistics • Assets - Vulnerabilities • Compromised Hosts • FortiSandbox Files • Quarantine • Top Endpoint Vulnerabilities • Top Failed Authentication • Top Threats • Top Threats - WAN
User & Authentication	• Assets • Assets - FortiClient • Firewall Users • FortiGuard Quota • FortiSwitch 802.1X Clients • Identities • Matched Devices • Top Admin Logins • Top Cloud Users
WiFi	• Channel Utilization • Clients By FortiAP • FortiAP Status • Historical Clients • Interfering SSIDs • Login Failures

Category	Widgets
	• Rogue APs • Signal Strength • Top WiFi Clients
WAN Opt. & Cache	• Cache Monitor • Peer Monitor • WAN Opt. Monitor

Using dashboards of widgets and monitors

Dashboards of widgets and monitors can be added, edited, and removed:

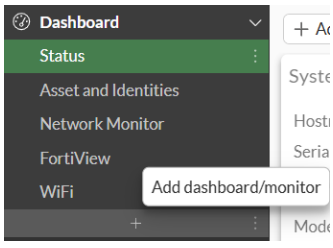
- Adding dashboards of widgets and monitors on page 192
- Editing dashboards of widgets and monitors on page 194
- Using widgets on page 195
- Resetting all dashboards on page 197
- Viewing device dashboards in the Security Fabric on page 197
- Creating a fabric system and license dashboard on page 198
- Adding an Agentless VPN dashboard on page 200
- Adding a FortiTelemetry dashboard on page 201

Adding dashboards of widgets and monitors

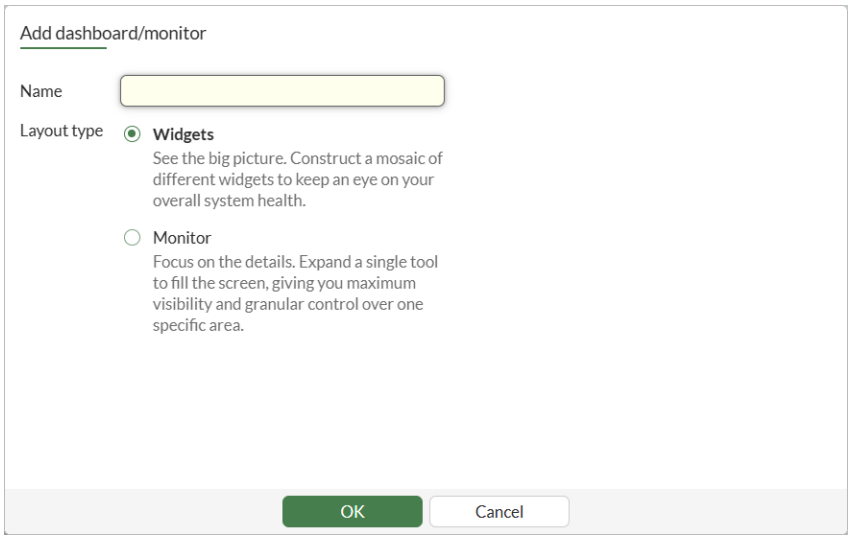
You can create dashboards of widgets and monitor tabs.

To add a new dashboard:

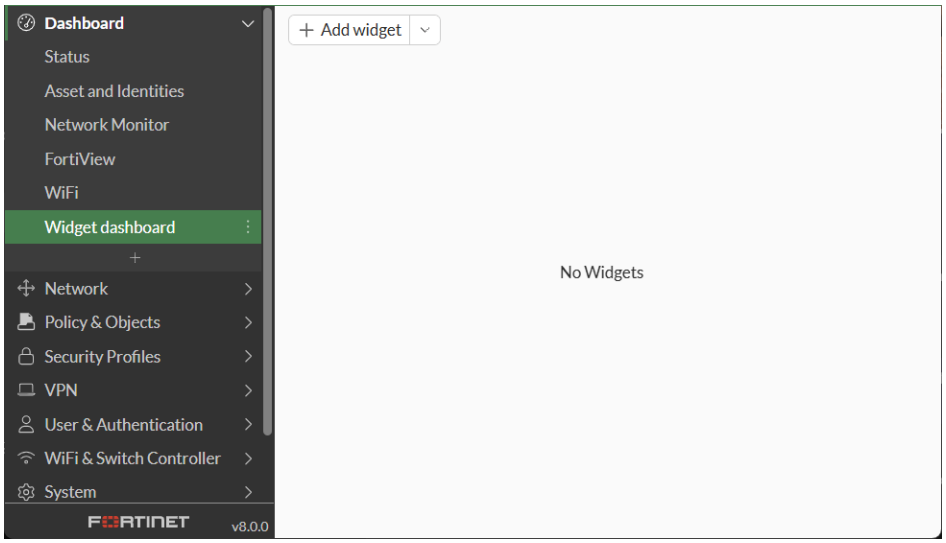
1. Under *Dashboard*, click the *+ Add Dashboard/monitor* button.



The *Add Dashboard/monitor* window opens.



2. Add widgets to the dashboard:
   a. Enter a name for the dashboard of widgets.
   b. Set *Layout type* to *Widgets*.
   c. In the *Name* field, enter a name for the tab of widgets and click *OK*. The new dashboard opens.



   d. Click *+ Add widget*. The *Add dashboard widget* pane is displayed.
   e. Select a widget. The *Edit dashboard* widget pane is displayed.
   f. Set the options, and click *OK*. The widget is added to the dashboard.
   g. Add additional widgets as needed.
   h. Click *Close*. The dashboard displays the tab of widgets.
3. Add tabs of monitors to the dashboard:
   a. Leave the Name field blank. It will be automatically populated with the name of the selected monitor.
   b. Beside *+ Add widget*, click the dropdown arrow, and select *+ Add tab*. The *Add dashboard tab* pane is displayed.
   c. In the *Name* field, enter a name for the monitor tab.
   d. Set *Layout type* to *Monitor*, and click *Choose monitor*. The *Add monitor* pane is displayed.

   e. Set the options, and click *OK*. The monitor name is displayed with a *Settings* button. Click *Settings* to edit the settings.
   f. Click *OK*. The monitor tab is added to the dashboard.
4. Select a tab of widgets and click *+ Add widget* to add more widgets.
5. Click *+ Add tab* to add more monitor tabs.

Editing dashboards of widgets and monitors

You can add widgets and monitor tabs to dashboards.

To add more widgets to a dashboard tab:

1. Go to *Dashboard* and select a dashboard in the tree menu.
2. Select a tab of widgets.
3. Click *+Add widget*. The *Add dashboard tab* pane opens.
4. Select a widget. The *Edit dashboard widget* pane is displayed.
5. Set the options, and click *OK*. The widget is added to the dashboard.
6. Add additional widgets as needed.
7. When done, click *Close*. The new widgets are displayed on the tab.
8. (Optional) Click and drag the widget to the desired location in the dashboard.

To add a new tab of widgets to a dashboard:

1. Go to *Dashboard* and select a dashboard in the tree menu.
2. In the banner, click *+Add tab*. The *Add dashboard tab* pane opens.
3. In the *Name* field, enter a name for the tab of widgets.
4. Set *Layout type* to *Widgets* and click *OK*. The new tab is displayed.
5. Add widgets:
   a. Click *+ Add widget*. The *Add dashboard widget* pane is displayed.
   b. Select a widget. The *Edit dashboard widget* pane is displayed.
   c. Set the options, and click *OK*. The widget is added to the dashboard.
   d. Add additional widgets as needed.
   e. Click *Close*. The widgets are displayed on the new tab.
6. (Optional) Click and drag the widget to the desired location in the dashboard.

To add monitor tabs to a dashboard:

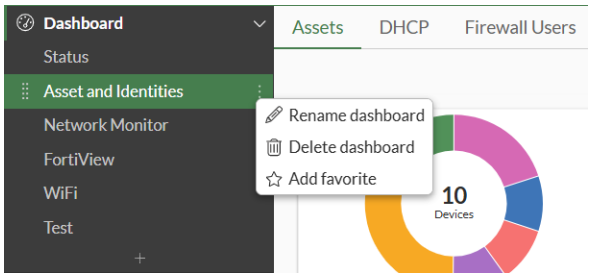
1. Go to *Dashboard* and select a dashboard in the tree menu.
2. Click *+ Add* tab. The *Add dashboard tab* pane is displayed.
3. Add tabs of monitors to the dashboard:
4. Set *Layout type* to *Monitor*, and click *Choose monitor*. The *Add monitor* pane is displayed.
5. Select a monitor. The *Add <monitor name> as new tab* pane is displayed.
6. Set the options, and click *OK*. The monitor name is displayed with a *Settings* button.
7. Click *OK* to add the monitor tab to the dashboard.

To manage tabs on a dashboard:

1. Go to *Dashboard* and select a dashboard in the tree menu.
2. Beside *+ Add tab*, click the arrow and select *Manage tabs*. The *Manage tabs* pane displays the tab names for the dashboard.
3. Beside a tab name, click *X* to delete the tab.
4. Under the tab names, click *+* to add a new tab.
5. Beside a tab name, click the *Settings* button to edit the tab settings:
   a. Change the layout between Widgets and Monitor.
   b. Edit the settings for the monitor.
   c. Click *OK* to save the settings.
6. Click *OK* to save changes to the dashboard.

To delete a dashboard:

1. Click the *Actions* menu next to the dashboard and select *Delete Dashboard*.



2. Click *Delete Dashboard* . The *Confirm* dialog opens.
3. *Click OK*.

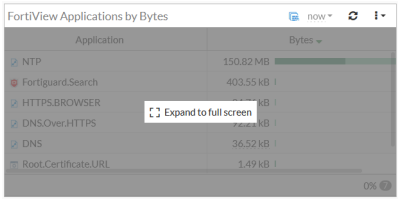
You cannot delete the *Status* dashboard.

Using widgets

You can temporarily expand a widget to a standalone monitor, change the view type, configure tables, and filter data.

To expand a dashboard widget to a monitor:

1. Hover over the widget and click *Expand to full screen*.

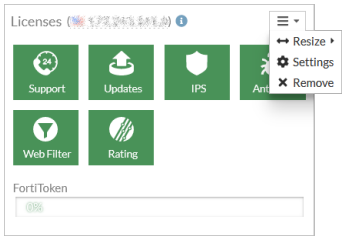


Full screen mode is not supported in all widgets.

2. On the top-left, click the back arrow to return to the widget.

To view the widget settings:

1. Click the menu dropdown at the right side of the widget and select *Settings*.

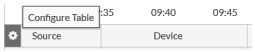


2. Configure the widget settings and click *OK*.

The settings will vary depending on the widget.

To configure a table in a monitor:

1. Hover over the left side of the table header and click *Configure Table*.



2. Configure the table options:

Option	Description
Best Fit Columns	Resizes all of the columns in a table to fit their content.
Reset Table	Resets the table to the default view.
Export	Export the table data to a CSV or JSON file. Only available for applicable tables.
Select Columns	Adds or removes columns from the view.

3. Click *Apply*.

To filter or configure a column in a table:

1. Hover over a column heading, and click *Filter/Configure Column*.

2. Configure the column options.

Option	Description
Resize to Contents	Resizes the column to fit the content.
Group by this Column	Groups the table rows by the contents in the selected column.

3. Click *Apply*.
4. To filter a column, enter a value in the *Filter* field, and click *Apply*.

> Filtering is not supported in all widgets.

See also Widgets in VDOM mode on page 189.

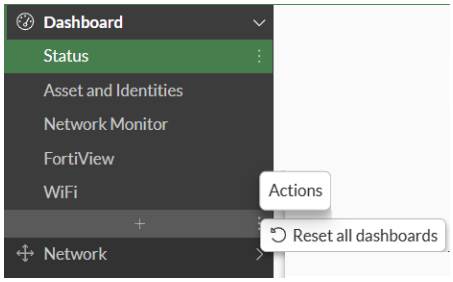
Resetting all dashboards

You can use the GUI to reset all dashboards to the default settings.

> Resetting all dashboards will delete any custom dashboards, monitors, and widgets.

To reset all dashboards:

1. Click the *Actions* menu next to *Add dashboard/monitor* and click *Reset all dashboards*.

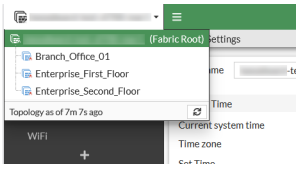


2. In the *Confirmation* window, click *OK*.

Viewing device dashboards in the Security Fabric

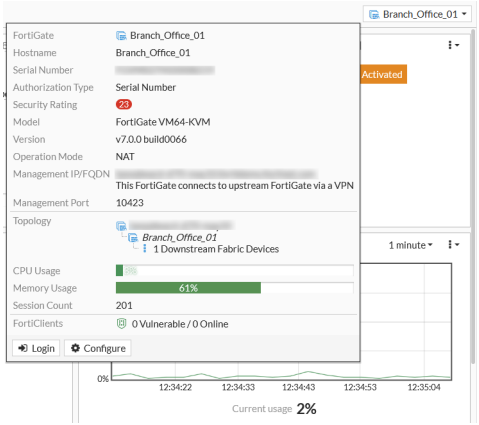
Use the device dropdown to view the dashboards in downstream Fabric devices. You can also create dedicated device dashboards or log in and configure Fabric devices.

To view the dashboards in Fabric devices, click the device dropdown at the left side of the page, and select a device from the list.



The device dropdown is available in the *Status*, *Security*, *Network*, *Assets & Identities*, and *WiFi* dashboards. You can also enable the dropdown when you create a dashboard.

To log in to or configure a Fabric device, hover over the device name until the device dialog opens and then select *Login* or *Configure*.

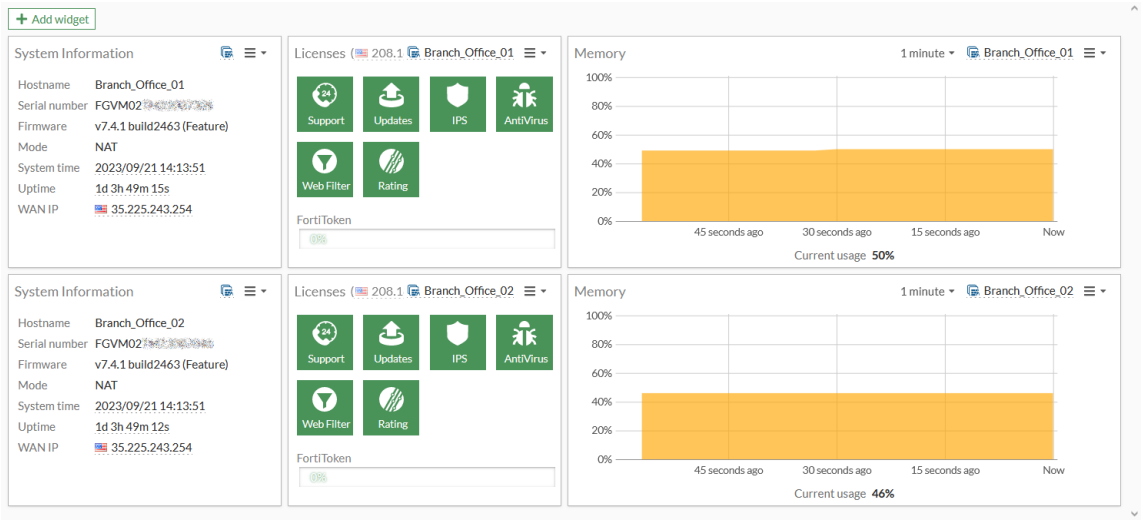


Creating a fabric system and license dashboard

Create a dashboard summary page to monitor all the Fabric devices in a single view. You can use this dashboard to monitor aspects of the devices such as system information, VPN and routing.

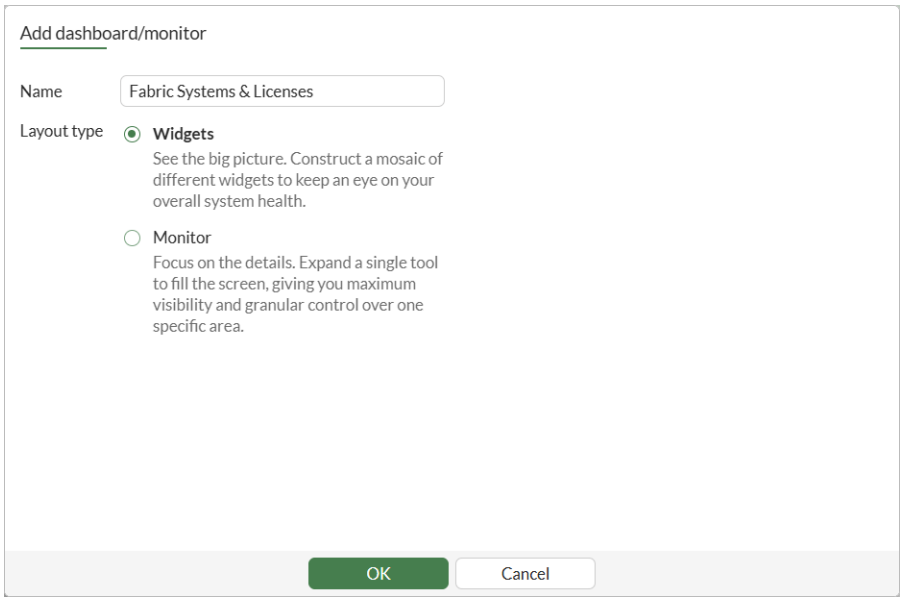
Example

The following image is an example of a *Fabric System & License* dashboard to monitor the *System Information*, *Licenses*, and *Memory* usage for *Branch_Office_01* and *Branch_Office_02*.



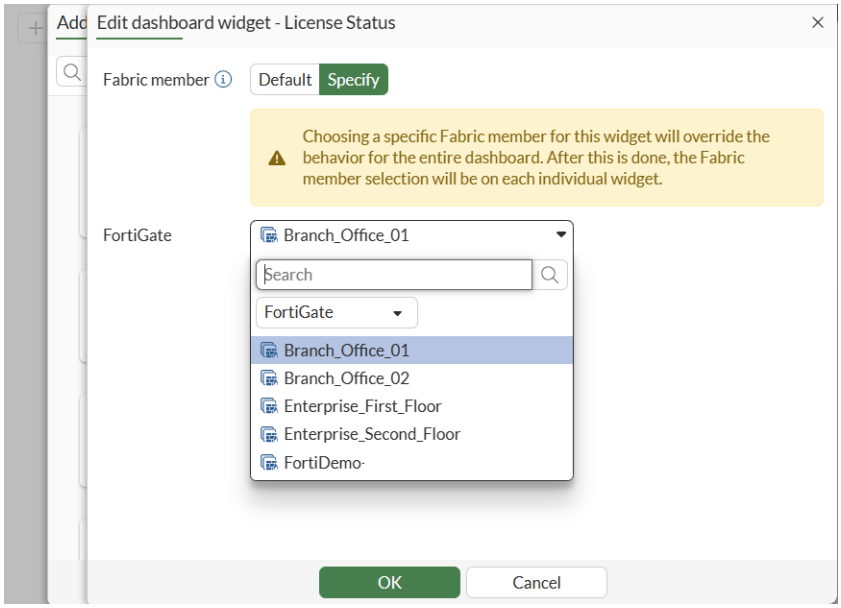
To create a system dashboard:

1. Click the *Add dashboard/monitor* button. The *Add dashboard/monitor* window opens.
2. In the *Name* field, enter a name such as *Fabric System & Licenses*, select *Widgets*, and click *OK*.



The new dashboard appears under *Dashboard* in the tree menu.

3. Add one or more widgets:
   a. In the banner, click *Add Widget*. The *Add dashboard widget* window opens.
      Use the *Search* field to search for a specific widget (for example, *License Status*, *System Information*, and *Memory Usage*).
   b. Click the *Add* button next to widget. The *Add Dashboard Widget* window opens.
   c. In the *Fabric member* area, select *Specify*, select a device in the Security Fabric, and click *OK*. The widget is added to the dashboard.

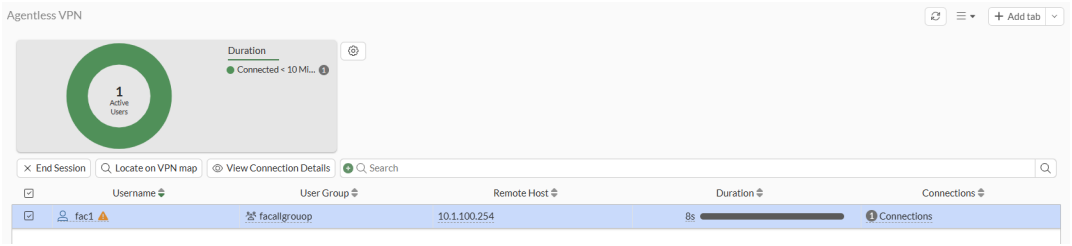


   d. Repeat this step for all the devices you want to view in the dashboard.

4. Click *Close*.
5. (Optional) Arrange the widgets in the dashboard by Fabric device.

Adding an Agentless VPN dashboard

The Agentless VPN monitor displays remote user logins and active connections. You can use the monitor to disconnect a specific connection. The monitor will notify you when VPN users have not enabled two-factor authentication.



To view the Agentless VPN monitor in the GUI:

1. Under *Dashboard*, click *+ (Add dashboard/monitor)*.
2. Enter a name for the dashboard/monitor.
3. Set *Layout type* to *Monitor*, and click *Choose monitor*.
4. In the *Search* box, type *Agentless VPN* and press Enter.
5. Click the *Agentless VPN* tile. The *Add Agentless VPN as new tab* pane is displayed.
6. Complete the options and click *OK*.
7. Click *OK*.

To filter or configure a column in the table, hover over the column heading and click the *Filter/Configure Column* button.

To disconnect a user:

1. Select a user in the table.
2. In the table, right-click the user, and click *End Session*. The Confirm window opens.
3. Click *OK*.

To monitor Agentless VPN users in the CLI:

```
# get vpn ssl monitor
```

Sample output

```
Agentless VPN Login Users:
Index   User    Group   Auth Type      Timeout         From     HTTP in/out    HTTPS in/out
0       amitchell       TAC     1(1)            296     10.100.64.101  3838502/11077721        0/0
1       mmiles          Dev     1(1)            292     10.100.64.101  4302506/11167442        0/0
```

```
Agentless VPN sessions:
Index   User    Group   Source IP      Duration        I/O Bytes       Tunnel/Dest IP
```

Adding a FortiTelemetry dashboard

This topic applies only to a FortiGate acting as a FortiTelemetry Controller with *FortiTelemetry* enabled on the *System > Feature Visibility* page. For a summary of FortiTelemetry-related tasks that you can do in FortiOS, see FortiTelemetry on page 139. See also *FortiTelemetry Administration Guide*.

Add FortiTelemetry monitors to view and monitor data collected by the agents and analyzed by FortiTelemetry Cloud. After FortiTelemetry Cloud analyzes the data, it returns to the FortiTelemetry Controller an application experience score and an application failure rate, which you can view in the FortiTelemetry monitor.

To add a FortiTelemetry dashboard:

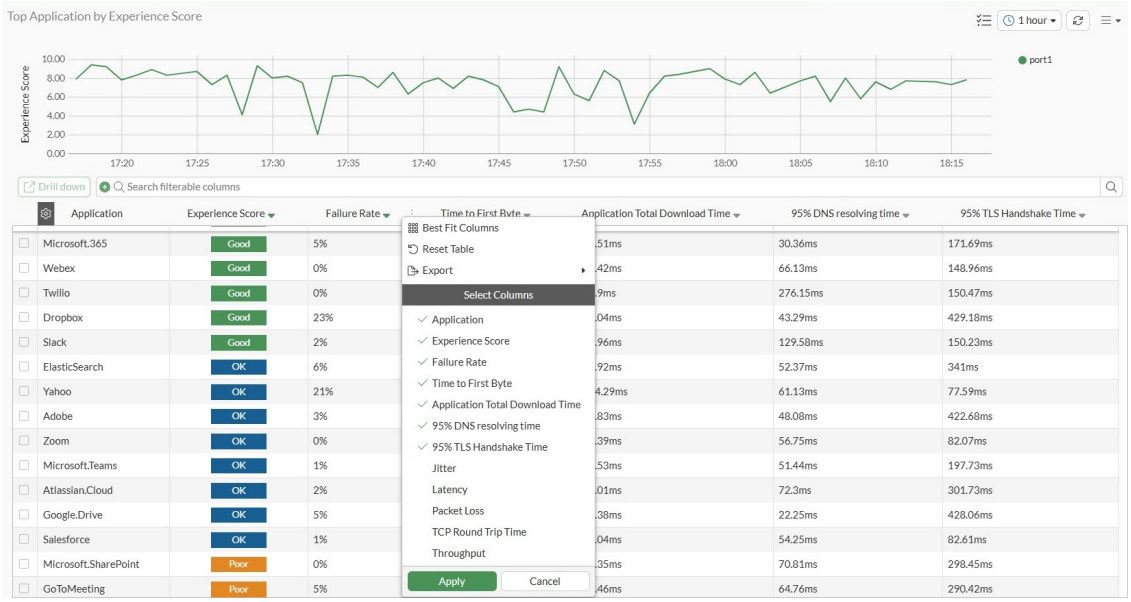
1. Under *Dashboard*, click *+ (Add dashboard/monitor)*. The *Add dashboard/monitor* window opens.
2. Enter a name for the dashboard/monitor.
3. Beside *Layout type*, select *Monitor*, and click *Choose monitor*. The *Add monitor* pane opens.
4. In the *Search* box, enter *FortiTelemetry* and press Enter.
5. Click *+ (Add)* next to *FortiTelemetry*. The *Add FortiTelemetry as new tab* pane opens.
6. Set the following options, and click *OK* to add the monitor:

FortiGate	Select the FortiGate acting as the FortiTelemetry Controller.
Source	Select one of the following sources: • Application • Agent • Source interface • Destination interface • Profile • Policy
Time period	Select a time range for the data: • 1 hour • 24 hours • 7 days
Sort by	Select how to sort the monitored data.

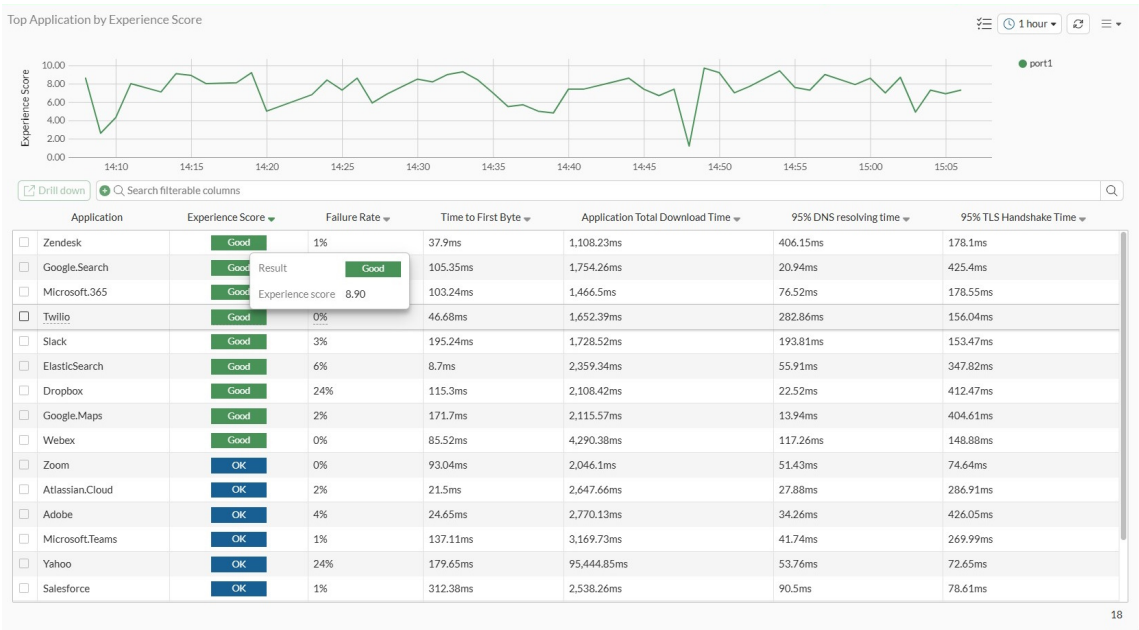
7. Click *OK* to close the *Add dashboard/monitor* window.

To view FortiTelemetry data:

1. Under *Dashboard*, click the *FortiTelemetry* monitor that you added.
2. Click the gear icon to customize what columns display:



3. Hover over the *Experience Score* rating to view the experience score for each application:



4. Select the application row in the table to display a *Drill Down* button, and click *Drill Down* to view details about the agent on the *Agent* tab:

5. You can continue drilling down to additional details. For example, select the agent to display a *Drill Down* button, and click *Drill Down* to view details about the destination interface on the *Destination Interface* tab, and so on.

Examples

This section provides the following dashboard examples:

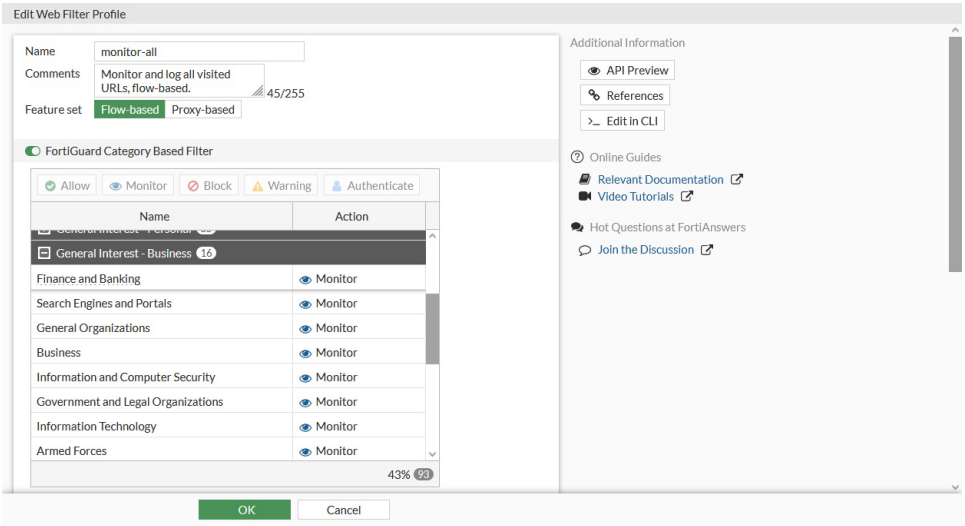
- Top websites and sources by category on page 203
- Cloud application view on page 206
- Application performance monitoring analytics NEW on page 216

Top websites and sources by category

You can use FortiGuard web categories to populate the category fields in various FortiView monitors such as *FortiView Web Categories*, *FortiView Websites* or *FortiView Sources*. To view the categories in a monitor, the web filter profile must be configured to at least monitor for a FortiGuard category based on a web filter and applied to a firewall policy for outbound traffic.

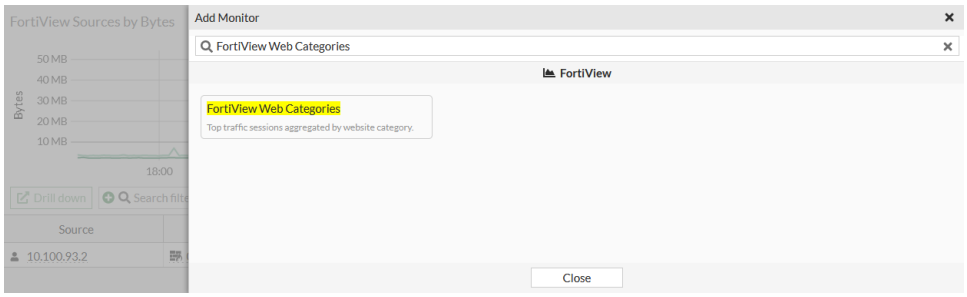
To verify the web filter profile is monitor-only:

1. Go to *Security Profiles* > *Web Filter*.
2. Double-click a web filter that is applied to an outbound traffic firewall policy. The *Edit Web Filter Profile* window opens.
3. Ensure *FortiGuard Category Based Filter* is enabled.
   In this example, the *General Interest - Business* categories are monitor-only.



To create a Web categories monitor:

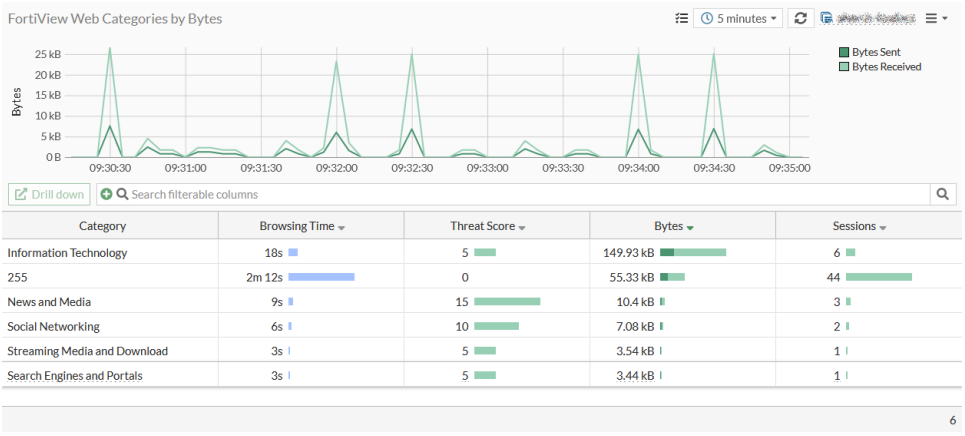
1. Under *Dashboard*, click *+ (Add dashboard/monitor*. The *Add dashboard tab* window opens.
2. Enter a name for the dashboard/monitor.
3. Set *Layout type* to *Monitor*, and click *Choose monitor*.
4. In the *Search* field, type *FortiView Web Categories* and click the *Add* button next to the monitor name.



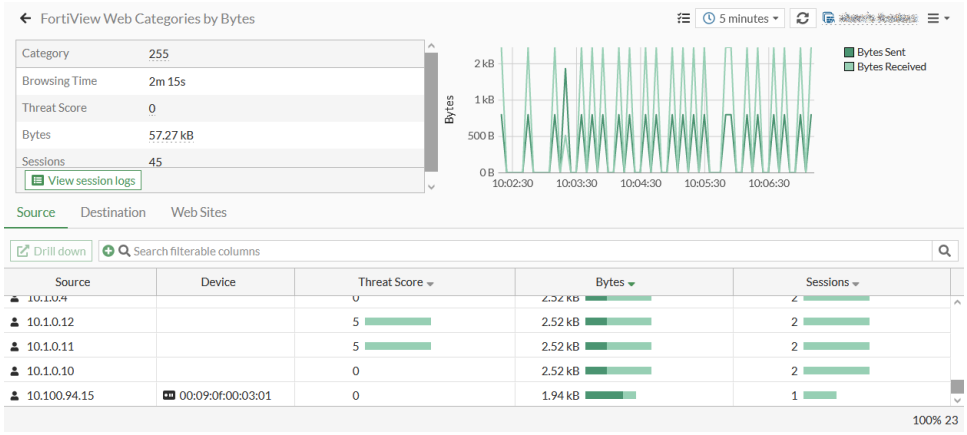
5. In the *FortiGate* area, select the FortiGate(s) from the dropdown.
6. In the *Data Source* area, click *Best Available Device* or *Specify* to select a device in the security fabric.
7. From the *Time Period* dropdown, select a time period greater than *Now*.
8. From the *Sort By* dropdown, select *Browsing Time*, *Threat Score*, *Bytes*, or *Sessions*.
9. Click *OK*. The monitor is added.
10. Click *OK* to close the *Add dashboard/monitor* tab.

Viewing the web filter category

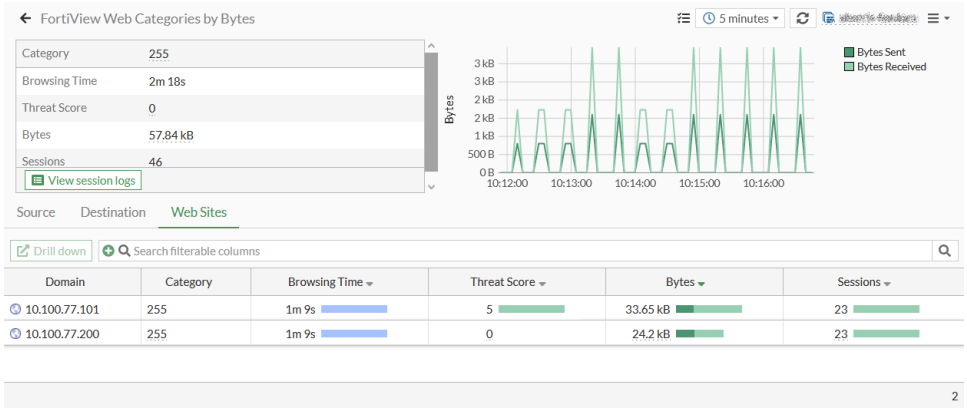
The web filter category name appears in the *Category* column of the dashboard.



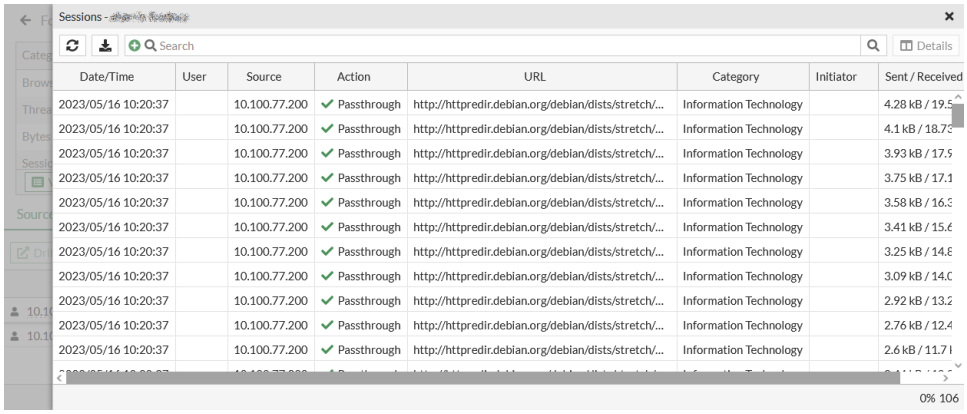
Drill down an entry in the table.



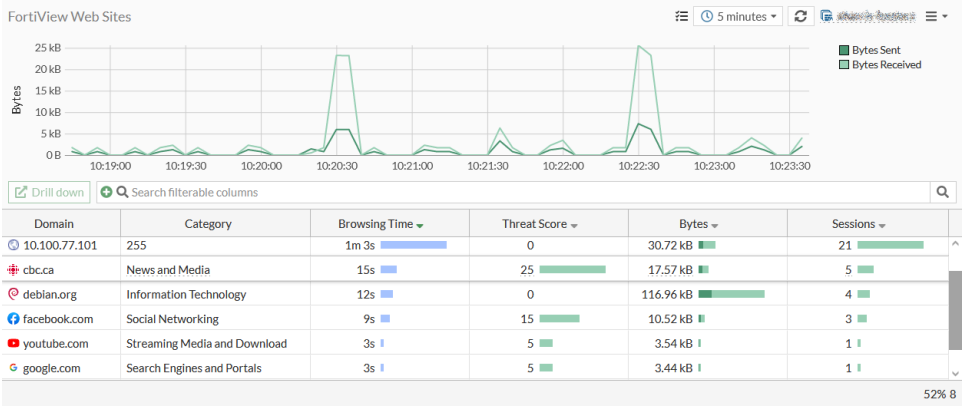
Click the *Web Sites* tab. The category name appears in the *Category* column.



Click *View session logs* to a view a list of the session logs. The category name appears in the *Category* column.



The category name also appears in the *Category* column in the *FortiView Websites* monitor and when drilling down in the *FortiView Sources* monitor.



Cloud application view

To see different cloud application views, set up the following:

- A FortiGate with a firewall policy that uses the *Application Control* security profile.
- A FortiGate with log data from the local disk or FortiAnalyzer.
- Optional but highly recommended: *SSL Inspection* set to *deep-inspection* in the related firewall policies.

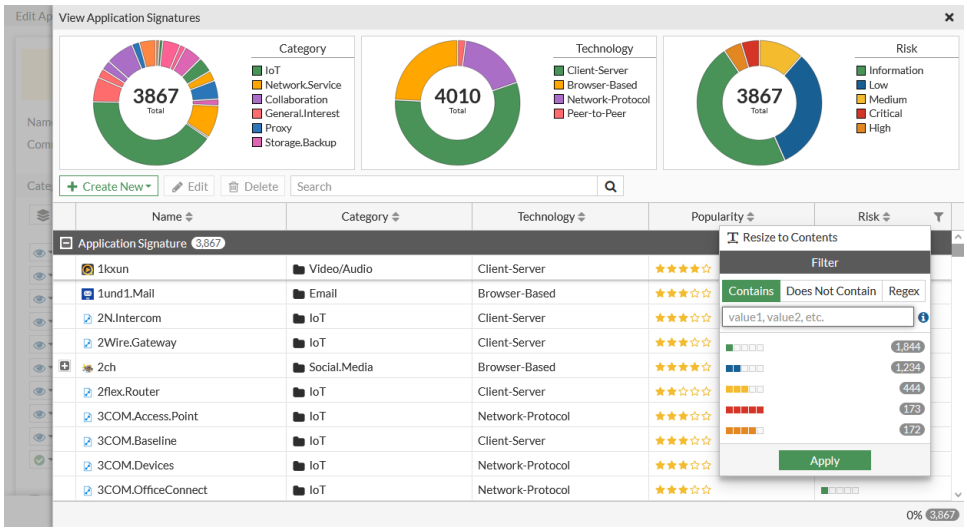
Viewing cloud applications

All cloud applications require *SSL Inspection* set to *deep-inspection* on the firewall policy. For example, `Facebook_File.Download` can monitor Facebook download behavior which requires *SSL deep-inspection* to parse the deep information in the network packets.

To view cloud applications:

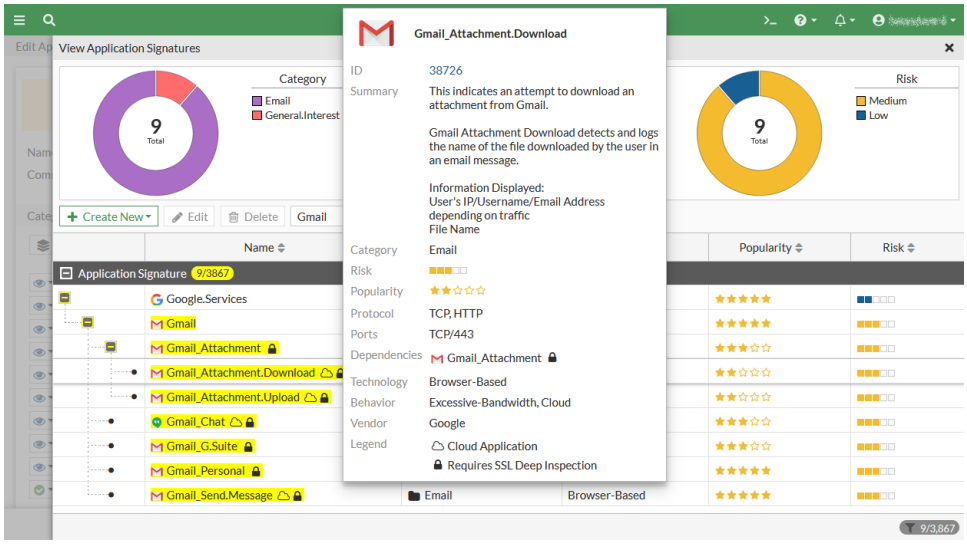
1. Go to *Security Profiles* > *Application Control*.
2. Select an Application Control profile that is used by the firewall policy and click *Edit*.
3. On the *Edit Application Sensor* page, click *View Application Signatures*.
4. Hover over a column heading or the *Application Signature* bar. On the right, click the filter icon to filter the applications.

Cloud applications have a cloud icon next to them. The lock icon indicates that the application requires SSL deep inspection.



5. Hover over an item to see its details.

   This example shows *Gmail_Attachment.Download*, a cloud application signature based sensor which requires SSL deep inspection. If any local network user behind the firewall logs into Gmail and downloads a Gmail attachment, that activity is logged.



Applications with cloud behavior

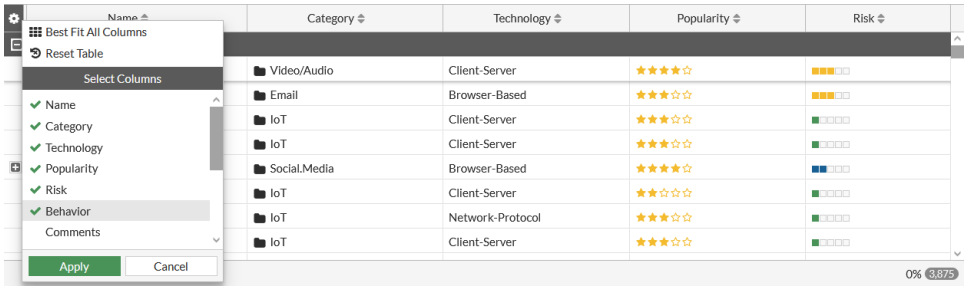
Applications with cloud behavior is a superset of cloud applications.

Some applications do not require SSL deep inspection, such as Facebook, Gmail, and YouTube. This means that if any traffic trigger application sensors for these applications, there is a FortiView cloud application view for that traffic.

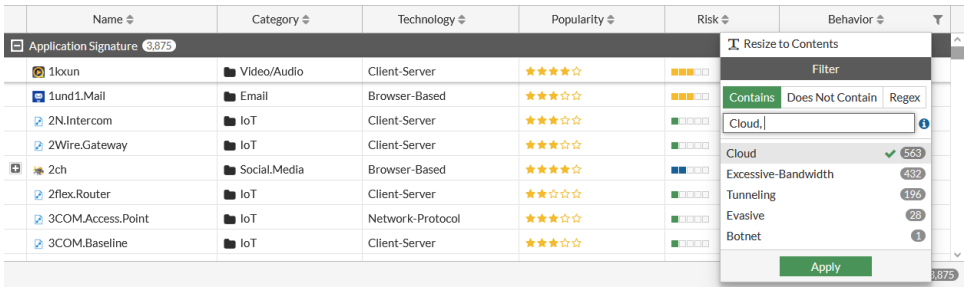
Other applications require SSL deep inspection, such as Gmail attachment, Facebook_Workplace, and so on.

To view applications with cloud behavior:

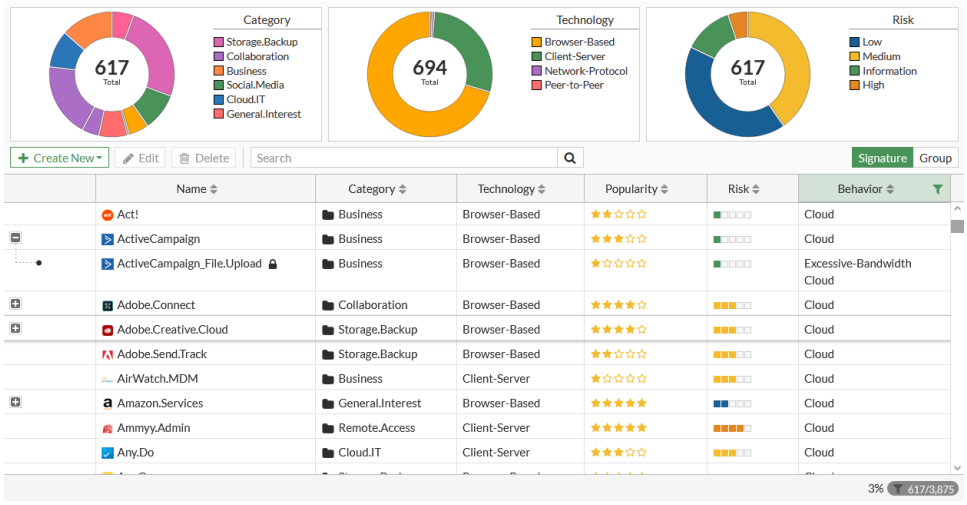
1. In the *Application Signature* page, ensure the *Behavior* column is displayed. If necessary, add the *Behavior* column.
   a. Hover over the left side of the table column headings to display the *Configure Table* icon.
   b. Click *Configure Table* and select *Behavior*.
   c. Click *Apply*.



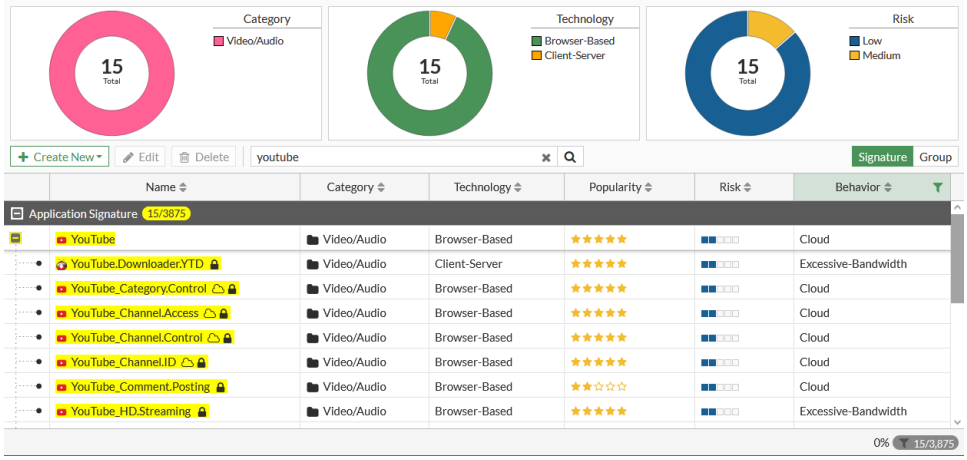
2. Click the filter icon in the *Behavior* column and select *Cloud* to filter by Cloud. Then click *Apply*.



3. The *Application Signature* page displays all applications with cloud behavior.

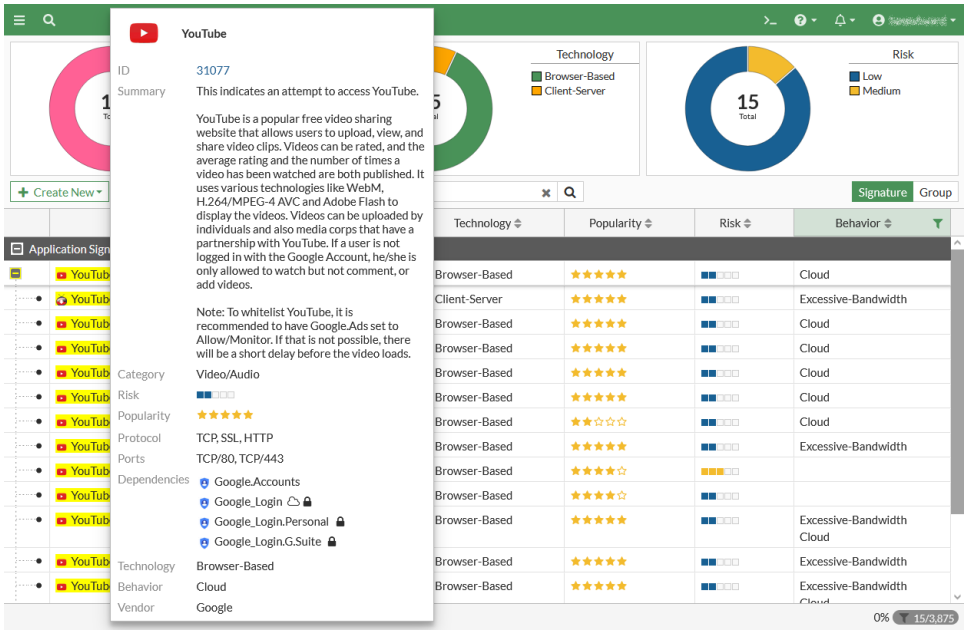


4. Use the *Search* box to search for applications. For example, you can search for *youtube*.



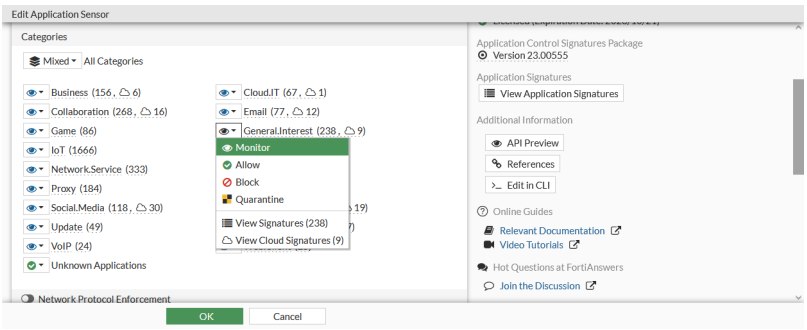
5. Hover over an item to see its details.

   This example shows an application sensor with no lock icon which means that this application sensor does not require SSL deep inspection. If any local network user behind the firewall tries to navigate to the YouTube website, that activity is logged.



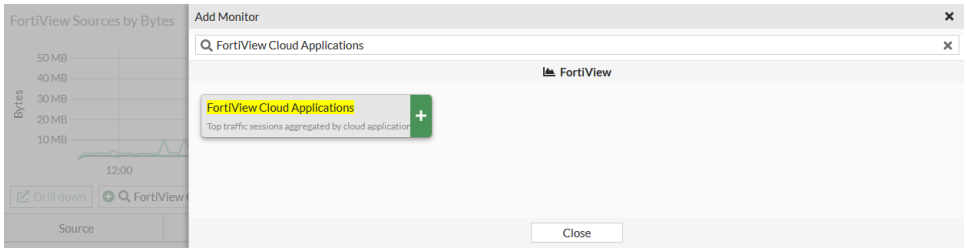
Configuring the Cloud Applications monitor

Go to *Security Profiles* > *Application Control* and edit a profile. On the *Edit Application Sensor* page in the *Categories* section, the eye icon next to a category means that category is monitored and logged.

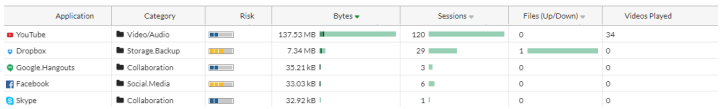


To add the Cloud Applications monitor in the GUI:

1. Under *Dashboard*, click *+ (Add dashboard/monitor)*. The *Add dashboard tab* window opens.
2. Enter a name for the dashboard/monitor.
3. Set *Layout type* to *Monitor*, and click *Choose monitor*.
4. In the *Search* field, enter *FortiView Cloud Applications* and click the *Add* button next to the monitor.



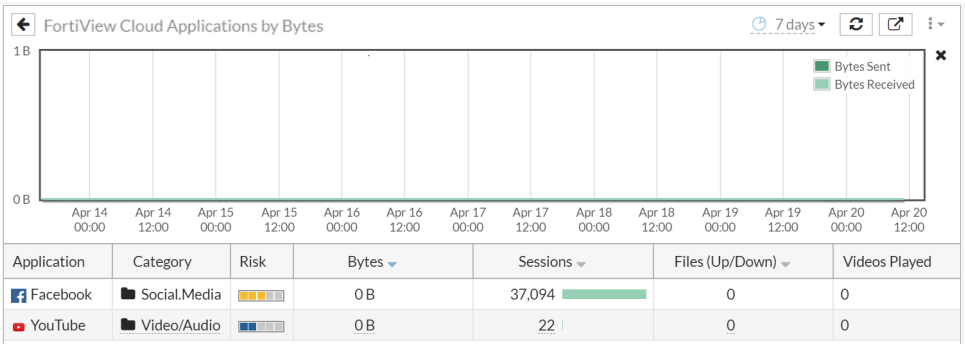
5. In the *FortiGate* area, select the FortiGate(s) from the dropdown.
6. In the *Data Source* area, click *Best Available Device* or *Specify* to select a device in the security fabric.
7. From the *Time Period* dropdown, select a time period greater than *Now*.
8. From the *Sort By* dropdown, select *Bytes*, *Sessions*, *Files (Up/Down)*, or *Videos Played*.
9. Click *OK*. The monitor is added.
10. Click *OK* to close the *Add dashboard* tab.
11. Open the monitor. If SSL deep inspection is enabled in the related firewall policy, then the monitor shows the additional details that are logged, such as *Files (Up/Down)* and *Videos Played*.
    - For YouTube, the *Videos Played* column is triggered by the *YouTube_Video.Play* cloud application sensor. This shows the number of local network users who logged into YouTube and played YouTube videos.
    - For Dropbox, the *Files (Up/Down)* column is triggered by *Dropbox_File.Download* and *Dropbox_File.Upload* cloud application sensors. This shows the number of local network users who logged into Dropbox and uploaded or downloaded files.



Using the Cloud Applications monitor

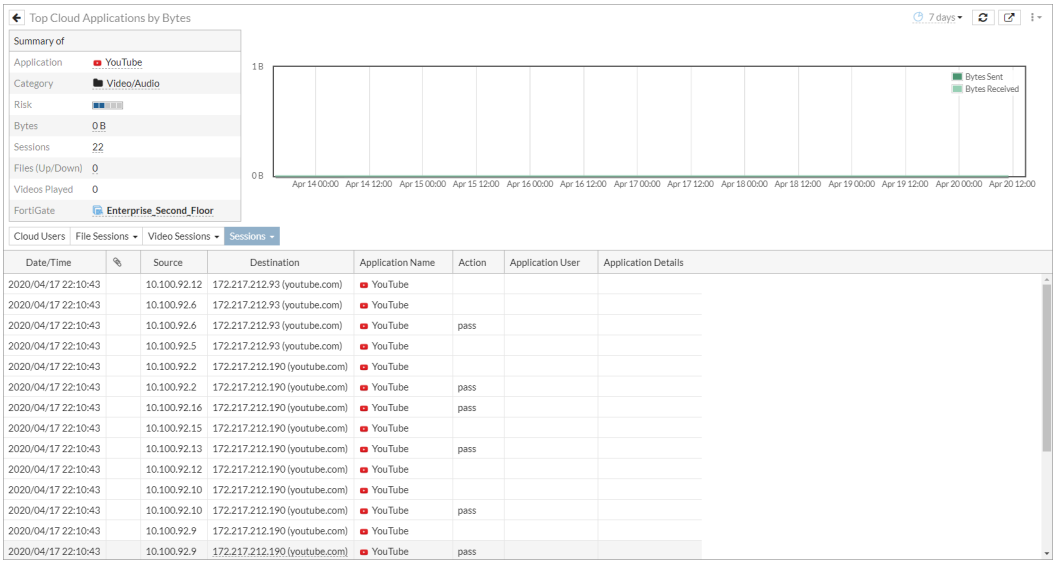
To see additional information in the Cloud Applications monitor:

1. Under *Dashboard*, click the new monitor to open it.



2. For details about a specific entry, double-click the entry or right-click the entry and select *Drill Down to Details*.
3. To see all the sessions for an application, click *Sessions*.

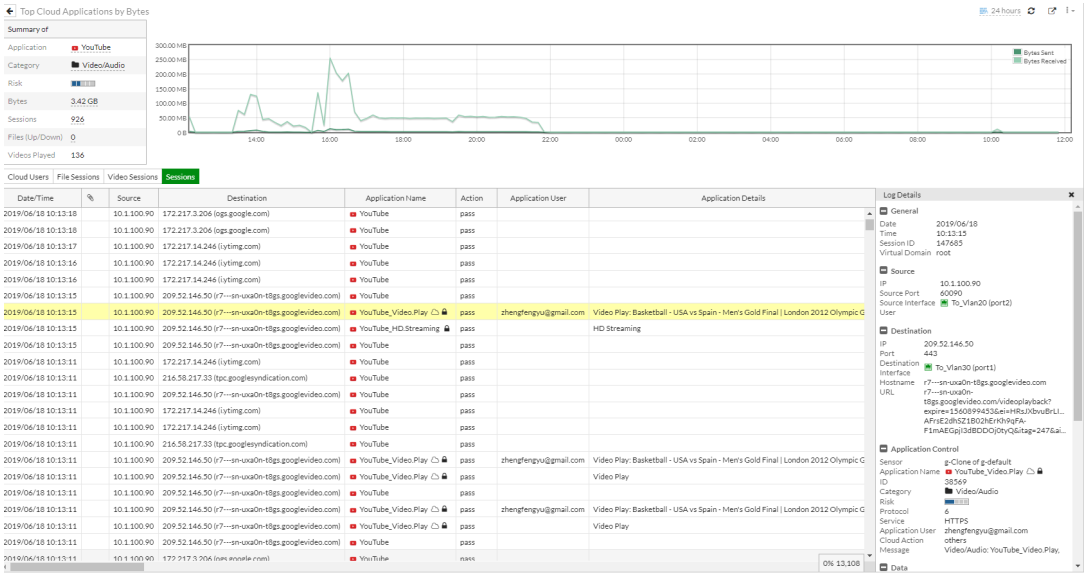
   In this example, the *Application Name* column shows all applications related to YouTube.



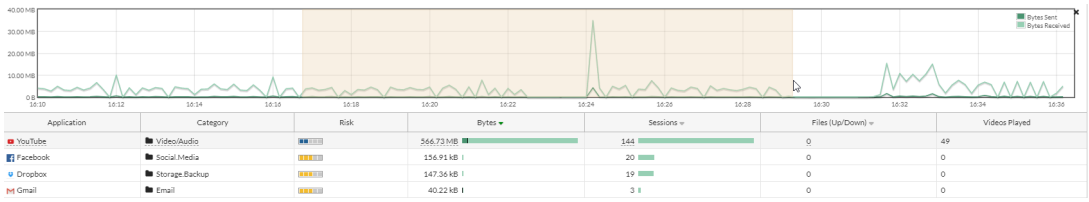
4. To view log details, double-click a session to display the *Log Details* pane.

Sessions monitored by SSL deep inspection (in this example, Youtube_Video.Play) captured deep information such as *Application User*, *Application Details*, and so on. The *Log Details* pane also shows additional deep information such as application *ID*, *Message*, and so on.

Sessions not monitored by SSL deep inspection (YouTube) did not capture the deep information.



5. To display a specific time period, select and drag in the timeline graph to display only the data for that time period.



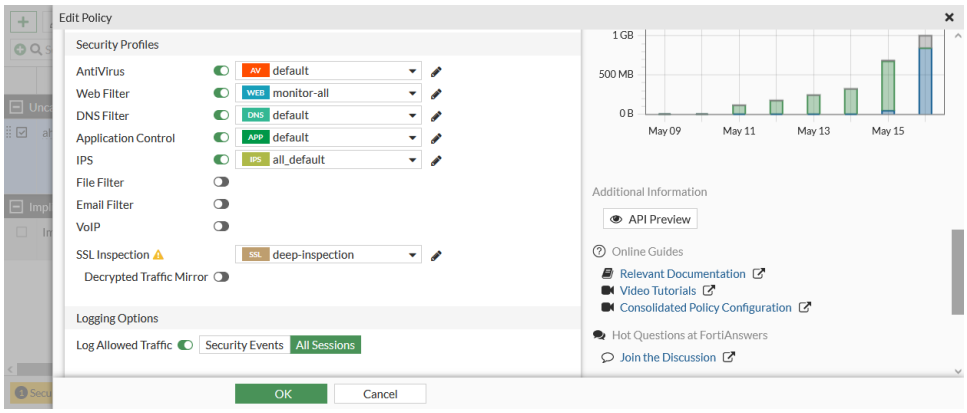
Top application: YouTube example

Monitoring network traffic with SSL deep inspection

This example describes how to monitor network traffic for YouTube using *FortiView Applications* view with SSL deep inspection.

To monitor network traffic with SSL deep inspection:

1. Create a firewall policy with the following settings:
   - *Application Control* is enabled.
   - *SSL Inspection* is set to *deep-inspection*.
   - *Log Allowed Traffic* is set to *All Sessions*.



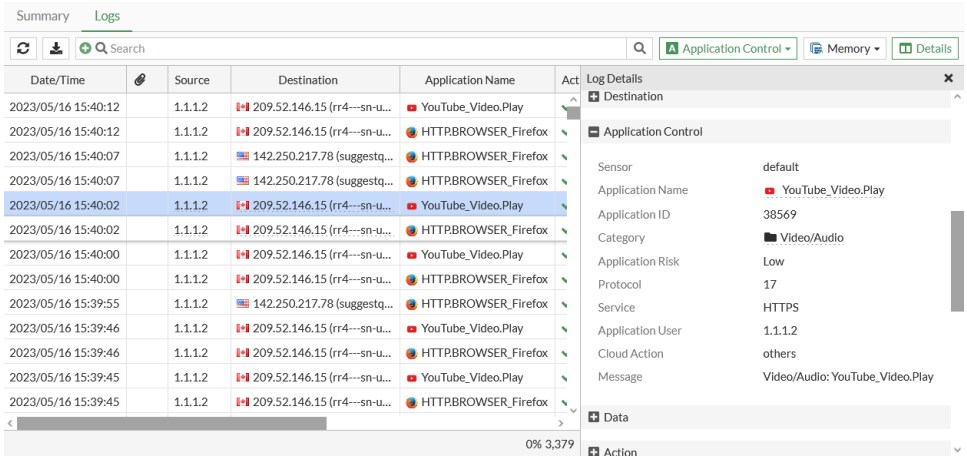
2. Go to *Security Profiles* > *Application Control*.
3. Select a related Application Control profile used by the firewall policy and click *Edit*.
4. Because YouTube cloud applications are categorized into *Video/Audio*, ensure the *Video/Audio* category is monitored. Monitored categories are indicate by an eye icon.
5. Click *View Application Signatures* and hover over YouTube cloud applications to view detailed information about YouTube application sensors.
6. Expand *YouTube* to view the Application Signatures associated with the application.

Application Signature	Description	Application ID
YouTube_Video.Access	An attempt to access a video on YouTube.	16420
YouTube_Channel.ID	An attempt to access a video on a specific channel on YouTube.	44956
YouTube_Comment.Posting	An attempt to post comments on YouTube.	31076
YouTube_HD.Streaming	An attempt to watch HD videos on YouTube.	33104
YouTube_Messenger	An attempt to access messenger on YouTube.	47858
YouTube_Video.Play	An attempt to download and play a video from YouTube.	38569
YouTube_Video.Upload	An attempt to upload a video to YouTube.	22564
YouTube	An attempt to access YouTube. This application sensor does not depend on SSL deep inspection so it does not have a cloud or lock icon.	31077
YouTube_Channel.Access	An attempt to access a video on a specific channel on YouTube.	41598

To view the application signature description, click the ID link in the information window.

7. On the test PC, log into YouTube and play some videos.
8. On the FortiGate, go to *Log & Report* > *Security Events*, select *Application Control*, and look for log entries for browsing and playing YouTube videos.

In this example, note the *Application User* and *Application Details*. Also note that the *Application Control ID* is *38569* showing that this entry was triggered by the application sensor *YouTube_Video.Play*.



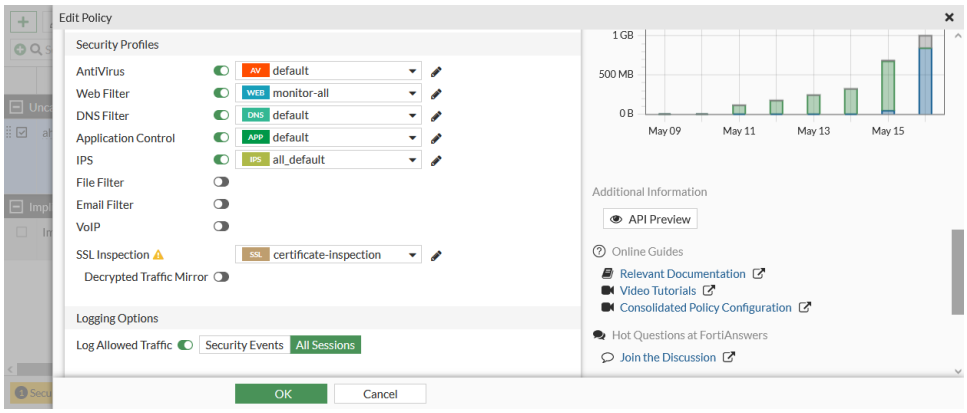
9. Go to *Dashboard* > *FortiView* > *Applications*.
10. In the *Applications* monitor, double-click *YouTube* to view the drilldown information.
11. Click *View session logs* to see all the entries for the videos played. Check the sessions for *YouTube_Video.Play* with the ID *38569*.

Monitoring network traffic without SSL deep inspection

This example describes how to monitor network traffic for YouTube using FortiView cloud application view without SSL deep inspection.

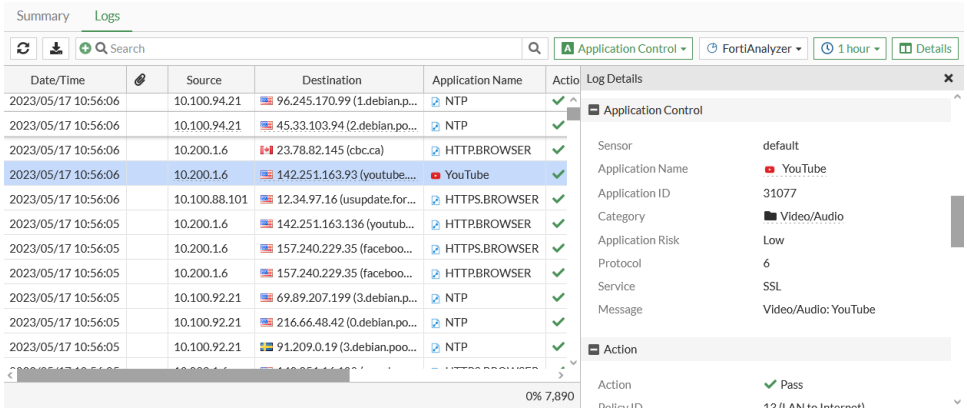
To monitor network traffic without SSL deep inspection:

1. Create a firewall policy with the following settings.
   - *Application Control* is enabled.
   - *SSL Inspection* is set to *certificate-inspection*.
   - *Log Allowed Traffic* is set to *All Sessions*.



2. On the test PC, log into YouTube and play some videos.
3. On the FortiGate, go to *Log & Report* > *Security Events* and look for log entries for browsing and playing YouTube videos in the *Application Control* card.

In this example, the log shows only applications with the name YouTube. The log cannot show YouTube application sensors which rely on SSL deep inspection.

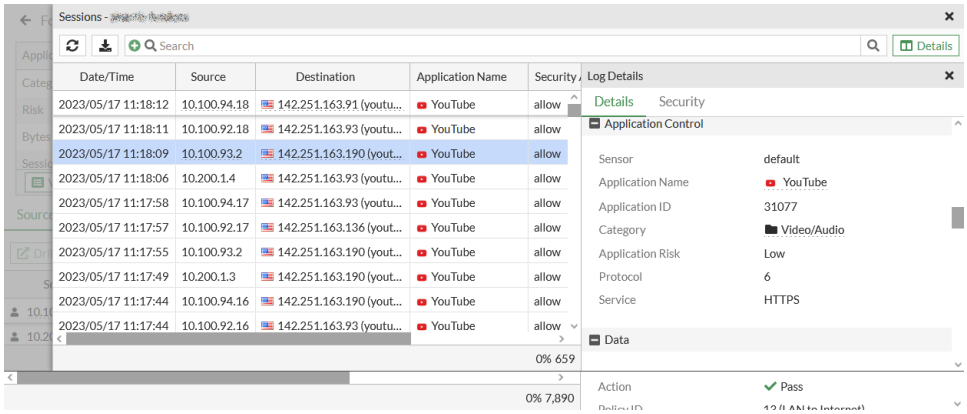


4. Go to *Dashboard* > *FortiView* > *Applications*.

   The *FortiView Application by Bytes* monitor shows the YouTube cloud application without the video played information that requires SSL deep inspection.

5. Double-click *YouTube* and click *View session logs*.

   These sessions were triggered by the application sensor *YouTube* with the ID *31077*. This is the application sensor with cloud behavior which does not rely on SSL deep inspection.



Application risk levels

Application pose different levels of risk to the network, represented by a color code. The application risk levels are shown on various FortiView monitors, such as FortiView Applications, and related GUI pages. See Cloud application view on page 206 for example.

Applications within the same category can have different levels defined by FortiGuard analysts. Considerations may include whether an application is known to be malicious, has known vulnerabilities, or is a trusted enterprise application.

Application risk levels

Indicator	Risk	Description
	Risk Level 1 *Low*	These applications have little to no risk level. This is the default value when no risk is specified. Example categories: Update, Business
	Risk Level 2 *Elevated*	These applications have an elevated risk level. Example categories: Game, Collaboration, Social Media, Video/Audio, VoIP, Industrial, General Interest, Network Service, Web Client, Mobile
	Risk Level 3 *Medium*	These applications have a medium risk level. Example categories: Email, Storage Backup, Cloud-IT
	Risk Level 4 *High*	These applications have a high risk level. Example categories: P2P, Remote Access
	Risk Level 5 *Critical*	These applications have the highest risk level and are most prone to malware or vulnerabilities. Example categories: Botnet, Proxy

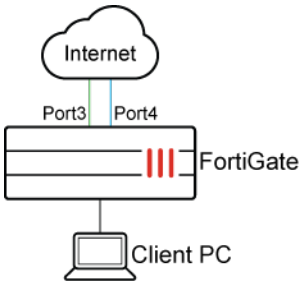
Application performance monitoring analytics NEW

The *FortiView SD-WAN Application Performance* widget leverages the passive WAN application performance monitoring of SD-WAN to log and monitor application performance metrics. See also Application performance monitoring on page 1219. These metrics are aggregated into the following FortiView report types:

- Application performance overview: Overview of different performance metrics of monitored applications.
- Application response time: Shows response time of a given application.
- Application connection stability: Shows application connection stability in terms of jitter.
- Application retransmission: Shows application traffic packet retransmission.
- Application reliability monitor: Shows packet loss and abrupt connection termination by TCP reset.

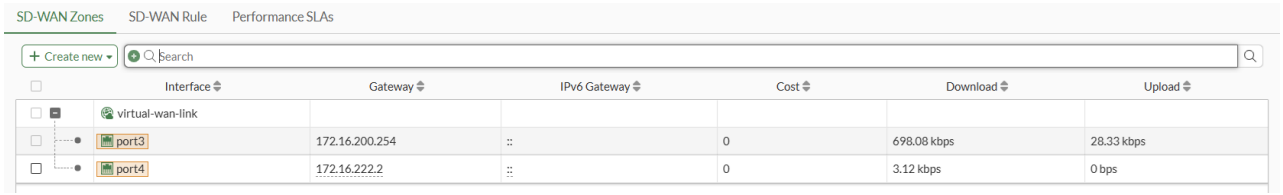
A new GUI option, *Log application health metrics*, is added to the firewall policy logging options section to enable the required logging for this widget.

Example



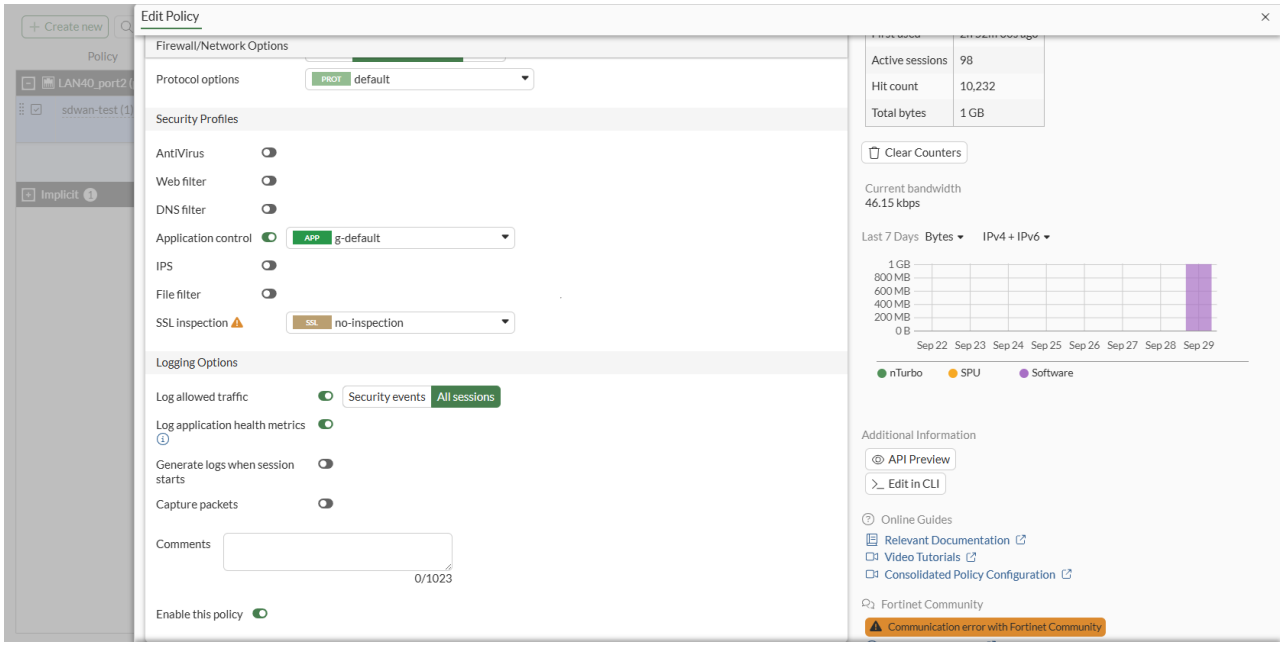
To test the application performance monitoring analytics in the GUI:

1. Create the SD-WAN zone and assign WAN ports.



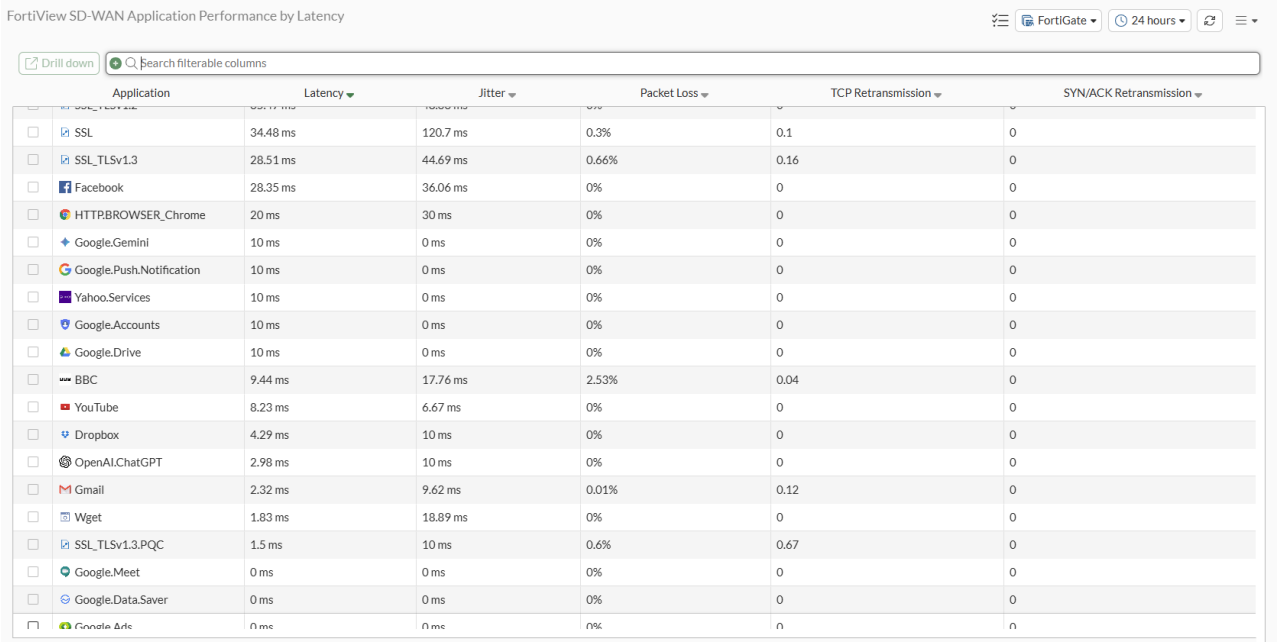
2. Create a policy with the following settings:
   - Set the *SD-WAN zone* as the destination interface.
   - Attach an *Application Control* profile to specify which applications are monitored.
   - Enable *Log application health metrics* under *Logging Options*.

     This replaces the *Passive health check* option under *Firewall > Network Options*. Enabling this toggle enables both *passive-wan-health-measurement* and *app-monitor* values in the policy. When it is disabled, both are turned off.



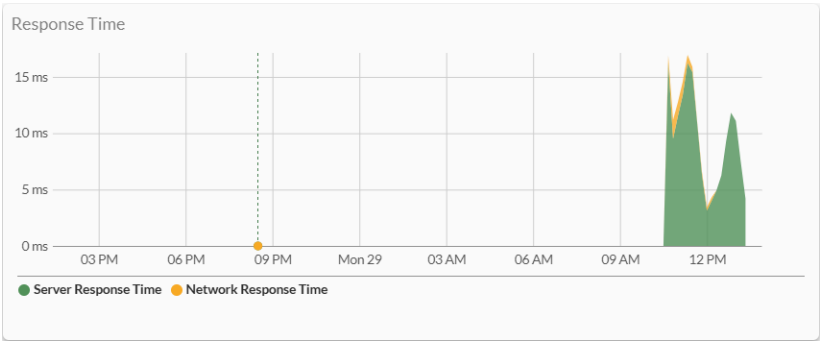
3. From a client PC, generate traffic using various applications, such as YouTube or ChatGPT.
4. Go to *Dashboard > FortiView* and add the *FortiView SD-WAN Application Performance Monitor*. See Editing dashboards of widgets and monitors on page 194.

   Verify that filtering by *Applications* works and check that sorting by *Latency*, *Jitter*, *Packet Loss*, *TCP Retransmission*, and *SYN/ACK Retransmission* is functional.

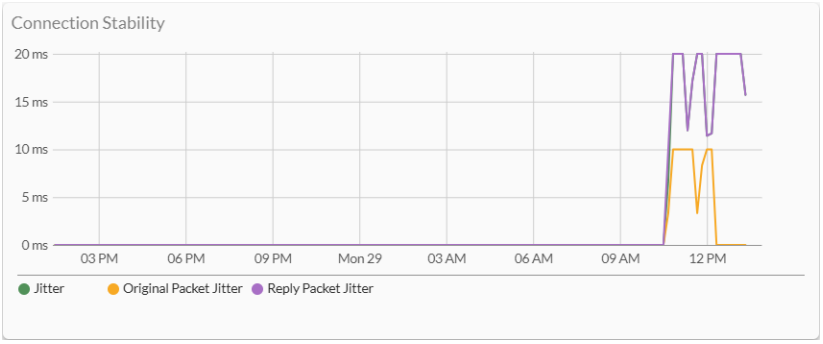


Application	Latency	Jitter	Packet Loss	TCP Retransmission	SYN/ACK Retransmission
SSL	34.48 ms	120.7 ms	0.3%	0.1	0
SSL_TLSv1.3	28.51 ms	44.69 ms	0.66%	0.16	0
Facebook	28.35 ms	36.06 ms	0%	0	0
HTTP.BROWSER_Chrome	20 ms	30 ms	0%	0	0
Google.Gemini	10 ms	0 ms	0%	0	0
Google.Push.Notification	10 ms	0 ms	0%	0	0
Yahoo.Services	10 ms	0 ms	0%	0	0
Google.Accounts	10 ms	0 ms	0%	0	0
Google.Drive	10 ms	0 ms	0%	0	0
BBC	9.44 ms	17.76 ms	2.53%	0.04	0
YouTube	8.23 ms	6.67 ms	0%	0	0
Dropbox	4.29 ms	10 ms	0%	0	0
OpenAI.ChatGPT	2.98 ms	10 ms	0%	0	0
Gmail	2.32 ms	9.62 ms	0.01%	0.12	0
Wget	1.83 ms	18.89 ms	0%	0	0
SSL_TLSv1.3.PQC	1.5 ms	10 ms	0.6%	0.67	0
Google.Meet	0 ms	0 ms	0%	0	0
Google.Data.Saver	0 ms	0 ms	0%	0	0

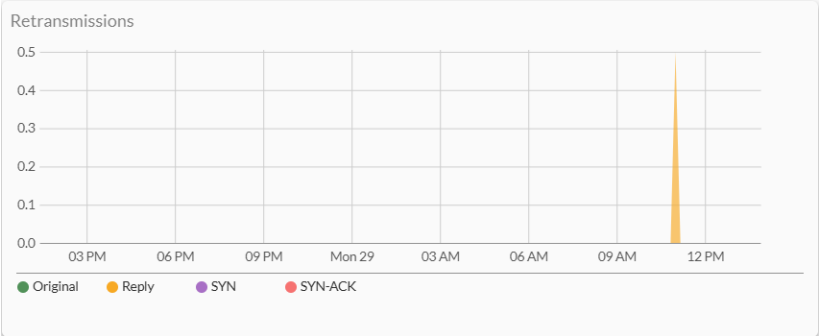
5. Drill down into an entry and verify that the following four charts are available:
   - **Response Time**



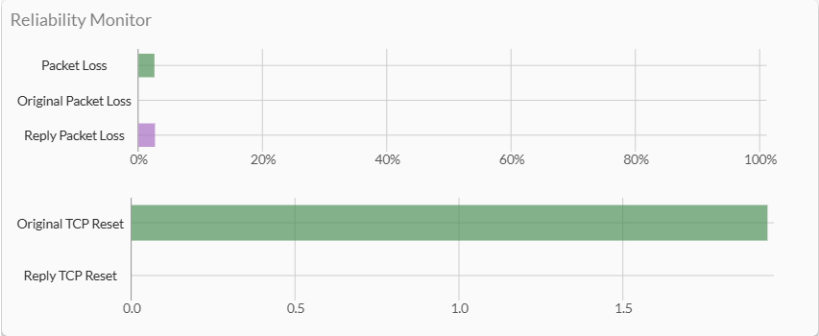
   - **Connection Stability**



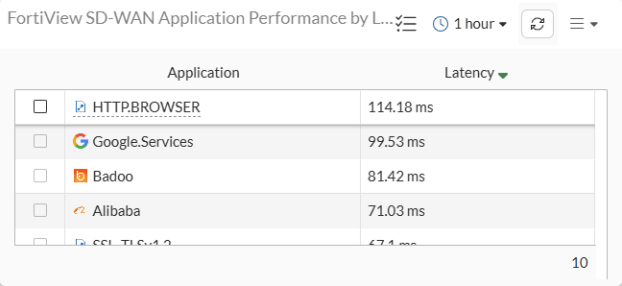
   - **Retransmissions**



- **Reliability Monitor**



6. Go to the Dashboard and confirm that the *FortiView SD-WAN Application Performance* widget can be added.



To test the application performance monitoring analytics in the CLI:

1. Create the SD-WAN zone and assign WAN ports:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "port3"
            set gateway 172.16.200.254
        next
        edit 2
```

```
            set interface "port4"
            set gateway 172.16.222.2
        next
    end
end
```

2. Create a policy:

```
config firewall policy
    edit 1
        set name "sdwan-test"
        set srcintf "port2"
        set dstintf "virtual-wan-link"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set passive-wan-health-measurement enable
        set app-monitor enable
        set utm-status enable
        set application-list "g-default"
        set logtraffic all
        set auto-asic-offload disable
        set nat enable
    next
end
```

3. Configure a health-check:

```
config system sdwan
    config health-check
        edit "test_h"
            set detect-mode passive
            set recoverytime 10
            set embed-measured-health enable
            set members 1 2
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 50
                next
            end
        next
    end
end
```

4. Configure an SD-WAN service and rule:

```
config system sdwan
    config service
        edit 1
```

```
            set name "sdwan_test"
            set mode sla
            set internet-service enable
            set internet-service-name "8X8-8X8.Cloud"
            set internet-service-app-ctrl 31077
            config sla
                edit "test_h"
                    set id 1
                next
            end
            set priority-zone "virtual-wan-link"
            set passive-measurement enable
        next
    end
end
```

5. From a client PC, generate traffic using YouTube.

Network

The following topics provide information about network settings:

- Interfaces on page 222
- DNS on page 358
- Explicit and transparent proxies on page 405
- SD-WAN on page 1037
- DHCP servers and relays on page 550
- Static routing on page 574
- Dynamic routing on page 605
- Multicast on page 742
- FortiExtender on page 766
- Cellular modems on page 769
- Upgrade LTE modem firmware directly from FortiGuard on page 786
- LLDP reception on page 788
- Virtual routing and forwarding on page 791
- NetFlow on page 832
- sFlow on page 867
- Link monitor on page 873
- IPv6 on page 887
- FortiGate LAN extension on page 995
- SCTP packets with zero checksum on the NP7 platform on page 1020
- Diagnostics on page 1025

Interfaces

Physical and virtual interfaces allow traffic to flow between internal networks, and between the internet and internal networks. FortiOS has options for configuring interfaces and groups of sub-networks that can scale as your organization grows. The following table lists commonly used interface types.

Interface type	Description
Physical	A physical interface can be connected to with either Ethernet or optical cables. Depending on the FortiGate model, there is a varying number of Ethernet or optical physical interfaces. Some FortiGates have a grouping of interfaces labeled as *lan* that have a built-in switch functionality. See Physical interface on page 249 for more information.

Interface type	Description
VLAN	A virtual local area network (VLAN) logically divides a local area network (LAN) into distinct broadcast domains using IEEE 802.1Q VLAN tags. A VLAN interface supports VLAN tagging and is associated with a physical interface that can be connected to a device, such as a switch or a router that supports these tags. VLANs can be used on a FortiGate in NAT or transparent mode, and the FortiGate functions differently depending on the operation mode See VLAN on page 252 for more information.
Aggregate	An aggregate interface uses a link aggregation method to combine multiple physical interfaces to increase throughput and to provide redundancy. FortiOS supports a link aggregation (LAG) interface using the Link Aggregation Control Protocol (LACP) based on IEEE 802.3ad/802.1ax. See Aggregation and redundancy on page 267 for more information.
Redundant	A redundant interface combines multiple physical interfaces where traffic only uses one of the interfaces at a time. Its primary purpose is to provide redundancy. This interface is typically used with a fully-meshed HA configuration. See Aggregation and redundancy on page 267 for more information.
Loopback	A loopback interface is a logical interface that is always up because it has no physical link dependency, and the attached subnet is always present in the routing table. It can be accessed through several physical or VLAN interfaces. See Loopback interface on page 276 for more information.
Software switch	A software switch is a virtual switch interface implemented in firmware that allows member interfaces to be added to it. Devices connected to member interfaces communicate on the same subnet, and packets are processed by the FortiGate's CPU. A software switch supports adding a wireless SSID as a member interface. See Software switch on page 278 for more information.
Hardware switch	A hardware switch is a virtual switch interface implemented at the hardware level that allows member interfaces to be added to it. Devices connected to member interfaces communicate on the same subnet. A hardware switch relies on specific hardware to optimize processing and supports the Spanning Tree Protocol (STP). See Hardware switch on page 280 for more information.
Zone	A zone is a logical group containing one or more physical or virtual interfaces. Grouping interfaces in zones can simplify firewall policy configurations. See Zone on page 289 for more information.

Interface type	Description
Virtual wire pair	A virtual wire pair (VWP) is an interface that acts like a virtual wire consisting of two interfaces, with an interface at each of the wire. No IP addressing is configured on a VWP, and communication is restricted between the two interfaces using firewall policies. See Virtual wire pair on page 291 for more information.
FortiExtender WAN extension	A FortiExtender WAN extension is a managed interface that allows a connected FortiExtender to provide WAN connectivity to the FortiGate. See FortiExtender on page 766 for more information.
FortiExtender LAN extension	A FortiExtender LAN extension is a managed interface that allows a connected FortiExtender to provide LAN connectivity to the FortiGate. See FortiExtender on page 766 for more information.
Enhanced MAC VLAN	An enhanced media access control (MAC) VLAN, or EMAC VLAN, interface allows a physical interface to be virtually subdivided into multiple virtual interfaces with different MAC addresses. In FortiOS, the EMAC VLAN functionality acts like a bridge. See Enhanced MAC VLAN on page 301 for more information.
VXLAN	A Virtual Extensible LAN (VXLAN) interface encapsulates layer 2 Ethernet frames within layer 3 IP packets and is used for cloud and data center networks. See VXLAN on page 304 for more information.
Tunnel	A tunnel virtual interface is used for IPsec interface-based or GRE tunnels and are created when configuring IPsec VPN and GRE tunnels, respectively. The tunnel interface can be configured with IP addresses on both sides of the tunnel since this is a requirement when using a tunnel interface with a dynamic routing protocol. See OSPF with IPsec VPN for network redundancy on page 3190, GRE over IPsec on page 2911, and Cisco GRE-over-IPsec VPN on page 2955 for more information. In some edge cases, users attempting to encapsulate an IPsec tunnel interface inside another (IPsec-over-IPsec) may be able to do so in the GUI. However, this is not a supported configuration.
WiFi SSID	A WiFi SSID interface is used to control wireless network user access to a wireless local radio on a FortiWiFi or to a wireless access point using a FortiAP. The SSID is created using the *WiFi & Switch Controller > SSIDs page*, and it appears in the *Network > Interfaces* page once it is created. See Defining a wireless network interface (SSID) in the FortiWiFi and FortiAP Configuration Guide for more information.
VDOM link	A VDOM link allows VDOMs to communicate internally without using additional physical interfaces. See Inter-VDOM routing for more information.

Interface settings

Administrators can configure both physical and virtual FortiGate interfaces in *Network* > *Interfaces*. There are different options for configuring interfaces when FortiGate is in NAT mode or transparent mode.

The available options will vary depending on feature visibility, licensing, device model, and other factors. The following list is not comprehensive.

To configure an interface in the GUI:

1. Go to *Network* > *Interfaces*.
2. Click *Create New* > *Interface*.
3. Configure the interface fields:

Interface Name	Physical interface names cannot be changed.
Alias	Enter an alternate name for a physical interface on the FortiGate unit. This field appears when you edit an existing physical interface. The alias does not appear in logs. The maximum length of the alias is 25 characters.
Type	The configuration type for the interface, such as VLAN, Software Switch, 802.3ad Aggregate, and others.
Interface	This field is available when *Type* is set to *VLAN*. Select the name of the physical interface that you want to add a VLAN interface to. Once created, the VLAN interface is listed below its physical interface in the *Interface* list. You cannot change the physical interface of a VLAN interface.
VLAN ID	This field is available when *Type* is set to *VLAN*. Enter the VLAN ID. The VLAN ID can be any number between 1 and 4094 and must match the VLAN ID added by the IEEE 802.1Q-compliant router or switch that is connected to the VLAN subinterface. The VLAN ID can be edited after the interface is added.
VRF ID	Virtual Routing and Forwarding (VRF) allows multiple routing table instances to coexist on the same router. One or more interface can have a VRF, and packets are only forwarded between interfaces with the same VRF.
Virtual Domain	Select the virtual domain to add the interface to. Only administrator accounts with the *super_admin* profile can change the *Virtual Domain*.
Interface Members	This section can have different formats depending on the *Type*. Members can be selected for some interface types: • *Software Switch* or *Hardware Switch*: Specify the physical and wireless interfaces joined into the switch. • *802.3ad Aggregate* or *Redundant Interface*: This field includes the

	available and selected interface lists.
Role	Set the role setting for the interface. Different settings will be shown or hidden when editing an interface depending on the role: • *LAN*: Used to connected to a local network of endpoints. It is default role for new interfaces. • *WAN*: Used to connected to the internet. When WAN is selected, the *Estimated bandwidth* setting is available, and the following settings are not: *DHCP server*, *Create address object matching subnet*, *Device detection*, *Security mode*, *One-arm sniffer*, *Dedicate to extension/fortiap modes*, and *Admission Control*.and will show Estimated Bandwidth settings. • *DMZ*: Used to connected to the DMZ. When selected, *DHCP server* and *Security mode* are not available. • *Undefined*: The interface has no specific role. When selected, *Create address object matching subnet* is not available.
Estimated bandwidth	The estimated WAN bandwidth. The values can be entered manually, or saved from a speed test executed on the interface. The values can be used in SD-WAN rules that use the Maximize Bandwidth or Best Quality strategy.
Traffic mode	This option is only available when *Type* is *WiFi SSID*. • *Tunnel*: Tunnel to wireless controller • *Bridge*: Local bridge with FortiAP's interface • *Mesh*: Mesh downlink
Address	
Addressing mode	Select the addressing mode for the interface. • *Manual*: Add an IP address and netmask for the interface. If IPv6 configuration is enabled, you can add both an IPv4 and an IPv6 address. • *DHCP*: Get the interface IP address and other network settings from a DHCP server. • *Auto-managed by IPAM*: Assign subnets to prevent duplicate IP addresses from overlapping within the same Security Fabric. See IPAM on page 524. • *PPPoE*: Get the interface IP address and other network settings from a PPPoE server. This option is only available on entry-level FortiGate models. • *One-Arm Sniffer*: Set the interface as a sniffer port so it can be used to detect attacks. See One-arm sniffer on page 234.
IP/Netmask	If *Addressing Mode* is set to *Manual*, enter an IPv4 address and subnet mask for the interface. FortiGate interfaces cannot have multiple IP addresses on the same subnet.
IPv6 addressing mode	Select the addressing mode for the interface: • *Manual*: Add an IP address and netmask for the interface.

	• *DHCP*: Get the interface IP address and other network settings from a DHCP server. • *Delegated*: Select an *IPv6 upstream interface* that has DHCPv6 prefix delegation enabled, and enter an *IPv6 subnet* if needed. The interface will get the IPv6 prefix from the upstream DHCPv6 server that is connected to the IPv6 upstream interface, and form the IPv6 address with the subnet configured on the interface.
IPv6 Address/Prefix	If *Addressing Mode* is set to *Manual* and IPv6 support is enabled, enter an IPv6 address and subnet mask for the interface. A single interface can have an IPv4 address, IPv6 address, or both.
Auto configure IPv6 address	Automatically configure an IPv6 address using Stateless Address Auto-configuration (SLAAC). This option is available when *IPv6 addressing mode* is set to *Manual*.
DHCPv6 prefix delegation	Enable/disable DHCPv6 prefix delegation, which can be used to delegate IPv6 prefixes from an upstream DHCPv6 server to another interface or downstream device. When enabled, there is an option to enable a *DHCPv6 prefix hint* that helps the DHCPv6 server provide the desired prefix.
Create address object matching subnet	This option is available and automatically enabled when *Role* is set to *LAN* or *DMZ*. This creates an address object that matches the interface subnet and dynamically updates the object when the IP/Netmask changes. See Interface subnet on page 2050 for more information.
Secondary IP Address	Add additional IPv4 addresses to this interface.
Administrative Access	
IPv4 Administrative Access	Select the types of administrative access permitted for IPv4 connections to this interface. See Configure administrative access to interfaces on page 229.
IPv6 Administrative Access	Select the types of administrative access permitted for IPv6 connections to this interface. See Configure administrative access to interfaces on page 229.
DHCP Server	Enable a DHCP server for the interface. See DHCP servers and relays on page 550.
Stateless Address Auto-configuration (SLAAC)	Enable to provide IPv6 addresses to connected devices using SLAAC.
DHCPv6 Server	Select to enable a DHCPv6 server for the interface. When enabled, you can configure *DNS service* settings: *Delegated* (delegate the DNS received from the upstream server), *Same as System DNS*, or *Specify* (up to four servers).

	You can also enable *Stateful server*to configure the DHCPv6 server to be stateful. Manually enter the IP range, or use Delegated mode to delegate IP prefixes from an upstream DHCPv6 server connected to the upstream interface.
Network	
Device Detection	Enable/disable passively gathering device identity information about the devices on the network that are connected to this interface.
Security Mode	Enable/disable captive portal authentication for this interface. After enabling captive portal authentication, you can configure the authentication portal, user and group access, custom portal messages, exempt sources and destinations/services, and redirect after captive portal.
DSL Settings	
Physical mode	Set to *ADSL* or *VDSL*.
Transfer mode	Set to *PTM* or *ATM*. If the *Transfer mode* is set to *ATM*, the *Virtual channel identification*, *Virtual path identification*, *ATM protocol*, and *MUX type* can be configured.
Traffic Shaping	
Outbound shaping profile	Enable/disable traffic shaping on the interface. This allows you to enforce bandwidth limits on individual interfaces. See Interface-based traffic shaping profile on page 2159 for more information.
Miscellaneous	
Comments	Enter a description of the interface of up to 255 characters.
Status	Enable/disable the interface. • *Enabled*: The interface is active and can accept network traffic. • *Disabled*: The interface is not active and cannot accept traffic.

4. Click *OK*.

To configure an interface in the CLI:

```
config system interface
    edit <name>
        set vdom <VDOM_name>
        set mode {static | dhcp | pppoe}
        set ip <IP_address/netmask>
        set security-mode {none | captive-portal | 802.1X}
        set egress-shaping-profile <profile>
        set device-identification {enable | disable}
        set allowaccess {ping https ssh http snmp telnet fgfm radius-acct probe-response fabric
ftm}
        set eap-supplicant {enable | disable}
        set eap-method {peap | tls}
```

```
        set eap-identity <identity>
        set eap-password <password>
        set eap-ca-cert <CA_cert>
        set eap-user-cert <user_cert>
        set secondary-IP enable
        config secondaryip
            edit 1
                set ip 9.1.1.2 255.255.255.0
                set allowaccess ping https ssh snmp http
            next
        end
    next
end
```

Configure administrative access to interfaces

You can configure the protocols that administrators can use to access interfaces on the FortiGate. This helps secure access to the FortiGate by restricting access to a limited number of protocols. It helps prevent users from accessing interfaces that you don't want them to access, such as public-facing ports.

As a best practice, you should configure administrative access when you're setting the IP address for a port.

To configure administrative access to interfaces in the GUI:

1. Go to *Network* > *Interfaces*.
2. Create or edit an interface.
3. In the *Administrative Access* section, select which protocols to enable for *IPv4* and *IPv6 Administrative Access*.

Industrial Connectivity	Allow Industrial Connectivity service access to proxy traffic between serial port and TCP/IP. Available with FortiGate Rugged models equipped with a serial RS-232 (DB9/RJ45) interface and when *Role* is set to *Undefined* or *WAN*. See Industrial Connectivity on page 1021.
HTTPS	Allow secure HTTPS connections to the FortiGate GUI through this interface. If configured, this option is enabled automatically.
HTTP	Allow HTTP connections to the FortiGate GUI through this interface. This option can only be enabled if HTTPS is already enabled.
PING	The interface responds to pings. Use this setting to verify your installation and for testing.
FMG-Access	Allow FortiManager authorization automatically during the communication exchanges between FortiManager and FortiGate devices.
SSH	Allow SSH connections to the CLI through this interface.
SNMP	Allow a remote SNMP manager to request SNMP information by connecting to this interface.

FTM	Allow FortiToken Mobile Push (FTM) access.
RADIUS Accounting	Allow RADIUS accounting information on this interface.
Probe Response	Allow probe responses. This option can only be enabled in the CLI.
Security Fabric Connection	Allow Security Fabric access. This enables FortiTelemetry and CAPWAP.

Speed Test	Allow this interface to listen to speed test sender requests.
	To allow the FortiGate to be configured as speed test server, configure the following:
	`config system global`
	`  set speedtest-server {enable | disable}`
	`end`
	For more detail, see Running speed tests from the hub to the spokes in dial-up IPsec tunnels on page 1241.

SCIM	Allow SCIM access.

Media type for interfaces that support transceiver modules

Some FortiGate models have interfaces that support a wide range of transceiver modules. These modules can generally be categorized as:

- Short-range (SR) - designed to transmit data over a short distance (up to 300 m) over an optical cable.
- Long-Range (LR) - designed to transmit data over long distances (up to 10 km) over an optical cable.
- Copper (CR) - designed to transmit data over various distances over a copper cable.

When a transceiver module is inserted into the FortiGate interface, the proper media type needs to be configured in the interface settings, based on the type of transceiver:

```
config system interface
    edit <interface>
        set mediatype {sr | lr | cr}
    next
end
```

The actual media types may differ between models and/or interfaces on the same model. For instance, some FortiGate models support sr4, lr4, and cr4, representing a 4-lane transceiver module.

In any case, select the media type that matches the module. Do not use, for example, the `cr` type for an optical transceiver or `sr`/`lr` for a copper one.

Additionally, the configured speed must be compatible with the chosen media type. Any mismatch between media type and speed will result in link failures or prevent the interface from coming up. FortiGate typically enforces these constraints to avoid misconfiguration, but it is best practice to verify compatibility before applying settings.

FEC implementations on 10G, 25G, 40G, and 100G interfaces

Only supported FEC (forward error correction) implementations are allowed to be configured on 10G, 25G, 40G, and 100G interfaces based on the speed that is selected.

- For 1000M, 10G, or 40G interfaces, FEC is not supported and the option is disabled.
- For 25G and 100G interfaces, FEC is automatically set to `cl91-rs-fec` by default.

To configure an interface for FEC:

```
config system interface
    edit <name>
        set speed {10000full | 1000full | 100Gauto | 100Gfull | 25000auto | 25000full | 40000full}
        set mediatype {sr4 | lr4 | cr4}
        set forward-error-correction {disable | cl91-rs-fec | cl74-fc-fec}
    next
end
```

Option	Description
`speed {10000full | 1000full | 100Gauto | 100Gfull | 25000auto | 25000full | 40000full}`	Set the interface speed: • 10000full: 10G full-duplex • 1000full: 1000M full-duplex • 100Gauto: 100G auto-negotiation • 100Gfull: 100G full-duplex • 25000auto: 25G auto-negotiation • 25000full: 25G full-duplex • 40000full: 40G full-duplex
`mediatype {sr4 | lr4 | cr4}`	Set the media type to use: • sr4: short-range transceiver (4-lane) • lr4: long-range transceiver (4-lane) • cr4: copper transceiver (4-lane)
`forward-error-correction {disable | cl91-rs-fec | cl74-fc-fec}`	Set the forward error correction type: • disable: disable forward error correction • cl91-rs-fec: Reed-Solomon (FEC CL91) • cl74-fc-fec: Firecode (FEC CL74)

To change the interface speed from 40G to 100G:

```
config system interface
    edit port26
        set speed 100Gfull
    next
end

The speed/mediatype/FEC of port26 will be changed from 40000full/sr4/disable to 100Gfull/sr4/cl91-
```

```
rs-fec.
Do you want to continue? (y/n) y
```

Since the speed changed to 100G, the `mediatype` setting automatically changes to `sr4`, and the `forward-error-correction` setting automatically changes to `cl91-rs-fec`. When the speed was 40G, the `forward-error-correction` setting was disabled.

Interface MTU packet size

Changing the maximum transmission unit (MTU) on FortiGate interfaces changes the size of transmitted packets. Most FortiGate device's physical interfaces support jumbo frames that are up to 9216 bytes, but some only support 9000 or 9204 bytes.

To avoid fragmentation, the MTU should be the same as the smallest MTU in all of the networks between the FortiGate and the destination. If the packets sent by the FortiGate are larger than the smallest MTU, then they are fragmented, slowing down the transmission. Packets with the DF flag set in the IPv4 header are dropped and not fragmented .

On many network and endpoint devices, the path MTU is used to determine the smallest MTU and to transmit packets within that size.

- ASIC accelerated FortiGate interfaces, such as NP6, NP7, SOC4 (np6xlite), and SOC5, support MTU sizes up to 9216 bytes.
- FortiGate VMs can have varying maximum MTU sizes, depending on the underlying interface and driver.
- Virtual interfaces, such as VLAN interfaces, inherit their MTU size from their parent interface.

To verify the supported MTU size:

```
config system interface
    edit <interface>
        set mtu-override enable
        set mtu <integer>
    next
end
```

To change the MTU size:

```
config system interface
    edit <interface>
        set mtu-override enable
        set mtu <max bytes>
    next
end
```

Maximum MTU size on a path

To manually test the maximum MTU size on a path, you can use the ping command on a Windows computer.

For example, you can send ICMP packets of a specific size with a DF flag, and iterate through increasing sizes until the ping fails.

- The `-f` option specifies the Do not Fragment (DF) flag.
- The `-l` option specifies the length, in bytes, of the Data field in the echo Request messages. This does not include the 8 bytes for the ICMP header and 20 bytes for the IP header. Therefore, if the maximum MTU is 1500 bytes, then the maximum supported data size is: 1500 - 8 - 20 = 1472 bytes.

To determine the maximum MTU size on a path:

1. In Windows command prompt, try a likely MTU size:

```
>ping 4.2.2.1 -l 1472 -f
Pinging 4.2.2.1 with 1472 bytes of data:
Reply from 4.2.2.1: bytes=1472 time=41ms TTL=52
Reply from 4.2.2.1: bytes=1472 time=42ms TTL=52
Reply from 4.2.2.1: bytes=1472 time=103ms TTL=52
Reply from 4.2.2.1: bytes=1472 time=38ms TTL=52

Ping statistics for 4.2.2.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 38ms, Maximum = 103ms, Average = 56ms
```

2. Increase the size and try the ping again:

```
>ping 4.2.2.1 -l 1473 -f
Pinging 4.2.2.1 with 1473 bytes of data:
Request timed out.

Ping statistics for 4.2.2.1:
    Packets: Sent = 1, Received = 0, Lost = 1 (100% loss),
```

The second test fails, so the maximum MTU size on the path is 1472 bytes + 8-byte ICMP header + 20-byte IP header = 1500 bytes

Maximum segment size

The TCP maximum segment size (MSS) is the maximum amount of data that can be sent in a TCP segment. The MSS is the MTU size of the interface minus the 20 byte IP header and 20 byte TCP header. By reducing the TCP MSS, you can effectively reduce the MTU size of the packet.

The TCP MSS can be configured in a firewall policy (see Configurations in the CLI on page 1853), or directly on an interface.

To configure the MSS on an interface:

```
config system interface
    edit "wan2"
        set vdom "root"
        set mode dhcp
        set allowaccess ping fgfm
        set type physical
        set tcp-mss 1448
        set role wan
```

```
    next
end
```

One-arm sniffer

You can use a one-arm sniffer to configure a physical interface as a one-arm intrusion detection system (IDS). Traffic sent to the interface is examined for matches to the configured security profile. The matches are logged, and then all received traffic is dropped. Sniffing only reports on attacks; it does not deny or influence traffic.

You can also use the one-arm sniffer to configure the FortiGate to operate as an IDS appliance to sniff network traffic for attacks without actually processing the packets. To configure a one-arm IDS, enable sniffer mode on a physical interface and connect the interface to the SPAN port of a switch or a dedicated network tab that can replicate the traffic to the FortiGate.

If the one-arm sniffer option is not available, this means the interface is in use. Ensure that the interface is not selected in any firewall policies, routes, virtual IPs, or other features where a physical interface is specified. The option also does not appear if the role is set to WAN. Ensure the role is set to LAN, DMZ, or undefined.

One-arm sniffer supports monitoring mirrored traffic that includes VLAN-tagged, VXLAN, and GRE encapsulated packets.

The following table lists some of the one-arm sniffer settings you can configure:

Field	Description
Security Profiles	The following profiles are configurable in the GUI and CLI: • Antivirus • Web filter • Application control • IPS • File filter The following profiles are only configurable in the CLI: • Email filter • DLP • IPS DoS

Each security profile has a predefined profile for *One-Arm Sniffer* called *sniffer-profile*. The *sniffer-profile* can be viewed or edited from the GUI through the *Edit Interface* page only. Please refer to the Example configuration on page 235 for a demonstration.

The sniffer traffic log is generated by the IPS engine. If no UTM profile is enabled in the sniffer policy, then the IPS engine is not running and no sniffer traffic log is generated. If a UTM profile is enabled in the sniffer policy:

- When *Log allowed traffic* is set to *Security events* (`utm` in the CLI), only security events are logged in the sniffer traffic log.
- When *Log allowed traffic* is set to *All sessions* (`all` in the CLI), clean traffic and security events are logged in sniffer traffic log.

CPU usage and packet loss

Traffic scanned on the one-arm sniffer interface is processed by the CPU, even if there is an SPU, such as NPU or CP, present. The one-arm sniffer may cause higher CPU usage and perform at a lower level than traditional inline scanning, which uses NTurbo or CP to accelerate traffic when present.

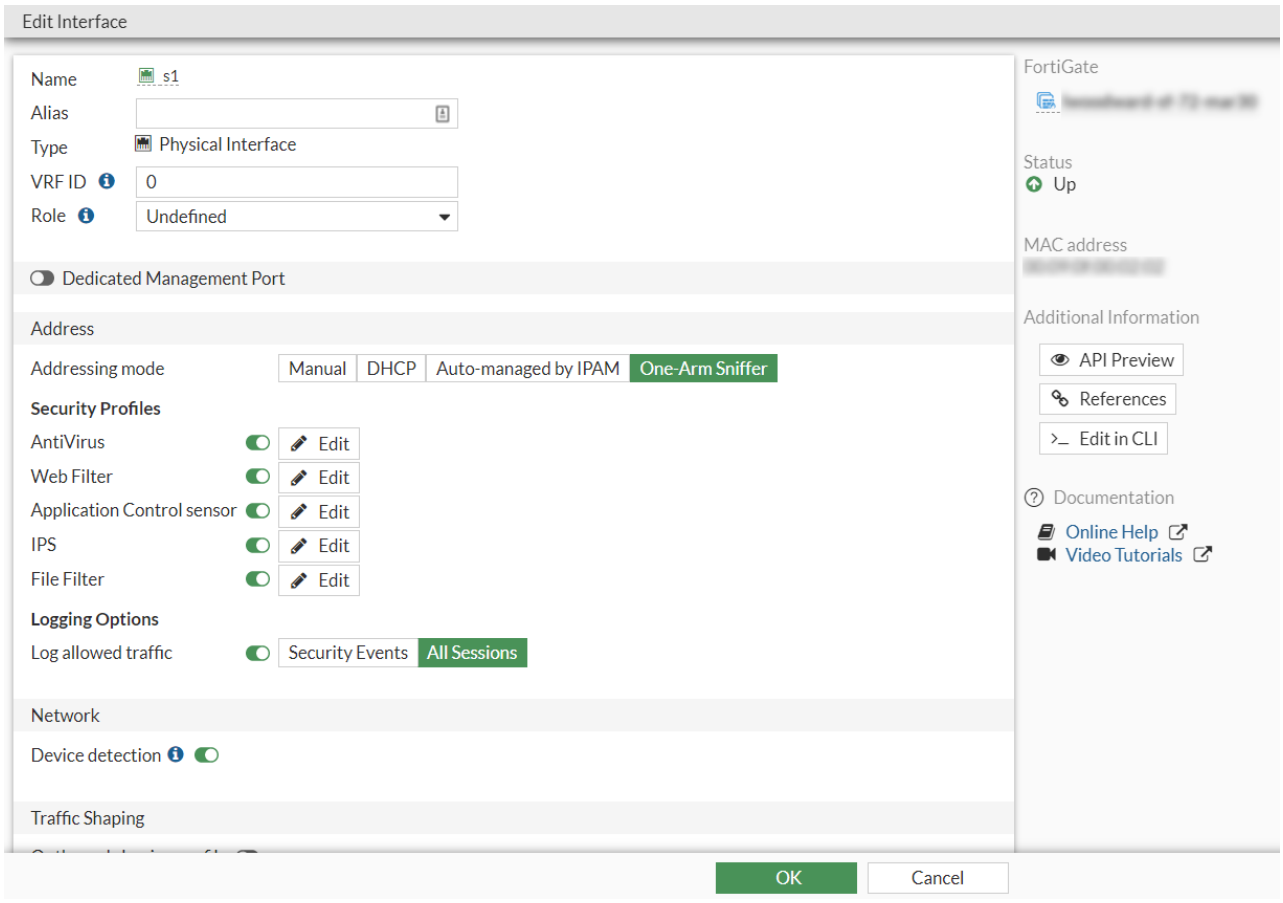
The absence of high CPU usage does not indicate the absence of packet loss. Packet loss may occur due to the capacity of the TAP devices hitting maximum traffic volume during mirroring, or on the FortiGate when the kernel buffer size is exceeded and it is unable to handle bursts of traffic.

Example configuration

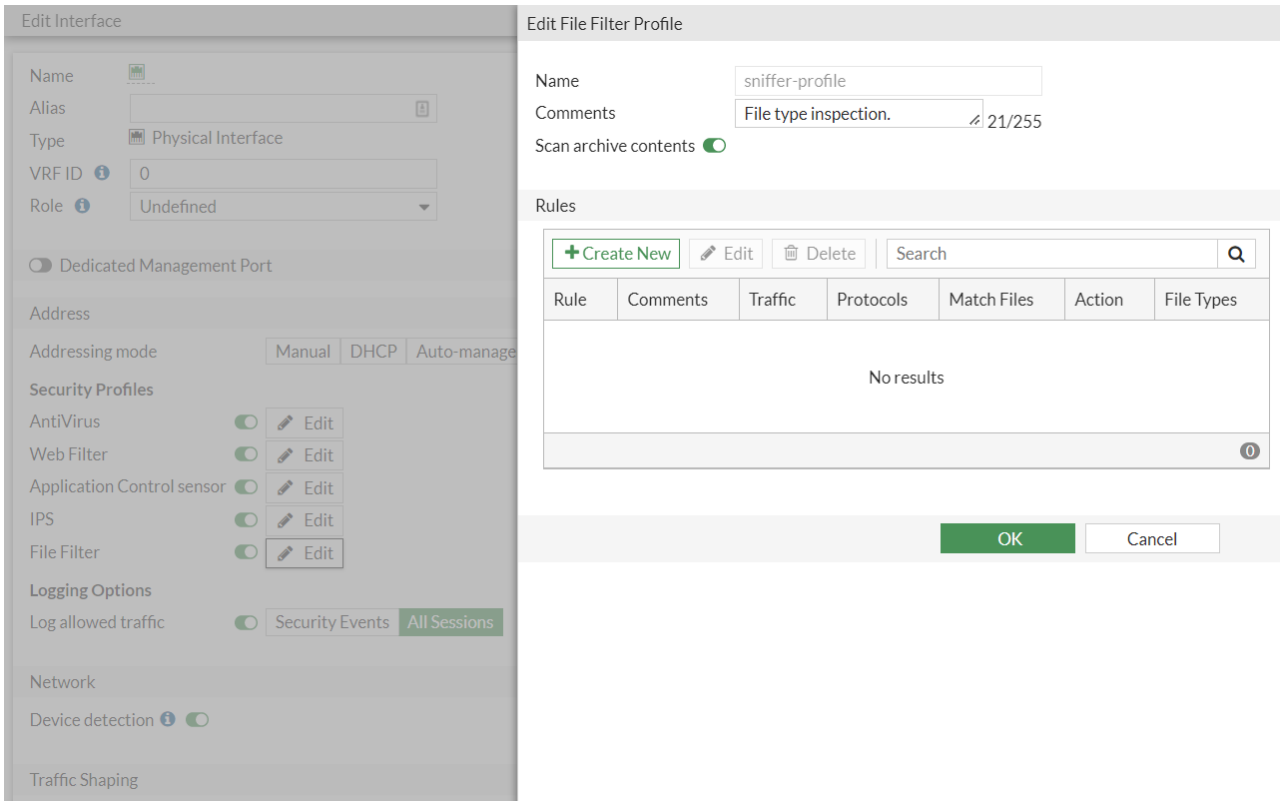
The following example shows how to configure a file filter profile that blocks PDF and RAR files used in a one-arm sniffer policy.

To configure a one-arm sniffer policy in the GUI:

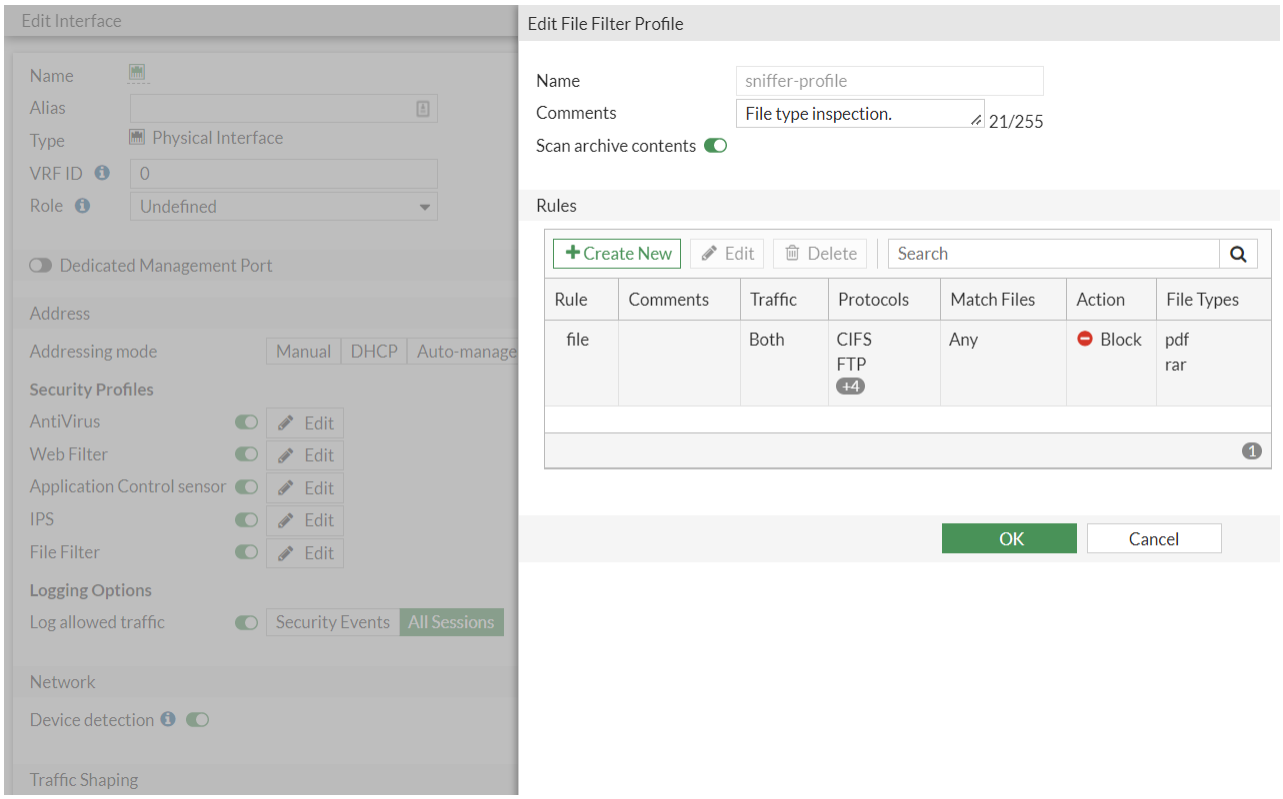
1. Go to *Network* > *Interfaces* and double-click a physical interface to edit it.
2. For *Role*, select either *LAN*, *DMZ*, or *Undefined*.
3. For *Addressing Mode*, select *One-Arm Sniffer*.



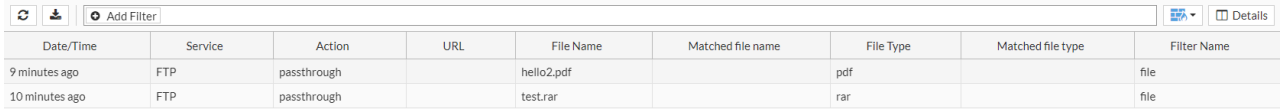
4. In the *Security Profiles* section, enable *File Filter* and click *Edit*. The *Edit File Filter Profile* pane opens.
5. In the *Rules* table, click *Create New*.



6. Configure the rule:
   a. For *File types*, click the + and select *pdf* and *rar*.
   b. For *Action*, select *Block*.
   c. Click *OK* to save the rule.
7. Click *OK* to save the file filter profile.



8. Click *OK* to save the interface settings.
9. Go to *Log & Report* > *Security Events* to view the *File Filter* logs.



To configure a one-arm sniffer policy in the CLI:

1. Configure the interface:

```
config system interface
    edit "s1"
        set vdom "root"
        set ips-sniffer-mode enable
        set type physical
        set role undefined
        set snmp-index 31
    next
end
```

2. Configure the file filter profile:

```
config file-filter profile
    edit "sniffer-profile"
        set comment "File type inspection."
        config rules
```

```
            edit "1"
                set protocol http ftp smtp imap pop3 cifs
                set action block
                set file-type "pdf" "rar"
            next
        end
    next
end
```

3. Configure the firewall sniffer policy:

```
config firewall sniffer
    edit 1
        set interface "s1"
        set file-filter-profile-status enable
        set file-filter-profile "sniffer-profile"
    next
end
```

4. View the log:

```
# execute log filter category  19
# execute log display
1 logs found.
1 logs returned.

1: date=2020-12-29 time=09:14:46 eventtime=1609262086871379250 tz="-0800" logid="1900064000"
type="utm" subtype="file-filter" eventtype="file-filter" level="warning" vd="root" policyid=1
sessionid=792 srcip=172.16.200.55 srcport=20 srcintf="s1" srcintfrole="undefined"
dstip=10.1.100.11 dstport=56745 dstintf="s1" dstintfrole="undefined" proto=6 service="FTP"
profile="sniffer-profile" direction="outgoing" action="blocked" rulename="1"
filename="hello.pdf" filesize=9539 filetype="pdf" msg="File was blocked by file filter."
```

Interface migration wizard

The *Integrate Interface* option on the *Network > Interfaces* page helps migrate a physical port into another interface or interface type such as aggregate, software switch, redundant, zone, or SD-WAN zone. The FortiGate will migrate object references either by replacing the existing instance with the new interface, or deleting the existing instance based on the user's choice. Users can also change the VLAN ID of existing VLAN sub-interface or FortiSwitch VLANs.

The interface migration wizard does not support turning an aggregate, software switch, redundant, zone, or SD-WAN zone interface back into a physical interface.

Integrating an interface

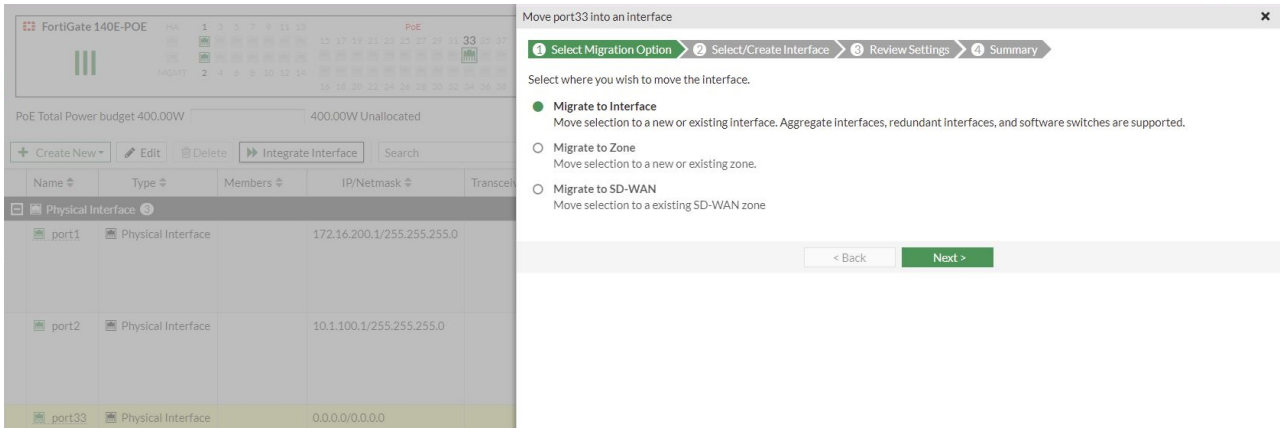
In this example, a DHCP server interface is integrated into a newly created redundant interface, which transfers the DHCP server to a redundant interface.

To integrate an interface:

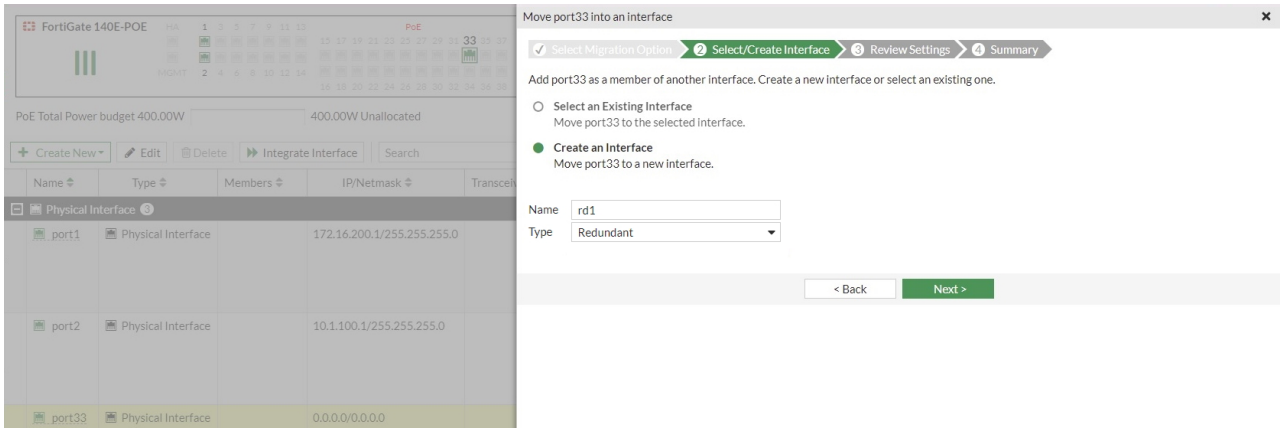
1. Go to *Network* > *Interfaces* and select an interface in the list.
2. Click *Integrate Interface*. The wizard opens.

   Alternatively, select an interface in the list. Then right-click and select *Integrate Interface*.

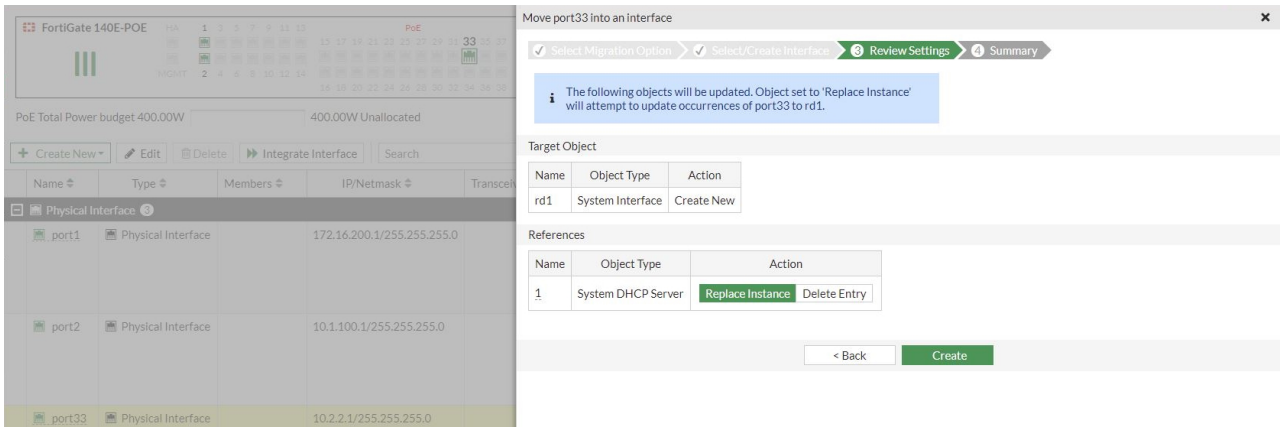
3. Select *Migrate to Interface* and click *Next*.



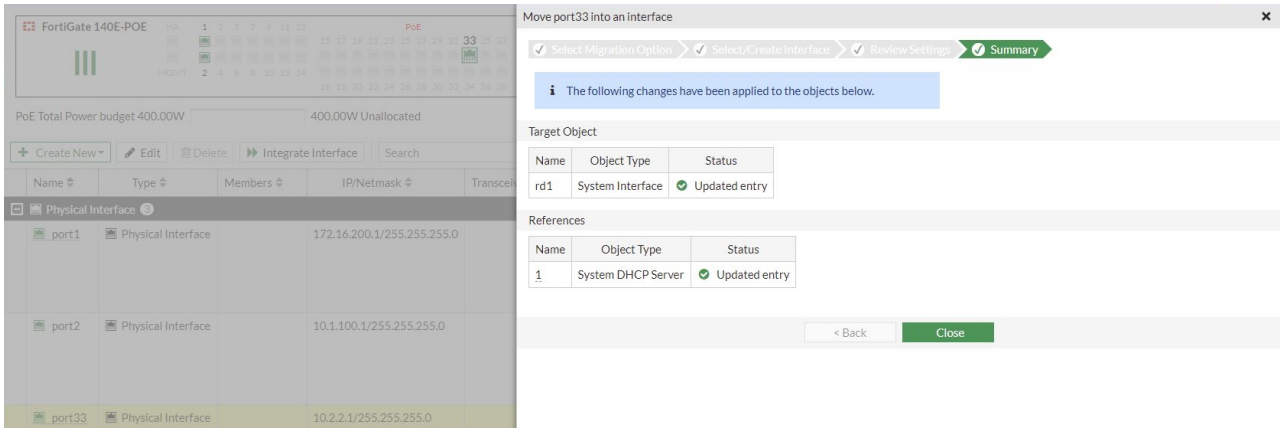
4. Select *Create an Interface*. Enter a name (*rd1*) and set the *Type* to *Redundant*.



5. Click *Next*. The *References* sections lists the associated services with options to *Replace Instance* or *Delete Entry*.
6. For the DHCP server *Action*, select *Replace Instance* and click *Create*.



7. The migration occurs automatically and the statuses for the object and reference change to *Updated entry*. Click *Close*.

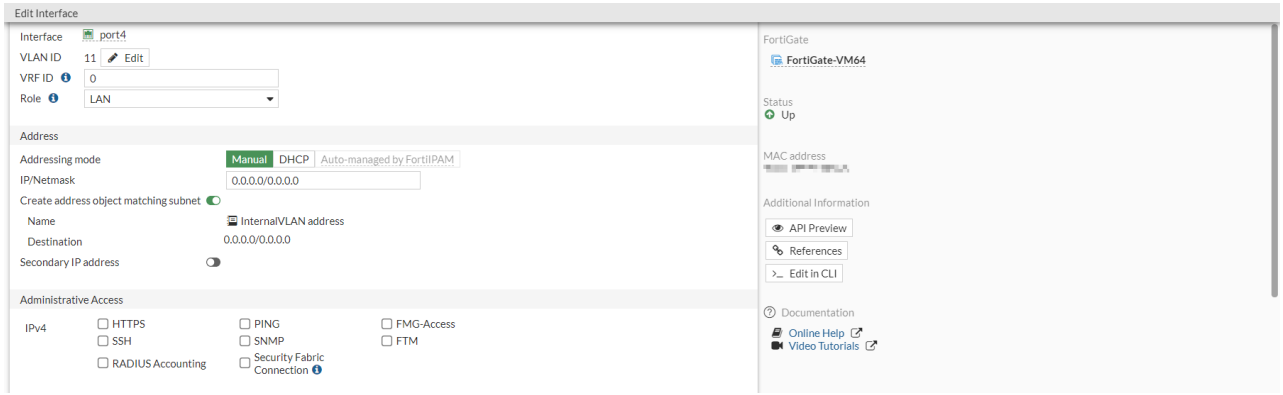


Changing the VLAN ID

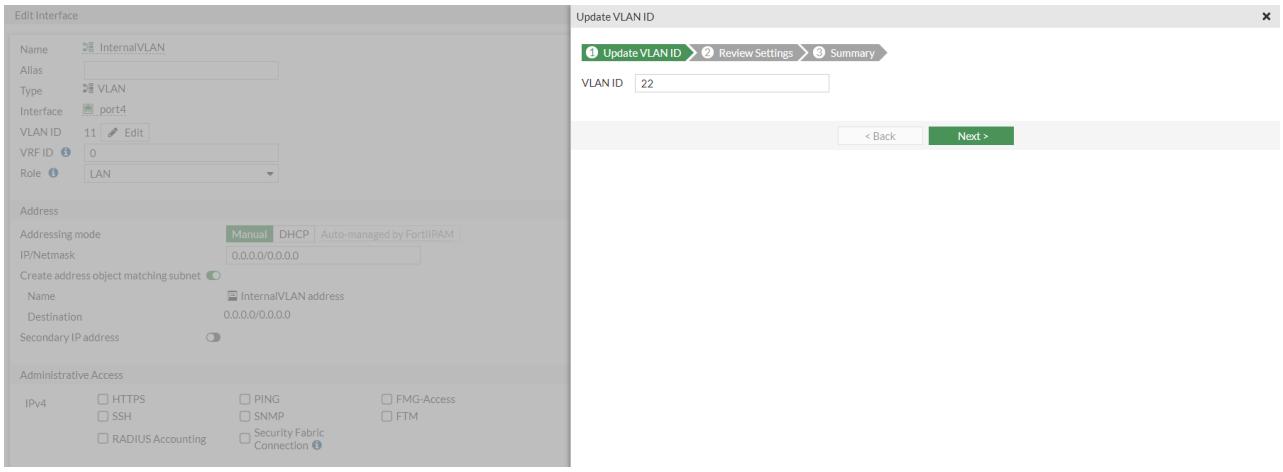
In this example, the VLAN ID of *InternalVLAN* is changed from 11 to 22.

To change the VLAN ID:

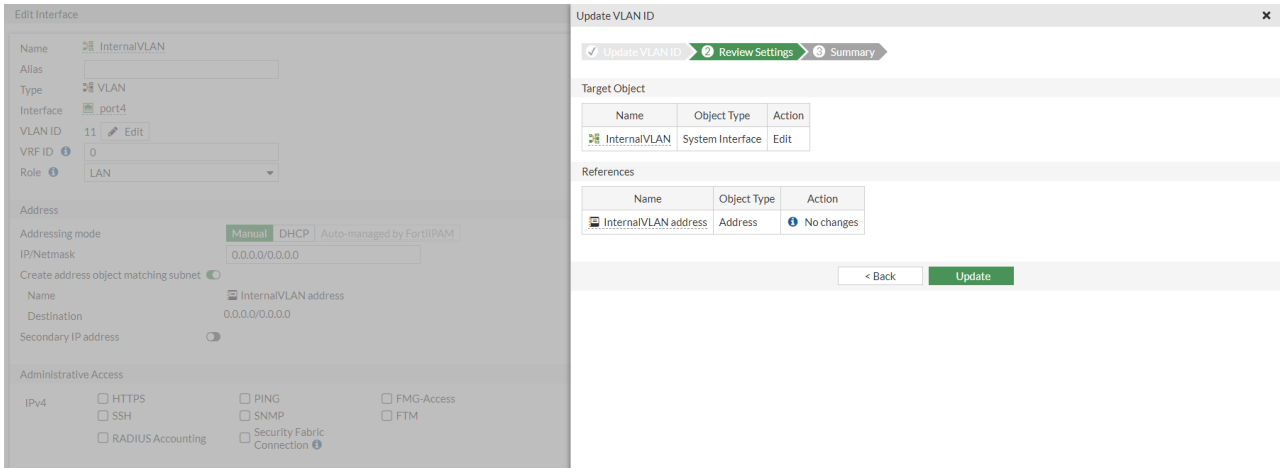
1. Go to *Network* > *Interfaces* and edit an existing interface.
2. Beside the *VLAN ID* field, click *Edit*. The *Update VLAN ID* window opens.



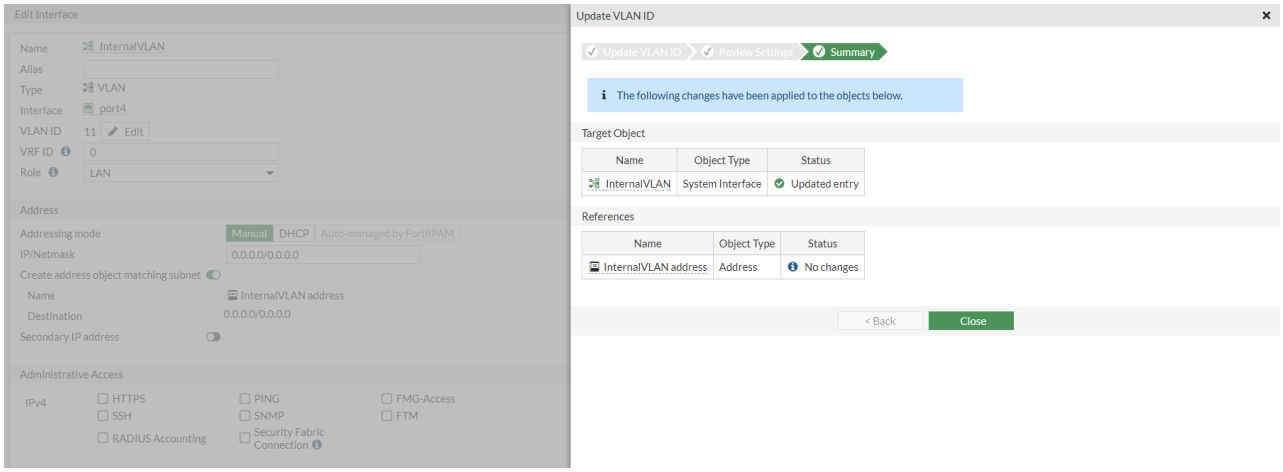
3. Enter the new ID (*22*) and click *Next*.



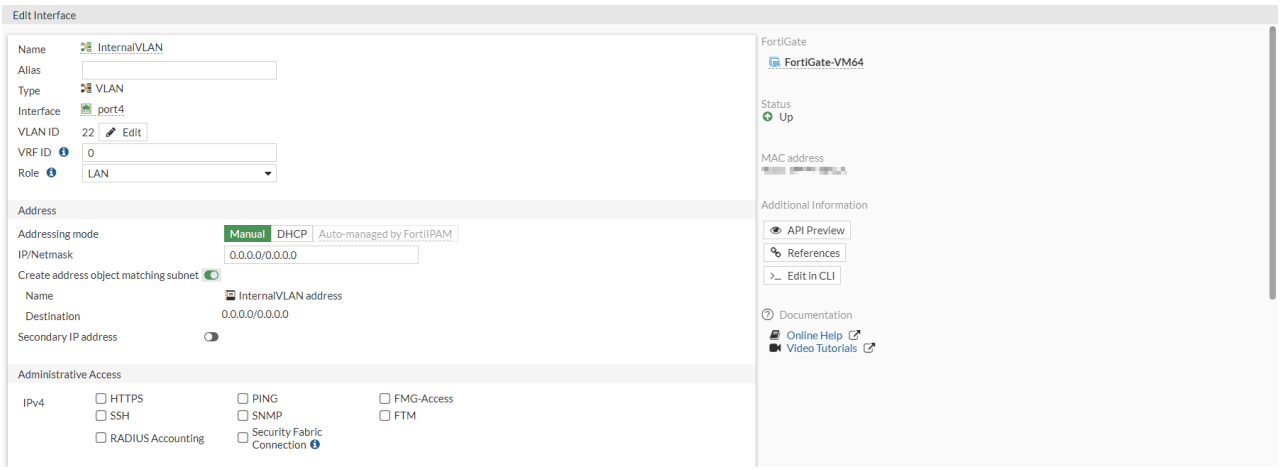
4. Verify the changes, then click *Update* and *OK*.



5. The target object status changes to *Updated entry*. Click *Close*.



In the interface settings, the ID displays as *22*.



Captive portals

A captive portal is used to enforce authentication before web resources can be accessed. Until a user authenticates successfully, any HTTP request returns the authentication page. After successfully authenticating, a user can access the requested URL and other web resources, as permitted by policies. The captive portal can also be configured to only allow access to members of specific user groups.

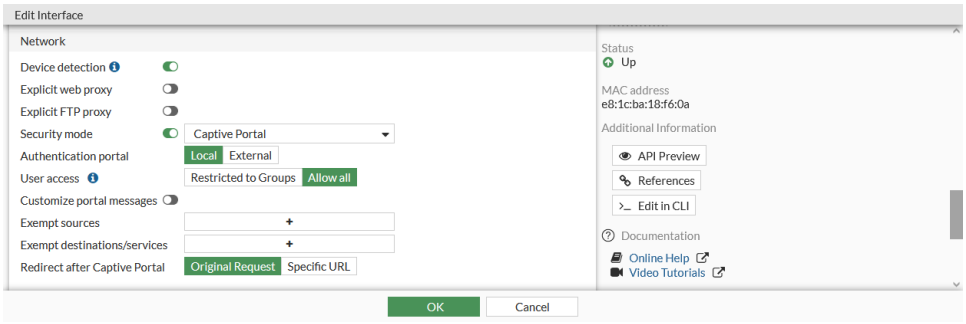
Captive portals can be hosted on the FortiGate or an external authentication server. They can be configured on any network interface, including VLAN and WiFi interfaces. On a WiFi interface, the access point appears open, and the client can connect to access point with no security credentials, but then sees the captive portal authentication page. See Captive Portal Security, in the FortiWiFi and FortiAP Configuration Guide for more information.

All users on the interface are required to authenticate. Exemption lists can be created for devices that are unable to authenticate, such as a printer that requires access to the internet for firmware upgrades.

When configuring a bridge mode SSID, you do not need to enable captive portal.

To configure a captive portal in the GUI:

1. Go to *Network* > *Interfaces* and edit the interface that the users connect to. The interface *Role* must be *LAN* or *Undefined*.
2. Enable *Security mode*.



3. Configure the following settings, then click *OK*.

Authentication Portal	Configure the location of the portal: • *Local*: the portal is hosted on the FortiGate unit. • *External*: enter the FQDN or IP address of external portal.
User access	Select if the portal applies to all users, or selected user groups: • *Restricted to Groups*: restrict access to the selected user groups. The *Login page* is shown when a user tries to log in to the captive portal. • *Allow all*: all users can log in, but access will be defined by relevant policies. The *Disclaimer page* is shown when a user tried to log in to the captive portal.
Customize portal messages	Enable to use custom portal pages, then select a replacement message group. See Custom captive portal pages on page 245.
Exempt sources	Select sources that are exempt from the captive portal. Each exemption is added as a rule in an automatically generated exemption list.
Exempt destinations/services	Select destinations and services that are exempt from the captive portal. Each exemption is added as a rule in an automatically generated exemption list.
Redirect after Captive Portal	Configure website redirection after successful captive portal authentication: • *Original Request*: redirect to the initially browsed to URL . • *Specific URL*: redirect to the specified URL. The *Redirect after Captive Portal* settings are ignored when the `CONTINUE_URL = "<web site>"` parameter is included in the captive portal request.

To configure a captive portal in the CLI:

1. If required, create a security exemption list:

```
config user security-exempt-list
    edit <list>
        config rule
            edit 1
                set srcaddr <source(s)>
                set dstaddr <source(s)>
                set service <service(s)>
            next
            edit 2
                set srcaddr <source(s)>
                set dstaddr <source(s)>
                set service <service(s)>
            next
        end
    next
end
```

2. Configure captive portal authentication on the interface:

```
config system interface
    edit <interface>
        set security-mode {none | captive-portal}
        set security-external-web <string>
        set replacemsg-override-group <group>
        set security-redirect-url <string>
        set security-exempt-list <list>
        set security-groups <group(s)>
    next
end
```

Custom captive portal pages

Portal pages are HTML files that can be customized to meet user requirements.

Most of the text and some of the HTML in the message can be changed. Tags are enclosed by double percent signs (%%); most of them should not be changed because they might carry information that the FortiGate unit needs. For information about customizing replacement messages, see Modifying replacement messages on page 3900.

The images on the pages can be replaced. For example, your organization's logo can replace the Fortinet logo. For information about uploading and using new images in replacement messages, see Replacement message images on page 3901.

The following pages are used by captive portals:

Login Page	Requests user credentials. The *%%QUESTION%%* tag provides the *Please enter the required information to continue*. text. This page is shown to users that are trying to log in when *User access* is set to *Restricted to Groups*.
Login Failed Page	Reports that incorrect credentials were entered, and requests correct credentials. The *%%FAILED_MESSAGE%%* tag provides the *Firewall authentication failed. Please try again*. text.
Disclaimer Page	A statement of the legal responsibilities of the user and the host organization that the user must agree to before proceeding. This page is shown users that are trying to log in when *User access* is set to *Allow all*.
Declined Disclaimer Page	Shown if the user does not agree to the statement on the Disclaimer page. Access is denied until the user agrees to the disclaimer.

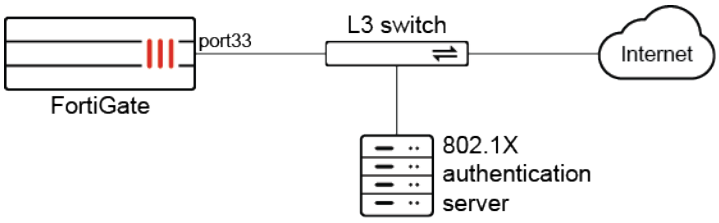
Configuring a FortiGate interface to act as an 802.1X supplicant

A FortiGate interface can be configured to act as a 802.1X supplicant. The settings can be enabled on the network interface in the CLI. The EAP authentication method can be either PEAP or TLS using a user certificate.

```
config system interface
    edit <interface>
        set eap-supplicant {enable | disable}
        set eap-method {peap | tls}
        set eap-identity <identity>
        set eap-password <password>
        set eap-ca-cert <CA_cert>
        set eap-user-cert <user_cert>
    next
end
```

Example

In this example, the FortiGate connects to an L3 switch that is not physically secured. All devices that connect to the internet through the L3 switch must be authenticated with 802.1X on the switch port by either a username and password (PEAP), or a user certificate (TLS). Configuration examples for both EAP authentication methods on port33 are shown.



To configure EAP authentication with PEAP:

1. Configure the interface:

```
config system interface
    edit "port33"
        set vdom "vdom1"
        set ip 7.7.7.2 255.255.255.0
        set allowaccess ping https ssh snmp http telnet fgfm radius-acct probe-response fabric
        set stpforward enable
        set type physical
        set snmp-index 42
        set eap-supplicant enable
        set eap-method peap
        set eap-identity "test1"
        set eap-password **********
    next
end
```

2. Verify the interface's PEAP authentication details:

```
# diagnose test app eap_supp 2
Interface: port33
status:Authorized
method: PEAP
identity: test1
```

```
ca_cert:
client_cert:
private_key:
last_eapol_src =70:4c:a5:3b:0b:c6
```

Traffic is able to pass because the status is authorized.

To configure EAP authentication with TLS:

1. Configure the interface:

```
config system interface
    edit "port33"
        set vdom "vdom1"
        set ip 7.7.7.2 255.255.255.0
        set allowaccess ping https ssh snmp http telnet fgfm radius-acct probe-response fabric
        set stpforward enable
        set type physical
        set snmp-index 42
        set eap-supplicant enable
        set eap-method tls
        set eap-identity "test2@fortiqa.net"
        set eap-ca-cert "root_G_CA_Cert_1.cer"
        set eap-user-cert "root_eap_client_global.cer"
    next
end
```

2. Verify the interface's TLS authentication details:

```
# diagnose test application eap_supp 2
Interface: port33
status:Authorized
method: TLS
identity: test2@fortiqa.net
ca_cert: /etc/cert/ca/root_G_CA_Cert_1.cer
client_cert: /etc/cert/local/root_eap_client_global.cer
private_key: /etc/cert/local/root_eap_client_global.key
last_eapol_src =70:4c:a5:3b:0b:c6
```

Traffic is able to pass because the status is authorized.

Auto speed negotiation for 10G Base-T on FortiGate 100xF devices

Auto speed negotiation on the 10G Base-T interface allows the 1G/10G copper ports on the FortiGate 100xF devices to automatically handle both 1G and 10G speeds and duplex settings, eliminating the need for manual adjustments.

```
config system interface
    edit "port1"
        set speed auto
```

```
    next
end
```

Example

- When the peer uses 1G and the speed is the default value:

  ```
  config system interface
      edit "port1"
          set vdom "root"
          set type physical
          set speed 10000auto
      next
  end
  ```

  The port is down because of the different speeds:

  ```
  # get sys int physical port1
  == [onboard]
          ==[port1]
                  mode: static
                  ip: 0.0.0.0 0.0.0.0
                  ipv6: ::/0
                  status: down
                  speed: n/a
                  FEC: none
                  FEC_cap: {none}
  ```

- After the speed is set to auto:

  ```
  config system interface
      edit "port1"
          set speed auto
      next
  end
  ```

  The port is up and auto negotiates to the correct speed and duplex setting:

  ```
  # get sys int physical port1
  == [onboard]
          ==[port1]
                  mode: static
                  ip: 0.0.0.0 0.0.0.0
                  ipv6: ::/0
                  status: up
                  speed: 1000Mbps (Duplex: full)
                  FEC: none
                  FEC_cap: {none}
  ```

Physical interface

A FortiGate has several physical interfaces that can connect to Ethernet or optical cables. Depending on the FortiGate model, it can have a varying combination of Ethernet, small form-factor pluggable (SFP), and enhanced small form-factor pluggable (SFP+) interfaces.

The port names, as labeled on the FortiGate, appear in the interfaces list on the *Network > Interfaces* page. Hover the cursor over a port to view information, such as the name and the IP address.

Refer to Configuring an interface for basic GUI and CLI configuration steps.

For information about SFP/SFP+ interfaces, see:

- Transceiver status information for SFP and SFP+ interfaces on page 249
- GPON support for FortiGates with SFP/SFP+ ports NEW on page 250

Transceiver status information for SFP and SFP+ interfaces

Transceiver status information for SFP and SFP+ interfaces installed on the FortiGate can be displayed in the GUI and CLI. For example, the type, vendor name, part number, serial number, and port name. The CLI output includes additional information that can be useful for diagnosing transmission problems, such as the temperature, voltage, and optical transmission power.

To view transceiver status information in the GUI:

1. Go to *Network > Interfaces*. The *Transceiver* column is visible in the table, which displays the transceiver vendor name and part number.
2. Hover the cursor over a transceiver to view more information.

To view transceiver status information in the CLI:

```
# get system interface transceiver
Interface port9 - SFP/SFP+
  Vendor Name  :            FINISAR CORP.
  Part No.     :            FCLF-8521-3
  Serial No.   :            PMS***
Interface port10 - Transceiver is not detected.
Interface port11 - SFP/SFP+
  Vendor Name  :            QNC
  Part No.     :            LCP-1250RJ3SRQN
  Serial No.   :            QNDT****
Interface port12 - SFP/SFP+
  Vendor Name  :            QNC
  Part No.     :            LCP-1250RJ3SRQN
  Serial No.   :            QNDT****
Interface s1 - SFP/SFP+
  Vendor Name  :            JDSU
  Part No.     :            PLRXPLSCS4322N
  Serial No.   :            CB26U****
Interface s2 - SFP/SFP+
```

```
  Vendor Name  :            JDSU
  Part No.     :            PLRXPLSCS4321N
  Serial No.   :            C825U****
Interface vw1 - Transceiver is not detected.
Interface vw2 - Transceiver is not detected.
Interface x1 - SFP/SFP+
  Vendor Name  :            Fortinet
  Part No.     :            LCP-10GRJ3SRFN
  Serial No.   :            19090910****
Interface x2 - Transceiver is not detected.
                                    Optical      Optical      Optical
SFP/SFP+     Temperature  Voltage      Tx Bias      Tx Power     Rx Power
Interface    (Celsius)    (Volts)      (mA)         (dBm)        (dBm)
------------ ------------ ------------ ------------ ------------ ------------
port9         N/A          N/A          N/A          N/A          N/A
port11        N/A          N/A          N/A          N/A          N/A
port12        N/A          N/A          N/A          N/A          N/A
s1            38.3         3.35         6.80         -2.3         -3.2
s2            42.1         3.34         7.21         -2.3         -3.0
x1            N/A          N/A          N/A          N/A          N/A
  ++ : high alarm, + : high warning, - : low warning, -- : low alarm, ? : suspect.
```

GPON support for FortiGates with SFP/SFP+ ports NEW

FortiGates with SFP/SFP+ ports support Gigabit Passive Optical Network (GPON) transceivers, enabling the firewall to operate directly as an ONU (Optical Network Unit) and allowing deployment in fiber-based access networks. This allows customers transitioning from legacy DSL infrastructure to adopt widely used GPON technology, improving interoperability and easing migration.

Use the following diagnose commands:

Command	Description
`diagnose gpon config loid [interface] [loid] [password]`	Configure Logical Identifier (LOID) username and password.
`diagnose gpon sysinfo [interface]`	Show transceiver system information.
`diagnose gpon state [interface]`	Show SFP GPON connection state.
`diagnose gpon traffic-stats [interface]`	Show SFP GPON traffic statistics.
`diagnose gpon reboot [interface]`	Reboot transceiver.

To detect a GPON transceiver:

```
# get system interface transceiver sfp1
Interface sfp1 - SFP/SFP+/SFP28, 1000BASE-LX
  Diagnostics  :            temperature, voltage, Tx bias, Rx power (avg), Tx power
  Vendor Name  :            FORTINET
  Part No.     :            WST-SFP-GPAU2MIF
  Serial No.   :            TK5ANK0006
```

```
  Measurement  Unit         Value        High Alarm   High Warning Low Warning  Low Alarm
  ------------ ------------ ------------ ------------ ------------ ------------ ------------

  Temperature  (Celsius)     45.8         88.0         85.0        -40.0        -45.0
  Voltage      (Volts)       3.34         3.60         3.47         3.00         2.90
  Tx Bias      (mA)         14.49        60.00        55.00         0.00         0.00
  Rx Power Avg (dBm)         -9.2         -7.0         -8.0        -28.5        -29.6
  Tx Power     (dBm)          1.3          6.5          5.5         -0.5         -1.0
  ++ : high alarm, + : high warning, - : low warning, -- : low alarm, ? : suspect.
```

To check the GPON system information:

```
# diagnose gpon sysinfo sfp1
Mac Address: 80:A5:79:50:11:1D
Model Name: WST-SFP-GPAU2MIF
Vendor: FORTINET
Hardware Version: V1.0
Software Version: V1.2.4
Gpon SN: TK5ANK0006
```

To check the GPON state:

```
# diagnose gpon state sfp1
State codes: O1 - Initial, O2 - Standby, O3 - Serial-Number, O4 - Ranging
             O5 - Operation, O6 - POPUP, O7 - Emergency-Stop

los = 0
state = O5
fec = 1
```

To check GPON traffic statistics:

```
# diagnose gpon traffic-stats sfp1
dsDropEvents = 0
usDropEvents = 0
dsOcts = 14395972
usOcts = 34376
dsPkts = 224922
usPkts = 669
dsBcastPkts = 0
usBcastPkts = 0
dsMcastPkts = 0
usMcastPkts = 0
dsCrcErrPkts = 0
usCrcErrPkts = 0
dsUnderSzPkts = 0
usUnderSzPkts = 0
dsOverSzPkts = 0
usOverSzPkts = 0
dsFragments = 0
```

```
usFragments = 0
dsJabbers = 0
usJabbers = 0
dsPkts_l64 = 0
dsPkts_64_127 = 224920
dsPkts_128_255 = 0
dsPkts_256_511 = 2
dsPkts_512_1023 = 0
dsPkts_1024_1518 = 0
usPkts_l64 = 0
usPkts_64_127 = 0
usPkts_128_255 = 0
usPkts_256_511 = 2
usPkts_512_1023 = 0
usPkts_1024_1518 = 0
dsDiscards = 0
usDiscards = 0
dsErr = 0
usErr = 0
```

VLAN

Virtual local area networks (VLANs) multiply the capabilities of your FortiGate and can also provide added network security. VLANs use ID tags to logically separate devices on a network into smaller broadcast domains. These smaller domains forward packets only to devices that are part of that VLAN domain. This reduces traffic and increases network security.

VLANs in NAT mode

In NAT mode, the FortiGate unit functions as a layer-3 device. In this mode, the FortiGate unit controls the flow of packets between VLANs and can also remove VLAN tags from incoming VLAN packets. The FortiGate unit can also forward untagged packets to other networks such as the Internet.

In NAT mode, the FortiGate unit supports VLAN trunk links with IEEE 802.1Q-compliant switches or routers. The trunk link transports VLAN-tagged packets between physical subnets or networks. When you add VLAN subinterfaces to the FortiGate's physical interfaces, the VLANs have IDs that match the VLAN IDs of packets on the trunk link. The FortiGate unit directs packets with VLAN IDs to subinterfaces with matching IDs.

You can define VLAN subinterfaces on all FortiGate physical interfaces. However, if multiple virtual domains are configured on the FortiGate unit, you only have access to the physical interfaces on your virtual domain. The FortiGate unit can tag packets leaving on a VLAN subinterface. It can also remove VLAN tags from incoming packets and add a different VLAN tag to outgoing packets.

Normally in VLAN configurations, the FortiGate unit's internal interface is connected to a VLAN trunk, and the external interface connects to an Internet router that is not configured for VLANs. In this configuration, the FortiGate unit can apply different policies for traffic on each VLAN interface connected to the internal interface, which results in less network traffic and better security.

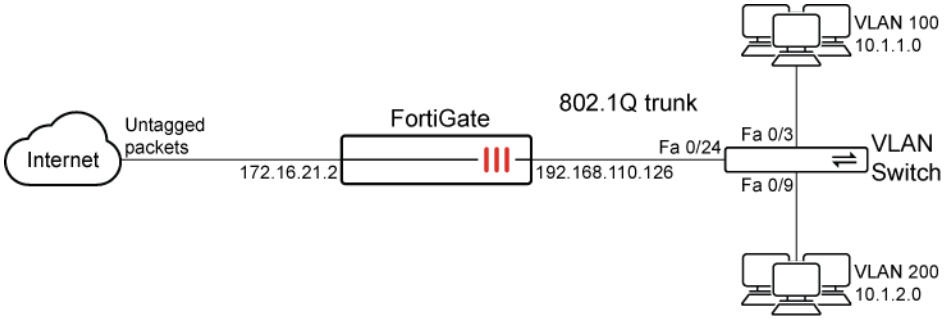
Sample topology

In this example, two different internal VLAN networks share one interface on the FortiGate unit and share the connection to the Internet. This example shows that two networks can have separate traffic streams while sharing a single interface. This configuration can apply to two departments in a single company or to different companies.

There are two different internal network VLANs in this example. VLAN_100 is on the 10.1.1.0/255.255.255.0 subnet, and VLAN_200 is on the 10.1.2.0/255.255.255.0 subnet. These VLANs are connected to the VLAN switch.

The FortiGate internal interface connects to the VLAN switch through an 802.1Q trunk. The internal interface has an IP address of 192.168.110.126 and is configured with two VLAN subinterfaces (VLAN_100 and VLAN_200). The external interface has an IP address of 172.16.21.2 and connects to the Internet. The external interface has no VLAN subinterfaces.

When the VLAN switch receives packets from VLAN_100 and VLAN_200, it applies VLAN ID tags and forwards the packets of each VLAN both to local ports and to the FortiGate unit across the trunk link. The FortiGate unit has policies that allow traffic to flow between the VLANs, and from the VLANs to the external network.



Sample configuration

In this example, both the FortiGate unit and the Cisco 2950 switch are installed and connected and basic configuration has been completed. On the switch, you need access to the CLI to enter commands. No VDOMs are enabled in this example.

General configuration steps include:

1. Configure the external interface.
2. Add two VLAN subinterfaces to the internal network interface.
3. Add firewall addresses and address ranges for the internal and external networks.
4. Add security policies to allow:
   - the VLAN networks to access each other.
   - the VLAN networks to access the external network.

To configure the external interface:

```
config system interface
    edit external
        set mode static
        set ip 172.16.21.2 255.255.255.0
```

```
    next
end
```

To add VLAN subinterfaces:

```
config system interface
    edit VLAN_100
        set vdom root
        set interface internal
        set type vlan
        set vlanid 100
        set mode static
        set ip 10.1.1.1 255.255.255.0
        set allowaccess https ping
    next
    edit VLAN_200
        set vdom root
        set interface internal
        set type vlan
        set vlanid 200
        set mode static
        set ip 10.1.2.1 255.255.255.0
        set allowaccess https ping
    next
end
```

To add the firewall addresses:

```
config firewall address
    edit VLAN_100_Net
        set type ipmask
        set subnet 10.1.1.0 255.255.255.0
    next
    edit VLAN_200_Net
        set type ipmask
        set subnet 10.1.2.0 255.255.255.0
    next
end
```

To add security policies:

Policies 1 and 2 do not need NAT enabled, but policies 3 and 4 do need NAT enabled.

```
config firewall policy
    edit 1
        set srcintf VLAN_100
        set srcaddr VLAN_100_Net
        set dstintf VLAN_200
        set dstaddr VLAN_200_Net
        set schedule always
```

```
        set service ALL
        set action accept
        set nat disable
        set status enable
    next
    edit 2
        set srcintf VLAN_200
        set srcaddr VLAN_200_Net
        set dstintf VLAN_100
        set dstaddr VLAN_100_Net
        set schedule always
        set service ALL
        set action accept
        set nat disable
        set status enable
    next
    edit 3
        set srcintf VLAN_100
        set srcaddr VLAN_100_Net
        set dstintf external
        set dstaddr all
        set schedule always
        set service ALL
        set action accept
        set nat enable
        set status enable
    next
    edit 4
        set srcintf VLAN_200
        set srcaddr VLAN_200_Net
        set dstintf external
        set dstaddr all
        set schedule always
        set service ALL
        set action accept
        set nat enable
        set status enable
    next
end
```

VLANs in transparent mode

In transparent mode, the FortiGate unit behaves like a layer-2 bridge but can still provide services such as antivirus scanning, web filtering, spam filtering, and intrusion protection to traffic. Some limitations of transparent mode is that you cannot use Agentless VPN, PPTP/L2TP VPN, DHCP server, or easily perform NAT on traffic. The limits in transparent mode apply to IEEE 802.1Q VLAN trunks passing through the unit.

You can insert the FortiGate unit operating in transparent mode into the VLAN trunk without making changes to your network. In a typical configuration, the FortiGate unit internal interface accepts VLAN packets on a VLAN trunk from a VLAN switch or router connected to internal network VLANs. The FortiGate external interface forwards VLAN-tagged packets through another VLAN trunk to an external VLAN switch or router and on to

external networks such as the Internet. You can configure the unit to apply different policies for traffic on each VLAN in the trunk.

To pass VLAN traffic through the FortiGate unit, you add two VLAN subinterfaces with the same VLAN ID, one to the internal interface and the other to the external interface. You then create a security policy to permit packets to flow from the internal VLAN interface to the external VLAN interface. If required, create another security policy to permit packets to flow from the external VLAN interface to the internal VLAN interface. Typically in transparent mode, you do not permit packets to move between different VLANs. Network protection features such as spam filtering, web filtering, and anti-virus scanning, are applied through the UTM profiles specified in each security policy, enabling very detailed control over traffic.

When the FortiGate unit receives a VLAN-tagged packet on a physical interface, it directs the packet to the VLAN subinterface with the matching VLAN ID. The VLAN tag is removed from the packet and the FortiGate unit then applies security policies using the same method it uses for non-VLAN packets. If the packet exits the FortiGate unit through a VLAN subinterface, the VLAN ID for that subinterface is added to the packet and the packet is sent to the corresponding physical interface.

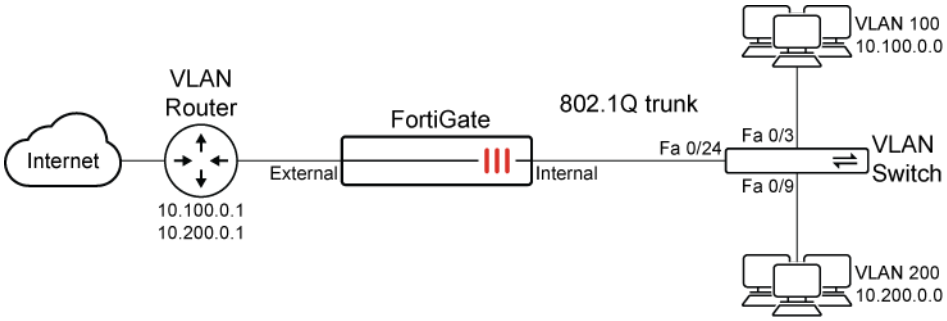
Sample topology

In this example, the FortiGate unit is operating in transparent mode and is configured with two VLANs: one with an ID of 100 and the other with ID 200. The internal and external physical interfaces each have two VLAN subinterfaces, one for VLAN_100 and one for VLAN_200.

The IP range for the internal VLAN_100 network is 10.100.0.0/255.255.0.0, and for the internal VLAN_200 network is 10.200.0.0/255.255.0.0.

The internal networks are connected to a Cisco 2950 VLAN switch which combines traffic from the two VLANs onto one in the FortiGate unit's internal interface. The VLAN traffic leaves the FortiGate unit on the external network interface, goes on to the VLAN switch, and on to the Internet. When the FortiGate units receives a tagged packet, it directs it from the incoming VLAN subinterface to the outgoing VLAN subinterface for that VLAN.

In this example, we create a VLAN subinterface on the internal interface and another one on the external interface, both with the same VLAN ID. Then we create security policies that allow packets to travel between the VLAN_100_int interface and the VLAN_100_ext interface. Two policies are required: one for each direction of traffic. The same is required between the VLAN_200_int interface and the VLAN_200_ext interface, for a total of four security policies.



Sample configuration

There are two main steps to configure your FortiGate unit to work with VLANs in transparent mode:

1. Add VLAN subinterfaces.
2. Add security policies.

You can also configure the protection profiles that manage antivirus scanning, web filtering, and spam filtering.

To add VLAN subinterfaces:

```
config system interface
    edit VLAN_100_int
        set type vlan
        set interface internal
        set vlanid 100
    next
    edit VLAN_100_ext
        set type vlan
        set interface external
        set vlanid 100
    next
    edit VLAN_200_int
        set type vlan
        set interface internal
        set vlanid 200
    next
    edit VLAN_200_ext
        set type vlan
        set interface external
        set vlanid 200
    next
end
```

To add security policies:

```
config firewall policy
    edit 1
        set srcintf VLAN_100_int
        set srcaddr all
        set dstintf VLAN_100_ext
        set dstaddr all
        set action accept
        set schedule always
        set service ALL
    next
    edit 2
        set srcintf VLAN_100_ext
        set srcaddr all
        set dstintf VLAN_100_int
        set dstaddr all
        set action accept
        set schedule always
        set service ALL
    next
```

```
    edit 3
        set srcintf VLAN_200_int
        set srcaddr all
        set dstintf VLAN_200_ext
        set dstaddr all
        set action accept
        set schedule always
        set service ALL
    next
    edit 4
        set srcintf VLAN_200_ext
        set srcaddr all
        set dstintf VLAN_200_int
        set dstaddr all
        set action accept
        set schedule always
        set service ALL
    next
end
```

Virtual VLAN switch

The hardware switch ports on FortiGate models that support virtual VLAN switches can be used as a layer 2 switch. Virtual VLAN switch mode allows 802.1Q VLANs to be assigned to ports, and the configuration of one interface as a trunk port.

The following FortiGate series are supported in FortiOS 8.0: 40F, 50G, 60F, 70F, 70G, 80F, 90G, 100F, 120G, 200F, 200G, 300E, 400E, 400F, 600F, 700G, 900G, 1100E, 1800F, 2600F, 3000F, 3200F, 3500F, 3700F, 4200F, 4400F, and 4800F. FortiWiFi 60F models are not supported.

The `virtual-switch-vlan` option must be enabled in the CLI to configure VLAN switch mode from the GUI or CLI.

To enable VLAN switches:

```
config system global
    set virtual-switch-vlan enable
end
```

After this setting is enabled, any previously configured hardware switches will appear in the *Network > Interfaces* page under *VLAN Switch*.

To enable VLAN switch mode in the GUI:

1. Go to *System > Settings > General*.
2. In the *System operation settings* section, enable *VLAN switch mode*.
3. Click *Apply*.

Basic configurations

Hardware switch ports can be configured as either a VLAN switch port or a trunk port. The available interfaces and allowable VLAN IDs that can be used depend on the FortiGate model. It is recommended to remove ports from the default VLAN switch before you begin configurations.

To create a new VLAN and assign ports in the GUI:

1. Go to *Network* > *Interfaces* and click *Create New* > *Interface*.
2. Enter a name and configure the following:
   a. Set the *Type* to *VLAN Switch*.
   b. Enter a *VLAN ID*.
   c. Click the *+* and add the *Interface Members*.
   d. Configure the *Address* and *Administrative Access* settings as needed.
3. Click *OK*.

To create a new VLAN and assign ports in the CLI:

1. Configure the VLAN:

```
config system virtual-switch
    edit "VLAN10"
        set physical-switch "sw0"
        set vlan 10
        config port
            edit "internal1"
            next
            edit "internal2"
            next
        end
    next
end
```

2. Configure the VLAN switch interface addressing:

```
config system interface
    edit "VLAN10"
        set vdom "root"
        set ip 192.168.10.99 255.255.255.0
        set allowaccess ping https ssh snmp http fgfm
        set type hard-switch
    next
end
```

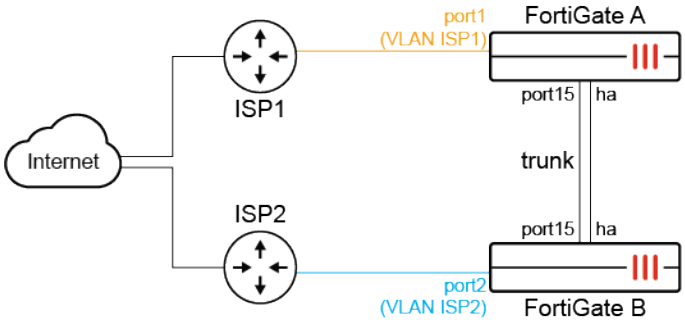
To designate an interface as a trunk port:

```
config system interface
    edit internal5
        set trunk enable
```

```
    next
end
```

Example 1: HA using a VLAN switch

In this example, two FortiGates in an HA cluster are connected to two ISP routers. Instead of connecting to external L2 switches, each FortiGate connects to each ISP router on the same hardware switch port on the same VLAN. A trunk port connects the two FortiGates to deliver the 802.1Q tagged traffic to the other. A full mesh between the FortiGate cluster and the ISP routers is achieved where no single point of failure will cause traffic disruptions.



This example assumes that the HA settings are already configured. The interface and VLAN switch settings are identical between cluster members and synchronized. See HA using a hardware switch to replace a physical switch on page 4168 for a similar example that does not use a VLAN switch.

To configure the VLAN switches:

1. Configure the ISP interfaces with the corresponding VLAN IDs:

```
config system virtual-switch
    edit "ISP1"
        set physical-switch "sw0"
        set vlan 2951
        config port
            edit "port1"
            next
        end
    next
    edit "ISP2"
        set physical-switch "sw0"
        set vlan 2952
        config port
            edit "port2"
            next
        end
    next
end
```

2. Configure the VLAN switch interface addressing:

```
config system interface
    edit "ISP1"
        set vdom "root"
        set ip 192.168.10.99 255.255.255.0
        set allowaccess ping
        set type hard-switch
    next
    edit "ISP2"
        set vdom "root"
        set ip 192.168.20.99 255.255.255.0
        set allowaccess ping
        set type hard-switch
    next
end
```

3. Designate port15 as the trunk port:

```
config system interface
    edit port15
        set trunk enable
    next
end
```

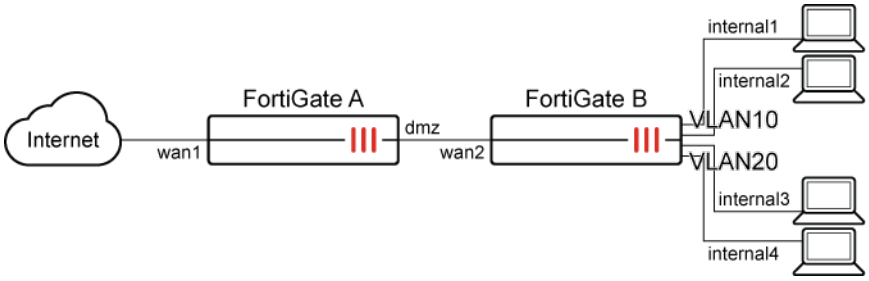
4. Configure firewall policies to allow outgoing traffic on the ISP1 and ISP2 interfaces:

```
config firewall policy
    edit 1
        set srcintf "port11"
        set dstintf "ISP1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set nat enable
    next
    edit 2
        set srcintf "port11"
        set dstintf "ISP2"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

Example 2: LAN extension

In this example, two hardware switch ports are assigned VLAN10, and two ports are assigned VLAN20 on FortiGate B. The wan2 interface is designated as the trunk port, and is connected to the upstream FortiGate A.

The corresponding VLAN subinterfaces VLAN10 and VLAN20 on the upstream FortiGate allow further access to other networks.



The available interfaces and VLAN IDs varies between FortiGate models. The FortiGate B in this example is a 60F model.

To configure FortiGate B:

1. Configure the VLAN interfaces:

```
config system virtual-switch
    edit "VLAN10"
        set physical-switch "sw0"
        set vlan 10
        config port
            edit "internal1"
            next
            edit "internal2"
            next
        end
    next
    edit "VLAN20"
        set physical-switch "sw0"
        set vlan 20
        config port
            edit "internal3"
            next
            edit "internal4"
            next
        end
    next
end
```

2. Configure the VLAN switch interface addressing:

```
config system interface
    edit "VLAN10"
        set vdom "root"
        set ip 192.168.10.99 255.255.255.0
        set allowaccess ping https ssh snmp http fgfm
        set type hard-switch
    next
```

```
    edit "VLAN20"
        set vdom "root"
        set ip 192.168.20.99 255.255.255.0
        set allowaccess ping https ssh snmp http fgfm
        set type hard-switch
    next
end
```

3. Designate wan2 as the trunk port:

```
config system interface
    edit wan2
        set trunk enable
    next
end
```

To configure FortiGate A:

1. Configure the VLAN subinterfaces:

```
config system interface
    edit "VLAN10"
        set ip 192.168.10.98 255.255.255.0
        set allowaccess ping https ssh
        set role lan
        set interface "dmz"
        set vlanid 10
    next
    edit "VLAN20"
        set ip 192.168.20.98 255.255.255.0
        set allowaccess ping https ssh
        set role lan
        set interface "dmz"
        set vlanid 20
    next
end
```

2. Configure the DHCP server on VLAN10:

```
config system dhcp server
    edit 0
        set dns-service default
        set default-gateway 192.168.10.98
        set netmask 255.255.255.0
        set interface "VLAN10 "
        config ip-range
            edit 1
                set start-ip 192.168.10.100
                set end-ip 192.168.10.254
            next
        end
        set timezone-option default
```

```
    next
end
```

3. Configure firewall policies that allow traffic from the VLAN10 and VLAN20 interfaces to the internet:

```
config firewall policy
    edit 0
        set name "VLAN10-out"
        set srcintf "VLAN10"
        set dstintf "wan1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set nat enable
    next
    edit 0
        set name "VLAN20-out"
        set srcintf "VLAN20"
        set dstintf "wan1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

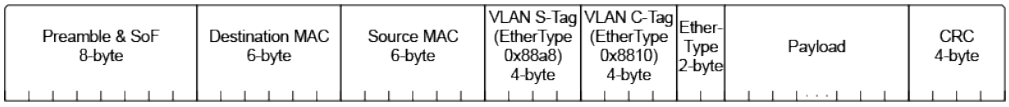
To test the connection:

1. Connect a PC to internal1 on FortiGate B.
2. Verify that it receives an IP address from FortiGate A's DHCP server.
3. From the PC, ping FortiGate B on 192.168.10.99.
4. Ping FortiGate A on 192.168.10.98.
5. Connect to the internet. Traffic is allowed by the VLAN10-out policy.

QinQ 802.1Q in 802.1ad

QinQ (802.1ad) allows multiple VLAN tags to be inserted into a single frame, and can be configured on supported FortiGate devices.

In this example, the customer connects to a provider that uses 802.1ad double-tagging to separate their customer VLANs. The FortiGate connecting to the provider double-tags its frames with an outer provider-tag (S-Tag) and an inner customer-tag (C-Tag).



The customer identifies itself with the provider-tag (S-Tag) 232 and uses the customer-tag (C-Tag) 444 for traffic to its VLAN.

QinQ is not supported on top of hardware switch interfaces.

To configure the interfaces:

1. Configure the interface to the provider that uses the outer tag (S-Tag):

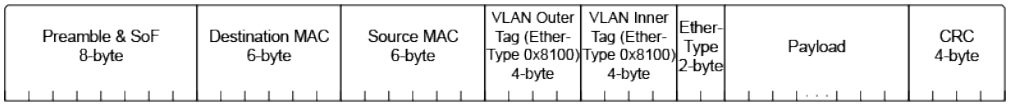
```
config system interface
    edit "vlan-8021ad"
        set vdom "root"
        set vlan-protocol 8021ad
        set device-identification enable
        set role lan
        set snmp-index 47
        set interface "PORT"
        set vlanid 232
    next
end
```

2. Configure a dynamic VLAN interface that uses the inner tag (C-Tag):

```
config system interface
    edit "DVLAN"
        set vdom "vdom1"
        set device-identification enable
        set role lan
        set snmp-index 48
        set interface "vlan-8021ad"
        set vlanid 444
    next
end
```

QinQ 802.1Q in 802.1Q

QinQ (802.1Q in 802.1Q) is supported for FortiGate VM models, where multiple VLAN tags can be inserted into a single frame.



In this example, the FortiGate VM is connected to a provider vSwitch and then a customer switch. The FortiGate encapsulates the frame with an outer 802.1Q tag of VLAN 100 and an inner 802.1Q tag of VLAN 200; port5 is used as the physical port. The provider vSwitch strips the outer tag and forwards traffic to the appropriate customer. Then the customer switch strips the inner tag and forwards the packet to the appropriate customer VLAN.

QinQ is not supported on top of hardware switch interfaces.

To configure the interfaces:

1. Configure the interface to the provider that uses the outer tag:

```
config system interface
    edit "vlan-8021q"
        set vdom "root"
        set device-identification enable
        set role lan
        set interface "port5"
        set vlan-protocol 8021q
        set vlanid 100
    next
end
```

2. Configure the interface to the provider that uses the inner tag:

```
config system interface
    edit "vlan-qinq8021q"
        set vdom "root"
        set ip 1.1.1.71 255.255.255.0
        set allowaccess ping https ssh snmp http
        set device-identification enable
        set role lan
        set interface "vlan-8021q"
        set vlanid 200
    next
end
```

To verify the traffic:

1. From the FortiGate, ping 1.1.1.72:

```
# execute ping 1.1.1.72
PING 1.1.1.72 (1.1.1.72): 56 data bytes
64 bytes from 1.1.1.72: icmp_seq=0 ttl=255 time=0.2 ms
64 bytes from 1.1.1.72: icmp_seq=1 ttl=255 time=0.1 ms
64 bytes from 1.1.1.72: icmp_seq=2 ttl=255 time=0.1 ms
64 bytes from 1.1.1.72: icmp_seq=3 ttl=255 time=0.1 ms
^C
--- 1.1.1.72 ping statistics ---
4 packets transmitted, 4 packets received, 0% packet loss
round-trip min/avg/max = 0.1/0.1/0.2 ms
```

2. Verify the packet capture frame header output captured from the FortiGate's port5:

```
Frame 2: 106 bytes on wire (848 bits), 106 bytes captured (848 bits)
Ethernet II, Src: VMware_93:ae:8f (00:50:56:93:ae:8f), Dst: VMware_93:e3:72
(00:50:56:93:e3:72)
    Destination: VMware_93:e3:72 (00:50:56:93:e3:72)
    Source: VMware_93:ae:8f (00:50:56:93:ae:8f)
    Type: 802.1Q Virtual LAN (0x8100)
802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 100
    000. .... .... .... = Priority: Best Effort (default) (0)
    ...0 .... .... .... = DEI: Ineligible
    .... 0000 0110 0100 = ID: 100
    Type: 802.1Q Virtual LAN (0x8100)
802.1Q Virtual LAN, PRI: 0, DEI: 0, ID: 200
    000. .... .... .... = Priority: Best Effort (default) (0)
    ...0 .... .... .... = DEI: Ineligible
    .... 0000 1100 1000 = ID: 200
    Type: IPv4 (0x0800)
Internet Protocol Version 4, Src: 1.1.1.71, Dst: 1.1.1.72
Internet Control Message Protocol
```

The outer tag (first tag) is an 802.1Q tag with VLAN ID 100. The inner tag (second tag) is also an 802.1Q tag with VLAN ID 200.

Aggregation and redundancy

Link aggregation (IEEE 802.3ad/802.1ax) enables you to bind two or more physical interfaces together to form an aggregated (combined) link. This new link has the bandwidth of all the links combined. If a link in the group fails, traffic is transferred automatically to the remaining interfaces. The only noticeable effect is reduced bandwidth.

This feature is similar to redundant interfaces. The major difference is a redundant interface group only uses one link at a time, where an aggregate link group uses the total bandwidth of the functioning links in the group, up to eight (or more).

An interface is available to be an aggregate interface if:

- It is a physical interface and not a VLAN interface or subinterface.
- It is not already part of an aggregate or redundant interface.
- It is in the same VDOM as the aggregated interface. Aggregate ports cannot span multiple VDOMs.
- It does not have an IP address and is not configured for DHCP or PPPoE.
- It is not referenced in any security policy, VIP, IP Pool, or multicast policy.
- It is not an HA heartbeat interface.
- It is not one of the FortiGate-5000 series backplane interfaces.

When an interface is included in an aggregate interface, it is not listed on the *Network* > *Interfaces* page. Interfaces still appear in the CLI although configuration for those interfaces do not take affect. You cannot configure the interface individually and it is not available for inclusion in security policies, VIPs, IP pools, or routing.

Example configuration

This example creates an aggregate interface on a FortiGate-140D POE using ports 3-5 with an internal IP address of 10.1.1.123, as well as the administrative access to HTTPS and SSH.

To create an aggregate interface in the GUI:

1. Go to *Network* > *Interfaces* and select *Create New* > *Interface*.
2. Set *Name* to *aggregate*.
3. Set *Type* to *802.3ad Aggregate*.
4. Set *Interface members* to *port4*, *port5*, and *port6*.
5. Set *Addressing mode* to *Manual*.
6. Set *IP/Netmask* to *10.1.1.123/24*.
7. For *Administrative Access*, select *HTTPS* and *SSH*.
8. Click *OK*.

To create an aggregate interface in the CLI:

```
config system interface
    edit "aggregate"
        set vdom "root"
        set ip 10.1.1.123 255.255.255.0
        set allowaccess https ssh
        set type aggregate
        set member "port4" "port5" "port6"
        set snmp-index 45
    next
end
```

Redundancy

In a redundant interface, traffic only goes over one interface at any time. This differs from an aggregated interface where traffic goes over all interfaces for increased bandwidth. This difference means redundant interfaces can have more robust configurations with fewer possible points of failure. This is important in a fully-meshed HA configuration.

An interface is available to be in a redundant interface if:

- It is a physical interface and not a VLAN interface.
- It is not already part of an aggregated or redundant interface.
- It is in the same VDOM as the redundant interface.
- It does not have an IP address and is not configured for DHCP or PPPoE.
- It has no DHCP server or relay configured on it.
- It does not have any VLAN subinterfaces.
- It is not referenced in any security policy, VIP, or multicast policy.
- It is not monitored by HA.
- It is not one of the FortiGate-5000 series backplane interfaces.

When an interface is included in a redundant interface, it is not listed on the *Network > Interfaces* page. You cannot configure the interface individually and it is not available for inclusion in security policies, VIPs, or routing.

Example configuration

To create a redundant interface in the GUI:

1. Go to *Network > Interfaces* and select *Create New > Interface*.
2. Set *Name* to *redundant*.
3. Set *Type* to *Redundant Interface*.
4. Set *Interface members* to *port4*, *port5*, and *port6*.
5. Set *Addressing mode* to *Manual*.
6. Set *IP/Netmask* to *10.13.101.100/24*.
7. For *Administrative Access*, select *HTTPS* and *SSH*.
8. Click *OK*.

To create a redundant interface in the CLI:

```
config system interface
    edit "redundant"
        set vdom "root"
        set ip 10.13.101.100 255.255.255.0
        set allowaccess https http
        set type redundant
        set member "port4" "port5" "port6"
        set snmp-index 9
    next
end
```

Enhanced hashing for LAG member selection

FortiGate models that have an internal switch that supports modifying the distribution algorithm can use enhanced hashing to help distribute traffic evenly, or load balance, across links on the Link Aggregation (LAG) interface.

The enhanced hashing algorithm is based on a 5-tuple of the IP protocol, source IP address, destination IP address, source port, and destination port.

Different computation methods allow for more variation in the load balancing distribution, in case one algorithm does not distribute traffic evenly between links across different XAUIs. The available methods are:

xor16	Use the XOR operator to make a 16 bit hash.
xor8	Use the XOR operator to make an 8 bit hash.
xor4	Use the XOR operator to make a 4 bit hash.
crc16	Use the CRC-16-CCITT polynomial to make a 16 bit hash.

The following NP6 non-service FortiGate models support this feature: 1500D, 1500DT, 3000D, 3100D, 3200D, 3700D, and 5001D.

To configure the enhanced hashing:

```
config system npu
    set lag-out-port-select {enable | disable}
    config sw-eh-hash
        set computation {xor4 | xor8 | xor16 | crc16}
        set ip-protocol {include | exclude}
        set source-ip-upper-16 {include | exclude}
        set source-ip-lower-16 {include | exclude}
        set destination-ip-upper-16 {include | exclude}
        set destination-ip-lower-16 {include | exclude}
        set source-port {include | exclude}
        set destination-port {include | exclude}
        set netmask-length {0 - 32}
    end
end
```

For example, to use XOR16 and include all of the fields in the 5-tuple to compute the link in the LAG interface that the packet is distributed to:

```
config system npu
    set lag-out-port-select enable
    config sw-eh-hash
        set computation xor16
        set ip-protocol include
        set source-ip-upper-16 include
        set source-ip-lower-16 include
        set destination-ip-upper-16 include
        set destination-ip-lower-16 include
        set source-port include
        set destination-port include
        set netmask-length 32
    end
end
```

LAG interface status signals to peer device

FortiGate can signal LAG (link aggregate group) interface status to the peer device. If the number of available links in the LAG on the FortiGate falls below the configured minimum number of links (`min-links`), the LAG interface goes down on both the FortiGate and the peer device.

When the minimum number of links is satisfied again, the LAG interface automatically resumes operation on both the FortiGate and the peer device. While the LAG interface is down, interface members are in the Link Aggregation Control Protocol (LACP) MUX state of *Waiting*.

Example

In this example, the LAG interface is configured on FGT_A and peered with FGT_B.

To verify the configuration:

1. On FGT_A, check the minimum number of links for the LAG interface named test_agg1.
   In the following example, `set min-links 1` indicates that a minimum of one alive interface member is required to keep the LAG interface up.

```
# show
config system interface
    edit "test_agg1"
        set vdom "vdom1"
        set ip 11.1.1.1 255.255.255.0
        set allowaccess ping https
        set type aggregate
        set member "port7" "port8" "port9"
        set device-identification enable
        set lldp-transmission enable
        set role lan
        set snmp-index 41
        set min-links 1
    next
end
```

2. Change the status of port9 to down.

```
config system interface
    edit port9
        set status down
    next
end
```

3. On FGT_A, test the LAG interface named test_agg1.
   The status is up for test_agg1 interface because two interface members (port7 and port8) are up, and only one interface member (port9) is down.

```
# diagnose netlink aggregate name test_agg1
LACP flags: (A|P)(S|F)(A|I)(I|O)(E|D)(E|D)
(A|P) - LACP mode is Active or Passive
(S|F) - LACP speed is Slow or Fast
(A|I) - Aggregatable or Individual
(I|O) - Port In sync or Out of sync
(E|D) - Frame collection is Enabled or Disabled
(E|D) - Frame distribution is Enabled or Disabled

status: up
npu: y
flush: n
asic helper: y
oid: 72
```

```
ports: 3
link-up-delay: 50ms
min-links: 1
ha: master
distribution algorithm: L4
LACP mode: active
LACP speed: slow
LACP HA: enable
aggregator ID: 1
actor key: 17
actor MAC address: d4:76:a0:01:e0:44
partner key: 17
partner MAC address: d4:76:a0:01:e8:1e

member: port7
  index: 0
  link status: up
  link failure count: 1
  permanent MAC addr: d4:76:a0:01:e0:44
  LACP state: established
  LACPDUs RX/TX: 4/17
  actor state: ASAIEE
  actor port number/key/priority: 1 17 255
  partner state: ASAIEE
  partner port number/key/priority: 1 17 255
  partner system: 1 d4:76:a0:01:e8:1e
  aggregator ID: 1
  speed/duplex: 1000 1
  RX state: CURRENT 6
  MUX state: COLLECTING_DISTRIBUTING 4

member: port8
  index: 1
  link status: up
  link failure count: 2
  permanent MAC addr: d4:76:a0:01:e0:45
  LACP state: established
  LACPDUs RX/TX: 216/222
  actor state: ASAIEE
  actor port number/key/priority: 2 17 255
  partner state: ASAIEE
  partner port number/key/priority: 2 17 255
  partner system: 1 d4:76:a0:01:e8:1e
  aggregator ID: 1
  speed/duplex: 1000 1
  RX state: CURRENT 6
  MUX state: COLLECTING_DISTRIBUTING 4

member: port9
  index: 2
  link status: down
  link failure count: 0
```

```
  permanent MAC addr: d4:76:a0:01:e0:46
```

4. On FGT_A, change the minimum number of links to 3.

```
config system interface
    edit "test_agg1"
        set vdom "vdom1"
        set ip 11.1.1.1 255.255.255.0
        set allowaccess ping https
        set type aggregate
        set member "port7" "port8" "port9"
        set device-identification enable
        set lldp-transmission enable
        set role lan
        set snmp-index 41
        set min-links 3
    next
end
```

5. On FGT_A, check the LAG interface named test_agg1:

   The status is down for test_agg1 interface because only two of the three required interface members are up. Interface members port7 and port8 are up, but interface member port9 is down.

```
# diagnose netlink aggregate name test_agg1
LACP flags: (A|P)(S|F)(A|I)(I|O)(E|D)(E|D)
(A|P) - LACP mode is Active or Passive
(S|F) - LACP speed is Slow or Fast
(A|I) - Aggregatable or Individual
(I|O) - Port In sync or Out of sync
(E|D) - Frame collection is Enabled or Disabled
(E|D) - Frame distribution is Enabled or Disabled

status: down
npu: y
flush: n
asic helper: y
oid: 230
ports: 3
link-up-delay: 50ms
min-links: 3
ha: master
distribution algorithm: L4
LACP mode: active
LACP speed: slow
LACP HA: enable
aggregator ID: 1
actor key: 17
actor MAC address: e8:1c:ba:b3:d0:df
partner key: 17
partner MAC address: e8:1c:ba:df:a0:ba

member: port7
```

```
  index: 0
  link status: up
  link failure count: 1
  permanent MAC addr: e8:1c:ba:b3:d0:df
  LACP state: negotiating
  LACPDUs RX/TX: 10/23
  actor state: ASAODD
  actor port number/key/priority: 1 17 255
  partner state: ASAIDD
  partner port number/key/priority: 1 17 255
  partner system: 61440 e8:1c:ba:df:a0:ba
  aggregator ID: 1
  speed/duplex: 1000 1
  RX state: CURRENT 6
  MUX state: WAITING 2

member: port8
  index: 1
  link status: up
  link failure count: 1
  permanent MAC addr: e8:1c:ba:b3:d0:e0
  LACP state: negotiating
  LACPDUs RX/TX: 222/228
  actor state: ASAODD
  actor port number/key/priority: 2 17 255
  partner state: ASAIDD
  partner port number/key/priority: 65 17 255
  partner system: 61440 e8:1c:ba:df:a0:ba
  aggregator ID: 1
  speed/duplex: 1000 1
  RX state: CURRENT 6
  MUX state: WAITING 2

member: port9
  index: 2
  link status: down
  link failure count: 0
  permanent MAC addr: e8:1c:ba:b3:d0:ed
```

6. On the peer FortiGate (FGT_B), check the LAG interface status.

   The status is down for test_agg2 interface due to FortiGate's ability to signal LAG interface status to the peer device. While interface members port7 and port8 are up, interface member port9 is down.

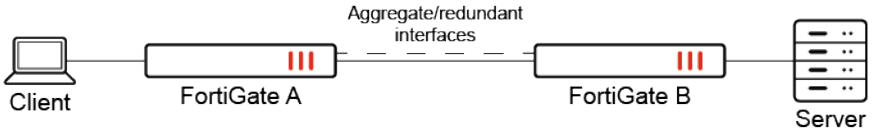
```
# diagnose netlink aggregate name test_agg2
LACP flags: (A|P)(S|F)(A|I)(I|O)(E|D)(E|D)
(A|P) - LACP mode is Active or Passive
(S|F) - LACP speed is Slow or Fast
(A|I) - Aggregatable or Individual
(I|O) - Port In sync or Out of sync
(E|D) - Frame collection is Enabled or Disabled
(E|D) - Frame distribution is Enabled or Disabled

status: down
```

```
npu: y
flush: n
asic helper: y
oid: 72
ports: 3
link-up-delay: 50ms
min-links: 1
ha: master
distribution algorithm: L4
LACP mode: active
LACP speed: slow
LACP HA: enable
aggregator ID: 1
actor key: 17
actor MAC address: d4:76:a0:01:e8:1e
partner key: 17
partner MAC address: d4:76:a0:01:e0:44

member: port7
  index: 0
  link status: up
  link failure count: 1
  permanent MAC addr: d4:76:a0:01:e8:1e
  LACP state: negotiating
  LACPDUs RX/TX: 13/14
  actor state: ASAIDD
  actor port number/key/priority: 1 17 255
  partner state: ASAODD
  partner port number/key/priority: 1 17 255
  partner system: 44237 d4:76:a0:01:e0:44
  aggregator ID: 1
  speed/duplex: 1000 1
  RX state: CURRENT 6
  MUX state: ATTACHED 3

member: port8
  index: 1
  link status: up
  link failure count: 1
  permanent MAC addr: d4:76:a0:01:e8:1f
  LACP state: negotiating
  LACPDUs RX/TX: 15/14
  actor state: ASAIDD
  actor port number/key/priority: 2 17 255
  partner state: ASAODD
  partner port number/key/priority: 2 17 255
  partner system: 44237 d4:76:a0:01:e0:44
  aggregator ID: 1
  speed/duplex: 1000 1
  RX state: CURRENT 6
  MUX state: ATTACHED 3
```

```
member: port9
  index: 2
  link status: down
  link failure count: 0
  permanent MAC addr: d4:76:a0:01:e8:20
```

Failure detection for aggregate and redundant interfaces

When an aggregate or redundant interface goes down, the corresponding fail-alert interface changes to down. When an aggregate or redundant interface comes up, the corresponding fail-alert interface changes to up.



Fail-detect for aggregate and redundant interfaces can be configured using the CLI.

To configure an aggregate interface so that port3 goes down with it:

```
config system interface
    edit "agg1"
        set vdom "root"
        set fail-detect enable
        set fail-alert-method link-down
        set fail-alert-interfaces "port3"
        set type aggregate
        set member "port1" "port2"
    next
end
```

To configure a redundant interface so that port4 goes down with it:

```
config system interface
    edit "red1"
        set vdom "root"
        set fail-detect enable
        set fail-alert-method link-down
        set fail-alert-interfaces "port4"
        set type redundant
        set member "port1" "port2"
    next
end
```

Loopback interface

A loopback interface is a logical interface that is always up. Its IP address does not depend on one specific physical port, and the attached subnet is always present in the routing table. Therefore, it can be accessed

through several physical or VLAN interfaces.

Typically, a loopback interface can be used with management access, BGP peering, PIM rendezvous points, and SD-WAN.

A loopback interface requires appropriate firewall policies to allow traffic to the interface. For example, see IPsec tunnel terminated on a loopback interface on page 277.

> GRE tunnels and other IP/IPv6 tunnels, where tunnel traffic is handled before the route lookup, do not require firewall policies to allow traffic to the interface. The ingress interface is changed to the tunnel interface by the time the policy is checked.

Multiple loopback interfaces can be configured in either non-VDOM mode or in each VDOM.

Dynamic routing protocols can be enabled on loopback interfaces. For example, loopback interfaces are a good practice for OSPF. To make it easier to troubleshoot OSPF, set the OSPF router ID to the same value as the loopback IP address to access a specific FortiGate using that IP address and SSH.

A loopback interface is configured using similar steps as a physical interface (see Configuring an interface).

IPsec tunnel terminated on a loopback interface

As mentioned above, a loopback interface requires appropriate firewall policies to allow traffic to the interface. In other words, traffic ingressing on an interface that is destined for the IP address associated with a loopback interface requires an appropriate firewall policy from that interface to the loopback interface otherwise the traffic will be dropped.

For example, consider the following topology where an IPsec tunnel is terminated on a loopback interface, VPN_LO, on the FortiGate FGT-1 and on a WAN interface on the FortiGate FGT-2.

We will focus on the configuration required for FortiGate FGT-1.

IPsec tunnel terminates on a loopback interface, VPN_LO, which has an associated IP address that the remote peer will use as its IPsec remote gateway address.

The IPsec tunnel uses wan1 as its underlay interface.

In this scenario, the administrator of the FortiGate FGT-1 device must configure a firewall policy from the wan1 interface to the VPN_LO interface that allows incoming traffic from the remote peer to reach the VPN_LO interface for proper IPsec tunnel connectivity.

For example:

```
config firewall policy
    edit 4
        set name "Loopback-In"
        set srcintf "wan1"
        set dstintf "VPN_LO"
        set srcaddr "all"
```

```
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic disable
    next
end
```

Software switch

A software switch is a virtual switch that is implemented at the software or firmware level and not at the hardware level. A software switch can be used to simplify communication between devices connected to different FortiGate interfaces. For example, a software switch lets you place the FortiGate interface connected to an internal network on the same subnet as your wireless interfaces. Then devices on the internal network can communicate with devices on the wireless network without any additional configuration on the FortiGate unit, such as additional security policies.

A software switch can also be useful if you require more hardware ports for the switch on a FortiGate unit. For example, if your FortiGate unit has a 4-port switch, WAN1, WAN2, and DMZ interfaces, and you need one more port, you can create a soft switch that can include the four-port switch and the DMZ interface, all on the same subnet. These types of applications also apply to wireless interfaces, virtual wireless interfaces, and physical interfaces such as those in FortiWiFi and FortiAP units.

Similar to a hardware switch, a software switch functions like a single interface. It has one IP address, and all the interfaces in the software switch are on the same subnet. Traffic between devices connected to each interface is not regulated by security policies, and traffic passing in and out of the switch is controlled by the same policy.

When setting up a software switch, consider the following:

- Ensure that you have a back up of the configuration.
- Ensure that you have at least one port or connection, such as the console port, to connect to the FortiGate unit. If you accidentally combine too many ports, you need a way to undo errors.
- The ports that you include must not have any link or relation to any other aspect of the FortiGate unit, such as DHCP servers, security policies, and so on.
- Ensure the *Create address object matching subnet* option is disabled, if any port *Role* is set to either *LAN* or *DMZ*.
- For increased security, you can create a captive portal for the switch to allow only specific user groups access to the resources connected to the switch.

Some of the difference between software and hardware switches are:

Feature	Software switch	Hardware switch
Processing	Packets are processed in software by the CPU.	Packets are processed in hardware by the hardware switch controller, or SPU where applicable.
STP	Not Supported	Supported
Wireless SSIDs	Supported	Not Supported

Feature	Software switch	Hardware switch
Intra-switch traffic	Allowed by default. Can be explicitly set to require a policy.	Allowed by default.
Active-active HA load balancing	Not supported	Supported

To create a software switch in the GUI:

1. Go to *Network* > *Interfaces*.
2. Click *Create New* > *Interface*.
3. Set *Type* to *Software Switch*.
4. Configure the *Name*, *Interface members*, and other fields as required.
   To add an interface to a software switch, it cannot be referenced by an existing, configuration and its IP address must be set to 0.0.0.0/0.0.0.0.
5. Click *OK*.

To create a software switch in the CLI:

```
config system switch-interface
    edit <interface>
        set vdom <vdom>
        set member <interface_list>
        set type switch
    next
end
config system interface
    edit <interface>
        set vdom <vdom>
        set type switch
        set ip <ip_address>
        set allowaccess https ssh ping
    next
end
```

To add an interface to a software switch, it cannot be referenced by an existing configuration and its IP address must be set to 0.0.0.0/0.0.0.0.

Example

For this example, the wireless interface (WiFi) needs to be on the same subnet as the DMZ1 interface to facilitate wireless synchronization between an iPhone and a local computer. Because synchronization between two subnets is problematic, putting both interfaces on the same subnet allows synchronization to work. The software switch will accomplish this.

1. Clear the interfaces and back up the configuration:
   a. Ensure the interfaces are not used for other security policy or for other use on the FortiGate unit.
   b. Check the WiFi and DMZ1 ports to ensure that DHCP is disabled and that there are no other dependencies on these interfaces.
   c. Save the current configuration so that it can be recovered if something goes wrong.
2. Merge the WiFi port and DMZ1 port to create a software switch named `synchro` with an IP address of 10.10.21.12 and administrative access for HTTPS, SSH and PING:

```
config system switch-interface
    edit synchro
        set vdom "root"
        set type switch
        set member dmz1 wifi
    next
end
config system interface
    edit synchro
        set ip 10.10.21.12 255.255.255.0
        set allowaccess https ssh ping
    next
end
```

After the switch is set up, add security policies, DHCP servers, and any other settings that are required.

Hardware switch

A hardware switch is a virtual switch interface that groups different ports together so that the FortiGate can use the group as a single interface. Supported FortiGate models have a default hardware switch called either *internal* or *lan*. The hardware switch is supported by the chipset at the hardware level.

Ports that are connected to the same hardware switch behave like they are on the same physical switch in the same broadcast domain. Ports can be removed from a hardware switch and assigned to another switch or used as standalone interfaces.

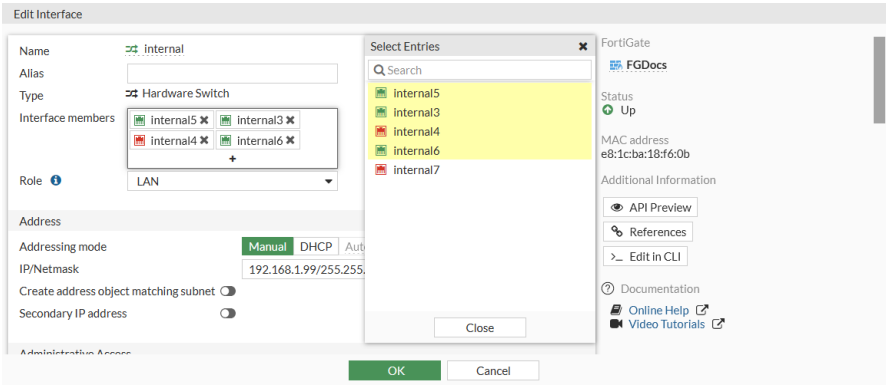
Some of the difference between hardware and software switches are:

Feature	Hardware switch	Software switch
Processing	Packets are processed in hardware by the hardware switch controller, or SPU where applicable.	Packets are processed in software by the CPU.
STP	Supported	Not Supported
802.1x	Supported on the following NP6 platforms: FG-30xE, FG-40xE, and FG-110xE	Supported on a virtual switch when added to a software switch
Wireless SSIDs	Not Supported	Supported
Intra-switch traffic	Allowed by default.	Allowed by default. Can be explicitly set to require a policy.

After ports are added to a virtual switch with STP or 802.1x enabled, you can enable or disable STP or 802.1x for each member port.

To change the ports in a hardware switch in the GUI:

1. Go to *Network* > *Interface* and edit the hardware switch.
2. Click inside the *Interface members* field.



3. Select interfaces to add or remove them from the hardware switch, then click *Close*.
   To add an interface to a hardware switch, it cannot be referenced by an existing configuration and its IP address must be set to 0.0.0.0/0.0.0.0.
4. Click *OK*.
   Removed interfaces will now be listed as standalone interfaces in the *Physical Interface* section.

To remove ports from a hardware switch in the CLI:

```
config system virtual-switch
    edit "internal"
        config port
            delete internal2
            delete internal7
            ...
        end
    next
end
```

To add ports to a hardware switch in the CLI:

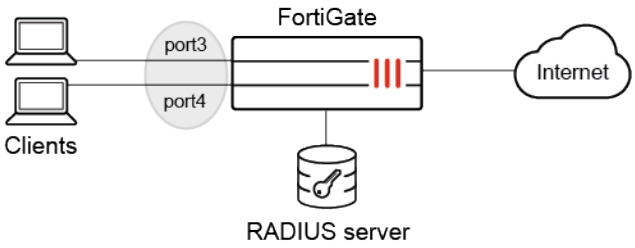
```
config system virtual-switch
    edit "internal"
        set physical-switch "sw0"
        config port
            edit "internal3"
            next
            edit "internal5"
            next
            edit "internal4"
            next
```

```
            edit "internal6"
            next
        end
    next
end
```

To add an interface to a hardware switch, it cannot be referenced by an existing configuration and its IP address must be set to 0.0.0.0/0.0.0.0.

Example of using 802.1X on virtual switches

In this example, port3 and port4 are part of a hardware switch interface. The hardware switch acts as a virtual switch so that devices can connect directly to these ports and perform 802.1X authentication on the port.

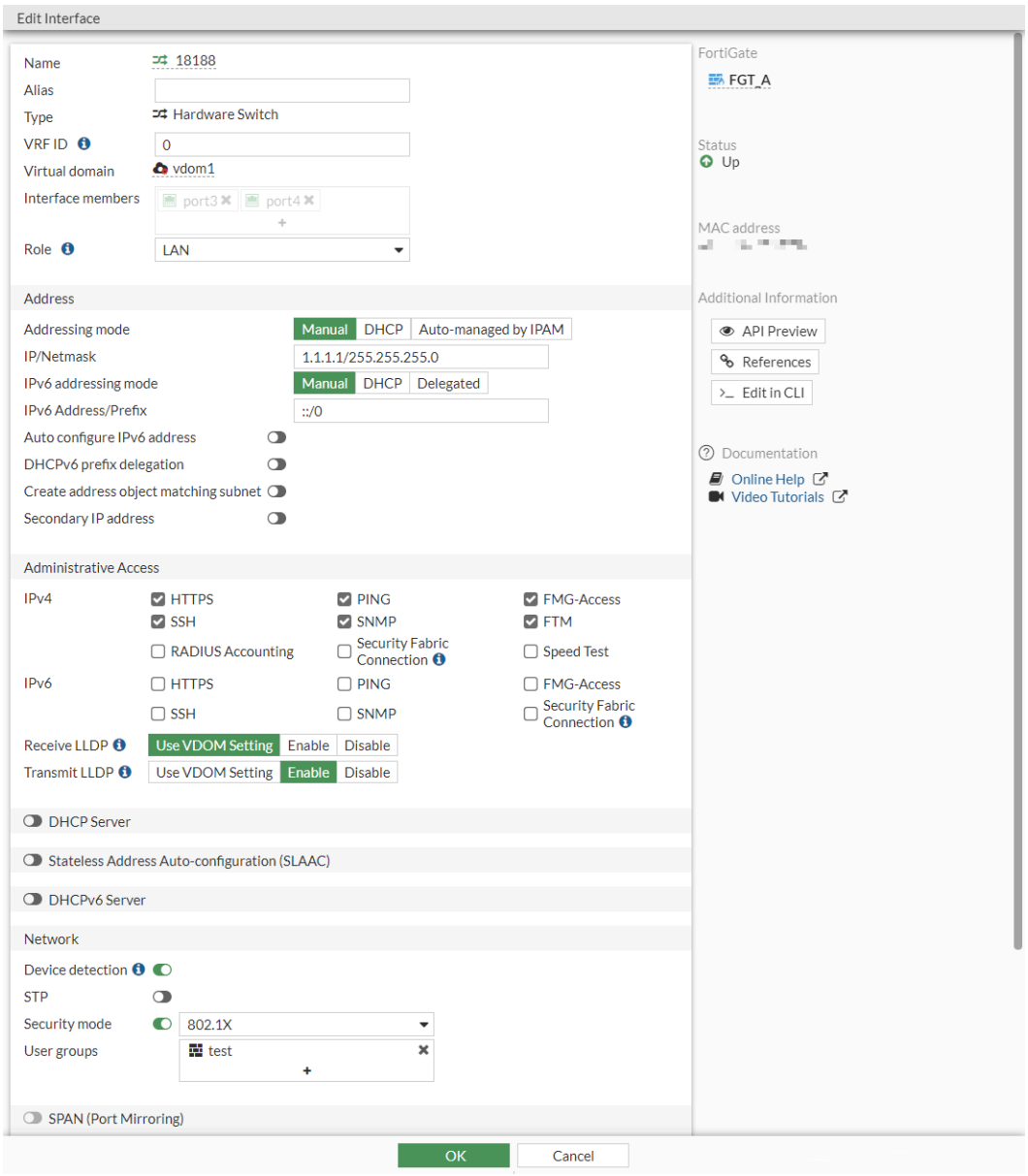


Prerequisites:

1. Configure a RADIUS server (see RADIUS servers on page 3527).
2. Define a user group named test to use the remote RADIUS server and for 802.1X authentication (see User definition, groups, and settings on page 3485).
3. Configure a hardware switch (named 18188) with port3 and port4 as the members.
4. Configure a firewall policy that allows traffic from the 18188 hardware switch to go to the internet.
5. Enable 802.1X authentication on the client devices.

To configure 802.1X authentication on a hardware switch in the GUI:

1. Go to *Network* > *Interfaces* and edit the hardware switch.
2. In the *Network* section, enable *Security mode* and select *802.1X*.
3. Click the *+* to add the *User group*.



4. Click *OK*.

To configure 802.1X authentication on a hardware switch in the CLI:

1. Configure the virtual hardware switch interfaces:

```
config system virtual-switch
    edit "18188"
        set physical-switch "sw0"
        config port
            edit "port3"
            next
            edit "port4"
            next
```

```
        end
    next
end
```

2. Configure 802.1X authentication:

```
config system interface
    edit "18188"
        set vdom "vdom1"
        set ip 1.1.1.1 255.255.255.0
        set allowaccess ping https ssh snmp fgfm ftm
        set type hard-switch
        set security-mode 802.1X
        set security-groups "test"
        set device-identification enable
        set lldp-transmission enable
        set role lan
        set snmp-index 52
    next
end
```

To verify the that the 802.1X authentication was successful:

1. Get a client connected to port3 to authenticate to access the internet.
2. In FortiOS, verify the 802.1X authentication port status:

```
# diagnose sys 802-1x status

Virtual switch '18188' (default mode) 802.1x member status:
  port3: Link up, 802.1X state: authorized
  port4: Link up, 802.1X state: unauthorized
```

Example of using 802.1X on a virtual switch when added to a software switch

802.1X authentication can be enabled under a virtual switch when it's added to a software switch. Customers can use firewall policies to regulate traffic within an 802.1X dynamic VLAN when setting software switch's `intra-switch-policy` to `explicit`.

To enable 802.1X authentication on a virtual switch:

1. From the FortiGate, create a hardware switch.

```
config system virtual-switch
    edit "trust"
        set physical-switch "sw0"
        config port
            edit "port5"
            next
```

```
        end
    next
    edit "vlan0200"
        set physical-switch "sw0"
    next
end
```

2. Create a software switch with `intra-switch-policy` set to `explicit`.

```
config system switch-interface
    edit "SW-switch"
        set vdom "vdom1"
        set member "port3" "trust" "vlan0200"
        set intra-switch-policy explicit
    next
end
```

3. Add the hardware switch you created as a member of the FortiGate.

```
config system interface
    edit "SW-switch"
        set vdom "vdom1"
        set ip 6.6.6.1 255.255.255.0
        set allowaccess ping https ssh snmp fgfm ftm
        set type switch
        set snmp-index 22
    next
end
```

4. Enable 802.1x authentication for the hardware switch.

```
config system interface
    edit "trust"
        set vdom "vdom1"
        set type hard-switch
        set security-mode 802.1X
        set security-8021x-mode dynamic-vlan
        set security-groups "group_radius"
        set snmp-index 20
    next
end

config system interface
    edit "vlan0200"
        set vdom "vdom1"
        set type hard-switch
        set security-mode 802.1X
        set security-8021x-mode slave
        set security-8021x-master "trust"
        set security-8021x-dynamic-vlan-id 200
        set snmp-index 21
```

```
    next
end
```

5. Verify that 802.1x is enabled and the port state is `unauthorized`.

```
# diagnose sys 802-1x status

Virtual switch 'vlan0200' (slave mode) 802.1x member status:

Virtual switch 'trust' (dynamic vlan mode) 802.1x member status:
  port5: Link up, 802.1X state: unauthorized
```

6. Configure a firewall policy to allow traffic between the software switch members.

```
config firewall policy
    edit 1
        set srcintf "vlan0200"
        set dstintf "port3"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
    next
end
```

7. Verify that a client device connected to the hardware switch cannot reach other software switch members without authentication.

```
fos@switch_pc01:~$ ping 6.6.6.3 -c 3
PING 6.6.6.3 (6.6.6.3): 56 data bytes
92 bytes from switch_pc01 (6.6.6.16): Destination Host Unreachable
92 bytes from switch_pc01 (6.6.6.16): Destination Host Unreachable
92 bytes from switch_pc01 (6.6.6.16): Destination Host Unreachable
--- 6.6.6.3 ping statistics ---
3 packets transmitted, 0 packets received, 100% packet loss
```

8. Perform 802.1x authentication on a client device.

```
EAPOL: Supplicant port status: Authorized
EAPOL: SUPP_BE entering state RECEIVE
EAPOL: SUPP_BE entering state SUCCESS
EAPOL: SUPP_BE entering state IDLE
EAPOL authentication completed - result=SUCCESS
```

9. After successful authentication, verify the port is authorized and moved to secondary switch (`slave mode`) on the FortiGate.

```
A (vdom1) (Interim)# diagnose sys 802-1x status

Virtual switch 'vlan0200' (slave mode) 802.1x member status:
  port5: Link up, 802.1X state: authorized
```

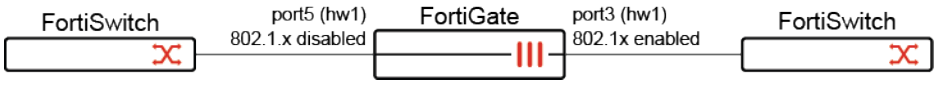
```
Virtual switch 'trust' (dynamic vlan mode) 802.1x member status:
```

10. The client device can now reach other software switch members.

```
fos@switch_pc01:~$ ping 6.6.6.3 -c 3
PING 6.6.6.3 (6.6.6.3): 56 data bytes
64 bytes from 6.6.6.3: icmp_seq=0 ttl=255 time=0.310 ms
64 bytes from 6.6.6.3: icmp_seq=1 ttl=255 time=0.297 ms
64 bytes from 6.6.6.3: icmp_seq=2 ttl=255 time=0.192 ms
--- 6.6.6.3 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.192/0.266/0.310/0.053 ms
```

Example of disabling 802.1x on one port

In this example, FortiGate is connected to two switches, and a virtual switch named hw1 is configured with two port members: port3 and port5. 802.1x authentication is enabled for port3 and disabled for port5.



To configure 802.1x authentication for individual ports:

1. Configure a virtual switch to use port3 and port5:

```
config system virtual-switch
    edit "hw1"
        set physical-switch "sw0"
        config port
            edit "port3"
            next
            edit "port5"
            next
        end
    next
end
```

2. Enable 802.1x authentication for the virtual switch:

```
config system interface
   edit "hw1"
        set vdom "vdom1"
        set ip 6.6.6.1 255.255.255.0
        set allowaccess ping https ssh
        set type hard-switch
        set security-mode 802.1X
        set security-groups "group_radius"
        set device-identification enable
        set lldp-transmission enable
```

```
        set role lan
        set snmp-index 55
        set ip-managed-by-fortiipam disable
    next
end
```

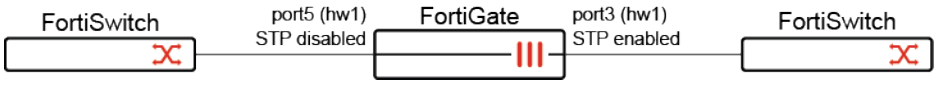
3. Disable 802.1x authentication on port5:

```
config system interface
    edit "port5"
        set vdom "vdom1"
        set type physical
        set security-8021x-member-mode disable
        set snmp-index 9
    next
end
```

802.1x authentication is disabled on port5 and remains enabled on port3.

Example of disabling STP on one port

In this example, FortiGate is connected to two switches, and a virtual switch named hw1 is configured with two port members: port3 and port5. STP is enabled for port3 and disabled for port5. Any STP sent to port5 is silently ignored. Port3 remains enabled for STP.



To configure STP for individual ports:

1. Configure a virtual switch to use port3 and port5:

```
config system virtual-switch
    edit "hw1"
        set physical-switch "sw0"
        config port
            edit "port3"
            next
            edit "port5"
            next
        end
    next
end
```

2. Enable STP for the virtual switch:

```
config system interface
    edit "hw1"
        set vdom "vdom1"
        set ip 6.6.6.1 255.255.255.0
        set allowaccess ping https ssh
```

```
        set type hard-switch
        set stp enable
        set device-identification enable
        set lldp-transmission enable
        set role lan
        set snmp-index 55
        set ip-managed-by-fortiipam disable
    next
end
```

3. Disable STP on port5 by enabling it as an STP edge port:

```
config system interface
    edit "port5"
        set vdom "vdom1"
        set type physical
        set stp-edge enable
        set snmp-index 9
    next
end
```

Port5 is enabled as an edge port with STP disabled. Port3 remains enabled for STP.

Zone

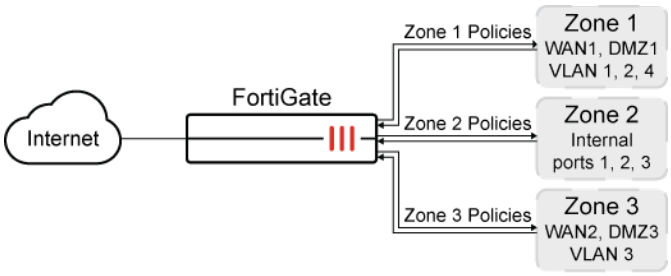
Zones are a group of one or more physical or virtual FortiGate interfaces that you can apply firewall policies to for controlling inbound and outbound traffic. Grouping interfaces and VLAN subinterfaces into zones simplifies creating firewall policies where a number of network segments can use the same policy settings and protection profiles.

When you add a zone, you select the names of the interfaces and VLAN subinterfaces to add to the zone. Each interface still has its own address. Routing is still done between interfaces, that is, routing is not affected by zones. You can use firewall policies to control the flow of intra-zone traffic.

For example, in the sample configuration below, the network includes three separate groups of users representing different entities on the company network. While each group has its own set of ports and VLANs in each area, they can all use the same firewall policy and protection profiles to access the Internet. Rather than the administrator making nine separate firewall policies, he can make administration simpler by adding the required interfaces to a zone and creating three policies.

Example configuration

You can configure policies for connections to and from a zone but not between interfaces in a zone. For this example, you can create a firewall policy to go between zone 1 and zone 3, but not between WAN2 and WAN1, or WAN1 and DMZ1.



To create a zone in the GUI:

1. Go to *Network* > *Interfaces*.

   If VDOMs are enabled, go to the VDOM to create a zone.

2. Click *Create New* > *Zone*.
3. Configure the *Name* and add the *Interface Members*.
4. Enable or disable *Block intra-zone traffic* as required.
5. Click *OK*.

To configure a zone to include the interfaces WAN1, DMZ1, VLAN1, VLAN2 and VLAN4 using the CLI:

```
config system zone
    edit zone_1
        set interface WAN1 DMZ1 VLAN1 VLAN2 VLAN4
        set intrazone {deny | allow}
    next
end
```

Using zone in a firewall policy

To configure a firewall policy to allow any interface to access the Internet using the CLI:

```
config firewall policy
    edit 2
        set name "2"
        set srcintf "zone_1"
        set dstintf "port15"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

Intra-zone traffic

In the zone configuration you can `set intrazone deny` to prohibit the different interfaces in the same zone to talk to each other.

For example, if you have ten interfaces in your zone and the `intrazone` setting is `deny`. You now want to allow traffic between a very small number of networks on different interfaces that are part of the zone but you do not want to disable the intra-zone blocking.

In this example, the zone VLANs are defined as: `192.168.1.0/24`, `192.168.2.0/24`, ... `192.168.10.0/24`.

This policy allows traffic from 192.168.1.x to 192.168.2.x even though they are in the same zone and intra-zone blocking is enabled. The intra-zone blocking acts as a default deny rule and you have to specifically override it by creating a policy within the zone.

To enable intra-zone traffic, create the following policy:

Source Interface	Zone-name, e.g., `Vlans`
Source Address	192.168.1.0/24
Destination	Zone-name (same as Source Interface, i.e., `Vlans`)
Destination Address	192.168.2.0/24

Virtual wire pair

A virtual wire pair consists of two interfaces that do not have IP addressing and are treated like a transparent mode VDOM. All traffic received by one interface in the virtual wire pair can only be forwarded to the other interface, provided a virtual wire pair firewall policy allows this traffic. Traffic from other interfaces cannot be routed to the interfaces in a virtual wire pair. Redundant and 802.3ad aggregate (LACP) interfaces can be included in a virtual wire pair.

Virtual wire pairs are useful for a typical topology where MAC addresses do not behave normally. For example, port pairing can be used in a Direct Server Return (DSR) topology where the response MAC address pair may not match the request's MAC address pair.

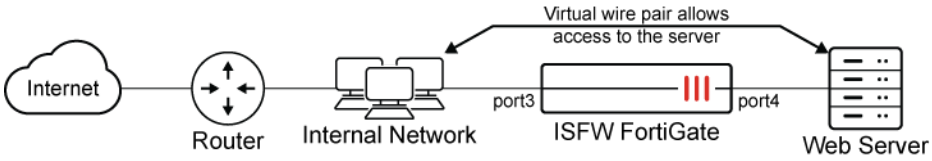
When creating a new virtual wire pair, the *Interface members* field displays interfaces without assigned addresses. Interfaces with assigned addresses are not displayed.

Therefore, you cannot add to a virtual wire pair an interface with *Addressing mode* set to *DHCP*. If you change the interface settings to *Manual* with *IP/Netmask* set to *0.0.0.0/0.0.0.0*, you can add the interface to a virtual wire pair.

Example

In this example, a virtual wire pair (port3 and port4) makes it easier to protect a web server that is behind a FortiGate operating as an Internal Segmentation Firewall (ISFW). Users on the internal network access the web server through the ISFW over the virtual wire pair.

Interfaces used in a virtual wire pair cannot be used to access the ISFW FortiGate. Before creating a virtual wire pair, make sure you have a different port configured to allow admin access using your preferred protocol.



To add a virtual wire pair using the GUI:

1. Go to *Network* > *Interfaces*.
2. Click *Create New* > *Virtual Wire Pair*.
3. Enter a name for the virtual wire pair.
4. Select the *Interface Members* to add to the virtual wire pair (*port3* and *port 4*).
   These interfaces cannot be part of a switch, such as the default LAN/internal interface.
5. If required, enable *Wildcard VLAN* and set the *VLAN Filter*.
6. Click *OK*.

To add a virtual wire pair using the CLI:

```
config system virtual-wire-pair
    edit "VWP-name"
        set member "port3" "port4"
        set wildcard-vlan disable
    next
end
```

To create a virtual wire pair policy using the GUI:

1. Go to *Policy & Objects* > *Firewall Virtual Wire Pair Policy*.
2. Click *Create New*.
3. In the *Virtual Wire Pair* field, click the + to add the virtual wire pair.
4. Select the direction (arrows) that traffic is allowed to flow.
5. Configure the other settings as needed.
6. Click *OK*.

To create a virtual wire pair policy using the CLI:

```
config firewall policy
    edit 1
        set name "VWP-Policy"
        set srcintf "port3" "port4"
        set dstintf "port3" "port4"
        set srcaddr "all"
        set dstaddr "all"
```

```
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set fsso disable
    next
end
```

Configuring multiple virtual wire pairs in a virtual wire pair policy

You can create a virtual wire pair policy that includes different virtual wire pairs in NGFW profile and policy mode. This reduces overhead to create multiple similar policies for each VWP. In NGFW policy mode, multiple virtual wire pairs can be configured in a *Security Virtual Wire Pair Policy* and *Virtual Wire Pair SSL Inspection & Authentication* policy.

The virtual wire pair settings must have wildcard VLAN enabled. When configuring a policy in the CLI, the virtual wire pair members must be entered in `srcintf` and `dstintf` as pairs.

To configure multiple virtual wire pairs in a policy in the GUI:

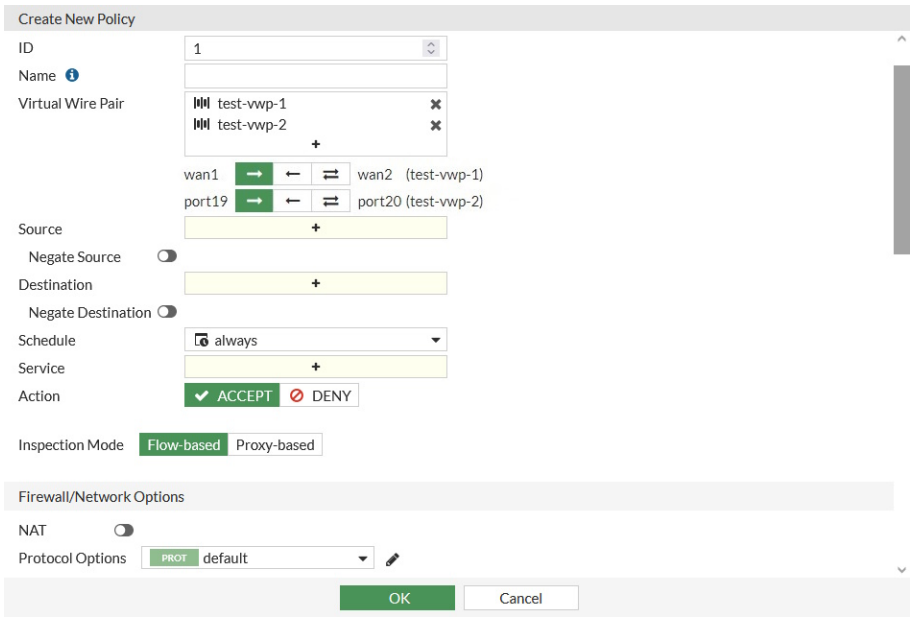
1. Configure the virtual wire pairs:
   a. Go to *Network* > *Interfaces* and click *Create New* > *Virtual Wire Pair*.
   b. Create a pair with the following settings:

Name	test-vwp-1
Interface members	wan1, wan2
Wildcard VLAN	Enable

   c. Click *OK*.
   d. Click *Create New* > *Virtual Wire Pair* and create another pair with the following settings:

Name	test-vwp-2
Interface members	port19, port20
Wildcard VLAN	Enable

   e. Click *OK*.
2. Configure the policy:
   a. Go to *Policy & Objects* > *Firewall Virtual Wire Pair Policy* and click *Create New*.
   b. In the *Virtual Wire Pair* field, click the *+* to add *test-vwp-1* and *test-vwp-2*. Select the direction for each of the selected virtual wire pairs.



c. Configure the other settings as needed.

d. Click *OK*.

To configure multiple virtual wire pairs in a policy in the CLI:

1. Configure the virtual wire pairs:

```
config system virtual-wire-pair
    edit "test-vwp-1"
        set member "wan1" "wan2"
        set wildcard-vlan enable
    next
    edit "test-vwp-2"
        set member "port19" "port20"
        set wildcard-vlan enable
    next
end
```

2. Configure the policy:

```
config firewall policy
    edit 1
        set name "vwp1&2-policy"
        set srcintf "port19" "wan1"
        set dstintf "port20" "wan2"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic all
    next
end
```

PRP handling in NAT mode with virtual wire pair

PRP (Parallel Redundancy Protocol) is supported in NAT mode for a virtual wire pair. This preserves the PRP RCT (redundancy control trailer) while the packet is processed by the FortiGate.

To configure PRP handling on a device in NAT mode:

1. Enable PRP in the VDOM settings:

```
(root) # config system settings
    set prp-trailer-action enable
end
```

2. Enable PRP in the NPU attributes:

```
(global) # config system npu
    set prp-port-in "port15"
    set prp-port-out "port16"
end
```

3. Configure the virtual wire pair:

```
(root) # config system virtual-wire-pair
    edit "test-vwp-1"
        set member "port15" "port16"
    next
end
```

Using VLAN sub-interfaces in virtual wire pairs

VLAN sub-interfaces, such as regular 802.1Q and 802.1ad (QinQ), are allowed to be members of a virtual wire pair.

Example

In this example, the FortiGate has two VLAN interfaces. The first interface is a QinQ (802.1ad) interface over the physical interface port3. The second interface is a basic 802.1Q VLAN interface over physical interface port5. These two interfaces are grouped in a virtual wire pair so that bi-directional traffic is allowed. This example demonstrates ICMP from the client (3.3.3.4) sent to the server (3.3.3.1).

To configure VLAN sub-interfaces in a virtual wire pair:

1. Configure the QinQ interfaces:

```
config system interface
    edit "8021ad-port3"
        set vdom "vdom1"
        set vlan-protocol 8021ad
        set device-identification enable
        set role lan
        set snmp-index 31
        set interface "port3"
        set vlanid 3
    next
    edit "8021Q"
        set vdom "vdom1"
        set device-identification enable
        set role lan
        set snmp-index 32
        set interface "8021ad-port3"
        set vlanid 33
    next
end
```

2. Configure the 802.1Q interface:

```
config system interface
    edit "8021q-port5"
        set vdom "vdom1"
        set device-identification enable
        set role lan
        set snmp-index 33
        set interface "port5"
        set vlanid 5
    next
end
```

3. Configure the virtual wire pair:

```
config system virtual-wire-pair
    edit "VWP1"
        set member "8021Q" "8021q-port5"
    next
end
```

4. Configure the firewall policy:

```
config firewall policy
    edit 1
        set name "1"
        set srcintf "8021Q" "8021q-port5"
        set dstintf "8021Q" "8021q-port5"
        set action accept
```

```
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
    next
end
```

To verify that bi-directional traffic passes through the FortiGate:

```
# diagnose sys session filter policy  1
# diagnose sys session list

session info: proto=1 proto_state=00 duration=18 expire=42 timeout=0 flags=00000000 socktype=0
sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=may_dirty br npu
statistic(bytes/packets/allow_err): org=168/2/1 reply=168/2/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=56->55/55->56 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 3.3.3.4:3072->3.3.3.1:8(0.0.0.0:0)
hook=post dir=reply act=noop 3.3.3.1:3072->3.3.3.4:0(0.0.0.0:0)
src_mac=08:5b:0e:71:bf:c6  dst_mac=d4:76:a0:5d:b2:de
misc=0 policy_id=1 pol_uuid_idx=534 auth_info=0 chk_client_info=0 vd=3
serial=00005f6c tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000c00 ofld-O ofld-R
npu info: flag=0x81/0x81, offload=8/8, ips_offload=0/0, epid=187/156, ipid=156/187,
vlan=0x0005/0x0021
vlifid=156/187, vtag_in=0x0005/0x0021 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=0/5
total session 1
```

DVLAN QinQ on NP7 platforms over virtual wire pairs

DVLAN 802.1ad and 802.1Q modes are supported on NP7 platforms over virtual wire pairs, which provides better performance and packet processing.

The default DVLAN mode is 802.1ad, but the DVLAN mode can be changed using `diagnose npu np7 dvlan-mode <dvlan_mode> {<npid> | all}`. The DVLAN mode can be applied to a specific NPID or all NPIDs. For example:

- `diagnose npu np7 dvlan-mode 802.1AD 0` will set NP0 to work in 802.1ad mode.
- `diagnose npu np7 dvlan-mode 802.1Q all` will set all NPUs to work in 802.1Q mode.

> ⚠ A reboot is required for custom DVLAN settings to take effect. To avoid any inconveniences or disruptions, changing the DVLAN settings should be done during a scheduled downtime or maintenance window.
>
> The DVLAN mode should only be changed if you are solely using the virtual wire pair (VWP) and are seeking to enhance performance. Enabling this feature may impact VLAN interfaces within your network.

In the virtual wire pair settings, the `outer-vlan-id` can be set. This is the same value as the outer provider-tag (S-Tag).

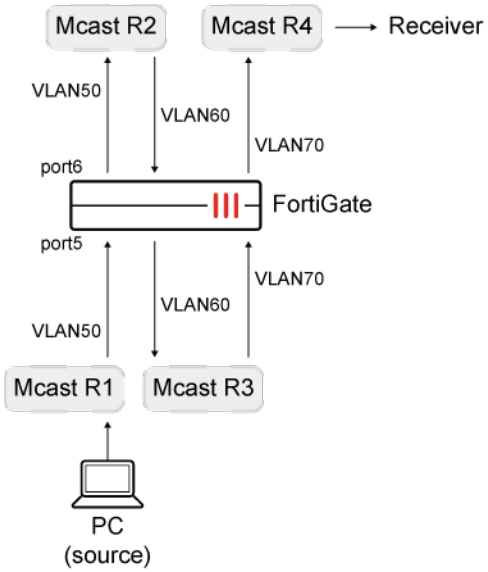
To configure the outer VLAN ID:

```
config system virtual-wire-pair
    edit "dvlan-test"
        set member "port33" "port34"
        set wildcard-vlan enable
        set outer-vlan-id 1234
    next
end
```

Enhanced multicast session key for virtual wire pair NEW

FortiOS now adds ingress interface and VLAN ID as additional session keys for multicast traffic in virtual wire pair (VWP) configurations with wildcard-vlan enabled. This enhancement enables the FortiGate to create distinct multicast sessions for traffic returning through the same VWP on different VLANs, improving accuracy and ensuring optimal session handling.

Example



In this example, a VWP with wildcard-vlan enabled is created. A firewall policy and multicast policy are created. Multicast traffic goes through FortiGate multiple times in different directions, and distinct multicast sessions are created for traffic returning through the same VWP on different VLANs.

To configure:

1. Configure a virtual wire pair:
   The VWP contains members a and b, and *wildcard-vlan* is enabled.

```
config system virtual-wire-pair
    edit "vwp"
        set member "a" "b"
        set wildcard-vlan enable
    next
end
```

2. Configure a firewall policy:
   The firewall policy specifies VWP members a and b as source and destination interfaces.

```
config firewall policy
    edit 1
        set uuid dfb4861a-f6ff-51f0-888b-682470ce488d
        set srcintf "a" "b"
        set dstintf "b" "a"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
    next
end
```

3. Configure a multicast policy:

```
config firewall multicast-policy
    edit 1
        set uuid 9d6c29a4-f720-51f0-f60b-07a30ec0aaac
        set srcintf "any"
        set dstintf "any"
        set srcaddr "all"
        set dstaddr "all"
        set logtraffic all
    next
end
```

4. After multicast traffic flows, view the different multicast sessions in traffic logs:

```
1: date=2026-01-21 time=15:38:42 eventtime=1769038717948452847 tz="-0800" logid="0002000012"
type="traffic" subtype="multicast" level="notice" vd="root" srcip=10.1.1.1 identifier=13
srcintf="b" srcintfrole="undefined" dstip=225.1.1.1 dstintf="a" dstintfrole="undefined"
vlan=60 srccountry="Reserved" dstcountry="Reserved" sessionid=4089 proto=1 action="accept"
policyid=1 policytype="multicast-policy" poluuid="9d6c29a4-f720-51f0-f60b-07a30ec0aaac"
```

```
service="PING" trandisp="noop" appcat="unscanned" duration=181 sentbyte=168 rcvdbyte=0
sentpkt=2 rcvdpkt=0

2: date=2026-01-21 time=15:38:42 eventtime=1769038717948449960 tz="-0800" logid="0002000012"
type="traffic" subtype="multicast" level="notice" vd="root" srcip=10.1.1.1 identifier=13
srcintf="a" srcintfrole="undefined" dstip=225.1.1.1 dstintf="b" dstintfrole="undefined"
vlan=50 srccountry="Reserved" dstcountry="Reserved" sessionid=4088 proto=1 action="accept"
policyid=1 policytype="multicast-policy" poluuid="9d6c29a4-f720-51f0-f60b-07a30ec0aaac"
service="PING" trandisp="noop" appcat="unscanned" duration=181 sentbyte=168 rcvdbyte=0
sentpkt=2 rcvdpkt=0

3: date=2026-01-21 time=15:38:42 eventtime=1769038717940450742 tz="-0800" logid="0002000012"
type="traffic" subtype="multicast" level="notice" vd="root" srcip=10.1.1.1 identifier=13
srcintf="a" srcintfrole="undefined" dstip=225.1.1.1 dstintf="b" dstintfrole="undefined"
vlan=70 srccountry="Reserved" dstcountry="Reserved" sessionid=4090 proto=1 action="accept"
policyid=1 policytype="multicast-policy" poluuid="9d6c29a4-f720-51f0-f60b-07a30ec0aaac"
service="PING" trandisp="noop" appcat="unscanned" duration=181 sentbyte=168 rcvdbyte=0
sentpkt=2 rcvdpkt=0
```

5. View the different multicast sessions using the `diagnose sys mcast-session list` command:

```
# diagnose sys mcast-session list

session info: id=4264 vf=0 vrf=0 proto=1 10.1.1.1.16->225.1.1.1.8
used=2 path=1 duration=167 expire=13 indev=11 vlanid=50 pkts=2 bytes=168
state=0000001a: tp npu-cap offloaded
session-npu-info: ipid/vlifid=9/9 vlanid/vtag_in=50/50 in_npuid=1 tae_index=0 qid=1 fwd_
map=0x00000000
path: log offloaded policy=1, outdev=12, tos=0xff
act-npu-info:  ipid/vlifid=9/9 vlanid/vtag_in=50/50 in_npu_id=1, out_npuid=1 epid=10 fwd=0

session info: id=4265 vf=0 vrf=0 proto=1 10.1.1.1.16->225.1.1.1.8
used=2 path=1 duration=167 expire=13 indev=12 vlanid=60 pkts=2 bytes=168
state=0000001a: tp npu-cap offloaded
session-npu-info: ipid/vlifid=10/10 vlanid/vtag_in=60/60 in_npuid=1 tae_index=0 qid=2 fwd_
map=0x00000000
path: log offloaded policy=1, outdev=11, tos=0xff
act-npu-info:  ipid/vlifid=10/10 vlanid/vtag_in=60/60 in_npu_id=1, out_npuid=1 epid=9 fwd=0

session info: id=4266 vf=0 vrf=0 proto=1 10.1.1.1.16->225.1.1.1.8
used=2 path=1 duration=167 expire=13 indev=11 vlanid=70 pkts=2 bytes=168
state=0000001a: tp npu-cap offloaded
session-npu-info: ipid/vlifid=9/9 vlanid/vtag_in=70/70 in_npuid=1 tae_index=0 qid=7 fwd_
map=0x00000000
path: log offloaded policy=1, outdev=12, tos=0xff
act-npu-info:  ipid/vlifid=9/9 vlanid/vtag_in=70/70 in_npu_id=1, out_npuid=1 epid=10 fwd=0
Total 3 sessions
```

Enhanced MAC VLAN

The Media Access Control (MAC) Virtual Local Area Network (VLAN) feature in Linux allows you to configure multiple virtual interfaces with different MAC addresses (and therefore different IP addresses) on a physical interface.

FortiGate implements an enhanced MAC VLAN consisting of a MAC VLAN with bridge functionality. Because each MAC VLAN has a unique MAC address, virtual IP addresses (VIPs) and IP pools are supported, and you can disable Source Network Address Translation (SNAT) in policies.

MAC VLAN cannot be used in a transparent mode virtual domain (VDOM). In a transparent mode VDOM, a packet leaves an interface with the MAC address of the original source instead of the interface's MAC address. FortiGate implements an enhanced version of MAC VLAN where it adds a MAC table in the MAC VLAN which learns the MAC addresses when traffic passes through.

If you configure a VLAN ID for an enhanced MAC VLAN, it won't join the switch of the underlying interface. When a packet is sent to this interface, a VLAN tag is inserted in the packet and the packet is sent to the driver of the underlying interface. When the underlying interface receives a packet, if the VLAN ID doesn't match, it won't deliver the packet to this enhanced MAC VLAN interface.

When using a VLAN ID, the ID and the underlying interface must be a unique pair, even if the belong to different VDOMs. This is because the underlying, physical interface uses the VLAN ID as the identifier to dispatch traffic among the VLAN and enhanced MAC VLAN interfaces.

If you use an interface in an enhanced MAC VLAN, do not use it for other purposes such as a management interface, HA heartbeat interface, or in Transparent VDOMs.

If a physical interface is used by an EMAC VLAN interface, you cannot use it in a Virtual Wire Pair.

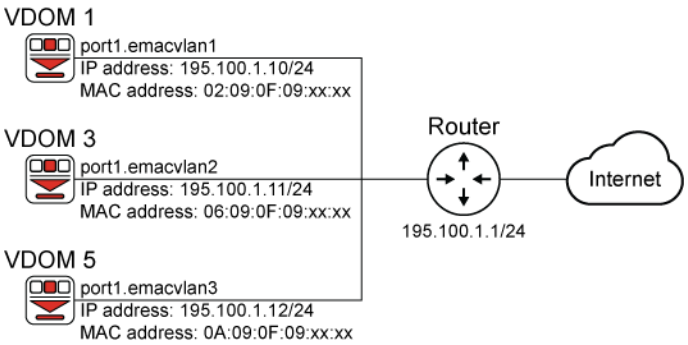
In high availability (HA) configurations, enhanced MAC VLAN is treated as a physical interface. It's assigned a unique physical interface ID and the MAC table is synchronized with the secondary devices in the same HA cluster.

In HA configurations, FortiGate assigns a virtual MAC to each interface. Virtual interfaces, such as EMAC VLAN interfaces with underlying NPU VLINK interface, are an exception and do not get assigned virtual MAC addresses.

Example 1: Enhanced MAC VLAN configuration for multiple VDOMs that use the same interface or VLAN

In this example, a FortiGate is connected, through port 1 to a router that's connected to the Internet. Three VDOMs share the same interface (port 1) which connects to the same router that's connected to the Internet. Three enhanced MAC VLAN interfaces are configured on port 1 for the three VDOMs. The enhanced MAC VLAN interfaces are in the same IP subnet segment and each have unique MAC addresses.

The underlying interface (port 1) can be a physical interface, an aggregate interface, or a VLAN interface on a physical or aggregate interface.



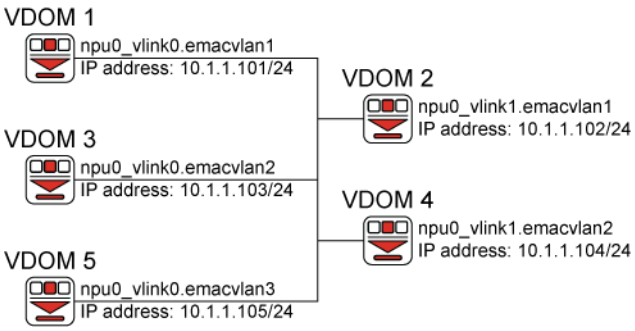
To configure enhanced MAC VLAN for this example in the CLI:

```
config system interface
    edit port1.emacvlan1
        set vdom VDOM1
        set type emac-vlan
        set interface port1
    next
    edit port 1.emacvlan2
        set vdom VDOM2
        set type emac-vlan
        set interface port1
    next
    edit port1.emacvlan3
        set vdom VDOM3
        set type emac-vlan
        set interface port1
    next
end
```

Example 2: Enhanced MAC VLAN configuration for shared VDOM links among multiple VDOMs

In this example, multiple VDOMs can connect to each other using enhanced MAC VLAN on network processing unit (NPU) virtual link (Vlink) interfaces.

FortiGate VDOM links (NPU-Vlink) are designed to be peer-to-peer connections and VLAN interfaces on NPU Vlink ports use the same MAC address. Connecting more than two VDOMs using NPU Vlinks and VLAN interfaces is not recommended as the VLAN interfaces share the same MAC address. To avoid overlapping MAC addresses on the same NPU Vlink, use EMAC VLANs instead.



To configure enhanced MAC VLAN for this example in the CLI:

```
config system interface
    edit npu0_vlink0.emacvlan1
        set vdom VDOM1
        set type emac-vlan
        set interface npu0_vlink0
    next
    edit npu0_vlink0.emacvlan2
        set vdom VDOM3
        set type emac-vlan
        set interface npu0_vlink0
    next
    edit npu0_vlink1.emacvlan1
        set vdom VDOM2
        set type emac-vlan
        set interface npu0_vlink1
    next
end
```

Example 3: Enhanced MAC VLAN configuration for unique MAC addresses for each VLAN interface on the same physical port

Some networks require a unique MAC address for each VLAN interface when the VLAN interfaces share the same physical port. In this case, the enhanced MAC VLAN interface is used the same way as normal VLAN interfaces.

To configure this, use the `set vlanid` command for the VLAN tag. The VLAN ID and interface must be a unique pair, even if they belong to different VDOMs.

To configure enhanced MAC VLAN:

```
config system interface
    edit <interface-name>
        set type emac-vlan
        set vlanid <VLAN-ID>
        set interface <physical-interface>
```

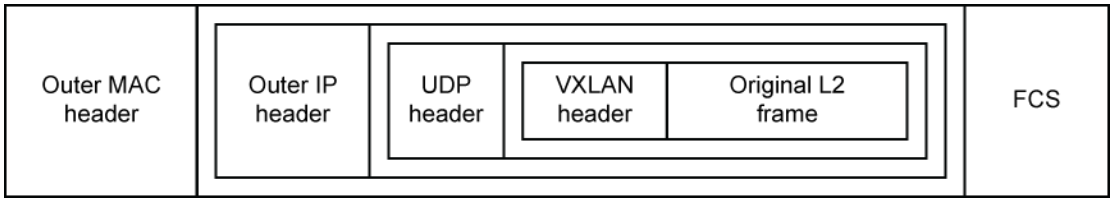
```
    next
end
```

FortiGate supports a maximum of 512 EMAC VLAN interfaces per underlying interface, and a maximum of 600 MAC addresses including EMAC VLAN interfaces.

VXLAN

Virtual Extensible LAN (VXLAN) is a network virtualization technology used in large cloud computing deployments. It encapsulates layer 2 Ethernet frames within layer 3 IP packets using the UDP transport protocol on port 4789. VXLAN endpoints that terminate VXLAN tunnels can be virtual or physical switch ports, and are known as VXLAN tunnel endpoints (VTEPs).

Sample VXLAN packet



A VXLAN packet encapsulation occurs by first inserting a VXLAN header in front of the original layer 2 frame. This VXLAN header uses 3 B for the VNID that is used to identify the VXLAN segment, meaning that there are 16,777,215 different possible VNIDs. This allows for more unique LAN segments than possible VLANs. The original frame and the VXLAN header are then encapsulated into the UDP payload. The outer IP header allows it to be routed and transported over a layer 3 network, thus providing a layer 2 overlay scheme over a layer 3 network.

This equates to 50 B of overhead over the original frame: 14 B (Ethernet) + 20 B (IPv4) + 8 B (UDP) + 8 B (VXLAN headers). Since fragmenting a VXLAN packet is not recommended, it is advisable to increase the MTU size to 1550 B or above if possible, or to decrease the TCP MSS size inside a firewall policy.

For more information about VXLAN, see RFC 7348.

The following topics provide information about VXLAN:

- General VXLAN configuration and topologies on page 305
- VLAN inside VXLAN on page 309
- Virtual wire pair with VXLAN on page 311
- VXLAN over IPsec tunnel with virtual wire pair on page 313
- VXLAN over IPsec using a VXLAN tunnel endpoint on page 317
- VXLAN with MP-BGP EVPN on page 323
- EVPN support for VXLAN Anycast gateway and type-5 routing New on page 336
- VXLAN troubleshooting on page 352

General VXLAN configuration and topologies

This topic describes general VXLAN configurations and commonly used topologies. In the most basic configuration, a FortiGate is configured as a VXLAN tunnel endpoint (VTEP).

To configure a FortiGate as a VTEP:

1. Configure the local interface:

```
config system vxlan
    edit <name>
        set interface <string>
        set vni <integer>
        set ip-version {ipv4-unicast | ipv6-unicast | ipv4-multicast | ipv6-multicast}
        set dstport <integer>
        set remote-ip <IP_address>
        set remote-ip6 <IP_address>
    next
end
```

`interface <string>`	Set the local outgoing interface for the VXLAN encapsulated traffic.
`vni <integer>`	Set the VXLAN network ID.
`ip-version {ipv4-unicast | ipv6-unicast | ipv4-multicast | ipv6-multicast}`	Set the IP version to use for the VXLAN device and communication over VXLAN (default = ipv4-unicast).
`dstport <integer>`	Set the VXLAN destination port (default = 4789).
`remote-ip <IP_address>`	Set the IPv4 address of the remote VXLAN endpoint.
`remote-ip6 <IP_address>`	Set the IPv6 address of the remote VXLAN endpoint.

   The VXLAN system interface is automatically created with a `vxlan` type.
2. Configure the VXLAN interface settings:

```
config system interface
    edit <name>
        set vdom <string>
        set type vxlan
        set ip <IP_address>
        set allowaccess {ping https ssh http telnet fgfm radius-acct probe-response fabric ftm
speed-test}
    next
end
```

3. Connect the internal interface and VXLAN interface to the same L2 network.
   - Connect using a software switch:

```
config system switch-interface
    edit <name>
```

```
        set vdom <string>
        set member <member_1> <member_2> ... <member_n>
        set intra-switch-policy {implicit | explicit}
    next
end
```

`member <member_1> <member_2> ... <member_n>`	Enter the VXLAN interface and other physical or virtual interfaces that will share the L2 network. When adding an interface member to a software switch, it cannot have an IP address or be referenced in any other settings. For newly created VLAN interfaces, it is advised to change the role from LAN to undefined so that an address is not automatically assigned.
`intra-switch-policy {implicit | explicit}`	Allow any traffic between switch interfaces or require firewall policies to allow traffic between switch interfaces: • `implicit`: traffic between switch members is implicitly allowed. • `explicit`: traffic between switch members must match firewall policies (explicit firewall policies are required to allow traffic between members). When in explicit mode, traffic can be offloaded to SOC4/SOC5/NP6/NP7 processors.

- Connect using a virtual wire pair:

```
config system virtual-wire-pair
    edit <name>
        set member <member_1> <member_2>
        set wildcard-vlan {enable | disable}
        set vlan-filter <filter>
    next
end
```

`member <member_1> <member_2>`	Enter the VXLAN interface and other physical or virtual interface that will share the L2 network.
`wildcard-vlan {enable | disable}`	Enable/disable wildcard VLAN. Disable to prevent VLAN-tagged traffic between the members of the virtual wire pair (default). Enable for VLAN tags to be allowed between the members.
`vlan-filter <filter>`	When wildcard-vlan is enabled, set the VLAN filter to specify which VLANs are allowed. By default, an empty vlan-filter allows all VLANs.

4. If using a virtual wire pair, configure a firewall policy that allows bi-directional traffic between the members of the virtual wire pair and inspection between them:

```
config firewall policy
    edit <id>
        set name <name>
        set srcintf <member_1> <member_2>
        set dstintf <member_1> <member_2>
        set srcaddr "all"
```

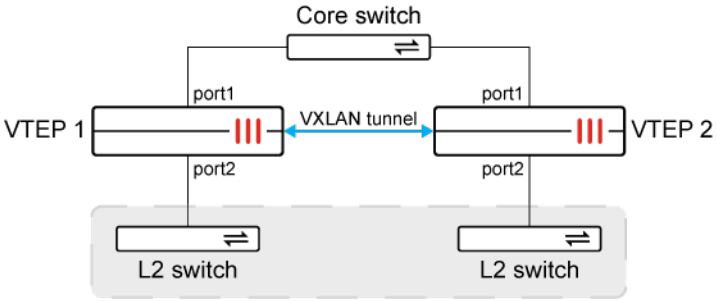
```
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end
```

Topologies

Many topologies can be deployed with VXLAN. A FortiGate can connect to VXLAN endpoints that are Fortinet devices or devices from other vendors. In the following topologies, it is assumed that at least one of the VTEPs is a FortiGate. The second VTEP can be any vendor.

Basic VXLAN between two VTEPs

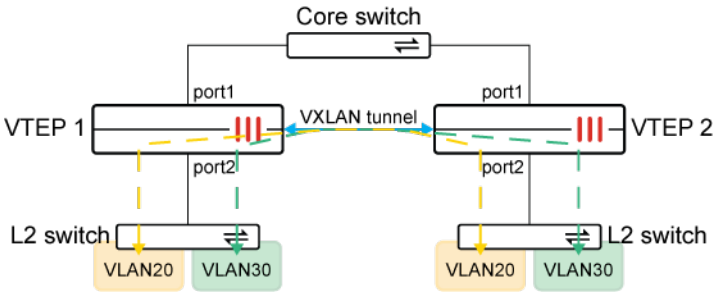
In this topology, a FortiGate (VTEP 1) is configured with a VXLAN interface over port1 where the `remote-ip` points to port1 of VTEP 2. The VXLAN interface and port2 can be associated with the same L2 network by making them members of either a software switch or a virtual wire pair. Devices under the L2 switches are part of the same L2 network.



See Virtual wire pair with VXLAN on page 311 for an example configuration.

VXLAN between two VTEPs with wildcard VLANs

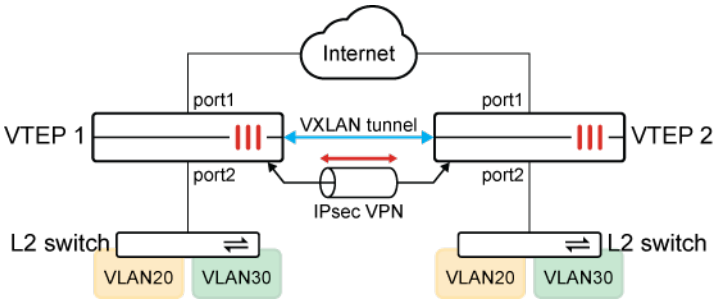
In this topology, a FortiGate (VTEP 1) is configured with a VXLAN interface over port1 where the `remote-ip` points to port1 of VTEP 2. The VXLAN interface is combined with port2 into the same L2 network using a virtual wire pair. The virtual wire pair allows wildcard VLANs to pass, which allows VLAN tags to be encapsulated over VXLAN. As a result, VLANs can span different switches over VXLAN.



Variations of these two scenarios can also be found in FortiGate to FortiSwitch FortiLink connections over VXLAN. See Deployment procedures in the FortiSwitch VXLAN Deployment Guide for example configurations.

VXLAN between two VTEPs over IPsec

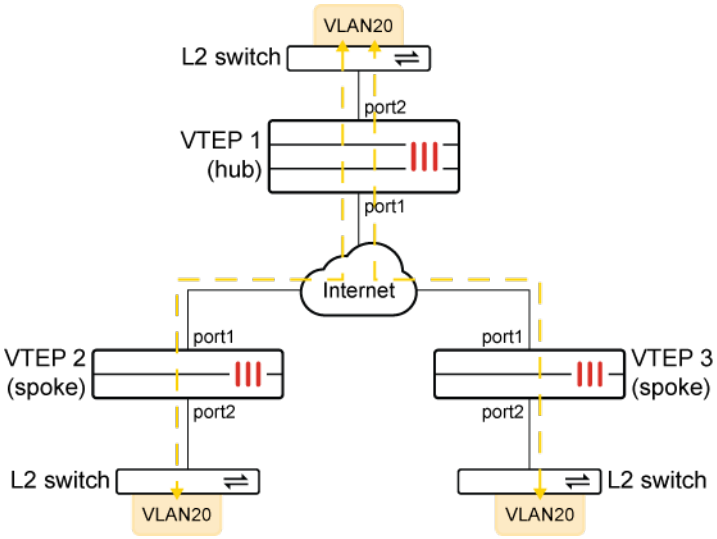
In scenarios where VTEPs are located in different sites and traffic must be secured between the sites, VXLAN will need to be encrypted over IPsec. The VXLAN interface must use the IPsec interface as its outgoing interface. The `remote-ip` must be configured as the IP of the remote IPsec gateway. The VXLAN interface can be combined with port2 into the same L2 network using a software switch or virtual wire pair. Devices under the L2 switches can communicate with each other.



See VXLAN over IPsec tunnel with virtual wire pair on page 313 for an example configuration. A variation of this scenario is explained in FortiGate LAN extension on page 995 and in FortiExtender as FortiGate LAN extension (FortiExtender FortiGate-Managed Administration Guide).

VXLAN between multiple VTEPs in an IPsec hub and spoke topology

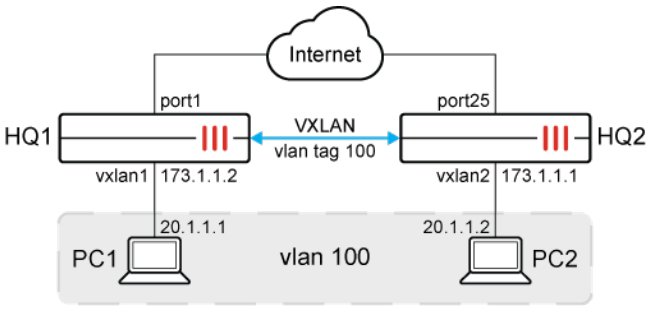
In this topology, an IPsec VPN hub and spoke overlay network is already configured between sites. To allow networks behind the hub and spokes to be connected together, each spoke has a VXLAN connection to the hub, and the hub allows interconnection between its private network and each of the VXLAN interfaces to the spokes. In this scenario, the private networks behind each spoke are actually on the same L2 network as the private network behind the hub.



See VXLAN over IPsec using a VXLAN tunnel endpoint on page 317 for an example configuration.

VLAN inside VXLAN

VLANs can be assigned to VXLAN interfaces. In a data center network where VXLAN is used to create an L2 overlay network and for multitenant environments, a customer VLAN tag can be assigned to VXLAN interface. This allows the VLAN tag from VLAN traffic to be encapsulated within the VXLAN packet.



To configure VLAN inside VXLAN on HQ1:

1. Configure VXLAN:

```
config system vxlan
    edit "vxlan1"
        set interface port1
        set vni 1000
        set remote-ip 173.1.1.1
    next
end
```

2. Configure system interface:

```
config system interface
    edit vlan100
        set vdom root
        set vlanid 100
        set interface dmz
    next
    edit vxlan100
        set type vlan
        set vlanid 100
        set vdom root
        set interface vxlan1
    next
end
```

3. Configure software-switch:

```
config system switch-interface
    edit sw1
        set vdom root
        set member vlan100 vxlan100
        set intra-switch-policy implicit
```

```
    next
end
```

The default `intra-switch-policy implicit` behavior allows traffic between member interfaces within the switch. Therefore, it is not necessary to create firewall policies to allow this traffic.

Instead of creating a software-switch, it is possible to use a virtual-wire-pair as well. See Virtual wire pair with VXLAN on page 311.

To configure VLAN inside VXLAN on HQ2:

1. Configure VXLAN:

```
config system vxlan
    edit "vxlan2"
        set interface port25
        set vni 1000
        set remote-ip 173.1.1.2
    next
end
```

2. Configure system interface:

```
config system interface
    edit vlan100
        set vdom root
        set vlanid 100
        set interface port20
    next
    edit vxlan100
        set type vlan
        set vlanid 100
        set vdom root
        set interface vxlan2
    next
end
```

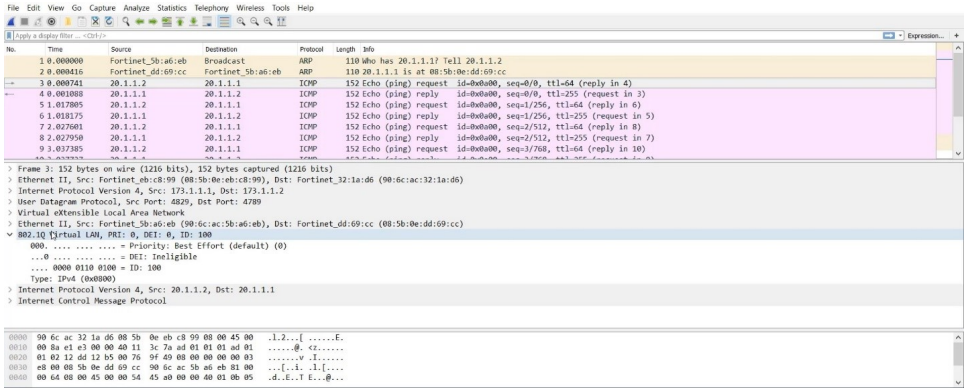
3. Configure software-switch:

```
config system switch-interface
    edit sw1
        set vdom root
        set member vlan100 vxlan100
    next
end
```

To verify the configuration:

Ping PC1 from PC2.

The following is captured on HQ2:

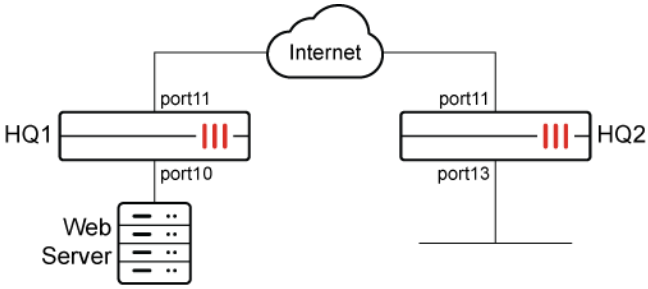


This captures the VXLAN traffic between 172.1.1.1 and 172.1.1.2 with the VLAN 100 tag inside.

Virtual wire pair with VXLAN

Virtual wire pairs can be used with VXLAN interfaces.

In this examples, VXLAN interfaces are added between FortiGate HQ1 and FortiGate HQ2, a virtual wire pair is added in HQ1, and firewall policies are created on both HQ1 and HQ2.



To create VXLAN interface on HQ1:

```
config system interface
    edit "port11"
        set vdom "root"
        set ip 10.2.2.1 255.255.255.0
        set allowaccess ping https ssh snmp telnet
    next
end
config system vxlan
    edit "vxlan1"
        set interface "port11"
        set vni 1000
        set remote-ip "10.2.2.2"
    next
end
```

To create VXLAN interface on HQ2:

```
config system interface
    edit "port11"
        set vdom "root"
        set ip 10.2.2.2 255.255.255.0
        set allowaccess ping https ssh snmp http
    next
end
config system vxlan
    edit "vxlan1"
        set interface "port11"
        set vni 1000
        set remote-ip "10.2.2.1"
    next
end
config system interface
    edit "vxlan1"
        set vdom "root"
        set ip 10.1.100.2 255.255.255.0
        set allowaccess ping https ssh snmp
    next
end
```

To create a virtual wire pair on HQ1:

```
config system virtual-wire-pair
    edit "vwp1"
        set member "port10" "vxlan1"
    next
end
```

To create a firewall policy on HQ1:

```
config firewall policy
    edit 5
        set name "vxlan-policy"
        set srcintf "port10" "vxlan1"
        set dstintf "port10" "vxlan1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "certificate-inspection"
        set av-profile "default"
        set webfilter-profile "default"
        set dnsfilter-profile "default"
        set ips-sensor "default"
        set application-list "default"
```

```
        set fsso disable
    next
end
```

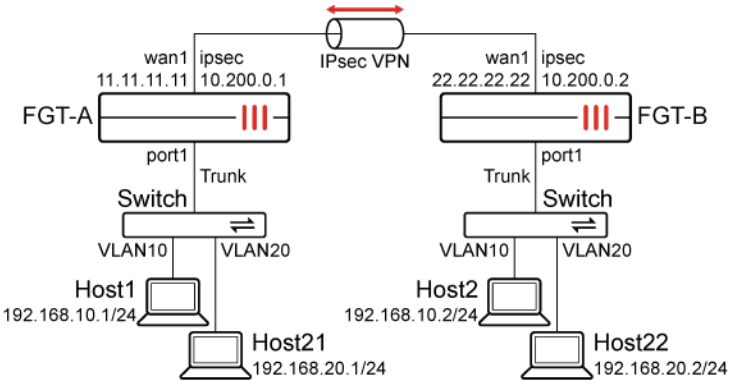
To create a firewall policy on HQ2:

```
config firewall policy
    edit 5
        set name "1"
        set srcintf "port13"
        set dstintf "vxlan1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set fsso disable
        set nat enable
    next
end
```

VXLAN over IPsec tunnel with virtual wire pair

VXLAN can be used to encapsulate VLAN traffic over a Layer 3 network. Using IPsec VPN tunnels to secure a connection between two sites, VXLAN can encapsulate VLAN traffic over the VPN tunnel to extend the VLANs between the two sites.

In this example, a site-to-site VPN tunnel is formed between two FortiGates. A VXLAN is configured over the IPsec interface. Multiple VLANs are connected to a switch behind each FortiGate. Host1 and Host2 are connected to VLAN10 on the switches on each site, and Host21 and Host22 are connected to VLAN20. Using virtual wire pairs, the internal interface (port1) will be paired with the VXLAN interface (vxlan) to allow VLAN traffic to pass through in either direction.



To configure FGT-A:

1. Configure the WAN interface:

```
config system interface
    edit "wan1"
        set vdom "root"
        set ip 11.11.11.11 255.255.255.0
        set allowaccess ping https ssh http fgfm
        set type physical
        set role wan
        set snmp-index 1
    next
end
```

2. Configure a static route to send all traffic out the WAN interface:

```
config router static
    edit 1
        set gateway 11.11.11.1
        set device "wan1"
    next
end
```

3. Configure the IPsec tunnel:

```
config vpn ipsec phase1-interface
    edit "ipsec"
        set interface "wan1"
        set peertype any
        set proposal aes256-sha1
        set remote-gw 22.22.22.22
        set psksecret **********
    next
end
config vpn ipsec phase2-interface
    edit "ipsec"
        set phase1name "ipsec"
        set proposal aes256-sha1
        set auto-negotiate enable
    next
end
```

4. Configure local and remote IP addresses for the IPsec interface:

```
config system interface
    edit "ipsec"
        set ip 10.200.0.1 255.255.255.255
        set remote-ip 10.200.0.2 255.255.255.252
    next
end
```

5. Configure the VXLAN interface and bind it to the IPsec interface:

```
config system vxlan
    edit "vxlan"
        set interface "ipsec"
        set vni 10
        set remote-ip "10.200.0.2"
    next
end
```

The remote IP address is the address of the remote IPsec peer.

6. Configure a virtual wire pair with the port1 and vxlan interfaces as members:

```
config system virtual-wire-pair
    edit "vwp"
        set member "port1" "vxlan"
        set wildcard-vlan enable
    next
end
```

The interfaces added to the virtual wire pair cannot be part of a switch, such as the default internal interface.

By enabling wildcard VLANs on the virtual wire pair, all VLAN tagged traffic that is allowed by the virtual wire pair firewall policies passes through the pair.

7. Configure a virtual wire pair firewall policy to allow traffic between the port1 and vxlan interfaces:

```
config firewall policy
    edit 4
        set name "vwp-pol"
        set srcintf "port1" "vxlan"
        set dstintf "port1" "vxlan"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end
```

To configure FGT-B

1. Configure the WAN interface:

```
config system interface
    edit "wan1"
        set vdom "root"
        set ip 22.22.22.22 255.255.255.0 255.255.255.0
        set allowaccess ping https ssh http fgfm
        set type physical
        set role wan
        set snmp-index 1
    next
end
```

2. Configure a static route to send all traffic out the WAN interface:

```
config router static
    edit 1
        set gateway 22.22.22.2
        set device "wan1"
    next
end
```

3. Configure the IPsec tunnel:

```
config vpn ipsec phase1-interface
    edit "ipsec"
        set interface "wan1"
        set peertype any
        set proposal aes256-sha1
        set remote-gw 11.11.11.11
        set psksecret **********
    next
end
config vpn ipsec phase2-interface
    edit "ipsec"
        set phase1name "ipsec"
        set proposal aes256-sha1
        set auto-negotiate enable
    next
end
```

4. Configure local and remote IP addresses for the IPsec interface:

```
config system interface
    edit "ipsec"
        set ip 10.200.0.2 255.255.255.255
        set remote-ip 10.200.0.1 255.255.255.252
    next
end
```

5. Configure the VXLAN interface and bind it to the IPsec interface:

```
config system vxlan
    edit "vxlan"
        set interface "ipsec"
        set vni 10
        set remote-ip "10.200.0.1"
    next
end
```

   The remote IP address is the address of the remote IPsec peer.

6. Configure a virtual wire pair with the port1 and vxlan interfaces as members:

```
config system virtual-wire-pair
    edit "vwp"
        set member "port1" "vxlan"
```

```
        set wildcard-vlan enable
    next
end
```

7. Configure a firewall policy to allow traffic between the port1 and vxlan interfaces:

```
config firewall policy
    edit 4
        set name "vwp-pol"
        set srcintf "port1" "vxlan"
        set dstintf "port1" "vxlan"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end
```

Test the configuration

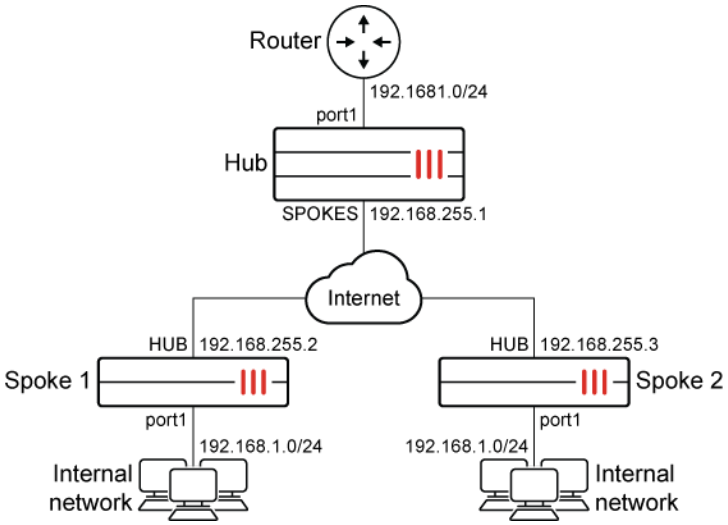
To test the configuration, ping Host2 (VLAN10: 192.168.10.2/24) from Host1 (VLAN10: 192.168.10.1/24):

```
C:\>ping 192.168.10.2

Pinging 192.168.10.2 with 32 bytes of data:
Reply from 192.168.10.2: bytes=32 time=8ms TTL=56
Reply from 192.168.10.2: bytes=32 time=8ms TTL=56
Reply from 192.168.10.2: bytes=32 time=8ms TTL=56
Reply from 192.168.10.2: bytes=32 time=11ms TTL=56

Ping statistics for 192.168.10.2:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 8ms, Maximum = 11ms, Average = 8ms
```

Host21 should also be able to ping Host22.

VXLAN over IPsec using a VXLAN tunnel endpoint

This example describes how to implement VXLAN over IPsec VPN using a VXLAN tunnel endpoint (VTEP).



This example uses a hub and spoke topology. Dialup VPN is used because it allows a single phase 1 dialup definition on the hub FortiGate. Additional spoke tunnels are added with minimal changes to the hub by adding a user account and VXLAN interface for each spoke. Spoke-to-spoke communication is established through the hub. This example assumes that the authentication users and user groups have already been created. While this topology demonstrates hub and spoke with dialup tunnels with XAuth authentication, the same logic can be applied to a static VPN with or without XAuth.

IPsec tunnel interfaces are used to support VXLAN tunnel termination. An IP address is set for each tunnel interface. Ping access is allowed for troubleshooting purposes.

VTEPs are created on the hub and each spoke to forward VXLAN traffic through the IPsec tunnels. VXLAN encapsulates OSI layer 2 Ethernet frames within layer 3 IP packets. You will need to either combine the internal port1 and VXLAN interface into a soft switch, or create a virtual wire pair so that devices behind port1 have direct layer 2 access to remote peers over the VXLAN tunnel. This example uses a switch interface on the hub and a virtual wire pair on the spokes to demonstrate the two different methods.

In order to apply an IPsec VPN interface on the VXLAN interface setting, `net-device` must be disabled in the IPsec VPN phase 1 settings.

To configure the hub FortiGate:

1. Configure the IPsec phase 1 interface:

```
config vpn ipsec phase1-interface
    edit "SPOKES"
        set type dynamic
        set interface "port2"
        set mode aggressive
        set peertype one
        set net-device disable
        set proposal aes256-sha256
        set xauthtype auto
        set authusrgrp "SPOKES"
        set peerid "SPOKES"
        set psksecret <secret>
```

```
   next
end
```

2. Configure the IPsec phase 2 interface:

```
config vpn ipsec phase2-interface
    edit "SPOKES"
        set phase1name "SPOKES"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
    next
end
```

3. Configure the IPsec VPN policy that allows VXLAN traffic between the spokes:

```
config firewall policy
    edit 1
        set name "VXLAN_SPOKE_to_SPOKE"
        set srcintf "SPOKES"
        set dstintf "SPOKES"
        set srcaddr "NET_192.168.255.0"
        set dstaddr "NET_192.168.255.0"
        set action accept
        set schedule "always"
        set service "UDP_4789"
        set logtraffic all
        set fsso disable
    next
end
```

4. Configure the IPsec tunnel interfaces (the remote IP address is not used, but it is necessary for this configuration):

```
config system interface
    edit "SPOKES"
        set vdom "root"
        set ip 192.168.255.1 255.255.255.255
        set allowaccess ping
        set type tunnel
        set remote-ip 192.168.255.254 255.255.255.0
        set snmp-index 12
        set interface "port2"
    next
end
```

5. Configure the VXLAN interfaces. Each spoke requires a VXLAN interface with a different VNI. The remote IP is the tunnel interfaces IP of the spokes.
   a. Spoke 1:

```
config system VXLAN
    edit "SPOKES_VXLAN1"
        set interface "SPOKES"
```

```
        set vni 1
        set remote-ip "192.168.255.2"
    next
end
```

b. Spoke 2:

```
config system VXLAN
    edit "SPOKES_VXLAN2"
        set interface "SPOKES"
        set vni 2
        set remote-ip "192.168.255.3"
    next
end
```

To configure the spoke FortiGates:

1. Configure the IPsec phase 1 interface:

```
config vpn ipsec phase1-interface
    edit "HUB"
        set interface "port2"
        set mode aggressive
        set peertype any
        set net-device disable
        set proposal aes256-sha256
        set localid "SPOKES"
        set xauthtype client
        set authusr "SPOKE1"
        set authpasswd <secret>
        set remote-gw <hub public IP>
        set psksecret <secret>
    next
end
```

2. Configure the IPsec phase 2 interface:

```
config vpn ipsec phase2-interface
    edit "HUB"
        set phase1name "HUB"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
        set auto-negotiate enable
        set src-subnet 192.168.255.2 255.255.255.255
    next
end
```

> The hub FortiGate inserts a reverse route pointing to newly established tunnel interfaces for any of the subnets that the spoke FortiGate's source quick mode selectors provides. This is why you should set the tunnel IP address here.

3. Configure the IPsec VPN policy:

```
config firewall policy
    edit 1
        set name "VTEP_IPSEC_POLICY"
        set srcintf "HUB"
        set dstintf "HUB"
        set srcaddr "none"
        set dstaddr "none"
        set action accept
        set schedule "always"
        set service "PING"
        set logtraffic disable
        set fsso disable
    next
end
```

4. Configure the IPsec tunnel interface:

```
config system interface
    edit "HUB"
        set vdom "root"
        set ip 192.168.255.2 255.255.255.255
        set allowaccess ping
        set type tunnel
        set remote-ip 192.168.255.1 255.255.255.0
        set snmp-index 12
        set interface "port2"
    next
end
```

5. Configure the VXLAN interfaces (the remote IP is the tunnel interface IP of the hub):
   a. Spoke 1:

```
config system VXLAN
    edit "HUB_VXLAN"
        set interface "HUB"
        set vni 1
        set remote-ip "192.168.255.1"
    next
end
```

   b. Spoke 2:

```
config system VXLAN
    edit "HUB_VXLAN"
        set interface "HUB"
        set vni 2
        set remote-ip "192.168.255.1"
    next
end
```

To bind the VXLAN interface to the internal interface:

1. Configure a switch interface on the hub:

```
config system switch-interface
    edit "SW"
        set vdom "root"
        set member "port1" "SPOKES_VXLAN1" "SPOKES_VXLAN2"
        set intra-switch-policy {implicit | explicit}
    next
end
```

> Allowing intra-switch traffic is implicitly allowed by default. Use `set intra-switch-policy explicit` to require firewall policies to allow traffic between switch interfaces.

2. Configure a virtual wire pair on the spokes:

```
config system virtual-wire-pair
    edit "VWP"
        set member "HUB_VXLAN" "port1"
    next
end
```

> The virtual wire pair requires an explicit policy to allow traffic between interfaces.

To test the configuration:

1. Ping the hub FortiGate from the spoke FortiGate:

```
user@pc-spoke1:~$ ping 192.168.1.1 -c 3
PING 192.168.1.1 (192.168.1.1) 56(84) bytes of data.
64 bytes from 192.168.1.1: icmp_seq=1 ttl=64 time=1.24 ms
64 bytes from 192.168.1.1: icmp_seq=2 ttl=64 time=0.672 ms
64 bytes from 192.168.1.1: icmp_seq=3 ttl=64 time=0.855 ms
--- 192.168.1.1 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002 ms
rtt min/avg/max/mdev = 0.672/0.923/1.243/0.239 ms
```

2. Sniff traffic on the hub FortiGate:

```
# diagnose sniffer packet any 'icmp or (udp and port 4789)' 4 0
interfaces=[any] filters=[icmp or (udp and port 4789)]
15:00:01.438230 SPOKES in 192.168.255.2.4790 -&gt; 192.168.255.1.4789: udp 106
15:00:01.438256 SPOKES_VXLAN1 in 192.168.1.2 -&gt; 192.168.1.1: icmp: echo request
15:00:01.438260 port1 out 192.168.1.2 -&gt; 192.168.1.1: icmp: echo request
15:00:01.438532 port1 in 192.168.1.1 -&gt; 192.168.1.2: icmp: echo reply
15:00:01.438536 SPOKES_VXLAN1 out 192.168.1.1 -&gt; 192.168.1.2: icmp: echo reply
15:00:01.438546 SPOKES out 192.168.255.1.4851 -&gt; 192.168.255.2.4789: udp 106
```

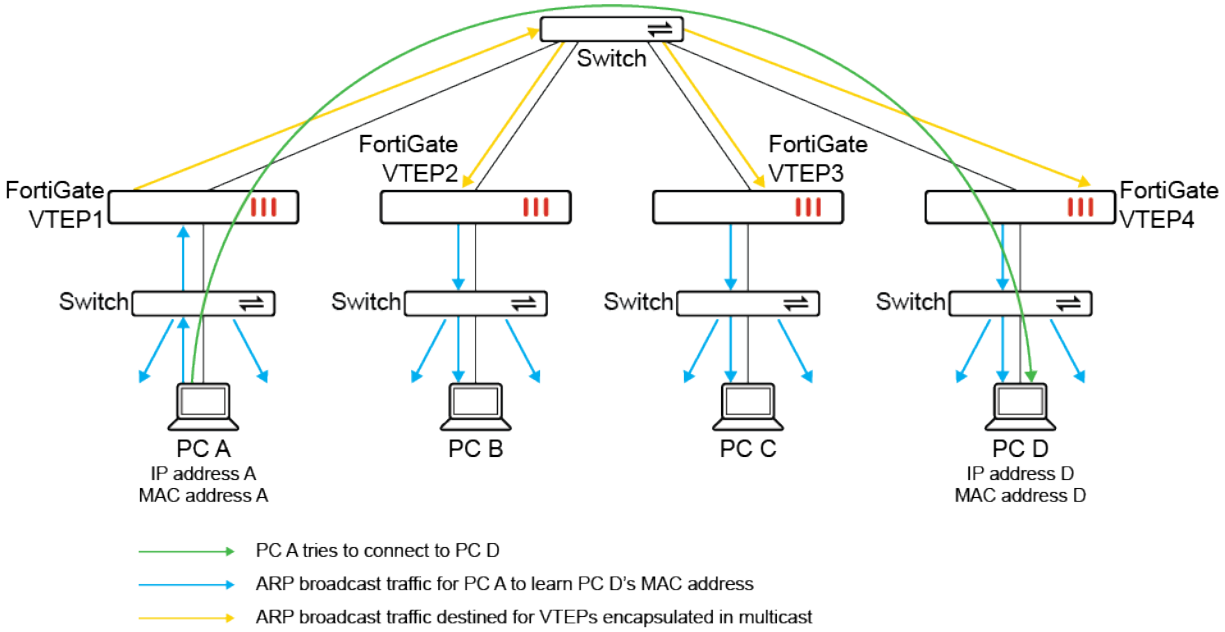
VXLAN with MP-BGP EVPN

FortiOS supports VXLAN as implemented according to RFC 7348. Currently, VXLAN relies on determining the MAC address of the destination host by using address resolution protocol (ARP) broadcast frames encapsulated in multicast packets.

- A multicast group is maintained with all the VXLAN tunnel endpoints (VTEPs) associated with the same VXLAN, namely, with the same VXLAN network identifier (VNI).
- The multicast packets that encapsulate ARP broadcast frames are sent to this multicast group, and then the destination host replies to the source host using unicast IP packet encapsulated using VXLAN.
- The source and destination FortiGates as VTEPs each maintain a mapping of MAC addresses to remote VTEPs.

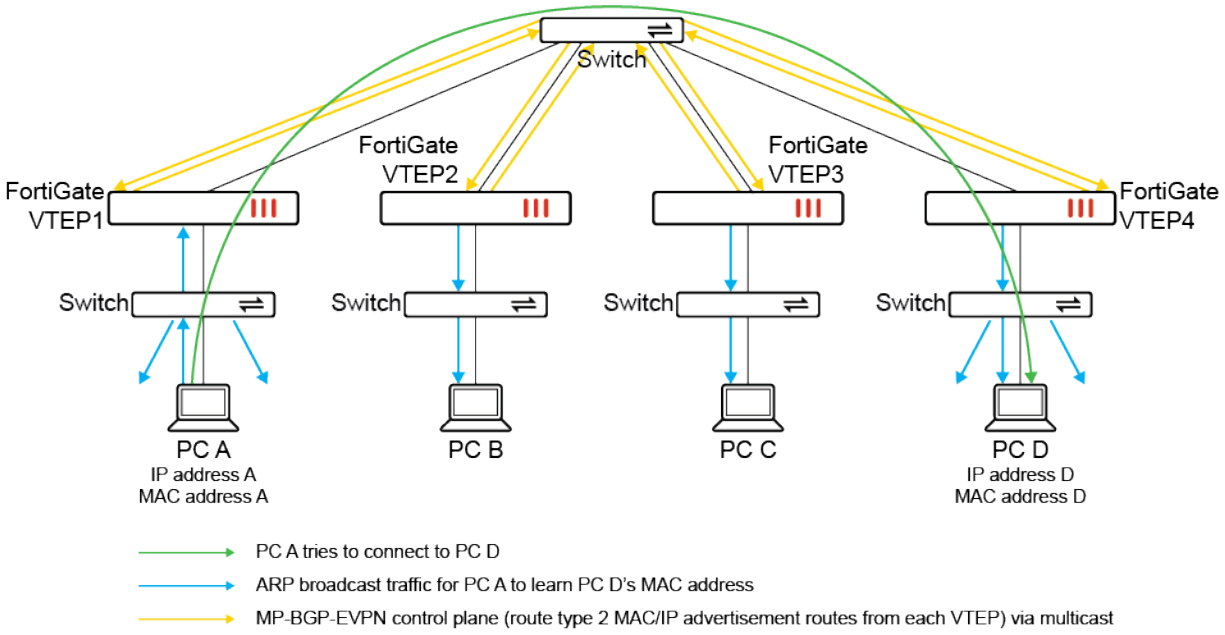
As with non-VXLAN traffic, VXLAN relies on the preceding ARP process, commonly known as flood-and-learn that floods the network with broadcast frames encapsulated as multicast packets to learn MAC addresses. In the RFC 7348 implementation of VXLAN, the data plane is simultaneously used as a control plane.

The following topology demonstrates how flood-and-learn uses ARP broadcast traffic flooded throughout the VXLAN for PC A to learn PC D's MAC address when PC A tries to connect to PC D.



Multiprotocol Border Gateway Protocol Ethernet Virtual Private Network (MP-BGP EVPN) support for VXLAN allows for learning MAC addresses in a way that is more suitable for large deployments than flood-and-learn.

MP-BGP EVPN is a standards-based control plane that supports the distribution of attached host MAC and IP addresses using MP-BGP, namely, using the EVPN address family and MAC addresses treated as routing entries in BGP. As a control plane that is separate from the data plane, MP-BGP EVPN avoids flood-and-learn in the network, and the wide use of BGP as an external gateway protocol on the internet proves its ability to scale well with large deployments. The following topology demonstrates how MP-BGP EVPN distributes route type 2 MAC/IP advertisement routes among VTEPs in the VXLAN, and minimizes ARP broadcast traffic required for PC A to learn PC D's MAC address when PC A tries to connect to PC D.



MP-BGP EVPN supports the following features:

- Route type 2 (MAC/IP advertisement route) and route type 3 (inclusive multicast Ethernet tag route)
- Intra-subnet communication
- Single-homing use cases
- VLAN-based service, namely, there is only one broadcast domain per EVPN instance (EVI). This is due to the current VXLAN design that supports a single VNI for a VXLAN interface.
- EVPN running on IPv4 unicast VXLAN
- Egress replication for broadcast, unknown unicast, and multicast (BUM) traffic
- VXLAN MAC learning from traffic
- IP address local learning
- ARP suppression

For more information about MP-BGP EVPN, see RFC 7432. For more information about EVPN and VXLAN, see RFC 8365.

Currently, MP-BGP EVPN supports only VRF 0.

Basic MP-BGP EVPN configuration

The MP-BGP EVPN feature builds on the CLI commands used for configuring VXLAN using a VXLAN tunnel endpoint (VTEP). See General VXLAN configuration and topologies on page 305 for more details.

After configuring VXLAN using a VTEP, the following CLI commands are configured to enable MP-BGP EVPN on each VTEP.

To configure MP-BGP EVPN on each VTEP:

1. Configure the EVPN settings:

```
config system evpn
    edit <id>
        set rd {AA | AA:NN | A.B.C.D:NN}
        set import-rt <AA:NN>
        set export-rt <AA:NN>
        set ip-local-learning {enable | disable}
        set arp-suppression {enable | disable}
    next
end
```

   The `ip-local-learning` setting is used to enable/disable monitoring the local ARP table of the switch interface to learn the IP/MAC bindings, and advertise them to neighbors. This setting is disabled by default, but must be enabled when configuring MP-BGP EVPN.

   The `arp-suppression` setting is used to enable/disable using proxy ARP to perform suppression of ARP discovery using the flood-and-learn approach. This setting is disabled by default. When enabled, proxy ARP entries are added on the switch interface to suppress the ARP flooding of known IP/MAC bindings, which were learned by the MP-BGP EVPN control plane.

2. Configure the EVPN settings within the VXLAN settings:

```
config system vxlan
    edit <name>
        set interface <string>
        set vni <integer>
        set evpn-id <integer>
        set learn-from-traffic {enable | disable}
    next
end
```

   The `learn-from-traffic` setting is used to enable/disable learning of remote VNIs from VXLAN traffic. This setting is disabled by default, and should only be enabled when local and all remote peers are using same VNI value, and some of the peers do not have MP-BGP EVPN capability.

3. Configure the BGP settings:

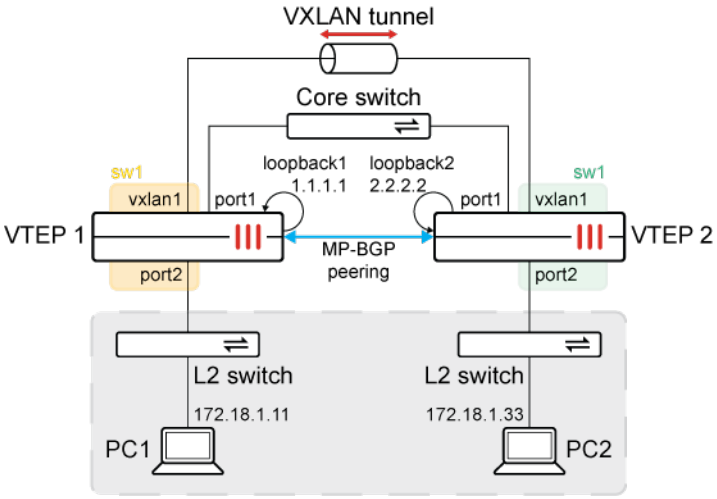
```
config router bgp
    set ibgp-multipath {enable | disable}
    set recursive-next-hop {enable | disable}
    set graceful-restart {enable | disable}
    config neighbor
        edit <WAN_IP_of_other_VTEP>
            set ebgp-enforce-multihop {enable | disable}
            set next-hop-self {enable | disable}
            set next-hop-self-vpnv4 {enable | disable}
            set soft-reconfiguration {enable | disable}
            set soft-reconfiguration-evpn {enable | disable}
            set remote-as <AS_number>
        next
    end
end
```

4. Configure the EVPN setting within the HA settings:

```
config system ha
    set evpn-ttl <integer>
end
```

Example

In this example, two FortiGates are configured as VXLAN tunnel endpoints (VTEPs). A VXLAN is configured to allow L2 connectivity between the networks behind each FortiGate. The VXLAN interface vxlan1 and port2 are placed on the same L2 network using a software switch (sw1). An L2 network is formed between PC1 and PC2. MP-BGP EVPN is used as the control plane to learn and distribute MAC address information within a single L2 domain identified using a specific VNI.



The VTEPs have the following MAC address tables:

Interface/endpoint	VTEP1	VTEP2
vxlan1	82:51:d1:44:bf:93	d2:21:00:c9:e6:98
port2	50:00:00:03:00:01	50:00:00:04:00:01
sw1	50:00:00:03:00:01	50:00:00:04:00:01

The MAC address of PC1 is 00:50:00:00:06:00. The MAC address of PC2 is 00:50:00:00:07:00.

This example assumes that the WAN interface and default route settings have already been configured on the VTEP 1 and VTEP 2 FortiGates. These configurations are omitted from the example. All peers are configured for MP-BGP EVPN.

To configure the VTEP1 FortiGate:

1. Configure the loopback interface:

```
config system interface
    edit "loopback1"
        set vdom "root"
```

```
        set ip 1.1.1.1 255.255.255.255
        set allowaccess ping https ssh http
        set type loopback
    next
end
```

2. Configure the EVPN settings:

```
config system evpn
    edit 100
        set rd "100:100"
        set import-rt "1:1"
        set export-rt "1:1"
        set ip-local-learning enable
        set arp-suppression enable
    next
end
```

3. Configure the local interface and EVPN settings within the VXLAN settings:

```
config system vxlan
    edit "vxlan1"
        set interface "loopback1"
        set vni 1000
        set evpn-id 100
    next
end
```

4. Configure the EVPN settings within the BGP settings:

```
config router bgp
    set as 65001
    set router-id 1.1.1.1
    set ibgp-multipath enable
    set recursive-next-hop enable
    set graceful-restart enable
    config neighbor
        edit "172.25.160.101"
            set ebgp-enforce-multihop enable
            set next-hop-self enable
            set next-hop-self-vpnv4 enable
            set soft-reconfiguration enable
            set soft-reconfiguration-evpn enable
            set remote-as 65001
        next
    end
    config network
        edit 1
            set prefix 1.1.1.1 255.255.255.255
        next
    end
end
```

172.27.16.237 is the WAN IP address of the VTEP2 FortiGate.

5. Configure the software switch:

```
config system switch-interface
    edit "sw1"
        set vdom "root"
        set member "port2" "vxlan1"
        set intra-switch-policy explicit
    next
end
```

6. Configure the software switch interface settings:

```
config system interface
    edit "sw1"
        set vdom "root"
        set ip 172.18.1.253 255.255.255.0
        set allowaccess ping
        set type switch
    next
end
```

7. Configure the firewall policies between the member interfaces in the software switch:

```
config firewall policy
     edit 1
         set srcintf "port2"
         set dstintf "vxlan1"
         set action accept
         set srcaddr "all"
         set dstaddr "all"
         set schedule "always"
         set service "ALL"
     next
     edit 2
         set srcintf "vxlan1"
         set dstintf "port2"
         set action accept
         set srcaddr "all"
         set dstaddr "all"
         set schedule "always"
         set service "ALL"
     next
end
```

To configure the VTEP2 FortiGate:

1. Configure the loopback interface:

```
config system interface
    edit "loopback2"
        set vdom "root"
```

```
        set ip 2.2.2.2 255.255.255.255
        set allowaccess ping https ssh http
        set type loopback
    next
end
```

2. Configure the EVPN settings:

```
config system evpn
    edit 100
        set rd "100:100"
        set import-rt "1:1"
        set export-rt "1:1"
        set ip-local-learning enable
        set arp-suppression enable
    next
end
```

3. Configure the local interface and EVPN settings within the VXLAN settings:

```
config system vxlan
    edit "vxlan1"
        set interface "loopback2"
        set vni 1000
        set evpn-id 100
    next
end
```

4. Configure the EVPN settings within the BGP settings:

```
config router bgp
    set as 65001
    set router-id 2.2.2.2
    set ibgp-multipath enable
    set recursive-next-hop enable
    set graceful-restart enable
    config neighbor
        edit "172.25.160.100"
            set ebgp-enforce-multihop enable
            set next-hop-self enable
            set next-hop-self-vpnv4 enable
            set soft-reconfiguration enable
            set soft-reconfiguration-evpn enable
            set remote-as 65001
        next
    end
    config network
        edit 1
            set prefix 2.2.2.2 255.255.255.255
        next
    end
end
```

172.27.16.236 is the WAN IP address of the VTEP1 FortiGate.

5. Configure the software switch:

```
config system switch-interface
    edit "sw1"
        set vdom "root"
        set member "port2" "vxlan1"
        set intra-switch-policy explicit
    next
end
```

6. Configure the software switch interface settings:

```
config system interface
    edit "sw1"
        set vdom "root"
        set ip 172.18.1.254 255.255.255.0
        set allowaccess ping
        set type switch
    next
end
```

7. Configure the firewall policies between the member interfaces in the software switch:

```
config firewall policy
     edit 1
         set srcintf "port2"
         set dstintf "vxlan1"
         set action accept
         set srcaddr "all"
         set dstaddr "all"
         set schedule "always"
         set service "ALL"
     next
     edit 2
         set srcintf "vxlan1"
         set dstintf "port2"
         set action accept
         set srcaddr "all"
         set dstaddr "all"
         set schedule "always"
         set service "ALL"
     next
end
```

To verify the MP-BGP EVPN status on the VTEP1 FortiGate:

1. From a host computer with IP address 172.18.1.11, perform the following.
   a. Check the ARP cache:

```
# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
172.18.1.253             ether   50:00:00:03:00:01   C                     ens3
```

b. Ping the host computer with IP address 172.18.1.33:

```
# ping 172.18.1.33 -c 4
PING 172.18.1.33 (172.18.1.33) 56(84) bytes of data.
64 bytes from 172.18.1.33: icmp_seq=1 ttl=64 time=1325 ms
64 bytes from 172.18.1.33: icmp_seq=2 ttl=64 time=319 ms
64 bytes from 172.18.1.33: icmp_seq=3 ttl=64 time=3.96 ms
64 bytes from 172.18.1.33: icmp_seq=4 ttl=64 time=1.66 ms

--- 172.18.1.33 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3007ms
rtt min/avg/max/mdev = 1.660/412.614/1325.209/542.530 ms
```

c. Check the ARP cache again:

```
# arp
Address                  HWtype  HWaddress           Flags Mask            Iface
172.18.1.33              ether   00:50:00:00:07:00   C                     ens3
172.18.1.253             ether   50:00:00:03:00:01   C                     ens3
```

2. On the VTEP1 FortiGate, run the switch and VXLAN debug commands.

a. Verify the forwarding database for vxlan1:

```
# diagnose sys vxlan fdb list vxlan1
mac=00:00:00:00:00:00 state=0x0082 remote_ip=2.2.2.2 port=4789 vni=1000 ifindex0
mac=00:50:00:00:07:00 state=0x0082 remote_ip=2.2.2.2 port=4789 vni=1000 ifindex0

total fdb num: 2
```

b. Verify the forwarding database statistics for vxlan1:

```
# diagnose sys vxlan fdb stat vxlan1
fdb_table_size=256 fdb_table_used=2 fdb_entry=2 fdb_max_depth=1 cleanup_idx=0 c2
```

c. Verify the bridging information for sw1:

```
# diagnose netlink brctl name host sw1
show bridge control interface sw1 host.
fdb: hash size=32768, used=5, num=5, depth=1, gc_time=4, ageing_time=3, arp-sups
Bridge sw1 host table
port no device  devname mac addr                ttl     attributes
  2     15      vxlan1  00:00:00:00:00:00       28       Hit(28)
  2     15      vxlan1  00:50:00:00:07:00       18       Hit(18)
  2     15      vxlan1  82:51:d1:44:bf:93       0       Local Static
  1     4       port2   00:50:00:00:06:00       14       Hit(14)
  1     4       port2   50:00:00:03:00:01       0       Local Static
```

3. Run the BGP EVPN commands and observe the route type 2 (MAC/IP advertisement route) and route type 3 (inclusive multicast Ethernet tag route).

a. Verify the BGP L2 VPN EVPN summary information:

```
# get router info bgp evpn summary

VRF 0 BGP router identifier 1.1.1.1, local AS number 65001
BGP table version is 2
1 BGP AS-PATH entries
0 BGP community entries

Neighbor       V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/Pd
172.25.160.101 4      65001       9       9        1    0    0 00:04:02        3

Total number of neighbors 1
```

b. Verify the BGP L2 VPN EVPN network information:

```
# get router info bgp evpn network
   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
Route Distinguisher: 100:100 (Default for VRF 0)
*> [2][0][48][00:50:00:00:06:00][0]/72
                    1.1.1.1             0              100  32768        0 i <-/>
*> [2][0][48][00:50:00:00:06:00][32][172.18.1.11]/104
                    1.1.1.1             0              100  32768        0 i <-/>
*>i[2][0][48][00:50:00:00:07:00][0]/72
                    2.2.2.2             0              100      0        0 i <-/>
*>i[2][0][48][00:50:00:00:07:00][32][172.18.1.33]/104
                    2.2.2.2             0              100      0        0 i <-/>
*> [3][0][32][1.1.1.1]/80
                    1.1.1.1             0              100  32768        0 i <-/>
*>i[3][0][32][2.2.2.2]/80
                    2.2.2.2             0              100      0        0 i <-/>

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
Route Distinguisher: 100:100 (received from VRF 0)
*>i[2][0][48][00:50:00:00:07:00][0]/72
                    2.2.2.2             0              100      0        0 i <-/>
*>i[2][0][48][00:50:00:00:07:00][32][172.18.1.33]/104
                    2.2.2.2             0              100      0        0 i <-/>
*>i[3][0][32][2.2.2.2]/80
                    2.2.2.2             0              100      0        0 i <-/>
```

c. Verify the BGP L2 VPN EVPN context:

```
# get router info bgp evpn context
L2VPN EVPN context for VRF 0
ID 100 vlan-based, RD is [100:100]
 Import RT: RT:1:1
 Export RT: RT:1:1
  Bridge domain 0 VNI 1000
   Encapsulation 8(VXLAN)
   Source interface loopback1
   Source address 1.1.1.1
```

d. Verify the BGP L2 VPN EVPN information for VRF 0:

```
# get router info bgp evpn vrf 0
   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
Route Distinguisher: 100:100 (Default for VRF 0)
*> [2][0][48][00:50:00:00:06:00][0]/72
                    1.1.1.1             0             100  32768        0 i <-/>
*> [2][0][48][00:50:00:00:06:00][32][172.18.1.11]/104
                    1.1.1.1             0             100  32768        0 i <-/>
*>i[2][0][48][00:50:00:00:07:00][0]/72
                    2.2.2.2             0             100      0        0 i <-/>
*>i[2][0][48][00:50:00:00:07:00][32][172.18.1.33]/104
                    2.2.2.2             0             100      0        0 i <-/>
*> [3][0][32][1.1.1.1]/80
                    1.1.1.1             0             100  32768        0 i <-/>
*>i[3][0][32][2.2.2.2]/80
                    2.2.2.2             0             100      0        0 i <-/>

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
Route Distinguisher: 100:100 (received from VRF 0)
*>i[2][0][48][00:50:00:00:07:00][0]/72
                    2.2.2.2             0             100      0        0 i <-/>
*>i[2][0][48][00:50:00:00:07:00][32][172.18.1.33]/104
                    2.2.2.2             0             100      0        0 i <-/>
*>i[3][0][32][2.2.2.2]/80
                    2.2.2.2             0             100      0        0 i <-/>
```

e. Verify the BGP L2 VPN EVPN information for RD 100:100:

```
# get router info bgp evpn rd 100:100
   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
Route Distinguisher: 100:100 (Default for VRF 0)
*> [2][0][48][00:50:00:00:06:00][0]/72
                    1.1.1.1             0             100  32768        0 i <-/>
*> [2][0][48][00:50:00:00:06:00][32][172.18.1.11]/104
                    1.1.1.1             0             100  32768        0 i <-/>
*>i[2][0][48][00:50:00:00:07:00][0]/72
                    2.2.2.2             0             100      0        0 i <-/>
*>i[2][0][48][00:50:00:00:07:00][32][172.18.1.33]/104
                    2.2.2.2             0             100      0        0 i <-/>
*> [3][0][32][1.1.1.1]/80
                    1.1.1.1             0             100  32768        0 i <-/>
*>i[3][0][32][2.2.2.2]/80
                    2.2.2.2             0             100      0        0 i <-/>

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
Route Distinguisher: 100:100 (received from VRF 0)
*>i[2][0][48][00:50:00:00:07:00][0]/72
                    2.2.2.2             0             100      0        0 i <-/>
*>i[2][0][48][00:50:00:00:07:00][32][172.18.1.33]/104
                    2.2.2.2             0             100      0        0 i <-/>
*>i[3][0][32][2.2.2.2]/80
                    2.2.2.2             0             100      0        0 i <-/>
```

f. Verify the neighbor EVPN advertised routes for 172.25.160.101:

```
# get router info bgp neighbors 172.25.160.101 advertised-routes evpn
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight RouteTag Path
Route Distinguisher: 100:100 (Default for VRF 0) (Default for VRF 0)
*>i[2][0][48][00:50:00:00:06:00][0]/72
                    1.1.1.1                            100  32768        0 i <-/>
*>i[2][0][48][00:50:00:00:06:00][32][172.18.1.11]/104
                    1.1.1.1                            100  32768        0 i <-/>
*>i[3][0][32][1.1.1.1]/80
                    1.1.1.1                            100  32768        0 i <-/>

Total number of prefixes 3
```

g. Verify the neighbor EVPN received routes for 172.25.160.101:

```
# get router info bgp neighbors 172.25.160.101 received-routes evpn
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight RouteTag Path
Route Distinguisher: 100:100 (received from VRF 0) (received from VRF 0)
*>i[2][0][48][00:50:00:00:07:00][0]/72
                    2.2.2.2                            100      0        0 i <-/>
*>i[2][0][48][00:50:00:00:07:00][32][172.18.1.33]/104
                    2.2.2.2                            100      0        0 i <-/>
*>i[3][0][32][2.2.2.2]/80
                    2.2.2.2                            100      0        0 i <-/>

Total number of prefixes 3
```

h. Verify the neighbor EVPN routes:

```
# get router info bgp neighbors 172.25.160.101 routes evpn
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight RouteTag Path
Route Distinguisher: 100:100 (Default for VRF 0) (Default for VRF 0)
*>i[2][0][48][00:50:00:00:07:00][0]/72
                    2.2.2.2             0              100      0        0 i <-/>
*>i[2][0][48][00:50:00:00:07:00][32][172.18.1.33]/104
                    2.2.2.2             0              100      0        0 i <-/>
*>i[3][0][32][2.2.2.2]/80
                    2.2.2.2             0              100      0        0 i <-/>
Route Distinguisher: 100:100 (received from VRF 0) (received from VRF 0)
*>i[2][0][48][00:50:00:00:07:00][0]/72
                    2.2.2.2             0              100      0        0 i <-/>
*>i[2][0][48][00:50:00:00:07:00][32][172.18.1.33]/104
```

```
                    2.2.2.2             0             100      0        0 i <-/>
*>i[3][0][32][2.2.2.2]/80
                    2.2.2.2             0             100      0        0 i <-/>

Total number of prefixes 6
```

4. Run the following EVPN get commands.
   a. Verify the EVPN instances:

```
# get l2vpn evpn instance
EVPN instance: 100
IP local learning enabled
ARP suppression enabled
HA primary
   Number of bridge domain: 1
   Bridge domain: TAGID 0 VNI 1000 ADDR 1.1.1.1 VXLAN vxlan1 SWITCH sw1
```

   b. Verify the EVPN table:

```
# get l2vpn evpn table
EVPN instance 100
 Broadcast domain VNI 1000 TAGID 0

  EVPN instance 100
 Broadcast domain VNI 1000 TAGID 0

  EVPN MAC table:
  MAC               VNI      Remote Addr     Binded Address
  00:50:00:00:07:00 1000     2.2.2.2         172.18.1.33
                    1000     2.2.2.2         -

  EVPN IP table:
  Address         VNI      Remote Addr     MAC
  172.18.1.33     1000     2.2.2.2         00:50:00:00:07:00

  EVPN Local MAC table:
  "Inactive" means this MAC/IP pair will not be sent to peer.
  Flag code: S - Static F - FDB. Trailing * means HA
  MAC               Flag Status   Binded Address
  00:50:00:00:06:00      Active   172.18.1.11
                    F    Active   -

  EVPN Local IP table:
  Address         MAC
  172.18.1.11     00:50:00:00:06:00

  EVPN PEER table:
  VNI      Remote Addr     Binded Address
  1000     2.2.2.2         2.2.2.2
```

5. Run the proxy ARP diagnose command:

```
# diagnose ip parp list
Address           Hardware Addr     Interface
172.18.1.33       00:50:00:00:07:00 sw1
```

EVPN support for VXLAN Anycast gateway and type-5 routing NEW

This feature adds support for VXLAN Anycast gateway, EVPN IRB (Ethernet VPN Integrated Routing and Bridging), and type-5 EVPN routes, enabling seamless L2/L3 service integration and scalable IP prefix advertisement across the VXLAN fabric. It simplifies mobility and routing in modern data center deployments. For more information about EVPN, see VXLAN with MP-BGP EVPN on page 323.

The `config router bgp` command has the following relevant options:

```
config router bgp
    config redistribute {connected | static | isis | ospf | rip}
        set status-evpn {enable | disable}
        set route-map-evpn <string>
    end
    config {neighbor | neighbor-group}
        edit <string>
            set adv-evpn-route {type2 | type5 | local}
        next
    end
end
```

Option	Description
`config redistribute {connected | static | isis | ospf | rip}`	Types of routes to be redistributed.
`status-evpn {enable | disable}`	Enable/disable EVPN redistribution status.
`route-map-evpn <string>`	Route map name for EVPN redistribution.
`adv-evpn-route {type2 | type5 | local}`	Types of EVPN routes that can be advertised to this neighbor as IPv4 routes: • `type2`: Type 2 EVPN routes. • `type5`: Type 5 EVPN routes. • `local`: Local EVPN routes.

The `config system evpn` command has the following relevant options:

```
config system evpn
    edit <id>
        set type {macvrf | ipvrf}
        set adv-default-gw {enable | disable}
        set virtual-mac-vrid <integer>
        set l3-instance <integer>
```

```
        set *interface <string>
        set distribute-local-route {enable | disable}
    next
end
```

Option	Description
`type {macvrf | ipvrf}`	EVPN instance type: • `macvrf`: MAC VRF. • `ipvrf`: L3 VRF.
`adv-default-gw {enable | disable}`	Enable/disable advertisement of default gateway.
`virtual-mac-vrid <integer>`	VRID used to generate virtual MAC address (1 to 255).
`l3-instance <integer>`	L3VRF EVPN instance ID (0 to 4294967295).
`*interface <string>`	Outgoing interface for all VxLANs in this L3VRF EVPN instance. Available when `type` is set to `ipvrf`.
`distribute-local-route {enable | disable}`	Enable/disable distribution of local EVPN routes as IPv4 routes to BGP.

A new `get` command is available for `default-gateway`:

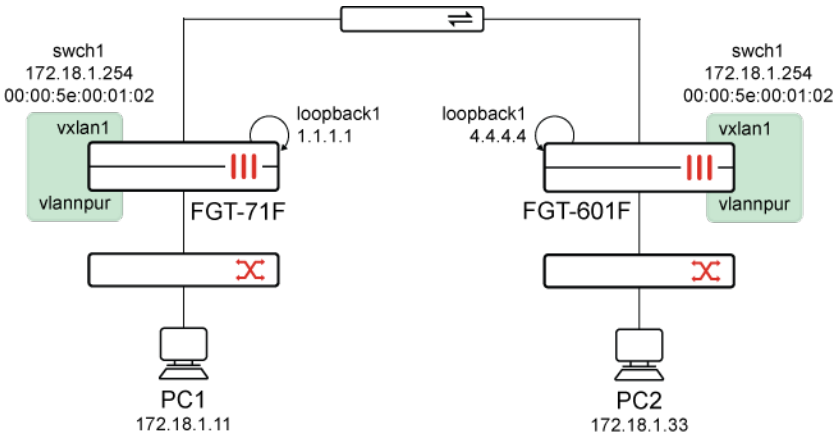
```
get l2vpn evpn default-gateway
```

Example

This section includes the following examples:

- Anycast gateway example on page 337
- IRB example on page 342
- Type-5 route example on page 347

Anycast gateway example



In a Security Fabric configured with EVPN, VXLAN, and an Anycast gateway (swch1), all leaf switches that connect endpoints in the same VLAN (or subnet) share the same virtual default gateway IP address. The leaf switches also share the same virtual MAC address for that gateway.

Endpoints (such as servers, VMs, containers) always see the same gateway regardless of which leaf they connect to. If a VM or host moves from Leaf1 to Leaf2, its default gateway stays the same and doesn't require any reconfiguration (for example re-ARP or a new IP address), which provides seamless mobility.

This example describes how to configure FortiGate-71F and FortiGate-601F in the topology. It also describes how to verify the configuration.

To configure FortiGate-71F:

1. Configure EVPN:

```
config system evpn
    edit 100
        set rd "100:100"
        set import-rt "1:1"
        set export-rt "1:1"
        set ip-local-learning enable
        set adv-default-gw enable
        set virtual-mac-vrid 2
    next
end
```

2. Configure VXLAN:

```
config system vxlan
    edit "vxlan1"
        set interface "loopback1"
        set vni 1000
        set evpn-id 100
        set learn-from-traffic enable
    next
end
```

3. Configure a system switch interface:

```
config system switch-interface
    edit "swch1"
        set vdom "root"
        set member "vxlan1" "vlannpur"
    next
end
```

4. Configure a system interface:

```
config system interface
    edit "swch1"
        set vdom "root"
        set ip 172.18.1.254 255.255.255.0
        set allowaccess ping
```

```
        set type switch
        set snmp-index 26
    next
end
```

5. Configure BGP:

```
config router bgp
    set as 65412
    set router-id 1.1.1.1
    set ibgp-multipath enable
    set network-import-check disable
    set recursive-next-hop enable
    set graceful-restart enable
    config neighbor
        edit "172.18.5.123"
            set ebgp-enforce-multihop enable
            set next-hop-self enable
            set next-hop-self-vpnv4 enable
            set soft-reconfiguration enable
            set soft-reconfiguration-evpn enable
            set remote-as 65412
            set graceful-shutdown-community "22:55"
        next
    end
    config network
        edit 1
            set prefix 1.1.1.1 255.255.255.255
        next
        edit 4
            set prefix 172.28.6.0 255.255.255.0
        next
    end
    config redistribute "connected"
    end
    config redistribute "rip"
    end
    config redistribute "ospf"
    end
    config redistribute "static"
    end
    config redistribute "isis"
    end
    config redistribute6 "connected"
    end
    config redistribute6 "rip"
    end
    config redistribute6 "ospf"
    end
    config redistribute6 "static"
    end
    config redistribute6 "isis"
```

```
    end
end
```

To configure FortiGate-601F:

1. Configure EVPN:

```
config system evpn
    edit 100
        set rd "100:100"
        set import-rt "1:1"
        set export-rt "1:1"
        set ip-local-learning enable
        set adv-default-gw enable
        set virtual-mac-vrid 2
    next
end
```

2. Configure VXLAN:

```
config system vxlan
    edit "vxlan1"
        set interface "loopback1"
        set vni 1000
        set evpn-id 100
        set learn-from-traffic enable
    next
end
```

3. Configure a system switch interface:

```
config system switch-interface
    edit "swch1"
        set vdom "root"
        set member "vxlan1" "vlannpur"
    next
end
```

4. Configure a system interface:

```
config system interface
    edit "swch1"
        set vdom "root"
        set ip 172.18.1.254 255.255.255.0
        set allowaccess ping
        set type switch
        set snmp-index 50
    next
end
```

5. Configure BGP:

```
config router bgp
    set as 65412
    set router-id 4.4.4.4
    set ibgp-multipath enable
    set recursive-next-hop enable
    config neighbor
        edit "172.18.5.121"
            set next-hop-self enable
            set next-hop-self-vpnv4 enable
            set soft-reconfiguration enable
            set soft-reconfiguration-evpn enable
            set remote-as 65412
        next
    end
    config network
        edit 1
            set prefix 4.4.4.4 255.255.255.255
        next
    end
    config redistribute "connected"
    end
    config redistribute "rip"
    end
    config redistribute "ospf"
    end
    config redistribute "static"
    end
    config redistribute "isis"
    end
    config redistribute6 "connected"
    end
    config redistribute6 "rip"
    end
    config redistribute6 "ospf"
    end
    config redistribute6 "static"
    end
end
```

To verify:

1. From FortiGate-601F, ping 172.18.1.11:

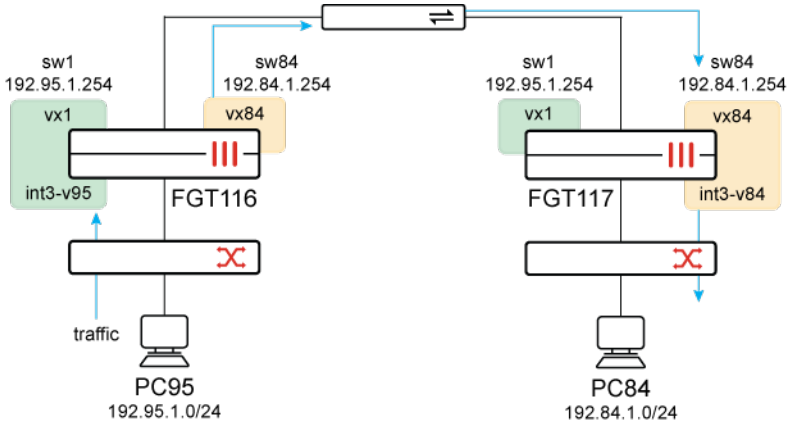
```
# execute ping 172.18.1.11
PING 172.18.1.11 (172.18.1.11): 56 data bytes
64 bytes from 172.18.1.11: icmp_seq=0 ttl=255 time=0.4 ms
64 bytes from 172.18.1.11: icmp_seq=1 ttl=255 time=0.3 ms
^C
--- 172.18.1.11 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.3/0.3/0.4 ms

FortiGate-601F (vdom1) (Interim)# diagnose ip arp list | grep 254
```

```
index=51 ifname=vlannpu1 172.18.1.254 00:00:5e:00:01:02 state=00000004 use=9094399
confirm=9100246 update=9094246 ref=0

FortiGate-601F (vdom1) (Interim)#
```

2. From FortiGate-71F, diagnose the IP ARP list:

```
# diagnose ip arp list
index=27 ifname=vlannpu1 172.18.1.33 86:39:8f:13:f7:01 state=00000002 use=2374 confirm=2374
update=1946 ref=3
index=27 ifname=vlannpu1 172.18.1.254 00:00:5e:00:01:02 state=00000002 use=435 confirm=435
update=435 ref=2
version count: 2
```

IRB example



IRB (Integrated Routing and Bridging) in EVPN means that FortiGate can both bridge (L2) and route (L3) traffic inside the Security Fabric configured with VXLAN/EVPN. EVPN IRB allows L2 services for intra-subnet communication and L3 services for inter-subnet communications.

This example uses the following VLAN, subnets, and VNI names:

PC	VLAN	Subnet	VNI
PC95	vlan95	192.95.1.0/24	vni 141
PC94	vlan84	192.84.1.0/24	vni 84

This example describes how to configure FGT116 and FGT117 in the topology to allow inter-subnet routing between vlan95 and vlan84 over the VXLANs between the two FortiGate VTEPs. This allows PC95 to reach PC84 using EVPN over VXLAN.

To configure FGT116:

1. Configure EVPN:

```
config system evpn
    edit 1
```

```
        set import-rt "1:1"
        set export-rt "1:1"
        set ip-local-learning enable
    next
    edit 84
        set import-rt "84:84"
        set export-rt "84:84"
        set ip-local-learning enable
    next
end
```

2. Configure VXLAN interfaces:

```
config system vxlan
    edit "vx1"
        set interface "loopback1"
        set vni 141
        set evpn-id 1
        set learn-from-traffic enable
    next
     edit "vx84"
        set interface "loopback1"
        set vni 84
        set evpn-id 84
        set learn-from-traffic enable
    next
end
```

3. Configure a system interface:

```
config system interface
    edit "int3-v95"
        set vdom "root"
        set interface "internal3"
        set vlanid 95
    next
end
```

4. Configure system switch interfaces and select the VXLAN interface:

```
config system switch-interface
    edit "sw1"
        set vdom "root"
        set member "int3-v95" "vx1"
        set intra-switch-policy explicit
    next
    edit "sw84"
        set vdom "root"
        set member "vx84"
        set intra-switch-policy explicit
    next
end
```

5. Configure system interfaces:

```
config system interface
    edit "sw1"
        set vdom "root"
        set ip 192.95.1.254 255.255.255.0
        set allowaccess ping ssh telnet
    next
    edit "sw84"
        set vdom "root"
        set ip 192.84.1.254 255.255.255.0
        set allowaccess ping ssh telnet
    next
end
```

6. Configure BGP:

```
config router bgp
    set as 65010
    set router-id 10.255.255.116
    config neighbor
        edit "10.1.2.117"
            set next-hop-self enable
            set soft-reconfiguration-evpn enable
            set remote-as 65010
            set connect-timer 1
        next
    end
    config network
        edit 1
            set prefix 10.255.255.116 255.255.255.255
        next
    end
end
```

To configure FGT117:

1. Configure EVPN:

```
config system evpn
    edit 1
        set import-rt "1:1"
        set export-rt "1:1"
        set ip-local-learning enable
    next
    edit 84
        set import-rt "84:84"
        set export-rt "84:84"
        set ip-local-learning enable
    next
end
```

2. Configure VXLAN interfaces:

```
config system vxlan
    edit "vx1"
        set interface "loopback1"
        set vni 141
        set evpn-id 1
        set learn-from-traffic enable
    next
    edit "vx84"
        set interface "loopback1"
        set vni 84
        set evpn-id 84
        set learn-from-traffic enable
    next
end
```

3. Configure a system switch interface and select the VXLAN interfaces:

```
config system switch-interface
    edit "sw1"
        set vdom "root"
        set member "vx1"
        set intra-switch-policy explicit
    next
    edit "sw84"
        set vdom "root"
        set member "int3-v84"  "vx84"
        set intra-switch-policy explicit
    next
end
```

4. Configure system interfaces:

```
config system interface
    edit "sw1"
        set vdom "root"
        set ip 192.95.1.254 255.255.255.0
        set allowaccess ping ssh telnet
    next
    edit "sw84"
        set vdom "root"
        set ip 192.84.1.254 255.255.255.0
        set allowaccess ping ssh telnet
    next
end
```

5. Configure BGP:

```
config router bgp
    set as 65010
    set router-id 10.255.255.117
    config neighbor
        edit "10.1.2.116"
```

```
            set next-hop-self enable
            set soft-reconfiguration-evpn enable
            set remote-as 65010
            set connect-timer 1
        next
    end
    config network
        edit 1
            set prefix 10.255.255.117 255.255.255.255
        next
    end
end
```

To verify:

1. On FGT116, view the routing table. The FortiGate learned both the subnet of vlan95 and vlan84:

```
# get router info routing-table  all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       V - BGP VPNv4, E - BGP EVPN
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 172.18.18.1, wan1, [1/0]
C       10.1.2.0/24 is directly connected, wan2
C       10.255.255.116/32 is directly connected, loopback1
B       10.255.255.117/32 [200/0] via 10.1.2.117 (recursive is directly connected, wan2),
20:28:32, [1/0]
C       172.18.18.0/24 is directly connected, wan1
C       192.84.1.0/24 is directly connected, sw84
C       192.95.1.0/24 is directly connected, sw1
C       192.96.0.0/16 is directly connected, int3-v96
```

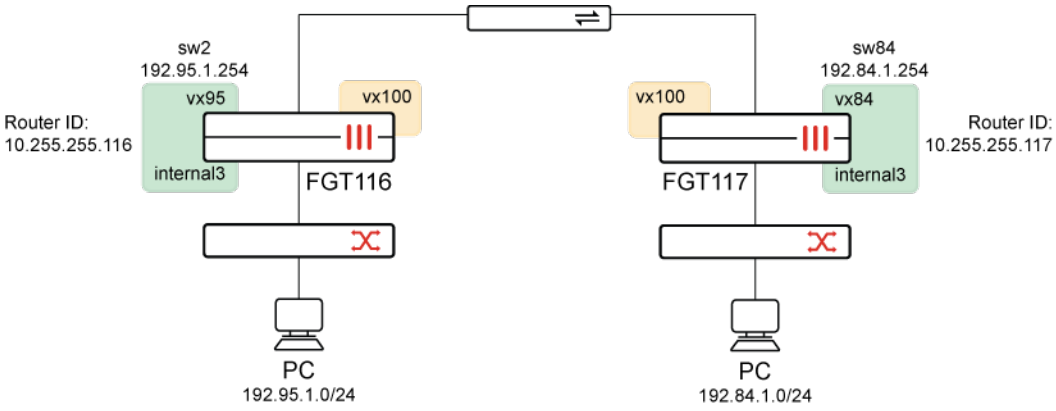
2. Look at the L2 EVPN table for VNI 84, FGT116 has learned the MAC address for PC84.
   The MAC address for PC84 is 00:0c:29:da:ca:07 84.

```
# get l2vpn evpn table 84

EVPN instance 84
 Broadcast domain VNI 84 TAGID 0

  EVPN MAC table:
  MAC               VNI      Remote Addr     Binded Address
  00:0c:29:da:ca:07 84       10.255.255.117  -
  00:11:92:d4:5e:8f 84       10.255.255.117  -
  04:d5:90:b1:9d:e8 84       10.255.255.117  -
  6a:a5:45:75:d5:65 84       10.255.255.117  -
```

```
EVPN Local MAC table:
"Inactive" means this MAC/IP pair will not be sent to peer.
Flag code: S - Static, F - FDB, L - Switch local MAC, G - Default Gateway.
Trailing * means HA
MAC               Flag Status   Binded Address
02:4c:63:08:3d:39 LG   Inactive 192.84.1.254
                  L    Active   -

EVPN Local IP table:
Address         MAC
192.84.1.254    02:4c:63:08:3d:39

EVPN PEER table:
VNI      Remote Addr     Binded Address
84       10.255.255.117  10.255.255.117
```

Type-5 route example

Instead of advertising MAC/IP bindings used in type-2 routes, type-5 EVPN routes advertise IP prefixes (subnets), allowing EVPN to carry L3 VPN routes across the Security Fabric configured with VXLAN to provide more than only L2 reachability.



This example uses the following VLAN and VNI names and IDs:

PC and L3 instance	VLAN	VNI
PC95	vlan95	vni 95
PC94	vlan84	vni 84
L3 instance	-	vni 100

This example describes how to configure FGT116 and FGT117 in the topology. It also describes how to verify the configuration.

To configure FGT116:

1. Configure EVPN:

```
config system evpn
    edit 95
        set import-rt "95:95"
        set export-rt "95:95"
        set l3-instance 100
    next
    edit 100
        set type ipvrf
        set rd "100:100"
        set import-rt "100:100"
        set export-rt "100:100"
        set interface "loopback1"
    next
end
```

2. Configure VXLAN interfaces:

```
config system vxlan
    edit "vx95"
        set vni 95
        set evpn-id 95
        set learn-from-traffic enable
    next
    edit "vx100"
        set vni 100
        set evpn-id 100
    next
end
```

3. Configure the system switch interface:

```
config system switch-interface
    edit "sw2"
        set vdom "root"
        set member "vx95" "internal3"
        set intra-switch-policy explicit
    next
end
```

4. Configure the system interface:

```
config system interface
    edit "sw2"
        set vdom "root"
        set ip 192.95.1.254 255.255.0.0
        set allowaccess ping
        set type switch
    next
end
```

5. Configure BGP:

```
config router bgp
    set as 65010
    set router-id 10.255.255.116
    config neighbor
        edit "10.1.2.117"
            set next-hop-self enable
            set soft-reconfiguration-evpn enable
            set remote-as 65010
            set connect-timer 1
        next
    end
    config network
        edit 1
            set prefix 10.255.255.116 255.255.255.255
        next
    end
    config redistribute "connected"
        set status evpn enable
    end
end
```

To configure FGT117:

1. Configure EVPN:

```
config system evpn
    edit 100
        set type ipvrf
        set rd "100:100"
        set import-rt "100:100"
        set export-rt "100:100"
        set interface "loopback1"
    next
    edit 84
        set import-rt "84:84"
        set export-rt "84:84"
        set l3-instance 100
    next
end
```

2. Configure VXLAN interfaces:

```
config system vxlan
    edit "vx84"
        set vni 84
        set evpn-id 84
        set learn-from-traffic enable
    next
    edit "vx100"
        set vni 100
```

```
        set evpn-id 100
        set learn-from-traffic enable
    next
end
```

3. Configure the system switch interface:

```
config system switch-interface
    edit "sw84"
        set vdom "root"
        set member "vx84" "internal3"
        set intra-switch-policy explicit
    next
end
```

4. Configure the system interface:

```
config system interface
    edit "sw84"
        set vdom "root"
        set ip 192.84.1.254 255.255.0.0
        set allowaccess ping
        set type switch
    next
end
```

To verify:

1. On FGT117, get router info for the BGP EVPN network:
   Type-5 routes are in bold.

```
FGT117# get router info bgp evpn network
<string>    L2VPN EVPN prefix

FGT117# get router info bgp evpn network
   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
Route Distinguisher: 100:100 (Default for VRF 0)
*> [5][0][16][192.84.0.0]/72
                    10.255.255.117      0             100  32768        0 ? <-/1>
*> [5][0][16][192.95.0.0]/72
                    10.255.255.116      0             100      0        0 ? <-/1>
*> [5][0][16][192.96.0.0]/72
                    10.255.255.116      0             100      0        0 ? <-/1>
*  [5][0][16][192.96.0.0]/72
                    10.255.255.117      0             100  32768        0 ? <-/->
*> [5][0][24][10.1.2.0]/72
                    10.255.255.116      0             100      0        0 ? <-/1>
*  [5][0][24][10.1.2.0]/72
                    10.255.255.117      0             100  32768        0 ? <-/->
*> [5][0][24][172.18.18.0]/72
                    10.255.255.116      0             100      0        0 ? <-/1>
*  [5][0][24][172.18.18.0]/72
```

```
                    10.255.255.117      0             100  32768        0 ? <-/->
*> [5][0][32][10.255.255.116]/72
                    10.255.255.116      0             100      0        0 ? <-/1>
*> [5][0][32][10.255.255.117]/72
                    10.255.255.117      0             100  32768        0 ? <-/1>

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
Route Distinguisher: 100:100 (received from VRF 0)
*>i[5][0][16][192.95.0.0]/72
                    10.255.255.116      0             100      0        0 ? <-/1>
*>i[5][0][16][192.96.0.0]/72
                    10.255.255.116      0             100      0        0 ? <-/1>
*>i[5][0][24][10.1.2.0]/72
                    10.255.255.116      0             100      0        0 ? <-/1>
*>i[5][0][24][172.18.18.0]/72
                    10.255.255.116      0             100      0        0 ? <-/1>
*>i[5][0][32][10.255.255.116]/72
                    10.255.255.116      0             100      0        0 ? <-/1>

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
Route Distinguisher: 0.0.0.0:84 (Default for VRF 0)
*> [2][0][48][00:0c:29:da:ca:07][0]/72
                    10.255.255.117      0             100  32768        0 i <-/1>
*> [2][0][48][00:11:92:d4:5e:8f][0]/72
                    10.255.255.117      0             100  32768        0 i <-/1>
*> [2][0][48][04:d5:90:b1:9d:e8][0]/72
                    10.255.255.117      0             100  32768        0 i <-/1>
*> [2][0][48][0e:61:aa:b2:bb:c3][0]/72
                    10.255.255.117      0             100  32768        0 i <-/1>
*> [3][0][32][10.255.255.117]/80
                    10.255.255.117      0             100  32768        0 i <-/1>
```

2. Get router info for the BGP EVPN network:

```
FGT117# get router info bgp evpn network [5][0][16][192.95.0.0]/72
Route Distinguisher: 100:100 (Default for VRF 0)
BGP routing table entry for [5][0][16][192.95.0.0]/72
  Original VRF 0
  Local
    10.255.255.116 from 10.1.2.116 (10.255.255.116)
      Origin incomplete distance 200 metric 0, localpref 100, valid, internal, evpn, best
      Extended Community: RT:100:100  RMAC:d2:22:98:41:e5:ba
      RD [100:100]
      LABEL1 100
      Last update: Fri Apr 11 14:08:56 2025

Route Distinguisher: 100:100 (received from VRF 0)
BGP routing table entry for [5][0][16][192.95.0.0]/72
  Original VRF 0
  Local
    10.255.255.116 from 10.1.2.116 (10.255.255.116)
      Origin incomplete distance 200 metric 0, localpref 100, valid, internal, evpn, best
```

```
        Extended Community: RT:100:100  RMAC:d2:22:98:41:e5:ba
        RD [100:100]
        LABEL1 100
        Last update: Fri Apr 11 14:08:56 2025
```

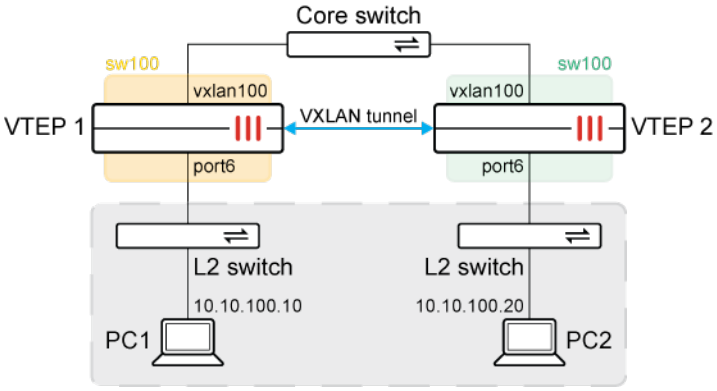
VXLAN troubleshooting

The following commands can be used to troubleshoot VXLAN connectivity:

- `diagnose sys vxlan fdb list <VXLAN_interface>`
- `diagnose sys vxlan fdb stat <VXLAN_interface>`
- `diagnose netlink brctl name host <switch_interface>`
- `diagnose sniffer packet any 'udp and port 4789' 4 0 l`
- `diagnose debug enable`
- `diagnose debug flow filter port 4789`
- `diagnose debug flow trace start <repeat_#>`

Topology

The following topology is used as an example configuration to demonstrate VXLAN troubleshooting steps.



In this example, two FortiGates are configured as VXLAN tunnel endpoints (VTEPs). A VXLAN is configured to allow L2 connectivity between the networks behind each FortiGate. The VXLAN interface and port6 are placed on the same L2 network using a software switch (sw100). An L2 network is formed between PC1 and PC2.

The VTEPs have the following MAC address tables:

Interface/endpoint	VTEP 1	VTEP 2
vxlan100	7e:f2:d1:84:75:0f	ca:fa:31:23:8d:c1

Interface/endpoint	VTEP 1	VTEP 2
port6	00:0c:29:4e:5c:1c	00:0c:29:d0:3e:0d
sw100	00:0c:29:4e:5c:1c	00:0c:29:d0:3e:0d

The MAC address of PC1 is 00:0c:29:90:4f:bf. The MAC address of PC2 is 00:0c:29:f0:88:2c.

To configure the VTEP 1 FortiGate:

1. Configure the local interface:

```
config system vxlan
    edit "vxlan100"
        set interface "port2"
        set vni 100
        set remote-ip "192.168.2.87"
    next
end
```

2. Configure the interface settings:

```
config system interface
    edit "port2"
        set vdom "root"
        set ip 192.168.2.86 255.255.255.0
        set allowaccess ping https ssh http fabric
    next
    edit "vxlan100"
        set vdom "root"
        set type vxlan
        set interface "port2"
    next
end
```

3. Configure the software switch:

```
config system switch-interface
    edit "sw100"
        set vdom "root"
        set member "port6" "vxlan100"
    next
end
```

4. Configure the software switch interface settings:

```
config system interface
    edit "sw100"
        set vdom "root"
        set ip 10.10.100.86 255.255.255.0
        set allowaccess ping
        set type switch
        set device-identification enable
```

```
        set lldp-transmission enable
        set role lan
    next
end
```

To configure the VTEP 2 FortiGate:

1. Configure the local interface:

```
config system vxlan
    edit "vxlan100"
        set interface "port2"
        set vni 100
        set remote-ip "192.168.2.86"
    next
end
```

2. Configure the interface settings:

```
config system interface
    edit "port2"
        set vdom "root"
        set ip 192.168.2.87 255.255.255.0
        set allowaccess ping https ssh snmp http
    next
    edit "vxlan100"
        set vdom "root"
        set type vxlan
        set interface "port2"
    next
end
```

3. Configure the software switch:

```
config system switch-interface
    edit "sw100"
        set vdom "root"
        set member "port6" "vxlan100"
    next
end
```

4. Configure the software switch interface settings:

```
config system interface
    edit "sw100"
        set vdom "root"
        set ip 10.10.100.87 255.255.255.0
        set allowaccess ping
        set type switch
        set device-identification enable
        set lldp-transmission enable
        set role lan
```

```
        set snmp-index 42
    next
end
```

To run diagnostics and debugs:

1. Start a ping from PC1 10.10.100.10 to PC2 10.10.100.20:

```
C:\Users\fortidocs>ping 10.10.100.20

Pinging 10.10.100.20 with 32 bytes of data:
Reply from 10.10.100.20: bytes=32 time=2ms TTL=128
Reply from 10.10.100.20: bytes=32 time=1ms TTL=128
Reply from 10.10.100.20: bytes=32 time=1ms TTL=128
Reply from 10.10.100.20: bytes=32 time<1ms TTL=128

Ping statistics for 10.10.100.20:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 1ms
```

2. Verify the ARP table:

```
C:\Users\fortidocs>arp /a

Interface: 10.10.100.10 --- 0x21
  Internet Address      Physical Address      Type
  10.10.100.20          00-0c-29-f0-88-2c     dynamic
  10.10.100.86          00-0c-29-4e-5c-1c     dynamic
  10.10.100.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
```

3. Run diagnostics on the VTEP 1 FortiGate.
   a. Verify the forwarding database of VXLAN interface vxlan100:

```
# diagnose sys vxlan fdb list vxlan100
mac=00:00:00:00:00:00 state=0x0082 remote_ip=192.168.2.87 port=4789 vni=100 ifindex=6
mac=00:0c:29:f0:88:2c state=0x0002 remote_ip=192.168.2.87 port=4789 vni=100 ifindex=6

total fdb num: 2
```

   The MAC address 00:0c:29:f0:88:2c is learned from PC2 10.10.100.20.

   b. Verify the summary of statistics from the VXLAN's forwarding database:

```
# diagnose sys vxlan fdb stat vxlan100
fdb_table_size=256 fdb_table_used=2 fdb_entry=2 fdb_max_depth=1 cleanup_idx=0 cleanup_
timer=252
```

   c. Verify the software switch's forwarding table:

```
# diagnose netlink brctl name host sw100
show bridge control interface sw100 host.
fdb: hash size=32768, used=6, num=6, depth=1, gc_time=4, ageing_time=3, simple=switch
Bridge sw100 host table
port no device  devname mac addr                ttl     attributes
  1     7       port6   00:0c:29:4e:5c:1c       0       Local Static
  2     33      vxlan100        7e:f2:d1:84:75:0f       0       Local Static
  2     33      vxlan100        00:00:00:00:00:00       26       Hit(26)
  1     7       port6   00:0c:29:90:4f:bf       0        Hit(0)
  1     7       port6   00:0c:29:d0:3e:ef       7        Hit(7)
  2     33      vxlan100        00:0c:29:f0:88:2c       0        Hit(0)
```

The MAC address of port6 is 00:0c:29:4e:5c:1c. The MAC address of vxlan100 is 7e:f2:d1:84:75:0f. The MAC address 00:0c:29:f0:88:2c of PC2 is learned from the remote network.

4. Run diagnostics on the VTEP 2 FortiGate.
   a. Verify the forwarding database of VXLAN interface vxlan100:

```
# diagnose sys vxlan fdb list vxlan100
mac=00:00:00:00:00:00 state=0x0082 remote_ip=192.168.2.86 port=4789 vni=100 ifindex=6
mac=00:0c:29:90:4f:bf state=0x0002 remote_ip=192.168.2.86 port=4789 vni=100 ifindex=6

total fdb num: 2
```

   The MAC address 00:0c:29:90:4f:bf is learned from PC1 10.10.100.10.

   b. Verify the summary of statistics from the VXLAN's forwarding database:

```
# diagnose sys vxlan fdb stat vxlan100
fdb_table_size=256 fdb_table_used=2 fdb_entry=2 fdb_max_depth=1 cleanup_idx=0 cleanup_
timer=304
```

   c. Verify the software switch's forwarding table:

```
# diagnose netlink brctl name host sw100
show bridge control interface sw100 host.
fdb: hash size=32768, used=5, num=5, depth=1, gc_time=4, ageing_time=3, simple=switch
Bridge sw100 host table
port no device  devname mac addr                ttl     attributes
  2     50      vxlan100        00:00:00:00:00:00       10       Hit(10)
  2     50      vxlan100        00:0c:29:90:4f:bf       2        Hit(2)
  1     7       port6   00:0c:29:d0:3e:0d       0       Local Static
  2     50      vxlan100        ca:fa:31:23:8d:c1       0       Local Static
  1     7       port6   00:0c:29:f0:88:2c       0        Hit(0)
```

   The MAC address of port6 is 00:0c:29:d0:3e:0d. The MAC address of vxlan100 is ca:fa:31:23:8d:c1. The MAC address 00:0c:29:90:4f:bf of PC1 is learned from the remote network.

5. Perform a sniffer trace on the VTEP 1 FortiGate to view the life of the packets as they pass through the FortiGate:

```
# diagnose sniffer packet any 'host 10.10.100.20 or (udp and host 192.168.2.87)' 4 0 l
Using Original Sniffing Mode
interfaces=[any]
```

```
filters=[host 10.10.100.20 or (udp and host 192.168.2.87)]
2022-11-04 14:35:18.567602 port6 in arp who-has 10.10.100.20 tell 10.10.100.10
2022-11-04 14:35:18.567629 vxlan100 out arp who-has 10.10.100.20 tell 10.10.100.10
2022-11-04 14:35:18.567642 port2 out 192.168.2.86.4804 -> 192.168.2.87.4789: udp 68
2022-11-04 14:35:18.567658 sw100 in arp who-has 10.10.100.20 tell 10.10.100.10
2022-11-04 14:35:18.568239 port2 in 192.168.2.87.4789 -> 192.168.2.86.4789: udp 68
2022-11-04 14:35:18.568263 vxlan100 in arp reply 10.10.100.20 is-at 00:0c:29:f0:88:2c
2022-11-04 14:35:18.568272 port6 out arp reply 10.10.100.20 is-at 00:0c:29:f0:88:2c
2022-11-04 14:35:18.568425 port6 in 10.10.100.10 -> 10.10.100.20: icmp: echo request
2022-11-04 14:35:18.568435 vxlan100 out 10.10.100.10 -> 10.10.100.20: icmp: echo request
2022-11-04 14:35:18.568443 port2 out 192.168.2.86.4805 -> 192.168.2.87.4789: udp 82
2022-11-04 14:35:18.568912 port2 in 192.168.2.87.4789 -> 192.168.2.86.4789: udp 68
2022-11-04 14:35:18.568925 vxlan100 in arp who-has 10.10.100.10 tell 10.10.100.20
2022-11-04 14:35:18.568935 port6 out arp who-has 10.10.100.10 tell 10.10.100.20
2022-11-04 14:35:18.568945 sw100 in arp who-has 10.10.100.10 tell 10.10.100.20
2022-11-04 14:35:18.569070 port6 in arp reply 10.10.100.10 is-at 00:0c:29:90:4f:bf
2022-11-04 14:35:18.569076 vxlan100 out arp reply 10.10.100.10 is-at 00:0c:29:90:4f:bf
2022-11-04 14:35:18.569081 port2 out 192.168.2.86.4806 -> 192.168.2.87.4789: udp 68
2022-11-04 14:35:18.569417 port2 in 192.168.2.87.4789 -> 192.168.2.86.4789: udp 82
2022-11-04 14:35:18.569427 vxlan100 in 10.10.100.20 -> 10.10.100.10: icmp: echo reply
2022-11-04 14:35:18.569431 port6 out 10.10.100.20 -> 10.10.100.10: icmp: echo reply
```

In the output, the following packet sequence is seen on the FortiGate:

a. The FortiGate receives an ARP request from PC1 10.10.100.10 on port6.

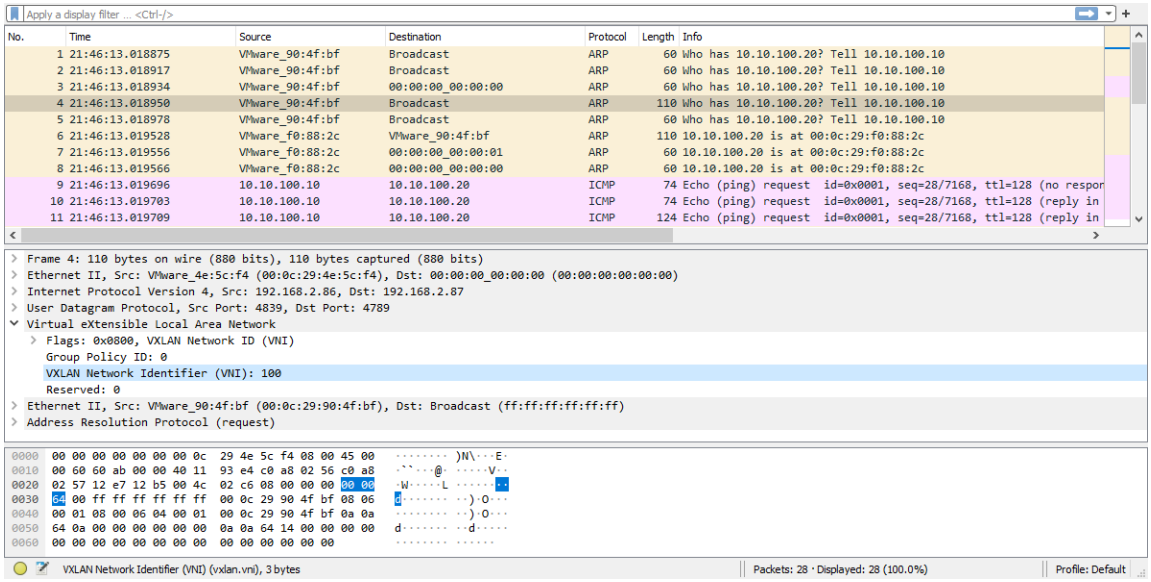
b. The ARP request is forwarded to vxlan100 on the same software switch, where it gets encapsulated and sent out as a UDP port 4789 packet on port2.

c. A reply is received on port2 from the remote VTEP with the ARP response encapsulated in UDP port 4789 again.

d. The ARP reply is forwarded back out of port6 to PC1.

e. PC1 sends the ICMP request using the same steps.

6. Perform the same sniffer trace filter with a level 6 verbose level. In this example, the packet capture is converted into a Wireshark file.



The packet that leaves the physical port2 is encapsulated in UDP and has a VXLAN header with VNI 100 as the identifier. There is an additional 50 B overhead of the UDP encapsulated VXLAN packets as opposed to the unencapsulated packets (for example, packet 4 versus packets 1 and 2).

DNS

Domain name system (DNS) is used by devices to locate websites by mapping a domain name to a website's IP address.

A FortiGate can serve different roles based on user requirements:

- A FortiGate can control what DNS server a network uses.
- A FortiGate can function as a DNS server.

FortiGuard Dynamic DNS (DDNS) allows a remote administrator to access a FortiGate's Internet-facing interface using a domain name that remains constant even when its IP address changes.

FortiOS supports DNS configuration for both IPv4 and IPv6 addressing. When a user requests a website, the FortiGate looks to the configured DNS servers to provide the IP address of the website in order to know which server to contact to complete the transaction.

The FortiGate queries the DNS servers whenever it needs to resolve a domain name into an IP address, such as for NTP or web servers defined by their domain names.

The following topics provide information about DNS:

- Important DNS CLI commands on page 358
- DNS domain list on page 363
- FortiGate DNS server on page 365
- DDNS on page 385
- DNS latency information on page 390
- DNS over TLS and HTTPS on page 392
- Transparent conditional DNS forwarder on page 396
- Interfaces in non-management VDOMs as the source IP address of the DNS conditional forwarding server on page 400
- DNS session helpers on page 402
- DNS troubleshooting on page 404

Important DNS CLI commands

DNS settings can be configured with the following CLI command:

```
config system dns
    set primary <ip_address>
    set secondary <ip_address>
    set protocol {cleartext dot doh}
    set ssl-certificate <string>
```

```
    set server-hostname <hostname>
    set domain <domains>
    set ip6-primary <ip6_address>
    set ip6-secondary <ip6_address>
    set timeout <integer>
    set retry <integer>
    set dns-cache-limit <integer>
    set dns-cache-ttl <integer>
    set cache-notfound-responses {enable | disable}
    set interface-select-method {auto | sdwan | specify}
    set interface <interface>
    set source-ip <class_ip>
    set source-ip-interface <string>
    set root-servers <DNS root name server IP address>
    set server-select-method {least-rtt | failover}
    set alt-primary <ip_address>
    set alt-secondary <ip_address>
    set log {disable |error | all}
    set fqdn-cache-ttl <integer>
    set fqdn-min-refresh <integer>
    set fqdn-max-refresh <integer>
end
```

For a FortiGate with multiple logical CPUs, you can set the DNS process number from 1 to the number of logical CPUs. The default DNS process number is 1.

```
config system global
    set dnsproxy-worker-count <integer>
end
```

DNS protocols

The following DNS protocols can be enabled:

- `cleartext`: Enable clear text DNS over port 53 (default).
- `dot`: Enable DNS over TLS.
- `doh`: Enable DNS over HTTPS.

For more information, see DNS over TLS and HTTPS on page 392.

cache-notfound-responses

When enabled, any DNS requests that are returned with `NOT FOUND` can be stored in the cache. The DNS server is not asked to resolve the host name for `NOT FOUND` entries. By default, this option is disabled.

dns-cache-limit

Set the number of DNS entries that are stored in the cache (0 to 4294967295, default = 5000). Entries that remain in the cache provide a quicker response to requests than going out to the Internet to get the same

information.

dns-cache-ttl

The duration that the DNS cache retains information, in seconds (60 to 86400 (1 day), default = 1800).

fqdn-cache-ttl

FQDN cache time to live (TTL), in seconds (0 - 86400, default = 0).

This is the amount of time an FQDN's address record can live if not refreshed. This setting applies globally, across all VDOMs, to FQDNs that have unspecified firewall address `cache-ttl` settings. If the `cache-ttl` value is configured for an FQDN address, it will supersede the `fqdn-cache-ttl` setting for that address.

For example, configure the FQDN cache TTL on the global VDOM:

```
config system dns
    set fqdn-cache-ttl 2000
end
```

```
# diagnose test application dnsproxy 6
```

```
...
vfid=0 name=test.bb.com ver=IPv4 wait_list=0 timer=985 min_ttl=1000 cache_ttl=2000 slot=-1 num=1
wildcard=0
         1.1.1.1 (ttl=1000:991:1991)
vfid=0 name=*.google.com ver=IPv4 wait_list=0 timer=0 min_ttl=0 cache_ttl=2000 slot=-1 num=0
wildcard=
...
```

Change the cache TTL in a VDOM for a specific address:

```
config firewall address
    edit "test.bb.com"
        set cache-ttl 1000
    next
end
```

```
# sudo global diagnose test application dnsproxy 6
...
vfid=0 name=test.bb.com ver=IPv4 wait_list=0 timer=864 min_ttl=1000 cache_ttl=1000 slot=-1 num=1
wildcard=0
1.1.1.1 (ttl=1000:870:1870)
vfid=0 name=*.google.com ver=IPv4 wait_list=0 timer=0 min_ttl=0 cache_ttl=2000 slot=-1 num=0
wildcard=1
...
```

fqdn-min-refresh

FQDN cache minimum refresh time, in seconds (10 - 3600, default = 60).

An FQDN normally requeries for updates according to the lowest TTL interval returned from all the DNS records in a DNS response. The FortiGate has a default minimum refresh interval of 60 seconds; if a TTL interval is shorter than 60 seconds, it still requires a minimum of 60 seconds for the FortiGate to requery for new addresses. The `fqdn-min-refresh` setting changes the interval. The settings could be shortened if there are FQDNs that require fast resolutions based on a short TTL interval.

For example, if `fqdn_min_refresh` is unspecified:

```
# diagnose test application dnsproxy 3
worker idx: 0
...
FQDN: min_refresh=60 max_refresh=3600
...
```

```
# diagnose test application dnsproxy 6
worker idx: 0
vfid=0 name=aa.com ver=IPv4 wait_list=0 timer=28 min_ttl=20 cache_ttl=0 slot=-1 num=1 wildcard=0
         23.202.195.114 (ttl=20:0:0)
```

The `min_refresh` is the default value of 60 seconds. Although the `min_ttl` (TTL returned) value is shorter, the FortiGate only requeries for updates based on the `min_refresh` value. the `timer` value is the countdown until the next refresh is triggered. The FortiGate triggers a refresh slightly earlier than the larger of the `min_refresh` or `min_ttl` value.

If `fqdn_min_refresh` is configured:

```
config system dns
    set fqdn-min-refresh 20
end
```

```
# diagnose test application dnsproxy 3
worker idx: 0
...
FQDN: min_refresh=20 max_refresh=3600
...
```

```
# diagnose test application dnsproxy 6
worker idx: 0
vfid=0 name=aa.com ver=IPv4 wait_list=0 timer=8 min_ttl=20 cache_ttl=0 slot=-1 num=1 wildcard=0
         23.202.195.114 (ttl=20:14:14)
```

This setting can be used in combination with `fqdn-cache-ttl` and `cache-ttl` to send more frequent queries and store more resolved addresses in cache. This is useful in scenarios where the FQDN has many resolutions and changes very frequently.

fqdn-max-refresh

FQDN cache maximum refresh time, in seconds (3600 - 86400, default = 3600).

The `fqdn-max-refresh` setting is used to control the global upper limit of the FQDN refresh timer. FQDN entries with a TTL interval that is longer than the `fqdn-max-refresh` value will have their refresh timer reduced to this

upper limit. This allows the FortiGate to dictate the upper limit in querying for DNS updates for its FQDN addresses.

VDOM DNS

When the FortiGate is in multi-vdom mode, DNS is handled by the management VDOM. However in some cases, administrators may want to configure custom DNS settings on a non-management VDOM. For example, in a multi-tenant scenario, each VDOM might be occupied by a different tenant, and each tenant might require its own DNS server. For more information on VDOM DNS, see Interfaces in non-management VDOMs as the source IP address of the DNS conditional forwarding server on page 400.

To configure a custom VDOM within a non-management VDOM:

```
config vdom
    edit <vdom>
        config system vdom-dns
            set vdom-dns enable
            set primary <primary_DNS>
            set secondary <secondary_DNS>
            set protocol {cleartext dot doh}
            set ip6-primary <primary_IPv6_DNS>
            set ip6-secondary <secondary_IPv6_DNS>
            set source-ip <IP_address>
            set interface-select-method {auto | sdwan | specify}
        end
    next
end
```

Source IP interface

Use the `source-ip-interface` option to synchronize the DNS setting across a cluster.

In this example, a private DNS is used. Port2 is configured with an IP address, and the private DNS is configured to use the IP address for port2 as its source IP address.

To set the source IP interface for a private DNS:

1. Configure port2 with an IP address. You can either specify an IP address or configure the interface to receive an IP address from a DHCP server.

   Example fixed IP address configuration:

   ```
   config system interface
       edit "port2"
           set vdom "vdom1"
           set ip 10.1.100.1 255.255.255.0
           set allowaccess ping https ssh snmp http telnet
           set type physical
           set snmp-index 4
   ```

```
        set secondary-IP enable
    next
end
```

Example DHCP configuration:

```
config system interface
     edit "port2"
         set mode dhcp
     next
end
```

2. Configure port2 as the source IP interface for DNS:

```
config system dns
    set primary 172.17.254.148
    set secondary 172.17.254.151
    set source-ip-interface "port2"
end
```

3. Sniff port2:

```
# diagnose sniffer packet port2 ""
....
3.336987 10.1.100.1.2264 -> 172.17.254.148.53: udp 43
```

DNS domain list

You can configure up to eight domains in the DNS settings using the GUI or the CLI.

When a FortiGate requests a URL that does not include an FQDN, FortiOS resolves the URL by traversing through the DNS domain list and performing a query for each domain until the first match is found.

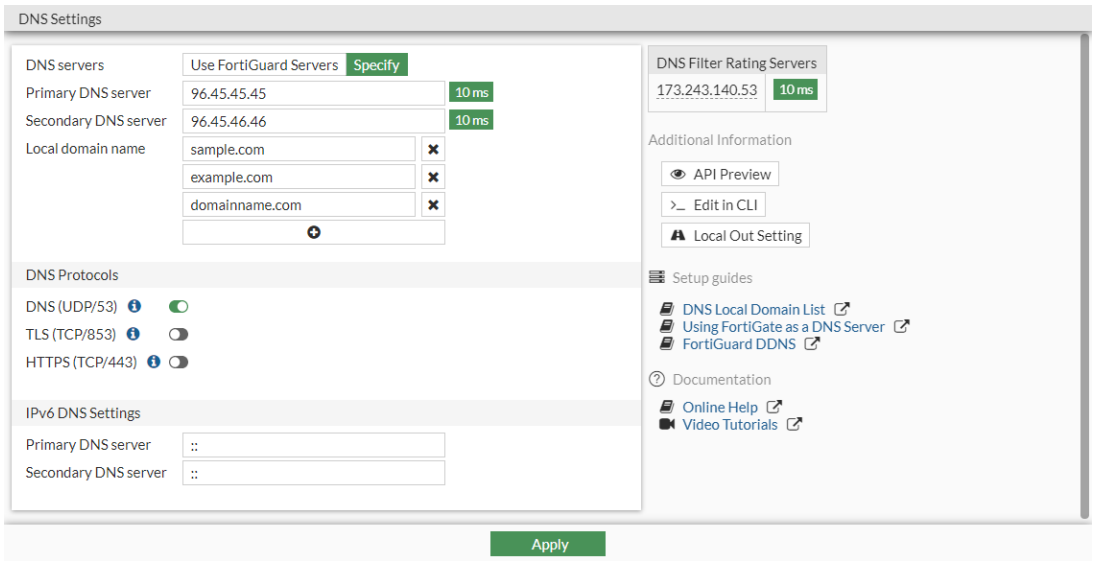
By default, FortiGates use FortiGuard's DNS servers:

- Primary: 96.45.45.45
- Secondary: 96.45.46.46

You can also customize the DNS timeout time and the number of retry attempts.

To configure a DNS domain list in the GUI:

1. Go to *Network* > *DNS*.
2. Set *DNS Servers* to *Specify*.
3. Configure the primary and secondary DNS servers as needed.
4. In the *Local Domain Name* field, enter the first domain (`sample.com` in this example).
5. Click the *+* to add more domains (`example.com` and `domainname.com` in this example). You can enter up to eight domains.
6. Configure additional DNS protocol and IPv6 settings as needed.



7. Click *Apply*.

To configure a DNS domain list in the CLI:

```
config system dns
    set primary 96.45.45.45
    set secondary 96.45.46.46
    set domain "sample.com" "example.com" "domainname.com"
end
```

Verify the DNS configuration

In the following example, the local DNS server has the entry for *host1* mapped to the FQDN of *host1.sample.com*, and the entry for *host2* is mapped to the FQDN of *host2.example.com*.

To verify that the DNS domain list is configured:

1. Open the FortiGate CLI.
2. Enter `execute ping host1`.
   The system returns the following response:

   ```
   PING host1.sample.com (1.1.1.1): 56 data bytes
   ```

   As the request does not include an FQDN, FortiOS traverses the configured DNS domain list to find a match. Because *host1* is mapped to the *host1.sample.com*, FortiOS resolves *host1* to *sample.com*, the first entry in the domain list.
3. Enter `execute ping host2`.
   The system returns the following response:

   ```
   PING host2.example.com (2.2.2.2): 56 data bytes
   ```

FortiOS traverses the domain list to find a match. It first queries sample.com, the first entry in the domain list, but does not find a match. It then queries the second entry in the domain list, example.com. Because *host2* is mapped to the FQDN of *host2.example.com*, FortiOS resolves *host2* to *example.com*.

DNS timeout and retry settings

The DNS timeout and retry settings can be customized using the CLI.

```
config system dns
    set timeout <integer>
    set retry <integer>
end
```

`timeout <integer>`	The DNS query timeout interval, in seconds (1 - 10, default = 5).
`retry <integer>`	The number of times to retry the DNS query (0 - 5, default - 2).

FortiGate DNS server

You can create local DNS servers for your network. Depending on your requirements, you can either manually maintain your entries (primary DNS server), or use it to refer to an outside source (secondary DNS server).

A local, primary DNS server requires that you to manually add all URL and IP address combinations. Using a primary DNS server for local services can minimize inbound and outbound traffic, and access time. Making it authoritative is not recommended, because IP addresses can change, and maintaining the list can become labor intensive.

A secondary DNS server refers to an alternate source to obtain URL and IP address combinations. This is useful when there is a primary DNS server where the entry list is maintained.

FortiGate as a DNS server also supports TLS and HTTPS connections to a DNS client. See DNS over TLS and HTTPS on page 392 for details.

DNS over QUIC (DoQ) and DNS over HTTP3 (DoH3) are supported in proxy mode inspection for transparent and local-in explicit modes. See DNS over QUIC and DNS over HTTP3 for transparent and local-in DNS modes on page 2416 for details.

See Basic DNS server configuration example on page 369 for a sample configuration.

By default, DNS server options are not available in the FortiGate GUI.

To enable DNS server options in the GUI:

1. Go to *System* > *Feature Visibility*.
2. Enable *DNS Database* in the *Additional Features* section.
3. Click *Apply*.

To configure the FortiGate as a DNS server in the GUI:

1. Go to *Network* > *DNS Servers*.
2. Enable DNS services on an interface:
   a. In the *DNS Service on Interface* table, click *Create New*.
   b. Configure the following:

Interface	Select the interface to enable DNS service on.
Mode	Set the DNS server mode: • *Recursive*: The system first checks for the requested record in the shadow DNS database. If the record is not found locally, the query is then forwarded to the system's DNS server for further lookup. This mode ensures a comprehensive search for the requested record, utilizing both local and system DNS resources. • *Non-Recursive*: Search is restricted to the Public DNS database only. If the requested record is not found, the query will not be forwarded to the system's DNS server. This mode is useful when you need to limit queries strictly to local resources. • *Forward to System DNS*: The local DNS database is bypassed and all queries are forwarded directly to the system's DNS server. This is beneficial when you need to rely solely on system-level DNS resources for resolving queries. • *Resolver*: As a resolver, the FortiGate can directly interact with root name servers, TLD name servers, and finally authoritative name servers to resolve DNS queries. The FortiGate will iterate through these DNS servers to get the final IP address for the FQDN, as opposed to forwarding the request to external resolvers in forwarder mode for example. Clients can then use the FortiGate as their DNS server to perform DNS resolution. See FortiGate as a recursive DNS resolver on page 372 for more information.
DNS Filter	Apply a DNS filter profile to DNS server. This option is not available when *Mode* is *Non-Recursive*. See Applying DNS filter to FortiGate DNS server on page 2411 for more information.
DNS over HTTPS	Enable DNS over HTTPS (DoH). DoH is a method of performing DNS resolution over a secure HTTPS connection. See DNS over TLS and HTTPS on page 392 for more information
DNS over HTTP3	Enable DNS over HTTP3 (DoH3). DoH3 is a method of performing DNS resolution over an HTTP3 connection. See DNS over QUIC and DNS over HTTP3 for transparent and local-in DNS modes on page 2416 for more information
DNS over QUIC	Enable DNS over QUIC (DoQ). DoQ is a method of performing DNS resolution over a QUICK UDP Internet Connection (QUIC) connection. See DNS over QUIC and DNS over HTTP3 for transparent and local-in DNS modes on page 2416 for more information

> In 8.0.0 and later, the TLS certificate used by FortiGate for encrypted DNS services is taken from the `ssl-cert` setting in the `config system dns-server` command. If no certificate is configured in `system dns-server` then the FortiGate automatically falls back to the `ssl-cert` setting in the `config web-proxy global` command.

c. Click *OK*.

3. Build the DNS database:

a. In the *DNS Database* table, click *Create New*.

b. Configure the following:

Type	Select the zone type: • *Primary*: The primary DNS zone, to manage entries directly. • *Secondary*: The secondary DNS zone, to import entries from other DNS zones. The purpose of a secondary DNS zone is to provide redundancy and load balancing. If the primary DNS server fails, the secondary DNS server can continue to resolve queries for the domain.
View	Select the zone view: • *Shadow*: This type of DNS zone is designed for both internal and external clients, allowing them to resolve DNS queries with the recursive DNS server on FortiGate. It creates a shadow of your public DNS records within your private network. • *Public*: This type of DNS zone is intended to serve external clients only, allowing them to resolve DNS queries with the non-recursive DNS server on FortiGate. It contains records that map the domain names of your publicly accessible services to their respective IP addresses. These records are propagated across the internet, allowing anyone in the world to find and connect to your services. • *Proxy*: This special type of shadow DNS zone is specifically designed for explicit proxy. It allows the explicit proxy to perform DNS lookups using a local database, providing faster and more efficient resolution of domain names. Internal users can experience improved performance and reduced latency when accessing websites and online services through the explicit proxy.
DNS Zone	The name of the DNS zone.
Domain Name	The domain name.
Hostname of Primary DNS	The domain name of the default DNS server for this zone. This option is only available when *Type* is *Primary*.
IP of Primary	The IP address of the primary DNS server. This option is only available when *Type* is *Secondary*.

Contact Email Address	The email address of the administrator for this zone. You can specify only the username, such as admin, or the full email address, such as admin@test.com. When using only a username, the domain of the email is the zone. This option is only available when *Type* is *Primary*.
TTL	The default time-to-live value for the entries of this DNS zone. This option is only available when *Type* is *Primary*.
Authoritative	Enabling *Authoritative* makes this server is the primary and sole source of information for this specific DNS zone. It prevents the FortiGate from seeking DNS records further upstream. Enabling authoritative is not recommended.
DNS Forwarder	A DNS forwarder routes DNS queries to specific servers based on the domain name. Specify one or two DNS zone forwarder IP addresses. Use the CLI to add more than two DNS forwarder addresses.

c. Add DNS entries:

i. In the *DNS Entries* table, click *Create New*.

ii. Configure the following:

Type	The resource record type. The availability of the subsequent settings vary depending on the selected type. • *Address (A)*: This is the host type. It maps a hostname to an IPv4 address in the DNS system, allowing a browser or other client to access a server using its domain name. • *Name Server (NS)*: This is the name server type. It indicates which DNS server is authoritative for that domain • *Canonical Name (CNAME)*: This is the canonical name type. It's used to alias one name to another. • *Mail Exchange (MX)*: This is the mail exchange type. It routes email to a specified mail server based on the information in the record. • *IPv6 Address (AAAA)*: This is the IPv6 host type. Similar to the *A* record, but it maps a hostname to an IPv6 address. • *IPv4 Pointer (PTR)*: This is the pointer type for IPv4. It provides a mapping of the IP address to a hostname, essentially the reverse of what an *A* record does. • *IPv6 Pointer (PTR)*: This is the pointer type for IPv6. It functions similarly to the IPv4 PTR record, but for IPv6 addresses.
TTL	The time-to-live for this entry.

iii. Click *OK*.

d. Click *OK*.

To configure the FortiGate as a DNS server in the CLI:

1. Configure DNS servers:

```
config system dns-server
    edit <name>
        set dnsfilter-profile {string}
        set doh {enable | disable}
        set doh3 {enable | disable}
        set doq {enable | disable}
        set mode {recursive | non-recursive | forward-only}
        set ssl-cert <certificate_name>
    next
end
```

   See config system dns-server in the CLI reference for a comprehensive list of commands.

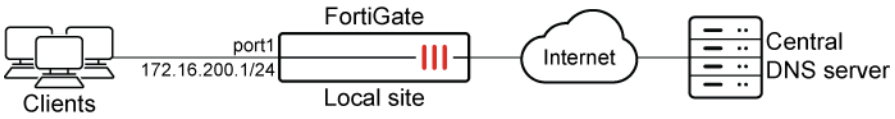
2. Configure DNS database:

```
config system dns-database
    edit <name>
        set authoritative {enable | disable}
        set contact {string}
        set domain {string}
        set forwarder {user}
        set primary-name {string}
        set ttl {integer}
        set type {primary | secondary}
        set view {shadow | public | shadow-ztna | proxy}
        config dns-entry
            edit <id>
                set status {enable | disable}
                set type {A | NS | CNAME | MX | AAAA | PTR | PTR_V6}
                set ttl {integer}
                set ip {ipv4-address-any}
                set ipv6 {ipv6-address}
                set hostname {string}
                set canonical-name {string}
            next
        end
    next
end
```

   See config system dns-database in the CLI reference for a comprehensive list of commands.

Basic DNS server configuration example

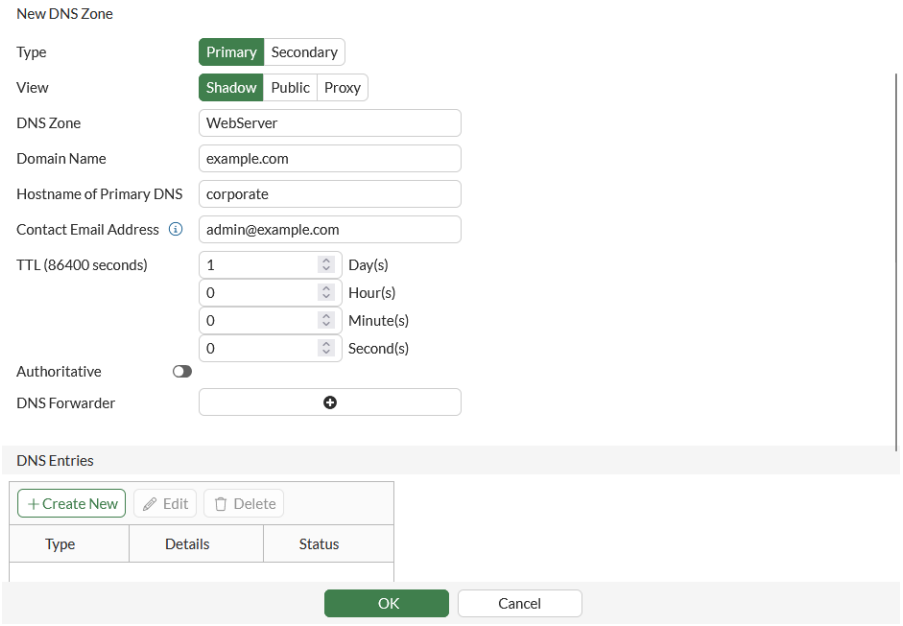
This section describes how to create a non-authoritative primary DNS server. The interface mode is recursive so that, if the request cannot be fulfilled, the external DNS servers will be queried.



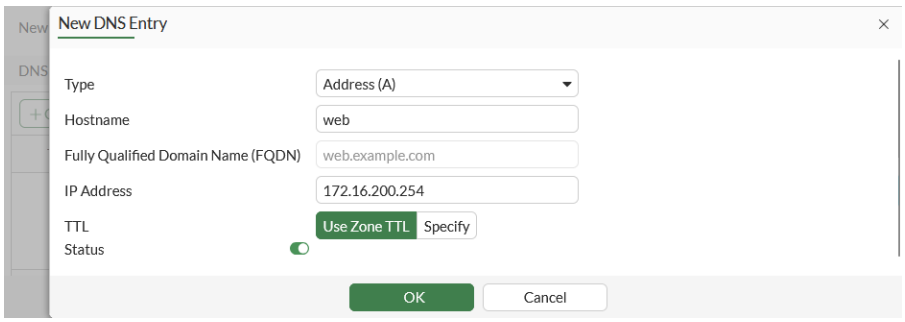
In this example, the Local site is configured as a non-authoritative primary DNS server.

To configure FortiGate as a primary DNS server in the GUI:

1. Go to *Network* > *DNS Servers*.
2. In the *DNS Database* table, click *Create New*.
3. Set *Type* to *Primary*.
4. Set *View* to *Shadow*.
   The *View* setting controls the accessibility of the DNS server. If you select *Public*, external users can access or use the DNS server. If you select *Shadow*, only internal users can use it.
5. Enter a *DNS Zone*, for example, *WebServer*.
6. Enter the *Domain Name* of the zone, for example, *example.com*.
7. Enter the *Hostname* of the DNS server, for example, *corporate*.
8. Enter the *Contact Email Address* for the administrator, for example, *admin@example.com*.
9. Disable *Authoritative*.

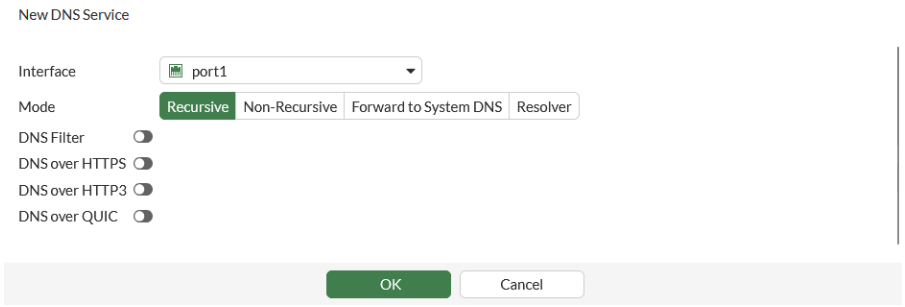


10. Add DNS entries:
    a. In the *DNS Entries* table, click *Create New*.
    b. Select a *Type*, for example *Address (A)*.
    c. Set the *Hostname*, for example *web*.



d. Configure the remaining settings as needed. The options might vary depending on the selected *Type*.
e. Click *OK*.

11. Add more DNS entries as needed.
12. Click *OK*.
13. Enable DNS services on an interface:
    a. Go to *Network* > *DNS Servers*.
    b. In the *DNS Service on Interface* table, click *Create New*.
    c. Select the *Interface* for the DNS server, such as *port1*.
    d. Set the *Mode* to *Recursive*.



    e. Click *OK*.

To configure FortiGate as a primary DNS server in the CLI:

```
config system dns-database
    edit WebServer
        set domain example.com
        set type primary
        set view shadow
        set ttl 86400
        set primary-name corporate
        set contact admin@example.com
        set authoritative disable
        config dns-entry
            edit 1
                set status enable
                set hostname web
                set type A
                set ip 172.16.200.254
            next
        end
    next
end
```

```
config system dns-server
    edit port1
        set mode recursive
    next
end
```

To verify the configuration:

1. Send a DNS query for a DNS entry configured locally on the Local site FortiGate:

```
C:\Users\demo>nslookup office.microsoft.com
Server:  Unknown
Address:  172.16.200.1
Non-authoritative answer:
Name:    web.example.com
Address:  172.16.200.254
```

   The query is resolved to the IP address configured in the shadow DNS database on the Local site FortiGate.
2. Send a DNS query for a domain that is not configured on the Local site FortiGate:

```
C:\Users\demo>nslookup facebook.com
Server:  Unknown
Address:  172.16.200.1
Non-authoritative answer:
Name:    facebook.com
Addresses:  157.240.22.35
```

   The query is resolved by the central DNS server.

FortiGate as a recursive DNS resolver

FortiOS supports being configured as a recursive DNS resolver. As a resolver, the FortiGate can directly interact with root name servers, Top-Level Domain (TLD) name servers, and finally authoritative name servers to resolve DNS queries. The FortiGate will iterate through these DNS servers to get the final IP address for the FQDN, as opposed to forwarding the request to external resolvers in forwarder mode for example. This can avoid hitting limitations from external resolvers which may limit the number of queries per second. Finally, clients can then use the FortiGate as their DNS server to perform DNS resolution.

To configure the recursive resolver mode in the CLI:

```
config system dns-server
    edit <DNS server name>
        set mode resolver
    next
end
```

To configure recursive resolver mode in the GUI:

1. Go to *Network* > *DNS Servers*.
2. Under *DNS Service on Interface*, click *Create New*.
3. Select the *Interface* to listen to DNS queries.
4. Set the *Mode* to *Resolver*.
5. Click *OK*.

Furthermore, FortiOS also adds support for prioritizing root name servers. By default, a list of 13 public root name servers are known to the FortiGate. These DNS servers can be viewed with `diagnose test application dnsproxy 19`.

Prioritized root name servers will be highlighted among the total list of root servers. Prioritized (highlighted) root servers will be queried in round robin fashion. Default (non-highlighted) root servers will only be queried if there are no prioritized root servers.

You may configure root servers from the list of 13 default servers, or you can configure your own custom root name server. Any custom root name servers you configure will be defined with an auto-generated name.

The root name servers can be prioritized and configured with the following:

```
config system dns
    set primary <class IP address>
    set root-servers <DNS root name server IP address>
end
```

At most, two root-servers can be configured.

Example 1

In the following example, we will configure the FortiGate's WAN interface (port3) in resolver mode and configure 1 DNS entry to return results for override.fortinet.com in the fortinet.com domain.

To configure the FortiGate as DNS resolver in the GUI:

1. Go to *Network* > *DNS Servers*.
2. Under *DNS Service on Interface*, click *Create New*.
3. For *Interface*, select *WAN (port3)*.
4. For *Mode*, select *Resolver*.
5. Click *OK*.
6. Under *DNS Database*, click *Create New* and configure the following:

Type	Primary
View	Shadow
DNS Zone	fortinet
Domain Name	fortinet.com
Authoritative	Disable

   a. Under *DNS Entries*, click *Create New* and configure the following:

Type	Address (A)
Hostname	override
IP Address	10.88.0.30

   b. Click *OK*.
7. Click *OK*.

To configure the FortiGate as DNS resolver in the CLI:

```
config system dns-server
    edit "port3"
        set mode resolver
    next
end
config system dns-database
    edit "fortinet"
        set domain "fortinet.com"
        set authoritative disable
        config dns-entry
            edit 1
                set hostname "override"
                set ip 10.88.0.30
            next
        end
    next
end
```

To verify:

1. On the FortiGate, enable dnsproxy debugs:

```
# diagnose debug application dnsproxy -1
# diagnose debug enable
```

2. From a client, use dig to issue a lookup for override.fortinet.com.

```
>dig @10.0.3.254 override.fortinet.com

; <<>> DiG 9.16.23 <<>> @10.0.3.254 override.fortinet.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 53910
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;override.fortinet.com.         IN      A

;; ANSWER SECTION:
override.fortinet.com.  86400   IN      A       10.88.0.30

;; Query time: 73 msec
;; SERVER: 10.0.3.254#53(10.0.3.254)
;; WHEN: Tue Jul 23 17:19:47 Pacific Daylight Time 2024
;; MSG SIZE  rcvd: 55
```

3. From the FortiGate debugs, observe the results are found locally.

```
FortiGate-VM64-KVM # [worker 0] batch_on_read()-3563
[worker 0] udp_receive_request()-3219: vfid=0, vrf=0, intf=5, len=62, alen=16,
10.0.3.2:51485=>10.0.3.254:53
[worker 0] handle_dns_request()-2497: vfid=0 real_vfid=0 id=0xd296 req_type=3
name=override.fortinet.com qtype=1
[worker 0] dns_nat64_ptr_lookup()-272
[worker 0] dns_nat64_update_request()-305
[worker 0] dns_local_lookup_common()-2578: vfid=0, real_vfid=0, view=2,
qname=override.fortinet.com, qtype=1, qclass=1, offset=39, map#=3 max_sz=512
[worker 0] dns_lookup_aa_zone()-627: vfid=0, fqdn=override.fortinet.com
[worker 0] dns_local_lookup_common()-2630: found zone=fortinet domain=fortinet.com
[worker 0] dnsentry_search()-507: domain=fortinet.com, name=override.fortinet.com, type=1
[worker 0] dnsentry_lookup()-431: domain=fortinet.com, name=override.fortinet.com, type=1
[worker 0] dnsentry_lookup()-441: found entry=override.fortinet.com
…
[worker 0] dns_send_response()-1626: domain=override.fortinet.com reslen=55
```

4. From the client, use dig to issue a lookup for www.fortinet.com.

```
C:\Users\tsmith.FORTIAD>dig @10.0.3.254 www.fortinet.com

; <<>> DiG 9.16.23 <<>> @10.0.3.254 www.fortinet.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 24497
;; flags: qr rd ra ad; QUERY: 1, ANSWER: 3, AUTHORITY: 0, ADDITIONAL: 0

;; QUESTION SECTION:
;www.fortinet.com.              IN      A

;; ANSWER SECTION:
www.fortinet.com.       60      IN      CNAME   fortinet.96983.fortiwebcloud.net.
fortinet.96983.fortiwebcloud.net. 60 IN CNAME   ipv6.lb-2.us-west-1.aws.waas-online.net.
ipv6.lb-2.us-west-1.aws.waas-online.net. 60 IN A 54.177.212.176

;; Query time: 226 msec
;; SERVER: 10.0.3.254#53(10.0.3.254)
;; WHEN: Tue Jul 23 17:11:04 Pacific Daylight Time 2024
;; MSG SIZE  rcvd: 146
```

5. From the FortiGate debugs, observe the FortiGate makes the resolver query to each DNS name server directly to resolve the address.

```
FortiGate-VM64-KVM # [worker 0] dns_query_check_timeout()-623: jiffies=60824828
[worker 0] batch_on_read()-3563
[worker 0] udp_receive_request()-3219: vfid=0, vrf=0, intf=5, len=57, alen=16,
10.0.3.2:52979=>10.0.3.254:53
[worker 0] handle_dns_request()-2497: vfid=0 real_vfid=0 id=0x5fb1 req_type=3
name=www.fortinet.com qtype=1
[worker 0] dns_nat64_ptr_lookup()-272
[worker 0] dns_nat64_update_request()-305
```

```
[worker 0] dns_local_lookup_common()-2578: vfid=0, real_vfid=0, view=2,
qname=www.fortinet.com, qtype=1, qclass=1, offset=34, map#=3
 max_sz=512
[worker 0] dns_lookup_aa_zone()-627: vfid=0, fqdn=www.fortinet.com
[worker 0] dns_local_lookup_common()-2630: found zone=fortinet domain=fortinet.com
[worker 0] dnsentry_search()-507: domain=fortinet.com, name=www.fortinet.com, type=1
[worker 0] dnsentry_lookup()-431: domain=fortinet.com, name=www.fortinet.com, type=1
[worker 0] dnsentry_lookup()-431: domain=fortinet.com, name=www.fortinet.com, type=5
[worker 0] dns_send_resol_request()-1322: orig id: 0x5fb1 local id: 0x5fb1
domain=www.fortinet.com
[worker 0] resolver_check_slist()-392: id=0x5fb1 domain=www.fortinet.com zone=fortinet.com
ns=ns3.fortinet.com:208.91.113.63
…
[worker 0] dns_send_resol_request()-1322: orig id: 0xa2d9 local id: 0xa2d9
domain=fortinet.96983.fortiwebcloud.net
[worker 0] resolver_check_slist()-392: id=0xa2d9 domain=fortinet.96983.fortiwebcloud.net
zone=fortiwebcloud.net ns=ns-111.awsdns-13.
com:205.251.192.111
…
[worker 0] dns_send_resol_request()-1322: orig id: 0x8850 local id: 0x8850 domain=ipv6.lb-
2.us-west-1.aws.waas-online.net
[worker 0] resolver_check_slist()-392: id=0x8850 domain=ipv6.lb-2.us-west-1.aws.waas-
online.net zone=waas-online.net ns=ns-131.awsdn
s-16.com:205.251.192.131
…
[worker 0] __udp_receive_response()-3419: vd-0: len=210, addr=205.251.192.131:53, rating=0
[worker 0] dns_query_handle_response()-2762: vfid=0 real_vfid=0 vrf=0 id=0x8850
domain=ipv6.lb-2.us-west-1.aws.waas-online.net pktle
n=210
…
[worker 0] dns_send_response()-1626: domain=www.fortinet.com reslen=146
```

6. From the FortiGate, change the DNS server setting for WAN (port3) to recursive mode.
7. Restart the dnsproxy:

```
# diagnose test application dnsproxy 99
```

8. From the client, use dig to issue a lookup for www.fortinet.com again
9. From the FortiGate debugs, observe the FortiGate now forwards the query to a resolver instead of resolving the query itself.

```
FortiGate-VM64-KVM # [worker 0] batch_on_read()-3563
[worker 0] udp_receive_request()-3219: vfid=0, vrf=0, intf=5, len=57, alen=16,
10.0.3.2:64510=>10.0.3.254:53
[worker 0] handle_dns_request()-2497: vfid=0 real_vfid=0 id=0x1acc req_type=3
name=www.fortinet.com qtype=1
[worker 0] dns_nat64_ptr_lookup()-272
[worker 0] dns_nat64_update_request()-305
[worker 0] dns_local_lookup_common()-2578: vfid=0, real_vfid=0, view=2,
qname=www.fortinet.com, qtype=1, qclass=1, offset=34, map#=3 max_sz=512
[worker 0] dns_lookup_aa_zone()-627: vfid=0, fqdn=www.fortinet.com
[worker 0] dns_local_lookup_common()-2630: found zone=fortinet domain=fortinet.com
```

```
[worker 0] dnsentry_search()-507: domain=fortinet.com, name=www.fortinet.com, type=1
[worker 0] dnsentry_lookup()-431: domain=fortinet.com, name=www.fortinet.com, type=1
[worker 0] dnsentry_lookup()-431: domain=fortinet.com, name=www.fortinet.com, type=5
[worker 0] dns_send_resol_request()-1322: orig id: 0x1acc local id: 0x1acc
domain=www.fortinet.com
[worker 0] dns_find_best_server()-653: found server: 208.91.112.52 (vfid=0 vrf=0)
[worker 0] dns_udp_forward_request()-1064: vdom=root req_type=3 domain=www.fortinet.com oif=0
[worker 0] dns_udp_forward_request()-1185: Send 57B to [208.91.112.52]:53 via fd=18 request:1
[worker 0] batch_on_read()-3563
[worker 0] __udp_receive_response()-3419: vd-0: len=157, addr=208.91.112.52:53, rating=0
[worker 0] dns_query_handle_response()-2762: vfid=0 real_vfid=0 vrf=0 id=0x1acc
domain=www.fortinet.com pktlen=157
[worker 0] dns_query_save_response()-2734: domain=www.fortinet.com pktlen=157
…
[worker 0] dns_send_response()-1626: domain=www.fortinet.com reslen=157
```

Example 2

In the following example, the root name servers that should be prioritized will be defined, and the DNS server will be highlighted from the list root name servers.

To define the root name servers:

1. Define the root name servers to prioritize:

```
config system dns
    set primary 8.8.8.8
    set root-servers "199.7.83.42" "192.33.4.12"
end
```

2. Display the DNS server for the root name servers:

```
# diagnose test application dnsproxy 19
worker idx: 0
name=. label_count=0 ns_count=13
    ns=a.root-servers.net A=198.41.0.4 use=1
    ns=b.root-servers.net A=199.9.14.201 use=1
    *ns=c.root-servers.net A=192.33.4.12 use=1
    ns=d.root-servers.net A=199.7.91.13 use=1
    ns=e.root-servers.net A=192.203.230.10 use=1
    ns=f.root-servers.net A=192.5.5.241 use=1
    ns=g.root-servers.net A=192.112.36.4 use=1
    ns=h.root-servers.net A=198.97.190.53 use=1
    ns=i.root-servers.net A=192.36.148.17 use=1
    ns=j.root-servers.net A=192.58.128.30 use=1
    ns=k.root-servers.net A=193.0.14.129 use=1
    *ns=l.root-servers.net A=199.7.83.42 use=1
    ns=m.root-servers.net A=202.12.27.33 use=1
```

Example 3

In the following example, the user configures a custom root server which will be added to the root zone cache. The name of the new user-defined root server will be auto-generated.

To configure a specific root name server:

1. Configure the user-defined root name server:

```
config system dns
    set primary 8.8.8.8
    set root-servers "172.16.200.55"
end
```

2. Display the DNS server for the user-defined root name server:

```
# diagnose test application dnsproxy 19
worker idx: 0
name=. label_count=0 ns_count=14
    ns=a.root-servers.net A=198.41.0.4 use=1
    ns=b.root-servers.net A=199.9.14.201 use=1
    ns=c.root-servers.net A=192.33.4.12 use=1
    ns=d.root-servers.net A=199.7.91.13 use=1
    ns=e.root-servers.net A=192.203.230.10 use=1
    ns=f.root-servers.net A=192.5.5.241 use=1
    ns=g.root-servers.net A=192.112.36.4 use=1
    ns=h.root-servers.net A=198.97.190.53 use=1
    ns=i.root-servers.net A=192.36.148.17 use=1
    ns=j.root-servers.net A=192.58.128.30 use=1
    ns=k.root-servers.net A=193.0.14.129 use=1
    ns=l.root-servers.net A=199.7.83.42 use=1
    ns=m.root-servers.net A=202.12.27.33 use=1
    *ns=a.user-root-servers.fgt A=172.16.200.55 use=1
```

Implement the interface name as the source IP address in RADIUS, LDAP, and DNS configurations

To account for dynamic IP address changes, such as those governed by SD-WAN rules, interface names can be used to define the source IP addresses in RADIUS, LDAP, and DNS configurations using the `source-ip-interface` command. The interface's current IP address will be used as the source IP address in the configuration; enhancing network flexibility and resolving potential connectivity issues. See RADIUS servers on page 3527and LDAP servers on page 3509 for more information.

The following examples demonstrate configuring the interface name as the source IP address in RADIUS and LDAP servers, and local DNS databases, respectively. The server configuration on the FortiGate will need to have a source IP address included. This source IP address can be any interface, including the IP address of a loopback interface.

Example 1: RADIUS server

In this example, the loopback interface is used as the source IP address and the interface method is set to `specify`.

To configure the interface name as the source IP address in a RADIUS server:

1. Configure the loopback interface:

```
config system interface
    edit "loop1"
        set vdom "vdom1"
        set ip 10.1.10.9 255.255.255.0
        set allowaccess ping
        set type loopback
    next
end
```

2. Configure the RADIUS user object:

```
config user radius
    edit "radius-142"
        set server "10.1.100.142"
        set secret XXXXXX
        set source-ip-interface "loop1"
        set interface-select-method specify
        set interface "testvlink1"
    next
end
```

3. Test the basic communication:
   a. Perform a local credential check with a known user and password:

```
# diagnose test authserver radius radius-142 pap test1 test1
authenticate 'test1' against 'pap' succeeded, server=primary assigned_rad_session_
id=105510201667592 session_timeout=3600 secs idle_timeout=300 secs!
Group membership(s) - group1
```

   b. Perform a sniffer check in a separate SSH session to verify that the source IP address contains the expected IP address of the loop interface:

```
# diagnose sniffer packet any 'host 10.1.100.142 and port 1812' 4
interfaces=[any]
filters=[host 10.1.100.142 and port 1812]
5.144791 testvlink1 out 10.1.10.9.17437 -> 10.1.100.142.1812: udp 110
5.144794 testvlink0 in 10.1.10.9.17437 -> 10.1.100.142.1812: udp 110
5.144812 port2 out 10.1.10.9.17437 -> 10.1.100.142.1812: udp 110
5.149570 port2 in 10.1.100.142.1812 -> 10.1.10.9.17437: udp 169
5.149581 testvlink0 out 10.1.100.142.1812 -> 10.1.10.9.17437: udp 169
5.149583 testvlink1 in 10.1.100.142.1812 -> 10.1.10.9.17437: udp 169
```

Example 2: LDAP server

In this example, a VDOM link is used as the source IP address and the interface method is set to `sdwan`.

To configure the interface name as the source IP address in an LDAP server:

1. Configure the VDOM link:

```
config system interface
    edit "testvlink1"
        set vdom "vdom1"
        set ip 10.12.1.10 255.255.255.0
        set allowaccess ping
        set type vdom-link
    next
end
```

2. Configure the LDAP user object:

```
config user ldap
    edit "ldap-2"
        set server "172.18.60.214"
        set source-ip-interface "testvlink1"
        set cnid "cn"
        set dn "dc=qafsso,dc=com"
        set type regular
        set username "cn=Manager,dc=qafsso,dc=com"
        set password ENC XXXXXXXXXXXXXXXXXX
        set interface-select-method sdwan
    next
end
```

3. Confirm in a packet capture that the correct IP address is used in the outgoing and incoming packets:

```
# diagnose sniffer packet any  'port 389' 4
interfaces=[any]
filters=[port 389]
11.356977 testvlink1 out 10.12.1.10.11742 -> 172.18.60.214.389: syn 1099805903
11.356979 testvlink0 in 10.12.1.10.11742 -> 172.18.60.214.389: syn 1099805903
11.357001 port1 out 172.16.200.9.11742 -> 172.18.60.214.389: syn 1099805903
11.357548 port1 in 172.18.60.214.389 -> 172.16.200.9.11742: syn 2083328609 ack 1099805904
11.357556 testvlink0 out 172.18.60.214.389 -> 10.12.1.10.11742: syn 2083328609 ack 1099805904
11.357558 testvlink1 in 172.18.60.214.389 -> 10.12.1.10.11742: syn 2083328609 ack 1099805904
11.357566 testvlink1 out 10.12.1.10.11742 -> 172.18.60.214.389: ack 2083328610
11.357564 testvlink0 in 10.12.1.10.11742 -> 172.18.60.214.389: ack 2083328610
11.357571 port1 out 172.16.200.9.11742 -> 172.18.60.214.389: ack 2083328610
```

Example 3: DNS database

In this example, the system DNS database uses a customized DNS server and a loopback interface as the source IP address.

To configure the interface name as the source IP address in a DNS database:

1. Configure the loopback interface:

```
config system interface
    edit "loop"
        set vdom "root"
        set ip 10.3.10.9 255.255.255.0
        set allowaccess ping
        set type loopback
        set role lan
        set snmp-index 28
    next
end
```

2. Configure the DNS database:

```
config system dns-database
    edit "1"
        set domain "fortinet-fsso.com"
        set authoritative disable
        set forwarder "10.1.100.150"
        set source-ip-interface "loop"
    next
end
```

3. Clear the DNS host cache and ping any FQDN in the DNS domain:

```
# execute ping login.fortinet-fsso.com
PING login.fortinet-fsso.com (10.1.100.5): 56 data bytes
64 bytes from 10.1.100.5: icmp_seq=0 ttl=255 time=0.1 ms
64 bytes from 10.1.100.5: icmp_seq=1 ttl=255 time=0.0 ms
64 bytes from 10.1.100.5: icmp_seq=2 ttl=255 time=0.0 ms
64 bytes from 10.1.100.5: icmp_seq=3 ttl=255 time=0.0 ms
64 bytes from 10.1.100.5: icmp_seq=4 ttl=255 time=0.0 ms

--- login.fortinet-fsso.com ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.0/0.0/0.1 ms
```

4. Perform a sniffer check on the FortiGate to confirm that the loopback interface was used as the source IP address in a DNS query:

```
# diagnose sniffer packet any 'host 10.1.100.150 and port 53' 4
interfaces=[any]
filters=[host 10.1.100.150 and port 53]
91.180362 port2 out 10.3.10.9.1328 -> 10.1.100.150.53: udp 41
91.180733 port2 in 10.1.100.150.53 -> 10.3.10.9.1328: udp 57
468.753163 port2 out 10.3.10.9.3990 -> 10.1.100.150.53: udp 41
468.753533 port2 in 10.1.100.150.53 -> 10.3.10.9.3990: udp 57
523.470007 port2 out 10.3.10.9.3990 -> 10.1.100.150.53: udp 44
523.470017 port2 out 10.3.10.9.3990 -> 10.1.100.150.53: udp 45
523.470025 port2 out 10.3.10.9.3990 -> 10.1.100.150.53: udp 47
```

```
523.470350 port2 in 10.1.100.150.53 -> 10.3.10.9.3990: udp 60
523.470380 port2 in 10.1.100.150.53 -> 10.3.10.9.3990: udp 85
523.470396 port2 in 10.1.100.150.53 -> 10.3.10.9.3990: udp 95
^C
10 packets received by filter
0 packets dropped by kernel
```

Optimizing hostname resolution in non-AD environments

In a non-Active Directory (AD) environment utilizing FortiGate as both the DNS and DHCP server, several key configurations are required for ensuring effective hostname resolution:

1. PTR records:

   These records are essential for reverse hostname resolution and enable IP addresses to be mapped back to their corresponding hostnames. For instance, if a device has the IP address 10.10.10.13 and the hostname pc1, a PTR record ensures that querying this IP returns pc1.pochiya.net. Without PTR records, reverse lookups would fail, potentially disrupting network services or applications that rely on this functionality.
2. Search domain configuration:

   In an AD environment, domain-joined devices automatically recognize the domain context, allowing hostname resolution without the need for the full FQDN. However, in a non-AD setup with FortiGate, explicit configuration of the search domain is necessary. This configuration informs client devices about the domain suffix to append when resolving hostnames, facilitating access using just the hostname.
3. DHCP integration:

   FortiGate can automate the distribution of DNS settings through DHCP options, reducing manual configuration. Devices receive the required configurations automatically when joining the network, streamlining administration and ensuring consistency across devices.
4. Manual configuration for static IPs:

   For devices with static IP assignments not managed by DHCP, manually adding the search domain to each device's DNS settings is required for hostname resolution.
5. Monitoring and maintenance:

   Regularly updating PTR records and verifying search domain configurations are critical to prevent DNS resolution issues.

Configuring PTR records and setting up proper search domains in a non-AD environment with FortiGate is crucial for reliable network services and seamless device communication. These configurations make sure that hostname resolution works efficiently, enhancing user convenience and overall network reliability.

Example

In this example, a small corporate network is connected to a FortiGate firewall that also serves as the DNS server for the internal domain pochiya.net and the DHCP server. To ensure accurate hostname resolution, the FortiGate is configured with reverse (PTR) records and proper search domain settings.

To configuring DNS server and DNS entries for a PTR and address record in the GUI:

1. Go to *Network* > *DNS Servers* and in the *DNS Database table*, click *Create New*.
2. Configure the following:

Field	Value
Type	Primary
View	Setting
DNS Zone	internal
Domain Name	pochiya.net
Hostname of Primary DNS	corporate
Contact Email Address	admin@pochiya.net
Authoritative	Disable

3. In the *DNS Entries* table, click *Create New* and configure the following to add a PTR record:

Field	Value
Type	IPv4 Pointer (PTR)
Hostname	pc1
IP Address	10.10.10.13
Status	Enable

4. Click *OK*.
5. In the *DNS Entries* table, click *Create New* again and configure the following to add an address record:

Field	Value
Type	Address (A)
Hostname	pc1
IP Address	10.10.10.13
Status	Enable

6. Click *OK*.
7. Click *OK*.
8. In the *DNS Service on Interface* table, click *Create New*.
9. Set *Interface* to *port2* and *Mode* to *Recursive* to enable DNS services on that interface.
10. Click *OK*.
11. In the CLI, configure a domain to push the DNS suffix automatically.

To configuring DNS server and DNS entries for a PTR and address record in the CLI:

```
config system dns-database
    edit "internal"
        set domain "pochiya.net"
        set authoritative disable
        config dns-entry
            edit 1
                set type PTR
                set hostname "pc1"
                set ip 10.10.10.13
            next
            edit 2
                set hostname "pc1"
                set ip 10.10.10.13
            next
        end
        set primary-name "corporate"
        set contact "admin@pochiya.net"
    next
end
config system dns-server
    edit "port2"
        set mode recursive
    next
end
```

To configure a domain to push the DNS suffix automatically:

Modify the existing DHCP server configuration that is responsible for distributing IP addresses to the corporate network:

```
config system dhcp server
    edit 2
        set domain "pochiya.net"
    next
end
```

The full DHCP configuration should look like:

```
config system dhcp server
    edit 2
        set dns-service default
        set domain "pochiya.net"
        set default-gateway 10.10.10.1
        set netmask 255.255.255.0
        set interface "port2"
        config ip-range
            edit 1
                set start-ip 10.10.10.10
                set end-ip 10.10.10.200
            next
```

```
        end
    next
end
```

To verify the configuration:

1. Check the PTR record function using nslookup on a client device:

```
#nslookup 10.10.10.13
Server:  UnKnown
Address:  10.10.10.1

Name:    pc1.pochiya.net
Address:  10.10.10.13
```

   The response should indicate that the IP address 10.10.10.13 is mapped to pc1.pochiya.net.

2. Test hostname resolution without specifying the FQDN using nslookup on a client device:

```
#nslookup pc1
Server:  UnKnown
Address:  10.10.10.1

Name:    pc1.pochiya.net
Address:  10.10.10.13
```

   The response should indicate that hostname resolution works without requiring users to enter the full FQDN. Client machines will automatically append the search domain (pochiya.net) when resolving hostnames.

DDNS

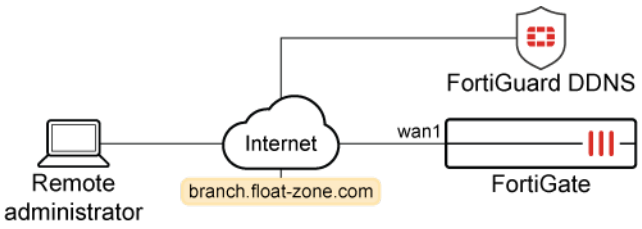
If your external IP address changes regularly and you want a static domain name, you can configure the external interface to use a dynamic DNS (DDNS) service. This ensures that external users and customers can always connect to your company firewall. You can configure FortiGuard as the DDNS server using the GUI or CLI.

Multiple DDNS interfaces can be configured in the GUI. The number of DDNS entries that can be configured is restricted by table size, with limits of 16, 32, and 64 entries for entry-level, mid-range, and high-end FortiGates respectively.

A license or subscription is not required to use the DDNS service.

DDNS is not supported in transparent mode.

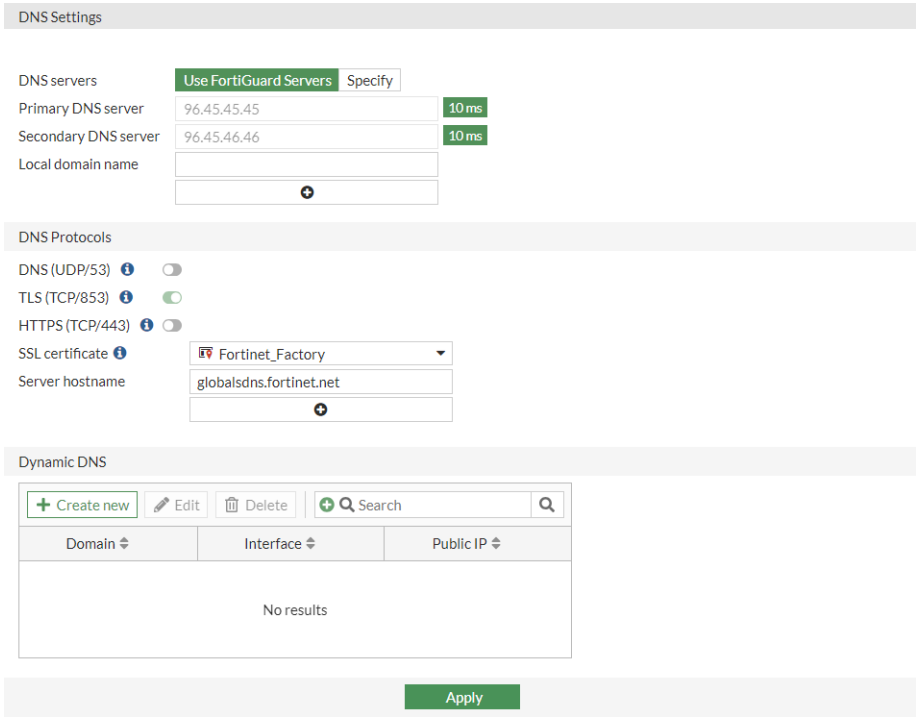
Sample topology



In this example, FortiGuard DDNS is enabled and the DDNS server is set to *float-zone.com*. Other DDNS server options include *fortiddns.com* and *fortidyndns.com*.

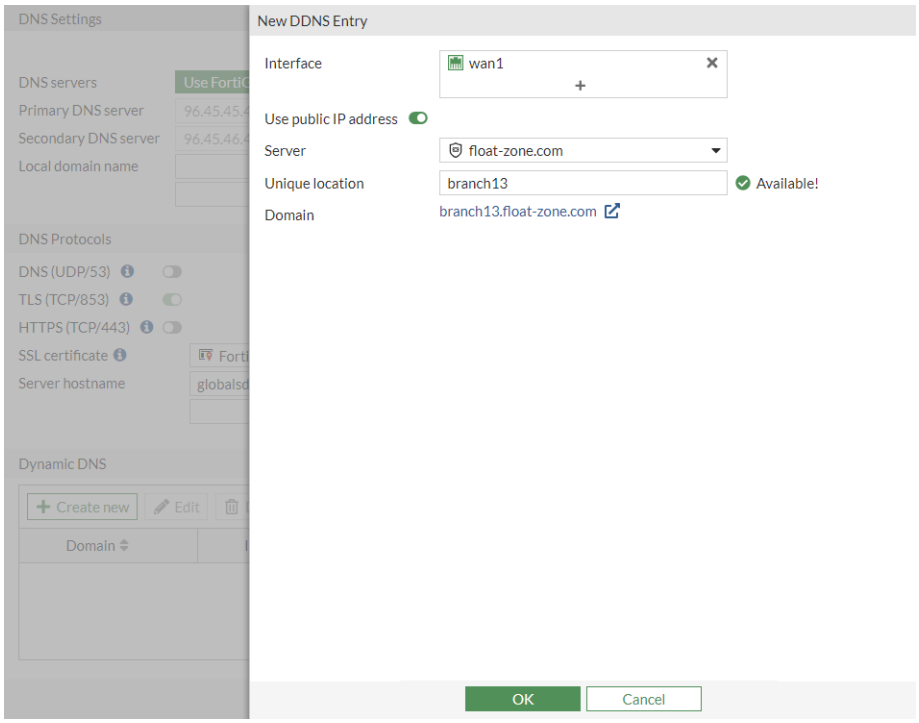
To configure multiple DDNS entries in the GUI:

1. Go to *Network* > *DNS*.
2. In the *Dynamic DNS* table, click *Create new*.



The *New DDNS Entry* pane opens.

3. Configure the DDNS entry settings:
   a. Select the *Interface* with the dynamic connection.
   b. Select the *Server* that you have an account with.
   c. Enter the *Unique Location*.



d. Click *OK*.

4. Click *Create new* and repeat step 3 to add more entries.
5. Click *Apply*.

To configure the FortiGuard DDNS service as an IPv4 DDNS server in the CLI:

```
config system ddns
    edit 1
        set ddns-server FortiGuardDDNS
        set server-type ipv4
        set ddns-domain "branch.float-zone.com"
        set addr-type ipv4
        set use-public-ip enable
        set monitor-interface "wan1"
    next
end
```

To configure the FortiGuard DDNS service as an IPv6 DDNS server in the CLI:

```
config system ddns
    edit 1
        set ddns-server FortiGuardDDNS
        set server-type ipv6
        set ddns-domain "fgtatest001.float-zone.com"
        set addr-type ipv6
        set monitor-interface "wan1"
    next
end
```

DDNS servers other than FortiGuard

If you do not have a FortiGuard subscription, or want to use a different DDNS server, you can configure a DDNS server for each interface. Only the first configure port appears in the GUI.

The available commands vary depending on the selected DDNS server.

To configure DDNS servers other than FortiGuard in the CLI:

```
config system ddns
    edit <DDNS_ID>
        set monitor-interface <external_interface>
        set ddns-server <ddns_server_selection>
        set server-type {ipv4 | ipv6}
        set ddns-server-addr <address>
        set addr-type ipv6 {ipv4 | ipv6}
    next
end
```

To configure an IPv6 DDNS client with generic DDNS on port 3 in the CLI:

```
config system ddns
    edit 1
        set ddns-server genericDDNS
        set server-type ipv6
        set ddns-server-addr "2004:16:16:16::2" "16.16.16.2" "ddns.genericddns.com"
        set ddns-domain "test.com"
        set addr-type ipv6
        set monitor-interface "port3"
    next
end
```

Refresh DDNS IP addresses

When using a public IP that is not assigned to the FortiGate, the FortiGate cannot trigger an update when the IP address changes. The FortiGate can be configured to refresh DDNS IP addresses by periodically checking the DDNS server at an update interval.

To configure FortiGate to refresh DDNS IP addresses in the CLI:

```
config system ddns
    edit 1
        set use-public-ip enable
        set update-interval <seconds>
    next
end
```

When `update-interval` is set to `0`:

- For FortiGuard DDNS, the interval is 300 seconds.
- For third part DDNS servers, the interval is assigned by the DDNS server.

Disable cleartext

When `clear-text` is disabled, FortiGate uses the SSL connection to send and receive DDNS updates.

To disable cleartext and set the SSL certificate in the CLI:

```
config system ddns
    edit 2
        set clear-text disable
        set ssl-certificate <cert_name>
    next
end
```

DDNS update override

A DHCP server has an override command option that allows DHCP server communications to go through DDNS to perform updates for the DHCP client. This enforces a DDNS update of the A field every time even if the DHCP client does not request it. This allows support for the `allow`, `ignore`, and `deny client-updates` options.

To enable DDNS update override in the CLI:

```
config system dhcp server
    edit 1
        set ddns-update enable
        set ddns-update-override enable
        set ddns-server-ip <ddns_server_ip>
        set ddns-zone <ddns_zone>
    next
end
```

Troubleshooting

To debug DDNS:

```
# diagnose debug application ddnscd -1
# diagnose debug enable
```

To check if a DDNS server is available:

```
# diagnose test application ddnscd 3
```

Not available:

```
FortiDDNS status:
ddns_ip=0.0.0.0, ddns_ip6=::, ddns_port=443 svr_num=0 domain_num=0
```

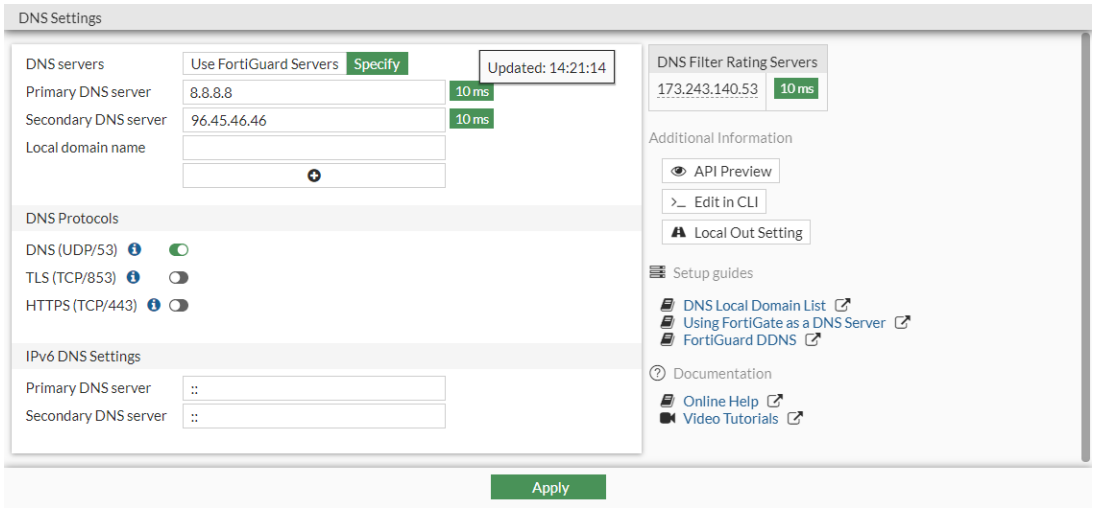
Available:

```
FortiDDNS status:
ddns_ip=208.91.113.230, ddns_ip6=::, ddns_port=443 svr_num=1 domain_num=3
svr[0]= 208.91.113.230
domain[0]= fortiddns.com
domain[1]= fortidyndns.com
domain[2]= float-zone.com
```

DNS latency information

High latency in DNS traffic can result in an overall sluggish experience for end-users. In the *DNS Settings* pane, you can quickly identify DNS latency issues in your configuration.

Go to *Network* > *DNS* to view DNS latency information in the right side bar. If you use FortiGuard DNS, latency information for DNS, DNS filter, web filter, and outbreak prevention servers is also visible. Hover your pointer over a latency value to see when it was last updated.



To view DNS latency information using the CLI:

```
# diagnose test application dnsproxy 2
worker idx: 0
worker: count=1 idx=0
retry_interval=500 query_timeout=1495
DNS latency info:
vfid=0 server=2001::1 latency=1494 updated=73311
vfid=0 server=96.45.46.46 latency=1405 updated=2547
vfid=0 server=8.8.8.8 latency=19 updated=91
SDNS latency info:
vfid=0 server=173.243.140.53 latency=1 updated=707681
```

```
DNS_CACHE: alloc=35, hit=26
RATING_CACHE: alloc=1, hit=49
DNS UDP: req=66769 res=63438 fwd=83526 alloc=0 cmp=0 retrans=16855 to=3233
          cur=111 switched=8823467 num_switched=294 v6_cur=80 v6_switched=7689041 num_v6_
switched=6
         ftg_res=8 ftg_fwd=8 ftg_retrans=0
DNS TCP: req=0, res=0, fwd=0, retrans=0 alloc=0, to=0
FQDN: alloc=45 nl_write_cnt=9498 nl_send_cnt=21606 nl_cur_cnt=0
Botnet: searched=57 hit=0 filtered=57 false_positive=0
```

To view the latency from web filter and outbreak protection servers using the CLI:

```
# diagnose debug rating
Locale   : english

Service  : Web-filter
Status   : Enable
License  : Contract

Service  : Antispam
Status   : Disable

Service  : Virus Outbreak Prevention
Status   : Disable

-=- Server List (Tue Jan 22 08:03:14 2019) -=-

IP               Weight RTT Flags  TZ  Packets  Curr Lost Total Lost  Updated Time
173.243.138.194  10     0 DI       -8  700      0         2           Tue Jan 22 08:02:44 2019
173.243.138.195  10     0          -8  698      0         4           Tue Jan 22 08:02:44 2019
173.243.138.198  10     0          -8  698      0         4           Tue Jan 22 08:02:44 2019
173.243.138.196  10     0          -8  697      0         3           Tue Jan 22 08:02:44 2019
173.243.138.197  10     1          -8  694      0         0           Tue Jan 22 08:02:44 2019
96.45.33.64      10     22 D       -8  701      0         6           Tue Jan 22 08:02:44 2019
64.26.151.36     40     62         -5  704      0         10          Tue Jan 22 08:02:44 2019
64.26.151.35     40     62         -5  703      0         9           Tue Jan 22 08:02:44 2019
209.222.147.43   40     70 D       -5  696      0         1           Tue Jan 22 08:02:44 2019
66.117.56.42     40     70         -5  697      0         3           Tue Jan 22 08:02:44 2019
66.117.56.37     40     71         -5  702      0         9           Tue Jan 22 08:02:44 2019
65.210.95.239    40     74         -5  695      0         1           Tue Jan 22 08:02:44 2019
65.210.95.240    40     74         -5  695      0         1           Tue Jan 22 08:02:44 2019
45.75.200.88     90     142        0   706      0         12          Tue Jan 22 08:02:44 2019
45.75.200.87     90     155        0   714      0         20          Tue Jan 22 08:02:44 2019
45.75.200.85     90     156        0   711      0         17          Tue Jan 22 08:02:44 2019
45.75.200.86     90     159        0   704      0         10          Tue Jan 22 08:02:44 2019
62.209.40.72     100    157        1   701      0         7           Tue Jan 22 08:02:44 2019
62.209.40.74     100    173        1   705      0         11          Tue Jan 22 08:02:44 2019
62.209.40.73     100    173        1   699      0         5           Tue Jan 22 08:02:44 2019
121.111.236.179  180    138        9   706      0         12          Tue Jan 22 08:02:44 2019
121.111.236.180  180    138        9   704      0         10          Tue Jan 22 08:02:44 2019
```

DNS over TLS and HTTPS

DNS over TLS (DoT) is a security protocol for encrypting and encapsulating DNS queries and responses over the TLS protocol. DoT increases user privacy and security by preventing eavesdropping and manipulation of DNS data via man-in-the-middle attacks. Similarly, DNS over HTTPS (DoH) provides a method of performing DNS resolution over a secure HTTPS connection. DoT and DoH are supported in explicit mode where the FortiGate acts as an explicit DNS server that listens for DoT and DoH requests. Local-out DNS traffic over TLS and HTTPS is also supported.

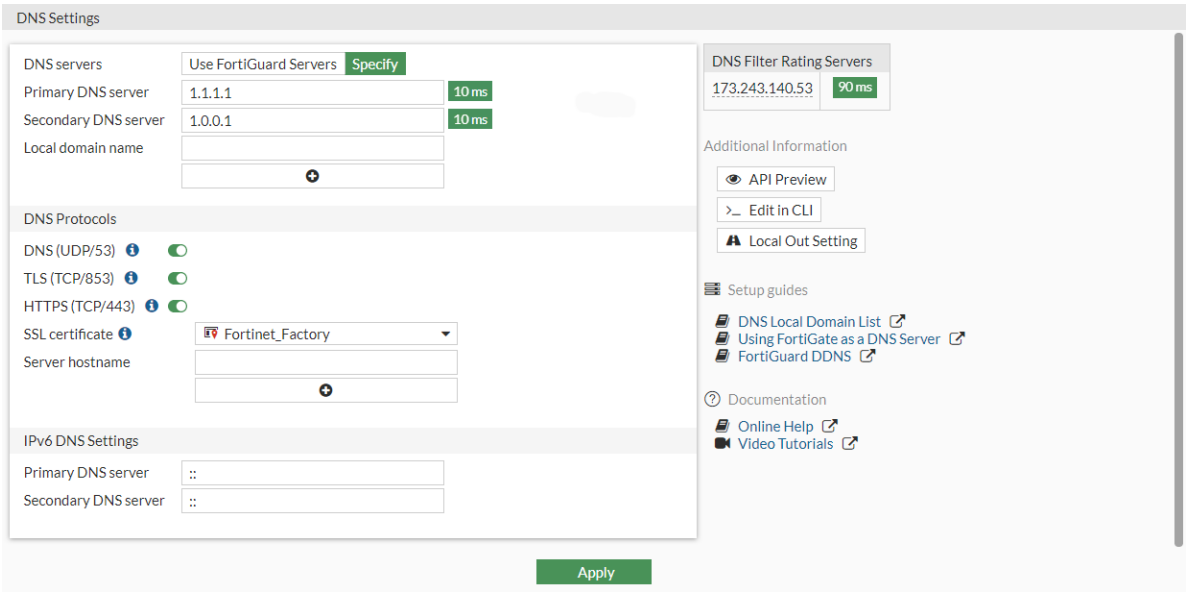
Basic configurations for enabling DoT and DoH for local-out DNS queries

Before enabling DoT or DoH, ensure that they are supported by the DNS servers. The legacy FortiGuard DNS servers (208.91.112.53 and 208.91.112.52) do not support DoT or DoH queries, and will drop these packets. At times, the latency status of the DNS servers might also appear high or unreachable.

Disabling DoT and DoH is recommended when they are not supported by the DNS servers.

To enable DoT and DoH DNS in the GUI:

1. Go to *Network* > *DNS*.
2. Enter the primary and secondary DNS server addresses.
3. In the *DNS Protocols* section, enable *TLS (TCP/853)* and *HTTPS (TCP/443)*.



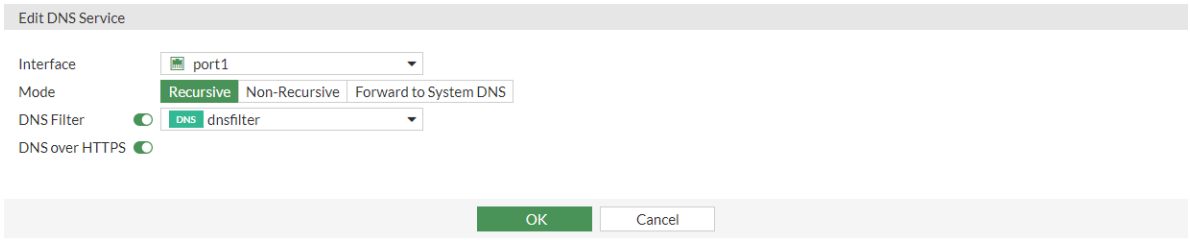
4. Configure the other settings as needed.
5. Click *Apply*.

To enable DoT and DoH DNS in the CLI:

```
config system dns
    set primary 1.1.1.1
    set secondary 1.0.0.1
    set protocol {cleartext dot doh}
end
```

To enable DoH on the DNS server in the GUI:

1. Go to *Network* > *DNS Servers*.
2. In the *DNS Service on Interface* section, edit an existing interface, or create a new one.
3. Select a *Mode*, and *DNS Filter* profile.
4. Enable *DNS over HTTPS*.



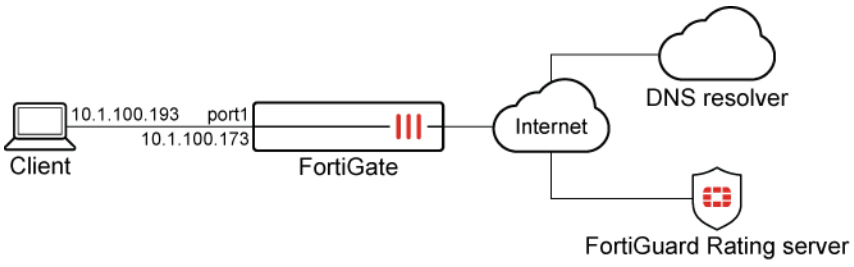
5. Click *OK*.

To enable DoH on the DNS server in the CLI:

```
config system dns-server
    edit "port1"
        set dnsfilter-profile "dnsfilter"
        set doh enable
    next
end
```

Examples

The following examples demonstrate how configure DNS settings to support DoT and DoH queries made to the FortiGate.



DoT

The following example uses a DNS filter profile where the education category is blocked.

To enable scanning DoT traffic in explicit mode with a DNS filter:

1. Configure the DNS settings:

```
config system dns
    set primary 1.1.1.1
    set secondary 1.0.0.1
    set protocol dot
end
```

2. Configure the DNS filter profile:

```
config dnsfilter profile
    edit "dnsfilter"
        config ftgd-dns
            config filters
                edit 1
                    set category 30
                    set action block
                next
            end
        end
    next
end
```

3. Configure the DNS server settings:

```
config system dns-server
    edit "port1"
        set dnsfilter-profile "dnsfilter"
    next
end
```

4. Send a DNS query over TLS (this example uses kdig on an Ubuntu client) using the FortiGate as the DNS server. The www.ubc.ca domain belongs to the education category:

```
root@client:/tmp# kdig -d @10.1.100.173 +tls +header +all www.ubc.ca
;; DEBUG: Querying for owner(www.ubc.ca.), class(1), type(1), server(10.1.100.173), port(853),
protocol(TCP)
;; DEBUG: TLS, received certificate hierarchy:
;; DEBUG:  #1,
C=US,ST=California,L=Sunnyvale,O=Fortinet,OU=FortiGate,CN=FG3H1E5818903681,EMAIL=support@forti
net.com
;; DEBUG:      SHA-256 PIN: Xhkpv9ABEhxDLtWG+lGEndNrBR7B1xjRYlGn2ltlkb8=
;; DEBUG:  #2, C=US,ST=California,L=Sunnyvale,O=Fortinet,OU=Certificate Authority,CN=fortinet-
subca2001,EMAIL=support@fortinet.com
;; DEBUG:      SHA-256 PIN: 3T8EqFBjpRSkxQNPFagjUNeEUghXOEYp904ROlJM8yo=
;; DEBUG:  #3, C=US,ST=California,L=Sunnyvale,O=Fortinet,OU=Certificate Authority,CN=fortinet-
```

```
ca2,EMAIL=support@fortinet.com
;; DEBUG:      SHA-256 PIN: /QfV4N3k5oxQR5RHtW/rbn/HrHgKpMLN0DEaeXY5yPg=
;; DEBUG: TLS, skipping certificate PIN check
;; DEBUG: TLS, skipping certificate verification
;; TLS session (TLS1.2)-(ECDHE-RSA-SECP256R1)-(AES-256-GCM)
;; ->>HEADER<<- opcode: QUERY; status: NOERROR; id: 56719
;; Flags: qr rd; QUERY: 1; ANSWER: 1; AUTHORITY: 0; ADDITIONAL: 0

;; QUESTION SECTION:
;; www.ubc.ca.                  IN      A

;; ANSWER SECTION:
www.ubc.ca.             60      IN      A       208.91.112.55

;; Received 44 B
;; Time 2021-03-12 23:11:27 PST
;; From 10.1.100.173@853(TCP) in 0.2 ms
root@client:/tmp#
```

The IP returned by the FortiGate for ubc.ca belongs to the FortiGuard block page, so the query was blocked successfully.

DoH

The following example uses a DNS filter profile where the education category is blocked.

To configure scanning DoH traffic in explicit mode with a DNS filter:

1. Configure the DNS settings:

```
config system dns
    set primary 1.1.1.1
    set secondary 1.0.0.1
    set protocol doh
end
```

2. Configure the DNS filter profile:

```
config dnsfilter profile
    edit "dnsfilter"
        config ftgd-dns
            config filters
                edit 1
                    set category 30
                    set action block
                next
            end
        end
    next
end
```

3. Configure the DNS server settings:

```
config system dns-server
    edit "port1"
        set dnsfilter-profile "dnsfilter"
        set doh enable
    next
end
```

4. In your browser, enable DNS over HTTPS.
5. On your computer, edit the TCP/IP settings to use the FortiGate interface address as the DNS server.
6. In your browser, go to a website in the education category (www.ubc.ca). The website is redirected to the block page.

Transparent conditional DNS forwarder

The transparent conditional DNS forwarder allows the FortiGate to intercept and reroute DNS queries for specific domains to a specific DNS server. For example, when a client's DNS is located in a distant location, in order to resolve destination addresses (such as SaaS applications) to the closest application server, the FortiGate can intercept and reroute the requests to a local DNS to resolve.

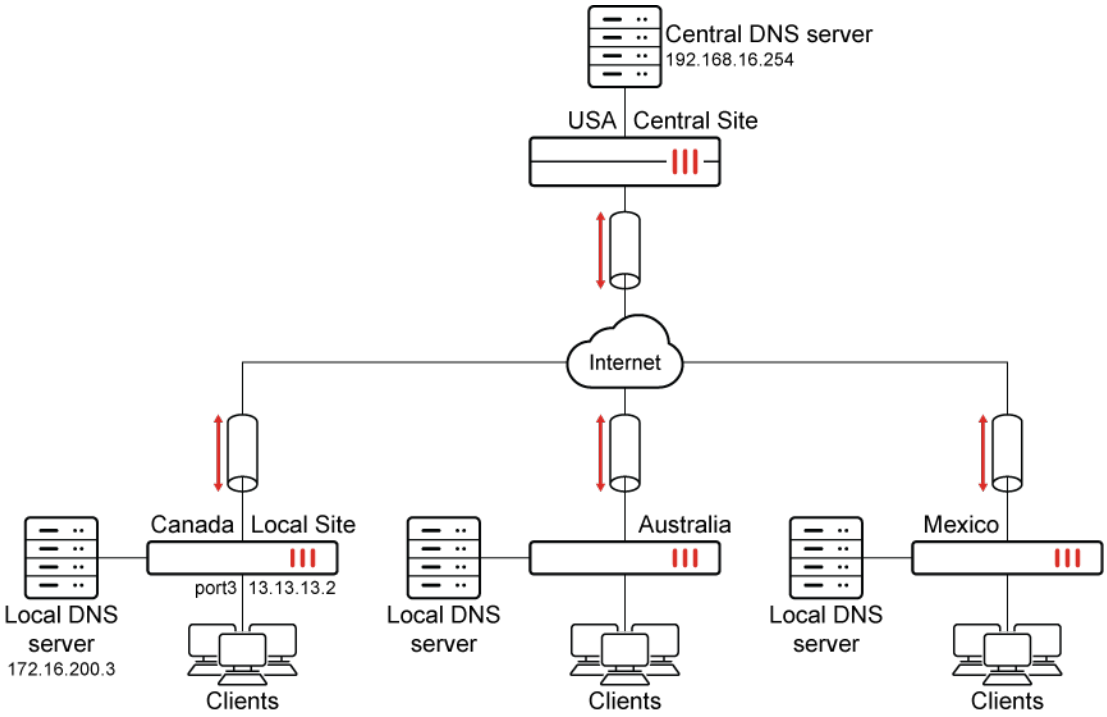
This is done by parsing entries and creating a list of filters based on the domain names of zones. When a DNS request matches one of these filters, the DNS proxy will retrieve the zone's data. The DNS request will then be handled based on the zone's forwarder settings and whether a local answer is available. It may be forwarded to the original destination address, the forwarder address, or not forwarded at all if a local answer is available.

This provides greater control over DNS requests, especially when the administrator is not managing the DNS server configuration of the client devices. This can improve network efficiency and performance by resolving IPs local to the client's PCs rather than IPs local to the central DNS server.

This feature is not supported on FortiGate models with 2 GB RAM or less. See Proxy-related features not supported on FortiGate 2 GB RAM models on page 126 for more information.

Example

In this example, FortiGates at various locations are connected to a central site by VPN tunnels where the corporate DNS server is located. Typically, DNS queries from different sites are sent to the central DNS server and resolved to an IP local to the central site, which might cause latency and performance issues for certain destinations, such as SaaS applications.



The Local Site FortiGate is configured with the Microsoft domain and a local DNS entry. Traffic matching the Microsoft domain is either forwarded to the local DNS server or resolved by the FortiGate, which resolves it to an IP local to the Local Site, thus improving performance.

This example assumes the following have been configured:

- A successfully operational site-to-site VPN between the Local Site and the Central Site FortiGates (see Site-to-site VPN on page 2857 for more information).
- Appropriate routing and network interfaces.
- The client PCs are configured to use the Central DNS Server.

The transparent conditional DNS forwarder feature only works with a proxy-based firewall policy.

By default, DNS server options are not available in the GUI.

To enable DNS server options in the GUI:

1. Go to *System* > *Feature Visibility*.
2. In the *Additional Features* section, enable *DNS Database*.
3. Click *Apply*.

To configure the DNS zone and local DNS entries on the Local Site FortiGate in the GUI:

1. Go to *Network* > *DNS Servers*.
2. In the *DNS Database* table, click *Create New*.
3. Enter a *DNS Zone* name (*SaaS_applications*).

4. Enter a *Domain Name* (*microsoft.com*).
5. Disable the *Authoritative* setting.
6. In the *DNS Forwarder* field, click the *+* and enter the DNS Forwarder address (*172.16.200.3*).
7. Configure the DNS entry:
   a. In the *DNS Entries* table, click *Create New*.
   b. Set the *Type* to *Address (A)*.
   c. Enter a *Hostname* (*office*).
   d. Configure the remaining settings as needed. The options vary depending on the selected *Type*.
   e. Click *OK*.
   f. Optionally, add more DNS entries if needed.
8. In the CLI, configure the source IP:

```
config system dns-database
    edit "SaaS_applications"
        set source-ip 13.13.13.2
    next
end
```

If the DNS server is accessed over a VPN, it may be necessary to specify a source IP for the FortiGate to reach the DNS server. See How to let the FortiGate access internal DNS through site-to-site IPsec VPN for more information.

Site-to-site VPN is not a mandatory requirement for this feature to work and is only applicable to this example.

To configure the DNS zone and local DNS entries on the Local Site FortiGate in the CLI:

```
config system dns-database
    edit "SaaS_applications"
        set domain "microsoft.com"
        set authoritative disable
        set forwarder "172.16.200.3"
        set source-ip 13.13.13.2
        config dns-entry
            edit 1
                set hostname "office"
                set ip 172.16.200.55
            next
        end
    next
end
```

To add the DNS database to a DNS filter profile:

```
config dnsfilter profile
    edit "SaaS"
        set transparent-dns-database "SaaS_applications"
```

```
    next
end
```

Multiple DNS databases can be selected for `transparent-dns-database`.

After selecting a DNS database, users are not permitted to modify the domain name of the zone. Before making any changes to the domain name, remove the reference from the `dnsfilter profile`.

To apply the DNS filter profile in a firewall policy in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy* and edit the outbound policy towards the IPsec VPN tunnel.
2. Set the *Inspection Mode* to *Proxy-based*.
3. In the *Security Profiles* section, enable *DNS filter* and select the profile created in the previous procedure (*SaaS*).
4. In the *Logging Options* section, enable *Log allowed traffic*.
5. Configure the remaining settings as needed.
6. Click *OK*.

To apply the DNS filter profile to the outbound policy towards the IPsec VPN tunnel in the CLI:

```
config firewall policy
    edit 1
        set name "outbound_VPN"
        ...
        set inspection-mode proxy
        set dnsfilter-profile "SaaS"
        set logtraffic enable
        ...
    next
end
```

To verify the configuration:

From one of the Windows client desktops, use the `nslookup` command to send various DNS queries.

1. Send a DNS query for a DNS entry configured locally on the Local Site FortiGate:

   ```
   C:\Users\demo>nslookup office.microsoft.com
   Server:  Unknown
   Address:  192.168.16.254
   Non-authoritative answer:
   Name:    osiprod-wus-pineapple-100.westus.cloudapp.azure.com
   Address:  172.16.200.55
   ```

   The query is resolved to the IP address configured on the Local Site FortiGate.
2. Send a DNS query for the domain configured on the Local Site FortiGate:

   ```
   C:\Users\demo>nslookup teams.microsoft.com
   Server:  Unknown
   ```

```
Address:  192.168.16.254
Non-authoritative answer:
Name:     s-0005.s-msedge.net
Address:  172.16.200.254
```

The query is resolved by the local DNS server.

3. Send a DNS query for a domain that is not configured on the Local Site FortiGate:

```
C:\Users\demo>nslookup facebook.com
Server:  Unknown
Address:  192.168.16.254
Non-authoritative answer:
Name:    facebook.com
Addresses:  157.240.249.35
```

The query is resolved by the central DNS server.

IPv6 support for conditional DNS forwarder

The configuration for IPv6 is similar to an IPv4 conditional DNS forwarder. When configuring the DNS forwarder address, the IPv6 address must be specified.

To configure a DNS forwarder:

```
config system dns-database
    edit <name>
        set source-ip6 <IPv6_address>
        set forwarder6 <IPv6_address>
    next
end
```

If the DNS server is accessed over a VPN, it may be necessary to specify a source IP for the FortiGate to reach the DNS server. See How to let the FortiGate access internal DNS through site-to-site IPsec VPN for more information.

Interfaces in non-management VDOMs as the source IP address of the DNS conditional forwarding server

Interfaces that are in non-management VDOMs can be the source IP address of the DNS conditional forwarding server.

- When `vdom-dns` is enabled in a VDOM, only the IP addresses of interfaces in that VDOM can be configured as the `source-ip`.
- When `vdom-dns` is disabled (default), only the IP address of interfaces in the management VDOM can be configured as the `source-ip`.

For more information on VDOM DNS, see Important DNS CLI commands on page 358.

In this example:

- vdom1 is a non-management VDOM
- port8 is assigned to vdom1 and has IP address 13.13.13.13
- port1 is assigned to the management VDOM (root) and has IP address 172.16.200.1

To configure the interfaces:

```
config global
    config system interface
        edit "port8"
            set vdom "vdom1"
            set ip 13.13.13.13 255.255.255.0
        next
        edit "port1"
            set vdom "root"
            set ip 172.16.200.1 255.255.255.0
        next
    end
end
```

To test configuring a source IP address when vdom-dns is disabled:

```
config vdom
    edit vdom1
        config system vdom-dns
            set vdom-dns disable
        end
    next
end
```

- port8 cannot be used as the source IP address in a DNS database because it is assigned to vdom1, and not to a management VDOM:

```
config vdom
    edit vdom1
        config system dns-database
            edit "1"
                set source-ip 13.13.13.13
13.13.13.13 does not match any interface ip in vdom root.
node_check_object fail! for source-ip 13.13.13.13
```

- port1 can be used as the source IP address in a DNS database because it is assigned to the management VDOM:

```
config vdom
    edit vdom1
        config system dns-database
            edit "1"
                set source-ip 172.16.200.1
            next
        end
```

```
    next
end
```

To test configuring a source IP address when `vdom-dns` is enabled:

```
config vdom
    edit vdom1
        config system vdom-dns
            set vdom-dns enable
        end
    next
end
```

- `port8` can be used as the source IP address in a DNS database because it is assigned to the `vdom1`:

```
config vdom
    edit vdom1
        config system dns-database
            edit "1"
                set source-ip 13.13.13.13
            next
        end
    next
end
```

- `port1` cannot be used as the source IP address in a DNS database because it is assigned to the management VDOM, and not to `vdom1`:

```
config vdom
    edit vdom1
        config system dns-database
            edit "1"
                set source-ip 172.16.200.1
172.16.200.1 does not match any interface ip in vdom vdom1.
node_check_object fail! for source-ip 172.16.200.1
```

DNS session helpers

DNS session helpers work in the background, passively learning hostnames and A/AAAA records from the DNS traffic that is being forwarded through the FortiGate. The learned address records are then stored as a hostname cache and FQDN addresses. These stored address records are subsequently utilized by the DNS proxy. This passive learning approach enables FortiOS to correspond with DNS traffic and acquire address records when possible, eliminating the need for active FQDN queries or reverse DNS lookups. This reduces the necessity to connect to the actual server, effectively diminishing the overall FortiGate traffic to the DNS server.

FortiOS incorporates two types of DNS session helpers: `dns-udp` and `dns-tcp`.

By default, FortiOS enables the `dns-udp` session helper and disables the `dns-tcp` session helper. This default configuration is based on the fact that the majority of DNS traffic occurs over UDP due to its lower overhead and faster response times. However, FortiOS provides the flexibility to enable `dns-tcp` if required.

To enable the DNS session helper from listening on TCP port 53:

```
config system session-helper
    edit 0
        set name dns-tcp
        set port 53
        set protocol 6
    next
end
```

Use the `show system session-helper` command to view the current session helper configuration.

To accept DNS sessions you must add a security policy with service set to *ALL* or to the DNS predefined service (which listens on TCP and UDP ports 53).

Disabling DNS session helper

In certain scenarios, you might consider disabling the DNS session helper. This action essentially removes it from the session-helper list, preventing the session helper from listening on port 53. Once the DNS session helper is disabled, the hostname cache and FQDN addresses will no longer be curated from the DNS traffic. Consequently, the DNS proxy will need to dispatch requests to servers to retrieve the necessary information, which could potentially increase the load on the DNS proxy.

DNS session helper is required for wildcard FQDN addresses, as they are initially empty. The FortiGate analyzes client DNS responses, adding any IP addresses found to the relevant wildcard FQDN object. See Using wildcard FQDN addresses in firewall policies on page 2040.

To disable the DNS session helper from listening on UDP port 53:

1. Enter the following command to find the DNS session helper entry that listens on UDP port 53:

```
#show system session-helper
…
    edit 14
        set name dns-udp
        set protocol 17
        set port 53
    next
…
```

2. Enter the following command to delete DNS session helper:

```
config system session-helper
    delete 14
end
```

Similarly, the DNS session helper can be disabled from listening on TCP port 53.

DNS troubleshooting

The following diagnose command can be used to collect DNS debug information. If you do not specify worker ID, the default worker ID is `0`.

```
# diagnose test application dnsproxy
worker idx: 0
1. Clear DNS cache
2. Show stats
3. Dump DNS setting
4. Reload FQDN
5. Requery FQDN
6. Dump FQDN
7. Dump DNS cache
8. Dump DNS DB
9. Reload DNS DB
10. Dump secure DNS policy/profile
11. Dump Botnet domain
12. Reload Secure DNS setting
13. Show Hostname cache
14. Clear Hostname cache
15. Show SDNS rating cache
16. Clear SDNS rating cache
17. DNS debug bit mask
18. DNS debug obj mem
99. Restart dnsproxy worker
```

To view useful information about the ongoing DNS connection:

```
# diagnose test application dnsproxy 3
worker idx: 0
vdom: root, index=0, is primary, vdom dns is disabled, mip-169.254.0.1 dns_log=1 tls=0 cert=
dns64 is disabled
vdom: vdom1, index=1, is primary, vdom dns is enabled, mip-169.254.0.1 dns_log=1 tls=0 cert=
dns64 is disabled
dns-server:96.45.45.220:45 tz=-480 tls=0 req=0 to=0 res=0 rt=0 rating=1 ready=0 timer=37 probe=9
failure=0 last_failed=0
dns-server:8.8.8.8:53 tz=0 tls=0 req=73 to=0 res=73 rt=5 rating=0 ready=1 timer=0 probe=0
failure=0 last_failed=0
dns-server:65.39.139.63:53 tz=0 tls=0 req=39 to=0 res=39 rt=1 rating=0 ready=1 timer=0 probe=0
failure=0 last_failed=0
dns-server:62.209.40.75:53 tz=60 tls=0 req=0 to=0 res=0 rt=0 rating=1 ready=0 timer=37 probe=9
failure=0 last_failed=0
dns-server:209.222.147.38:53 tz=-300 tls=0 req=0 to=0 res=0 rt=0 rating=1 ready=0 timer=37 probe=9
failure=0 last_failed=0
dns-server:173.243.138.221:53 tz=-480 tls=0 req=0 to=0 res=0 rt=0 rating=1 ready=0 timer=37
probe=9 failure=0 last_failed=0
dns-server:45.75.200.89:53 tz=0 tls=0 req=0 to=0 res=0 rt=0 rating=1 ready=0 timer=37 probe=9
failure=0 last_failed=0
DNS_CACHE: hash-size=2048, ttl=1800, min-ttl=60, max-num=-1
DNS FD: udp_s=12 udp_c=17:18 ha_c=22 unix_s=23, unix_nb_s=24, unix_nc_s=25
```

```
        v6_udp_s=11, v6_udp_c=20:21, snmp=26, redir=13, v6_redir=14
DNS FD: tcp_s=29, tcp_s6=27, redir=31 v6_redir=32
FQDN: hash_size=1024, current_query=1024
DNS_DB: response_buf_sz=131072
LICENSE: expiry=2015-04-08, expired=1, type=2
FDG_SERVER:96.45.45.220:45
FGD_CATEGORY_VERSION:8
SERVER_LDB: gid=eb19, tz=-480, error_allow=0
FGD_REDIR_V4:208.91.112.55 FGD_REDIR_V6:
```

Important fields include:

`tls`	1 if the connection is TLS, 0 if the connection is not TLS.
`rt`	The round trip time of the DNS latency.
`probe`	The number of probes sent.

To dump the second DNS worker's cache:

```
# diagnose test application dnsproxy 7 1
```

To enable debug on the second worker:

```
# diagnose debug application dnsproxy -1 1
```

To enable debug on all workers by specifying -1 as worker ID:

```
# diagnose debug application dnsproxy -1 -1
```

Explicit and transparent proxies

This section contains instructions for configuring explicit and transparent proxies.

- Explicit web proxy on page 406
- Transparent proxy on page 414
- FTP proxy on page 410
- Proxy policy addresses on page 416
- IPv6 proxy address and address group objects New on page 423
- Proxy policy security profiles on page 424
- Explicit proxy authentication on page 429
- Transparent web proxy forwarding on page 435
- Transparent web proxy forwarding over IPv6 on page 439
- Upstream proxy authentication in transparent proxy mode on page 443
- Multiple dynamic header count on page 444

- Restricted SaaS access on page 447
- Explicit proxy and FortiGate Cloud Sandbox on page 456
- Proxy chaining on page 459
- WAN optimization SSL proxy chaining on page 464
- Agentless NTLM authentication for web proxy on page 472
- Multiple LDAP servers in Kerberos keytabs and agentless NTLM domain controllers on page 476
- Learn client IP addresses on page 477
- Explicit proxy authentication over HTTPS on page 478
- mTLS client certificate authentication on page 480
- CORS protocol in explicit web proxy when using session-based, cookie-enabled, and captive portal-enabled SAML authentication on page 486
- Display CORS content in an explicit proxy environment on page 489
- HTTP connection coalescing and concurrent multiplexing for explicit proxy on page 491
- Secure explicit proxy on page 493
- Secure explicit proxy with client certificates on page 496
- Explicit proxy logging on page 498
- Configuring fast fallback for explicit proxy on page 508
- Forward HTTPS requests to a web server without the need for an HTTP CONNECT message on page 512
- Specifying outgoing interface and VRF for a web proxy forward server or isolator server on page 514
- Isolator servers in proxy policies on page 516
- Secure explicit proxy with client certificate blocklist enforcement NEW on page 520
- Session and user limits on page 523

This feature is not supported on FortiGate models with 2 GB RAM or less. See Proxy-related features not supported on FortiGate 2 GB RAM models on page 126 for more information.

Explicit web proxy

Explicit web proxy can be configured on FortiGate for proxying HTTP and HTTPS traffic.

To deploy explicit proxy, individual client browsers can be manually configured to send requests directly to the proxy, or they can be configured to download proxy configuration instructions from a Proxy Auto-Configuration (PAC) file.

When explicit proxy is configured on an interface, the interface IP address can be used by client browsers to forward requests directly to the FortiGate. FortiGate also supports PAC file configuration.

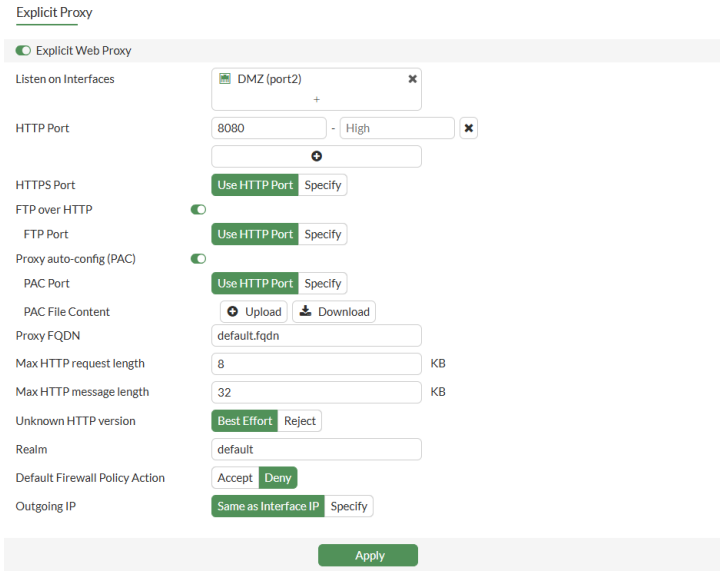
Isolator servers can be applied in explicit proxy policies to distinguish isolated traffic from normal traffic in logs. See Isolator servers in proxy policies on page 516.

To configure explicit web proxy in the GUI:

1. Enable feature visibility in the GUI:
   a. Go to *System* > *Feature Visibility*.
   b. Under *Security Features*, enable *Explicit Proxy*.

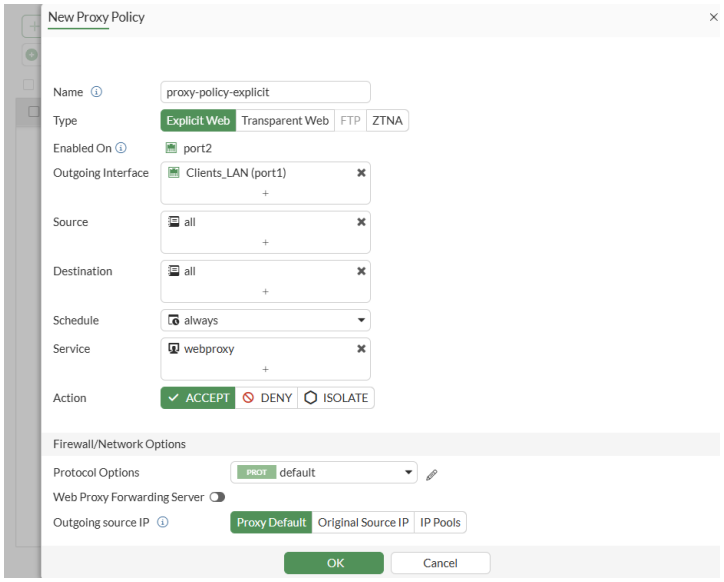
c. Click *Apply*.

2. Enable and configure explicit web proxy:
   a. Go to *Network* > *Explicit Proxy*.
   b. Enable *Explicit Web Proxy*.
   c. Select *port2* as the *Listen on Interfaces* and set the *HTTP Port* to *8080*.
   d. Configure the remaining settings as needed.



   e. Click *Apply*.

3. Create an explicit web proxy policy:
   a. Go to *Policy & Objects* > *Proxy Policy*.
   b. Click *Create New*.
   c. Set *Proxy Type* to *Explicit Web* and *Outgoing Interface* to *port1*.
   d. Also set *Source* and *Destination* to *all*, *Schedule* to *always*, *Service* to *webproxy*, and *Action* to *ACCEPT*.



e. Click *OK* to create the policy.

This example creates a basic policy. If required, security profiles can be enabled, and deep SSL inspection can be selected to inspect HTTPS traffic.

4. Configure a client to use the FortiGate explicit proxy:
   Set the FortiGate IP address as the proxy IP address in the browser, or use an automatic configuration script for the PAC file.

To configure explicit web proxy in the CLI:

1. Enable and configure explicit web proxy:

```
config web-proxy explicit
    set status enable
    set ftp-over-http enable
    set socks enable
    set http-incoming-port 8080
    set ipv6-status enable
    set unknown-http-version best-effort
end
config system interface
    edit "port2"
        set vdom "vdom1"
        set ip 10.1.100.1 255.255.255.0
        set allowaccess ping https ssh snmp http telnet
        set type physical
        set explicit-web-proxy enable
        set snmp-index 12
        end
    next
end
```

2. Create an explicit web proxy policy:

```
config firewall proxy-policy
    edit 1
        set name "proxy-policy-explicit"
        set proxy explicit-web
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set service "webproxy"
        set action accept
        set schedule "always"
        set logtraffic all
    next
end
```

This example creates a basic policy. If required, security profiles can be enabled, and deep SSL inspection can be selected to inspect HTTPS traffic.

3. Configure a client to use the FortiGate explicit web proxy:
   Set the FortiGate IP address as the proxy IP address in the browser, or use an automatic configuration script for the PAC file.

Downloading a PAC file using HTTPS

PAC files can be downloaded for an explicit proxy through the FortiGate's captive portal using HTTPS to ensure a secure download.

In this example, a Windows PC has an HTTPS URL configured in its proxy settings to download a PAC file from a FortiGate by using a download link, https://cp.myqalab.local:7831/proxy.pac, through a captive portal. Once the PAC file is securely downloaded using HTTPS, browsers installed on the PC can use the proxy in the PAC file to visit a website.

The global web proxy settings must be configured to use a customized SSL certificate because the default Fortinet_Factory certificate will not be accepted by Windows due to security restrictions. The customized SSL certificate is used as the HTTPS server's certificate on the FortiGate. All CA certificates in the server certificate must be installed and trusted on the Windows PC.

To download a PAC file using HTTPS:

1. Configure the explicit web proxy to get a PAC file through HTTPS:

```
config web-proxy explicit
    set pac-file-server-status enable
    unset pac-file-server-port
    set pac-file-name "proxy.pac"
    set pac-file-data "function FindProxyForURL(url, host) {
   // testtest
   return \"PROXY 10.1.100.1:8080\";
}
"
    set pac-file-through-https enable
end
```

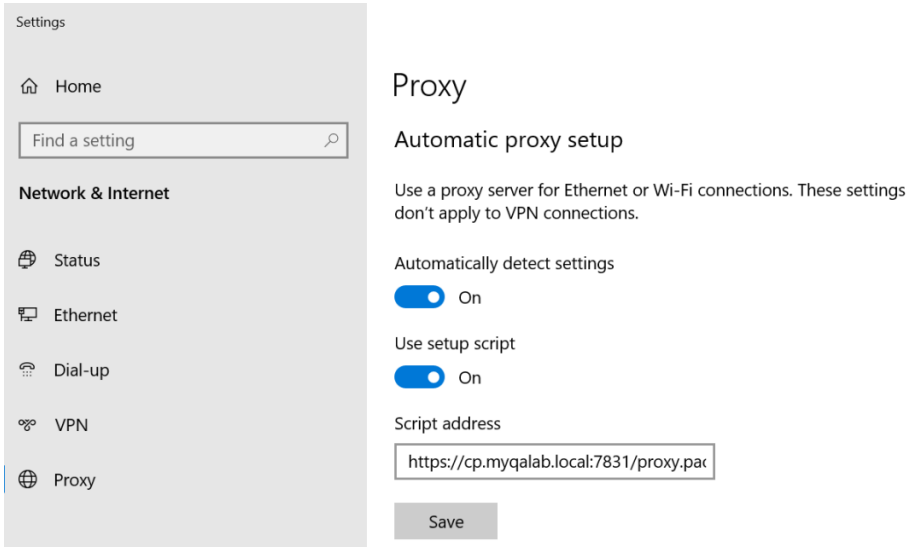
2. Configure the captive portal to be used as an HTTPS server to provide the service to download the PAC file:

```
config authentication setting
    set captive-portal-type ip
    set captive-portal-ip 10.1.100.1
    set captive-portal-ssl-port 7831
end
```

3. Configure the global web proxy settings to use a customized SSL certificate:

```
config web-proxy global
    set ssl-cert "server_cert"
end
```

4. On the Windows PC, go to *Settings* > *Network & Internet* > *Proxy*.



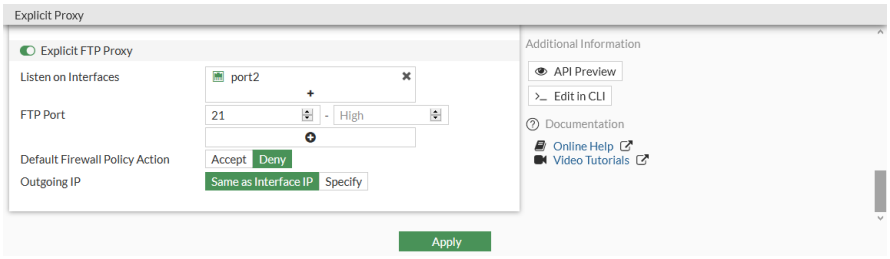
5. In the *Automatic proxy setup* section, click *Save* to trigger the PAC file download from the HTTPS URL.

FTP proxy

FTP proxies can be configured on the FortiGate so that FTP traffic can be proxied. When the FortiGate is configured as an FTP proxy, FTP client applications should be configured to send FTP requests to the FortiGate.

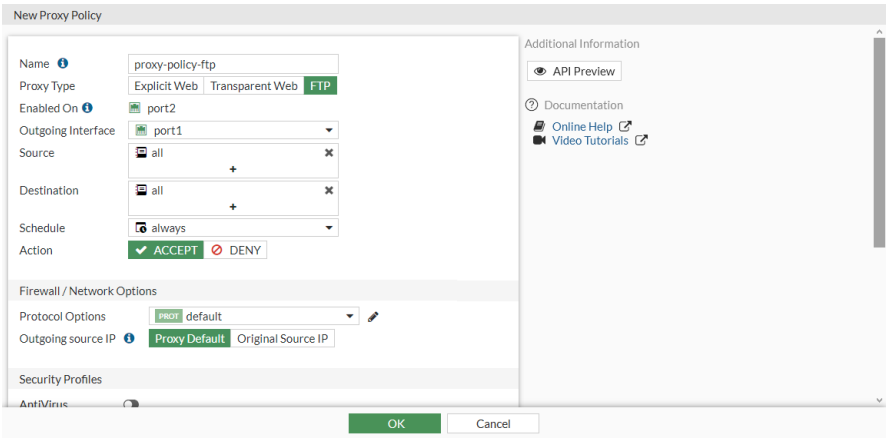
To configure explicit FTP proxy in the GUI:

1. Enable and configure explicit FTP proxy:
   a. Go to *Network* > *Explicit Proxy*.
   b. Enable *Explicit FTP Proxy*.
   c. Select *port2* as the *Listen on Interfaces* and set the *HTTP Port* to *21*.
   d. Configure the *Default Firewall Policy Action* as needed.



   e. Click *Apply*.
2. Create an explicit FTP proxy policy:

a. Go to *Policy & Objects* > *Proxy Policy*.
b. Click *Create New*.
c. Set *Proxy Type* to *FTP* and *Outgoing Interface* to *port1*.
d. Also set *Source* and *Destination* to *all*, *Schedule* to *always*, and *Action* to *ACCEPT*.



e. Click *OK* to create the policy.

This example creates a basic policy. If required, security profiles can be enabled.

3. Configure the FTP client application to use the FortiGate IP address.

To configure explicit FTP proxy in the CLI:

1. Enable and configure explicit FTP proxy:

```
config ftp-proxy explicit
    set status enable
    set incoming-port 21
end
config system interface
    edit "port2"
        set vdom "vdom1"
        set ip 10.1.100.1 255.255.255.0
        set allowaccess ping https ssh snmp http telnet
        set type physical
        set explicit-ftp-proxy enable
        set snmp-index 12
    next
end
```

2. Create an explicit FTP proxy policy:

```
config firewall proxy-policy
    edit 4
        set name "proxy-policy-ftp"
        set proxy ftp
        set dstintf "port1"
        set srcaddr "all"
```

```
        set dstaddr "all"
        set action accept
        set schedule "always"
    next
end
```

This example creates a basic policy. If required, security profiles can be enabled.

3. Configure the FTP client application to use the FortiGate IP address.

Changing the FTP mode from active to passive for explicit proxy

An explicit FTP proxy can convert an active FTP connection initiated by an FTP client to a passive FTP connection between the explicit FTP proxy and FTP server.

```
config ftp-proxy explicit
    set server-data-mode {client | passive}
end
```

`server-data-mode {client | passive}`	Set the data selection mode on the FTP server side: • `client`: use the same transmission mode for client and server data sessions (default). • `passive`: use passive mode for server data sessions.

In this example, a client that only supports active mode FTP connects to a remote FTP server through the explicit FTP proxy to download a text file (test1.txt). The explicit FTP proxy converts the active FTP connection to a passive connection between the explicit FTP proxy and the FTP server.

To configure passive mode for FTP server data sessions:

1. Configure the web proxy:

```
config ftp-proxy explicit
    set status enable
    set incoming-port 21
    set server-data-mode passive
end
```

2. Enable the explicit FTP proxy on port1:

```
config system interface
    edit "port1"
        set ip 10.1.100.2 255.255.255.0
        set explicit-ftp-proxy enable
```

```
    next
end
```

3. Configure the firewall policy:

```
config firewall proxy-policy
    edit 1
        set proxy ftp
        set dstintf "port3"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
    next
end
```

4. Get the client to download the text file from the FTP server (NcFTP is used in this example):

```
ncftpget -E -r 0 -d stdout -u pc4user1@172.16.200.44 -p 123456 10.1.100.2 ./
/home/pc4user1/test1.txt
...
Cmd: PORT 10,1,100,11,151,115
200: PORT command successful. Consider using PASV.
Cmd: RETR /home/pc4user1/test1.txt
```

5. In the FTP server logs, verify that the explicit FTP proxy converted the active FTP connection to a passive connection:

```
...
2023-01-28 01:56:39,909 webproxy_pc04 proftpd[1104] webproxy_pc04 (172.16.200.2
[172.16.200.2]): dispatching PRE_CMD command 'PASV' to mod_exec
2023-01-28 01:56:39,909 webproxy_pc04 proftpd[1104] webproxy_pc04 (172.16.200.2
[172.16.200.2]): dispatching PRE_CMD command 'PASV' to mod_rewrite
2023-01-28 01:56:39,909 webproxy_pc04 proftpd[1104] webproxy_pc04 (172.16.200.2
[172.16.200.2]): dispatching PRE_CMD command 'PASV' to mod_tls
2023-01-28 01:56:39,909 webproxy_pc04 proftpd[1104] webproxy_pc04 (172.16.200.2
[172.16.200.2]): dispatching PRE_CMD command 'PASV' to mod_core
2023-01-28 01:56:39,909 webproxy_pc04 proftpd[1104] webproxy_pc04 (172.16.200.2
[172.16.200.2]): dispatching PRE_CMD command 'PASV' to mod_core
2023-01-28 01:56:39,909 webproxy_pc04 proftpd[1104] webproxy_pc04 (172.16.200.2
[172.16.200.2]): dispatching CMD command 'PASV' to mod_core
2023-01-28 01:56:39,909 webproxy_pc04 proftpd[1104] webproxy_pc04 (172.16.200.2
[172.16.200.2]): in dir_check_full(): path = '/home/pc4user1', fullpath = '/home/pc4user1'
2023-01-28 01:56:39,909 webproxy_pc04 proftpd[1104] webproxy_pc04 (172.16.200.2
[172.16.200.2]): Entering Passive Mode (172,16,200,44,175,61).
2023-01-28 01:56:39,910 webproxy_pc04 proftpd[1104] webproxy_pc04 (172.16.200.2
[172.16.200.2]): dispatching POST_CMD command 'PASV' to mod_exec
2023-01-28 01:56:39,910 webproxy_pc04 proftpd[1104] webproxy_pc04 (172.16.200.2
[172.16.200.2]): dispatching LOG_CMD command 'PASV' to mod_log
2023-01-28 01:56:39,911 webproxy_pc04 proftpd[1104] webproxy_pc04 (172.16.200.2
[172.16.200.2]): dispatching PRE_CMD command 'RETR /home/pc4user1/test1.txt' to mod_exec
```

Transparent proxy

In a transparent proxy deployment, the user's client software, such as a browser, is unaware that it is communicating with a proxy.

Users request internet content as usual, without any special client configuration, and the proxy serves their requests. FortiGate also allows users to configure in transparent proxy mode.

To redirect HTTPS traffic, SSL deep inspection is required.

HTTP and HTTPS traffic that is successfully redirected to a proxy policy is subject to security profiles configured on the proxy policy, not the base firewall policy. Security profiles configured on the base firewall policy still apply to other traffic, such as FTP.

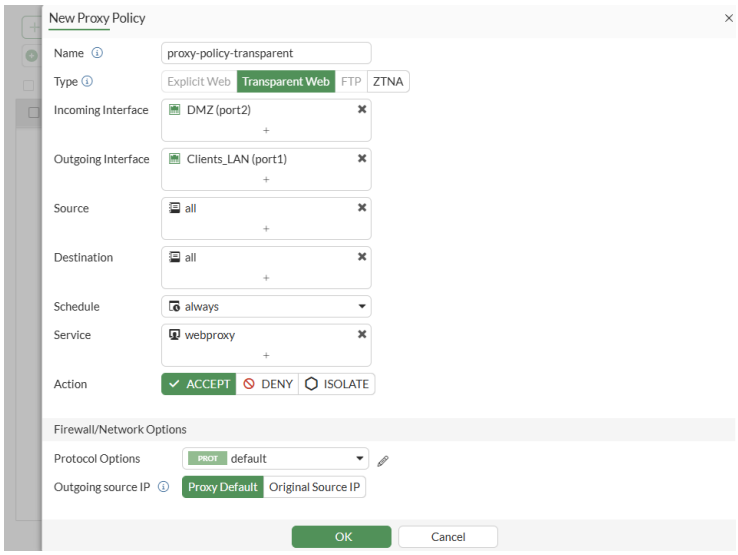
Isolator servers can be applied in transparent proxy policies to distinguish isolated traffic from normal traffic in logs. See Isolator servers in proxy policies on page 516.

To configure transparent proxy in the GUI:

1. Configure a regular firewall policy with HTTP redirect:
   a. Go to *Policy & Objects* > *Firewall Policy*.
   b. Click *Create New*.
   c. Name the policy appropriately, set the *Incoming interface* to *port2*, and set the *Outgoing interface* to *port1*.
   d. Also set *Source* and *Destination* to *all*, *Schedule* to *always*, *Service* to *ALL*, and *Action* to *ACCEPT*.
   e. Set *Inspection Mode* to *Proxy-based* and *SSL Inspection* to *deep-inspection*.
   f. Configure the remaining settings as needed.
   g. Click *OK*.

   By default, HTTP redirect can only be enabled in the CLI. Enable *Policy Advanced Options* in *Feature Visibility* to configure it in the GUI. See Feature visibility on page 3952 on page 1 for more information.

2. Configure a transparent proxy policy:
   a. Go to *Policy & Objects* > *Proxy Policy*.
   b. Click *Create New*.
   c. Set *Proxy Type* to *Transparent Web*, set the *Incoming Interface* to *port2*, and set the *Outgoing Interface* to *port1*.
   d. Also set *Source* and *Destination* to *all*, *Schedule*to *always*, *Service* to *webproxy*, and *Action* to *ACCEPT*.



e. Configure the remaining settings as needed.

f. Click *OK* to create the policy.

This example creates a basic policy. If required, security profiles can be enabled, and deep SSL inspection can be selected to inspect HTTPS traffic.

3. No special configuration is required on the client to use FortiGate transparent proxy. As the client is using the FortiGate as its default gateway, requests will first hit the regular firewall policy, and then be redirected to the transparent proxy policy.

To configure transparent proxy in the CLI:

1. Configure a regular firewall policy with HTTP redirect:

```
config firewall policy
    edit 1
        set name "LAN To WAN"
        set srcintf "port2"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set inspection-mode proxy
        set http-policy-redirect enable
        set fsso disable
        set ssl-ssh-profile "deep-inspection"
        set nat enable
    next
end
```

2. Configure a transparent proxy policy:

```
config firewall proxy-policy
    edit 5
        set name "proxy-policy-transparent"
        set proxy transparent-web
        set srcintf "port2"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set service "webproxy"
        set action accept
        set schedule "always"
    next
end
```

This example creates a basic policy. If required, security profiles can be enabled, and deep SSL inspection can be selected to inspect HTTPS traffic.

3. No special configuration is required on the client to use FortiGate transparent proxy. As the client is using the FortiGate as its default gateway, requests will first hit the regular firewall policy, and then be redirected to the transparent proxy policy.

Proxy policy addresses

Proxy addresses are designed to be used only by proxy policies. The following address types are available:

- Host regex match on page 417
- URL pattern on page 417
- URL category on page 418
- HTTP method on page 419
- HTTP header on page 420
- User agent on page 420
- Advanced (source) on page 421
- Advanced (destination) on page 422

Fast policy match

The fast policy match function improves the performance of IPv4 explicit and transparent web proxies on FortiGate devices.

When enabled, after the proxy policies are configured, the FortiGate builds a fast searching table based on the different proxy policy matching criteria. When fast policy matching is disabled, web proxy traffic is compared to the policies one at a time from the beginning of the policy list.

Fast policy matching is enabled by default, and can be configured with the following CLI command:

```
config web-proxy global
    set fast-policy-match {enable | disable}
end
```

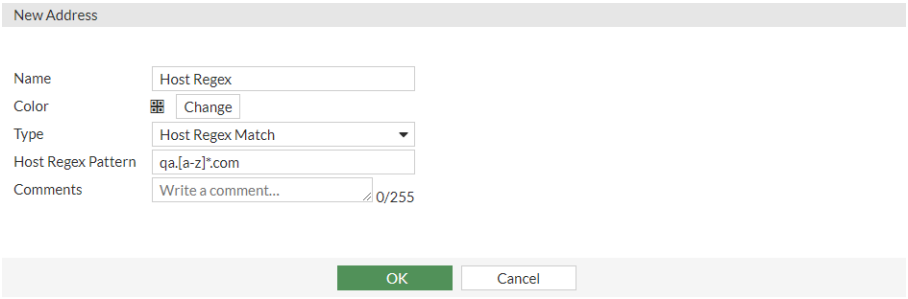
Host regex match

In this address type, a user can create a hostname as a regular expression to match the Host field in the Layer 7 header of a packet. Once created, the hostname address can be selected as a destination of a proxy policy. This means that a policy will only allow or block requests that match the regular expression.

This example creates a host regex match address with the pattern *qa.[a-z]*.com*.

To create a host regex match address in the GUI:

1. Go to *Policy & Objects* > *Addresses* and select *Proxy* > *Address*.
2. Click *Create new*.
3. Set the following:
   - *Name* to *Host Regex*,
   - *Type* to *Host Regex Match*, and
   - *Host Regex Pattern* to *qa.[a-z]*.com*.



4. Click *OK*.

To create a host regex match address in the CLI:

```
config firewall proxy-address
    edit "Host Regex"
        set type host-regex
        set host-regex "qa.[a-z]*.com"
    next
end
```

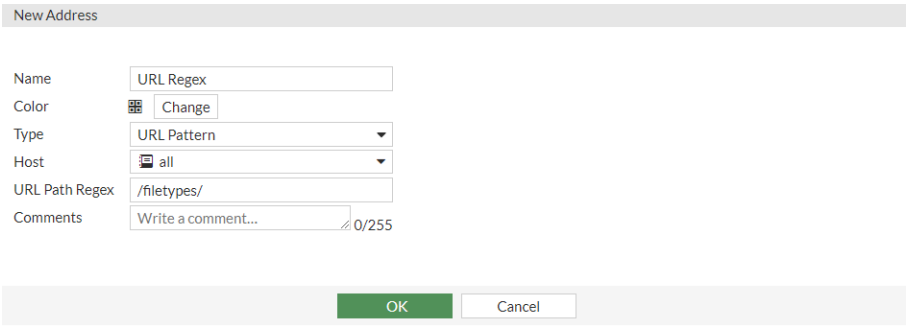
URL pattern

In this address type, a user can create a URL path as a regular expression. Once created, the path address can be selected as a destination of a proxy policy. This means that a policy will only allow or block requests that match the regular expression.

This example creates a URL pattern address with the pattern */filetypes/*.

To create a URL pattern address in the GUI:

1. Go to *Policy & Objects* > *Addresses* and select *Proxy* > *Address*.
2. Click *Create new*.
3. Set the following:
   - *Name* to *URL Regex*,
   - *Type* to *URL Pattern*,
   - *Host* to *all*, and
   - *URL Path Regex* to */filetypes/*.



4. Click *OK*.

To create a URL pattern address in the CLI:

```
config firewall proxy-address
    edit "URL Regex"
        set type url
        set host "all"
        set path "/filetypes/"
    next
end
```

URL category

In this address type, a user can create a URL category based on a FortiGuard URL ID. Once created, the address can be selected as a destination of a proxy policy. This means that a policy will only allow or block requests that match the URL category.

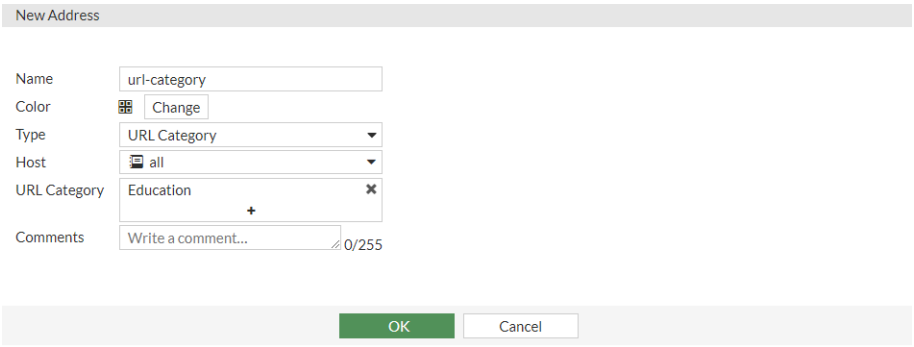
The example creates a URL category address for URLs in the *Education* category. For more information about categories, see https://fortiguard.com/webfilter/categories.

For information about creating and using custom local and remote categories, see Web rating override on page 2736, Using local and remote categories on page 2745, and External feeds on page 4695.

To create a URL category address in the GUI:

1. Go to *Policy & Objects* > *Addresses* and select *Proxy* > *Address*.
2. Click *Create new*.
3. Set the following:

- *Name* to *url-category*,
- *Type* to *URL Category*,
- *Host* to *all*, and
- *URL Category* to *Education*.



4. Click *OK*.

To create a URL category address in the CLI:

```
config firewall proxy-address
    edit "url-category"
        set type category
        set host "all"
        set category 30
    next
end
```

To see a list of all the categories and their numbers, when editing the address, enter `set category ?`.

HTTP method

In this address type, a user can create an address based on the HTTP request methods that are used. Multiple method options are supported, including: *CONNECT*, *DELETE*, *GET*, *HEAD*, *OPTIONS*, *OTHER*, *PATCH*, *POST*, *PUT*, *TRACE*, and *UPDATE*. Once created, the address can be selected as a source of a proxy policy. This means that a policy will only allow or block requests that match the selected HTTP method.

The example creates a HTTP method address that uses the GET method.

To create a HTTP method address in the GUI:

1. Go to *Policy & Objects* > *Addresses* and select *Proxy* > *Address*.
2. Click *Create new*.
3. Set the following:
   - *Name* to *method_get*,
   - *Type* to *HTTP Method*,
   - *Host* to *all*, and
   - *Request Method* to *GET*.
4. Click *OK*.

To create a HTTP method address in the CLI:

```
config firewall proxy-address
    edit "method_get"
        set type method
        set host "all"
        set method get
    next
end
```

HTTP header

In this address type, a user can create a HTTP header as a regular expression. Once created, the header address can be selected as a source of a proxy policy. This means that a policy will only allow or block requests where the HTTP header matches the regular expression.

This example creates a HTTP header address with the pattern *Q[A-B]*.

To create a HTTP header address in the GUI:

1. Go to *Policy & Objects* > *Addresses* and select *Proxy* > *Address*.
2. Click *Create new*.
3. Set the following:
   - *Name* to *HTTP-header*,
   - *Type* to *HTTP Header*,
   - *Host* to *all*,
   - *Header Name* to *Header_Test*, and
   - *Header Regex* to *Q[A-B]*.
4. Click *OK*.

To create a HTTP header address in the CLI:

```
config firewall proxy-address
    edit "method_get"
        set type header
        set host "all"
        set header-name "Header_Test"
        set header "Q[A-B]"
    next
end
```

User agent

In this address type, a user can create an address based on the names of the browsers that are used as user agents. Multiple browsers are supported, such as Chrome, Firefox, Internet Explorer, and others. Once created, the address can be selected as a source of a proxy policy. This means that a policy will only allow or block requests from the specified user agent.

This example creates a user agent address for Google Chrome.

To create a user agent address in the GUI:

1. Go to *Policy & Objects* > *Addresses* and select *Proxy* > *Address*.
2. Click *Create new*.
3. Set the following:
   - *Name* to *UA-Chrome*,
   - *Type* to *User Agent*,
   - *Host* to *all*, and
   - *User Agent* to *Google Chrome*.
4. Click *OK*.

To create a user agent address in the CLI:

```
config firewall proxy-address
    edit "UA-Chrome"
        set type ua
        set host "all"
        set ua chrome
    next
end
```

Browser version control

For security reasons, the user can restrict the browser version by specifying a range of the supported versions which can be set from the CLI using `set ua-min-ver` and `set ua-max-ver`. This option is available when the address *Type* is either *User Agent* or *Advanced (Source)*.

To restrict the browser version:

```
config firewall proxy-address
    edit "ua-ver"
        set type ua
        set ua firefox
        set ua-min-ver "100.0.1"
        set ua-max-ver "160"
    next
end
```

Advanced (source)

In this address type, a user can create an address based on multiple parameters, including HTTP method, User Agent, and HTTP header. Once created, the address can be selected as a source of a proxy policy. This means that a policy will only allow or block requests that match the selected address.

This example creates an address that uses the get method, a user agent for Google Chrome, and an HTTP header with the pattern *Q[A-B]*.

To create an advanced (source) address in the GUI:

1. Go to *Policy & Objects* > *Addresses* and select *Proxy* > *Address*.
2. Click *Create new*.
3. Set the following:
   - *Name* to *advanced_src*,
   - *Type* to *Advanced (Source)*,
   - *Host* to *all*,
   - *Request Method* to *GET*,
   - *User Agent* to *Google Chrome*, and
   - *HTTP header* to *Header_Test* : *Q[A-B]*.
4. Click *OK*.

To create an advanced (source) address in the CLI:

```
config firewall proxy-address
    edit "advance_src"
        set type src-advanced
        set host "all"
        set method get
        set ua chrome
        config header-group
            edit 1
                set header-name "Header_Test"
                set header "Q[A-B]"
            next
        end
    next
end
```

Advanced (destination)

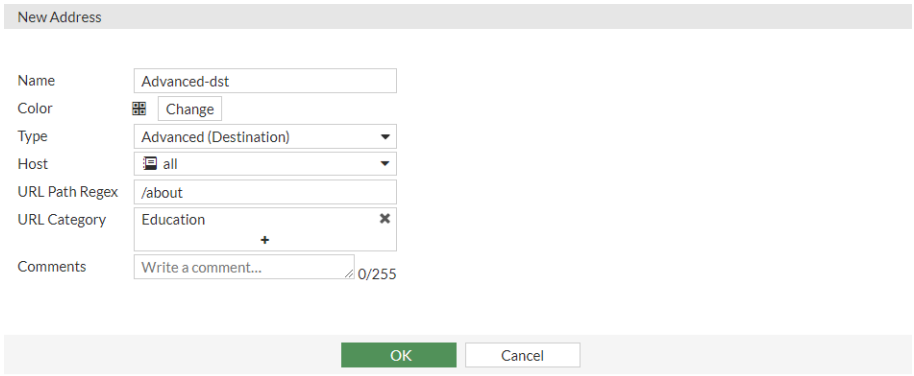
In this address type, a user can create an address based on URL pattern and URL category parameters. Once created, the address can be selected as a destination of a proxy policy. This means that a policy will only allow or block requests that match the selected address.

This example creates an address with the URL pattern */about* that are in the *Education* category. For more information about categories, see https://fortiguard.com/webfilter/categories.

To create an advanced (destination) address in the GUI:

1. Go to *Policy & Objects* > *Addresses* and select *Proxy* > *Address*.
2. Click *Create new*.
3. Set the following:
   - *Name* to *Advanced-dst*,
   - *Type* to *Advanced (Destination)*,
   - *Host* to *all*,

- *URL Path Regex* to */about*, and
- *URL Category* to *Education*.



4. Click *OK*.

To create an advanced (destination) address in the CLI:

```
config firewall proxy-address
    edit "Advanced-dst"
        set type dst-advanced
        set host "ubc"
        set path "/about"
        set category 30
    next
end
```

IPv6 proxy address and address group objects NEW

FortiGate supports IPv6 proxy address and proxy address group objects, enabling proxy-based policy matching for IPv6 traffic.

IPv6 proxy address and address group objects can be configured using the CLI and used as IPv6 address objects in proxy policies and authentication rules. They provide the same matching capabilities as IPv4 proxy address and address group objects, except that the host definitions use IPv6 firewall address objects.

IPv6 proxy address and address group objects can be applied to both the `srcaddr6` and `dstaddr6` options, allowing granular IPv6 traffic matching based on URL, header, method, user agent, category, SaaS application, and other proxy criteria.

IPv6 proxy address and address group objects:

```
config firewall proxy-address6
```

```
config firewall proxy-addrgrp6
```

IPv6 proxy objects in proxy policies:

```
config firewall proxy-policy
    set srcaddr6 {proxy-address6 | proxy-addrgrp6}
    set dstaddr6 {proxy-address6 | proxy-addrgrp6}
end
```

IPv6 proxy objects in authentication rules:

```
config authentication rule
    set srcaddr6 {proxy-address6 | proxy-addrgrp6}
    set dstaddr6 {proxy-address6 | proxy-addrgrp6}
end
```

Example

To configure an IPv6 proxy address object and use it in a proxy policy:

1. Configure an IPv6 proxy address:

```
config firewall proxy-address6
    edit "ipv6-proxy-url"
        set type url
        set path "/example"
    next
end
```

2. Apply the IPv6 proxy address to a proxy policy:

```
config firewall proxy-policy
    edit 1
        set srcaddr6 "ipv6-proxy-url"
        set dstaddr6 "all"
    next
end
```

Proxy policy security profiles

Web proxy policies support most security profile types.

Security profiles must be created before they can be used in a policy, see Security Profiles on page 2212 for information.

Explicit web proxy policy

The security profiles supported by explicit web proxy policies are:

- *AntiVirus*
- *Web Filter*
- *Video Filter*
- *DNS Filter*
- *Application Control*
- *IPS*
- *DLP Profile*
- *ICAP*
- *Web Application Firewall*
- *File Filter*
- *SSL Inspection*

To configure security profiles on an explicit web proxy policy in the GUI:

1. Go to *Policy & Objects* > *Proxy Policy*.
2. Click *Create New*.
3. Set the following:

Proxy Type	Explicit Web
Outgoing Interface	port1
Source	all
Destination	all
Schedule	always
Service	webproxy
Action	ACCEPT

4. In the *Firewall / Network Options* section, set *Protocol Options* to *default*.
5. In the *Security Profiles* section, make the following selections (for this example, these profiles have all already been created):

AntiVirus	av
Web Filter	urlfiler
Application Control	app
IPS	Sensor-1
DLP Profile	dlp
ICAP	default
Web Application Firewall	default

SSL Inspection	deep-inspection

6. Click *OK* to create the policy.

To configure security profiles on an explicit web proxy policy in the CLI:

```
config firewall proxy-policy
    edit 1
        set proxy explicit-web
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set service "web"
        set action accept
        set schedule "always"
        set utm-status enable
        set av-profile "av"
        set webfilter-profile "urlfilter"
        set dlp-profile "dlp"
        set ips-sensor "sensor-1"
        set application-list "app"
        set icap-profile "default"
        set waf-profile "default"
        set ssl-ssh-profile "deep-inspection"
    next
end
```

Transparent proxy

The security profiles supported by transparent proxy policies are:

- *AntiVirus*
- *Web Filter*
- *Video Filter*
- *DNS Filter*
- *Application Control*
- *IPS*
- *DLP Profile*
- *ICAP*
- *Web Application Firewall*
- *File Filter*
- *SSL Inspection*

To configure security profiles on a transparent proxy policy in the GUI:

1. Go to *Policy & Objects* > *Proxy Policy*.
2. Click *Create New*.
3. Set the following:

Proxy Type	Transparent Web
Incoming Interfae	port2
Outgoing Interface	port1
Source	all
Destination	all
Schedule	always
Service	webproxy
Action	ACCEPT

4. In the *Firewall / Network Options* section, set *Protocol Options* to *default*.
5. In the *Security Profiles* section, make the following selections (for this example, these profiles have all already been created):

AntiVirus	av
Web Filter	urlfiler
Application Control	app
IPS	Sensor-1
DLP Profile	dlp
ICAP	default
Web Application Firewall	default
SSL Inspection	deep-inspection

6. Click *OK* to create the policy.

To configure security profiles on a transparent proxy policy in the CLI:

```
config firewall proxy-policy
    edit 2
        set proxy transparent-web
        set srcintf "port2"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set service "webproxy"
        set action accept
        set schedule "always"
        set utm-status enable
        set av-profile "av"
        set webfilter-profile "urlfilter"
        set dlp-profile "dlp"
        set ips-sensor "sensor-1"
        set application-list "app"
        set icap-profile "default"
```

```
        set waf-profile "default"
        set ssl-ssh-profile "certificate-inspection"
    next
end
```

FTP proxy

The security profiles supported by FTP proxy policies are:

- *AntiVirus*
- *Application Control*
- *IPS*
- *File Filter*
- *DLP Profile*

To configure security profiles on an FTP proxy policy in the GUI:

1. Go to *Policy & Objects* > *Proxy Policy*.
2. Click *Create New*.
3. Set the following:

Proxy Type	FTP
Outgoing Interface	port1
Source	all
Destination	all
Schedule	always
Action	ACCEPT

4. In the *Firewall / Network Options* section, set *Protocol Options* to *default*.
5. In the *Security Profiles* section, make the following selections (for this example, these profiles have all already been created):

AntiVirus	av
Application Control	app
IPS	Sensor-1
DLP Profile	dlp

6. Click *OK* to create the policy.

To configure security profiles on an FTP proxy policy in the CLI:

```
config firewall proxy-policy
    edit 3
        set proxy ftp
```

```
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set utm-status enable
        set av-profile "av"
        set dlp-profile "dlp"
        set ips-sensor "sensor-1"
        set application-list "app"
    next
end
```

Explicit proxy authentication

FortiGate supports multiple authentication methods. This topic explains using an external authentication server with Kerberos as the primary and NTLM as the fallback.

To configure Explicit Proxy with authentication:

1. Enable and configure the explicit proxy on page 429.
2. Configure the authentication server and create user groups on page 430.
3. Create an authentication scheme and rules on page 432.
4. Create an explicit proxy policy and assign a user group to the policy on page 433.
5. Verify the configuration on page 434.

Enable and configure the explicit proxy

To enable and configure explicit web proxy in the GUI:

1. Go to *Network* > *Explicit Proxy*.
2. Enable *Explicit Web Proxy*.
3. Select *port2* as the *Listen on Interfaces* and set the *HTTP Port* to *8080*.
4. Configure the remaining settings as needed.
5. Click *Apply*.

To enable and configure explicit web proxy in the CLI:

```
config web-proxy explicit
    set status enable
    set ftp-over-http enable
    set socks enable
    set http-incoming-port 8080
    set ipv6-status enable
    set unknown-http-version best-effort
end
```

```
config system interface
    edit "port2"
        set vdom "vdom1"
        set ip 10.1.100.1 255.255.255.0
        set allowaccess ping https ssh snmp http telnet
        set type physical
        set explicit-web-proxy enable
        set snmp-index 12
        end
    next
end
```

Configure the authentication server and create user groups

Since we are using an external authentication server with Kerberos authentication as the primary and NTLM as the fallback, Kerberos authentication is configured first and then FSSO NTLM authentication is configured.

For successful authorization, the FortiGate checks if user belongs to one of the groups that is permitted in the security policy.

⚠ When configuring an LDAP connection to an Active Directory server, an administrator must provide Active Directory user credentials.

- To secure this connection, use LDAPS on both the Active Directory server and FortiGate. See Configuring an LDAP server on page 3509 and Configuring client certificate authentication on the LDAP server on page 3524.
- Apply the principle of least privilege. For the LDAP regular bind operation, do not use credentials that provide full administrative access to the Windows server when using credentials. See Configuring least privileges for LDAP admin account authentication in Active Directory on page 3517.

To configure an authentication server and create user groups in the GUI:

1. Configure Kerberos authentication:
   a. Go to *User & Authentication* > *LDAP Servers*.
   b. Click *Create New*.
   c. Set the following:

Name	ldap-kerberos
Server IP	172.18.62.220
Server Port	389
Common Name Identifier	cn
Distinguished Name	dc=fortinetqa,dc=local

   d. Click *OK*
2. Define Kerberos as an authentication service. This option is only available in the CLI. For information on generating a keytab, see Generating a keytab on a Windows server on page 435.

3. Configure FSSO NTLM authentication:

   FSSO NTLM authentication is supported in a Windows AD network. FSSO can also provide NTLM authentication service to the FortiGate unit. When a user makes a request that requires authentication, the FortiGate initiates NTLM negotiation with the client browser, but does not process the NTLM packets itself. Instead, it forwards all the NTLM packets to the FSSO service for processing.

   a. Go to *Security Fabric > External Connectors*.
   b. Click *Create New* and select *FSSO Agent on Windows AD* from the *Endpoint/Identity* category.
   c. Set the *Name* to *FSSO*, *Primary FSSO Agent* to *172.16.200.220*, and enter a password.
   d. Click *OK*.
4. Create a user group for Kerberos authentication:
   a. Go to *User & Authentication > User Groups*.
   b. Click *Create New*.
   c. Set the *Name* to *Ldap-Group*, and *Type* to *Firewall*.
   d. In the *Remote Groups* table, click *Add*, and set the *Remote Server* to the previously created *ldap-kerberos* server.
   e. Click *OK*.
5. Create a user group for NTLM authentication:
   a. Go to *User & Authentication > User Groups*.
   b. Click *Create New*.
   c. Set the *Name* to *NTLM-FSSO-Group*, *Type* to *Fortinet Single Sign-On (FSSO)*, and add *FORTINETQA/FSSO* as a member.
   d. Click *OK*.

To configure an authentication server and create user groups in the CLI:

1. Configure Kerberos authentication:

```
config user ldap
    edit "ldap-kerberos"
        set server "172.18.62.220"
        set cnid "cn"
        set dn "dc=fortinetqa,dc=local"
        set type regular
        set username "CN=root,CN=Users,DC=fortinetqa,DC=local"
        set password *********
    next
end
```

2. Define Kerberos as an authentication service:

```
config user krb-keytab
    edit "http_service"
        set pac-data disable
        set principal "HTTP/FGT.FORTINETQA.LOCAL@FORTINETQA.LOCAL"
        set ldap-server "ldap-kerberos"
        set keytab
"BQIAAABFAAIAEEZPUlRJTkVUUUEuTE9DQUwABEhUVFAAFEZHVC5GT1JUSU5FVFFBLkxPQ0FMAAAAAQAAAAAEAAEACKLCM
```

```
onpitnVAAAARQACABBGT1JUSU5FVFFBLkxPQ0FMAARIVFRQABRGR1QuRk9SVElORVRRQS5MT0NBTAAAAAEAAAAABAADAAi
iwjKJ6YrZ1QAAAE0AAgAQRk9SVElORVRRQS5MT0NBTAAESFRUUAAURkdULkZPUlRJTkVUUUEuTE9DQUwAAAABAAAAAAQAF
wAQUHo9uqR9cSkzyxdzKCEXdwAAAF0AAgAQRk9SVElORVRRQS5MT0NBTAAESFRUUAAURkdULkZPUlRJTkVUUUEuTE9DQUw
AAAABAAAAAAQAEgAgzee854Aq1HhQiKJZvV4tL2Poy7hMIARQpK8MCB//BIAAAABNAAIAEEZPUlRJTkVUUUEuTE9DQUwAB
EhUVFAAFEZHVC5GT1JUSU5FVFFBLkxPQ0FMAAAAAQAAAAAEABEAEG49vHEiiBghr63Z/lnwYrU="
    next
end
```

For information on generating a keytab, see Generating a keytab on a Windows server on page 435.

3. Configure FSSO NTLM authentication:

```
config user fsso
    edit "1"
        set server "172.18.62.220"
        set password *********
    next
end
```

4. Create a user group for Kerberos authentication:

```
config user group
    edit "Ldap-Group"
        set member "ldap" "ldap-kerberos"
    next
end
```

5. Create a user group for NTLM authentication:

```
config user group
    edit "NTLM-FSSO-Group"
        set group-type fsso-service
        set member "FORTINETQA/FSSO"
    next
end
```

Create an authentication scheme and rules

Explicit proxy authentication is managed by authentication schemes and rules. An authentication scheme must be created first, and then the authentication rule.

To create an authentication scheme and rules in the GUI:

1. Create an authentication scheme:
   a. Go to *Policy & Objects* > *Authentication* and select the *Authentication Schemes* tab.
   b. Click *Create New*.
   c. Set the *Name* to *Auth-scheme-Negotiate* and select *Negotiate* as the *Method*.
   d. Click *OK*.
2. Create an authentication rule:

a. Go to *Policy & Objects* > *Authentication* and select the *Authentication Rules* tab.
b. Click *Create New*.
c. Set the *Name* to *Auth-Rule*, *Source Address* to *all*, and *Protocol* to *HTTP*.
d. Enable *Authentication Scheme*, and select the just created *Auth-scheme-Negotiate* scheme.
e. Click *OK*.

To create an authentication scheme and rules in the CLI:

1. Create an authentication scheme:

```
config authentication scheme
    edit "Auth-scheme-Negotiate"
        set method negotiate       <<< Accepts both Kerberos and NTLM as fallback
    next
end
```

2. Create an authentication rule:

```
config authentication rule
    edit "Auth-Rule"
        set status enable
        set protocol http
        set srcaddr "all"
        set ip-based enable
        set active-auth-method "Auth-scheme-Negotiate"
        set comments "Testing"
    next
end
```

Create an explicit proxy policy and assign a user group to the policy

To create an explicit proxy policy and assign a user group to it in the GUI:

1. Go to *Policy & Objects* > *Proxy Policy*.
2. Click *Create New*.
3. Set *Proxy Type* to *Explicit Web* and *Outgoing Interface* to *port1*.
4. Set *Source* to *all*, and the just created user groups *NTLM-FSSO-Group* and *Ldap-Group*.
5. Also set *Destination* to *all*, *Schedule* to *always*, *Service* to *webproxy*, and *Action* to *ACCEPT*.
6. Click *OK*.

To create an explicit proxy policy and assign a user group to it in the CLI:

```
config firewall proxy-policy
    edit 1
        set proxy explicit-web
        set dstintf "port1"
        set srcaddr "all"
```

```
        set dstaddr "all"
        set service "web"
        set action accept
        set schedule "always"
        set logtraffic all
        set groups "NTLM-FSSO-Group" "Ldap-Group"
        set av-profile "av"
        set ssl-ssh-profile "deep-custom"
    next
end
```

Verify the configuration

Log in using a domain and system that would be authenticated using the Kerberos server, then enter the `diagnose wad user list` CLI command to verify:

```
# diagnose wad user list
ID: 8, IP: 10.1.100.71, VDOM: vdom1
  user name    : test1@FORTINETQA.LOCAL
  duration     : 389
  auth_type    : IP
  auth_method : Negotiate
  pol_id       : 1
  g_id         : 1
  user_based   : 0
  expire       : no
  LAN:
    bytes_in=4862 bytes_out=11893
  WAN:
    bytes_in=7844 bytes_out=1023
```

Log in using a system that is not part of the domain. The NTLM fallback server should be used:

```
# diagnose wad user list
ID: 2, IP: 10.1.100.202, VDOM: vdom1
  user name    : TEST31@FORTINETQA
  duration     : 7
  auth_type    : IP
  auth_method : NTLM
  pol_id       : 1
  g_id         : 5
  user_based   : 0
  expire       : no
  LAN:
    bytes_in=6156 bytes_out=16149
  WAN:
    bytes_in=7618 bytes_out=1917
```

Generating a keytab on a Windows server

A keytab is used to allow services that are not running Windows to be configured with service instance accounts in the Active Directory Domain Service (AD DS). This allows Kerberos clients to authenticate to the service through Windows Key Distribution Centers (KDCs).

For an explanation of the process, see https://docs.microsoft.com/en-us/windows-server/administration/windows-commands/ktpass.

To generate a keytab on a Windows server:

1. On the server, create a user for the FortiGate:
   - The service name is the FQDN for the explicit proxy interface, such as the hostname in the client browser proxy configuration. In this example, the service name is *FGT*.
   - The account only requires *domain users* membership.
   - The password must be very strong.
   - The password is set to never expire.
2. Add the FortiGate FQDN in to the Windows DNS domain, as well as in-addr.arpa.
3. Generate the Kerberos keytab using the `ktpass` command on Windows servers and many domain workstations:

   ```
   # ktpass -princ HTTP/<domain name of test fgt>@realm -mapuser <user> -pass <password> -crypto all -ptype KRB5_NT_PRINCIPAL -out fgt.keytab
   ```

   For example:

   ```
   ktpass -princ HTTP/FGT.FORTINETQA.LOCAL@FORTINETQA.LOCAL -mapuser FGT -pass ********** -crypto all -ptype KRB5_NT_PRINCIPAL -out fgt.keytab
   ```

   If the FortiGate is handling multiple keytabs in Kerberos authentication, use different passwords when generating each keytab.

4. Encode the keytab to base64 in a text file:
   - On Windows: `certutil -encode fgt.keytab tmp.b64 && findstr /v /c:- tmp.b64 > fgt.txt`
   - On Linux: `base64 fgt.keytab > fgt.txt`
   - On MacOS: `base64 -i fgt.keytab -o fgt.txt`
5. Use the code in `fgt.txt` as the keytab parameter when configuring the FortiGate.

Transparent web proxy forwarding

In FortiOS, there is an option to enable proxy forwarding for transparent web proxy policies and regular firewall policies for HTTP and HTTPS.

In previous versions of FortiOS, you could forward proxy traffic to another proxy server (proxy chaining) with explicit proxy. Now, you can forward web traffic to the upstream proxy without having to reconfigure your browsers or publish a proxy auto-reconfiguration (PAC) file.

Once configured, the FortiGate forwards traffic generated by a client to the upstream proxy. The upstream proxy then forwards it to the server.

To configure proxy forwarding:

1. Configure the web proxy forwarding server:

```
config web-proxy forward-server
    edit "upStream_proxy_1"
        set ip 172.16.200.20
        set healthcheck enable
        set monitor "http://www.google.ca"
    next
end
```

2. Append the web proxy forwarding server to a firewall policy:

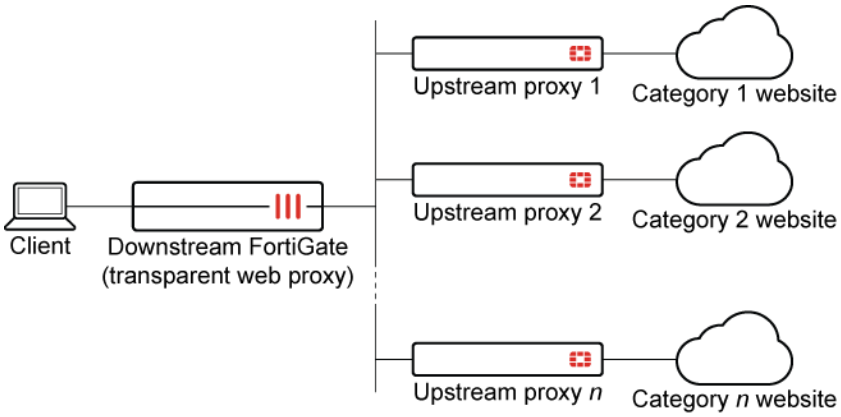
```
config firewall policy
    edit 1
        set name "LAN To WAN"
        set srcintf "port10"
        set dstintf "port9"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set logtraffic all
        set webproxy-forward-server "upStream_proxy_1"
        set fsso disable
        set av-profile "av"
        set ssl-ssh-profile "deep-custom"
        set nat enable
    next
end
```

Selectively forward web requests to a transparent web proxy

Web traffic over HTTP/HTTPS can be forwarded selectively by the FortiGate's transparent web proxy to an upstream web proxy to avoid overwhelming the proxy server. Traffic can be selected by specifying the proxy address, which can be based on a FortiGuard URL category.

The FortiGuard web filter service must be enabled on the downstream FortiGate.

Topology



Forwarding behavior

The forward server will be ignored if the proxy policy matching for a particular session needs the FortiGate to see authentication information inside the HTTP (plain text) message. For example, assume that user authentication is required and a forward server is configured in the transparent web proxy, and the authentication method is an active method (such as basic). When the user or client sends the HTTP request over SSL with authentication information to the FortiGate, the request cannot be forwarded to the upstream proxy. Instead, it will be forwarded directly to the original web server (assuming deep inspection and `http-policy-redirect` are enabled in the firewall policy).

The FortiGate will close the session before the client request can be forwarded if all of the following conditions are met:

- The certificate inspection is configured in the firewall policy that has the `http-policy-redirect` option enabled.
- A previously authenticated IP-based user record cannot be found by the FortiGate's memory during the SSL handshake.
- Proxy policy matching needs the FortiGate to see the HTTP request authentication information.

This means that in order to enable user authentication and use `webproxy-forward-server` in the transparent web proxy policy at the same time, the following best practices should be followed:

- In the firewall policy that has the `http-policy-redirect` option enabled, set `ssl-ssh-profile` to use the `deep-inspection` profile.
- Use IP-based authentication rules; otherwise, the `webproxy-forward-server` setting in the transparent web proxy policy will be ignored.
- Use a passive authentication method such as FSSO. With FSSO, once the user is authenticated as a domain user by a successful login, the web traffic from the user's client will always be forwarded to the upstream proxy as long as the authenticated user remains unexpired. If the authentication method is an active authentication method (such as basic, digest, NTLM, negotiate, form, and so on), the first session containing authentication information will bypass the forward server, but the following sessions will be connected through the upstream proxy.

Sample configuration

On the downstream FortiGate proxy, there are two category proxy addresses used in two separate transparent web proxy policies as the destination address:

- In the policy with `upstream_proxy_1` as the forward server, the proxy address `category_infotech` is used to match URLs in the information technology category.
- In the policy with `upstream_proxy_2` as the forward server, the proxy address `category_social` is used to match URLs in the social media category.

To configure forwarding requests to transparent web proxies:

1. Configure the proxy forward servers:

```
config web-proxy forward-server
    edit "upStream_proxy_1"
        set ip 172.16.200.20
    next
    edit "upStream_proxy_2"
        set ip 172.16.200.46
    next
end
```

2. Configure the web proxy addresses:

```
config firewall proxy-address
    edit "category_infotech"
        set type category
        set host "all"
        set category 52
    next
    edit "category_social"
        set type category
        set host "all"
        set category 37
    next
end
```

3. Configure the firewall policy:

```
config firewall policy
    edit 1
        set srcintf "port10"
        set dstintf "port9"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set http-policy-redirect enable
```

```
        set ssl-ssh-profile "deep-inspection"
        set av-profile "av"
        set nat enable
    next
end
```

4. Configure the proxy policies:

```
config firewall proxy-policy
    edit 1
        set proxy transparent-web
        set srcintf "port10"
        set dstintf "port9"
        set srcaddr "all"
        set dstaddr "category_infotech"
        set service "webproxy"
        set action accept
        set schedule "always"
        set logtraffic all
        set webproxy-forward-server "upStream_proxy_1"
        set utm-status enable
        set ssl-ssh-profile "deep-inspection"
        set av-profile "av"
    next
    edit 2
        set proxy transparent-web
        set srcintf "port10"
        set dstintf "port9"
        set srcaddr "all"
        set dstaddr "category_social"
        set service "webproxy"
        set action accept
        set schedule "always"
        set logtraffic all
        set webproxy-forward-server "upStream_proxy_2"
        set utm-status enable
        set ssl-ssh-profile "deep-inspection"
        set av-profile "av"
    next
end
```

Transparent web proxy forwarding over IPv6

The IPv6-enabled forward server works the same way as the IPv4 forward server. For example, you can configure an IPv6 address or an FQDN that resolves to an IPv6 address for the forward server, and you can also use the IPv6 forward server in a forward server group.

```
config web-proxy forward-server
    edit <name>
        set addr-type {ip | ipv6 | fqdn}
```

```
        set ipv6 <IPv6-address>
    next
end
```

Command	Description
`addr-type`	Specify the type of IP address for the web proxy forward server: • `ip`: use an IPv4 address. • `ipv6`: use an IPv6 address. • `fqdn`: use a fully qualified domain name (FQDN).
`ipv6`	Specify the IPv6 address for the web proxy forward server. Available when `addr-type` is set to `ipv6`.

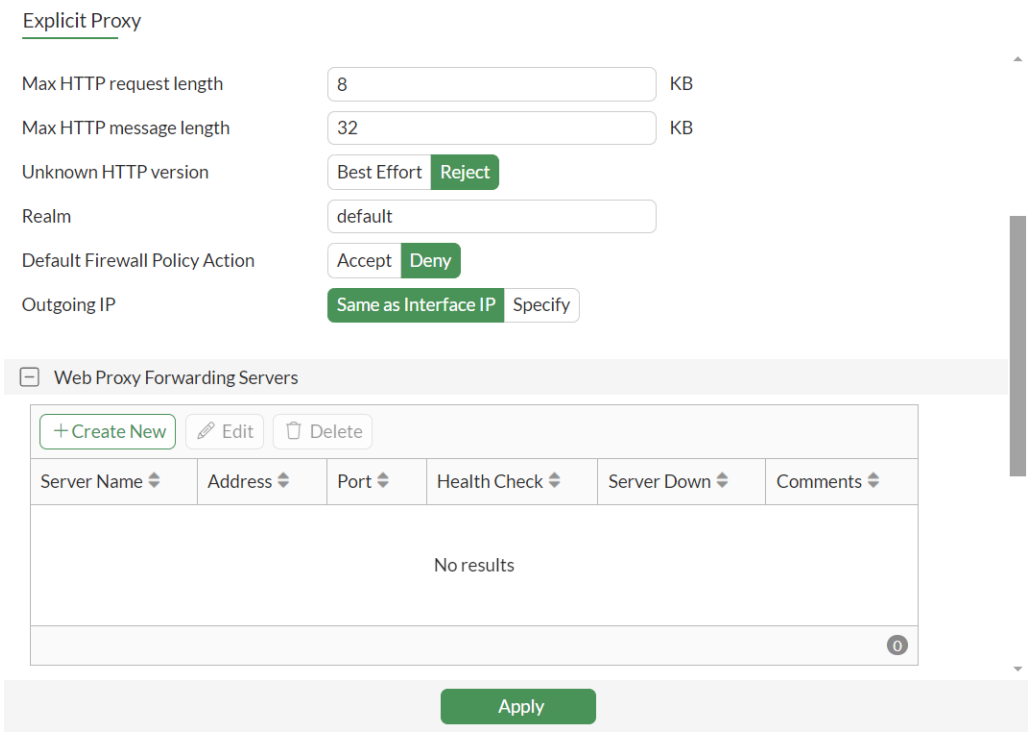
Example

In this example, an explicit web proxy with a forward server can be reached by an IPv6 address, and a client PC uses this explicit web proxy forward server to access a website, such as www.google.com.

The IPv6 address is configured for the web proxy forward server, and then the configuration is added to a proxy policy. The web proxy forward server configuration could also be added to a proxy mode policy or a transparent web proxy policy.

To configure an IPv6 address in the GUI:

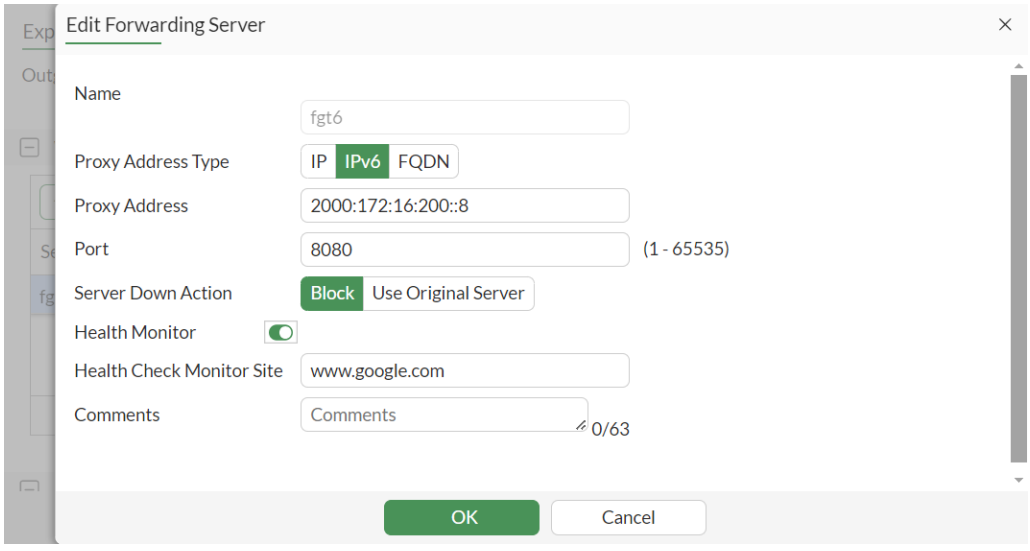
1. Go to *Network* > *Explicit Proxy*.
2. If disabled, enable *Explicit Web Proxy*.
3. Under *Web Proxy Forwarding Servers*, click *Create New*.



The *New Forwarding Server* pane opens.

4. Set the following options, and click *OK* to create the forwarding server.

Proxy Address Type	IPv6
Proxy Address	2000:172:16:200::8
Port	8080
Health Monitor	Enable
Health Check Monitor Site	www.google.com



5. Set the remaining options as needed, and click *OK* to save the explicit web proxy.

6. Add the web proxy forward server to a proxy policy.

To configure an IPv6 address in the CLI:

1. Configure an IPv6 address for the web proxy forward server.
   In this example, address type is set to IPv6, and an IPv6 address is specified in a configuration (fgt6) for a web proxy forward server.

```
config web-proxy forward-server
    edit "fgt6"
        set addr-type ipv6
        set ipv6 2000:172:16:200::8
        set port 8080
    next
end
```

2. Add the web proxy forward server to a proxy policy.
   The web proxy forward server configuration (fgt6) is added to the firewall proxy policy.

```
config firewall proxy-policy
    edit 1
        set uuid 560d8520-fa7b-51ed-e06a-df05ec145542
        set proxy explicit-web
        set dstintf "port3"
        set srcaddr "all"
        set dstaddr "all"
        set service "webproxy"
        set action accept
        set schedule "always"
        set logtraffic all
        set srcaddr6 "all"
        set dstaddr6 "all"
        set webproxy-forward-server "fgt6"
        set utm-status enable
        set ssl-ssh-profile "deep-custom"
        set av-profile "av"
    next
end
```

3. View the traffic logs.
   An HTTP request to www.google.com was sent through the web proxy forward server over IPv6.

```
12: date=2023-08-10 time=23:44:43 eventtime=1691736283529768562 tz="-0700" logid="0000000010"
type="traffic" subtype="forward" level="notice" vd="vdom1" srcip=2000:10:1:100::11
srcport=44190 srcintf="port1" srcintfrole="undefined" dstcountry="United States"
srccountry="Reserved" dstip=2607:f8b0:400a:807::2004 dstport=80 dstintf="port3"
dstintfrole="undefined" sessionid=391251274 service="HTTP" proxyapptype="web-proxy" proto=6
action="accept" policyid=1 policytype="proxy-policy" poluuid="560d8520-fa7b-51ed-e06a-
df05ec145542" trandisp="snat+dnat" tranip=2000:172:16:200::8 tranport=8080
transip=2000:172:16:200::2 transport=21344 duration=22 wanin=2385 rcvdbyte=2385 wanout=369
lanin=129 sentbyte=129 lanout=795 appcat="unscanned"
```

Upstream proxy authentication in transparent proxy mode

A downstream proxy FortiGate that needs to be authenticated by the upstream web proxy can use the basic authentication method to send its username and password, in the base64 format, to the upstream web proxy for authentication. If the authentication succeeds, web traffic that is forwarded from the downstream proxy FortiGate to the upstream proxy can be accepted and forwarded to its destinations.

In this example, a school has a FortiGate acting as a downstream proxy that is configured with firewall policies for each user group (students and staff). In each policy, a forwarding server is configured to forward the web traffic to the upstream web proxy.

The username and password that the upstream web proxy uses to authenticate the downstream proxy are configured on the forwarding server, and are sent to the upstream web proxy with the forwarded HTTP requests.

	Username	Password
student.proxy.local:8080	students	ABC123
staff.proxy.local:8081	staff	123456

On the downstream FortiGate, configure forwarding servers with the usernames and passwords for authentication on the upstream web proxy, then apply those servers to firewall policies for transparent proxy. For explicit web proxy, the forwarding servers can be applied to proxy policies.

When the transparent proxy is configured, clients can access websites without configuring a web proxy in their browser. The downstream proxy sends the username and password to the upstream proxy with forwarded HTTP requests to be authenticated.

To configure the forwarding server on the downstream FortiGate:

```
config web-proxy forward-server
    edit "Student_Upstream_WebProxy"
        set addr-type fqdn
        set fqdn "student.proxy.local"
        set port 8080
        set username "student"
        set password ABC123
    next
    edit "Staff_Upstream_WebProxy"
        set addr-type fqdn
        set fqdn "staff.proxy.local"
        set port 8081
        set username "staff"
        set password 123456
    next
end
```

To configure firewall policies for transparent proxy:

```
config firewall policy
    edit 1
        set srcintf "Vlan_Student"
        set dstintf "port9"
        set srcaddr "Student_Subnet"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "deep-inspection"
        set av-profile "av"
        set webproxy-forward-server "Student_Upstream_WebProxy"
        set nat enable
    next
    edit 2
        set srcintf "Vlan_Staff"
        set dstintf "port9"
        set srcaddr "Staff_Subnet"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "deep-inspection"
        set av-profile "av"
        set webproxy-forward-server "Staff_Upstream_WebProxy"
        set nat enable
    next
end
```

Multiple dynamic header count

Multiple dynamic headers are supported for web proxy profiles, as well as Base64 encoding and the append/new options.

Administrators only have to select the dynamic header in the profile. The FortiGate will automatically display the corresponding static value. For example, if the administrator selects the `$client-ip` header, the FortiGate will display the actual client IP address.

The supported headers are:

`$client-ip`	Client IP address
`$user`	Authentication user name
`$domain`	User domain name

$local_grp	Firewall group name
$remote_grp	Group name from authentication server
$proxy_name	Proxy realm name

To configure dynamic headers using the CLI:

Since authentication is required, FSSO NTLM authentication is configured in this example.

1. Configure LDAP:

```
config user ldap
    edit "ldap-kerberos"
        set server "172.18.62.220"
        set cnid "cn"a
        set dn "dc=fortinetqa,dc=local"
        set type regular
        set username "CN=root,CN=Users,DC=fortinetqa,DC=local"
        set password *********
    next
end
```

2. Configure FSSO:

```
config user fsso
    edit "1"
        set server "172.18.62.220"
        set password *********
    next
end
```

3. Configure a user group:

```
config user group
    edit "NTLM-FSSO"
        set group-type fsso-service
        set member "FORTINETQA/FSSO"
    next
end
```

4. Configure an authentication scheme:

```
config authentication scheme
    edit "au-sch-ntlm"
        set method ntlm
    next
end
```

5. Configure an authentication rule:

```
config authentication rule
    edit "au-rule-fsso"
        set srcaddr "all"
```

```
        set active-auth-method "au-sch-ntlm"
    next
end
```

6. Create a web proxy profile that adds a new dynamic and custom `Via` header:

```
config web-proxy profile
    edit "test"
        set log-header-change enable
        config headers
            edit 1
                set name "client-ip"
                set content "$client-ip"
            next
            edit 2
                set name "Proxy-Name"
                set content "$proxy_name"
            next
            edit 3
                set name "user"
                set content "$user"
            next
            edit 4
                set name "domain"
                set content "$domain"
            next
            edit 5
                set name "local_grp"
                set content "$local_grp"
            next
            edit 6
                set name "remote_grp"
                set content "$remote_grp"
            next
            edit 7
                set name "Via"
                set content "Fortigate-Proxy"
            next
        end
    next
end
```

7. In the proxy policy, append the web proxy profile created in the previous step:

```
config firewall proxy-policy
    edit 1
        set proxy explicit-web
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set service "web"
        set action accept
```

```
        set schedule "always"
        set logtraffic all
        set groups "NTLM-FSSO"
        set webproxy-profile "test"
        set utm-status enable
        set av-profile "av"
        set webfilter-profile "content"
        set ssl-ssh-profile "deep-custom"
    next
end
```

8. Once traffic is being generated from the client, look at the web filter logs to verify that it is working.
   The corresponding values for all the added header fields are shown in the *Web Filter* card at *Log & Report* > *Security Events*, in the *Change headers* section at the bottom of the *Log Details* pane.

```
1: date=2019-02-07 time=13:57:24 logid="0344013632" type="utm" subtype="webfilter"
eventtype="http_header_change" level="notice" vd="vdom1" eventtime=1549576642 policyid=1
transid=50331689 sessionid=1712788383 user="TEST21@FORTINETQA" group="NTLM-FSSO"
profile="test" srcip=10.1.100.116 srcport=53278 dstip=172.16.200.46 dstport=80 srcintf="port2"
srcintfrole="undefined" dstintf="port1" dstintfrole="undefined" proto=6 service="HTTP"
url="http://172.16.200.46/" agent="curl/7.22.0" chgheaders="Added=client-ip:
10.1.100.116|Proxy-Name: 1.1 100D.qa|user: TEST21|domain: FORTINETQA|local_grp: NTLM-
FSSO|remote_grp: FORTINETQA/FSSO|Via: Fortigate-Proxy"
```

Restricted SaaS access

Large organizations may want to restrict SaaS access to resources like Microsoft Office 365, Google Workspace, and Dropbox by tenant to block non-company login attempts and secure the users from accessing non-approved cloud resources. Many cloud vendors enable this by applying tenant restrictions for access control. For example, users accessing Microsoft 365 applications with tenant restrictions through the corporate proxy will only be allowed to log in as the company's tenant and access the organization's applications.

To implement this, access requests from the clients pass through the company's web proxy, which inserts headers to notify the SaaS service to apply tenant restrictions with the permitted tenant list. Users are redirected the SaaS service login page, and are only allowed to log in if they belong to the permitted tenant list.

For more information, refer to the vendor-specific documentation:

- Office 365: Restrict access to a tenant
- Google Workspace: Block access to consumer accounts
- Dropbox: Network control

Basic configuration

A web proxy profile can specify access permissions for Microsoft Office 365, Google Workspace, and Dropbox by inserting vendor-defined headers that restrict access to the specific accounts. Custom headers can also be inserted for any destination. The web proxy profile can then be applied to a firewall policy to control the header's insertion.

To implement Office 365 tenant restriction, Google Workspace account access control, and Dropbox network access control:

1. Configure a web proxy profile according to the vendors' specifications:
   a. Set the header name (defined by the service provider).
   b. Set the traffic destination (the service provider).
   c. Set the HTTP header content to be inserted into the traffic (defined by your settings).

```
config web-proxy profile
    edit <name>
        config headers
            edit <id>
                set name <string>
                set dstaddr <address>
                set action add-to-request
                set base64-encoding disable
                set add-option new
                set protocol https http
                set content <string>
            next
        end
    next
end
```

2. Apply the web proxy profile to a policy. SSL deep inspection must be used in the firewall policy:

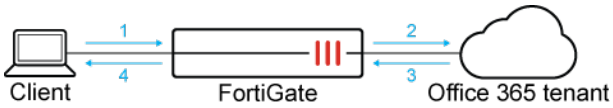
The following table lists the vendor-specific `config headers` settings that must be configured in the web proxy profile (`config web-proxy profile`):

Setting	Vendor specification		
	Microsoft Office 365	Google Workspace	Dropbox
`name <string>`	• `Restrict-Access-To-Tenants` • `Restrict-Access-Context`	• `X-GoogApps-Allowed-Domains`	• `X-Dropbox-allowed-Team-Ids`
`dstaddr <address>`	• Use the built-in `Microsoft Office 365` address.	• Use the built-in `G Suite` address.	• Use the built-in `wildcard.dropbox.com` address.
`content <string>`	• Enter the domain for `Restrict-Access-To-Tenants`. • Enter the directory ID for `Restrict-Access-Context`.	• Enter the domain.	• Enter the Dropbox team ID.

Due to vendors' changing requirements, these settings may no longer comply with the vendors' official guidelines. See the vendor documentation for more details.

Microsoft Office 365 example

In this example, a web proxy profile is created to control permissions for Microsoft Office 365 to allow corporate domains and deny personal accounts, such as Hotmail and Outlook that are accessed through login.live.com.



1. When a user attempts to access login.microsoftonline.com, login.microsoft.com, or login.windows.net, the traffic will match a proxy inspection mode firewall policy with the assigned web proxy profile.
2. The web proxy profile adds new headers to the customer tenant, indicating the allowed domain and restricted access for personal accounts. Next, the FortiGate starts a new connection with the Microsoft Office 365 domain controller including the new headers.
3. The Microsoft Office 365 domain controller assesses this data and will allow or deny this access, then sends a reply to the FortiGate.
4. The FortiGate sends a reply to the client.

The FortiGate will only indicate the correct domains to be allowed or denied through the headers to Microsoft. The custom sign-in portal in the browser is generated by Microsoft.

Configuration summary

The following must be configured in FortiOS:

- An FQDN address for login.live.com
- An SSL inspection profile that applies deep inspection for login.live.com

  Ensure that the firewall certificate is installed on the client machines. A company certificate signed by an internal CA is recommended.

- A web filter profile in proxy mode with static URL filters for the SNI URLs
- A web proxy profile that adds new headers to the customer tenant
- A firewall policy using proxy mode inspection that applies the configured SSL SSL inspection, web filter, and web proxy profiles

The `Restrict-Access-To-Tenants` and `Restrict-Access-Context` headers are inserted for incoming requests to: login.microsoftonline.com, login.microsoft.com, and login.windows.net, which are part of the `Microsoft Office 365` address group.

To restrict access to personal accounts using the login.live.com domain, the `sec-Restrict-Tenant-Access-Policy` header is inserted and uses `restrict-msa` as the header content.

Before configuring the FortiGate, collect the information related to the company domain in the Office 365 contract.

- `Restrict-Access-To-Tenants`: your <domain.com>
- `Restrict-Access-Context`: Directory ID

To find the Directory ID related to the domain, locate it in the Azure portal, or use the whatismytenantid.com open tool.

To configure the FortiGate:

1. Add the FQDN address for login.live.com:

```
config firewall address
    edit "login.live.com"
        set type fqdn
        set fqdn "login.live.com"
    next
end
```

2. Configure the SSL inspection profile. In this example, the `deep-inspection` profile is cloned, and the `live.com` FQDN is removed from the exemption list.
   a. Clone the `deep-inspection` profile:

```
config firewall ssl-ssh-profile
    clone "deep-inspection" to "Tenant"
end
```

   b. Edit the `Tenant` profile and remove `live.com` from the `config ssl-exempt` list.
3. Configure the URL filter list:

```
config webfilter urlfilter
    edit 1
        set name "Auto-webfilter-urlfilter"
        config entries
            edit 1
                set url "login.microsoftonline.com"
                set action allow
            next
            edit 2
                set url "login.microsoft.com"
                set action allow
            next
            edit 3
                set url "login.windows.net"
                set action allow
            next
            edit 4
                set url "login.live.com"
                set action allow
            next
        end
    next
end
```

4. Configure the web filter profile:

```
config webfilter profile
    edit "Tenant"
        set comment "Office 365"
        set feature-set proxy
        config web
            set urlfilter-table 1
        end
    next
end
```

5. Configure the web proxy profile (enter the header names exactly as shown):

```
config web-proxy profile
    edit "SaaS-Tenant-Restriction"
        set header-client-ip pass
        set header-via-request pass
        set header-via-response pass
        set header-x-forwarded-for pass
        set header-x-forwarded-client-cert pass
        set header-front-end-https pass
        set header-x-authenticated-user pass
        set header-x-authenticated-groups pass
        set strip-encoding disable
        set log-header-change disable
        config headers
            edit 1
                set name "Restrict-Access-To-Tenants"
                set dstaddr "login.microsoft.com" "login.microsoftonline.com"
"login.windows.net"
                set action add-to-request
                set base64-encoding disable
                set add-option new
                set protocol https http
                set content <domain>
            next
            edit 2
                set name "Restrict-Access-Context"
                set dstaddr "login.microsoftonline.com" "login.microsoft.com"
"login.windows.net"
                set action add-to-request
                set base64-encoding disable
                set add-option new
                set protocol https http
                set content <directory_ID>
            next
            edit 3
                set name "sec-Restrict-Tenant-Access-Policy"
                set dstaddr "login.live.com"
                set action add-to-request
                set base64-encoding disable
                set add-option new
                set protocol https http
```

```
                set content "restrict-msa"
            next
        end
    next
end
```

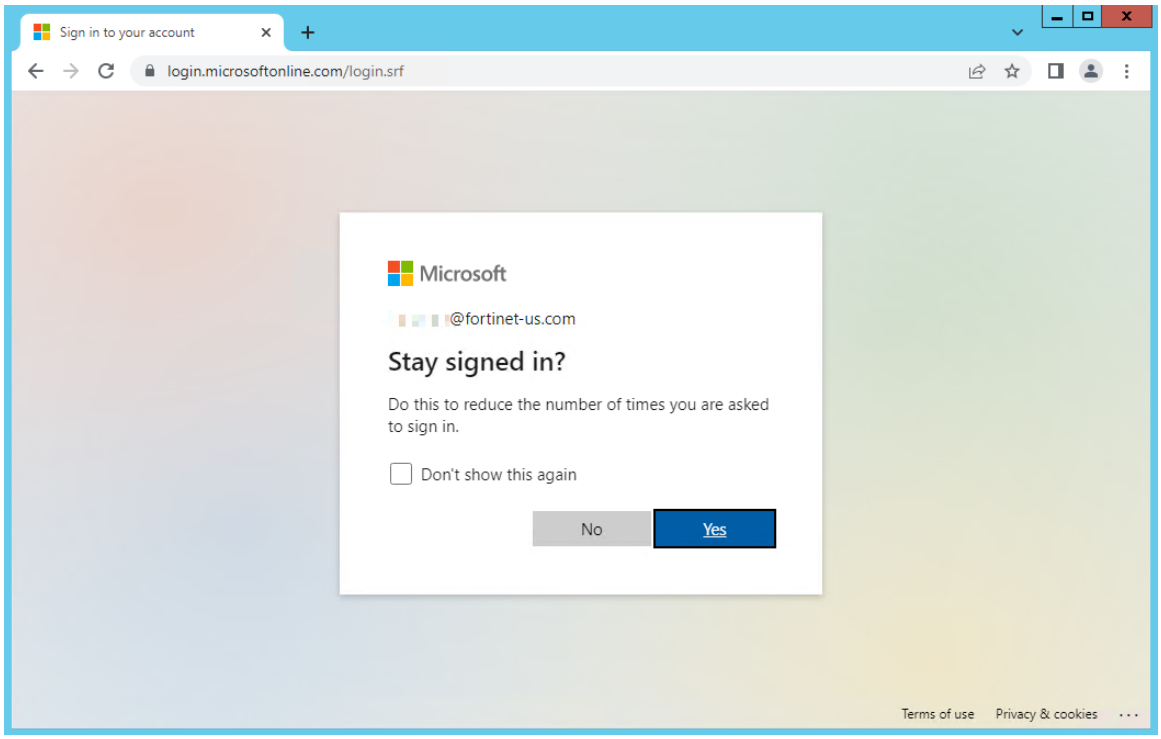
6. Configure the firewall policy:

```
config firewall policy
    edit 10
        set name "Tenant"
        set srcintf "port2"
        set dstintf "port1"
        set action accept
        set srcaddr "users-lan"
        set dstaddr "login.microsoft.com" "login.microsoftonline.com" "login.windows.net"
"login.live.com"
        set schedule "always"
        set service "HTTP" "HTTPS"
        set utm-status enable
        set inspection-mode proxy
        set webproxy-profile "SaaS-Tenant-Restriction"
        set ssl-ssh-profile "Tenant"
        set webfilter-profile "Tenant"
        set logtraffic all
        set nat enable
    next
end
```

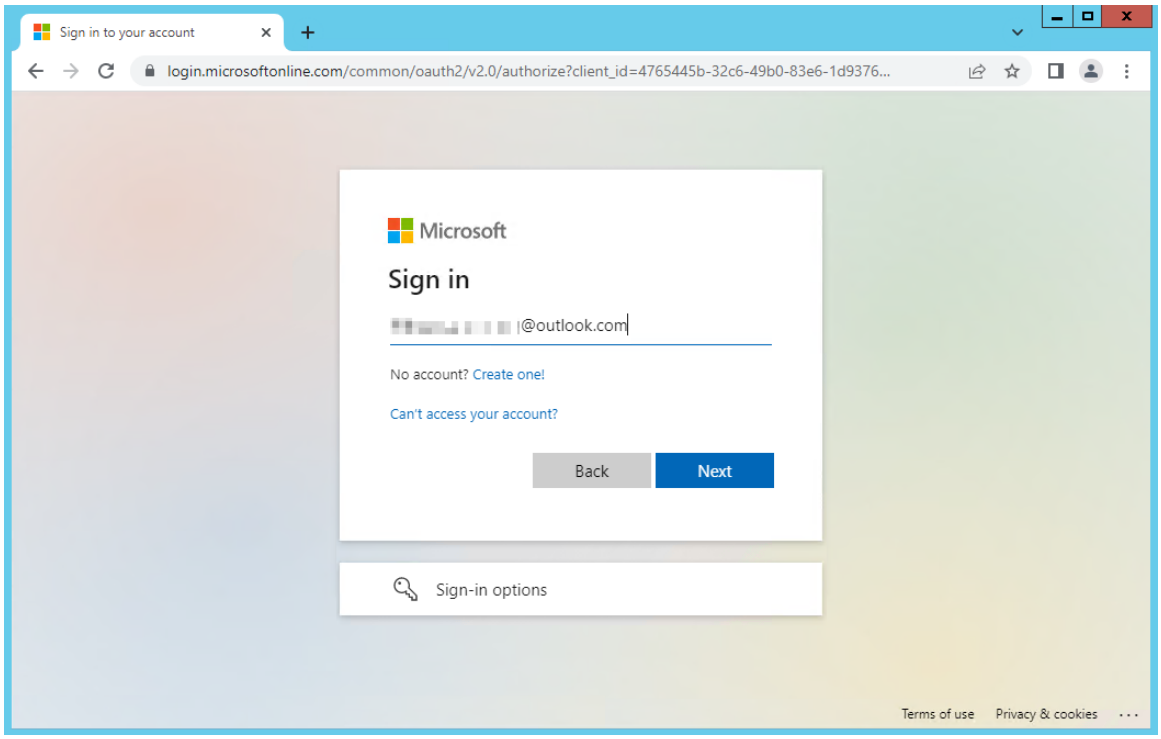
Testing the access

To test the access to corporate domains and personal accounts:

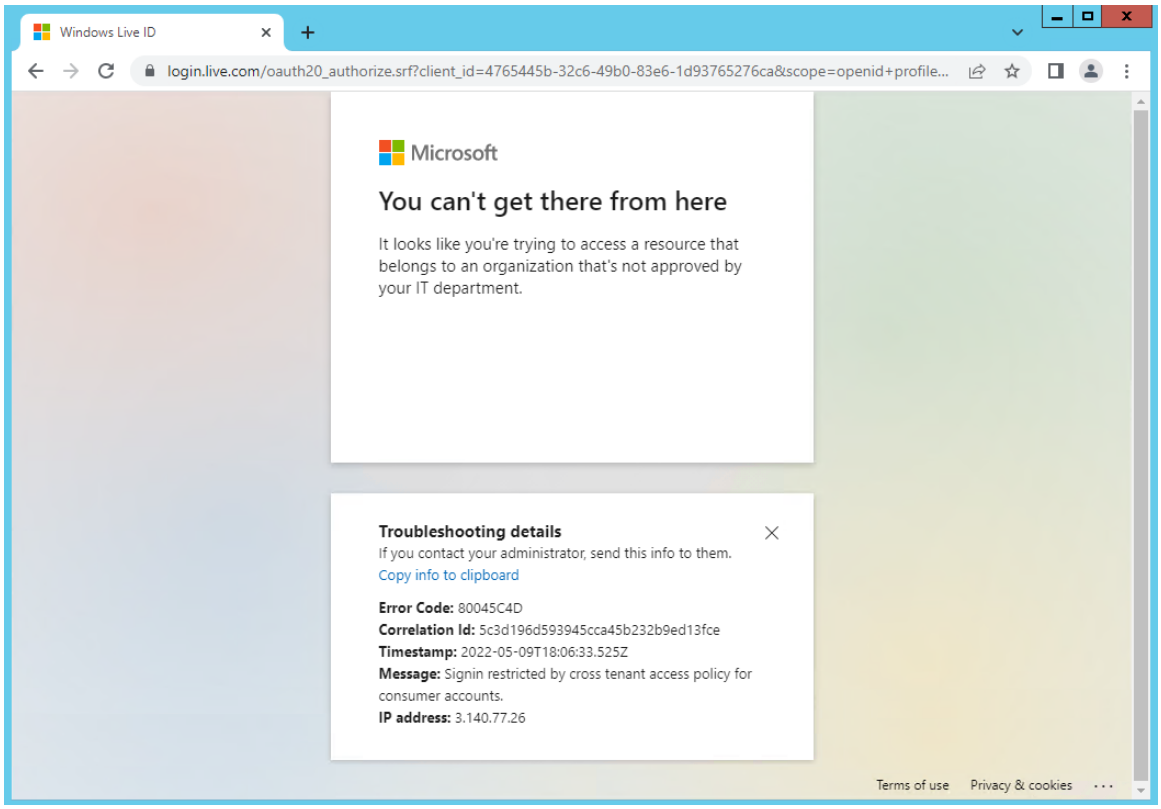
1. Get a client to log in with their corporate email using the login.microsoftonline.com domain.



2. The client is able to enter their credentials and log in successfully.
3. Get a client to log in to their personal Outlook account.



4. After the client enters their credentials, a message appears that they cannot access this resource because it is restricted by the cross-tenant access policy.



Verifying the header insertion

To verify the header insertion for corporate domains and personal accounts:

1. On the FortiGate, start running the WAD debugs:

```
# diagnose wad debug enable category http
# diagnose wad debug enable level info
# diagnose debug enable
```

2. After a client attempts to access corporate domains, verify that the header information is sent to the Microsoft Active Directory:

```
[I][p:234][s:2481][r:33] wad_dump_fwd_http_req             :2567  hreq=0x7fc75f0cd468 Forward
request to server:
POST /common/GetCredentialType?mkt=en-US HTTP/1.1
Host: login.microsoftonline.com
Connection: keep-alive
Content-Length: 1961
sec-ch-ua: " Not A;Brand";v="99", "Chromium";v="101", "Google Chrome";v="101"
hpgrequestid: d7f706a8-1143-4cdd-ad52-1cc69dc7bb00
sec-ch-ua-mobile: ?0
User-Agent: Mozilla/5.0 (Windows NT 6.3; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/101.0.4951.54 Safari/537.36
client-request-id: 5c3d196d-5939-45cc-a45b-232b9ed13fce
...
Restrict-Access-To-Tenants: fortinet-us.com
Restrict-Access-Context: ********-****-452f-8535-************
```

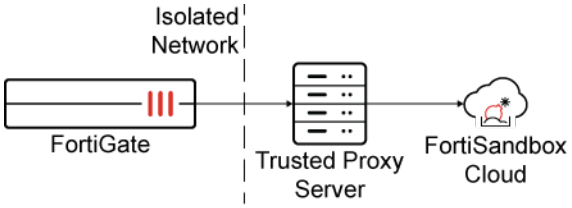
3. After a client attempts to access a personal account, verify that the header information is sent to the Microsoft Active Directory:

```
[I][p:234][s:2519][r:34] wad_dump_fwd_http_req             :2567  hreq=0x7fc75f0ce6a8 Forward
request to server:
GET /oauth20_authorize.srf?client_id=4765445b-32c6-49b0-83e6-
1d93765276ca&scope=openid+profile+https%3a%2f%2fwww.office.com%2fv2%2fOfficeHome.All&redirect_
uri=https%3a%2f%2fwww.office.com%2flandingv2&response_type=code+id_
token&state=7tAtndYhcA3132S--UOTyLVEtyIZs8FgndTpeYM9mJ1EeA-
X5nfqrSalnnPH41cHxfHGug6N5cbliK676v6xZgszgH_
JARVKrptZwBvjI2cbnZ4mttYNNdK1FTlbEtu5VBjgtBOX2u6v3F_
9g7UikCpGTnBRGhvO2pyTndT3EEIyAHvhg9LsKRtY3kxce8dQkfk1iDjLcc3q-01r4rpxSx2xZSbwg_
KkAN3kCRQ9uLfE0ziHAcpvunuKmzGBWKnBhC4sJJkXrMEfXwCg4nsOjg&response_mode=form_
post&nonce=637877163655610380.MjNjZmM4NzQtOTU5My00OGZlLTk0NTItZTE5NDU2YjVlODdjNjViOTQwYmUtOTZl
MS00M2Y5LTkyN2MtN2QyMjgwNjcxY2Uz&x-client-SKU=ID_NETSTANDARD2_0&x-client-
Ver=6.12.1.0&uaid=5c3d196d593945cca45b232b9ed13fce&msproxy=1&issuer=mso&tenant=common&ui_
locales=en-US&epct=AQABAAAAAAD--DLA3VO7QrddgJg7WevrfA6SLaDsJUcjb1Bg9OKonF3d_
lfNJsdDAIH5hlJdUSGejEBIqsko-A7JX67PzaGdEJgOIGa37VhJzGTYBZ-KgATe9FHssnNmLjM_
dojr0dAT83xDhiqQTN2-UcYdcP2s3vPainF7Nqes5ecXRaEoE9Vw9-
sN7jfASOkPRWW03aI6buz0niABvA860YOWDb98vdJWPGkWE-euDr6n8_
zI5iAA&jshs=0&username=***************%40outlook.com&login_hint=***************%40outlook.com
HTTP/1.1
Host: login.live.com
```

```
Connection: keep-alive
...
Referer: https://login.microsoftonline.com/
Accept-Encoding: gzip, deflate, br
Accept-Language: en-US,en;q=0.9
sec-Restrict-Tenant-Access-Policy: restrict-msa
```

Explicit proxy and FortiGate Cloud Sandbox

Explicit proxy connections can leverage FortiGate Cloud Sandbox for advanced threat scanning and updates. This allows FortiGates behind isolated networks to connect to FortiCloud services.



To configure FortiGuard services to communicate with an explicit proxy server:

```
config system fortiguard
    set proxy-server-ip 172.16.200.44
    set proxy-server-port 3128
    set proxy-username "test1"
    set proxy-password *********
end
```

To verify the explicit proxy connection to FortiGate Cloud Sandbox:

```
# diagnose debug application forticldd -1
Debug messages will be on for 30 minutes.
# diagnose debug enable
[2942] fds_handle_request: Received cmd 23 from pid-2526, len 0
[40] fds_queue_task: req-23 is added to Cloud-sandbox-controller
[178] fds_svr_default_task_xmit: try to get IPs for Cloud-sandbox-controller
[239] fds_resolv_addr: resolve aptctrl1.fortinet.com
[169] fds_get_addr: name=aptctrl1.fortinet.com, id=32, cb=0x2bc089
[101] dns_parse_resp: DNS aptctrl1.fortinet.com -&gt; 172.16.102.21
[227] fds_resolv_cb: IP-1: 172.16.102.21
[665] fds_ctx_set_addr: server: 172.16.102.21:443
[129] fds_svr_default_pickup_server: Cloud-sandbox-controller: 172.16.102.21:443
[587] fds_https_start_server: server: 172.16.102.21:443
[579] ssl_new: SSL object is created
[117] https_create: proxy server 172.16.200.44 port:3128
[519] fds_https_connect: https_connect(172.16.102.21) is established.
[261] fds_svr_default_on_established: Cloud-sandbox-controller has connected to ip=172.16.102.21
[268] fds_svr_default_on_established: server-Cloud-sandbox-controller handles cmd-23
```

```
[102] fds_pack_objects: number of objects: 1
[75] fds_print_msg: FCPC: len=109
[81] fds_print_msg: Protocol=2.0
[81] fds_print_msg: Command=RegionList
[81] fds_print_msg: Firmware=FG101E-FW-6.02-0917
[81] fds_print_msg: SerialNumber=FG101E4Q17002429
[81] fds_print_msg: TimeZone=-7
[75] fds_print_msg: http req: len=248
[81] fds_print_msg: POST https://172.16.102.21:443/FCPService HTTP/1.1
[81] fds_print_msg: User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
[81] fds_print_msg: Host: 172.16.102.21:443
[81] fds_print_msg: Cache-Control: no-cache
[81] fds_print_msg: Connection: close
[81] fds_print_msg: Content-Type: application/octet-stream
[81] fds_print_msg: Content-Length: 301
[524] fds_https_connect: http request to 172.16.102.21: header=248, ext=301.
[257] fds_https_send: sent 248 bytes: pos=0, len=248
[265] fds_https_send: 172.16.102.21: sent 248 byte header, now send 301-byte body
[257] fds_https_send: sent 301 bytes: pos=0, len=301
[273] fds_https_send: sent the entire request to server: 172.16.102.21:443
[309] fds_https_recv: read 413 bytes: pos=413, buf_len=2048
[332] fds_https_recv: received the header from server: 172.16.102.21:443, [HTTP/1.1 200
Content-Type: application/octet-stream
Content-Length: 279
Date: Thu, 20 Jun 2019 16:41:11 GMT
Connection: close]
[396] fds_https_recv: Do memmove buf_len=279, pos=279
[406] fds_https_recv: server: 172.16.102.21:443, buf_len=279, pos=279
[453] fds_https_recv: received a packet from server-172.16.102.21:443: sz=279, objs=1
[194] __ssl_data_ctx_free: Done
[839] ssl_free: Done
[830] ssl_disconnect: Shutdown
[481] fds_https_recv: obj-0: type=FCPR, len=87
[294] fds_svr_default_on_response: server-Cloud-sandbox-controller handles cmd-23
[75] fds_print_msg: fcpr:  len=83
[81] fds_print_msg: Protocol=2.0
[81] fds_print_msg: Response=202
[81] fds_print_msg: ResponseItem=Region:Europe,Global,Japan,US
[81] fds_print_msg: existing:Japan
[3220] aptctrl_region_res: Got rsp: Region:Europe,Global,Japan,US
[3222] aptctrl_region_res: Got rsp: Region existing:Japan
[439] fds_send_reply: Sending 28 bytes data.
[395] fds_free_tsk: cmd=23; req.noreply=1
# [136] fds_on_sys_fds_change: trace
[2942] fds_handle_request: Received cmd 22 from pid-170, len 0
[40] fds_queue_task: req-22 is added to Cloud-sandbox-controller
[587] fds_https_start_server: server: 172.16.102.21:443
[579] ssl_new: SSL object is created
[117] https_create: proxy server 172.16.200.44 port:3128
[519] fds_https_connect: https_connect(172.16.102.21) is established.
[261] fds_svr_default_on_established: Cloud-sandbox-controller has connected to ip=172.16.102.21
[268] fds_svr_default_on_established: server-Cloud-sandbox-controller handles cmd-22
```

```
[102] fds_pack_objects: number of objects: 1
[75] fds_print_msg: FCPC: len=146
[81] fds_print_msg: Protocol=2.0
[81] fds_print_msg: Command=UpdateAPT
[81] fds_print_msg: Firmware=FG101E-FW-6.02-0917
[81] fds_print_msg: SerialNumber=FG101E4Q17002429
[81] fds_print_msg: TimeZone=-7
[81] fds_print_msg: TimeZoneInMin=-420
[81] fds_print_msg: DataItem=Region:US
[75] fds_print_msg: http req: len=248
[81] fds_print_msg: POST https://172.16.102.21:443/FCPService HTTP/1.1
[81] fds_print_msg: User-Agent: Mozilla/4.0 (compatible; MSIE 6.0; Windows NT 5.1)
[81] fds_print_msg: Host: 172.16.102.21:443
[81] fds_print_msg: Cache-Control: no-cache
[81] fds_print_msg: Connection: close
[81] fds_print_msg: Content-Type: application/octet-stream
[81] fds_print_msg: Content-Length: 338
[524] fds_https_connect: http request to 172.16.102.21: header=248, ext=338.
[257] fds_https_send: sent 248 bytes: pos=0, len=248
[265] fds_https_send: 172.16.102.21: sent 248 byte header, now send 338-byte body
[257] fds_https_send: sent 338 bytes: pos=0, len=338
[273] fds_https_send: sent the entire request to server: 172.16.102.21:443
[309] fds_https_recv: read 456 bytes: pos=456, buf_len=2048
[332] fds_https_recv: received the header from server: 172.16.102.21:443, [HTTP/1.1 200
Content-Type: application/octet-stream
Content-Length: 322
Date: Thu, 20 Jun 2019 16:41:16 GMT
Connection: close]
[396] fds_https_recv: Do memmove buf_len=322, pos=322
[406] fds_https_recv: server: 172.16.102.21:443, buf_len=322, pos=322
[453] fds_https_recv: received a packet from server-172.16.102.21:443: sz=322, objs=1
[194] __ssl_data_ctx_free: Done
[839] ssl_free: Done
[830] ssl_disconnect: Shutdown
[481] fds_https_recv: obj-0: type=FCPR, len=130
[294] fds_svr_default_on_response: server-Cloud-sandbox-controller handles cmd-22
[75] fds_print_msg: fcpr:  len=126
[81] fds_print_msg: Protocol=2.0
[81] fds_print_msg: Response=202
[81] fds_print_msg: ResponseItem=Server1:172.16.102.51:514
[81] fds_print_msg: Server2:172.16.102.52:514
[81] fds_print_msg: Contract:20210215
[81] fds_print_msg: NextRequest:86400
[615] parse_apt_contract_time_str: The APTContract is valid to Mon Feb 15 23:59:59 2021
[616] parse_apt_contract_time_str: FGT current local time is Thu Jun 20 09:41:16 2019
[3289] aptctrl_update_res: Got rsp: APT=172.16.102.51:514 APTAlter=172.16.102.52:514 next-
upd=86400
[395] fds_free_tsk: cmd=22; req.noreply=1
```

Proxy chaining

For the explicit web proxy you can configure web proxy forwarding servers to use proxy chaining to redirect web proxy sessions to other proxy servers. Proxy chaining can be used to forward web proxy sessions from the FortiGate unit to one or more other proxy servers on your network or on a remote network. You can use proxy chaining to integrate the FortiGate explicit web proxy with a web proxy solution that you already have in place.

A FortiGate unit can forward sessions to most web proxy servers including a remote FortiGate unit with the explicit web proxy enabled. No special configuration of the explicit web proxy on the remote FortiGate unit is required.

You can deploy the explicit web proxy with proxy chaining in an enterprise environment consisting of small satellite offices and a main office. If each office has a FortiGate unit, users at each of the satellite offices can use their local FortiGate unit as an explicit web proxy server. The satellite office FortiGate units can forward explicit web proxy sessions to an explicit web proxy server at the central office. From here the sessions can connect to web servers on the Internet.

FortiGate proxy chaining does not support web proxies in the proxy chain authenticating each other.

The following examples assume explicit web proxy has been enabled.

To enable explicit web proxy in the GUI:

1. Go to *System* > *Feature Visibility*.
2. In the *Security Features* column, enable *Explicit Proxy*.
3. Configure the explicit web proxy settings. See Explicit web proxy on page 406.

To add a web proxy forwarding server in the GUI:

1. Go to *Network* > *Explicit Proxy*. The *Explicit Proxy* page opens.
2. In the *Web Proxy Forwarding Servers* section, click *Create New*.
3. Configure the server settings:

Name	Enter the name of the forwarding server.
Proxy Address Type	Select the type of IP address of the forwarding server. A forwarding server can have an *FQDN* or *IP* address.
Proxy Address	Enter the IP address of the forwarding server.
Port	Enter the port number on which the proxy receives connections. Traffic leaving the FortiGate explicit web proxy for this server has its destination port number changed to this number.
Server Down Action	Select the action the explicit web proxy will take if the forwarding server is down. • *Block*: Blocks the traffic if the remote server is down. • *Use Original Server*: Forwards the traffic from the FortiGate to its destination as if no forwarding server is configured.
Health Monitor	Select to enable health check monitoring.
Health Check Monitor Site	Enter the address of a remote site.

4. Click *OK*.

Example

The following example adds a web proxy forwarding server named `fwd-srv` at address `proxy.example.com` and port `8080`.

To add a web proxy forwarding server in the CLI:

```
config web-proxy forward-server
    edit fwd-srv
        set addr-type fqdn
        set fqdn proxy.example.com
        set port 8080
    next
end
```

Web proxy forwarding server monitoring and health checking

By default, a FortiGate unit monitors a web proxy forwarding server by forwarding a connection to the remote server every 10 seconds. The remote server is assumed to be down if it does not respond to the connection. FortiGate continues checking the server. The server is assumed to be back up when the server sends a response. If you enable health checking, the FortiGate unit attempts to get a response from a web server every 10 seconds by connecting through the remote forwarding server.

You can configure health checking for each remote server and specify a different website to check for each one.

If the remote server is found to be down you can configure the FortiGate unit to block sessions until the server comes back up or to allow sessions to connect to their destination, bypassing the remote forwarding server. You cannot configure the FortiGate unit to fail over to another remote forwarding server.

To configure proxy server monitor and health checking in the GUI:

1. Go to *Network* > *Explicit Proxy*. The *Explicit Proxy* page opens.
2. In the *Web Proxy Forwarding Servers* section, edit a server.
3. Configure the *Server Down Action* and *Health Monitor* settings.

Server Down Action	Select the action the explicit web proxy will take if the forwarding server is down. • *Block*: Blocks the traffic if the remote server is down. • *Use Original Server*: Forwards the traffic from the FortiGate to its destination as if no forwarding server configured.
Health Monitor	Select to enable health check monitoring.
Health Check Monitor Site	Enter the address of a remote site.

4. Click *OK*.

Example

The following example enables health checking for a web proxy forwarding server and sets the server down option to bypass the forwarding server if it is down.

To configure proxy server monitor and health checking in the CLI:

```
config web-proxy forward-server
    edit fwd-srv
        set healthcheck enable
        set monitor http://example.com
        set server-down-option pass
    next
end
```

Grouping forwarding servers and load balancing traffic to the servers

You can add multiple web proxy forwarding servers to a forwarding server group and then add the server group to an explicit web proxy policy instead of adding a single server. Forwarding server groups are created from the FortiGate CLI but can be added to policies from the web-based manager (or from the CLI).

When you create a forwarding server group you can select a load balancing method to control how sessions are load balanced to the forwarding servers in the server group. Two load balancing methods are available:

- *Weighted* load balancing sends more sessions to the servers with higher weights. You can configure the weight for each server when you add it to the group.
- *Least-session* load balancing sends new sessions to the forwarding server that is processing the fewest sessions.

When you create a forwarding server group you can also enable *affinity*. Enable affinity to have requests from the same client processed by the same server. This can reduce delays caused by using multiple servers for a single multi-step client operation. Affinity takes precedence over load balancing.

You can also configure the behavior of the group if all of the servers in the group are down. You can select to block traffic or you can select to have the traffic pass through the FortiGate explicit proxy directly to its destination instead of being sent to one of the forwarding servers.

Example

The following example adds a forwarding server group that uses weighted load balancing to load balance traffic to three forwarding servers. Server weights are configured to send most traffic to `server2`. The group has `affinity` enabled and blocks traffic if all of the forward servers are down.

To configure load balancing in the CLI:

```
config web-proxy forward-server
    edit server_1
        set ip 172.20.120.12
```

```
        set port 8080
    next
    edit server_2
        set ip 172.20.120.13
        set port 8000
    next
    edit server_3
        set ip 172.20.120.14
        set port 8090
    next
end
```

```
config web-proxy forward-server-group
    edit New-fwd-group
        set affinity enable
        set ldb-method weighted
        set group-down-option block
        config server-list
            edit server_1
                set weight 10
            next
            edit server_2
                set weight 40
            next
            edit server_3
                set weight 10
            next
        end
    next
end
```

Adding proxy chaining to an explicit web proxy policy

You can enable proxy chaining for web proxy sessions by adding a web proxy forwarding server or server group to an explicit web proxy policy. In a policy you can select one web proxy forwarding server or server group. All explicit web proxy traffic accepted by this security policy is forwarded to the specified web proxy forwarding server or server group.

To add an explicit web proxy forwarding server in the GUI:

1. Go to *Policy & Objects* > *Proxy Policy* and click *Create New*.
2. Configure the policy settings:

Proxy Type	Explicit Web
Outgoing Interface	wan1
Source	Internal_subnet
Destination	all

Schedule	always
Service	webproxy
Action	Accept

3. Enable *Web Proxy Forwarding Server* and select the forwarding server, (for example,*fwd-srv*).
4. Click *OK*.

Example

The following example adds a security policy that allows all users on the `10.31.101.0` subnet to use the explicit web proxy for connections through the `wan1` interface to the Internet. The policy forwards web proxy sessions to a remote forwarding server named *fwd-srv*.

To add an explicit web proxy forwarding server in the CLI:

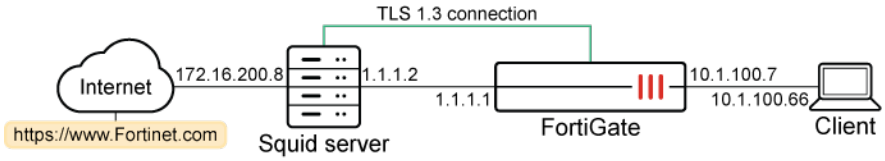
```
config firewall proxy-policy
    edit 0
        set proxy explicit-web
        set dstintf "wan1"
        set srcaddr "Internal_subnet"
        set dstaddr "all"
        set service "webproxy"
        set action accept
        set schedule "always"
        set webproxy-forward-server "fwd-srv"
    next
end
```

Using TLS 1.3 with web proxy forward servers

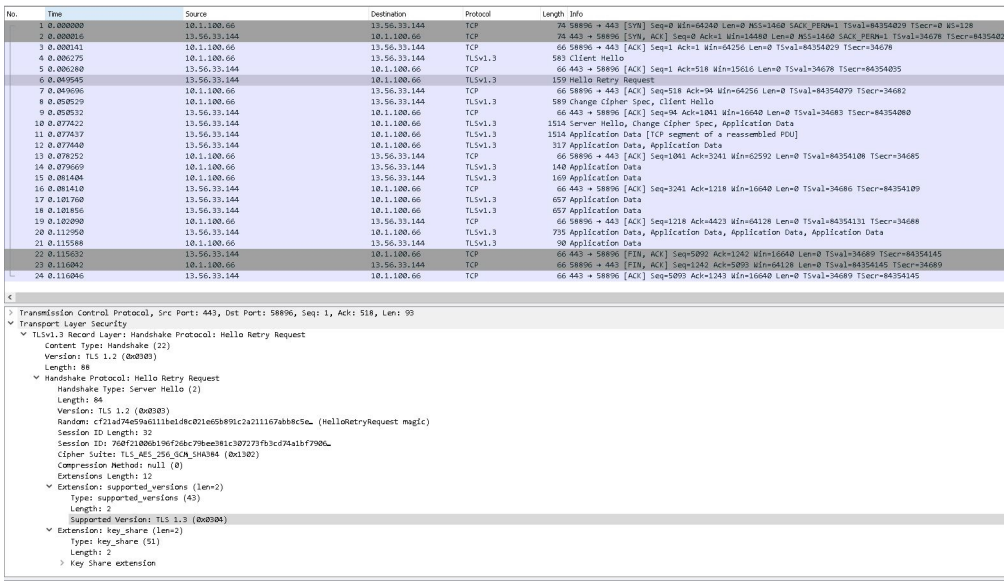
A FortiGate can handle TLS 1.3 traffic in both deep and certificate inspection modes.

Example

The following example demonstrates that the Squid server and the FortiGate can handle TLS 1.3 traffic.



The following output from the Squid server demonstrates that the FortiGate supports TLS 1.3 traffic and forwards the hello retry request back to the client PC. The client PC then sends the client hello again, and the connection is successfully established.



WAN optimization SSL proxy chaining

An SSL server does not need to be defined for WAN optimization (WANOpt) SSL traffic offloading (traffic acceleration). The server side FortiGate uses an SSL profile to resign the HTTP server's certificate, both with and without an external proxy, without an SSL server configured. GCM and ChaCha ciphers can also be used in the SSL connection.

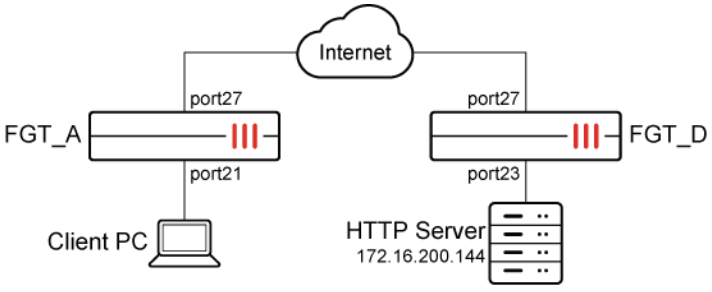
Examples

In these examples, HTTPS traffic is accelerated without configuring an SSL server, including with a proxy in between, and when the GCM or ChaCha ciphers are used.

Example 1

In this example, the server certificate is resigned by the server side FortiGate, and HTTPS traffic is accelerated without configuring an SSL server.

HTTPS traffic with the GCM or ChaCha cipher can pass though WANOpt tunnel.



To configure FGT_A:

1. Configure the hard disk to perform WANOpt:

```
config system storage
    edit "HDD2"
        set status enable
        set usage wanopt
        set wanopt-mode mix
    next
end
```

2. Configure the WANOpt peer and profile:

```
config wanopt peer
    edit "FGT-D"
        set ip 120.120.120.172
    next
end
```

```
config wanopt profile
    edit "test"
        config http
            set status enable
            set ssl enable
        end
    next
end
```

3. Create an SSL profile with deep inspection on HTTPS port 443:

```
config firewall ssl-ssh-profile
    edit "ssl"
        config https
            set ports 443
            set status deep-inspection
        end
    next
end
```

4. Configure a firewall policy in proxy mode with WANOpt enabled and the WANOpt profile selected:

```
config firewall policy
    edit 1
        set name "WANOPT-A"
        set srcintf "port21"
        set dstintf "port27"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
```

```
        set inspection-mode proxy
        set profile-protocol-options "protocol"
        set ssl-ssh-profile "ssl"
        set wanopt enable
        set wanopt-profile "test"
        set nat enable
    next
end
```

To configure FGT_D:

1. Configure the hard disk to perform WANOpt:

```
config system storage
    edit "HDD2"
        set status enable
        set usage wanopt
        set wanopt-mode mix
    next
end
```

2. Configure the WANOpt peer:

```
config wanopt peer
    edit "FGT-A"
        set ip 110.110.110.171
    next
end
```

3. Create an SSL profile with deep inspection on HTTPS port 443. The default *Fortinet_CA_SSL* certificate is used to resign the server certificate:

```
config firewall ssl-ssh-profile
    edit "ssl"
        config https
            set ports 443
            set status deep-inspection
        end
    next
end
```

4. Configure a firewall policy in proxy mode with WANOpt enabled and passive WANOpt detection:

```
config firewall policy
    edit 1
        set name "WANOPT-B"
        set srcintf "port27"
        set dstintf "port23"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
```

```
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set wanopt enable
        set wanopt-detection passive
        set nat enable
    next
end
```

5. Configure a proxy policy to apply the SSL profile:

```
config firewall proxy-policy
    edit 100
        set proxy wanopt
        set dstintf "port23"
        set srcaddr "all"
        set dstaddr "all"
        set service "ALL"
        set action accept
        set schedule "always"
        set utm-status enable
        set profile-protocol-options "protocol"
        set ssl-ssh-profile "ssl"
    next
end
```

To confirm that traffic is accelerated:

1. On the client PC, curl a 10MB test sample for the first time:

```
root@client:/tmp# curl -k https://172.16.200.144/test_10M.pdf -O
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 9865k  100 9865k    0     0   663k      0  0:00:14  0:00:15 --:--:-- 1526k
```

   It takes 15 seconds to finish the download.

2. On FGT_A, check the WAD statistics:

```
# diagnose wad stats worker.tunnel
comp.n_in_raw_bytes                  10155840
comp.n_in_comp_bytes                 4548728
comp.n_out_raw_bytes                 29624
comp.n_out_comp_bytes                31623
```

```
# diagnose wad stats worker.protos.http
wan.bytes_in                         0
wan.bytes_out                        0
lan.bytes_in                         760
lan.bytes_out                        10140606
tunnel.bytes_in                      4548728
tunnel.bytes_out                     31623
```

3. Curl the same test sample a second time:

```
root@client:/tmp# curl -k https://172.16.200.144/test_10M.pdf -O
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 9865k  100 9865k    0     0   663k      0  0:00:01  0:00:01 --:--:-- 1526k
```

It now takes less than one second to finish the download.

4. On FGT_A, check the WAD statistics again:

```
# diagnose wad stats worker.tunnel
comp.n_in_raw_bytes                  10181157
comp.n_in_comp_bytes                 4570331
comp.n_out_raw_bytes                 31627
comp.n_out_comp_bytes                34702
```

```
# diagnose wad stats worker.protos.http
wan.bytes_in                         0
wan.bytes_out                        0
lan.bytes_in                         1607
lan.bytes_out                        20286841
tunnel.bytes_in                      4570331
tunnel.bytes_out                     34702
```

The tunnel bytes are mostly unchanged, but the LAN bytes are doubled. This means that the bytes of the second curl come from the cache, showing that the traffic is accelerated.

To confirm that a curl using the GCM cipher is accepted and accelerated:

1. On the client PC, curl a 10MB test sample with the GCM cipher:

```
root@client:/tmp# curl -v -k --ciphers DHE-RSA-AES128-GCM-SHA256 https://172.16.200.144/test_
10M.pdf -O
*   Trying 172.16.200.144...
* TCP_NODELAY set
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0* Connected to
172.16.200.144 (172.16.200.144) port 443 (#0)
* ALPN, offering h2
* ALPN, offering http/1.1
* Cipher selection: DHE-RSA-AES128-GCM-SHA256
* successfully set certificate verify locations:
*   CAfile: /etc/ssl/certs/ca-certificates.crt
  CApath: none
} [5 bytes data]
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
} [512 bytes data]
* TLSv1.3 (IN), TLS handshake, Server hello (2):
{ [100 bytes data]
* TLSv1.2 (IN), TLS handshake, Certificate (11):
{ [1920 bytes data]
```

```
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
{ [783 bytes data]
* TLSv1.2 (IN), TLS handshake, Server finished (14):
{ [4 bytes data]
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
} [262 bytes data]
* TLSv1.2 (OUT), TLS change cipher, Change cipher spec (1):
} [1 bytes data]
* TLSv1.2 (OUT), TLS handshake, Finished (20):
} [16 bytes data]
* TLSv1.2 (IN), TLS handshake, Finished (20):
{ [16 bytes data]
* SSL connection using TLSv1.2 / DHE-RSA-AES128-GCM-SHA256
* ALPN, server accepted to use http/1.1
* Server certificate:
*  subject: CN=ubuntu
*  start date: Sep 20 21:38:01 2018 GMT
*  expire date: Sep 17 21:38:01 2028 GMT
*  issuer: C=US; ST=California; L=Sunnyvale; O=Fortinet; OU=Certificate Authority; CN=Fortinet
Untrusted CA; emailAddress=support@fortinet.com
*  SSL certificate verify result: self signed certificate in certificate chain (19),
continuing anyway.
} [5 bytes data]
> GET /test_10M.pdf HTTP/1.1
> Host: 172.16.200.144
> User-Agent: curl/7.64.1
> Accept: */*
>
{ [5 bytes data]
< HTTP/1.1 200 OK
< Date: Sat, 12 Jun 2021 00:31:08 GMT
< Server: Apache/2.4.37 (Ubuntu)
< Upgrade: h2,h2c
< Connection: Upgrade
< Last-Modified: Fri, 29 Jan 2021 20:10:25 GMT
< ETag: "9a2572-5ba0f98404aa5"
< Accept-Ranges: bytes
< Content-Length: 10102130
< Content-Type: application/pdf
<
{ [5 bytes data]
100 9865k  100 9865k    0     0  16.7M      0 --:--:-- --:--:-- --:--:-- 16.8M
* Connection #0 to host 172.16.200.144 left intact
* Closing connection 0
```

To confirm that a curl using the ChaCha cipher is accepted and accelerated:

1. On the client PC, curl a 10MB test sample with the ChaCha cipher:

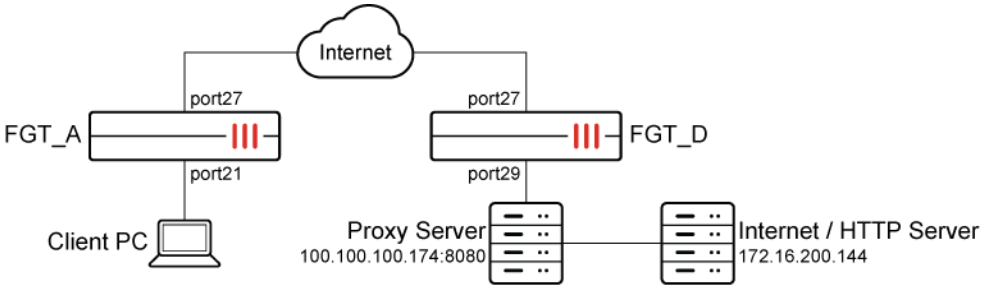
```
root@client:/tmp# curl -v -k --ciphers ECDHE-RSA-CHACHA20-POLY1305
https://172.16.200.144/test.doc -O
*   Trying 172.16.200.144...
```

```
* TCP_NODELAY set
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0* Connected to
172.16.200.144 (172.16.200.144) port 443 (#0)
* ALPN, offering h2
* ALPN, offering http/1.1
* Cipher selection: ECDHE-RSA-CHACHA20-POLY1305
* successfully set certificate verify locations:
*   CAfile: /etc/ssl/certs/ca-certificates.crt
  CApath: none
} [5 bytes data]
* TLSv1.3 (OUT), TLS handshake, Client hello (1):
} [512 bytes data]
* TLSv1.3 (IN), TLS handshake, Server hello (2):
{ [100 bytes data]
* TLSv1.2 (IN), TLS handshake, Certificate (11):
{ [1920 bytes data]
* TLSv1.2 (IN), TLS handshake, Server key exchange (12):
{ [300 bytes data]
* TLSv1.2 (IN), TLS handshake, Server finished (14):
{ [4 bytes data]
* TLSv1.2 (OUT), TLS handshake, Client key exchange (16):
} [37 bytes data]
* TLSv1.2 (OUT), TLS change cipher, Change cipher spec (1):
} [1 bytes data]
* TLSv1.2 (OUT), TLS handshake, Finished (20):
} [16 bytes data]
* TLSv1.2 (IN), TLS handshake, Finished (20):
{ [16 bytes data]
* SSL connection using TLSv1.2 / ECDHE-RSA-CHACHA20-POLY1305
* ALPN, server accepted to use http/1.1
* Server certificate:
*  subject: CN=ubuntu
*  start date: Sep 20 21:38:01 2018 GMT
*  expire date: Sep 17 21:38:01 2028 GMT
*  issuer: C=US; ST=California; L=Sunnyvale; O=Fortinet; OU=Certificate Authority; CN=Fortinet
Untrusted CA; emailAddress=support@fortinet.com
*  SSL certificate verify result: self signed certificate in certificate chain (19),
continuing anyway.
} [5 bytes data]
> GET /test.doc HTTP/1.1
> Host: 172.16.200.144
> User-Agent: curl/7.64.1
> Accept: */*
>
{ [5 bytes data]
< HTTP/1.1 200 OK
< Date: Sat, 12 Jun 2021 00:32:11 GMT
< Server: Apache/2.4.37 (Ubuntu)
< Upgrade: h2,h2c
```

```
< Connection: Upgrade
< Last-Modified: Wed, 05 May 2021 21:59:49 GMT
< ETag: "4c00-5c19c504b63f4"
< Accept-Ranges: bytes
< Content-Length: 19456
< Content-Type: application/msword
<
{ [5 bytes data]
100 19456  100 19456    0     0   137k      0 --:--:-- --:--:-- --:--:--  138k
* Connection #0 to host 172.16.200.144 left intact
* Closing connection 0
```

Example 2

In this example, an external proxy is added to the configuration in Example 1.



To reconfigure FGT_A:

```
config firewall profile-protocol-options
    edit "protocol"
        config http
            set ports 80 8080
            unset options
            unset post-lang
        end
    next
end
```

To reconfigure FGT_D:

1. Configure a new firewall policy for traffic passing from port27 to port29:

```
config firewall policy
    edit 1
        set name "WANOPT-B"
        set srcintf "port27"
        set dstintf "port29"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
```

```
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set wanopt enable
        set wanopt-detection passive
        set nat enable
    next
end
```

2. Configure a proxy policy for traffic on destination interface port29:

```
config firewall proxy-policy
    edit 100
        set proxy wanopt
        set dstintf "port29"
        set srcaddr "all"
        set dstaddr "all"
        set service "ALL"
        set action accept
        set schedule "always"
        set profile-protocol-options "protocol"
        set ssl-ssh-profile "ssl"
    next
end
```

To confirm that HTTPS traffic is still being accelerated:

1. On the client PC, curl the same 10MB test sample through the explicit proxy:

```
root@client:/tmp# curl -x 100.100.100.174:8080 -v -k https://172.16.200.144/test_10M.pdf -O
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 9865k  100 9865k    0     0   663k      0  0:00:01  0:00:01 --:--:-- 1526k
```

   It takes less than a second to finish the download.

Agentless NTLM authentication for web proxy

Agentless Windows NT LAN Manager (NTLM) authentication includes support for the following items:

- Multiple servers
- Individual users

You can use multiple domain controller servers for the agentless NTLM. They can be used for load balancing and high service stability.

You can also use user-based matching in groups for Kerberos and agentless NTLM. In these scenarios, FortiOS matches the user's group information from an LDAP server.

To support multiple domain controllers for agentless NTLM using the CLI:

1. Configure an LDAP server:

```
config user ldap
    edit "ldap-kerberos"
        set server "172.18.62.177"
        set cnid "cn"
        set dn "dc=fortinetqa,dc=local"
        set type regular
        set username "CN=root,CN=Users,DC=fortinetqa,DC=local"
        set password *********
    next
end
```

2. Configure multiple domain controllers:

```
config user domain-controller
    edit "dc1"
        set ip-address 172.18.62.177
        config extra-server
            edit 1
                set ip-address 172.18.62.220
            next
        end
        set ldap-server "ldap-kerberos"
    next
end
```

3. Create an authentication scheme and rule:

```
config authentication scheme
    edit "au-ntlm"
        set method ntlm
        set domain-controller "dc1"
    next
end
```

```
config authentication rule
    edit "ru-ntlm"
        set srcaddr "all"
        set ip-based disable
        set active-auth-method "au-ntlm"
    next
end
```

4. In the proxy policy, append the user group for authorization:

```
config firewall proxy-policy
    edit 1
        set proxy explicit-web
        set dstintf "port1"
        set srcaddr "all"
```

```
        set dstaddr "all"
        set service "web"
        set action accept
        set schedule "always"
        set groups "ldap-group"
        set utm-status enable
        set av-profile "av"
        set ssl-ssh-profile "deep-custom"
    next
end
```

This configuration uses a round-robin method. When the first user logs in, the FortiGate sends the authentication request to the first domain controller. Later when another user logs in, the FortiGate sends the authentication request to another domain controller.

5. Verify the behavior after the user successfully logs in:

```
# diagnose wad user list
ID: 1825, IP: 10.1.100.71, VDOM: vdom1
    user name   : test1
    duration    : 497
    auth_type   : Session
    auth_method : NTLM
    pol_id      : 1   g_id        : 5
    user_based  : 0   e
    xpire      : 103
    LAN:
        bytes_in=2167 bytes_out=7657
    WAN:
        bytes_in=3718 bytes_out=270
```

To support individual users for agentless NTLM using the CLI:

1. Configure an LDAP server:

```
config user ldap
    edit "ldap-kerberos"
        set server "172.18.62.177"
        set cnid "cn"
        set dn "dc=fortinetqa,dc=local"
        set type regular
        set username "CN=root,CN=Users,DC=fortinetqa,DC=local"
        set password *********
    next
end
```

2. Configure the user group and allow user-based matching:

```
config user group
    edit "ldap-group"
        set member "ldap" "ldap-kerberos"
        config match
            edit 1
```

```
                set server-name "ldap-kerberos"
                set group-name "test1"
            next
        end
    next
end
```

3. Create an authentication scheme and rule:

```
config authentication scheme
    edit "au-ntlm"
        set method ntlm
        set domain-controller "dc1"
    next
end
```

```
config authentication rule
    edit "ru-ntlm"
        set srcaddr "all"
        set ip-based disable
        set active-auth-method "au-ntlm"
    next
end
```

4. In the proxy policy, append the user group for authorization:

```
config firewall proxy-policy
    edit 1
        set proxy explicit-web
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set service "web"
        set action accept
        set schedule "always"
        set groups "ldap-group"
        set utm-status enable
        set av-profile "av"
        set ssl-ssh-profile "deep-custom"
    next
end
```

This implementation lets you configure a single user instead of a whole group. The FortiGate will now allow the user named `test1`.

To verify the configuration using the CLI:

```
diagnose wad user list
    ID: 1827, IP: 10.1.15.25, VDOM: vdom1
    user name   : test1
    duration    : 161
```

```
    auth_type   : Session
    auth_method : NTLM
    pol_id      : 1
    g_id        : 5
    user_based  : 0
    expire      : 439
    LAN:
           bytes_in=1309 bytes_out=4410
      WAN:
           bytes_in=2145 bytes_out=544
```

Multiple LDAP servers in Kerberos keytabs and agentless NTLM domain controllers

Multiple LDAP servers can be configured in Kerberos keytabs and agentless NTLM domain controllers for multi-forest deployments.

To use multiple LDAP servers in Kerberos keytabs and agentless NTLM domain controllers:

1. Add multiple LDAP servers:

```
config user ldap
    edit "ldap-kerberos"
        set server "172.16.200.98"
        set cnid "cn"
        set dn "dc=fortinetqa,dc=local"
        set type regular
        set username "CN=root,CN=Users,DC=fortinetqa,DC=local"
        set password xxxxxxxxx
    next
    edit "ldap-two"
        set server "172.16.106.128"
        set cnid "cn"
        set dn "OU=Testing,DC=ad864r2,DC=com"
        set type regular
        set username "cn=Testadmin,cn=users,dc=AD864R2,dc=com"
        set password xxxxxxxxx
    next
end
```

2. Configure a Kerberos keytab entry that uses both LDAP servers:

```
config user krb-keytab
    edit "http_service"
        set pac-data disable
        set principal "HTTP/FGT.FORTINETQA.LOCAL@FORTINETQA.LOCAL"
        set ldap-server "ldap-kerberos" "ldap-two"
        set keytab xxxxxxxxx
```

```
    next
end
```

3. Configure a domain controller that uses both LDAP servers:

```
config user domain-controller
    edit "dc1"
        set ip-address 172.16.200.98
        set ldap-server "ldap-two" "ldap-kerberos"
    next
end
```

Learn client IP addresses

Learning the actual client IP addresses is imperative for authorization. This function identifies the real client IP address when there is a NATing device between the FortiGate and the client.

```
config web-proxy global
    set learn-client-ip {enable | disable}
    set learn-client-ip-from-header {true-client-ip | x-real-ip | x-forwarded-for}
    set learn-client-ip-srcaddr <address> ... <address>
end
```

learn-client-ip {enable | disable}	Enable/disable learning the client's IP address from headers.
learn-client-ip-from-header {true-client-ip | x-real-ip | x-forwarded-for}	Learn client IP addresses from the specified headers.
learn-client-ip-srcaddr <address> ... <address>	The source address names.

Example

In this example, the real client IP address is used to match a policy for FSSO authentication.

To enable learning the client IP address:

```
config web-proxy global
    set proxy-fqdn "default.fqdn"
    set webproxy-profile "default"
    set learn-client-ip enable
       set learn-client-ip-from-header x-forwarded-for
    set learn-client-ip-srcaddr "all"
end
```

To configure the proxy policy:

```
config firewall proxy-policy
    edit 1
        set proxy explicit-web
        set dstintf "mgmt1"
        set srcaddr "all"
        set dstaddr "all"
        set service "w"
        set action accept
        set schedule "always"
        set groups "fsso1"
        set utm-status enable
        set av-profile "default"
        set dlp-profile "default"
        set profile-protocol-options "default"
        set ssl-ssh-profile "deep-inspection"
    next
end
```

To configure the authentication scheme and rule:

```
config authentication scheme
    edit "scheme1"
        set method fsso
    next
end
```

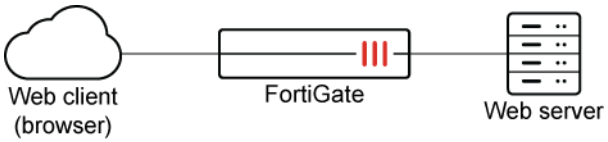
```
config authentication rule
    edit "rule1"
        set srcaddr "all"
        set sso-auth-method "scheme1"
    next
end
```

Explicit proxy authentication over HTTPS

When a HTTP request requires authentication in an explicit proxy, the authentication can be redirected to a secure HTTPS captive portal. Once authentication is complete, the client can be redirected back to the original destination over HTTP.

Example

A user visits a website via HTTP through the explicit web proxy on a FortiGate. The user is required to authenticate by either basic or form IP-based authentication for the explicit web proxy service. The user credentials need to be transmitted over the networks in a secured method over HTTPS rather than in plain text. The user credentials are protected by redirecting the client to a captive portal of the FortiGate over HTTPS for authentication where the user credentials are encrypted and transmitted over HTTPS.



In this example, explicit proxy authentication over HTTPS is configured with form IP-based authentication. Once configured, you can enable authorization for an explicit web proxy by configuring users or groups in the firewall proxy policy.

To configure explicit proxy authentication over HTTPS:

1. Configure the authentication settings:

```
config authentication setting
    set captive-portal-type fqdn
    set captive-portal "fgt-cp"
    set auth-https enable
end
```

2. Configure the authentication scheme:

```
config authentication scheme
    edit "form"
        set method form
        set user-database "local-user-db"
    next
end
```

3. Configure the authentication rule:

```
config authentication rule
    edit "form"
        set srcaddr "all"
        set active-auth-method "form"
    next
end
```

If a session-based basic authentication method is used, enable `web-auth-cookie`.

4. Configure the firewall address:

```
config firewall address
    edit "fgt-cp"
        set type fqdn
        set fqdn "fgt.fortinetqa.local"
    next
end
```

5. Configure the interface:

```
config system interface
    edit "port10"
```

```
        set ip 10.1.100.1 255.255.255.0
        set explicit-web-proxy enable
        set proxy-captive-portal enable
    next
end
```

6. Configure a firewall proxy policy with users or groups (see Explicit web proxy on page 406).

Verification

When a client visits a HTTP website, the client will be redirected to the captive portal for authentication by HTTPS. For example, the client could be redirected to a URL by a HTTP 303 message similar to the following:

HTTP/1.1 303 See Other

Connection: close

Content-Type: text/html

Cache-Control: no-cache

Location: https://fgt.fortinetqa.local:7831/XX/YY/ZZ/cpauth?scheme=http&4Tmthd=0&host=172.16.200.46&port=80&rule=75&uri=Lw==&

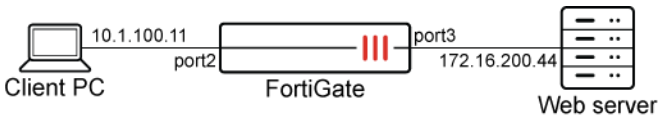
Content-Length: 0

The captive portal URL used for authentication is *https://fgt.fortinetqa.local:7831/...*. Once the authentication is complete with all user credentials protected by HTTPS, the client is redirected to the original HTTP website they intended to visit.

mTLS client certificate authentication

FortiGate supports client certificate authentication used in mutual Transport Layer Security (mTLS) communication between a client and server. Clients are issued certificates by the CA, and an access proxy configured on the FortiGate uses the new certificate method in the authentication scheme to identify and approve the certificate provided by the client when they try to connect to the access proxy. The FortiGate can also add the HTTP header X-Forwarded-Client-Cert to forward the certificate information to the server.

Examples



In these examples, the access proxy VIP IP address is 10.1.100.200.

Example 1

In this example, clients are issued unique client certificates from your CA. The FortiGate authenticates the clients by their user certificate before allowing them to connect to the access proxy. The access server acts as a reverse proxy for the web server that is behind the FortiGate.

This example assumes that you have already obtained the public CA certificate from your CA, the root CA of the client certificate has been imported (`CA_Cert_1`), and the client certificate has been distributed to the endpoints.

To configure the FortiGate:

1. Configure user authentication. Both an authentication scheme and rule must be configured, as the authentication is applied on the access proxy:

```
config authentication scheme
    edit "mtls"
        set method cert
        set user-cert enable
    next
end
```

```
config authentication rule
    edit "mtls"
        set srcintf "port2"
        set srcaddr "all"
        set dstaddr "all"
        set active-auth-method "mtls"
    next
end
```

2. Select the CA or CAs used to verify the client certificate:

```
config authentication setting
    set user-cert-ca "CA_Cert_1"
end
```

3. Configure the users. Users can be matched based on either the common-name on the certificate or the trusted issuer.
   - Verify the user based on the common name on the certificate:

```
config user certificate
    edit "single-certificate"
        set type single-certificate
        set common-name "client.fortinet.com"
    next
end
```

   - Verify the user based on the CA issuer:

```
config user certificate
    edit "trusted-issuer"
        set type trusted-issuer
```

```
        set issuer "CA_Cert_1"
    next
end
```

4. Configure the access proxy VIP. The SSL certificate is the server certificate that is presented to the user as they connect. To request the client certificate for authentication, `client-cert` is enabled:

```
config firewall vip
    edit "mTLS"
        set type access-proxy
        set extip 10.1.100.200
        set extintf "port2"
        set server-type https
        set empty-cert-action accept
        set client-cert enable
        set extport 443
        set ssl-certificate "Fortinet_SSL"
    next
end
```

5. Configure the access proxy policy, including the real server to be mapped:

```
config firewall access-proxy
    edit "mTLS-access-proxy"
        set vip "mTLS"
        config api-gateway
            edit 1
                config realservers
                    edit 1
                        set ip 172.16.200.44
                    next
                end
            next
        end
    next
end
```

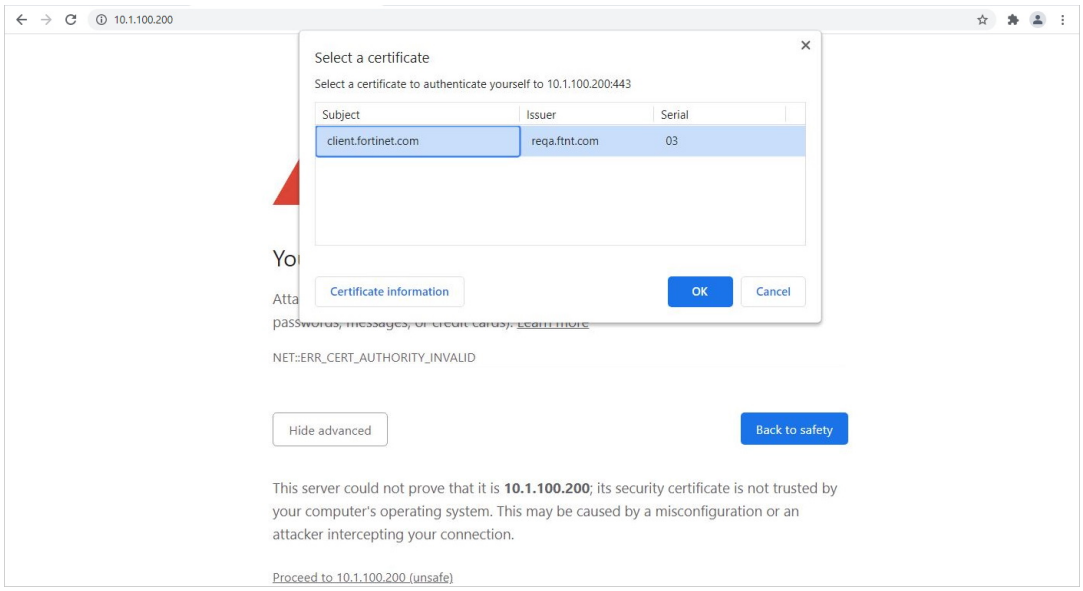
6. Configure the proxy policy to apply authentication and the security profile, selecting the appropriate user object depending on the user type:

```
config firewall proxy-policy
    edit 3
        set proxy access-proxy
        set access-proxy "mTLS-access-proxy"
        set srcintf "port2"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set users {"single-certificate" | "trusted-issuer"}
        set utm-status enable
        set ssl-ssh-profile "deep-inspection-clone"
```

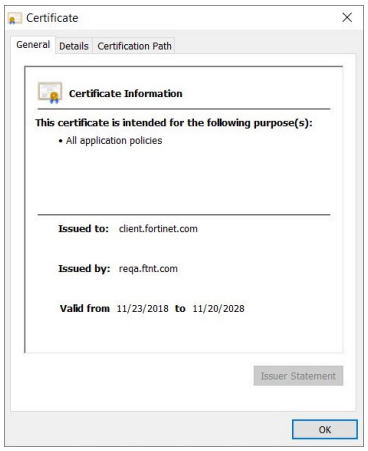
```
        set av-profile "av"
    next
end
```

To verify the results:

1. In a web browser, access the VIP address. This example uses Chrome.
2. When prompted, select the client certificate, then click *OK*.



3. Click *Certificate information* to view details about the certificate.



4. On the FortiGate, check the traffic logs.
   - If client certificate authentication passes:

```
1: date=2021-06-03 time=15:48:36 eventtime=1622760516866635697 tz="-0700"
logid="0000000010" type="traffic" subtype="forward" level="notice" vd="vdom1"
srcip=10.1.100.11 srcport=45532 srcintf="port2" srcintfrole="undefined"
dstcountry="Reserved" srccountry="Reserved" dstip=172.16.200.44 dstport=443
dstintf="vdom1" dstintfrole="undefined" sessionid=154900 service="HTTPS"
wanoptapptype="web-proxy" proto=6 action="accept" policyid=3 policytype="proxy-policy"
```

```
poluuid="af5e2df2-c321-51eb-7d5d-42fa58868dcb" duration=0 user="single-certificate"
wanin=2550 rcvdbyte=2550 wanout=627 lanin=4113 sentbyte=4113 lanout=2310
appcat="unscanned"
```

- If the CA issuer is used to verify the client:

```
1: date=2021-06-03 time=15:43:02 eventtime=1622760182384776037 tz="-0700"
logid="0000000010" type="traffic" subtype="forward" level="notice" vd="vdom1"
srcip=10.1.100.11 srcport=45514 srcintf="port2" srcintfrole="undefined"
dstcountry="Reserved" srccountry="Reserved" dstip=10.1.100.200 dstport=443 dstintf="vdom1"
dstintfrole="undefined" sessionid=153884 service="HTTPS" wanoptapptype="web-proxy" proto=6
action="accept" policyid=3 policytype="proxy-policy" poluuid="af5e2df2-c321-51eb-7d5d-
42fa58868dcb" duration=0 user="trusted-issuer" wanin=0 rcvdbyte=0 wanout=0 lanin=4089
sentbyte=4089 lanout=7517 appcat="unscanned" utmaction="block" countweb=1 crscore=30
craction=8 utmref=65535-0
```

- If the client certificate authentication fails, and the traffic is blocked:

```
1: date=2021-06-03 time=15:45:53 eventtime=1622760353789703671 tz="-0700"
logid="0000000013" type="traffic" subtype="forward" level="notice" vd="vdom1"
srcip=10.1.100.11 srcport=45518 srcintf="port2" srcintfrole="undefined"
dstip=172.16.200.44 dstport=443 dstintf="vdom1" dstintfrole="undefined"
srccountry="Reserved" dstcountry="Reserved" sessionid=154431 proto=6 action="deny"
policyid=0 policytype="proxy-policy" user="single-certificate" service="HTTPS"
trandisp="noop" url="https://10.1.100.200/" agent="curl/7.68.0" duration=0 sentbyte=0
rcvdbyte=0 sentpkt=0 rcvdpkt=0 appcat="unscanned" crscore=30 craction=131072
crlevel="high" msg="Traffic denied because of explicit proxy policy"
```

Example 2

In this example, the same configuration as in Example 1 is used, with a web proxy profile added to enable adding the client certificate to the HTTP header X-Forwarded-Client-Cert. The header is then forwarded to the server.

To configure the FortiGate:

1. Repeat steps 1 to 6 of Example 1, using the common name on the certificate to verify the user.
2. Configure a web proxy profile that adds the HTTP x-forwarded-client-cert header in forwarded requests:

```
config web-proxy profile
    edit "mtls"
        set header-x-forwarded-client-cert add
    next
end
```

3. Configure the proxy policy to apply authentication, the security profile, and web proxy profile:

```
config firewall proxy-policy
    edit 3
        set uuid af5e2df2-c321-51eb-7d5d-42fa58868dcb
        set proxy access-proxy
        set access-proxy "mTLS-access-proxy"
```

```
        set srcintf "port2"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set logtraffic all
        set users "single-certificate"
        set webproxy-profile "mtls"
        set utm-status enable
        set ssl-ssh-profile "deep-inspection-clone"
        set av-profile "av"
    next
end
```

To verify the results:

The WAD debug shows that the FortiGate adds the client certificate information to the HTTP header. The added header cannot be checked using the sniffer, because the FortiGate encrypts the HTTP header to forward it to the server.

1. Enable WAD debug on all categories:

```
# diagnose wad debug enable category all
```

2. Set the WAD debug level to verbose:

```
# diagnose wad debug enable level verbose
```

3. Enable debug output:

```
# diagnose debug enable
```

4. Check the debug output.
   - When the FortiGate receives the client HTTP request:

```
[0x7fc8d4bc4910] Received request from client: 10.1.100.11:45544

GET / HTTP/1.1
Host: 10.1.100.200
User-Agent: curl/7.68.0
Accept: */*
```

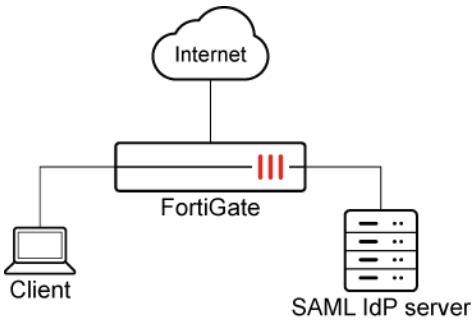
   - When the FortiGate adds the client certificate in to the HTTP header and forwards the client HTTP request:

```
[0x7fc8d4bc4910] Forward request to server:
GET / HTTP/1.1
Host: 172.16.200.44
User-Agent: curl/7.68.0
Accept: */*
X-Forwarded-Client-Cert: -----BEGIN CERTIFICATE-----
MIIFXzCCA0egAwI...aCFHDHlR+wb39s=
```

```
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
MIIFpTCCA42gAwI...OtDtetkNoFLbvb
-----END CERTIFICATE-----
```

CORS protocol in explicit web proxy when using session-based, cookie-enabled, and captive portal-enabled SAML authentication

The FortiGate explicit web proxy supports the Cross-Origin Resource Sharing (CORS) protocol, which allows the FortiGate to process a CORS preflight request and an actual CORS request properly, in addition to a simple CORS request when using session-based, cookie-enabled, and captive portal-enabled SAML authentication. This allows a FortiGate explicit web proxy user with this specific configuration to properly view a web page requiring CORS with domains embedded in it other than its own domain.



To configure the FortiGate:

1. Configure the authentication rule:

```
config authentication rule
    edit "saml"
        set srcaddr "all"
        set ip-based disable
        set active-auth-method "saml"
        set web-auth-cookie enable
    next
end
```

2. Configure the captive portal:

```
config authentication setting
    set captive-portal "fgt9.myqalab.local"
end
```

3. Configure the proxy policy

```
config firewall proxy-policy
    edit 3
        set proxy explicit-web
        set dstintf "port9"
        set srcaddr "all"
        set dstaddr "all"
        set service "webproxy"
        set action accept
        set schedule "always"
        set logtraffic all
        set groups "ldap-group-saml"
        set utm-status enable
        set profile-protocol-options "protocol"
        set ssl-ssh-profile "deep-custom"
        set av-profile "av"
        set application-list "fff"
    next
end
```

CORS request scenarios

Preflight CORS request

The client sends the initial CORS preflight request (`OPTIONS` with the origin header) to the web server through FortiGate's web proxy and receives a CORS `200 OK` response (with headers, such as `Access-Control-Allow-Origin`). The FortiGate will not redirect the client to the captive capital for authentication:

```
> OPTIONS /bidRequest HTTP/1.1
> Host: c2shb.pubgw.yahoo.com
> User-Agent: curl/7.61.1
> Accept: */*
> Access-Control-Request-Method: GET
> Access-Control-Request-Headers: content-type,x-openrtb-version
> Origin: https://www.cnn.com
...
< HTTP/1.1 200 OK
< Date: Thu, 19 May 2022 01:49:17 GMT
< Content-Length: 0
< Server: ATS/9.1.0.46
< Access-Control-Allow-Origin: https://www.cnn.com
< Access-Control-Allow-Methods: GET,POST,OPTIONS
< Access-Control-Allow-Headers: X-Requested-With,Content-Type,X-Openrtb-Version
< Access-Control-Allow-Credentials: true
< Access-Control-Max-Age: 600
< Age: 0
< Connection: keep-alive
< Set-Cookie: A3=d=AQABBB2ihWICEIUyD_Du5ol8tMdKKWxspR8FEgEBAQHzhmKPYgAAAAAA_
eMAAA&S=AQAAAlU0dAheQx6euvcPs8ErK4I; Expires=Fri, 19 May 2023 07:49:17 GMT; Max-Age=31557600;
Domain=.yahoo.com; Path=/; SameSite=None; Secure; HttpOnly
```

Real CORS request

Once the initial preflight request for the client is successful, the client sends the real CORS request (`GET` request with origin header) to the FortiGate, The FortiGate then replies with a 30x response to redirect the client to the captive portal. The 30x response includes CORS headers such as `Access-Control-Allow-Origin`:

```
> GET /bidRequest HTTP/1.1
> Host: c2shb.pubgw.yahoo.com
> User-Agent: curl/7.61.1
> Accept: */*
> Origin: https://www.cnn.com
...
< HTTP/1.1 303 See Other
< Access-Control-Max-Age: 1
< Access-Control-Allow-Origin: https://www.cnn.com
< Access-Control-Allow-Credentials: true
< Set-Cookie: FTNT-EP-
FG900D3915800054=pqWlpdswdcCnpaWli6WlpcjEwszGmJbGksbBwMCVwcPBlpKRnMGTl52QxJeUwYPW18aYlJWLlIuUlZWLl
JalpQ==; Path=/; Domain=.pubgw.yahoo.com; HttpOnly; SameSite=None; Secure
< Connection: close
< Content-Type: text/html
< Cache-Control: no-cache
< Location:
https://fgt9.myqalab.local:7831/test/saml/login/?cptype=ckauth&scheme=https&4Tmthd=0&host=c2shb.pu
bgw.yahoo.com&port=443&rule=98&uri=L2JpZFJlcXVlc3Q=&cdata=pqWlpdswdcCnpaWli6WlpcjEwszGmJbGksbBwMCV
wcPBlpKRnMGTl52QxJeUwYPW18aYlJWLlIuUlZWLlJalpQ==
< Content-Length: 0
```

Redirection to captive portal

Once the client's real CORS request is redirected to the captive portal, the client senda another preflight to the captive portal. The captive portal then replies with a 20x response, which includes CORS headers such as `Access-Control-Allow-Origin`:

```
> OPTIONS
/test/saml/login/?cptype=ckauth&scheme=https&4Tmthd=1&host=gql.reddit.com&port=443&rule=98&uri=Lw=
=&cdata=pqWlpQM5dcCnpaWliqWlpcjEwszGmJbGksbAk5WTl8aTwJDGnJ2Tl52QxpHDkYPW18aYlJWLlIuUlZWLlJGWpQ==
HTTP/1.1
> Host: fgt9.myqalab.local:7831
> Connection: keep-alive
> Accept: */*
> Access-Control-Request-Method: GET
> Access-Control-Request-Headers: authorization,content-type,x-reddit-compression,x-reddit-loid,x-
reddit-session
> Origin: null
> User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/100.0.4896.75 Safari/537.36 Edg/100.0.1185.36
> Sec-Fetch-Mode: cors
> Sec-Fetch-Site: cross-site
> Sec-Fetch-Dest: empty
> Referer: https://www.reddit.com/
> Accept-Encoding: gzip, deflate, br
```

```
> Accept-Language: en-US,en;q=0.9
...
< HTTP/1.1 204 No Content
< Access-Control-Max-Age: 86400
< Access-Control-Allow-Methods: GET
< Access-Control-Allow-Headers: authorization,content-type,x-reddit-compression,x-reddit-loid,x-
reddit-session
< Access-Control-Allow-Origin: null
< Access-Control-Allow-Credentials: true
```

Simple CORS request

If a simple CORS request (no preflight request sent before it) is used, when the FortiGate receives the simple request, it replies with a 30x response that includes CORS headers, such as `Access-Control-Allow-Origin`:

```
> Host: www.yahoo.com
> User-Agent: curl/7.61.1
> Accept: */*
> Origin: https://www.cnn.com
...
< HTTP/1.1 303 See Other
< Access-Control-Max-Age: 1
< Access-Control-Allow-Origin: https://www.cnn.com
< Access-Control-Allow-Credentials: true
< Set-Cookie: FTNT-EP-
FG900D3915800000=pqWlpaw7dcCnpaWli6WlpcjEwszGmJbGksbAkpOcxMDDlpbGlMSTl52QwcGcl4PW18aYlJWLlIuUlZWLl
JalpQ==; Path=/; Domain=.yahoo.com; HttpOnly; SameSite=None; Secure
< Connection: close
< Content-Type: text/html
< Cache-Control: no-cache
< Location:
https://fgt9.myqalab.local:7831/test/saml/login/?cptype=ckauth&scheme=https&4Tmthd=0&host=www.yaho
o.com&port=443&rule=98&uri=Lw==&cdata=pqWlpaw7dcCnpaWli6WlpcjEwszGmJbGksbAkpOcxMDDlpbGlMSTl52QwcGc
l4PW18aYlJWLlIuUlZWLlJalpQ==
< Content-Length: 0
```

Display CORS content in an explicit proxy environment

Webpages can display Cross-Origin Resource Sharing (CORS) content in an explicit proxy environment when using session-based, cookie-enabled, and captive portal assisted authentication. This ensures that webpages are displayed correctly and improves the user experience.

```
config authentication rule
    edit <name>
        set web-auth-cookie enable
        set cors-stateful {enable | disable}
        set cors-depth <integer>
    next
end
```

`cors-stateful {enable | disable}`	Enable/disable allowing CORS access (default = disable). This setting is only available when `web-auth-cookie` is enabled.
`cors-depth <integer>`	Set the depth to allow CORS access (1 - 8, default = 3). For example, when visiting domain A, the returned web page may refer the browser to a cross-origin domain B (depth of 1). When the browser visits domain B, the returned web content may further refer the browser to another cross-origin domain C (depth of 2).

Example

CORS access is enabled in this example. When a user access the Microsoft *Sign in* page using an explicit proxy, the page appears and the user can log in. This example assumes the web proxy and user group have already been configured, and that the proxy captive portal setting has been enabled on the appropriate interface.

To view CORS content in an explicit proxy environment:

1. Configure the authentication scheme:

```
config authentication scheme
    edit "form"
        set method form
        set user-database "local-user-db"
    next
end
```

2. Configure the authentication rule:

```
config authentication rule
    edit "form"
        set srcaddr "all"
        set ip-based disable
        set active-auth-method "form"
        set web-auth-cookie enable
        set cors-stateful enable
        set cors-depth 3
    next
end
```

3. Configure the captive portal:

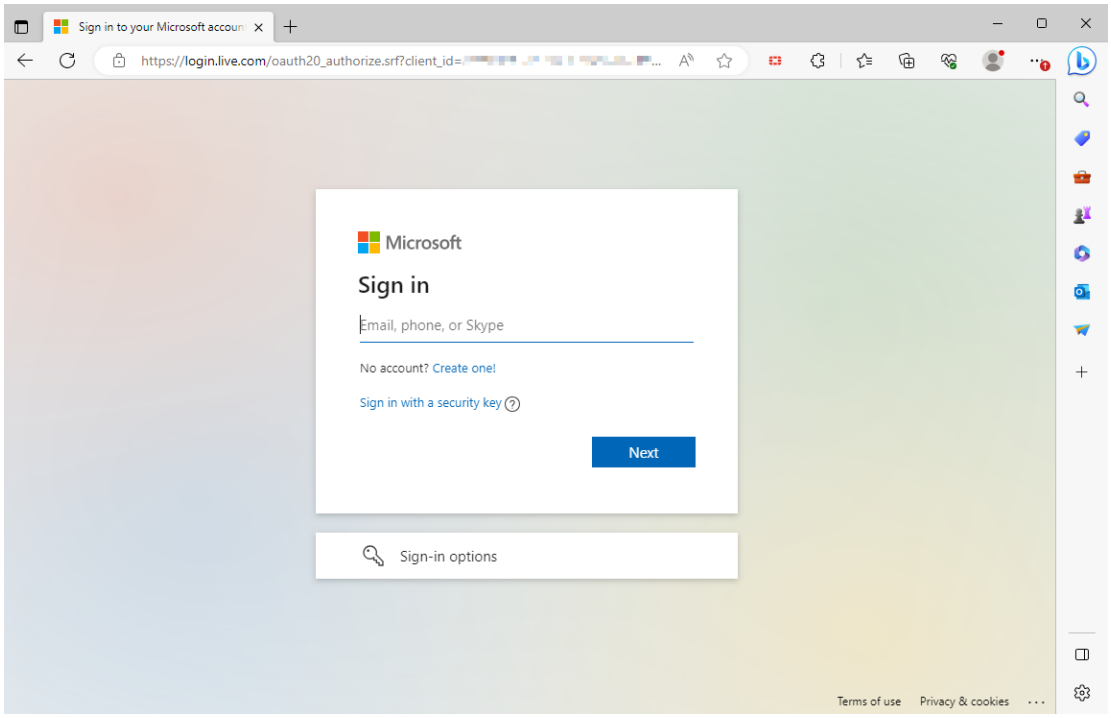
```
config authentication setting
    set captive-portal "fgt9.myqalab.local"
end
```

4. Configure the proxy policy:

```
config firewall proxy-policy
    edit 1
        set proxy explicit-web
        set dstintf "port9"
```

```
        set srcaddr "all"
        set dstaddr "all"
        set service "webproxy"
        set action accept
        set schedule "always"
        set logtraffic all
        set groups "localgroup"
        set utm-status enable
        set ssl-ssh-profile "deep-custom"
        set av-profile "av"
    next
end
```

5. Get a user to access login.microsoftonline.com trough the explicit web proxy. The *Sign in* page appears, and the user can log in.



If CORS access (`cors-stateful`) was disabled, the browser would load a blank page.

HTTP connection coalescing and concurrent multiplexing for explicit proxy

HTTP connection coalescing and concurrent multiplexing allows multiple HTTP requests to share the same TCP three-way handshake when the destination IP is the same.

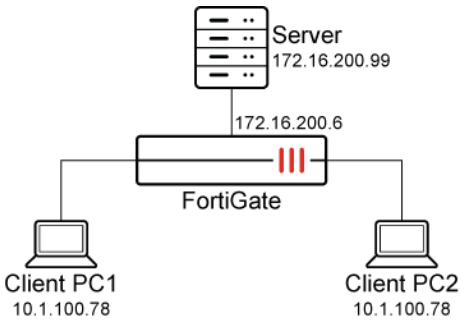
To configure the explicit web proxy:

```
config web-proxy explicit
    set http-connection-mode {static | multiplex | serverpool}
end
```

`http-connection-mode {static | multiplex | serverpool}`	Set the HTTP connection mode: • static: only one server connection exists during the proxy session (default). • multiplex: hold established connections until the proxy session ends. • serverpool: share established connections with other proxy sessions.

Example

In this example, multiple clients submit requests in HTTP. The requests hit the VIP address, and then FortiGate opens a session between itself (172.16.200.6) and the server (172.16.200.99). The coalescing occurs in this session as the multiple streams share the same session to connect to the same destination server.



To configure connection coalescing and concurrent multiplexing with an explicit proxy:

1. Configure the explicit web proxy:

```
config web-proxy explicit
    set status enable
    set http-incoming-port 8080
    set http-connection-mode serverpool
end
```

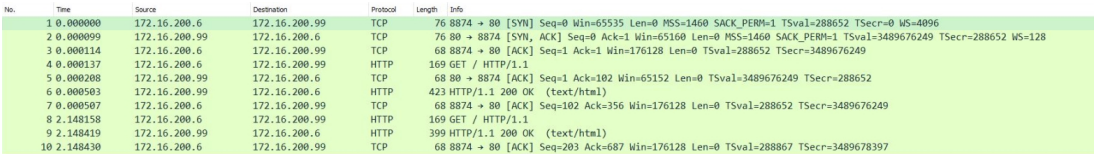
2. Enable explicit web proxy on port2:

```
config system interface
    edit "port2"
        set ip 10.1.100.6 255.255.255.0
        set explicit-web-proxy enable
    next
end
```

3. Configure the proxy policy:

```
config firewall proxy-policy
    edit 1
        set proxy explicit-web
        set dstintf "port3"
        set srcaddr "all"
        set dstaddr "all"
        set service "web"
        set action accept
        set schedule "always"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set utm-status enable
        set profile-protocol-options "default-clone"
        set ssl-ssh-profile "deep-inspection-clone"
    next
end
```

4. Get the clients to access the server through the explicit web proxy (10.1.100.6:8080). The FortiGate shares the first connection TCP three-way handshake with later connections that connect to same destination address.
5. Verify the sniffer packet capture on the FortiGate server side. There is one TCP three-way handshake, but there are two HTTP connections.

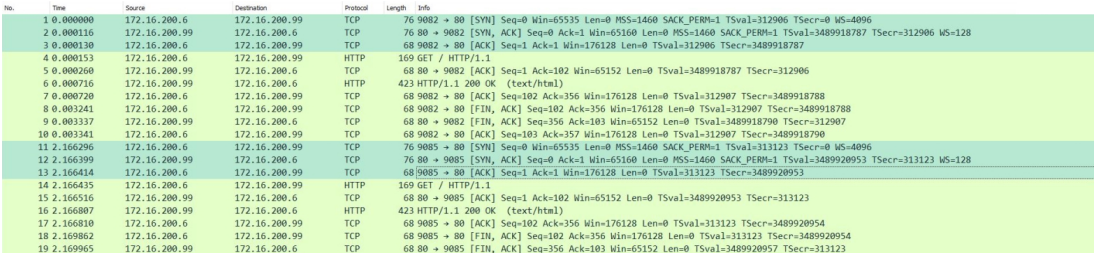


No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.16.200.6	172.16.200.99	TCP	76	8874 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1 TSval=288652 TSecr=0 WS=4096
2	0.000099	172.16.200.99	172.16.200.6	TCP	76	80 → 8874 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=3489676249 TSecr=288652 WS=128
3	0.000114	172.16.200.6	172.16.200.99	TCP	68	8874 → 80 [ACK] Seq=1 Ack=1 Win=176128 Len=0 TSval=288652 TSecr=3489676249
4	0.000137	172.16.200.6	172.16.200.99	HTTP	169	GET / HTTP/1.1
5	0.000208	172.16.200.99	172.16.200.6	TCP	68	80 → 8874 [ACK] Seq=1 Ack=102 Win=65152 Len=0 TSval=3489676249 TSecr=288652
6	0.000503	172.16.200.99	172.16.200.6	HTTP	423	HTTP/1.1 200 OK (text/html)
7	0.000507	172.16.200.6	172.16.200.99	TCP	68	8874 → 80 [ACK] Seq=102 Ack=356 Win=176128 Len=0 TSval=288652 TSecr=3489676249
8	2.148158	172.16.200.6	172.16.200.99	HTTP	169	GET / HTTP/1.1
9	2.148419	172.16.200.99	172.16.200.6	HTTP	399	HTTP/1.1 200 OK (text/html)
10	2.148430	172.16.200.6	172.16.200.99	TCP	68	8874 → 80 [ACK] Seq=203 Ack=687 Win=176128 Len=0 TSval=288867 TSecr=3489678397

6. Change the HTTP connection mode to static:

```
config web-proxy explicit
    set status enable
    set http-incoming-port 8080
    set http-connection-mode static
end
```

7. Verify the sniffer packet capture. This time, the FortiGate establishes a TCP connection for each client.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.16.200.6	172.16.200.99	TCP	76	9082 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1 TSval=312906 TSecr=0 WS=4096
2	0.000116	172.16.200.99	172.16.200.6	TCP	76	80 → 9082 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=3489918787 TSecr=312906 WS=128
3	0.000130	172.16.200.6	172.16.200.99	TCP	68	9082 → 80 [ACK] Seq=1 Ack=1 Win=176128 Len=0 TSval=312906 TSecr=3489918787
4	0.000153	172.16.200.6	172.16.200.99	HTTP	169	GET / HTTP/1.1
5	0.000260	172.16.200.99	172.16.200.6	TCP	68	80 → 9082 [ACK] Seq=1 Ack=102 Win=65152 Len=0 TSval=3489918787 TSecr=312906
6	0.000716	172.16.200.99	172.16.200.6	HTTP	423	HTTP/1.1 200 OK (text/html)
7	0.000720	172.16.200.6	172.16.200.99	TCP	68	9082 → 80 [ACK] Seq=102 Ack=356 Win=176128 Len=0 TSval=312907 TSecr=3489918788
8	0.003241	172.16.200.6	172.16.200.99	TCP	68	9082 → 80 [FIN, ACK] Seq=102 Ack=356 Win=176128 Len=0 TSval=312907 TSecr=3489918788
9	0.003337	172.16.200.99	172.16.200.6	TCP	68	80 → 9082 [FIN, ACK] Seq=356 Ack=103 Win=65152 Len=0 TSval=3489918790 TSecr=312907
10	0.003341	172.16.200.6	172.16.200.99	TCP	68	9082 → 80 [ACK] Seq=103 Ack=357 Win=176128 Len=0 TSval=312907 TSecr=3489918790
11	2.166296	172.16.200.6	172.16.200.99	TCP	76	9085 → 80 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1 TSval=313123 TSecr=0 WS=4096
12	2.166399	172.16.200.99	172.16.200.6	TCP	76	80 → 9085 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=3489920953 TSecr=313123 WS=128
13	2.166414	172.16.200.6	172.16.200.99	TCP	68	9085 → 80 [ACK] Seq=1 Ack=1 Win=176128 Len=0 TSval=313123 TSecr=3489920953
14	2.166435	172.16.200.6	172.16.200.99	HTTP	169	GET / HTTP/1.1
15	2.166516	172.16.200.99	172.16.200.6	TCP	68	80 → 9085 [ACK] Seq=1 Ack=102 Win=65152 Len=0 TSval=3489920953 TSecr=313123
16	2.166807	172.16.200.99	172.16.200.6	HTTP	423	HTTP/1.1 200 OK (text/html)
17	2.166810	172.16.200.6	172.16.200.99	TCP	68	9085 → 80 [ACK] Seq=102 Ack=356 Win=176128 Len=0 TSval=313123 TSecr=3489920954
18	2.169862	172.16.200.6	172.16.200.99	TCP	68	9085 → 80 [FIN, ACK] Seq=102 Ack=356 Win=176128 Len=0 TSval=313123 TSecr=3489920954
19	2.169965	172.16.200.99	172.16.200.6	TCP	68	80 → 9085 [FIN, ACK] Seq=356 Ack=103 Win=65152 Len=0 TSval=3489920957 TSecr=313123

Secure explicit proxy

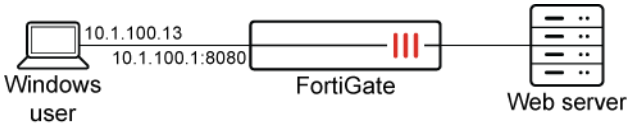
Secure explicit web proxy with HTTPS connections is supported between web clients and the FortiGate.

```
config web-proxy explicit
    set secure-web-proxy {disable | enable | secure}
    set secure-web-proxy-cert <certificate1> <certificate2> ...
    set ssl-dh-bits {768 | 1024 | 1536 | 2048}
end
```

`secure-web-proxy {disable | enable | secure}`	Enable/disable/require the secure web proxy for HTTP and HTTPS session. • `disable`: disable secure web proxy (default) • `enable`: enable secure web proxy access, allowing both HTTPS and HTTP connections to the explicit proxy • `secure`: require secure web proxy access, allowing only HTTPS connections to the explicit proxy
`secure-web-proxy-cert <certificate1> <certificate2> ...`	Enter the names of the server certificates in the local certificate store of the FortiGate used to establish a TLS connection between the user's browser and the FortiGate. Multiple server certificates can be configured, and different certificate types can be used. The SNI tries to match the right server certificate for the connection. If the SNI cannot not match with the certificates' CN or SAN, the first server certificate will be offered.
`ssl-dh-bits {768 | 1024 | 1536 | 2048}`	Set the bit size of Diffie-Hellman (DH) prime used in the DHE-RSA negotiation. • `768`: use 768-bit Diffie-Hellman prime • `1024`: use 1024-bit Diffie-Hellman prime • `1536`: use 1536-bit Diffie-Hellman prime • `2048`: use 2048-bit Diffie-Hellman prime (default)

Example

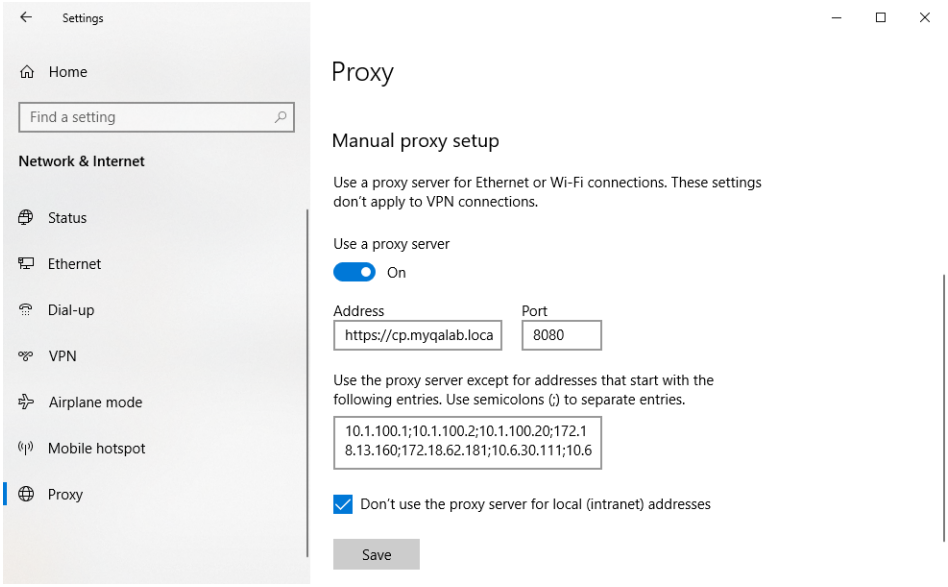
In this example, a Windows PC user configures an HTTPS URL (https://cp.myqalab.local) as the proxy address for the explicit web proxy. When the user opens a browser (such as Edge or Chrome), the browser will use the HTTPS URL to connect to the explicit web proxy and send any HTTP requests to the proxy over HTTPS. The certificate (server_cert) contains the explicit web proxy's name (cp.myqalab.local) as its CN, so the browser will accept this certificate for the TLS connection.



To configure the Windows proxy settings:

1. On the Windows PC, go to *Settings* > *Network & Internet* > *Proxy*.
2. In the *Manual proxy setup* section configure the following:
   a. Enable *Use a proxy server*.
   b. Set the *Address* to *https://cp.myqalab.local*.
   c. Set the *Port* to *8080*.

d. If needed, enter any addresses to exempt in the text box (use a semicolon to separate entries).
e. Enable *Don't use the proxy server for local (intranet) addresses*.



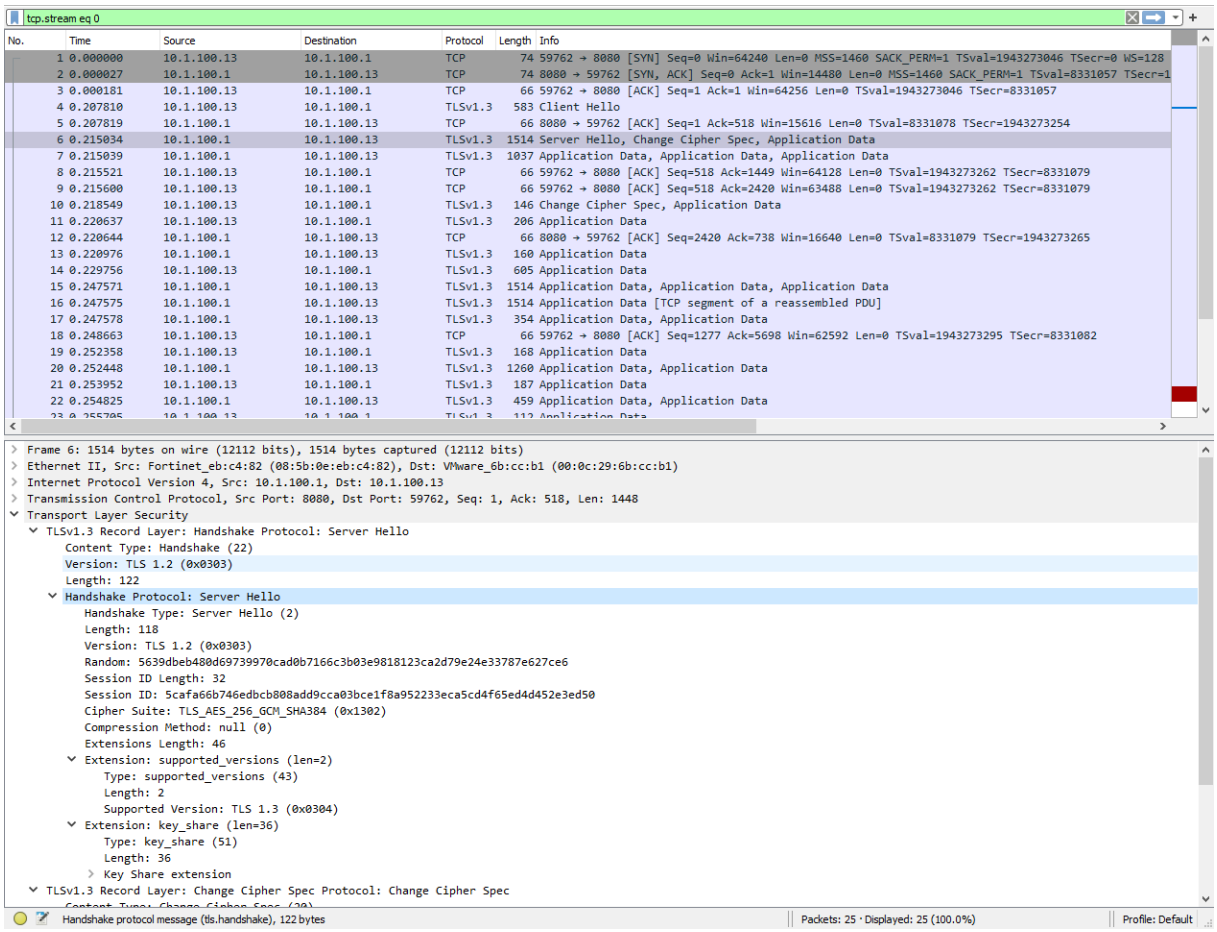
3. Click *Save*.

To configure the secure explicit web proxy:

```
config web-proxy explicit
    set status enable
    set secure-web-proxy enable
    set ftp-over-http enable
    set socks enable
    set http-incoming-port 8080
    set secure-web-proxy-cert "server_cert"
    set socks-incoming-port 1080
    set ipv6-status enable
    set unknown-http-version best-effort
    set pac-file-server-status enable
    set pac-file-data "function FindProxyForURL(url, host) {
   // testtest
   return \"PROXY 10.1.100.1:8080\";
}
"
    set pac-file-through-https enable
end
```

To verify the TLS connection:

1. Perform a packet capture of HTTPS traffic between the web client and the web server. Wireshark is used in this example.
2. Locate the exchange between the web client (10.1.100.13) and the explicit web proxy (10.1.100.1:8080):



After the client initiates the TLS connection to the explicit web proxy with a client hello packet, the web proxy is able to respond appropriately with a server hello packet to establish a TLS connection first before any HTTP messages are exchanged, and all HTTP messages will be protected by the TLS connection.

Secure explicit proxy with client certificates

The explicit web proxy policy can use client certificates for validation. In this example, a CA signs a client certificate. The client certificate is installed on an endpoint, and the root CA is imported to FortiGate. A web proxy policy is configured to require the client certificate.

When the user accesses a web site, the explicit web proxy policy uses the client certificate from the endpoint device to authenticate the user and grant access to the web site.

To configure client certificates with explicit proxies:

1. Prepare the certificate:
   a. Use a CA to sign the client certificate.
   b. Import the root CA certificate that signed the client certificate to FortiGate. In this scenario, the certificate is root_ca.
   c. Install the client certificate on an endpoint.

2. Configure the explicit web-proxy policy to request the client certificate from the endpoint.

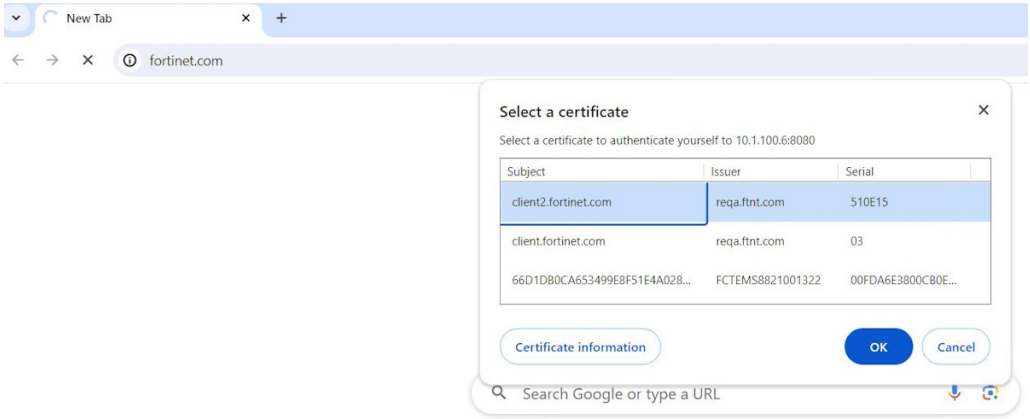
```
config web-proxy explicit
    set secure-web-proxy enable
    set secure-web-proxy-cert "proxy"
    set client-cert enable
    set empty-cert-action block
end
```

3. Configure verification of the client certificate with the root CA.

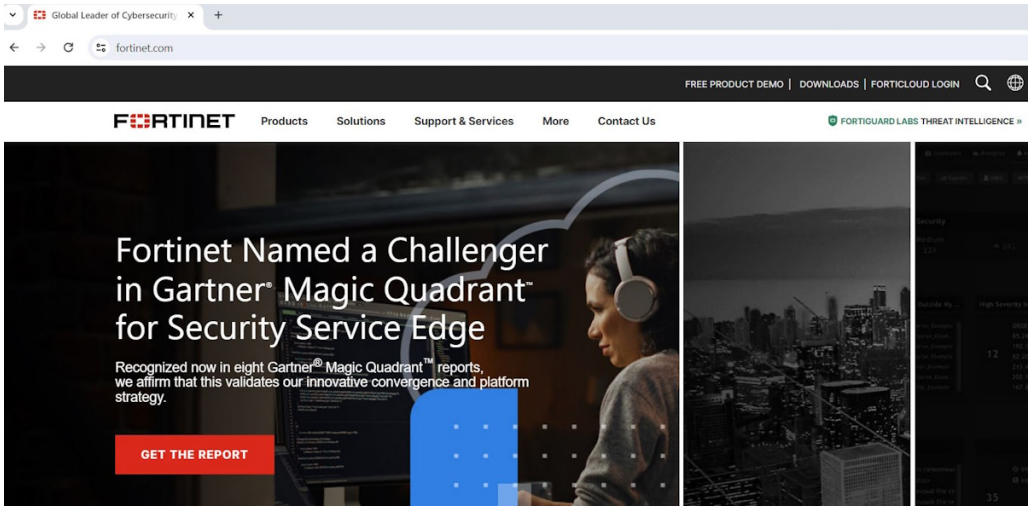
```
config authentication setting
    set user-cert-ca "root_ca"
end
```

When the user accesses a web site:

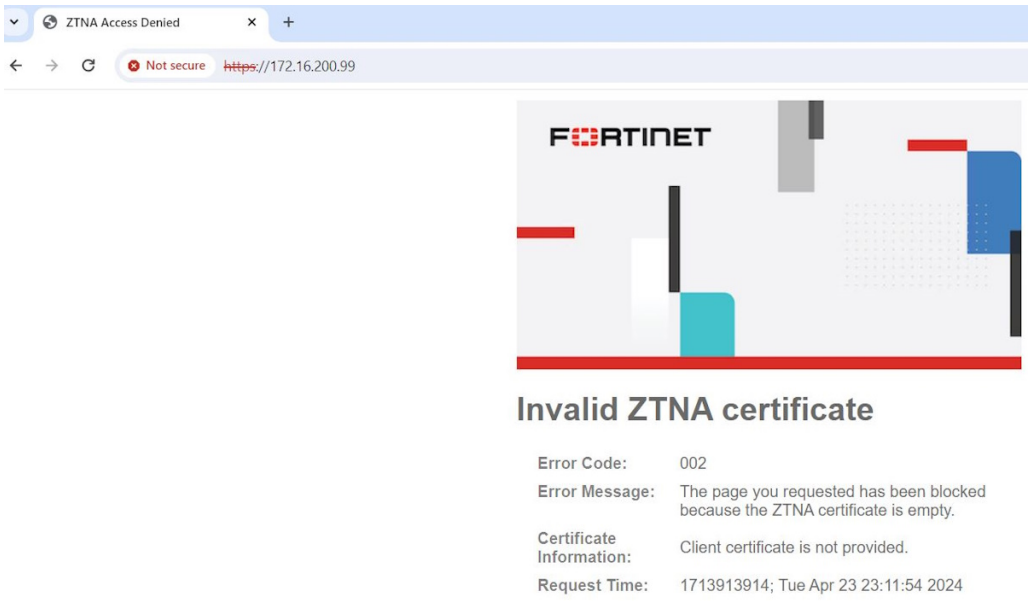
1. FortiGate requests client certificate authentication, and the web browser displays the available certificates. The user selects a client certificate (client2.fortinet.com) issued by the reqa.ftnt.com CA, and clicks *OK*.



2. Once the client certificate is successfully verified against the root CA certificate imported on the FortiGate, access to the web site is granted.



When the endpoint device fails to present a client certificate, a message is displayed, and access to the web site is blocked.



Explicit proxy logging

Explicit proxy traffic logging can be used to troubleshoot the HTTP proxy status for each HTTP transaction with the following:

- Monitor HTTP header requests and responses in the UTM web filter log. This requires an SSL deep inspection profile to be configured in the corresponding firewall policy.
- Log the explicit web proxy forward server name using `set log-forward-server`, which is disabled by default.

```
config web-proxy global
    set log-forward-server {enable | disable}
end
```

- Log TCP connection failures in the traffic log when a client initiates a TCP connection to a remote host through the FortiGate and the remote host is unreachable.
- Log HTTP transaction details

Basic configuration

The following FortiGate configuration is used in the three explicit proxy traffic logging use cases in this topic.

To configure the FortiGate:

1. Configure the web proxy profile:

```
config web-proxy profile
    edit "header"
        config headers
```

```
            edit 1
                set name "test_request_header"
                set action monitor-request
            next
            edit 2
                set name "ETag"
                set action monitor-response
            next
        end
    next
end
```

2. Enable forward server name logging in traffic:

```
config web-proxy global
    set proxy-fqdn "100D.qa"
    set log-forward-server enable
end
```

3. Configure the web filter banned word table to block any HTTP response containing the text, `works`:

```
config webfilter content
    edit 1
        set name "default"
        config entries
            edit "works"
                set status enable
                set action block
            next
        end
    next
end
```

4. Configure the web filter profile:

```
config webfilter profile
    edit "header"
        set feature-set proxy
        config web
            set bword-table 1
        end
        config ftgd-wf
            unset options
        end
        set log-all-url enable
        set extended-log enable
        set web-extended-all-action-log enable
    next
end
```

5. Configure the web proxy forwarding server:

```
config web-proxy forward-server
    edit "fgt-b"
        set ip 172.16.200.20
    next
end
```

6. Configure the firewall policy:

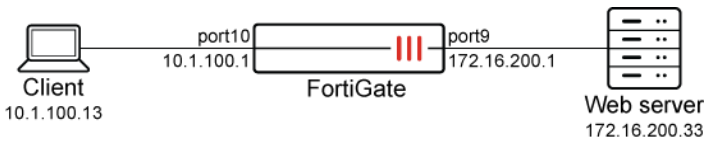
```
config firewall policy
    edit 1
        set srcintf "port10"
        set dstintf "port9"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set webproxy-profile "header"
        set webproxy-forward-server "fgt-b"
        set ssl-ssh-profile "deep-inspection"
        set webfilter-profile "header"
        set logtraffic all
        set nat enable
    next
end
```

A firewall policy is used in this basic configuration example and the specific examples that follow. This feature also works for the explicit web proxy or transparent web proxy with proxy policies, and the configurations are similar:

- Example 1: apply the `web-proxy profile` and `webfilter profile` to the proxy policy.
- Example 2: apply the `webproxy-forward-server`.

Example 1: monitoring HTTP header requests

In this example, the user wants to monitor some HTTP headers in HTTP messages forwarded through a FortiGate proxy (either transparent or explicit proxy with a firewall policy in proxy mode or a proxy policy). When the monitored headers are detected, they will be logged in the UTM web filter log.



In the web proxy profile configuration, the following HTTP headers are monitored:

- `test_request_header`: this is a user-customized HTTP header.
- `ETag`: this is a HTTP header returned by the web server's 200 OK response.

Based on the web filter profile configuration, the monitored headers in the web proxy profile will only be logged when the HTTP response received by the FortiGate triggers a block action by the banned word table. The `log-all-url`, `extended-log`, and `web-extended-all-action-log` settings in the web filter profile must be enabled.

The following settings are required in the firewall policy:

- `set inspection-mode proxy`
- `set webproxy-profile "header"`
- `set ssl-ssh-profile "deep-inspection"`
- `set webfilter-profile "header"`
- `set logtraffic all`

To verify the configuration:

1. Send a HTTP request from the client:

   ```
   curl -kv https://172.16.200.33 -H "test_request_header: aaaaa"
   ```

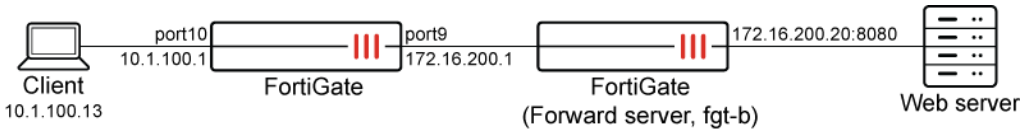
   This command sends a HTTP request with the header `test_request_header: aaaaa` through the FortiGate. Since the response from the web server contains the word `works`, the response will be blocked by the web filter profile (`header`). During this process, two logs will be generated.
2. On the FortiGate, check the traffic logs:

   ```
   # execute log filter category 3
   1: date=2023-04-19 time=19:01:19 eventtime=1681956079146481995 tz="-0700" logid="0314012288"
   type="utm" subtype="webfilter" eventtype="content" level="warning" vd="vdom1" policyid=1
   poluuid="4d8dc396-46e3-51ea-7f3f-ee328a5bd07b" policytype="policy" sessionid=40980
   srcip=10.1.100.13 srcport=54512 srccountry="Reserved" srcintf="port10" srcintfrole="undefined"
   srcuuid="6ce0b8ca-30ae-51ea-a388-ceacbb4fb045" dstip=172.16.200.33 dstport=443
   dstcountry="Reserved" dstintf="port9" dstintfrole="undefined" dstuuid="6ce0b8ca-30ae-51ea-
   a388-ceacbb4fb045" proto=6 httpmethod="GET" service="HTTPS" hostname="172.16.200.33"
   agent="curl/7.61.1" profile="header" reqtype="direct" url="https://172.16.200.33/" sentbyte=0
   rcvdbyte=0 direction="incoming" action="blocked" banword="works" msg="URL was blocked because
   it contained banned word(s)." rawdata="[REQ] test_request_header=aaaaa||[RESP] Content-
   Type=text/html|ETag=\"34-5b23b9d3b67f4\""

   2: date=2023-04-19 time=19:01:19 eventtime=1681956079144896978 tz="-0700" logid="0319013317"
   type="utm" subtype="webfilter" eventtype="urlmonitor" level="notice" vd="vdom1" policyid=1
   poluuid="4d8dc396-46e3-51ea-7f3f-ee328a5bd07b" policytype="policy" sessionid=40980
   srcip=10.1.100.13 srcport=54512 srccountry="Reserved" srcintf="port10" srcintfrole="undefined"
   srcuuid="6ce0b8ca-30ae-51ea-a388-ceacbb4fb045" dstip=172.16.200.33 dstport=443
   dstcountry="Reserved" dstintf="port9" dstintfrole="undefined" dstuuid="6ce0b8ca-30ae-51ea-
   a388-ceacbb4fb045" proto=6 httpmethod="GET" service="HTTPS" hostname="172.16.200.33"
   agent="curl/7.61.1" profile="header" action="passthrough" reqtype="direct"
   url="https://172.16.200.33/" sentbyte=724 rcvdbyte=2769 direction="outgoing" msg="URL has been
   visited" ratemethod="ip" cat=255 rawdata="[REQ] test_request_header=aaaaa"
   ```

   Log 1 is for the blocked HTTP response that contains both monitored headers, `test_request_header` and `ETag`, and their values, `aaaaa` and `34-5b23b9d3b67f4`, respectively. Log 2 is for the HTTP request passing through the FortiGate proxy that contains `test_request_header` and its `aaaaa` value in the `rawdata` field.

Example 2: logging the explicit web proxy forward server name

In this example, the user wants to see the name of the web proxy forward server in the traffic log when the traffic is forwarded by a web proxy forward server.



In the global web proxy settings, `log-forward-server` must be enabled.

The following settings are required in the firewall policy:

- `set inspection-mode proxy`
- `set webproxy-forward-server "fgt-b"`
- `set logtraffic all`

When a HTTP request is sent through the FortiGate proxy, the request will be forwarded by the FortiGate to the upstream proxy (fgt-b), and the forward server's name will be logged in the traffic log.

To verify the configuration:

1. Send a HTTP request from the client:

```
curl -kv https://www.google.com
```

2. On the FortiGate, check the traffic logs:

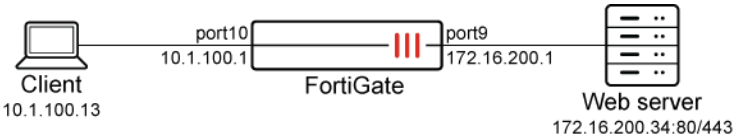
```
# execute log filter category 3
1: date=2023-04-19 time=19:51:33 eventtime=1681959093510003961 tz="-0700" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="vdom1" srcip=10.1.100.13 srcport=49762
srcintf="port10" srcintfrole="undefined" dstip=142.250.217.100 dstport=443 dstintf="port9"
dstintfrole="undefined" srccountry="Reserved" dstcountry="United States" sessionid=43292
proto=6 action="client-rst" policyid=1 policytype="policy" poluuid="4d8dc396-46e3-51ea-7f3f-
ee328a5bd07b" service="HTTPS" trandisp="snat" transip=172.16.200.1 transport=49762
duration=120 sentbyte=0 rcvdbyte=37729 sentpkt=0 rcvdpkt=33 appcat="unscanned" wanin=3779
wanout=682 lanin=879 lanout=36005 fwdsrv="fgt-b" utmaction="block" countssl=1 utmref=65506-14
```

Example 3: logging TCP connection failures

In this example, a client initiates a TCP connection to a remote network node through the FortiGate. The connection fails because the IP address or port of the remote node is unreachable. A `Connection Failed` message appears in the logs. In the firewall policy configuration, the `inspection-mode` can be set to either `proxy` or `flow` mode.

Based on the basic FortiGate configuration used in examples 1 and 2, the forward server may need to be removed from the firewall policy if the forward server's TCP IP port is actually reachable. If the forward server proxy tries to set up back-to-back TCP connections with the downstream FortiGate and the remote server as in the case of deep-inspection, then when the client tries to connect to a remote node (even if the IP address or port is unreachable), the downstream FortiGate is able to establish a TCP connection with the upstream forward server, so there will be no `Connection Failed` message in the downstream FortiGate's log.

Currently, the `Connection Failed` message in the downstream FortiGate's log is visible for the case when there is an unreachable TCP port only when explicit web proxy with a proxy policy is configured. Therefore, the following example that makes use of a firewall policy demonstrates this log message is only supported for the unreachable IP address case.



To verify the configuration:

1. Send a HTTP request from the client to an unreachable IP:

```
curl -kv https://172.16.200.34
```

2. On the FortiGate, check the traffic logs:

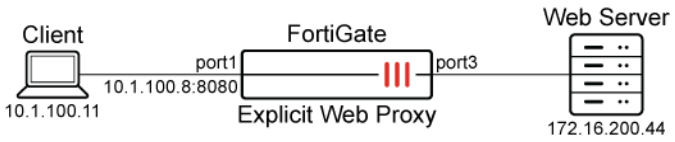
```
# execute log filter category 3
1: date=2023-04-19 time=20:25:55 eventtime=1681961155100007061 tz="-0700" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="vdom1" srcip=10.1.100.13 srcport=52452
srcintf="port10" srcintfrole="undefined" dstip=172.16.200.34 dstport=443 dstintf="port9"
dstintfrole="undefined" srccountry="Reserved" dstcountry="Reserved" sessionid=44903 proto=6
action="server-rst" policyid=1 policytype="policy" poluuid="4d8dc396-46e3-51ea-7f3f-
ee328a5bd07b" service="HTTPS" trandisp="snat" transip=172.16.200.1 transport=52452 duration=20
sentbyte=180 rcvdbyte=164 sentpkt=3 rcvdpkt=3 appcat="unscanned" wanin=0 wanout=0 lanin=0
lanout=0 crscore=5 craction=262144 crlevel="low" msg="Connection Failed"
```

Example 4: logging HTTP transaction details

HTTP transaction details are logged in a traffic log when HTTP traffic is routed through a proxy. After an HTTP transaction is proxied through the FortiGate, traffic logs of the http-transaction subtype are generated in addition to the forward subtype log. HTTP transaction logs are based on each transaction, such as an HTTP request and response pair. When there are multiple HTTP transactions completed over the TCP connection there will be multiple http-transaction logs and only one forward traffic log.

HTTP transaction logging can be enabled in explicit-web proxy, transparent-web proxy, access-proxy, and proxy-mode firewall policies.

```
config firewall proxy-policy
    edit 1
        set proxy {explicit-web | transparent-web | access-proxy}
        logtraffic {utm | all}
        set log-http-transaction {enable | disable}
    next
end
config firewall policy
    edit 1
        set inspection-mode proxy
        logtraffic {utm | all}
        set log-http-transaction {enable | disable}
    next
end
```



One http-transaction log is generated for each HTTP transaction. A TCP connection can have multiple HTTP transactions, so there can be multiple http-transaction logs for one forward traffic log.

- Explicit-web proxy logs:

  In the http-transaction logs (logs 2 and 3), transaction information such as `httpmethod` and `statuscode` are recorded.

```
1: date=2024-05-21 time=20:06:17 eventtime=1716347177537010993 tz="-0700" logid="0000000010"
type="traffic" subtype="forward" level="notice" vd="vdom1" srcip=10.1.100.11 srcport=42694
srcintf="port1" srcintfrole="undefined" dstcountry="Reserved" srccountry="Reserved"
dstip=172.16.200.44 dstport=80 dstintf="port3" dstintfrole="undefined" sessionid=316483733
service="HTTP" proxyapptype="web-proxy" proto=6 action="accept" policyid=1 policytype="proxy-
policy" poluuid="1e1e0b2e-14c1-51ef-7b4c-6b789be487f2" trandisp="snat" transip=172.16.200.8
transport=12204 duration=30 wanin=760 rcvdbyte=760 wanout=211 lanin=163 sentbyte=163
lanout=36194 appcat="unscanned" utmaction="block" countav=1 utmref=65515-14
```

```
2: date=2024-05-21 time=20:06:17 eventtime=1716347177536946272 tz="-0700" logid="0006000026"
type="traffic" subtype="http-transaction" level="notice" vd="vdom1" srcip=10.1.100.11
srcport=42694 dstip=172.16.200.44 dstport=80 sessionid=316483733 transid=50331679
action="accept" policyid=1 policytype="proxy-policy" poluuid="1e1e0b2e-14c1-51ef-7b4c-
6b789be487f2" url="http://172.16.200.44/eicar.com" agent="curl/7.68.0" duration=0 reqlength=86
resplength=392 rcvdbyte=760 sentbyte=163 scheme="http" hostname="172.16.200.44"
resptype="cached" httpmethod="GET"statuscode="403" reqtime=1716347177 resptime=0
respfinishtime=1716347177 appcat="unscanned" utmaction="block" countav=1 utmref=65515-0
```

```
3: date=2024-05-21 time=20:06:06 eventtime=1716347166400042072 tz="-0700" logid="0006000026"
type="traffic" subtype="http-transaction" level="notice" vd="vdom1" srcip=10.1.100.11
srcport=42694 dstip=172.16.200.44 dstport=80 sessionid=316483733 transid=50331678
action="accept" policyid=1 policytype="proxy-policy" poluuid="1e1e0b2e-14c1-51ef-7b4c-
6b789be487f2" url="http://172.16.200.44/" agent="curl/7.68.0" duration=0 reqlength=77
resplength=368 rcvdbyte=368 sentbyte=77 scheme="http" hostname="172.16.200.44"
```

```
resptype="normal" httpmethod="GET"statuscode="200" reqtime=1716347166 resptime=1716347166
respfinishtime=1716347166 appcat="unscanned"
```

When the EICAR test file in the response is blocked by utm-av, `utmref` information referring to the corresponding utm-av log is included:

```
# execute log detail 2 "65515-0"
1 logs found.
1 logs returned.
```

```
1: date=2024-05-21 time=20:06:17 eventtime=1716347177536848145 tz="-0700" logid="0211008192"
type="utm" subtype="virus" eventtype="infected" level="warning" vd="vdom1" policyid=1
poluuid="1e1e0b2e-14c1-51ef-7b4c-6b789be487f2" policytype="proxy-policy" msg="File is
infected." action="blocked" service="HTTP" sessionid=316483733 transid=50331679
srcip=10.1.100.11 dstip=172.16.200.44 srcport=42694 dstport=80 srccountry="Reserved"
dstcountry="Reserved" srcintf="port1" srcintfrole="undefined" dstintf="port3"
dstintfrole="undefined" proto=6 direction="incoming" filename="eicar.com"
quarskip="Quarantine-disabled" virus="EICAR_TEST_FILE" viruscat="Virus" dtype="av-engine"
itype="infected" ref="https://fortiguard.com/encyclopedia/virus/2172" virusid=2172
url="http://172.16.200.44/eicar.com" profile="av" agent="curl/7.68.0" httpmethod="GET"
analyticssubmit="false" crscore=50 craction=2 crlevel="critical"
```

- Forward traffic log and http-transaction logs for transparent-web proxy policy, access-proxy proxy policy, and proxy-mode firewall policy:
  - Transparent-web proxy policy:

```
1: date=2024-05-23 time=23:22:36 eventtime=1716531756508124889 tz="-0700"
logid="0000000010" type="traffic" subtype="forward" level="notice" vd="vdom1"
srcip=10.1.100.11 srcport=34326 srcintf="port1" srcintfrole="undefined"
dstcountry="Reserved" srccountry="Reserved" dstip=172.16.200.44 dstport=443
dstintf="port3" dstintfrole="undefined" sessionid=90954 service="HTTPS" proxyapptype="web-
proxy" proto=6 action="accept" policyid=2 policytype="proxy-policy" poluuid="1f986ef4-
14c1-51ef-7d4f-fd482cefde73" trandisp="snat" transip=172.16.200.8 transport=34326
duration=0 wanin=3051 rcvdbyte=3051 wanout=618 lanin=842 sentbyte=842 lanout=4551
appid=40568 app="HTTPS.BROWSER" appcat="Web.Client" apprisk="medium" utmaction="allow"
countssl=2 utmref=65516-0
```

```
2: date=2024-05-23 time=23:22:36 eventtime=1716531756507534974 tz="-0700"
logid="0006000026" type="traffic" subtype="http-transaction" level="notice" vd="vdom1"
srcip=10.1.100.11 srcport=34326 dstip=172.16.200.44 dstport=443 sessionid=90954
transid=50331679 srcuuid="bdba900e-14c0-51ef-1328-c1b8329857ef" dstuuid="bdba900e-14c0-
51ef-1328-c1b8329857ef" action="accept" policyid=2 policytype="proxy-policy"
poluuid="1f986ef4-14c1-51ef-7d4f-fd482cefde73" url="https://172.16.200.44/"
agent="curl/7.68.0" duration=0 reqlength=38 resplength=185 rcvdbyte=3051 sentbyte=787
scheme="https" hostname="172.16.200.44" resptype="normal" httpmethod="GET"
statuscode="200" reqtime=1716531756 resptime=1716531756 respfinishtime=1716531756
appcat="unscanned"
```

  - Access-proxy proxy policy:

```
1: date=2024-05-23 time=23:24:10 eventtime=1716531850437771247 tz="-0700"
logid="0005000024" type="traffic" subtype="ztna" level="notice" vd="vdom1"
srcip=10.1.100.11 srcport=51076 srcintf="port1" srcintfrole="undefined"
dstcountry="Reserved" srccountry="Reserved" dstip=172.16.200.44 dstport=443
dstintf="port3" dstintfrole="undefined" sessionid=91243 service="HTTPS"
proxyapptype="http" proto=6 action="accept" policyid=3 policytype="proxy-policy"
poluuid="2853b1ba-195e-51ef-2e7c-93b136b9505a" duration=0 gatewayid=1 realserverid=1
vip="vip-ztna" accessproxy="ztna" clientdevicemanageable="unknown" clientcert="no"
wanin=2981 rcvdbyte=2981 wanout=712 lanin=824 sentbyte=824 lanout=2000 appcat="unscanned"
```

```
2: date=2024-05-23 time=23:24:10 eventtime=1716531850436959338 tz="-0700"
logid="0006000026" type="traffic" subtype="http-transaction" level="notice" vd="vdom1"
srcip=10.1.100.11 srcport=51076 dstip=172.16.200.44 dstport=443 sessionid=91243
transid=50331680 srcuuid="bdba900e-14c0-51ef-1328-c1b8329857ef" action="accept" policyid=3
policytype="proxy-policy" poluuid="2853b1ba-195e-51ef-2e7c-93b136b9505a"
url="https://a.ftnt.com/" agent="curl/7.68.0" duration=0 reqlength=36 resplength=185
rcvdbyte=2981 sentbyte=800 scheme="https" hostname="a.ftnt.com" resptype="normal"
httpmethod="GET" statuscode="200" reqtime=1716531850 resptime=1716531850
respfinishtime=1716531850 appcat="unscanned"
```

- Proxy-mode firewall policy:

```
1: date=2024-05-23 time=23:32:11 eventtime=1716532331148593220 tz="-0700"
logid="0000000013" type="traffic" subtype="forward" level="notice" vd="vdom1"
srcip=10.1.100.11 srcport=49424 srcintf="port1" srcintfrole="undefined"
dstip=172.16.200.44 dstport=443 dstintf="port3" dstintfrole="undefined"
srccountry="Reserved" dstcountry="Reserved" sessionid=92757 proto=6 action="close"
policyid=1 policytype="policy" poluuid="1ab9cbbc-14c1-51ef-55cd-4b90ff9a117a"
service="HTTPS" trandisp="snat" transip=172.16.200.8 transport=49424 duration=1
sentbyte=1406 rcvdbyte=5235 sentpkt=15 rcvdpkt=13 appcat="unscanned" wanin=3051 wanout=618
lanin=842 lanout=4551 utmaction="allow" countssl=2 utmref=65511-0
```

```
2: date=2024-05-23 time=23:32:10 eventtime=1716532330081764186 tz="-0700"
logid="0006000026" type="traffic" subtype="http-transaction" level="notice" vd="vdom1"
srcip=10.1.100.11 srcport=49424 dstip=172.16.200.44 dstport=443 sessionid=92757
transid=50331681 srcuuid="bdba900e-14c0-51ef-1328-c1b8329857ef" dstuuid="bdba900e-14c0-
51ef-1328-c1b8329857ef" action="pending" policyid=1 policytype="policy" poluuid="1ab9cbbc-
14c1-51ef-55cd-4b90ff9a117a" url="https://172.16.200.44/" duration=0 reqlength=38
resplength=185 rcvdbyte=3051 sentbyte=787 scheme="https" hostname="172.16.200.44"
resptype="normal" reqtime=1716532330 resptime=1716532330 respfinishtime=1716532330
appcat="unscanned"
```

For HTTPS with explicit-web proxy, there is an additional piece of http-transaction log for each CONNECT request and response:

```
3: date=2024-05-21 time=20:34:44 eventtime=1716348884524284243 tz="-0700" logid="0006000026"
type="traffic" subtype="http-transaction" level="notice" vd="vdom1" srcip=10.1.100.11
srcport=46030 dstip=172.16.200.44 dstport=443 sessionid=316483736 transid=50331683 action="accept"
policyid=1 policytype="proxy-policy" poluuid="1e1e0b2e-14c1-51ef-7b4c-6b789be487f2"
url="https://172.16.200.44/" agent="curl/7.68.0" duration=0 reqlength=118 resplength=0 rcvdbyte=0
sentbyte=118 scheme="https" hostname="172.16.200.44" resptype="generated" httpmethod="CONNECT"
statuscode="200" reqtime=1716348884 resptime=0 respfinishtime=1716348884 appcat="unscanned"
```

For HTTPS with certificate-inspection or no inspection, there is only one http-transaction log for each TCP connection because the encrypted HTTP messages are not decrypted:

- Firewall policy:

```
2: date=2024-05-23 time=21:38:56 eventtime=1716525535340969183 tz="-0700" logid="0006000026"
type="traffic" subtype="http-transaction" level="notice" vd="vdom1" srcip=10.1.100.11
srcport=46462 dstip=172.16.200.44 dstport=443 sessionid=70593 transid=1 srcuuid="bdba900e-
14c0-51ef-1328-c1b8329857ef" dstuuid="bdba900e-14c0-51ef-1328-c1b8329857ef" policyid=1
policytype="policy" poluuid="1ab9cbbc-14c1-51ef-55cd-4b90ff9a117a" url="172.16.200.44"
duration=0 reqlength=842 resplength=3100 rcvdbyte=3100 sentbyte=842 scheme="https"
hostname="172.16.200.44" resptype="N/A" reqtime=1716525535 resptime=1716525535
respfinishtime=1716525535 appcat="unscanned"
```

- Explicit-web proxy policy:

```
2: date=2024-05-23 time=21:36:18 eventtime=1716525378802239534 tz="-0700" logid="0006000026"
type="traffic" subtype="http-transaction" level="notice" vd="vdom1" srcip=10.1.100.11
srcport=56986 dstip=172.16.200.44 dstport=443 sessionid=1369348242 transid=50331673
action="accept" policyid=1 policytype="proxy-policy" poluuid="1e1e0b2e-14c1-51ef-7b4c-
6b789be487f2" url="https://172.16.200.44/" agent="curl/7.68.0" duration=0 reqlength=118
resplength=0 rcvdbyte=0 sentbyte=118 scheme="https" hostname="172.16.200.44"
resptype="generated" httpmethod="CONNECT" statuscode="200" reqtime=1716525378 resptime=0
respfinishtime=1716525378 appcat="unscanned"
```

For SOCKS proxy, there is one http-transaction log for each HTTP transaction per TCP connection:

```
1: date=2024-05-23 time=22:50:34 eventtime=1716529833463518327 tz="-0700" logid="0006000026"
type="traffic" subtype="http-transaction" level="notice" vd="vdom1" srcip=10.1.100.143
srcport=63744 dstip=34.107.221.82 dstport=80 sessionid=1369348358 transid=117441403
srcuuid="bdba900e-14c0-51ef-1328-c1b8329857ef" dstuuid="bdba900e-14c0-51ef-1328-c1b8329857ef"
action="pending" policyid=1 policytype="proxy-policy" poluuid="1e1e0b2e-14c1-51ef-7b4c-
6b789be487f2" url="http://detectportal.firefox.com/success.txt?ipv4" agent="Mozilla/5.0 (Windows
NT 10.0; Win64; x64; rv:125.0) Gecko/20100101 Firefox/125.0" duration=0 reqlength=305
resplength=216 rcvdbyte=7128 sentbyte=20143 scheme="http" hostname="detectportal.firefox.com"
resptype="normal" httpmethod="GET" statuscode="200" reqtime=1716529833 resptime=1716529833
respfinishtime=1716529833 appcat="unscanned"
```

```
2: date=2024-05-23 time=22:49:40 eventtime=1716529779381260298 tz="-0700" logid="0000000010"
type="traffic" subtype="forward" level="notice" vd="vdom1" srcip=10.1.100.143 srcport=63842
srcintf="port1" srcintfrole="undefined" dstcountry="Reserved" srccountry="Reserved"
dstip=172.16.200.44 dstport=80 dstintf="port3" dstintfrole="undefined" sessionid=1369348430
service="webproxy" proxyapptype="web-proxy" proto=6 action="accept" policyid=1 policytype="proxy-
policy" poluuid="1e1e0b2e-14c1-51ef-7b4c-6b789be487f2" trandisp="snat" transip=172.16.200.8
transport=16135 duration=0 wanin=392 rcvdbyte=392 wanout=356 lanin=725 sentbyte=725 lanout=71832
appcat="unscanned" utmaction="block" countav=1 utmref=65517-14
```

```
3: date=2024-05-23 time=22:49:40 eventtime=1716529779381204800 tz="-0700" logid="0006000026"
type="traffic" subtype="http-transaction" level="notice" vd="vdom1" srcip=10.1.100.143
srcport=63842 dstip=172.16.200.44 dstport=80 sessionid=1369348430 transid=117441401
srcuuid="bdba900e-14c0-51ef-1328-c1b8329857ef" dstuuid="bdba900e-14c0-51ef-1328-c1b8329857ef"
action="pending" policyid=1 policytype="proxy-policy" poluuid="1e1e0b2e-14c1-51ef-7b4c-
6b789be487f2" url="http://172.16.200.44/eicar.com" agent="Mozilla/5.0 (Windows NT 10.0; Win64;
```

```
x64; rv:125.0) Gecko/20100101 Firefox/125.0" duration=0 reqlength=356 resplength=392 rcvdbyte=392
sentbyte=725 scheme="http" hostname="172.16.200.44" resptype="cached" httpmethod="GET"
statuscode="403" reqtime=1716529779 resptime=0 respfinishtime=1716529779 appcat="unscanned"
utmaction="block" countav=1 utmref=65517-0
```

UTM logs that do not belong to an HTTP transaction are only associated with the forward traffic log, and not the http-transaction log:

Only the forward traffic log is associated to the utm-ssl log by the `utmref`.

- forward traffic log:

```
1: date=2024-05-23 time=22:01:43 eventtime=1716526903039789335 tz="-0700" logid="0000000010"
type="traffic" subtype="forward" level="notice" vd="vdom1" srcip=10.1.100.11 srcport=57928
srcintf="port1" srcintfrole="undefined" dstcountry="Reserved" srccountry="Reserved"
dstip=172.16.200.44 dstport=443 dstintf="port3" dstintfrole="undefined" sessionid=1369348243
service="HTTPS" proxyapptype="web-proxy" proto=6 action="accept" policyid=1 policytype="proxy-
policy" poluuid="1e1e0b2e-14c1-51ef-7b4c-6b789be487f2" trandisp="snat" transip=172.16.200.8
transport=1837 duration=0 wanin=3051 rcvdbyte=3051 wanout=618 lanin=960 sentbyte=960
lanout=4623 appcat="unscanned" utmaction="allow" countssl=2 utmref=65523-0
```

- http-transaction logs:

```
2: date=2024-05-23 time=22:01:43 eventtime=1716526903038941661 tz="-0700" logid="0006000026"
type="traffic" subtype="http-transaction" level="notice" vd="vdom1" srcip=10.1.100.11
srcport=57928 dstip=172.16.200.44 dstport=443 sessionid=1369348243 transid=50331675
action="accept" policyid=1 policytype="proxy-policy" poluuid="1e1e0b2e-14c1-51ef-7b4c-
6b789be487f2" url="https://172.16.200.44/" agent="curl/7.68.0" duration=0 reqlength=38
resplength=185 rcvdbyte=3051 sentbyte=936 scheme="https" hostname="172.16.200.44"
resptype="normal" httpmethod="GET" statuscode="200" reqtime=1716526903 resptime=1716526903
respfinishtime=1716526903 appcat="unscanned"
```

```
3: date=2024-05-23 time=22:01:43 eventtime=1716526903015933382 tz="-0700" logid="0006000026"
type="traffic" subtype="http-transaction" level="notice" vd="vdom1" srcip=10.1.100.11
srcport=57928 dstip=172.16.200.44 dstport=443 sessionid=1369348243 transid=50331674
action="accept" policyid=1 policytype="proxy-policy" poluuid="1e1e0b2e-14c1-51ef-7b4c-
6b789be487f2" url="https://172.16.200.44/" agent="curl/7.68.0" duration=0 reqlength=118
resplength=0 rcvdbyte=0 sentbyte=118 scheme="https" hostname="172.16.200.44"
resptype="generated" httpmethod="CONNECT" statuscode="200" reqtime=1716526903 resptime=0
respfinishtime=1716526903 appcat="unscanned"
```

Configuring fast fallback for explicit proxy

The fast fallback (also named "Happy Eyeballs") algorithm, as outlined in RFC 8305, is supported for explicit web proxy. This feature operates by attempting to connect to a web server that is available at multiple IPv4 and IPv6 addresses, either sequentially or simultaneously. As a result, the web server can be connected with reduced user-visible delay, which enhances the overall browsing experience.

```
config web-proxy fast-fallback
    edit <name>
        set status {enable | disable}
```

```
        set connection-mode {sequentially | simultaneously}
        set protocol {IPv4-first | IPv6-first | IPv4-only | IPv6-only}
        set connection-timeout <integer>
    next
end
```

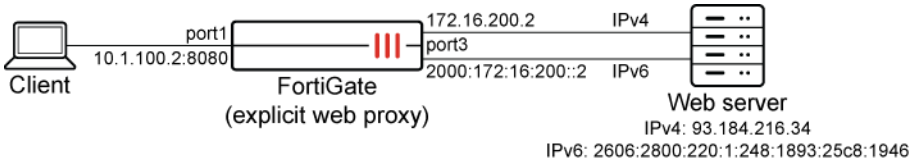
`status {enable | disable}`	Enable/disable the fast fallback entry (default = `enable`).
`connection-mode {sequentially | simultaneously}`	Set the connection mode for multiple destinations. • `sequentially`: connect the different destinations sequentially (default). • `simultaneously`: connect the different destinations simultaneously.
`protocol {IPv4-first | IPv6-first | IPv4-only | IPv6-only}`	Set the connection protocols for multiple destinations. • `IPv4-first`: connect IPv4 destinations first (default). • `IPv6-first`: connect IPv6 destinations first. • `IPv4-only`: connect IPv4 destinations only. • `IPv6-only`: connect IPv6 destinations only.
`connection-timeout <integer>`	Start another connection if a connection takes longer than the timeout value, in milliseconds (200 - 1800000, default = 200).

Based on the settings for `connection-mode` and `protocol`, the explicit web proxy will try connecting to the web server in different ways:

- If the `connection-mode` is set to `sequential` (default), then the explicit web proxy will try connecting to the web server by IPv4 first, or by IPv6 first depending on the `protocol` setting. If the connection attempt over IPv4 or IPv6 succeeds, then the connection is kept; but if the connection fails, then it falls back to try a connection over IPv6 or IPv4 instead.
- If the `connection-mode` is set to `simultaneously`, then the explicit web proxy will try connecting to the web server by IPv4 and IPv6 at the same time. If the connection over IPv4 is established first, then the connection is kept for the session and the IPv6 connection is discarded and vice-versa.
- If the user only wants to connect by IPv4 but not IPv6, or by IPv6 but not IPv4, then the `protocol` option can be set to `IPv4-only` or `IPv6-only` accordingly. The explicit web proxy will try connecting to the web server only by IPv4 or IPv6, even though both IPv4 and IPv6 may work.

Example

In this example, a client visits a web server through a FortiGate explicit web proxy that has IPv4 and IPv6 connections to the web server (www.example.com), which can resolve to IPv4 address 93.184.216.34 and IPv6 address 2606:2800:220:1:248:1893:25c8:1946.



The configuration uses sequential connection mode, the IPv4 first protocol, and the default connection timeout (200 ms).

To configure the FortiGate:

1. Configure the IPv4 static route:

```
config router static
    edit 1
        set gateway 172.16.200.251
        set device "port3"
    next
end
```

2. Configure the IPv6 static route:

```
config router static6
    edit 1
        set gateway 2000:172:16:200::254
        set device "port3"
    next
end
```

3. Configure the proxy destination connection fast fallback:

```
config web-proxy fast-fallback
    edit "ffbk"
        set status enable
        set connection-mode sequentially
        set protocol IPv4-first
        set connection-timeout 200
    next
end
```

4. Configure the exempt URL of the web server from web proxy forwarding and caching:

```
config web-proxy url-match
    edit "ffbk"
        set url-pattern "example.com"
        set fast-fallback "ffbk"
    next
end
```

5. Configure the proxy policy:

```
config firewall proxy-policy
    edit 1
        set proxy explicit-web
        set dstintf "port3"
        set srcaddr "all"
        set dstaddr "all"
        set service "webproxy"
        set action accept
        set schedule "always"
        set logtraffic all
        set srcaddr6 "all"
```

```
        set dstaddr6 "all"
        set utm-status enable
        set ssl-ssh-profile "deep-custom"
        set av-profile "av"
    next
end
```

Verifying the connection

Scenario 1:

The TCP connection from the explicit web proxy to the web server is established successfully over IPv4 within 200 ms.

As shown in the forward traffic log, the web session data is transmitted over IPv4 between the explicit web proxy and the web server.

```
2: date=2023-06-26 time=18:46:18 eventtime=1687830378260927765 tz="-0700" logid="0000000010"
type="traffic" subtype="forward" level="notice" vd="vdom1" srcip=10.1.100.11 srcport=33304
srcintf="port1" srcintfrole="undefined" dstcountry="United States" srccountry="Reserved"
dstip=93.184.216.34 dstport=80 dstintf="port3" dstintfrole="undefined" sessionid=1688881487
service="HTTP" proxyapptype="web-proxy" proto=6 action="accept" policyid=1 policytype="proxy-
policy" poluuid="560d8520-fa7b-51ed-e06a-df05ec145542" trandisp="snat" transip=0.0.0.0 transport=0
duration=0 wanin=0 rcvdbyte=0 wanout=0 lanin=131 sentbyte=131 lanout=1591 appcat="unscanned"
```

Scenario 2:

The TCP connection from the explicit web proxy to the web server is not established over IPv4 within 200 ms and falls back to IPv6 successfully.

The IPv4 path to the server is interrupted, and the TCP connection between the explicit web proxy and web server cannot be established. The explicit web proxy waits until the 200 ms connection timeout timer expires, then attempts to connect to the server by IPv6, which is successful. The web session data is transmitted over IPv6, as shown in the forward traffic log.

```
2: date=2023-06-26 time=18:47:27 eventtime=1687830447277653089 tz="-0700" logid="0000000010"
type="traffic" subtype="forward" level="notice" vd="vdom1" srcip=10.1.100.11 srcport=36636
srcintf="port1" srcintfrole="undefined" dstcountry="United States" srccountry="Reserved"
dstip=2606:2800:220:1:248:1893:25c8:1946 dstport=80 dstintf="port3" dstintfrole="undefined"
sessionid=1688881488 service="HTTP" proxyapptype="web-proxy" proto=6 action="accept" policyid=1
policytype="proxy-policy" poluuid="560d8520-fa7b-51ed-e06a-df05ec145542" trandisp="snat"
transport=0 duration=1 wanin=0 rcvdbyte=0 wanout=0 lanin=131 sentbyte=131 lanout=1591
appcat="unscanned"
```

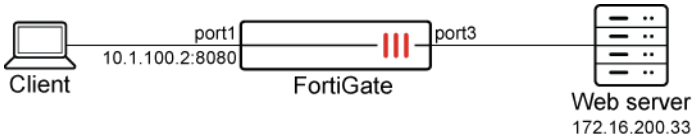
Forward HTTPS requests to a web server without the need for an HTTP CONNECT message

An explicit web proxy can forward HTTPS requests to a web server without the need for an HTTP CONNECT message. The FortiGate explicit web proxy can be configured to detect the HTTPS scheme in the request line of a plain text HTTP request and forward it as an HTTPS request to the web server. This allows applications that cannot use the CONNECT message for sending an HTTPS request to communicate with the web server through an explicit web proxy.

```
config firewall proxy-policy
    edit <id>
        set detect-https-in-http-request {enable | disable}
    next
end
```

Example

Based on the following topology, an HTTPS request is sent to a web server through an explicit web proxy.



To enable detection of HTTPS in an HTTP request:

1. Configure the explicit web proxy:

```
config web-proxy explicit
    set status enable
    set ftp-over-http enable
    set socks enable
    set http-incoming-port 8080
    set ipv6-status enable
    set unknown-http-version best-effort
end
```

2. Enable the explicit web proxy on port1:

```
config system interface
    edit "port1"
        set ip 10.1.100.2 255.255.255.0
        set explicit-web-proxy enable
    next
end
```

3. Configure the proxy policy:

```
config firewall proxy-policy
    edit 1
        set proxy explicit-web
        set dstintf "port3"
        set srcaddr "all"
        set dstaddr "all"
        set service "webproxy"
        set action accept
        set schedule "always"
        set logtraffic all
        set utm-status enable
        set ssl-ssh-profile "deep-inspection"
        set av-profile "av"
        set detect-https-in-http-request enable
    next
end
```

An SSL-SSH profile with deep inspection must be applied in order to decrypt the server response in HTTPS and forward the response to the client by HTTP.

4. Using Telnet, send an HTTP request with an HTTPS scheme as follows:

```
telnet 10.1.100.2 8080
Trying 10.1.100.2...
Connected to 10.1.100.2.
Escape character is '^]'.
POST https://172.16.200.33/ HTTP/1.1
Host: 172.16.200.33
User-Agent: curl/7.68.0
Accept: */*
Content-Type: application/x-www-form-urlencoded
Content-Length: 0

HTTP/1.1 200 OK
```

5. Verify the traffic log. The HTTP request is forwarded to the server successfully by HTTPS:

```
# execute log filter category 3
...
2: date=2023-07-31 time=16:02:22 eventtime=1690844541296891542 tz="-0700" logid="0000000010"
type="traffic" subtype="forward" level="notice" vd="vdom1" srcip=10.1.100.11 srcport=46074
srcintf="port1" srcintfrole="undefined" dstcountry="Reserved" srccountry="Reserved"
dstip=172.16.200.33 dstport=443 dstintf="port3" dstintfrole="undefined" sessionid=1799884153
service="HTTPS" proxyapptype="web-proxy" proto=6 action="accept" policyid=1 policytype="proxy-
policy" poluuid="73379360-2d21-51ee-77d8-154efc517a6a" trandisp="snat" transip=172.16.200.2
transport=2713 duration=4 wanin=3053 rcvdbyte=3053 wanout=757 lanin=169 sentbyte=169
lanout=279 appcat="unscanned"
```

Specifying outgoing interface and VRF for a web proxy forward server or isolator server

You can specify the outgoing interface and VRF for a web proxy forward server or a web proxy isolator server, such as FortiIsolator. See Virtual routing and forwarding on page 791.

The following CLI command options can be configured:

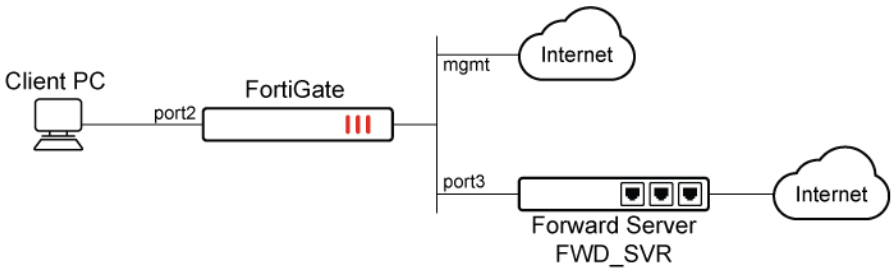
```
config web-proxy {forward-server | isolator-server}
   edit <name>
    set interface-select-method specify
    set interface <port>
    set masquerade {enable | disable}
    set vrf-select <vrf-id>
  next
end
```

Example

In the following example, a forward server is applied to the FortiGate in an explicit proxy policy. An interface that is not in the policy, such as port3, can be specified to forward traffic.

> Without this feature, the FortiGate would have to forward traffic through the management interface which is the destination interface of the policy.

This example uses the following topology:



To specify an outgoing interface and VRF for a web proxy forward server:

1. Enable and configure the explicit web proxy:

```
config web-proxy explicit
    set status enable
    set ftp-over-http enable
    set http-incoming-port 8080
    set ipv6-status enable
    set unknown-http-version best-effort
end
```

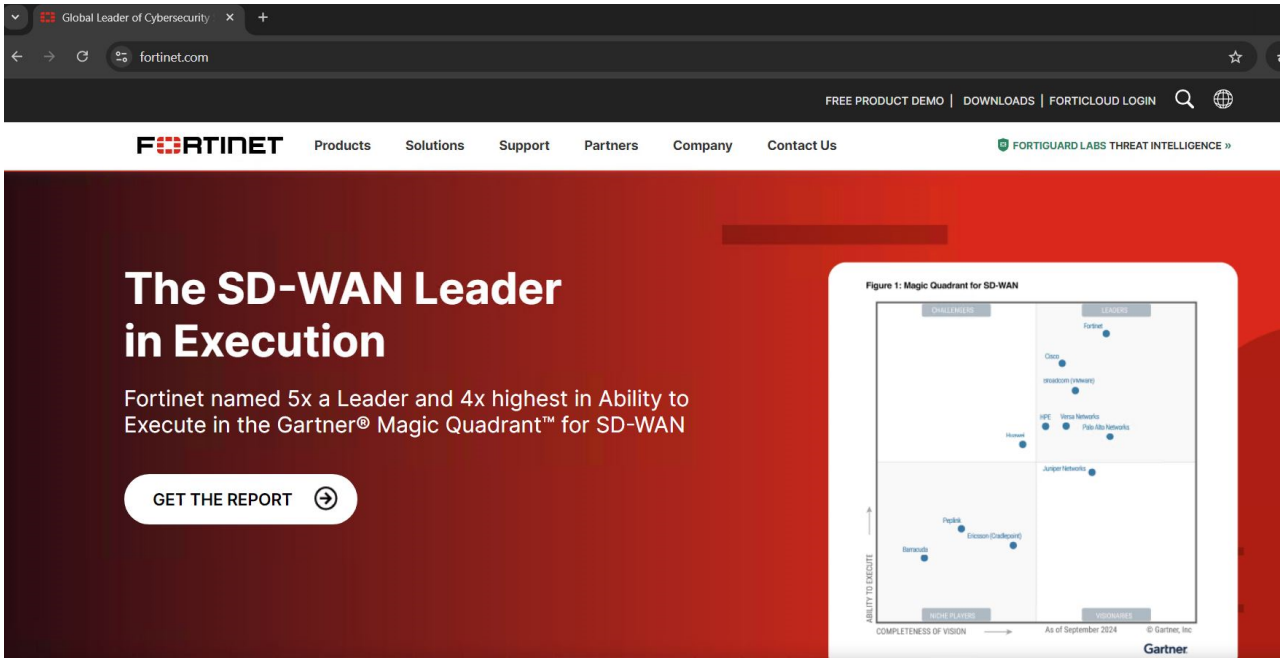
2. Configure the web proxy forward server with an interface that is not included in the policy:

```
config web-proxy forward-server
    edit "FWD_SVR"
        set ip 172.16.200.7
        set port 8080
        set interface-select-method specify
        set interface "port3"
        set vrf-select 10
    next
end
```

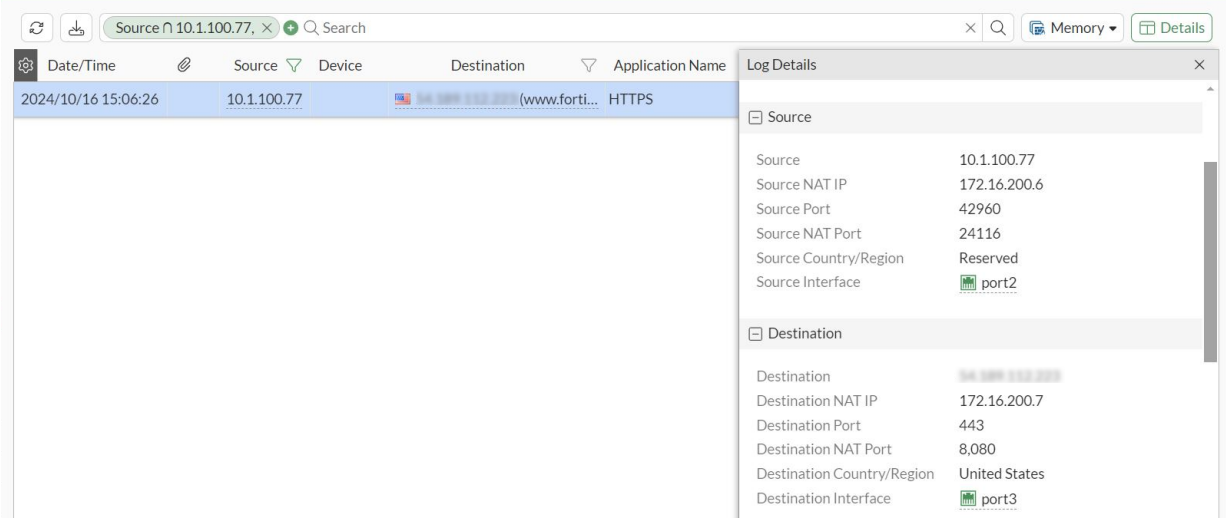
3. Specify the destination interface and web proxy forward server in the proxy policy:

```
config firewall proxy-policy
    edit 1
        set proxy explicit-web
        set dstintf "mgmt"
        set srcaddr "all"
        set dstaddr "all"
        set service "webproxy"
        set action accept
        set schedule "always"
        set logtraffic all
        set webproxy-forward-server "FWD_SVR"
    next
end
```

4. Access a website in the browser, such as www.fortinet.com.



5. Go to *Log & Report* > *Forward Traffic* and review the traffic log. The *Destination Interface* is *port3* instead of the management interface.



Isolator servers in proxy policies

Web proxy isolator servers, such as FortiIsolator, are supported in proxy policies. Isolators are fundamentally the same as web proxy forward servers because both will redirect HTTP and HTTPS requests to an HTTP or HTTPS proxy server. However, isolators have the specific function of isolating potentially unsafe traffic from a user environment.

The `isolate` action in proxy policies can be used to distinguish isolated traffic from normal traffic in logs. Isolator servers can only be applied in explicit and transparent proxy policies. See Explicit web proxy on page 406 and Transparent proxy on page 414.

To configure an isolator server in the CLI:

1. Configure the isolator server:

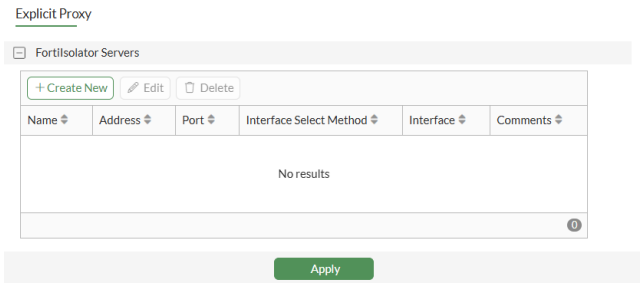
```
config web-proxy isolator-server
    edit <name>
        set addr-type {ip | ipv6 | fqdn}
        set ip <any_ip>
        set ipv6 <IPv6 address>
        set fqdn <string>
        set port <port>
    next
end
```

2. Apply the server to a proxy policy:

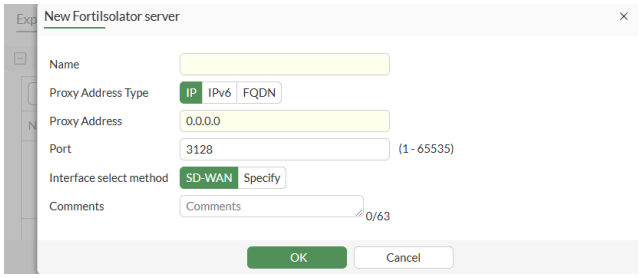
```
config firewall proxy-policy
    edit <id>
        set action isolate
        set isolator-server <name>
    next
end
```

To configure an isolator server in the GUI:

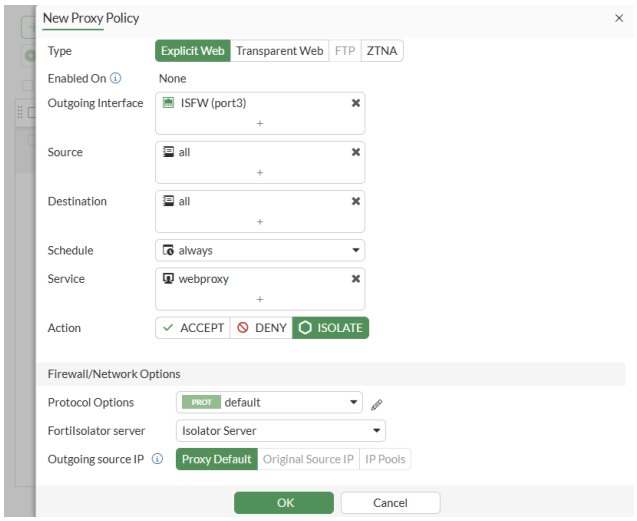
1. Go to *Network* > *Explicit Proxy*.
2. Enable *Explicit Web Proxy* and scroll to *FortiIsolator Servers*.



3. Click *Create New*.



4. Configure the isolator server.
5. Click *OK*.
6. Configure the other explicit proxy fields, as needed, and click *Apply*.
7. Apply the isolator server to a proxy policy:
   a. Go to *Policy & Objects* > *Proxy Policy*.
   b. Click *Create New*.
   c. Set the *Type* to *Explicit Web* or *Transparent Web*.
   d. Set *Action* to *Isolate*.
   e. Set *FortiIsolator server* to the new server you configured.
   f. Configure other proxy policy fields, as needed.
   g. Click *OK*.



Example

The following example demonstrates how to apply an isolator server to a proxy policy. Two explicit proxy policies are configured:

- A web proxy forward server is applied to one policy with the action set to accept.
- An isolator server is applied to the other policy with the action set to isolate.

Each proxy policy uses a different destination IP address to separate traffic. Once traffic passes, logs are generated for the specific actions.

To configure an isolator server in the CLI:

1. Configure the isolator server:

```
config web-proxy isolator-server
    edit "isolator"
        set ip 172.16.200.7
        set port 8080
    next
end
```

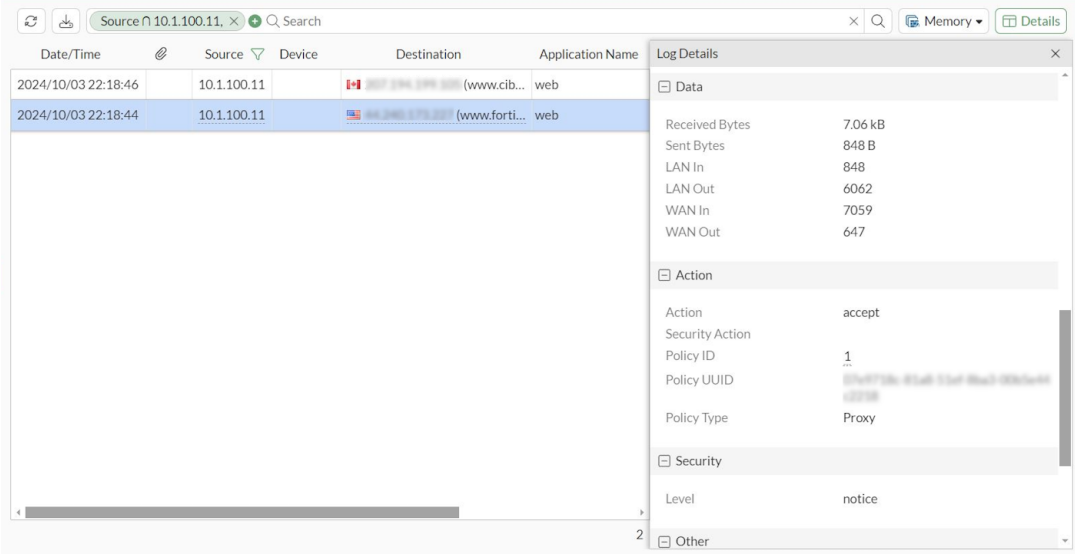
2. Configure the forward server:

```
config web-proxy forward-server
    edit "fgt-b"
        set ip 172.16.200.7
        set port 8080
    next
end
```

3. Apply each server to a proxy policy:

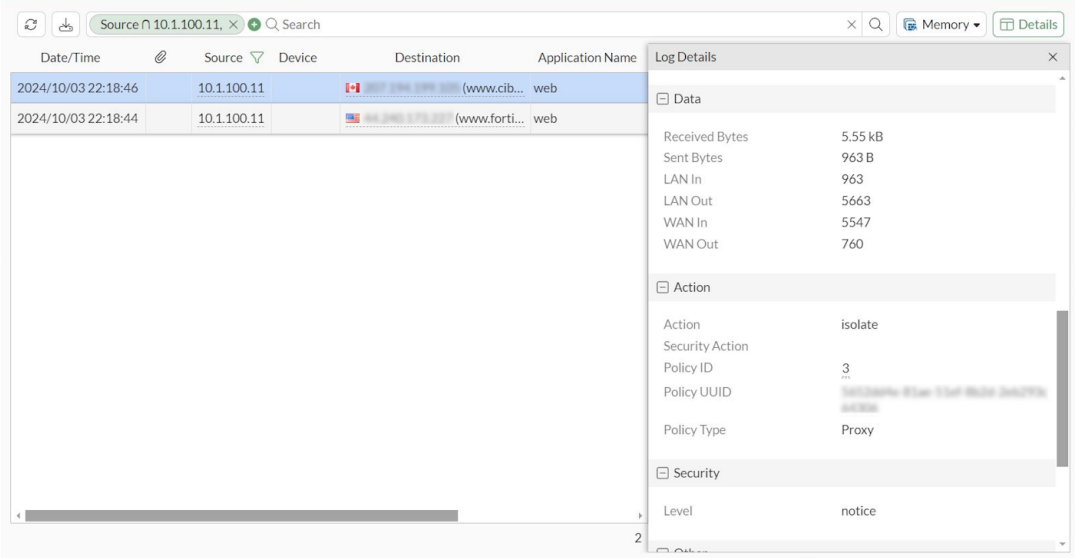
```
config firewall proxy-policy
    edit 1
        set proxy explicit-web
        set dstintf "port3"
        set srcaddr "all"
        set dstaddr "IT"
        set service "webproxy"
        set action accept
        set schedule "always"
        set logtraffic all
        set webproxy-forward-server "fgt-b"
        set utm-status enable
        set ssl-ssh-profile "deep-custom"
        set av-profile "av"
    next
    edit 3
        set proxy explicit-web
        set dstintf "port3"
        set srcaddr "all"
        set dstaddr "Finance"
        set service "webproxy"
        set action isolate
        set schedule "always"
        set logtraffic all
        set isolator-server "isolator"
        set utm-status enable
        set ssl-ssh-profile "deep-custom"
        set av-profile "av"
    next
end
```

4. Generate traffic for the proxy policies and go to *Log & Report > Forward Traffic* in the GUI to review the logs:
   a. When accessing www.fortinet.com, the traffic hits proxy policy 1 and is accepted.
      Since the traffic matches the destination address, it goes to the forward server and then to the internet. A traffic log is generated with the action set to *accept*.



b. When accessing www.cibc.com, the traffic hits proxy policy 3 and is isolated.

Since the traffic matches the destination address, it goes to the isolator server and then to the internet. A traffic log is generated with the action set to *isolate*.



Secure explicit proxy with client certificate blocklist enforcement NEW

This section describes how to configure a secure explicit web proxy that enforces the FortiGate client certificate blocklist during the TLS handshake, ensuring that client certificates listed in the Malicious Certificate Database (MCDB) are denied access.

This behavior is supported in FortiOS 8.0.0 and later.

Overview

Secure explicit proxy with client certificates allows FortiGate to authenticate proxy clients using X.509 certificates. When client certificate blocklist, enforcement is enabled, FortiGate checks the presented client certificate against the MCDB during the TLS handshake.

If the client certificate is listed in the MCDB:

- The TLS handshake is aborted
- The proxy connection is denied
- A log entry is generated

This prevents malicious certificates from accessing web resources through the proxy.

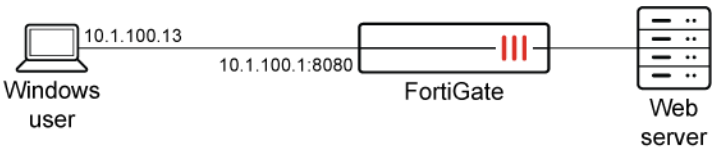
The config web-proxy explicit command includes a new option:

```
config web-proxy explicit
    set client-certificate-blocklist {enable | disable}
end
```

Option	Description
`client-certificate-blocklist {enable | disable}`	Enable/disable blocking client malicious certificates list by FortiGuard during TLS (default = enabled). • `enable`: Enable client certificate blocklist during TLS handshake. • `disable`: Disable client certificate blocklist during TLS handshake.

Example

The following example demonstrates client certificate blocklist enforcement using a secure explicit proxy.



Prerequisite:

An active FortiGuard subscription is required to use the Malicious Certificate Database (MCDB). The MCDB is used by FortiGate's SSL/TLS inspection and certificate reputation checks to detect and block SSL/TLS sessions that present malicious certificates.

To configure client certificate and blocklist enforcement with explicit proxies:

1. Prepare the certificate:
   a. Use a CA to sign the client certificate.
   b. Import the root CA certificate that signed the client certificate to FortiGate. In this scenario, the certificate is `root_ca`.
   c. Install the client certificate on an endpoint.

2. Configure the explicit web-proxy policy to request the client certificate from the endpoint.

```
config web-proxy explicit
    set status enable
    set secure-web-proxy secure
    set http-incoming-port 8080
    set secure-web-proxy-cert "proxyserver"
    set client-cert enable
    set client-certificate-blocklist enable
end
```

3. Configure verification of the client certificate with the root CA.

```
config authentication setting
    set user-cert-ca "root_ca"
end
```

4. Configure explicit proxy policy.

```
config firewall proxy-policy
    edit 1
        set proxy explicit-web
        set srcaddr "all"
        set dstaddr "all"
        set service "web"
        set action accept
        set schedule "always"
        set logtraffic all
        set utm-status enable
        set ssl-ssh-profile "deep-custom"
    next
end
```

Verification

On Client PC, initiate a connection with blocklisted client certificate:

```
# curl -v -k --proxy-insecure --proxy-cert /root/CA/client_blocklist.p12:123456 --proxy-cert-type
P12 -x https://10.1.100.1:8080 http://172.16.200.99
```

TLS handshake fails, connection is denied, and following log is generated on the FortiGate:

```
1: date=2026-04-02 time=17:33:31 eventtime=1775176410797092890 tz="-0700" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="vdom1" srcip=10.1.100.13 srcport=53684
srcintf="port2" srcintfrole="undefined" dstip=10.1.100.1 dstport=8080 dstintf="unknown-0"
dstintfrole="undefined" srccountry="Reserved" dstcountry="Reserved" sessionid=268435461 proto=6
action="deny" policyid=0 policytype="policy" service="tcp/8080" trandisp="noop" appcat="unscanned"
duration=0 sentbyte=1965 rcvdbyte=0 sentpkt=0 rcvdpkt=0 crscore=30 craction=131072 crlevel="high"
msg="Traffic denied because of client certificate blocklisted"
```

Session and user limits

Web browsers and web servers open and close multiple sessions with the explicit web proxy. Some sessions are short-lived, while others may persist for longer periods due to HTTP persistent connection behavior. Sessions can remain on the explicit web proxy session list after a user has stopped using the proxy (for example, after closing their browser). If an explicit web proxy session is idle for more than 3600 seconds (one hour) it is torn down by the explicit web proxy. Modern HTTP versions (HTTP/1.1, HTTP/2, HTTP/3) use connection persistence mechanisms to improve performance, but the proxy enforces its own session lifecycle independent of the client

This section describes proxy sessions and user limits for both the explicit web proxy and the explicit FTP proxy. Session and user limits for the two proxies are counted and calculated together. However, in most cases if both proxies are active there will be many more web proxy sessions than FTP proxy sessions.

FortiOS adds two sessions to the session table for every explicit proxy session that is initiated by a client:

- One for the connection from the client to the explicit proxy, typically over a configured proxy port using HTTP or HTTPS.
- The other for the proxied connection from FortiOS to the destination server, which may use HTTP, HTTPS (TLS-encrypted), or other application protocols, depending on the request.

These outbound sessions use the FortiOS interface IP as the source address and connect to the destination service using its negotiated port and protocol, rather than assuming fixed ports.

FortiOS limits the number of explicit proxy users, including both explicit FTP proxy and explicit web proxy users. The number of users varies by FortiGate model from 1000 to 128000 for high end models. This limit cannot be raised.

This feature is not supported on FortiGate models with 2 GB RAM or less. See Proxy-related features not supported on FortiGate 2 GB RAM models on page 126 for more information.

If your FortiGate unit is configured for multiple VDOMs, you can go to *System* > *Global Resources* to view the maximum number of *Concurrent explicit proxy users* and, optionally, reduce the limit. The following command can also be used:

```
config global
    config system resource-limits
        set proxy 50
    end
end
```

To limit the number of explicit proxy users for a VDOM, in the GUI enable multiple VDOMs then go to *System* > *VDOM* and edit a VDOM, or use the following command to change the number of explicit web proxy users for VDOM_1:

```
config global
    config system vdom-property
        edit VDOM_1
            set proxy 25
        next
    end
end
```

Use the `diagnose wad user list` command to view the non-anonymous explicit web proxy users. Users may be displayed with this command even if they are no longer actively using the proxy. All idle sessions timeout after 3600 seconds.

Use the `diagnose wad user clear` command to clear currently authenticated explicit proxy users. You can also use the `diagnose wad user clear <user-name>` command to clear individual users. This deletes information about all users and forces them re-authenticate.

Use the `diagnose wad process worker -1 157` command to view the details of each used concurrent proxy connection for every WAD worker. Alternatively, the first output line from the `diagnose wad process worker -1 118` command shows summarized statistics for each WAD worker.

Users that authenticate with explicit web-proxy or ftp-proxy security policies appears in the *Dashboard > Asset and Identities > Firewall Users list* when you switch the view from the default Firewall view to Proxy view.

Determining the number of concurrent explicit proxy users depends on their authentication method:

- Session-based authenticated users: Each authenticated user is counted as a single user. Because multiple users can have the same user name, the proxy attempts to identify users according to their authentication membership (based upon whether they were authenticated using RADIUS, LDAP, FSAE, local database, or similar). If a user of one session has the same name and membership as a user of another session, the explicit proxy assumes that it is one user.
- IP-based authentication, no authentication, or if no web-proxy security policy has been added: The source IP address is used to determine a user. All sessions from a single source address are assumed to be from the same user.

The explicit proxy does not limit the number of active sessions for each user. As a result, the actual explicit proxy session count is usually much higher than the number of explicit web proxy users. If an excessive number of explicit web proxy sessions is compromising system performance, you can limit the amount of users if the FortiGate unit is operating with multiple VDOMs.

IPAM

IP address management (IPAM) provides centralized control over IP address allocation within the network. It streamlines administration by automatically assigning IP addresses to interfaces, detecting and resolving conflicts, and maintaining an organized address structure. This enables easier administration for widely used interfaces in the network and reduces complexity, which usually arises when there are a large number of interfaces to be managed in the network. By using IPAM, network administrators can easily keep track of the various interfaces in their network and ensure that they are properly configured and functioning as intended. This can save time and effort, and helps prevent issues that may arise from misconfigured or improperly managed interfaces.

This section includes information about IPAM configurations and examples:

- IPAM introduction on page 525
- Advanced configurations on page 530
- Configuration examples on page 540
- Diagnostics on page 549

IPAM introduction

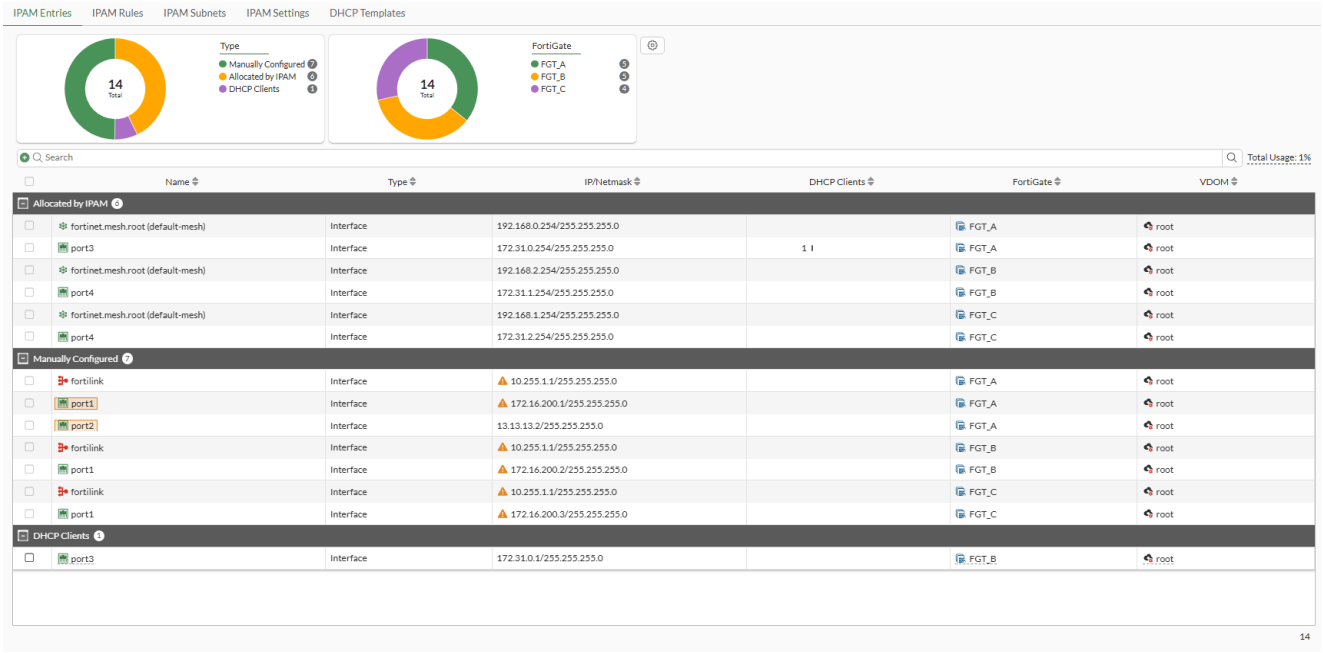
IPAM is available locally on the FortiGate and is disabled by default. When enabled, FortiGate manages IP pools, assign addresses to interfaces, and configure DHCP settings automatically. This reduces manual configuration and ensures efficient use of IP resources. A standalone FortiGate, or a Fabric root in the Security Fabric, can act as the IPAM server.

This section includes information about configuring and using IPAM:

- Configuring IPAM on page 525
- Default IPAM configuration after a factory reset on page 528
- Using IPAM on page 529

Configuring IPAM

IPAM can be configured on the *Network* > *IPAM* page using the *IPAM Settings*, *IPAM Rules*, *IPAM Interfaces*, *IPAM Subnets*, and *DHCP Templates* tabs.



IPAM Interfaces, *IPAM Rules*, and *IPAM Subnets* can only be viewed or configured from the root FortiGate in a Security Fabric or from a standalone FortiGate. Additionally, *IPAM Settings* can only be enabled or disabled from the root FortiGate in a Security Fabric or from a standalone FortiGate.

To configure IPAM settings in the GUI:

1. Go to *Network* > *IPAM* > *IPAM Settings*.
2. Enable or disable the following settings:

- *Status*
- *Auto-resolve conflicts*
- *Require subnet size match*
- *Interfaces with LAN role*
- *FortiAP SSIDs*
- *FortiExtender LAN extensions*

3. Click *OK*.

   Once IPAM is enabled, users should be able to configure additional IPAM settings, such as subnets and rules.

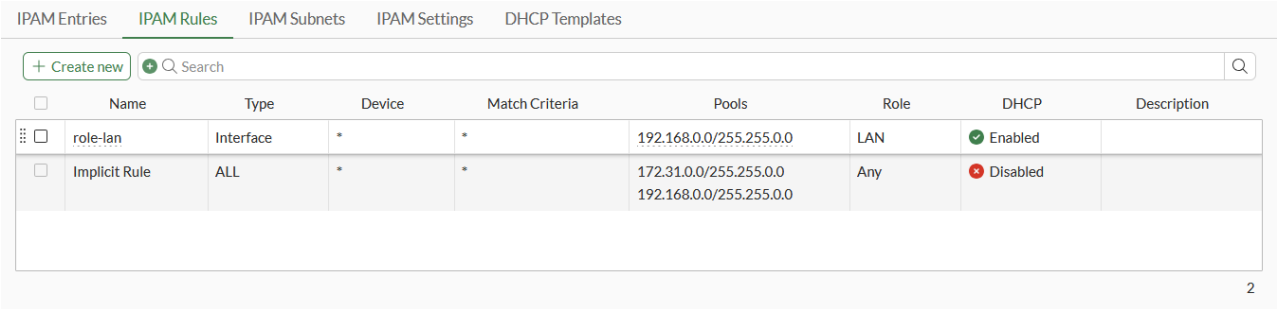
To configure IPAM subnets in the GUI:

1. Go to *Network* > *IPAM* > *IPAM Subnets*.
2. Set *Subnets* to the subnet IP address and netmask. Click the plus field to add more subnets.

   For the IPAM pool subnet, only class A or class B subnet can be configured.

3. Set *Excluded subnets* to the IP address and netmask of subnets that will be excluded. These subnets will not be assigned by private IP assignment. Click the plus field to add more.
4. Click *OK*. A chart is displayed showing free space and IP address overlap between *IPAM-Managed and unallocated*, *Allocated by IPAM*, *Manually Configured*, *DHCP Clients*, and *Reserved* IP addresses.

   Once subnets are added, they are automatically included in the implicit IPAM rule called *Implicit Rule*. This rule encompasses all subnets with the role set to *ANY* and DHCP disabled. The *Implicit Rule* cannot be edited or deleted.



To configure DHCP template in the GUI:

1. Go to *Network* > *IPAM* > *DHCP Templates* and click *Create new*.
2. Configure the new DHCP template. See IPAM DHCP templates New on page 533 for more information.
3. Click *OK*.
4. Configure Security Fabric global object:

```
config system dhcp template
    edit <name>
        set fabric-object {enable | disable}
    next
end
```

Enabling `fabric-object` sets the DHCP template as a Security Fabric–wide global object that is synchronized to downstream FortiGates. Disabling `fabric-object` sets the object as local to the current Security Fabric member.

See config system dhcp template in the CLI Reference for information about the CLI commands.

To configure IPAM rules in the GUI:

1. Go to *Network > IPAM > IPAM Rules*.
2. Click *Create New* and configure the settings:

Option	Description
Name	Set a name.
Description	Set a description.
Apply rule to	Configure whether item is an interface or an address.
FortiGates	Set the serial number or wildcard of a FortiGate to match. Click the plus field to add more FortiGates.
Role	Set the role of the interface to match.
Interfaces	Set the name or wildcard of an interface to match. Click the plus field to add more interfaces.
DHCP	Enabling this option will enable DHCP server on the matching interfaces. Disabling this option will ignore the current DHCP server settings for the interfaces.
Pools	Click the plus field to add one or more subnets.
DHCP template	Enable to use DHCP template with IPAM. Interfaces configured with IPAM inherit DHCP settings from the template. When disabled, interfaces do not inherit DHCP settings from the DHCP template.

3. Click *OK*.

To configure IPAM settings in the CLI:

```
config system ipam
    set status {enable | disable}
    set automatic-conflict-resolution {enable | disable}
    set manage-lan-addresses {enable | disable}
    set manage-lan-extension-addresses {enable | disable}
    set manage-ssid-addresses {enable | disable}
    set require-subnet-match {enable | disable}
    config pools
        edit <pool_name>
            set subnet <IP address/netmask>
            config exclude
                edit <ID>
                    set exclude-subnet <IP address/netmask>
```

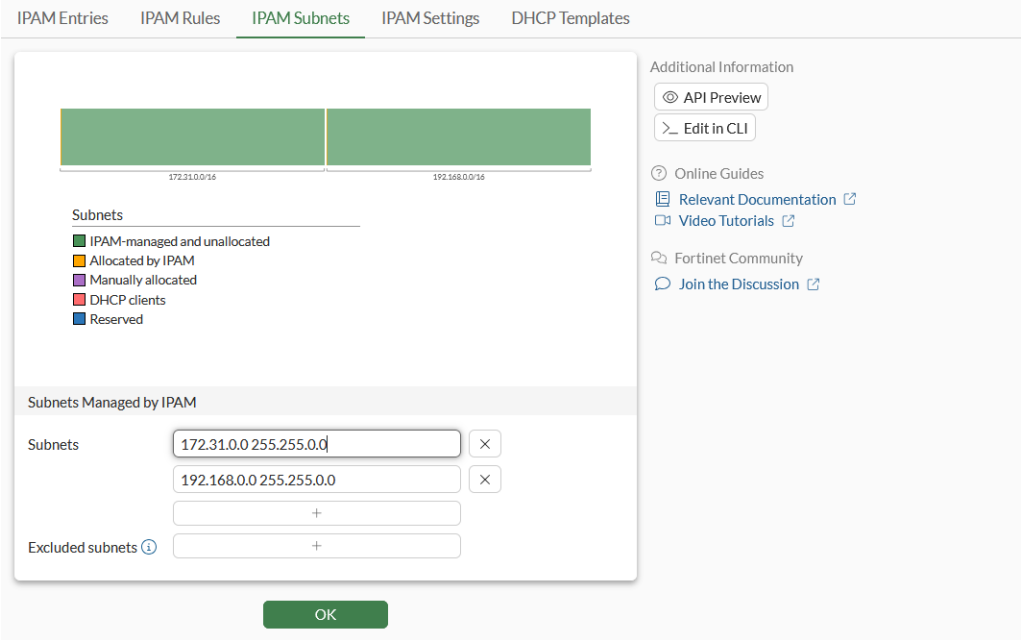
```
                next
            end
        next
    end
    config rules
        edit <rule_name>
            set item-type {interface | address}
            set item-name <name1> <name2> ...
            set device <name1> <name2> ...
            set role {any | lan | wan | dmz | undefined}
            set dhcp {enable | disable}
            set pool <pool_name>
            set dhcp-template <template_name>
        next
    end
end
```

Command	Description
`status {enable | disable}`	Enable/disable IP address management services.
`automatic-conflict-resolution {enable | disable}`	Enable/disable automatic conflict resolution. When `automatic-conflict-resolution` is enabled, IPAM will periodically check and validate the addresses of all interfaces. In case of any conflicts, IPAM will automatically attempt to obtain a new address for the affected interface managed by IPAM, ensuring no address duplication.
`manage-lan-addresses {enable | disable}`*	Enable/disable default management of LAN interface addresses.
`manage-lan-extension-addresses {enable | disable}`*	Enable/disable default management of FortiExtender LAN extension interface addresses.
`manage-ssid-addresses {enable | disable}`*	Enable/disable default management of FortiAP SSID addresses.
`require-subnet-match {enable | disable}`	Enable/disable reassignment of subnets to make requested and actual sizes match.
`config pools`	Set the subnet for the IP pool. Optionally, set the exclude-subnet for the exclude IP pool.
`config rules`	Set the device, interface, role, dhcp, IP pool and DHCP template for IPAM rules.

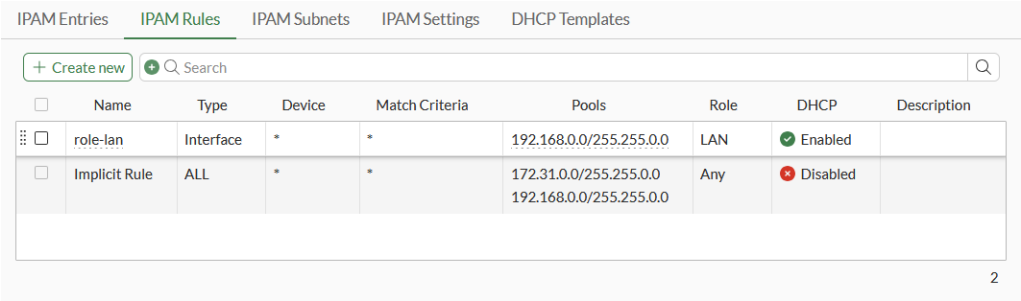
* See IPAM manage options on page 531 for more information about the `manage` options.

Default IPAM configuration after a factory reset

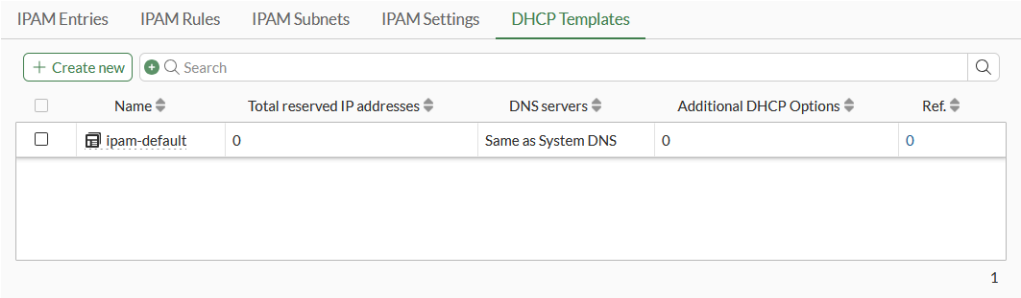
After performing a factory reset, enabled IPAM status will auto-generate two subnets. These subnets can be modified and deleted.



Two IPAM Rules will be automatically generated: one named *role-lan* for the subnet 192.168.0.0/255.255.0.0 with the role set to LAN and DHCP enabled, and another called *Implicit Rule* that includes both subnets with the role set to ANY and DHCP disabled. The *Implicit Rule* cannot be modified or deleted.



In addition, a DHCP template named *ipam-default* is generated. This template is a Security Fabric–wide global object with forced synchronization enabled. The template can be modified or deleted.



Using IPAM

Interfaces configured for IPAM receive an address from the IPAM server's address/subnet pool. DHCP Server can be enabled using IPAM Rules, and the address range is populated by IPAM. Users can customize the

address pool subnet and the size of a subnet that an interface can request. When a DHCP template is enabled through IPAM rules, interfaces configured with IPAM inherit DHCP settings from the template.

Interfaces with a LAN role, wireless network interfaces (vap-switch type), and FortiExtender LAN extension interfaces (lan-extension type) can receive an IP address from an IPAM server without any additional configuration at the interface level. See IPAM manage options on page 531 for more information

IPAM detects and resolves any IP conflicts that may occur on the interfaces that it manages. Users have the option to manually edit the interface or reallocate the IP. See IPAM automatic conflict resolution on page 532 for more information.

To configure an interface to receive an IP address through IPAM in the GUI:

1. On the FortiGate, go to *Network* > *Interfaces* and edit the required port.
2. Set *Role* to match the *Role* option set in the *IPAM Rules*.
3. For *Addressing Mode*, select *IPAM*.

   When the *Addressing mode* for an interface with a *LAN* role is set to *IPAM*, an additional option appears: *When to use IPAM*. This option is set to *Always* if the *Interfaces with LAN role* option under *IPAM Settings* is disabled, or to *Inherit IPAM auto-manage settings* option if *Interfaces with LAN role* is enabled. See IPAM manage options on page 531 for more information.

4. For *Network size*, select the subnet as per your network requirements.
5. Click *OK* to save the settings.

To configure an interface to receive an IP address through IPAM in the CLI:

```
config system interface
    edit <interface>
        set ip-managed-by-fortiipam
        set managed-subnetwork-size {4 | 8 | 16 | 32 | 64 | 128 | 256 | 512 | 1024 | 2048 | 4096 |
8192 | 16384 | 32768 | 65536}
    next
end
```

Advanced configurations

This section includes the following:

- IPAM manage options on page 531
- IPAM automatic conflict resolution on page 532
- IPAM conflict markers on page 533
- IPAM DHCP templates New on page 533

IPAM manage options

Interfaces with a LAN role, wireless network interfaces (vap-switch type), and FortiExtender LAN extension interfaces (lan-extension type) can also receive an IP address from an IPAM server without any additional configuration at the interface level.

See Interfaces on page 222 for more information.

When a manage option is enabled, any interface that meets the specified criteria will automatically receive an IP address from IPAM. However, if this option is disabled, interfaces that meet the criteria will not be configured by IPAM. All manage options are disabled by default. The central FortiIPAM configuration can be overridden at the interface level.

To configure IPAM manage options in the GUI:

1. Go to *Network > IPAM > IPAM Settings*.
2. Configure the following settings:

Option	Description
Interfaces with LAN role	Enable/disable LAN interface address management by default.
FortiAP SSIDs	Enable/disable FortiAP SSID address management by default.
FortiExtender LAN extensions	Enable/disable FortiExtender LAN extension interface address management by default.

3. Click *OK*.

To configure IPAM manage options in the CLI:

```
config system ipam
    set manage-lan-addresses {enable | disable}
    set manage-lan-extension-addresses {enable | disable}
    set manage-ssid-addresses {enable | disable}
end
```

To override the central FortiIPAM configuration at the interface level:

```
config system interface
    edit <name>
        set ip-managed-by-fortiipam {enable | disable | inherit-global}
    next
end
```

The default setting is to inherit from the global configuration (`inherit-global`) through the relevant `manage` option under `config system ipam`.

See Example 2: IPAM with automatically managed interfaces on page 547.

IPAM automatic conflict resolution

IPAM can detect and resolve any IP conflicts that may occur on the interfaces that it manages.

When `automatic-conflict-resolution` is enabled, IPAM will periodically check and validate the addresses of all interfaces. In case of any conflicts, IPAM will automatically attempt to obtain a new address for the affected interface managed by IPAM, ensuring no address duplication. Additionally, it also prevent conflicts when new FortiGates join the Security Fabric.

To configure automatic conflict resolution in the GUI:

1. Go to *Network* > *IPAM* > *IPAM Settings*.
2. Enable *Auto-resolve conflicts*.
3. Click *OK*.

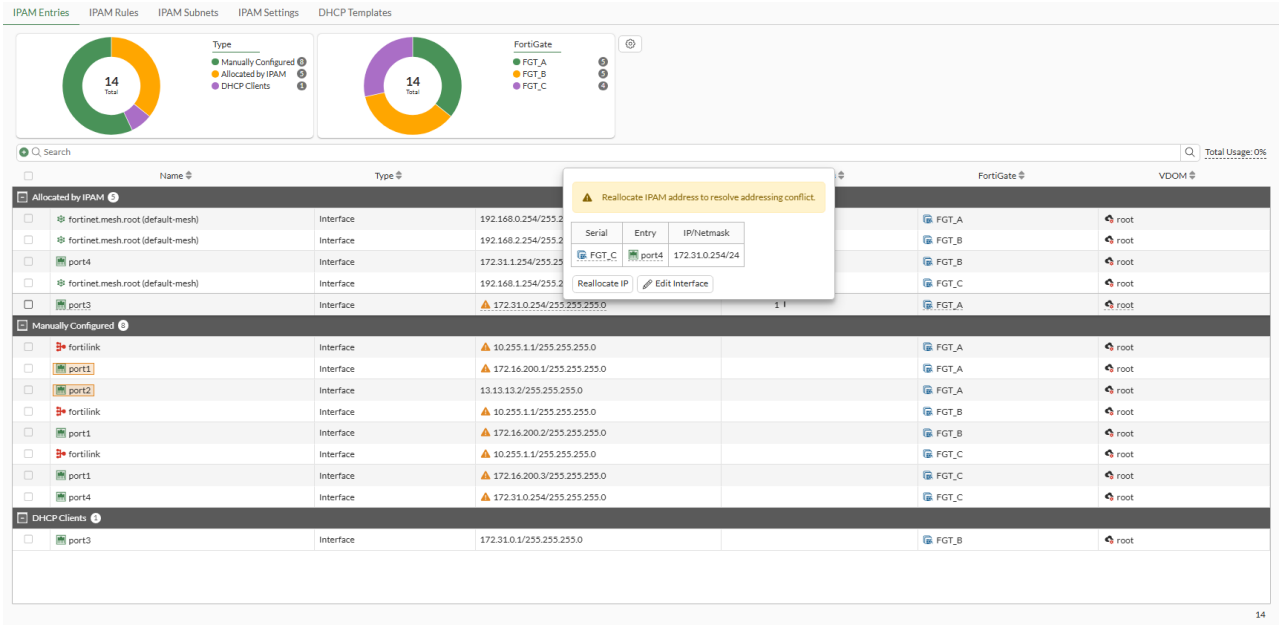
To configure automatic conflict resolution in the CLI:

```
config system ipam
    set automatic-conflict-resolution enable
end
```

When `automatic-conflict-resolution` is disabled, the conflicts do no resolve automatically.

To resolve conflicts in the GUI for IP addresses allocated by IPAM:

1. Go to *Network* > *IPAM* > *IPAM Entries*.
2. Hover your mouse over the conflict marker and select *Reallocate IP* to manually reallocate the IP address.



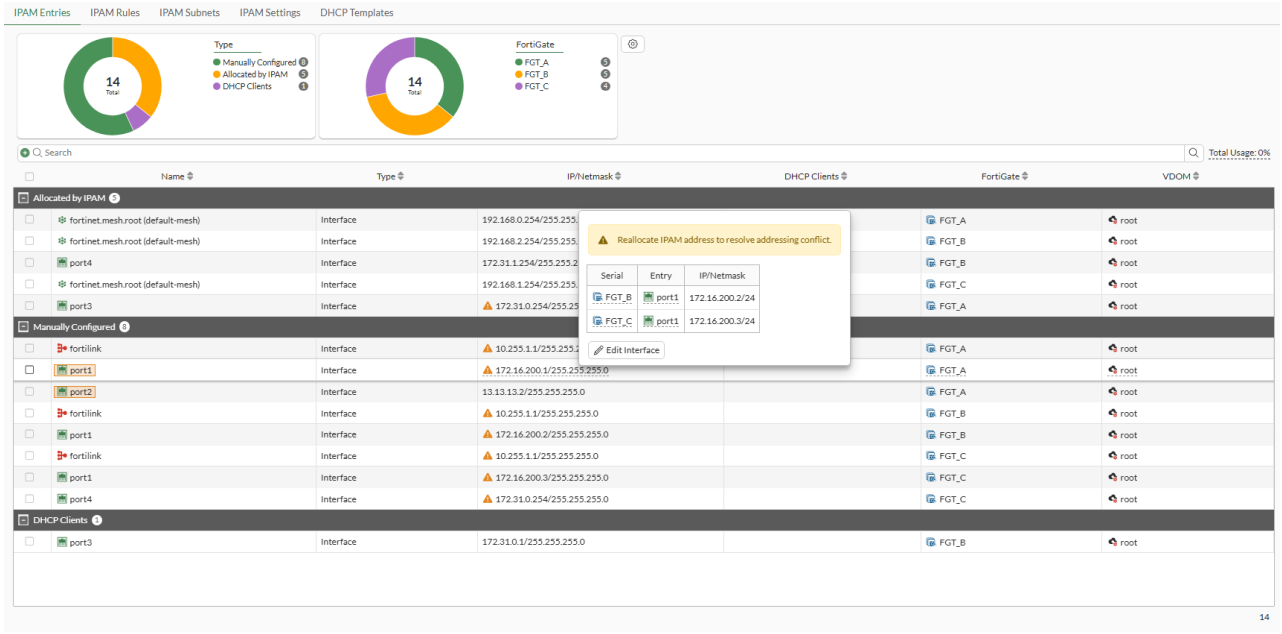
See Example 3: IPAM with auto-resolve conflicts on page 548.

IPAM conflict markers

The *IPAM Entries* tab displays conflict markers when there are IP pool IP address conflicts with manually configured IP addresses. Administrators can use the *Edit Interface* dialog to manually resolve the conflict.

To resolve conflicts in the GUI:

1. Go to *Network* > *IPAM* > *IPAM Entries*.
2. Hover your mouse over the conflict marker. The conflict marker information is displayed.



3. Click *Edit Interface*. The *Edit Interface* pane opens.
4. Enter a new IP address and netmask in the *IP/Netmask* field.
5. Click *OK*.

IPAM DHCP templates NEW

DHCP template support allows interfaces to inherit DHCP settings from reusable templates. This introduces a DHCP Template table and extends DHCP server attributes to support template-based configurations. IPAM rules and interfaces can now reference a DHCP template, enabling consistent and scalable DHCP configuration across interfaces.

For example, when the same DHCP server configuration is required on multiple interfaces, templates eliminate the need to configure the DHCP server settings on each interface manually. Instead, the shared template can be applied, ensuring consistency while reducing configuration effort and errors.

This feature works on a standalone FortiGate or in a Security Fabric.

DHCP Templates can be configured in the GUI on the *Network* > *IPAM* > *DHCP Templates* page (IPAM does not need to be enabled), or in the CLI using the `config system dhcp template` command; see config system dhcp template in the CLI Reference for details.

The `config dhcp server` and `config system ipam` commands are also updated:

- DHCP server:

```
config system dhcp server
    edit <server>
        set template <string>
        set template-subnet <class_ip&net_netmask>
        set template-subnet-from-interface {enable | disable}
        set dhcp-settings-from-fortiipam {enable | disable}
    next
end
```

Command	Description
`template <string>`	DHCP template associated with the server.
`template-subnet <class_ip&net_netmask>`	Configure template subnet.
`template-subnet-from-interface {enable | disable}`	Enable/disable using interface subnet as DHCP template subnet.
`dhcp-settings-from-fortiipam {enable | disable}`	Enable/disable populating of DHCP server settings from FortiIPAM.

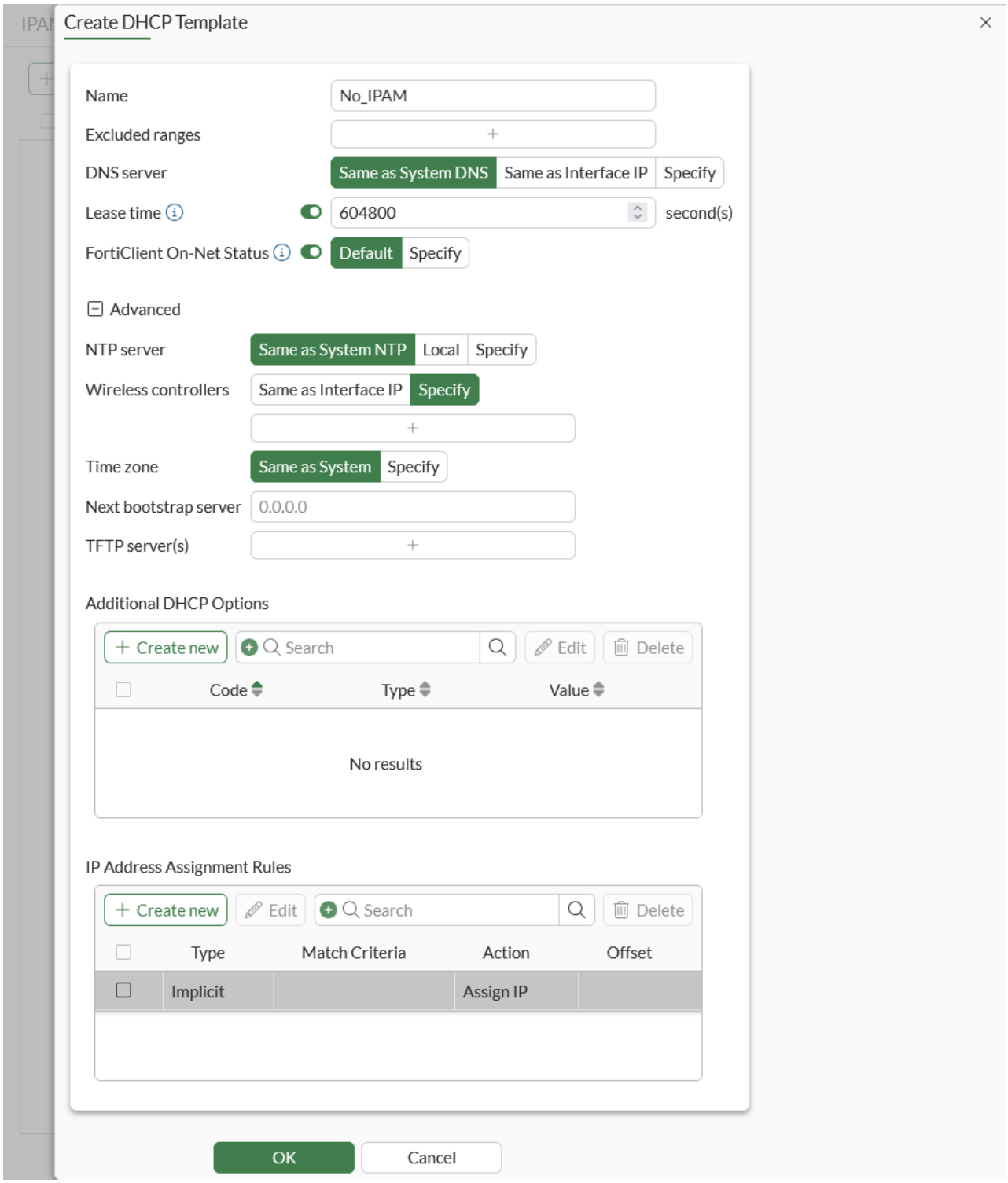
- IPAM rule:

```
config system ipam
    config rules
        edit <rule>
            set dhcp-template <string>
        next
    end
end
```

Command	Description
`dhcp-template <string>`	DHCP template for assigned interface.

To create a DHCP template in the GUI:

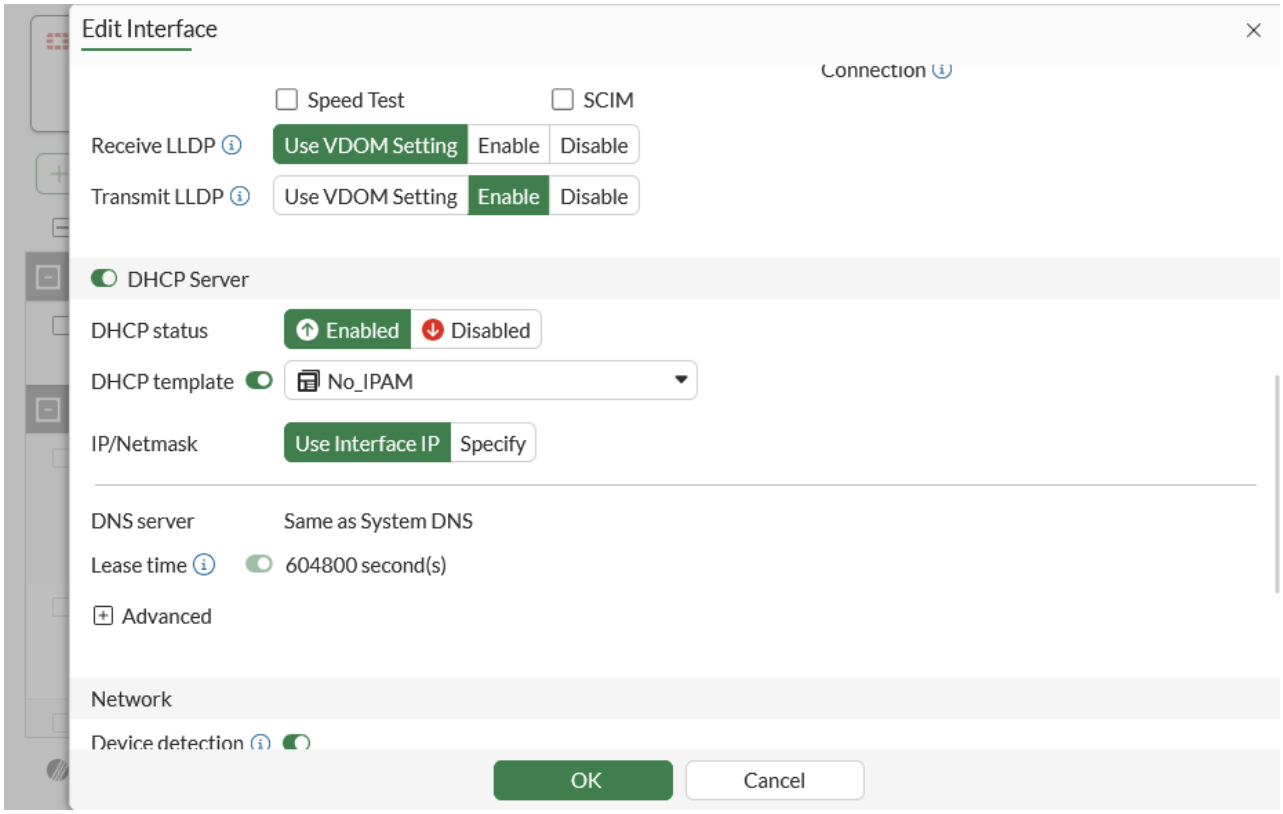
1. Go to *Network* > *IPAM* > *DHCP Templates* and click *Create new*.
2. Configure the new DHCP template.


Create DHCP Template
Name
No_IPAM
Excluded ranges
DNS server
Same as System DNS
Same as Interface IP
Specify
Lease time
604800
second(s)
FortiClient On-Net Status
Default
Specify
Advanced
NTP server
Same as System NTP
Local
Specify
Wireless controllers
Same as Interface IP
Specify
Time zone
Same as System
Specify
Next bootstrap server
0.0.0.0
TFTP server(s)
Additional DHCP Options
Create new
Search
Edit
Delete
Code
Type
Value
No results
IP Address Assignment Rules
Create new
Edit
Search
Delete
Type
Match Criteria
Action
Offset
Implicit
Assign IP
OK
Cancel

3. Click *OK*.

To use the DHCP template as the DHCP server in an interface:

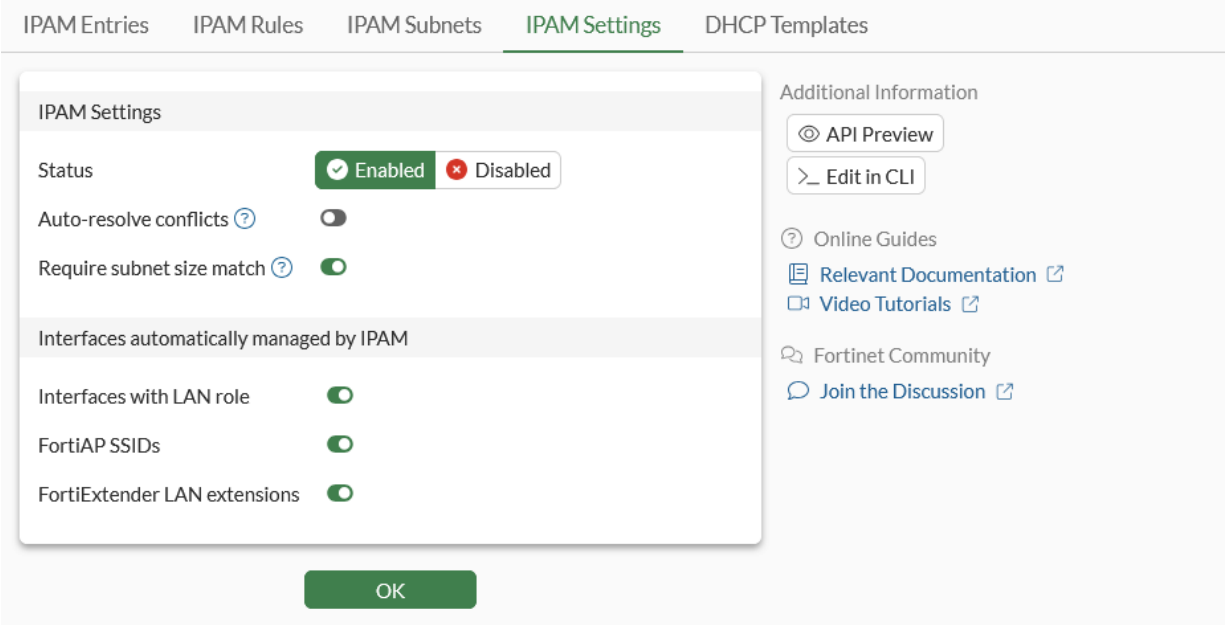
1. Go to *Network* > *Interfaces* and edit an interface.
2. Enable DHCP template and select the DHCP template.



3. Click *OK*.

To use a DHCP template with IPAM:

1. Enable IPAM:
   a. Go to *Network* > *IPAM* > *IPAM Settings*.
   b. Set *Status* to *Enabled* and then click *OK*.

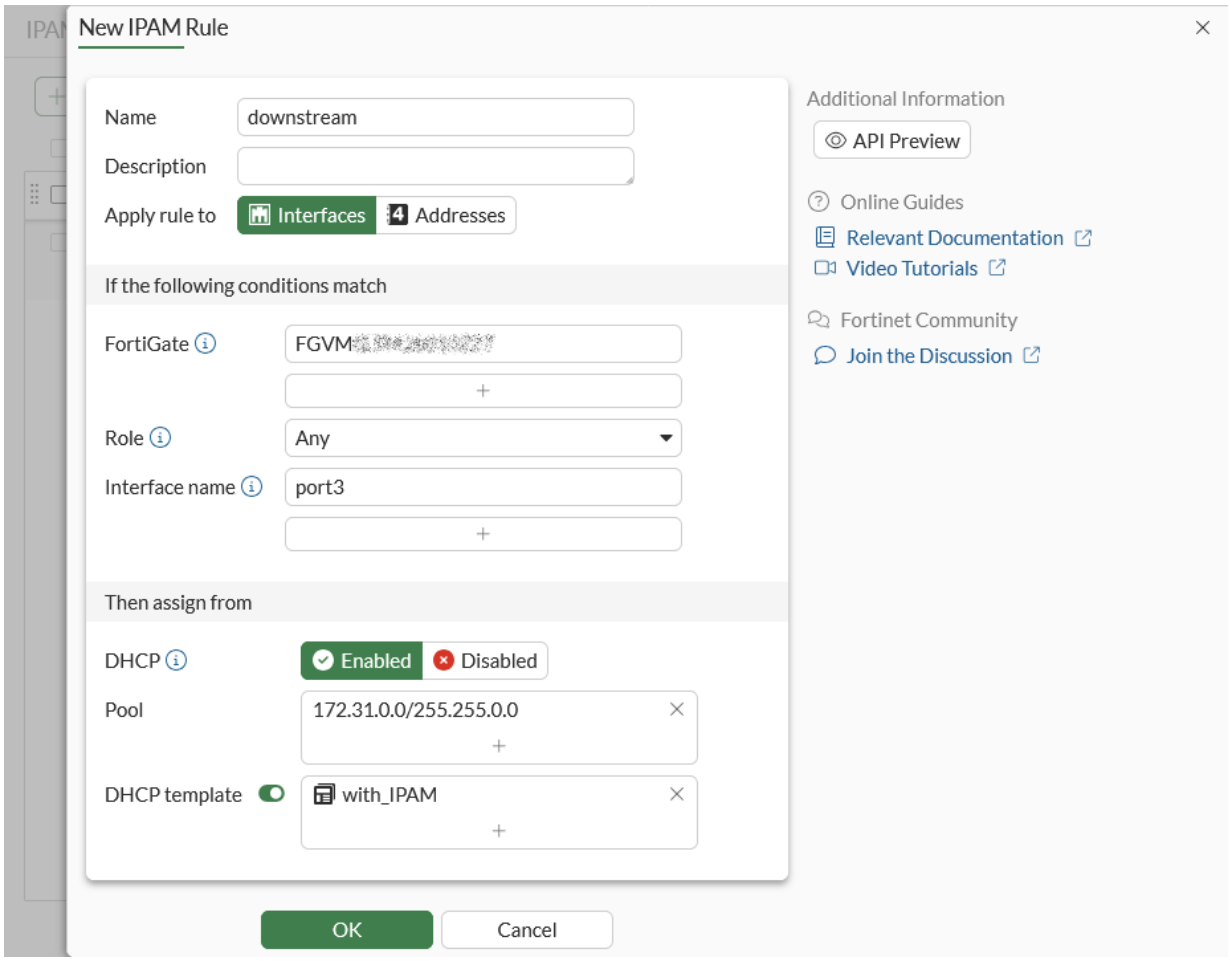


2. Create a DHCP Template and enable Security Fabric global object for it:

```
config system dhcp template
    edit "with_IPAM"
        set fabric-object enable
    next
end
```

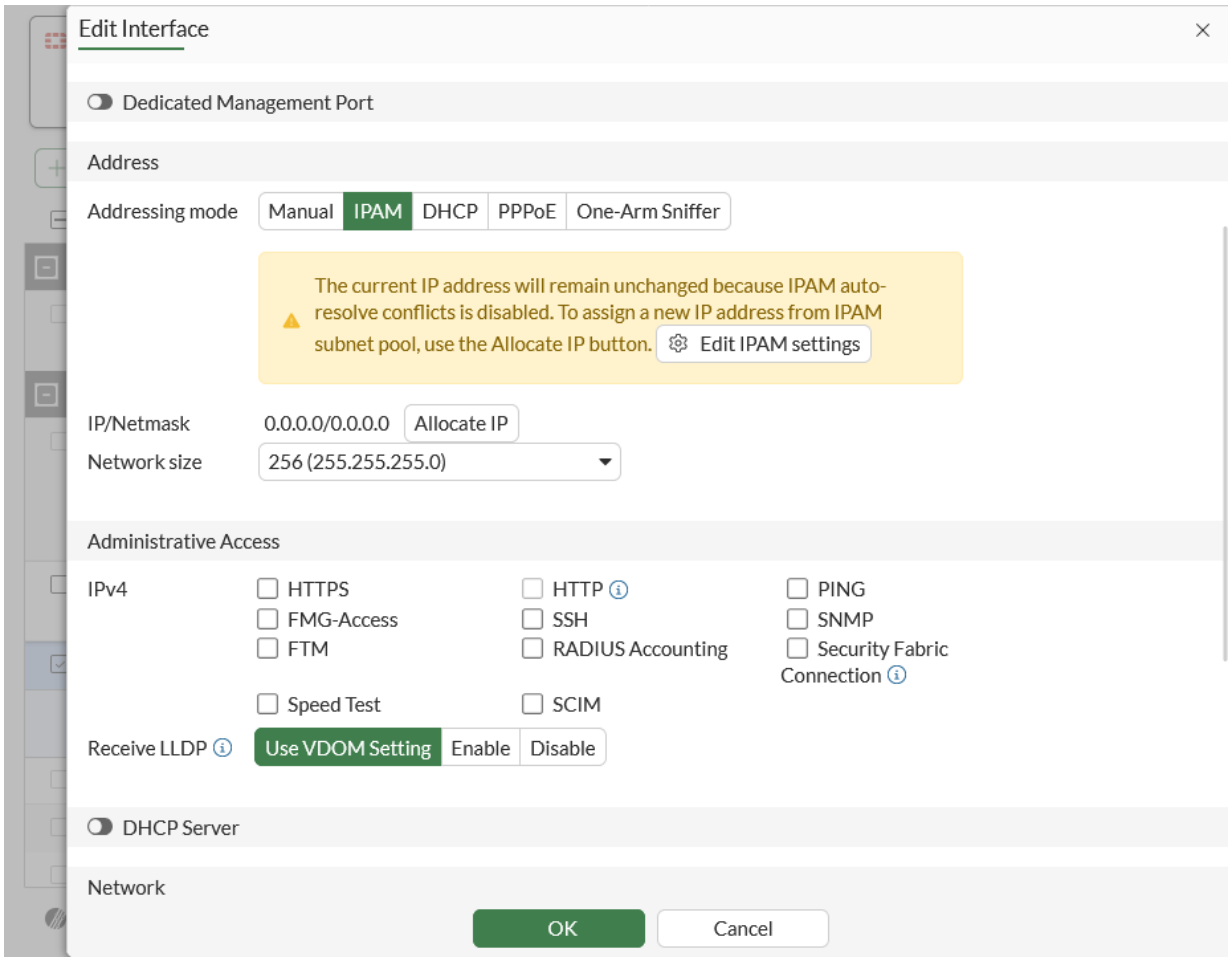
In a Security Fabric, the downstream FortiGate will inherit the upstream DHCP template when `fabric-object` is enabled.

3. Create an IPAM rule:
   a. Go to *Network* > *IPAM* > *IPAM Rules* and click *Create new*.
   b. Set *Interface name* to *port3*.
   c. Enable *DHCP* and set *Pool* to *172.31.0.0/255.255.0.0*.
   d. Enable *DHCP template* and then select the DHCP template.
   e. Configuring the remaining settings as needed.



f. Click *OK*.

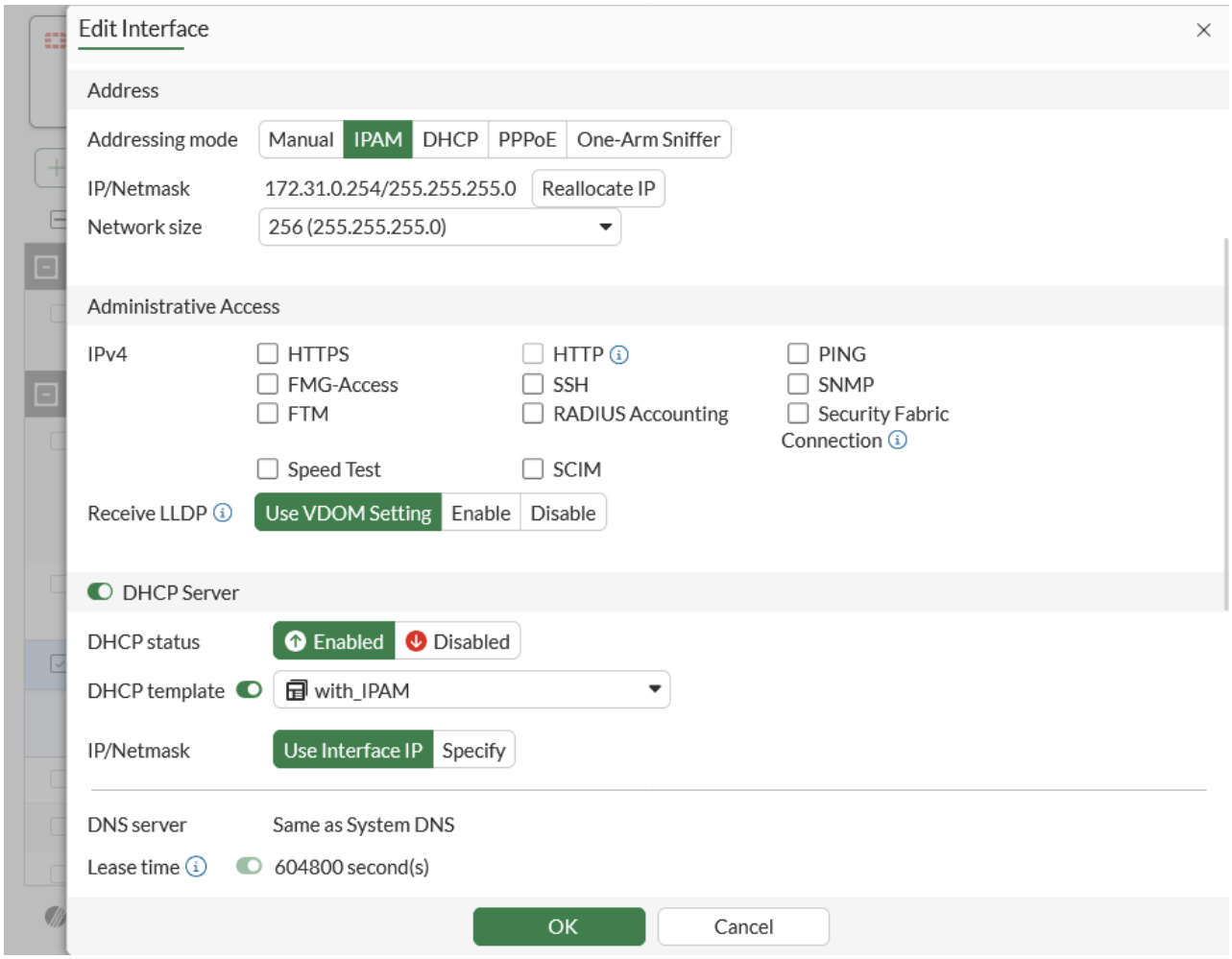
4. Assign an IP address and DHCP template to an interface using the IPAM:
   a. Go to *Network* > *Interfaces* and edit *port3*.
   b. Set *Addressing Mode* to *IPAM* and leave *Network size* as the default subnet.



c. Click *OK*.

5. Edit port3 again:

   Because IPAM is managing a 172.31.0.0/16 network, the *IP/Netmask* field is automatically assigned a class C IP by IPAM. The *DHCP Server* section is also automatically configured by IPAM using the DHCP template.



6. Click *OK*.

Configuration examples

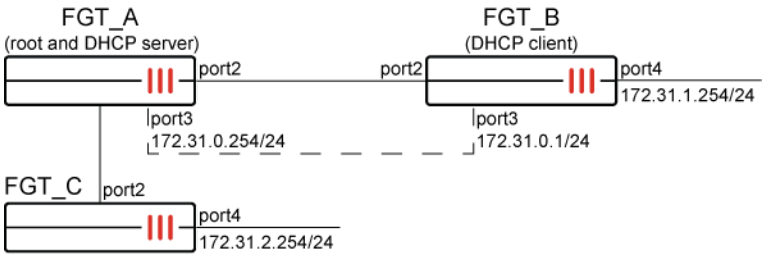
This section includes the following examples:

- Example 1: IPAM in a Security Fabric setup on page 540
- Example 2: IPAM with automatically managed interfaces on page 547
- Example 3: IPAM with auto-resolve conflicts on page 548

Example 1: IPAM in a Security Fabric setup

In this example, FGT_A is the Security Fabric root with IPAM enabled. FGT_B and FGT_C are downstream Fabric devices and retrieve IPAM information from FGT_A.

The Fabric interface on all FortiGates is port2. FGT_A acts as the DHCP server, and FGT_B acts as the DHCP client.



General configuration steps

This section breaks down the configuration for this example into smaller procedures:

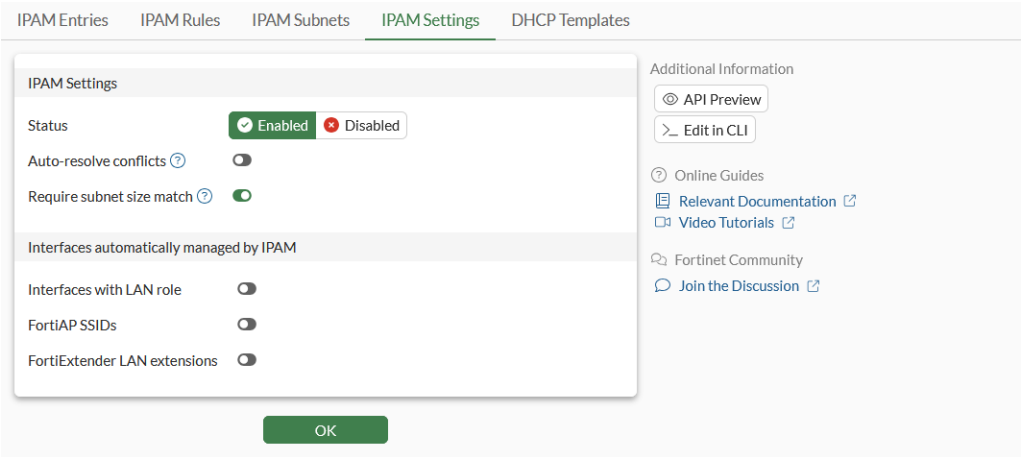
1. Configure the root FortiGate unit:
   a. Configure the IPAM Settings.
   b. Configure the IPAM Subnets.
   c. Configure the IPAM Rules.
   d. Assign an IP address using the IPAM.
2. Configure the downstream FortiGate units:
   a. Configure FGT_B and FGT_C to receive an IP address using the IPAM.
   b. Configure FGT_B to receive an IP address using the DHCP.
3. Verify IPAM allocations and DHCP clients.

Configuring IPAM in the GUI

Use the following steps to configure the example configuration in the GUI.

To configure the root FortiGate (FGT_A):

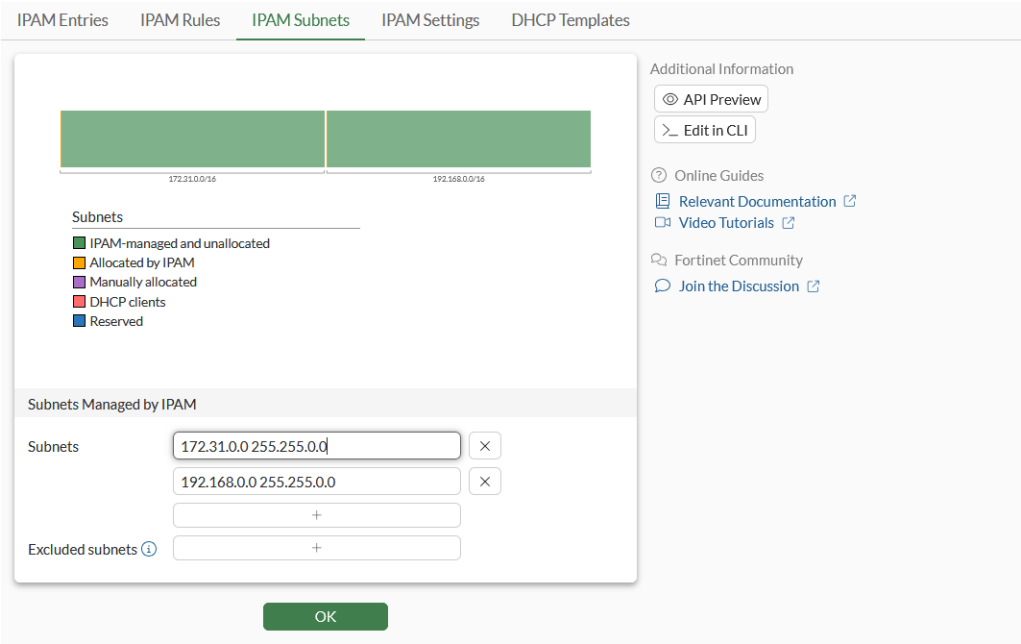
1. Configure the IPAM settings:
   a. Go to *Network* > *IPAM* > *IPAM Settings*.
   b. Set *Status* to *Enabled*.



   c. Click *OK*.
2. Configure the IPAM subnets:

a. Go to *Network* > *IPAM* > *IPAM Subnets*.

b. Add subnet 172.31.0.0/16, if not already present.



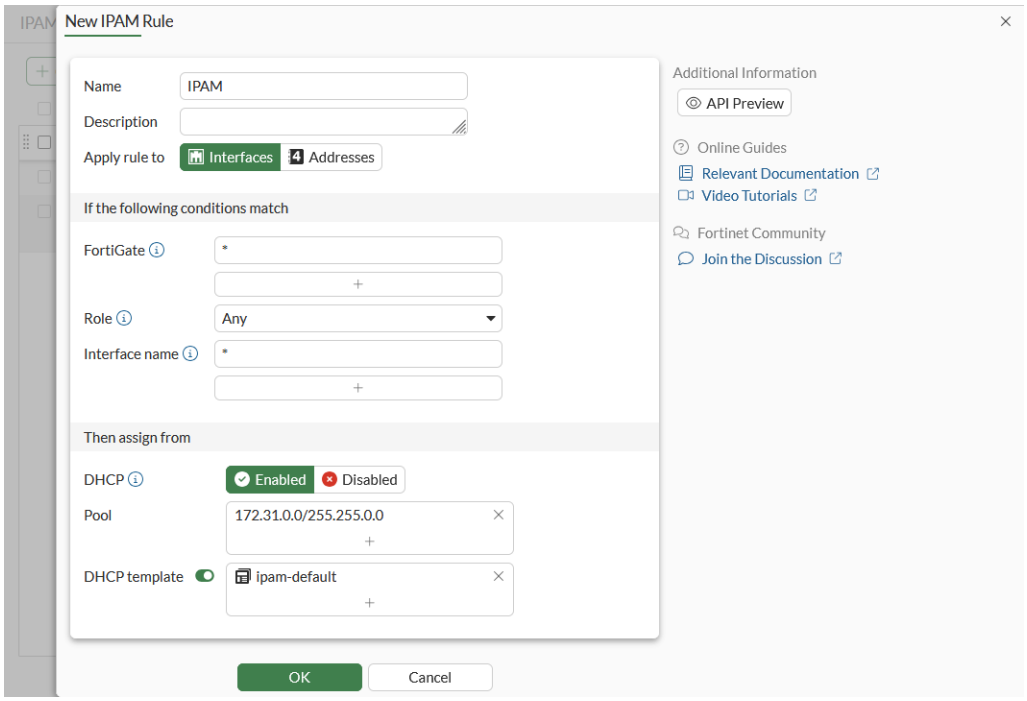
c. Click *OK*.

3. Configure the IPAM rules:

a. Go to *Network* > *IPAM* > *IPAM Rules*.

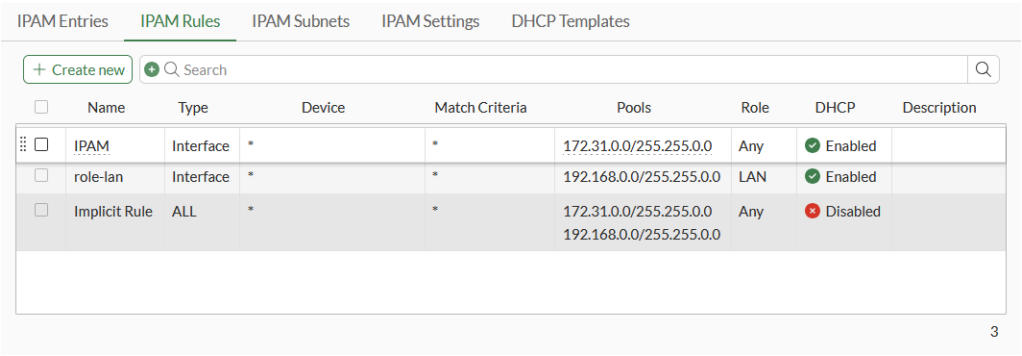
b. Click *Create New* and configure the settings:

Option	Value
Name	IPAM
Description	Optional
Apply rule to	Interfaces
FortiGates	*
Role	Any
Interfaces	*
DHCP	Enabled
Pools	172.31.0.0/255.255.0.0
DHCP template	Enabled Choose *ipam-default*



c. Click *OK*.

4. Drag the new IPAM rule to the top of the list, if it is not already there.

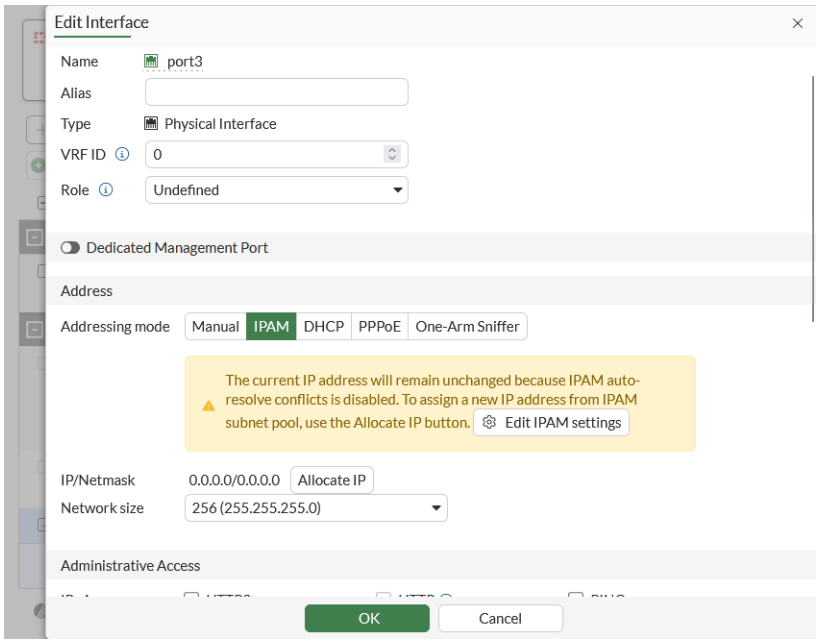


5. Assign an IP address using the IPAM:

a. Go to *Network* > *Interfaces* and edit port3.

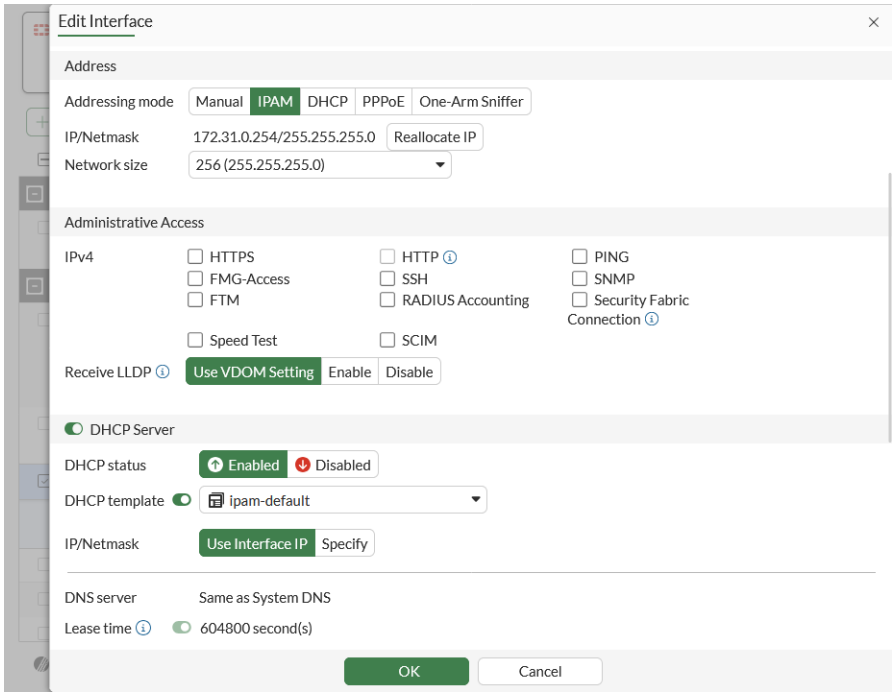
b. Set *Addressing Mode* to *IPAM*

c. Leave *Network size* as the default subnet.



d. Click *OK*.

Since IPAM is managing a 172.31.0.0/16 network, the *IP/Netmask* field in the *Address* section is automatically assigned a class C IP by IPAM.In the DHCP Server section, the DHCP template is enabled and automatically assigned by IPAM.



To configure the downstream FortiGate units:

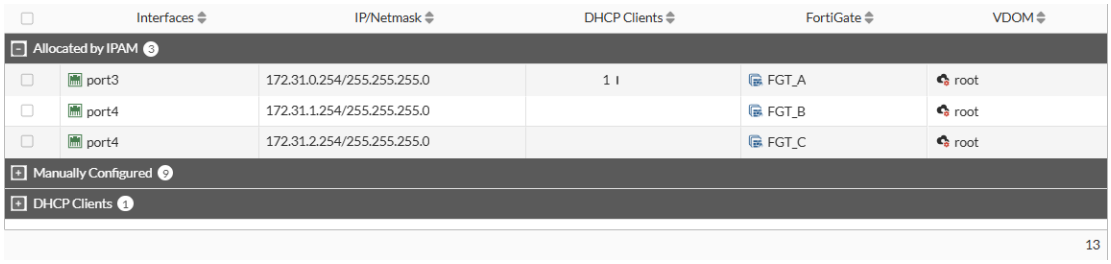
1. Configure FGT_B and FGT_C to receive an IP address using the IPAM:
   a. Log in to FGT_B and set the *Addressing Mode* of port4 to *IPAM*. The subnet assigned from the pool on the root is 172.31.1.254/24.
   b. Log in to FG_C and set the *Addressing Mode* of port4 to *IPAM*. The subnet assigned from the pool on the root is 172.31.2.254/24.

   Any interface on a downstream FortiGate can be managed by the IPAM server. The interface does not have to be directly connected to the Fabric root FortiGate.

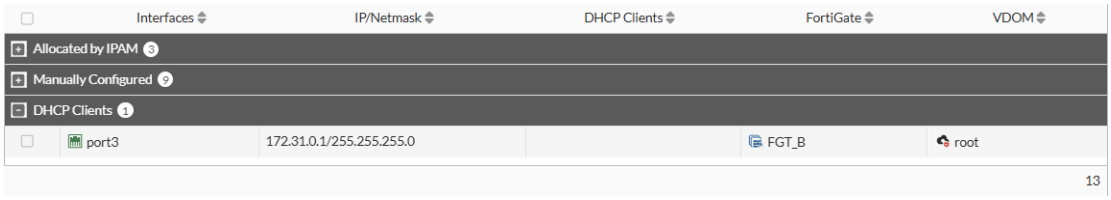
2. Configure FGT_B to receive an IP address using the DHCP:
   a. Log in to FGT_B and set the *Addressing Mode* of port3 to *DHCP*.
   b. Click *OK*.
      The IP assigned from the DHCP pool on the root is 172.31.0.1/24.

To verify IPAM allocations and DHCP clients:

Log in to FGT_A, go to *Network* > *IPAM* > *IPAM Entries* to view the IP addresses allocated by IPAM (port3, port4, and port4)



FGT_B, port3 is a DHCP client and the DHCP server interface (FGT_A port3) is managed by IPAM, so it is displayed in the *DHCP Clients* section:



Configuring IPAM in the CLI

Use the following steps to configure the example configuration in the CLI.

To configure the root FortiGate (FGT_A):

Enable IPAM and configure IPAM subnets and rules:

```
config system ipam
    set status enable
    config pools
```

```
        edit "1"
            set subnet 172.31.0.0 255.255.0.0
        next
    end
    config rules
        edit "IPAM"
            set device "*"
            set item-name "*"
            set pool "default-pool"
            set dhcp enable
            set dhcp-template "ipam-default"
        next
    end
end
```

To configure the downstream FortiGate units:

1. On both the FGT_B and FGT_C, receive an IP address using the IPAM:

```
config system interface
    edit "port4"
        set ip-managed-by-fortiipam enable
    next
end
```

2. On FGT_B, receive an IP address using the DHCP:

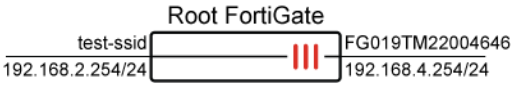
```
config system interface
    edit "port3"
        set mode dhcp
    next
end
```

To verify IPAM allocations and DHCP clients:

```
# diagnose sys ipam list entries
Entries: (sn, vdom, interface, subnet/mask, conflict)
  IPAM Entries:
  FGVMSLTM25016113 root port3 172.31.0.254/24
  FGVMSLTM25016114 root port4 172.31.1.254/24
  FGVMSLTM25016115 root port4 172.31.2.254/24

  Static Entries:
  FGVMSLTM25016113 root fortilink 10.255.1.1/24 C
  FGVMSLTM25016113 root port1 172.16.200.1/24 C
…
  DHCP Entries:
  FGVMSLTM25016114 root port3 172.31.0.1/24
```

Example 2: IPAM with automatically managed interfaces

In this example, the FortiGate serves as the Security Fabric root and has two interfaces: test-ssid (vap-switch type) and FG019TM22004646 (lan-extension type). Currently, neither interface has an IP address assigned to it.



To configure IPAM on the root FortiGate:

1. Go to *Network* > *IPAM* > *IPAM Settings*.
2. Enable the *Status*, *Interfaces with LAN role*, *FortiAP SSIDs*, and *FortiExtender LAN extensions* settings.

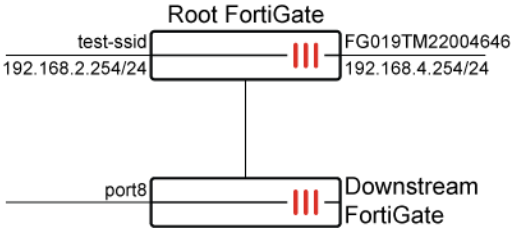
   IPAM is disabled by default, so all these options are disabled by default. Each option must be activated individually to function, and they do not depend on one another.

3. Click *OK*.

   After enabling IPAM on the root FortiGate with the specified settings, FortiGates that are part of the Security Fabric and have an interface set to either the LAN role, `vap-switch` type, or `lan-extension` type will automatically receive an IP assignment from the IPAM server without requiring any additional configuration at the interface level.
4. Verify the list of IPAM entries:

```
# diagnose sys ipam list entries
Entries: (sn, vdom, interface, subnet/mask, conflict)

IPAM Entries:
  FGVM08TM22004645 root FG019TM22004646 192.168.4.254/24
  FGVM08TM22004645 root test-ssid 192.168.2.254/24
```

   When a downstream FortiGate joins the Security Fabric, port8 is set to a LAN role with no IP address assigned. The IPAM server assigns an IP to port8 of the downstream FortiGate since its role was set to LAN.

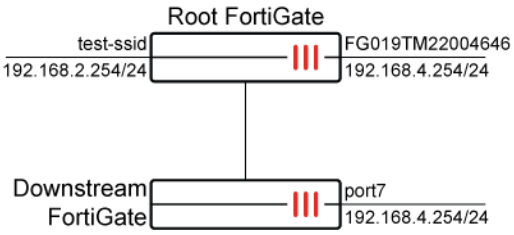


5. Verify the list of IPAM entries in the CLI:

```
# diagnose sys ipam list entries
Entries: (sn, vdom, interface, subnet/mask, conflict)

  IPAM Entries:
  FGVM08TM22004645 root FG019TM22004646 192.168.4.254/24
  FGVM08TM22004645 root test-ssid 192.168.2.254/24
  FGVM08TM22004647 root port8 192.168.3.254/24
```

Example 3: IPAM with auto-resolve conflicts

In this example, the FortiGate serves as the Security Fabric root with IPAM and Auto-resolve conflicts enabled and has two interfaces: test-ssid (vap-switch type) and FG019TM22004646 (lan-extension type) managed by IPAM. When a downstream FortiGate joins the Security Fabric, the port7 interface is configured with a static IP (192.168.4.254/24). It is observed that the FG019TM22004646 interface of the root FortiGate conflicts with port7 of the downstream FortiGate.

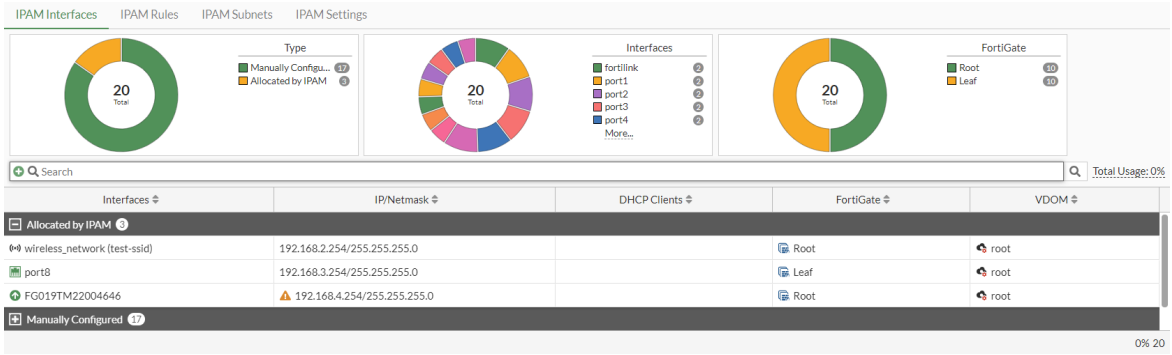


To configure *Auto-resolve conflicts* on the root FortiGate:

1. Go to *Network* > *IPAM* > *IPAM Settings*.
2. Enable *Auto-resolve conflicts*.
3. Click *OK*.

To verify the IP address conflict resolution:

1. On the root FortiGate, go to *Network* > *IPAM* > *IPAM Entries*.

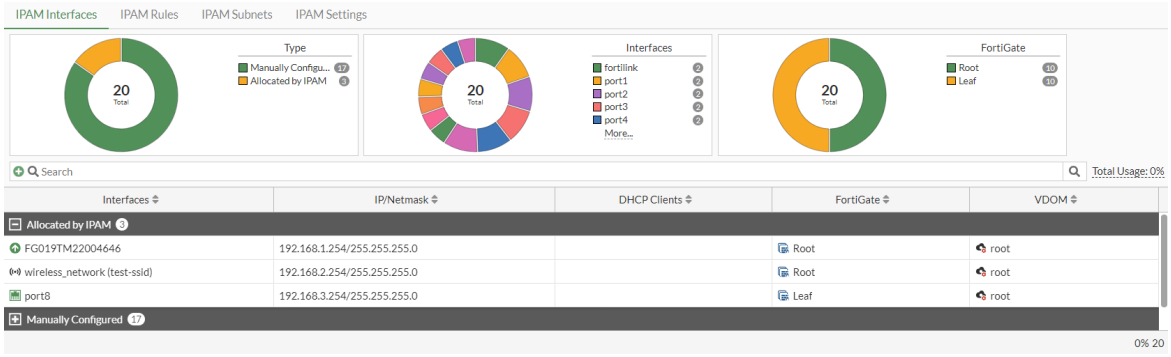


   There is a conflict marker (warning icon) beside the IP address of *FG019TM22004646* due to a conflict between the IPAM-assigned interface FG019TM22004646 of the root FortiGate and the manually configured interface of the downstream FortiGate.
2. Verify the list of IPAM entries in the CLI:

```
# diagnose sys ipam list entries
Entries: (sn, vdom, interface, subnet/mask, conflict)

  IPAM Entries:
  FGVM08TM22004645 root test-ssid 192.168.2.254/24
  FGVM08TM22004647 root port8 192.168.3.254/24
  FGVM08TM22004645 root FG019TM22004646 192.168.4.254/24 C
```

3. After some time, since *Auto-resolve conflicts* is enabled in the IPAM settings, the conflict is resolved automatically.

   FG019TM22004646 has been assigned a new IP address of 192.168.1.254/24.



4. Verify the list of IPAM entries in the CLI:

```
# diagnose sys ipam list entries
Entries: (sn, vdom, interface, subnet/mask, conflict)

  IPAM Entries:
  FGVM08TM22004645 root FG019TM22004646 192.168.1.254/24
  FGVM08TM22004645 root test-ssid 192.168.2.254/24
  FGVM08TM22004647 root port8 192.168.3.254/24
```

Diagnostics

Use the following commands to view IPAM related diagnostics.

To view the largest available subnet size:

```
# diagnose sys ipam largest-available-subnet
Largest available subnet is a /17.
```

To verify IPAM allocation information:

```
# diagnose sys ipam list entries
IPAM Entries: (sn, vdom, interface, subnet/mask, flag)
  F140EP4Q17000000 root port4 172.31.2.1/24 0
  FG5H1E5818900001 root port3 172.31.0.1/24 0
  FG5H1E5818900002 root port4 172.31.1.1/24 0
  FG5H1E5818900003 root port3 172.31.0.2/24 1
```

To verify the available subnets:

```
# diagnose sys ipam list free-subnets
IPAM free subnets: (subnet/mask)
  172.31.3.0/24
```

```
  172.31.4.0/22
  172.31.8.0/21
  172.31.16.0/20
  172.31.32.0/19
  172.31.64.0/18
  172.31.128.0/17
```

To remove a device from IPAM in the Security Fabric:

```
# diagnose sys ipam delete device F140EP4Q17000000
Successfully removed device F140EP4Q17000000 from ipam
```

To review IPAM statistics with SNMP:

Use the `snmpwalk` command to retrieve diagnostic statistics for IP pools, including allocated addresses, total available addresses, and usage percentages, using the IPAM OID branch (1.3.6.1.4.1.12356.101.32).

```
# snmpwalk -v2c -c <community_string> <IP_address> 1.3.6.1.4.1.12356.101.32
FORTINET-FORTIGATE-MIB::fnFortiGateMib.32.1.1.0 = INTEGER: 17
FORTINET-FORTIGATE-MIB::fnFortiGateMib.32.1.2.0 = INTEGER: 6336
FORTINET-FORTIGATE-MIB::fnFortiGateMib.32.1.3.0 = INTEGER: 1536
FORTINET-FORTIGATE-MIB::fnFortiGateMib.32.1.4.0 = INTEGER: 7872
FORTINET-FORTIGATE-MIB::fnFortiGateMib.32.1.5.0 = INTEGER: 262144
FORTINET-FORTIGATE-MIB::fnFortiGateMib.32.1.6.0 = INTEGER: 3
```

OID Suffix	Description
.32.1.1	Largest Subnet Available.
.32.1.2	Number of IPAM allocated IP addresses.
.32.1.3	Number of manually allocated IP addresses.
.32.1.4	Sum of IPAM and manually allocated IP addresses.
.32.1.5	Total number of IP addresses in IPAM pools.
.32.1.6	Percentage of IP addresses used from IPAM pools.

DHCP servers and relays

A DHCP server dynamically assigns IP addresses to hosts on the network connected to the interface. The host computers must be configured to obtain their IP addresses using DHCP. You can configure one or more DHCP servers on any FortiGate interface.

A DHCP server can be in server or relay mode. In server mode, you can define up to ten address ranges to assign addresses from, and options such as the default gateway, DNS server, lease time, and other advanced settings. In relay mode, the interface forwards DHCP requests from DHCP clients to an external DHCP server

and returns the responses to the DHCP clients. The DHCP server must have appropriate routing so that its response packets to the DHCP clients arrive at the unit.

If an interface is connected to multiple networks through routers, you can add a DHCP server for each network. The IP range of each DHCP server must match the network address range. The routers must be configured for DHCP relay.

Default DHCP server for entry-level FortiGates

On entry-level FortiGates, a DHCP server is configured on the internal interface, by default, with the following values:

Field	Value
Address Range	192.168.1.110 to 192.168.1.210
Netmask	255.255.255.0
Default Gateway	192.168.1.99
Lease Time	7 days
DNS Server 1	192.168.1.99

These settings are appropriate for the default internal interface IP address of 192.168.1.99. If you change this address to a different network, you need to change the DHCP server settings to match.

- Basic configuration on page 551
- DHCP options on page 555
- DHCP addressing mode on an interface on page 563
- VCI pattern matching for DHCP assignment on page 566
- DHCP shared subnet on page 568
- Multiple DHCP relay servers on page 570
- DHCP smart relay on interfaces with a secondary IP on page 571
- FortiGate DHCP works with DDNS to allow FQDN connectivity to leased IP addresses on page 574

Basic configuration

The following contains information on basic configurations.

Configure a DHCP server on an interface

A DHCP server can be configured on an interface in the GUI from *Network* > *Interfaces*.

To configure a DHCP server in the GUI:

1. Go to *Network* > *Interfaces*.
2. Edit an interface.
3. Enable the *DHCP Server* option and configure the settings.
4. Click *OK*.

Field	Description
Address Range	By default, the FortiGate unit assigns an address range based on the address of the interface for the complete scope of the address. For example, if the interface address is 172.20.120.230, the default range created is 172.20.120.231 to 172.20.120.254. Select the range and select *Edit* to adjust the range or select *Create New* to add a different range.
Netmask	Enter the netmask of the addresses that the DHCP server assigns.
Default Gateway	Select this to use either *Same as Interface IP* or select *Specify* and enter the IP address of the default gateway that the DHCP server assigns to DHCP clients.
DNS Server	Select this to use *Same as system DNS*, *Same as Interface IP* or select *Specify* and enter the IP address of the DNS server.
Mode	Select the type of DHCP server FortiGate will be. By default, it is a *Server*. Select *Relay* if needed. When *Relay* is selected, the above configuration is replaced by a field to enter the *DHCP Server IP* address.
DHCP Server IP	This appears only when *Mode* is *Relay*. Enter the IP address of the DHCP server where FortiGate obtains the requested IP address.
Type	Select this to use the DHCP in *Regular* or *IPsec* mode.
Additional DHCP Options	Use this to create new DHCP options.
Add from DHCP Client List	If the client is currently connected and using an IP address from the DHCP server, you can select this option to select the client from the list.

To configure a DHCP server in the CLI:

```
config system dhcp server
    edit 1
        set dns-service default
        set default-gateway 192.168.1.2
        set netmask 255.255.255.0
        set interface "port1"
        config ip-range
            edit 1
                set start-ip 192.168.1.1
                set end-ip 192.168.1.1
            next
            edit 2
```

```
                set start-ip 192.168.1.3
                set end-ip 192.168.1.254
            next
        end
        set timezone-option default
        set tftp-server "172.16.1.2"
    next
end
```

Configure a DHCP relay on an interface

To configure a DHCP relay in the GUI:

1. Go to *Network* > *Interfaces*.
2. Edit an interface.
3. Expand the *Advanced* section and set *Mode* to *Relay*.
4. Enter the *DHCP Server IP*.
5. Click *OK*.

To configure a DHCP relay in the CLI:

1. Configure the interface, making sure to configure set `dhcp-relay-ip`:

```
config system interface
    edit "port2"
        set vdom "root"
        set dhcp-relay-service enable
        set ip 10.1.1.5 255.255.255.0
        set allowaccess ping https ssh fabric
        set type physical
        set snmp-index 4
        set dhcp-relay-ip "192.168.20.10"
    next
end
```

Configure a DHCP server and relay on an interface

A FortiGate interface can be configured to work in DHCP server mode to lease out addresses, and at the same time relay the DHCP packets to another device, such as a FortiNAC to perform device profiling.

The DHCP message to be forwarded to the relay server under the following conditions:

- `dhcp-relay-request-all-server` is enabled
- Message type is either DHCPDISCOVER or DHCPINFORM
- Client IP address in client message is 0
- Server ID is NULL in the client message
- Server address is a broadcast address (255.255.255.255)
- Server address is 0

Configuring a DHCP server and relay on the same interface is currently only supported in the CLI.

To configure a DHCP server and relay in the CLI:

1. Configure the interface:

```
config system interface
    edit "port2"
        set vdom "root"
        set dhcp-relay-service enable
        set ip 10.1.1.5 255.255.255.0
        set allowaccess ping https ssh fabric
        set type physical
        set snmp-index 4
        set dhcp-relay-ip "192.168.20.10"
        set dhcp-relay-request-all-server enable
    next
end
```

2. Configure the DHCP server settings:

```
config system dhcp server
    edit 17
        set status enable
        set dns-service default
        set default-gateway 10.1.1.5
        set netmask 255.255.255.0
        set interface "port2"
        config ip-range
            edit 1
                set start-ip 10.1.1.6
                set end-ip 10.1.1.254
            next
        end
    next
end
```

Excluding addresses in DHCP

If you have a large address range for the DHCP server, you can block a range of addresses that will not be included in the available addresses for the connecting users using the `config exclude-range` subcommand.

To exclude addresses in DHCP:

```
config system dhcp server
    edit <id>
        config exclude-range
            edit <sequence_number>
                set start-ip <address>
```

```
                set end-ip <address>
            next
        end
    next
end
```

Viewing information about DHCP server connections

To view information about DHCP server connections, go to *Dashboard* > *Asset and Identities* > *DHCP*. On this page you can also add IP addresses to the reserved IP address list.

DHCP options

When adding a DHCP server, you can include DHCP options. The DHCP options are BOOTP vendor information fields that provide additional vendor-independent configuration parameters to manage the DHCP server. For example, you might need to configure a FortiGate DHCP server that gives out a separate option as well as an IP address, such as an environment that needs to support PXE boot with Windows images. The *Option code* is specific to the application. The documentation for the application indicates the values to use. The *Option code* is a value between 1 and 255.

For detailed information about DHCP options, see RFC 2132, DHCP Options and BOOTP Vendor Extensions.

The FortiOS DHCP server supports up to a maximum of 30 options per DHCP server. These optional fields can be set in either the GUI or CLI.

DHCP server options are not available in transparent mode.

The DHCP options include:

- Common DHCP options on page 555
- Additional DHCP options on page 558
- IP address assignment with relay agent information option on page 561

Common DHCP options

All FortiGate models come with predefined DHCP options. These DHCP options are widely used and required in most scenarios. The following DHCP options can be set straight from the *DHCP server* section of the *Edit Interface* dialog:

Option Code	Option Name	Purpose
***1**	*Netmask*	Assign subnet mask to the DHCP client.
***3**	*Default Gateway*	Assign default gateway to the DHCP client.
6	*DNS server*	Assign DNS server to the DHCP client.

Option Code	Option Name	Purpose
42	*NTP server*	Assign NTP server to the DHCP client.
***51**	*Lease time*	Lease time for the DHCP client.
138	*Wireless controllers*	Assign CAPWAP Access Controller addresses to the DHCP client.
150	*TFTP server(s)*	Assign TFTP server to the DHCP client.

The parameter marked with an asterisk (*) are mandatory and must be filled in.

Configuring the lease time

This configuration implements DHCP option code 51. The global lease time (measured in seconds, 300 - 864000) determines the length of time an IP address remains assigned to a client. Once the lease expires, the address is released for allocation to the next client that requests an IP address.

To configure the global lease time:

```
config system dhcp server
    edit <id>
        set interface <interface>
        set netmask <netmask>
        set lease-time <integer>
    next
end
```

The default lease time is seven days (604800 seconds). To have an unlimited lease time, set the value to zero.

The lease time can also be configured in the GUI in the *Lease time* field within the *DHCP server* section of the *Edit Interface* dialog.

Configuring the lease time for IP ranges

The lease time can be also be configured for an IP range. Measured in seconds, the range is similar to the global lease time (300 - 864000), but the default value is zero (0). If the default (0) is used for an IP range, it applies the global DHCP server lease time value.

To configure the lease time for an IP range:

```
config system dhcp server
    edit <id>
        config ip-range
            edit <id>
                set lease-time <integer>
            next
        end
    next
end
```

This setting can only be configured in the CLI.

Customizing DHCP lease backup during power cycles

FortiOS allows customization of the backup interval of DHCP leases during power cycles using the `dhcp-lease-backup-interval` command. This provides enhanced control and flexibility, ensuring lease preservation during events like outages or reboots. After a power cycle, expired IP addresses are released from the lease list and unexpired IP addresses are retained.

The backup interval can be set between 10 and 3600 seconds, with the default value being 60.

```
config system global
    set dhcp-lease-backup-interval <integer>
end
```

Breaking an address lease

If you need to end an IP address lease, you can break the lease. This is useful if you have limited addresses and longer lease times when some leases are no longer necessary, for example, with corporate visitors.

To break a lease:

```
# execute dhcp lease-clear <ip_address>
```

To break a lease for all IP addresses for the DHCP servers in the current VDOM:

```
# execute dhcp lease-clear all
```

Configuring NTP servers

This configuration implements DHCP option code 42. NTP server can be used by the client to synchronize their time which is very important as for many features to work, including scheduling, logging, and SSL-dependent features, the FortiOS system time must be accurate. This option specifies a list of the NTP servers available to the client by IP address.

To configure NTP servers:

```
config system dhcp server
    edit 2
        set ntp-service {local | default | specify}
        set ntp-server1 <class_ip>
        set ntp-server2 <class_ip>
        set ntp-server3 <class_ip>
    next
end
```

NTP servers can also be configured in the GUI in the *NTP server* field within the *DHCP server > Advanced* section of the *Edit Interface* dialog.

`ntp-service {local | default | specify}`	Set the option for assigning NTP servers to DHCP clients: • `local`: the IP address of the interface that the DHCP server is added to

becomes the client's NTP server IP address.
- `default`: clients are assigned the FortiGate's configured NTP servers.
- `specify`: specify up to three NTP servers in the DHCP server configuration.

Configuring TFTP servers

This configuration implements DHCP option code 150. TFTP server are used by VoIP phones to obtain the VoIP Configuration. You can configure multiple TFTP servers for a DHCP server. For example, you may want to configure a main TFTP server and a backup TFTP server.

The `tftp-server` command allows you to configure the TFTP servers, using either their hostnames or IP addresses. Separate multiple server entries with spaces.

To configure TFTP servers:

```
config system dhcp server
    edit <id>
        set interface <interface>
        set netmask <netmask>
        set tftp-server <hostname/IP address> <hostname/IP address>
    next
end
```

TFTP servers can also be configured in the GUI in the *TFTP server(s)* field within the *DHCP server > Advanced* section of the *Edit Interface* dialog.

Additional DHCP options

The FortiGate can be used to provide additional DHCP options that can be useful for different scenarios.

A few of the options are explained below:

- Option 82 on page 560
- Option 77 on page 560

To configure the DHCP options in the GUI:

1. Go to *Network > Interfaces*, click *Create New* or *Edit* the existing interface.
2. Enable *DHCP Server*.
3. Expand the *Advanced* section and select *Create New* under *Additional DHCP options*.
4. Select a predefined *Option code* from the list or select *Specify* to enter a custom *Option code*.
5. Configure the rest of the parameters as required and click *OK* to save the options.
6. Click *OK* to save the setting.

To configure the DHCP options in the CLI:

```
config system dhcp server
    edit <id>
        config options
            edit <integer>
                set code <integer>
                set type {hex | string | ip | fqdn}
                set value <string>
            next
        end
    next
end
```

Variable	Description
`code <integer>`	DHCP client option code (0 - 255, default = 0). See Dynamic Host Configuration Protocol (DHCP) and Bootstrap Protocol (BOOTP) Parameters for a list of possible options.
`type {hex | string | ip | fqdn}`	DHCP server option type (default = hex).
`value <string>`	DHCP server option value.
`ip <ip address>`	DHCP server option IP address. This option is only available when `type` is `ip`.

Example

To configure option 252 with value http://192.168.1.1/wpad.dat:

```
config system dhcp server
    edit <id>
        config options
            edit <id>
                set code 252
                set type hex
                set value 687474703a2f2f3139322e3136382e312e312f777061642e646174
            next
        end
    next
end
```

In the example above, `687474703a2f2f3139322e3136382e312e312f777061642e646174` is the hexadecimal equivalent of the ASCII text `http://192.168.1.1/wpad.dat`.

Option 82

The DHCP relay agent information option (option 82 in RFC 3046) helps protect the FortiGate against attacks such as spoofing (forging) of IP addresses and MAC addresses, and DHCP IP address starvation.

This option is disabled by default. However, when `dhcp-relay-service` is enabled, `dhcp-relay-agent-option` becomes enabled.

To configure the DHCP relay agent option:

```
config system interface
    edit <interface>
        set vdom root
        set dhcp-relay-service enable
        set dhcp-relay-ip <ip>
        set dhcp-relay-agent-option enable
        set vlanid <id>
    next
end
```

See IP address assignment with relay agent information option on page 561 for an example.

Option 77

This option can be used for User Class information (UCI) matching. When enabled, only DHCP requests with a matching UCI are served with the specified range.

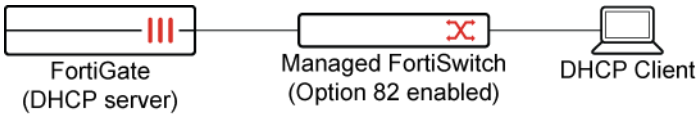
To configure UCI matching:

```
config system dhcp server
    edit <id>
        config ip-range
            edit <id>
                set uci-match {enable | disable}
                set uci-string <string>
            next
        end
        config options
            edit <id>
                set uci-match {enable | disable}
                set uci-string <string>
            next
        end
    next
end
```

`uci-match {enable | disable}`	Enable/disable User Class information (UCI) matching for option 77.
`uci-string <string>`	Enter one or more UCI strings in quotation marks separated by spaces.

IP address assignment with relay agent information option

Option 82 (DHCP relay information option) helps protect the FortiGate against attacks such as spoofing (or forging) of IP and MAC addresses, and DHCP IP address starvation.



The following CLI commands are included in the `config system dhcp server` > `config reserved-address` command:

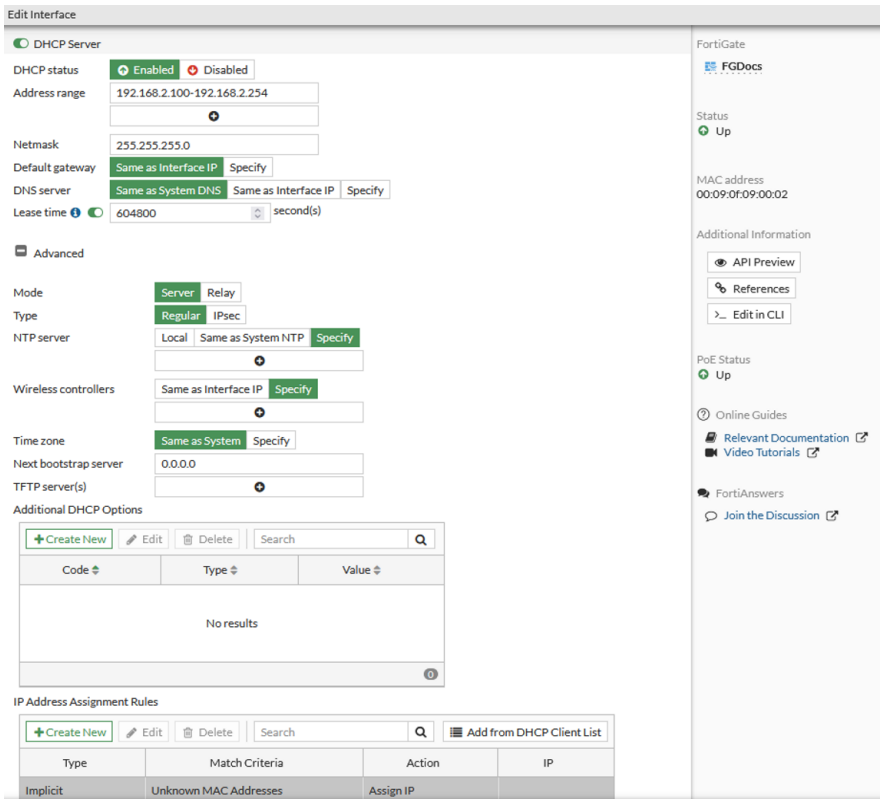
Command	Description
`circuit-id-type {hex | string}`	DHCP option type; hex or string (default).
`circuit-id <value>`	Option 82 circuit ID of the client that will get the reserved IP address. Format: *vlan-mod-port* • vlan: VLAN ID (2 bytes) • mod: 1 = snoop, 0 = relay (1 byte) • port: port number (1 byte)
`remote-id-type {hex | string}`	DHCP option type; hex or string (default).
`remote-id <value>`	Option 82 remote ID of the client that will get the reserved IP address. Format: the MAC address of the client.
`type {mac | option82}`	The DHCP reserved address type; mac (default) or option82.

To create an IP address assignment rule using option 82 in the GUI:

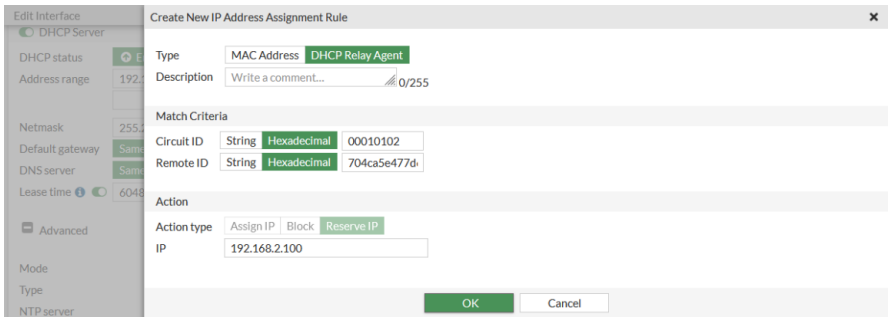
1. Go to *Network* > *Interfaces*.
2. Edit an existing port, or create a new one.

   The port *Role* must be *LAN* or *Undefined*.

3. Enable *DHCP Server*.
4. Configure the address ranges and other settings as needed.
5. Click *+* to expand the *Advanced* options.



6. In the *IP Address Assignment Rules* table, click *Create New*.
   The *Create New IP Address Assignment Rule* pane opens.
7. Configure the new rule:
   a. For the *Type*, select *DHCP Relay Agent*.
   b. Enter the *Circuit ID* and *Remote ID*.
   c. Enter the *IP* address that will be reserved.



8. Click *OK*.

To create an IP address assignment rule using option 82 with the CLI:

```
config system dhcp server
    edit 1
        set netmask 255.255.255.0
        set interface "port4"
        config ip-range
```

```
            edit 1
                set start-ip 192.168.2.100
                set end-ip 192.168.2.254
            next
        end
        config reserved-address
            edit 1
                set type option82
                set ip 192.168.2.100
                set circuit-id-type hex
                set circuit-id "00010102"
                set remote-id-type hex
                set remote-id "704ca5e477d6"
            next
        end
    next
end
```

DHCP addressing mode on an interface

Any FortiGate interface can be configured to obtain an IP address dynamically using DHCP. If you configure DHCP on an interface on the FortiGate, the FortiGate automatically broadcasts a DHCP request from the interface. The interface is configured with the IP address, any DNS server addresses, and the default gateway address that the DHCP server provides.

Configuring an interface as a DHCP Client

You can configure interface as a DHCP client.

To configure an interface as a DHCP client in the GUI:

1. Go to *Network* > *Interfaces*.
2. Edit an interface.
3. Select the *DHCP* option in the *Addressing mode*.
4. Configure the rest of the setting as required.
5. Click *OK*.

The following table describes the DHCP status information when DHCP is configured for an interface.

Field	Description
Status	Displays DHCP status messages as the interface connects to the DHCP server and gets addressing information. Status can be one of the following values: • *Initializing*: No activity. • *Connecting*: Interface attempts to connect to the DHCP server. • *Connected*: Interface retrieves an IP address, netmask, and other

Field	Description
	settings from the DHCP server. • *Failed*: Interface was unable to retrieve an IP address and other settings from the DHCP server.
Obtained IP/Netmask	The IP address and netmask leased from the DHCP server. This is only displayed if the *Status* is *Connected*.
Renew	Select this to renew the DHCP license for this interface. This is only displayed if the *Status* is *Connected*.
Expiry Date	The time and date when the leased IP address and netmask is no longer valid for the interface. The IP address is returned to the pool to be allocated to the next user request for an IP address. This is only displayed if the *Status* is *Connected*.
Default Gateway	The IP address of the gateway defined by the DHCP server. This is displayed only if the *Status* is *Connected*, and if *Retrieve default gateway from server* is enabled.
Acquired DNS	The DNS server IP defined by the DHCP server. This is displayed only if the *Status* is *Connected*.
Retrieve default gateway from server	Enable this to retrieve a default gateway IP address from the DHCP server. The default gateway is added to the static routing table.
Distance	Enter the administrative distance for the default gateway retrieved from the DHCP server. The administrative distance is an integer from 1 to 255 that specifies the relative priority of a route when there are multiple routes to the same destination. A lower administrative distance indicates a more preferred route.
Override internal DNS	Enable this to use the DNS addresses retrieved from the DHCP server instead of the DNS server IP addresses on the DNS page. When VDOMs are enabled, you can override the internal DNS only on the management VDOM.

To configure an interface as a DHCP client in the CLI:

```
config system interface
    edit <name>
        set mode dhcp
        set defaultgw {enable | disable}
        set distance <integer>
        set dns-server-override {enable | disable}
    next
end
```

Configuring the DHCP renew time

You can set a minimum DHCP renew time for an interface acting as a DHCP client. This option is available only when mode is set to DHCP.

To set the DHCP renew time:

```
config system interface
    edit <name>
        set vdom <vdom>
        set interface <interface>
        set mode dhcp
        set dhcp-renew-time <integer>
    next
end
```

The possible values for `dhcp-renew-time` are 300 to 605800 seconds (five minutes to seven days). To use the renew time that the server provides, set this entry to 0.

DHCP client options

When an interface is in DHCP addressing mode, DHCP client options can be configured in the CLI. For example, a vendor class identifier (usually DCHP client option 60) can be specified so that a request can be matched by a specific DHCP offer.

Multiple options can be configured, but any options not recognized by the DHCP server are discarded.

To configure client option 60 - vendor class identifier:

```
config system interface
    edit port1
        set vdom vdom1
        set mode dhcp
        config client-options
            edit 1
                set code 60
                set type hex
                set value aabbccdd
            next
        end
        set type physical
        set snmp-index 4
    next
end
```

Variable	Description
`code <integer>`	DHCP client option code (0 - 255, default = 0).

Variable	Description
	See Dynamic Host Configuration Protocol (DHCP) and Bootstrap Protocol (BOOTP) Parameters for a list of possible options.
`type {hex | string | ip | fqdn}`	DHCP client option type (default = hex).
`value <string>`	DHCP client option value.
`ip <ip>`	DHCP client option IP address. This option is only available when `type` is `ip`.

VCI pattern matching for DHCP assignment

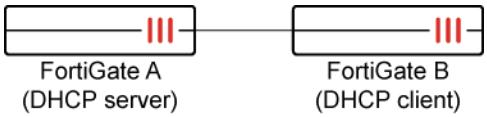
VCIs (vendor class identifiers) are supported in DHCP to allow VCI pattern matching as a condition for IP or DHCP option assignment. A single IP address, IP ranges of a pool, and dedicated DHCP options can be mapped to a specific VCI string.

```
config system dhcp server
    edit <id>
        config ip-range
            edit <id>
                set vci-match {enable | disable}
                set vci-string <string>
            next
        end
        config options
            edit <id>
                set vci-match {enable | disable}
                set vci-string <string>
            next
        end
    next
end
```

`vci-match {enable | disable}`	Enable/disable VCI matching. When enabled, only DHCP requests with a matching VCI are served with this range.
`vci-string <string>`	Set the VCI string. Enter one or more VCI strings in quotation marks separated by spaces.

Example

In this example, any DHCP client that matches the FortiGate-201F VCI will get their IP from the pool of 10.2.2.133-10.2.2.133, and options 42 (NTP servers) and 150 (TFTP server address). Any DHCP client that matches the FortiGate-101F VCI will get their IP from the default pool (10.2.2.132-10.2.2.132/10.2.2.134-10.2.2.254) and only get the 150 option.



To configure VCI pattern matching on FortiGate A:

```
config system dhcp server
    edit 1
        set dns-service default
        set default-gateway 10.2.2.131
        set netmask 255.255.255.0
        set interface "port3"
        config ip-range
            edit 1
                set start-ip 10.2.2.132
                set end-ip 10.2.2.132
            next
            edit 2
                set start-ip 10.2.2.133
                set end-ip 10.2.2.133
                set vci-match enable
                set vci-string "FortiGate-201F"
            next
            edit 3
                set start-ip 10.2.2.134
                set end-ip 10.2.2.254
            next
        end
        config options
            edit 1
                set code 42
                set type ip
                set vci-match enable
                set vci-string "FortiGate-201F"
                set ip "8.8.8.8"
            next
            edit 2
                set code 150
                set type ip
                set ip "172.16.200.55"
            next
        end
        set vci-match enable
        set vci-string "FortiGate-201F" "FortiGate-101F"
    next
end
```

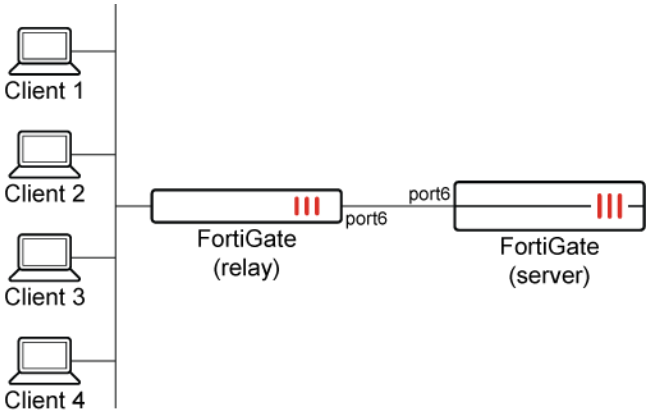
DHCP shared subnet

A FortiGate can act as a DHCP server and assign IP addresses from different subnets to clients on the same interface or VLAN based on the requests coming from the same DHCP relay agent. A FortiGate may have more than one server and pool associated with the relay agent, and it can assign IP addresses from the next server when the current one is exhausted. This way, the FortiGate can allocate IP addresses more efficiently and avoid wasting unused addresses in each subnet.

```
config system dhcp server
    edit <id>
        set shared-subnet {enable | disable}
        set relay-agent <ip_address>
    next
end
```

Example

In this example, there are two DHCP servers configured on the FortiGate. The first two clients (1 and 2) get their IP from the DHCP server 1. Once the DHCP server 1's IP pool is exhausted, subsequent clients (3 and 4) get their IP from DHCP server 2.



To configure a DHCP shared subnet:

1. Configure the DHCP servers:

```
config system dhcp server
    edit 1
        set default-gateway 10.18.0.10
        set netmask 255.255.255.0
        set interface "p2_vl3819"
        config ip-range
            edit 1
                set start-ip 10.18.0.110
                set end-ip 10.18.0.111
            next
```

```
        end
        set shared-subnet enable
        set relay-agent 10.18.0.10
        set dns-server1 8.8.8.8
    next
    edit 2
        set default-gateway 10.18.1.130
        set netmask 255.255.255.128
        set interface "p2_vl3819"
        config ip-range
            edit 1
                set start-ip 10.18.1.200
                set end-ip 10.18.1.201
            next
        end
        set shared-subnet enable
        set relay-agent 10.18.0.10
        set dns-server1 8.8.8.8
    next
end
```

2. Verify the DHCP lease list:

```
# execute dhcp lease-list
port6
  IP           MAC-Address      Hostname    VCI    SSID    AP    SERVER-ID    Expiry
  10.18.0.110  00:50:56:02:92:11                                 1            Fri Jan 13 15:37:35
2023
  10.18.0.111  00:50:56:02:92:12                                 1            Fri Jan 13 15:37:38
2023

Result: PASS
```

Clients 1 and 2 get their IP from the DHCP server 1.

When the IP pool is exhausted, the DHCP daemon assigns the IP from other pools that have the same relay agent.

3. Verify the DHCP lease list:

```
# execute dhcp lease-list
port6
  IP           MAC-Address      Hostname    VCI    SSID    AP    SERVER-ID    Expiry
  10.18.0.110  00:50:56:02:92:11                                 1            Fri Jan 13 15:37:35
2023
  10.18.0.111  00:50:56:02:92:12                                 1            Fri Jan 13 15:37:38
2023
  10.18.1.200  00:50:56:02:92:13                                 2            Fri Jan 13 15:38:05
2023
  10.18.1.201  00:50:56:02:92:14                                 2            Fri Jan 13 15:38:06
2023
```

Clients 3 and 4 get their IP from DHCP server 2, since the server 1 IP pool is exhausted.

Multiple DHCP relay servers

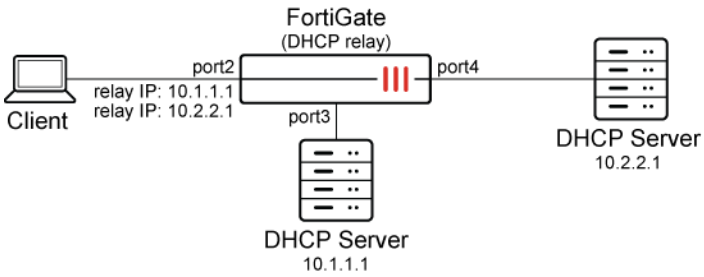
Multiple DHCP relays can be configured on an interface. After receiving a DHCP request from a client, the FortiGate forwards it to all configured servers simultaneously without waiting for any response. Each server sends back an IP address for the client, but the client usually uses the IP address from the first response that is receives.

This allows the FortiGate to forward DHCP requests to all configured servers simultaneously, reducing wait times and potential bottlenecks.

```
config system interface
    edit <name>
        set dhcp-relay-service {enable | disable}
        set dhcp-relay-ip <ip-address>
    next
end
```

Example

In this example, two DHCP relay servers are configured on port2, with DHCP relay IP addresses 10.1.1.1 and 10.2.2.1.



To configure the DHCP relay servers:

```
config system interface
    edit "port2"
        set dhcp-relay-service enable
        set dhcp-relay-ip 10.1.1.1 10.2.2.1
    next
end
```

To check the debug messages to verify that the DHCP relay is working:

```
# diagnose debug application dhcprelay -1
```

```
Debug messages will be on for 30 minutes.
# (xid:d7d00b58) L2 socket: received request message from 0.0.0.0:68 to 255.255.255.255 at port2
(xid:d7d00b58) got a DHCPDISCOVER
(xid:d7d00b58) Warning! can't get server id from client message
```

```
Insert option(82), len(7)
found route to 10.1.1.1 via 10.1.1.254 iif=6 oif=9/port3, mode=auto, ifname=
(xid:d7d00b58) forwarding dhcp request from 10.10.10.12:67 to 10.1.1.1:67
found route to 10.2.2.1 via 10.2.2.254 iif=6 oif=11/port4, mode=auto, ifname=
(xid:d7d00b58) forwarding dhcp request from 10.10.10.12:67 to 10.2.2.1:67
(xid:d7d00b58) got a DHCPOFFER
(xid:d7d00b58) from server 10.1.1.1
(xid:d7d00b58) sending dhcp reply from 10.10.10.12:67 to 255.255.255.255:68
(xid:d7d00b58) L2 socket: received request message from 0.0.0.0:68 to 255.255.255.255 at port2
(xid:d7d00b58) got a DHCPREQUEST
Insert option(82), len(7)
found route to 10.1.1.1 via 10.1.1.254 iif=6 oif=9/port3, mode=auto, ifname=
(xid:d7d00b58) forwarding dhcp request from 10.10.10.12:67 to 10.1.1.1:67
found route to 10.2.2.1 via 10.2.2.254 iif=6 oif=11/port4, mode=auto, ifname=
(xid:d7d00b58) forwarding dhcp request from 10.10.10.12:67 to 10.2.2.1:67
(xid:d7d00b58) got a DHCPOFFER
(xid:d7d00b58) from server 10.2.2.1
(xid:d7d00b58) sending dhcp reply from 10.10.10.12:67 to 255.255.255.255:68
(xid:d7d00b58) got a DHCPACK
(xid:d7d00b58) from server 10.1.1.1
(xid:d7d00b58) sending dhcp reply from 10.10.10.12:67 to 255.255.255.255:68
```

The debug output shows the following information:

`got a DHCPDISCOVER`
`forwarding dhcp request from 10.10.10.12:67 to 10.1.1.1:67`
`forwarding dhcp request from 10.10.10.12:67 to 10.2.2.1:67`
	FortiGate received a DHCPDISCOVER message from the DHCP client and forwarded it to both DHCP servers.

`got a DHCPOFFER` `from server 10.1.1.1`	FortiGate received a DHCPOFFER message from server 10.1.1.1.

`got a DHCPREQUEST`
`forwarding dhcp request from 10.10.10.12:67 to 10.1.1.1:67`
`forwarding dhcp request from 10.10.10.12:67 to 10.2.2.1:67`
	FortiGate received a DHCPREQUEST message from the client and forwarded it to both servers again.

`got a DHCPOFFER` `from server 10.2.2.1`	FortiGate received another DHCPOFFER message from server 10.2.2.1.
`got a DHCPACK` `from server 10.1.1.1`	FortiGate received a DHCPACK message from server 10.1.1.1. Because the DHCP server 10.1.1.1 was the first to send response, the client accepts the DHCP configuration from this server.

DHCP smart relay on interfaces with a secondary IP

DHCP relays can be configured on interfaces with secondary IP addresses. The FortiGate will track the number of unanswered DHCP requests for a client on the interface's primary IP. After three unanswered DHCP requests,

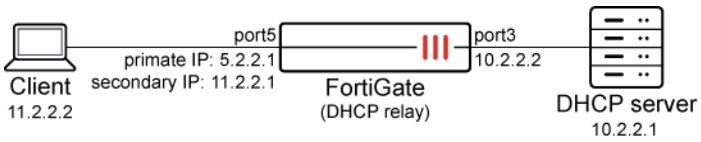
the FortiGate will forward DHCP requests to DHCP relays configured under the secondary IP using the secondary IP address as the source. After three unanswered DHCP requests, the FortiGate will return to using the primary IP and restart the process.

```
config system interface
    edit <name>
        set dhcp-smart-relay {enable | disable}
        config secondaryip
            edit <id>
                set secip-relay-ip <secondary_dhcp_relay_IP_1> <secondary_dhcp_relay_IP_2>
            next
        end
    next
end
```

DHCP relay targets under both the primary and secondary IP may be the same or unique. If smart relay is not configured, all requests are forwarded using the primary IP address on the interface.

Example

In this example, DHCP smart relay is configured on port5 with a DHCP relay IP address of 10.2.2.1.



To configure DHCP smart relay on interfaces with a secondary IP:

1. Configure DHCP relay on the interfaces:

```
config system interface
    edit "port3"
        set vdom "vdom1"
        set ip 10.2.2.2 255.255.255.0
        set allowaccess ping https ssh snmp http telnet
        set type physical
        set snmp-index 5
    next
    edit "port5"
        set vdom "vdom1"
        set dhcp-relay-service enable
        set dhcp-smart-relay enable
        set ip 5.2.2.1 255.255.255.0
        set allowaccess ping https ssh snmp http
        set type physical
        set snmp-index 7
        set secondary-IP enable
        set dhcp-relay-ip "10.2.2.1"
        config secondaryip
            edit 1
```

```
                set ip 11.2.2.1 255.255.255.0
                set secip-relay-ip "10.2.2.1"
                set allowaccess ping https ssh snmp http
            next
        end
    next
end
```

2. Verify the debug messages to check that the DHCP relay is working. After three unanswered DHCP requests, the request is forwarded to the secondary IP DHCP relay target:

```
# diagnose debug application dhcprelay -1
Debug messages will be on for 30 minutes.

(xid:7ea80e4b) received request message from 0.0.0.0:68 to 255.255.255.255 at port5
(xid:7ea80e4b) got a DHCPDISCOVER
(xid:7ea80e4b) Warning! can't get server id from client message
Insert option(82), len(7)
found route to 10.2.2.1 via 10.2.2.2 iif=11 oif=9/port3, mode=auto, ifname=
(xid:7ea80e4b) forwarding dhcp request from 5.2.2.1:67 to 10.2.2.1:67
(xid:7ea80e4b) received request message from 0.0.0.0:68 to 255.255.255.255 at port5
(xid:7ea80e4b) got a DHCPDISCOVER
(xid:7ea80e4b) Warning! can't get server id from client message
Insert option(82), len(7)
found route to 10.2.2.1 via 10.2.2.2 iif=11 oif=9/port3, mode=auto, ifname=
(xid:7ea80e4b) forwarding dhcp request from 5.2.2.1:67 to 10.2.2.1:67
(xid:7ea80e4b) received request message from 0.0.0.0:68 to 255.255.255.255 at port5
(xid:7ea80e4b) got a DHCPDISCOVER
(xid:7ea80e4b) Warning! can't get server id from client message
Insert option(82), len(7)
found route to 10.2.2.1 via 10.2.2.2 iif=11 oif=9/port3, mode=auto, ifname=
(xid:7ea80e4b) forwarding dhcp request from 11.2.2.1:67 to 10.2.2.1:67
(xid:7ea80e4b) received request message from 10.2.2.1:67 to 11.2.2.1 at port3
(xid:7ea80e4b) got a DHCPOFFER
(xid:7ea80e4b) from server 10.2.2.1
(xid:7ea80e4b) sending dhcp reply from 11.2.2.1:67 to 255.255.255.255:68
(xid:7ea80e4b) received request message from 0.0.0.0:68 to 255.255.255.255 at port5
(xid:7ea80e4b) got a DHCPREQUEST
Insert option(82), len(7)
found route to 10.2.2.1 via 10.2.2.2 iif=11 oif=9/port3, mode=auto, ifname=
(xid:7ea80e4b) forwarding dhcp request from 11.2.2.1:67 to 10.2.2.1:67
(xid:7ea80e4b) received request message from 10.2.2.1:67 to 11.2.2.1 at port3
(xid:7ea80e4b) got a DHCPACK
(xid:7ea80e4b) from server 10.2.2.1
(xid:7ea80e4b) sending dhcp reply from 11.2.2.1:67 to 255.255.255.255:68
```

FortiGate DHCP works with DDNS to allow FQDN connectivity to leased IP addresses

As clients are assigned IP addresses, they send back information that would be found in an A record to the FortiGate DHCP server, which can take this information and pass it back to a corporate DNS server so that even devices using leased IP address can be reached using FQDNs. You can configure the settings for this feature using the `ddns-update` CLI command and some other DDNS related options. Please refer to *DDNS update override* in the DDNS on page 385 topic for further details.

Static routing

Static routing is one of the foundations of firewall configuration. It is a form of routing in which a device uses manually-configured routes. In the most basic setup, a firewall will have a default route to its gateway to provide network access. In a more complex setup with dynamic routing, ADVPN, or SD-WAN involved, you would still likely find static routes being deployed.

This section explores concepts in using static routing and provides examples in common use cases:

- Routing concepts on page 575
- Policy routes on page 590
- Equal cost multi-path on page 595
- Dual internet connections on page 599

The following topics include additional information about static routes:

- Deploying the Security Fabric on page 4423
- Security Fabric over IPsec VPN on page 4445
- Routing on page 1043
- IPsec VPN in an HA environment on page 3198
- IPsec VPN to Azure with virtual network gateway on page 2942
- FortiGate as dialup client on page 2962
- ADVPN with BGP as the routing protocol on page 3244
- ADVPN with OSPF as the routing protocol on page 3255
- ADVPN with RIP as the routing protocol on page 3264
- Basic site-to-site VPN with pre-shared key on page 2857
- Basic site-to-site VPN with digital certificate on page 2867
- Site-to-site VPN with overlapping subnets on page 2879
- Tunneled Internet browsing on page 2891
- Multiple concurrent SDN connectors on page 4671
- Packet distribution and redundancy for aggregate IPsec tunnels on page 3205
- Using BGP tags with SD-WAN rules on page 1434

Routing concepts

This section contains the following topics:

- Default route on page 575
- Adding or editing a static route on page 575
- Configuring FQDNs as a destination address in static routes on page 576
- Routing table on page 577
- Viewing the routing database on page 580
- Kernel routing table on page 580
- Route look-up on page 582
- Blackhole routes on page 582
- Reverse path look-up on page 583
- Asymmetric routing on page 584
- Routing changes on page 586
- Static route tags on page 587
- Defining a preferred source IP for local-out egress interfaces on page 588

Default route

The default route has a destination of `0.0.0.0/0.0.0.0`, representing the least specific route in the routing table. It is a catch all route in the routing table when traffic cannot match a more specific route. Typically this is configured with a static route with an administrative distance of `10`. In most instances, you will configure the next hop interface and the gateway address pointing to your next hop. If your FortiGate is sitting at the edge of the network, your next hop will be your ISP gateway. This provides internet access for your network.

Sometimes the default route is configured through DHCP. On some entry-level models, the WAN interface is preconfigured in DHCP mode. Once the WAN interface is plugged into the network modem, it will receive an IP address, default gateway, and DNS server. FortiGate will add this default route to the routing table with a distance of `5`, by default. This will take precedence over any default static route with a distance of `10`. Therefore, take caution when you are configuring an interface in DHCP mode, where *Retrieve default gateway from server* is enabled. You may disable it and/or change the distance from the *Network* > *Interfaces* page when you edit an interface.

Adding or editing a static route

To add a static route using the GUI:

1. Go to *Network* > *Static Routes* and click *Create New*.
2. Enter the following information:

Destination	• Subnet Enter the destination IP address and netmask. A value of `0.0.0.0/0.0.0.0` creates a default route. • Named Address

	Select an address or address group object. Only addresses with static route configuration enabled will appear on the list. This means a geography type address cannot be used. • Internet Service Select an Internet Service. These are known IP addresses of popular services across the Internet.
Interface	Select the name of the interface that the static route will connect through.
Gateway Address	Enter the gateway IP address. If a DHCP/PPPoE interface is selected, select *Dynamic* to automatically retrieve the interface's dynamic gateway, or select *Specify* to manually enter the gateway IP address. If an IPsec VPN interface or SD-WAN creating a blackhole route is selected, the gateway cannot be specified.
Administrative Distance	Enter the distance value, which will affect which routes are selected first by different protocols for route management or load balancing. The default is 10.
Advanced Options	Optionally, expand *Advanced Options* and enter a *Priority*. When two routes have an equal distance, the route with a lower priority number will take precedence. The default is 1.

3. Click *OK*.

Configuring FQDNs as a destination address in static routes

You can configure FQDN firewall addresses as destination addresses in a static route, using either the GUI or the CLI.

In the GUI, to add an FQDN firewall address to a static route in the firewall address configuration, enable the *Static Route Configuration* option. Then, when you configure the static route, set *Destination* to *Named Address*.

To configure an FQDN as a destination address in a static route using the CLI:

```
config firewall address
    edit 'Fortinet-Documentation-Website'
        set type fqdn
        set fqdn docs.fortinet.com
        set allow-routing enable
    next
end
```

```
config router static
    edit 0
        set dstaddr Fortinet-Documentation-Website
        ...
```

```
    next
end
```

Routing table

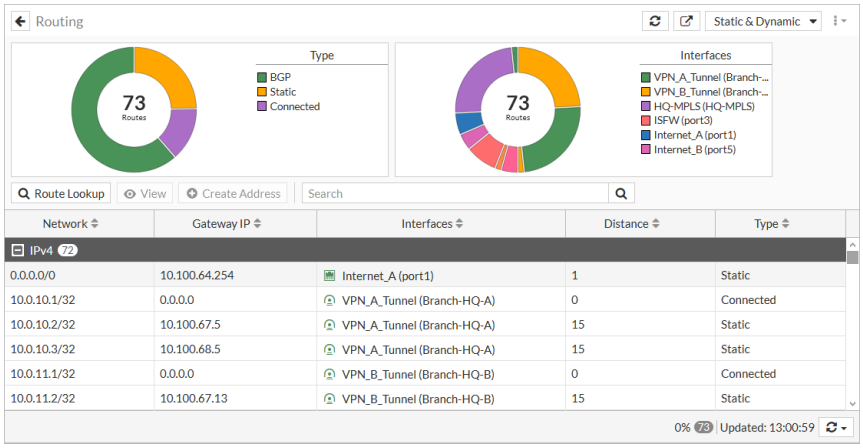
A routing table consists of only the best routes learned from the different routing protocols. The most specific route always takes precedence. If there is a tie, then the route with a lower administrative distance will be injected into the routing table. If administrative distances are also equal, then all the routes are injected into the routing table, and *Cost* and *Priority* become the deciding factors on which a route is preferred. If these are also equal, then FortiGate will use Equal cost multi-path on page 595 to distribute traffic between these routes.

Viewing the routing table in the GUI

You can view routing tables in the FortiGate GUI under *Dashboard* > *Network* > *Static & Dynamic Routing* by default. Expand the widget to see the full page. Additionally, if you want to convert the widget into a dashboard, click on the *Save as Monitor* icon on the top right of the page.

You can also monitor policy routes by toggling from *Static & Dynamic* to *Policy* on the top right corner of the page. The active policy routes include policy routes that you created, SD-WAN rules, and Internet Service static routes. It also supports downstream devices in the Security Fabric.

The following figure show an example of the static and dynamic routes in the Routing Monitor:



To view more columns, right-click on the column header to select the columns to be displayed:

Field	Description
IP Version	Shows whether the route is IPv4 or IPv6.
Network	The IP addresses and network masks of destination networks that the FortiGate can reach.
Gateway IP	The IP addresses of gateways to the destination networks.
Interfaces	The interface through which packets are forwarded to the gateway of the destination network.

Field	Description
Distance	The administrative distance associated with the route. A lower value means the route is preferable compared to other routes to the same destination.
Type	The type values assigned to FortiGate routes (Static, Connected, RIP, OSPF, or BGP): • *Connected*: All routes associated with direct connections to FortiGate interfaces • *Static*: The static routes that have been added to the routing table manually • *RIP*: All routes learned through RIP • *RIPNG*: All routes learned through RIP version 6 (which enables the sharing of routes through IPv6 networks) • *BGP*: All routes learned through BGP • *OSPF*: All routes learned through OSPF • *OSPF6*: All routes learned through OSPF version 6 (which enables the sharing of routes through IPv6 networks) • *IS-IS*: All routes learned through IS-IS • *HA*: RIP, OSPF, and BGP routes synchronized between the primary unit and the subordinate units of a high availability (HA) cluster. HA routes are maintained on subordinate units and are visible only if you're viewing the router monitor from a virtual domain that is configured as a subordinate virtual domain in a virtual cluster.
Metric	The metric associated with the route type. The metric of a route influences how the FortiGate dynamically adds it to the routing table. The following are types of metrics and the protocols they are applied to: • *Hop count*: Routes learned through RIP • *Relative cost*: Routes learned through OSPF • *Multi-Exit Discriminator (MED)*: Routes learned through BGP. By default, the MED value associated with a BGP route is zero. However, the MED value can be modified dynamically. If the value was changed from the default, the Metric column displays a non-zero value.
Priority	In static routes, priorities are 1 by default. When two routes have an equal distance, the route with the lower priority number will take precedence.
VRF	Virtual routing and forwarding (VRF) allows multiple routing table instances to co-exist. VRF can be assigned to an Interface. Packets are only forwarded between interfaces with the same VRF.
Up Since	The total accumulated amount of time that a route learned through RIP, OSPF, or BGP has been reachable.

Viewing the routing table in the CLI

Viewing the routing table using the CLI displays the same routes as you would see in the GUI.

If VDOMs are enabled on the FortiGate, all routing-related CLI commands must be run within a VDOM and not in the global context.

To view the routing table using the CLI:

```
# get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2

E1 - OSPF external type 1, E2 - OSPF external type 2
 i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
 * - candidate default
Routing table for VRF=0
S*      0.0.0.0/0 [1/0] via 172.31.0.1, MPLS [1/0]via 192.168.2.1, port1 [1/0] via 192.168.122.1,
port2
S       1.2.3.4/32 [10/0] via 172.16.100.81, VLAN100
C       10.10.2.0/24 is directly connected, hub
C       10.10.2.1/32 is directly connected, hub
O       10.10.10.0/24 [110/101] via 192.168.2.1, port1, 01:54:18
C       10.253.240.0/20 is directly connected, wqt.root
S       110.2.2.122/32 [22/0] via 2.2.2.2, port2, [3/3]
C       172.16.50.0/24 is directly connected, WAN1-VLAN50
C       172.16.60.0/24 is directly connected, WAN2-VLAN60
C       172.16.100.0/24 is directly connected, VLAN100
C       172.31.0.0/30 is directly connected, MPLS
C       172.31.0.2/32 is directly connected, MPLS
B       192.168.0.0/24 [20/0] via 172.31.0.1, MPLS, 00:31:43
C       192.168.2.0/24 is directly connected, port1
C       192.168.20.0/24 is directly connected, port3
C       192.168.99.0/24 is directly connected, Port1-VLAN99
C       192.168.122.0/24 is directly connected, port2
Routing table for VRF=10
C       172.16.101.0/24 is directly connected, VLAN101
```

Examining an entry:

```
B       192.168.0.0/24 [20/0] via 172.31.0.1, MPLS, 00:31:43
```

Value	Description
`B`	BGP. The routing protocol used.
`192.168.0.0/24`	The destination of this route, including netmask.
`[20/0]`	`20` indicates an administrative distance of `20` out of a range of `0` to `255`. `0` is an additional metric associated with this route, such as in OSPF.
`172.31.0.1`	The gateway or next hop.
`MPLS`	The interface that the route uses.
`00:31:43`	The age of the route in `HH:MM:SS`.

Viewing the routing database

The routing database consists of all learned routes from all routing protocols before they are injected into the routing table. This likely lists more routes than the routing table as it consists of routes to the same destinations with different distances. Only the best routes are injected into the routing table. However, it is useful to see all learned routes for troubleshooting purposes.

To view the routing database using the CLI:

```
# get router info routing-table database
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area

N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
     E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       > - selected route, * - FIB route, p - stale info
Routing table for VRF=0
S    *> 0.0.0.0/0 [1/0] via 172.31.0.1, MPLS
     *>           [1/0] via 192.168.2.1, port1
     *>           [1/0] via 192.168.122.1, port2
S    *> 1.2.3.4/32 [10/0] via 172.16.100.81, VLAN100
C    *> 10.10.2.0/24 is directly connected, hub
C    *> 10.10.2.1/32 is directly connected, hub
O    *> 10.10.10.0/24 [110/101] via 192.168.2.1, port1, 02:10:17
C    *> 10.253.240.0/20 is directly connected, wqt.root
S    *> 110.2.2.122/32 [22/0] via 2.2.2.2, port2, [3/3]
C    *> 172.16.50.0/24 is directly connected, WAN1-VLAN50
C    *> 172.16.60.0/24 is directly connected, WAN2-VLAN60
C    *> 172.16.100.0/24 is directly connected, VLAN100
O       172.31.0.0/30 [110/201] via 192.168.2.1, port1, 00:47:36
C    *> 172.31.0.0/30 is directly connected, MPLS
```

Selected routes are marked by the > symbol. In the above example, the OSPF route to destination `172.31.0.0/30` is not selected.

Kernel routing table

The kernel routing table makes up the actual Forwarding Information Base (FIB) that used to make forwarding decisions for each packet. The routes here are often referred to as kernel routes. Parts of this table are derived from the routing table that is generated by the routing daemon.

To view the kernel routing table using the CLI:

```
# get router info kernel
tab=254 vf=0 scope=0 type=1 proto=11 prio=0 0.0.0.0/0.0.0.0/0->0.0.0.0/0 pref=0.0.0.0
        gwy=172.31.0.1 flag=04 hops=0 oif=31(MPLS)
        gwy=192.168.2.1 flag=04 hops=0 oif=3(port1)
        gwy=192.168.122.1 flag=04 hops=0 oif=4(port2)
```

```
tab=254 vf=0 scope=0 type=1 proto=17 prio=0 192.168.122.98/255.255.255.255/0->1.1.1.1/32
pref=0.0.0.0 gwy=192.168.122.1 dev=4(port2)
tab=254 vf=0 scope=0 type=1 proto=17 prio=0 172.31.0.2/255.255.255.255/0->1.1.1.1/32 pref=0.0.0.0
gwy=172.31.0.1 dev=31(MPLS)
tab=254 vf=0 scope=0 type=1 proto=17 prio=0 192.168.2.5/255.255.255.255/0->1.1.1.1/32 pref=0.0.0.0
gwy=192.168.2.1 dev=3(port1)
tab=254 vf=0 scope=0 type=1 proto=11 prio=0 0.0.0.0/0.0.0.0/0->1.2.3.4/32 pref=0.0.0.0
gwy=172.16.100.81 dev=20(VLAN100)
tab=254 vf=0 scope=0 type=1 proto=17 prio=0 192.168.122.98/255.255.255.255/0->8.8.8.8/32
pref=0.0.0.0 gwy=192.168.122.1 dev=4(port2)
```

The kernel routing table entries are:

Value	Description
`tab`	Table number: It will either be `254` (unicast) or `255` (multicast).
`vf`	Virtual domain of the firewall: It is the VDOM index number. If VDOMs are not enabled, this number is `0`.
`type`	Type of routing connection. Valid values include: • `0` - unspecific • `1` - unicast • `2` - local • `3` - broadcast • `4` - anycast • `5` - multicast • `6` - blackhole • `7` - unreachable • `8` - prohibited
`proto`	Type of installation that indicates where the route came from. Valid values include: • `0` - unspecific • `2` - kernel • `11` - ZebOS routing module • `14` - FortiOS • `15` - HA • `16` - authentication based • `17` - HA1
`prio`	Priority of the route. Lower priorities are preferred.
`->0.0.0.0/0 (->x.x.x.x/mask)`	The IP address and subnet mask of the destination.
`pref`	Preferred next hop along this route.
`gwy`	Gateway: The address of the gateway this route will use.

Value	Description
dev	Outgoing interface index: This number is associated with the interface for this route. If VDOMs are enabled, the VDOM is also included here. If an interface alias is set for this interface, it is also displayed here.

Route look-up

Route look-up typically occurs twice in the life of a session. Once when the first packet is sent by the originator and once more when the first reply packet is sent from the responder. When a route look-up occurs, the routing information is written to the session table. If routing changes occur during the life of a session, additional routing look-ups may occur.

FortiGate performs a route look-up in the following order:

1. Policy-based routes: If a match occurs and the action is to forward, traffic is forwarded based on the policy route.
2. Forwarding Information Base, otherwise known as the kernel routing table.
3. If no match occurs, the packet is dropped.

Searching the routing table

When there are many routes in your routing table, you can perform a quick search by using the search bar to specify your criteria, or apply filters on the column header to display only certain routes. For example, if you want to only display static routes, you may use "static" as the search term, or filter by the *Type* field with value *Static*.

Route look-up on the other hand provides a utility for you to enter criteria such as *Destination*, *Destination Port*, *Source*, *Protocol* and/or *Source Interface*, in order to determine the route that a packet will take. Once you click *Search*, the corresponding route will be highlighted.

You can also use the CLI for a route look-up. The CLI provides a basic route look-up tool.

To look-up a route in the CLI:

```
# get router info routing-table details 4.4.4.4
Routing table for VRF=0
Routing entry for 0.0.0.0/0
      Known via "static", distance 1, metric 0, best
      * 172.31.0.1, via MPLS distance 0
      * 192.168.2.1, via port1 distance 0
      * 192.168.122.1, via port2 distance 0
```

Blackhole routes

Sometimes upon routing table changes, it is not desirable for traffic to be routed to a different gateway. For example, you may have traffic destined for a remote office routed through your IPsec VPN interface. When the VPN is down, traffic will try to re-route to another interface. However, this may not be viable and traffic will

instead be routed to your default route through your WAN, which is not desirable. Traffic may also be routed to another VPN, which you do not want. For such scenarios, it is good to define a blackhole route so that traffic is dropped when your desired route is down. Upon reconnection, your desired route is once again added to the routing table and your traffic will resume routing to your desired interface. For this reason, blackhole routes are created when you configure an IPsec VPN using the IPsec wizard.

To create a blackhole route in the GUI:

1. Go to *Network* > *Static Routes*.
2. Click *Create New*. The *New Static Route* screen appears.
3. Specify a *Destination* type.
4. Select *Blackhole* from the *Interface* field.
5. Type the desired *Administrative Distance*.
6. Click *OK*.

Route priority for a *Blackhole* route can only be configured from the CLI.

Reverse path look-up

Whenever a packet arrives at one of the interfaces on a FortiGate, the FortiGate determines whether the packet was received on a legitimate interface by doing a reverse look-up using the source IP address in the packet header. This protects against IP spoofing attacks. If the FortiGate does not have a route to the source IP address through the interface on which the packet was received, the FortiGate drops the packet as per Reverse Path Forwarding (RPF) check. There are two modes of RPF: feasible path and strict. The default feasible RPF mode checks only for the existence of at least one active route back to the source using the incoming interface. The strict RPF check ensures the best route back to the source is used as the incoming interface.

To configure a strict Reverse Path Forwarding check:

```
config system settings
    set strict-src-check enable
end
```

For reply packets, use the following command:

```
config system settings
    set src-check-reply enable
end
```

You can remove RPF state checks without needing to enable asymmetric routing by disabling state checks for traffic received on specific interfaces. Disabling state checks makes a FortiGate less secure and should only be done with caution for troubleshooting purposes.

To remove Reverse Path Forwarding checks from the state evaluation process:

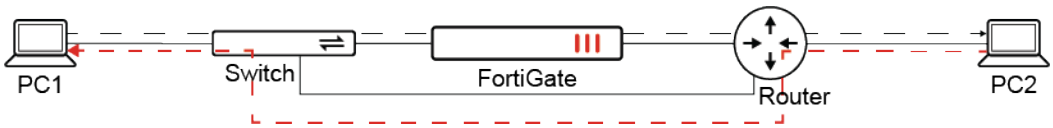
```
config system interface
    edit <interface_name>
```

```
        set src-check disable
    next
end
```

Asymmetric routing

Asymmetric routing occurs when request and response packets follow different paths that do not cross the same firewall.

In the following topology, traffic between PC1 and PC2 takes two different paths.



Traffic from PC1 to PC2 goes through the FortiGate, while traffic from PC2 to PC1 does not.

In TCP, if the packets in the request and response directions follow different paths, the FortiGate will block the packets, since the TCP three-way handshake is not established through the FortiGate.

Scenario 1: PC1 starts a TCP connection with PC2

1. The TCP SYN is allowed by the FortiGate.
2. The TCP SYN/ACK bypasses the FortiGate.
3. The TCP ACK is blocked by the FortiGate.
4. Subsequent TCP packets are blocked by the FortiGate.

Scenario 2: PC2 starts a TCP connection with PC1

1. The TCP SYN bypasses the FortiGate.
2. The TCP SYN/ACK is blocked by the FortiGate.
3. Subsequent TCP packets are blocked by the FortiGate.

In ICMP, consider the following scenarios.

Scenario 1: PC1 pings PC2

1. The ICMP request passes through the FortiGate. A session is created.
2. The ICMP reply bypasses the FortiGate, but reaches PC1. The ping is successful.
3. The ICMP request passes through the FortiGate, and it matches the previous session.
4. The ICMP reply bypasses the FortiGate, but it reaches PC1. The ping is successful.
5. Subsequent ICMP requests are allowed by the FortiGate.

Scenario 2: PC2 pings PC1

1. The ICMP request bypasses the FortiGate, but it reaches PC1.
2. The ICMP reply passes through the FortiGate. No session is matched, and the packet is dropped.
3. Subsequent ICMP replies are blocked by the FortiGate.

If an ICMP request does not pass through the FortiGate, but the response passes through the FortiGate, then by default it blocks the packet as invalid.

Permitting asymmetric routing

If required, the FortiGate can be configured to permit asymmetric routing.

To permit asymmetric routing:

```
config system settings
    set asymroute enable
end
```

This setting should be used only when the asymmetric routing issue cannot be resolved by ensuring both directions of traffic pass through the FortiGate.

When asymmetric routing is enabled and occurs, the FortiGate cannot inspect all traffic. Potentially malicious traffic may pass through and compromise the security of the network.

Asymmetric routing behaves as follows when it is permitted by the FortiGate:

TCP packets

Scenario 1: PC1 starts a TCP connection with PC2

1. The TCP SYN is allowed by the FortiGate. The FortiGate creates a session, checks the firewall policies, and applies the configuration from the matching policy (UTM inspection, NAT, traffic shaping, and so on).
2. The TCP SYN/ACK bypasses the FortiGate.
3. The TCP ACK is allowed by the FortiGate. The packet matches the previously created session.
4. Subsequent TCP packets are allowed by the FortiGate. The packets in the session can also be offloaded where applicable.

Scenario 2: PC2 starts a TCP connection with PC1

1. The TCP SYN bypasses the FortiGate.
2. The TCP SYN/ACK is allowed by the FortiGate. No session is matched. The packet passes to the CPU and is forwarded based on the routing table.
3. The TCP ACK bypasses the FortiGate.
4. Subsequent TCP packets are allowed by the FortiGate. The FortiGate acts as a router that only makes routing decisions. No security inspection is performed.

ICMP packets

Scenario 1: PC1 pings PC2

1. There is no difference from when asymmetric routing is disabled.

Scenario 2: PC2 pings PC1

1. The ICMP request bypasses the FortiGate, but it reaches PC1.
2. The ICMP reply passes through the FortiGate. No session is matched. The packet passes to the CPU and is forwarded based on the routing table.
3. Subsequent ICMP replies are allowed by the FortiGate. The FortiGate acts as a router that only makes routing decisions. No security inspection is performed.

UDP packets

Asymmetric routing does not affect UDP packets. UDP packets are checked by the session table regardless of asymmetric routing. A policy is required to allow UDP.

Routing changes

When routing changes occur, routing look-up may occur on an existing session depending on certain configurations.

Routing changes without SNAT

When a routing change occurs, FortiGate flushes all routing information from the session table and performs new routing look-up for all new packets on arrival by default. You can modify the default behavior using the following commands:

```
config system interface
    edit <interface>
        set preserve-session-route enable
    next
end
```

By enabling `preserve-session-route`, the FortiGate marks existing session routing information as persistent. Therefore, routing look-up only occurs on new sessions.

Routing changes with SNAT

When SNAT is enabled, the default behavior is opposite to that of when SNAT is not enabled. After a routing change occurs, sessions with SNAT keep using the same outbound interface as long as the old route is still active. This may be the case if the priority of the static route was changed. You can modify this default behavior using the following commands. Separate controls are available for IPv4 and IPv6:

For IPv4:

```
config system global
    set snat-route-change enable
end
```

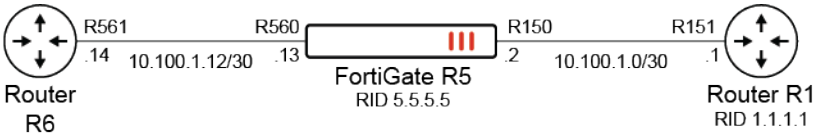
For IPv6:

```
config system global
    set ipv6-snat-route-change enable
end
```

By enabling `snat-route-change` or `ipv6-snat-route-change`, sessions with SNAT will require new route look-up when a routing change occurs. This will apply a new SNAT to the session.

Static route tags

When a static route is configured with a route tag, it is matched in the route map, and then used to set the route's metric and advertise to the BGP neighbor. In the following example, route tag 565 is used, and router R1 receives the advertised route from the FortiGate router R5.



To configure the FortiGate:

1. Configure the static route:

```
config router static
    edit 1
        set dst 77.7.7.7 255.255.255.255
        set distance 2
        set device "R560"
        set tag 565
    next
end
```

2. Configure the route map:

```
config router route-map
    edit "map1"
        config rule
            edit 2
                set match-tag 565
                set set-metric 2301
            next
        end
    next
end
```

3. Configure the BGP neighbor:

```
config router bgp
    config neighbor
        edit "10.100.1.2"
            set route-map-out "map1"
        next
```

```
    end
end
```

On its neighbor side, router R1 receives the advertised route from the FortiGate router R5.

4. Verify the BGP routing table:

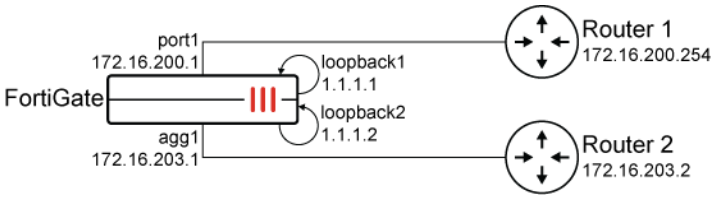
```
# get router info routing-table bgp
Routing table for VRF=0
B       77.7.7.7/32 [20/2301] via 10.100.1.1 (recursive is directly connected, R150),
03:18:53, [1/0]
```

5. Verify the network community:

```
# get router info bgp network 77.7.7.7/32
VRF 0 BGP routing table entry for 77.7.7.7/32
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
   2.2.2.2 3.3.3.3 10.100.1.5 2000::2:2:2:2
  Original VRF 0
  20
    10.100.1.1 from 10.100.1.1 (5.5.5.5)
      Origin incomplete metric 2301, localpref 200, valid, external, best
      Last update: Wed Oct  5 16:48:28 2022
```

Defining a preferred source IP for local-out egress interfaces

The preferred source IP can be configured on a static route so that local-out traffic is sourced from that IP. In the following example, a source IP is defined per static route. Local traffic that uses the static route will use the source IP instead of the interface IP associated with the route.



To configure preferred source IPs for static routes:

1. Configure the static routes:

```
config router static
    edit 22
        set dst 172.17.254.0 255.255.255.0
        set gateway 172.16.200.254
        set preferred-source 1.1.1.1
        set distance 2
        set device "port1"
    next
    edit 23
        set dst 172.17.254.0 255.255.255.0
```

```
        set gateway 172.16.203.2
        set preferred-source 1.1.1.2
        set distance 2
        set device "agg1"
    next
end
```

2. Configure the primary DNS server IP address:

```
config system dns
    set primary 172.17.254.148
end
```

To verify the configuration:

1. Verify the kernel routing table:

```
# get router info kernel
...
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0->172.17.254.0/24 pref=0.0.0.0
        gwy=172.16.200.254 flag=14 hops=0 oif=9(port1) pref=1.1.1.1
        gwy=172.16.203.2 flag=14 hops=0 oif=33(agg1) pref=1.1.1.2
```

2. Verify the routing table for 172.17.254.148:

```
# get router info routing-table details 172.17.254.148
Routing table for VRF=0
Routing entry for 172.17.254.0/24
  Known via "static", distance 2, metric 0, best
  * vrf 0 172.16.200.254, via port1, prefsrc 1.1.1.1
  * vrf 0 172.16.203.2, via agg1, prefsrc 1.1.1.2
```

3. Run a sniffer trace after some traffic passes:

```
# diagnose sniffer packet any "host 172.17.254.148" 4
interfaces=[any]
filters=[host 172.17.254.148]
1.319811 port1 out 1.1.1.1.1371 -> 172.17.254.148.53: udp 43
1.320095 port1 in 172.17.254.148.53 -> 1.1.1.1.1371: udp 310
1.921718 port1 out 1.1.1.1.1371 -> 172.17.254.148.53: udp 27
2.031520 port1 in 172.17.254.148.53 -> 1.1.1.1.1371: udp 213
```

When DNS traffic leaves the FortiGate and is routed through port1, the source address 1.1.1.1 is used.

Individual source IP commands, such as FortiManager source IP and FortiAnalyzer source IP, take precedence over these preferred-source configurations.

Policy routes

Policy routing allows you to specify an interface to route traffic. This is useful when you need to route certain types of network traffic differently than you would if you were using the routing table. You can use the incoming traffic's protocol, source or destination address, source interface, or port number to determine where to send the traffic.

When a packet arrives, the FortiGate starts at the top of the policy route list and attempts to match the packet with a policy. For a match to be found, the policy must contain enough information to route the packet. At a minimum, this requires either the outgoing interface to forward the traffic, or the gateway to route the traffic to, or both.

If one or both of these are not specified in the policy route, then the FortiGate searches the routing table to find the best active route that corresponds to the policy route:

- If only the outgoing interface is specified, FortiGate will look up the routing table to find the gateway, filtered by the outgoing interface.
- If only the gateway is specified, FortiGate will look up the routing table to find the outgoing interface, filtered by the gateway

If either of these cannot be found, then the policy route does not match the packet.

When both the outgoing interface and gateway are specified, the FortiGate must still find a route in the routing table ensuring that the gateway is routable over the outgoing interface. If a route cannot be found, then the policy route again does not match the packet.

In any of these scenarios, the FortiGate continues down the policy route list until it reaches the end. If no matches are found, then the FortiGate does a route lookup using the routing table.

Policy routes are sometimes referred to as Policy-based routes (PBR).

Configuring a policy route

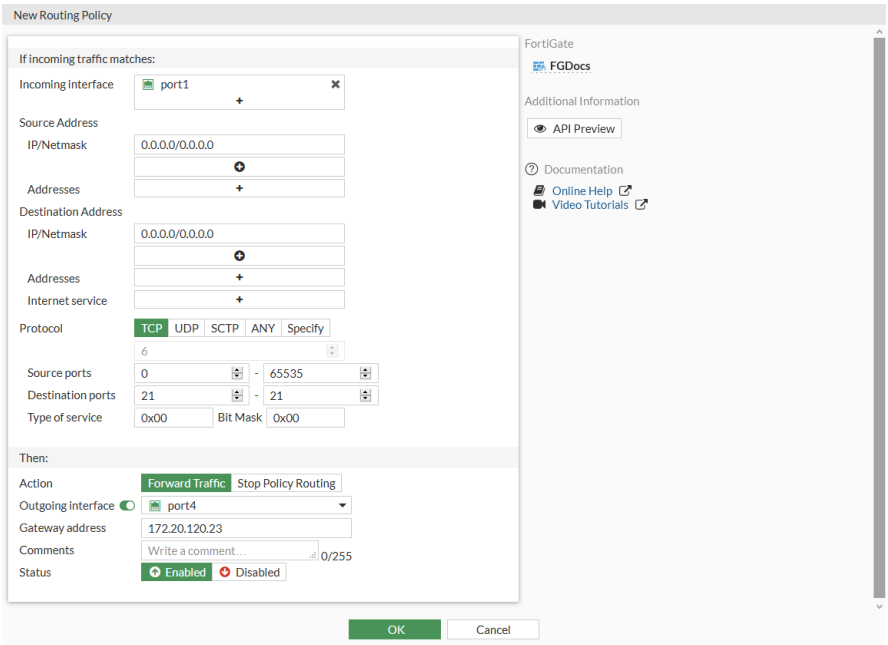
In this example, a policy route is configured to send all FTP traffic received at port1 out through port4 and to a next hop router at 172.20.120.23. To route FTP traffic, the protocol is set to TCP (6) and the destination ports are set to 21 (the FTP port).

To configure a policy route in the GUI:

1. Go to *Network* > *Policy Routes*.
2. Click *Create New* > *Policy Route*.
3. Configure the following fields:

Incoming interface	port1
Source Address	0.0.0.0/0.0.0.0
Destination Address	0.0.0.0/0.0.0.0
Protocol	TCP

Destination ports	21 - 21
Type of service	0x00
Bit Mask	0x00
Outgoing interface	Enable and select port4
Gateway address	172.20.120.23



4. Click *OK*.

To configure a policy route in the CLI:

```
config router policy
    edit 1
        set input-device "port1"
        set src "0.0.0.0/0.0.0.0"
        set dst "0.0.0.0/0.0.0.0"
        set protocol 6
        set start-port 21
        set end-port 21
        set gateway 172.20.120.23
        set output-device "port4"
        set tos 0x00
        set tos-mask 0x00
    next
end
```

Configuring users and groups in policy routes

Users and groups can be added in policy routes, allowing administrators to use users and user groups as source filters. This provides granular control over network traffic, enabling organizations to prioritize resources for specific users or groups.

To configure user and groups in policy routes:

1. Configure the users:

```
config user local
    edit "test1"
        set type password
        set passwd ********
    next
    edit "test2"
        set type password
        set passwd ********
    next
    edit "test3"
        set type password
        set passwd ********
    next
end
```

2. Configure the user group:

```
config user group
    edit "group1"
        set member "test1" "test2"
    next
end
```

3. Configure the IPv4 routing policy with both a user and a user group as source filters:

```
config router policy
    edit 22
        set dst "10.1.100.1/255.255.255.255"
        set gateway 172.16.200.55
        set output-device "port1"
        set users "test3"
        set groups "group1"
    next
end
```

4. Configure the IPv6 routing policy with two users as source filters:

```
config router policy6
    edit 23
        set dst "2000::55/128"
        set gateway 2000:172:16:200::2
        set output-device "port1"
```

```
        set users "test1" "test2"
    next
end
```

5. Verify that the IPv4 routing policy is using both the user and user group as source filters:

```
# diagnose firewall proute list
list route policy info(vf=root):

id=22(0x16) dscp_tag=0xfc 0xfc flags=0x0 tos=0x00 tos_mask=0x00 protocol=0 port=src(0->0):dst
(0->0) iif=0(any)
path(1): oif=7(port1) gwy=172.16.200.55
users and groups(2): test3(16777219) group1(1)
source wildcard(1): 0.0.0.0/0.0.0.0
destination wildcard(1): 10.1.100.1/255.255.255.255
hit_count=0 rule_last_used=2025-02-12 11:40:34
```

6. Verify that the IPv6 routing policy is using both of the assigned users as source filters:

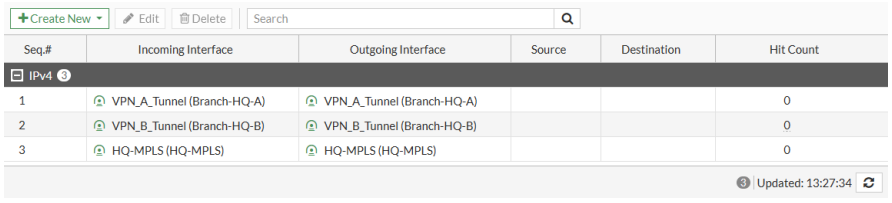
```
# diagnose firewall proute6 list
list route policy6 info(vf=root):

id=23(0x17) flags=0x0 tos=0x00 tos_mask=0x00 protocol=0 port=src(0->0):dst(0->0) iif=0(any)
path(1): oif=7(port1) gwy=2000:172:16:200::2
users and groups(2): test1(16777217) test2(16777218)
destination(1): 2000::55/128
source wildcard(1):  ADDR(::)/ADDR(::)
hit_count=0 last_used=2025-02-12 11:47:07
```

Moving a policy route

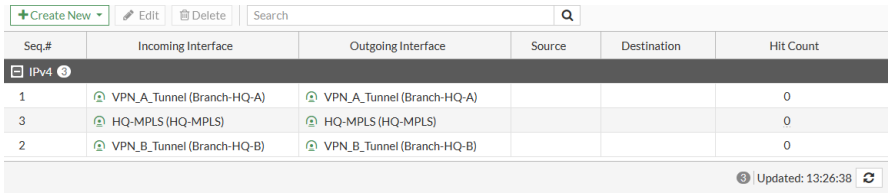
A routing policy is added to the bottom of the table when it is created. Routing policies can be moved to a different location in the table to change the order of preference. In this example, routing policy 3 will be moved before routing policy 2.

To move a policy route in the GUI:

1. Go to *Network* > *Policy Routes*.
2. In the table, select the policy route.



3. Drag the selected policy route to the desired position.



To move a policy route in the CLI:

```
config router policy
    move 3 after 1
end
```

ISDB route

An Internet service database (ISDB) route is a special kind of policy route that can be configured on the *Static Routes* page, where the destination is specifically defined as an Internet service from the ISDB.

To configure an ISDB route in the GUI:

1. Go to *Network* > *Static Routes* and click *Create new*.
2. Set *Destination* to *Internet Service* and select a custom or predefined service from the drop-down list.
3. Specify the *Gateway Address* and *Interface*.
4. Click *OK*.

To configure an ISDB route in the CLI:

```
config router static
    edit <id>
        set gateway <address of next hop>
        set device <outgoing interface>
        set internet-service <Internet service id>
    next
end
```

To check the order of the route with respect to other policy routes:

```
# diagnose firewall proute list
list route policy info(vf=root):
id=2113929222(0x7e000006) static_route=6 dscp_tag=0xfc 0xfc flags=0x0 tos=0x00 tos_mask=0x00
protocol=0 port=src(0->0):dst(1->65535) iif=0(any)
path(1): oif=3(port1) gwy=10.152.95.88
source wildcard(1): 0.0.0.0/0.0.0.0
destination wildcard(1): 0.0.0.0/0.0.0.0
internet service(1): BuiltIn-Alibaba-DingTalk(4278190117,0,0,0)
hit_count=0 rule_last_used=2025-02-12 09:38:36
```

Policy routes order of precedence

Policy routes can take different forms, including the policy-based, ISDB, and SD-WAN routes. When these routes are evaluated, the following order of precedence is applied:

Policy-based routes > ISDB routes > SD-WAN routes

Equal cost multi-path

Equal cost multi-path (ECMP) is a mechanism that allows a FortiGate to load-balance routed traffic over multiple gateways. Just like routes in a routing table, ECMP is considered after policy routing, so any matching policy routes will take precedence over ECMP.

ECMP pre-requisites are as follows:

- Routes must have the same destination and costs. In the case of static routes, costs include distance and priority
- Routes are sourced from the same routing protocol. Supported protocols include static routing, OSPF, and BGP

ECMP and SD-WAN implicit rule

ECMP and SD-WAN implicit rule are essentially similar in the sense that an SD-WAN implicit rule is processed after SD-WAN service rules are processed. See Implicit rule on page 1114 to learn more.

The following table summarizes the different load-balancing algorithms supported by each:

ECMP	SD-WAN		Description
	GUI	CLI	
`source-ip-based`	Source IP	`source-ip-based`	Traffic is divided equally between the interfaces. Sessions that start at the same source IP address use the same path. This is the default selection.
`weight-based`	Sessions	`weight-based`	The workload is distributed based on the number of sessions that are connected through the interface. The weight that you assign to each interface is used to calculate the percentage of the total sessions allowed to connect through an interface, and the sessions are distributed to the interfaces accordingly.

ECMP	SD-WAN		Description
	GUI	CLI	
usage-based	Spillover	usage-based	The interface is used until the traffic bandwidth exceeds the ingress and egress thresholds that you set for that interface. Additional traffic is then sent through the next interface member.
source-dest-ip-based	Source-Destination IP	source-dest-ip-based	Traffic is divided equally between the interfaces. Sessions that start at the same source IP address and go to the same destination IP address use the same path.
Not supported	Volume	measured-volume-based	This mode is supported in SD-WAN only. The workload is distributed based on the number of packets that are going through the interface.

To configure the ECMP algorithm from the CLI:

- At the VDOM level:

```
config system settings
    set v4-ecmp-mode {source-ip-based* | weight-based | usage-based | source-dest-ip-based}
end
```

- If SD-WAN is enabled, the above option is not available and ECMP is configured under the SD-WAN settings:

```
config system sdwan
    set status enable
    set load-balance-mode {source-ip-based* | weight-based | usage-based | source-dest-ip-
based | measured-volume-based}
end
```

For ECMP in IPv6, the mode must also be configured under SD-WAN:

```
# diagnose sys vd list
system fib version=63
list virtual firewall info:
name=root/root index=0 enabled fib_ver=40 use=168 rt_num=46 asym_rt=0 sip_helper=0, sip_nat_
trace=1, mc_fwd=0, mc_ttl_nc=0, tpmc_sk_pl=0
ecmp=source-ip-based, ecmp6=source-ip-based asym_rt6=0 rt6_num=55 strict_src_check=0 dns_log=1
ses_num=20 ses6_num=0 pkt_num=19154477
```

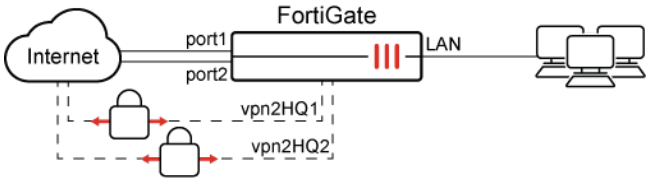
To change the number of paths allowed by ECMP:

```
config system settings
    set ecmp-max-paths <number of paths>
```

```
end
```

Setting `ecmp-max-paths` to the lowest value of `1` is equivalent to disabling ECMP.

ECMP configuration examples



The following examples demonstrate the behavior of ECMP in different scenarios:

- Example 1: Default ECMP on page 597
- Example 2: Same distance, different priority on page 597
- Example 3: Weight-based ECMP on page 598
- Example 4: Load-balancing BGP routes on page 599

Example 1: Default ECMP

```
config router static
    edit 1
        set gateway 172.16.151.1
        set device "port1"
    next
    edit 2
        set gateway 192.168.2.1
        set device "port2"
    next
end
```

```
# get router info routing-table all
Routing table for VRF=0
S*    0.0.0.0/0 [10/0] via 172.16.151.1, port1
                [10/0] via 192.168.2.1, port2
C    172.16.151.0/24 is directly connected, port1
C    192.168.2.0/24 is directly connected, port2
```

Result:

Both routes are added to the routing table and load-balanced based on the source IP.

Example 2: Same distance, different priority

```
config router static
    edit 1
```

```
        set gateway 172.16.151.1
        set priority 5
        set device "port1"
    next
    edit 2
        set gateway 192.168.2.1
        set device "port2"
    next
end
```

```
# get router info routing-table all
Routing table for VRF=0
S*    0.0.0.0/0 [10/0] via 192.168.2.1, port2
                [10/0] via 172.16.151.1, port1, [5/0]
C    172.16.151.0/24 is directly connected, port1
C    192.168.2.0/24 is directly connected, port2
```

Result:

Both routes are added to the routing table, but traffic is routed to `port2` which has a lower priority value with a default of `0`.

Example 3: Weight-based ECMP

```
config router static
    edit 3
        set dst 10.10.30.0 255.255.255.0
        set weight 80
        set device "vpn2HQ1"
    next
    edit 5
        set dst 10.10.30.0 255.255.255.0
        set weight 20
        set device "vpn2HQ2"
    next
end
```

```
# get router info routing-table all
Routing table for VRF=0
...
S    10.10.30.0/24 [10/0] is directly connected, vpn2HQ1, [0/80]
                   [10/0] is directly connected, vpn2HQ2, [0/20]
C    172.16.151.0/24 is directly connected, port1
C    192.168.0.0/24 is directly connected, port3
C    192.168.2.0/24 is directly connected, port2
```

Result:

Both routes are added to the routing table, but 80% of the sessions to `10.10.30.0/24` are routed to `vpn2HQ1`, and 20% are routed to `vpn2HQ2`.

Example 4: Load-balancing BGP routes

eBGP routes that are advertised to iBGP neighbors by multiple paths may be aggregated if the tie breakers specified in RFC 4271 > 9.1.2.2 have the same attribute. This includes having equal cost IGP metric. RFC conditions f) and g) are not evaluated when forming ECMP this way.

```
config router bgp
    set as 64511
    set router-id 192.168.2.86
    set ebgp-multipath enable
    config neighbor
        edit "192.168.2.84"
            set remote-as 64512
        next
        edit "192.168.2.87"
            set remote-as 64512
        next
   end
end
# get router info routing-table all
Routing table for VRF=0
...
C    172.16.151.0/24 is directly connected, port1
C    192.168.0.0/24 is directly connected, port3
C    192.168.2.0/24 is directly connected, port2
B    192.168.80.0/24 [20/0] via 192.168.2.84, port2, 00:00:33
                     [20/0] via 192.168.2.87, port2, 00:00:33
```

Result:

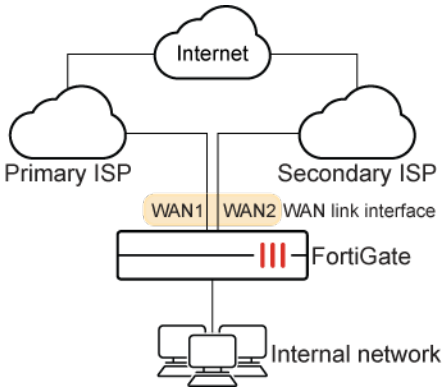
The network `192.168.80.0/24` is advertised by two BGP neighbors. Both routes are added to the routing table, and traffic is load-balanced based on Source IP.

For multiple BGP paths to be added to the routing table, you must enable `ebgp-multipath` for eBGP or `ibgp-multipath` for iBGP. These settings are disabled by default.

Dual internet connections

Dual internet connections, also referred to as dual WAN or redundant internet connections, refers to using two FortiGate interfaces to connect to the Internet. This is generally accomplished with SD-WAN, but this legacy solution provides the means to configure dual WAN without using SD-WAN. You can use dual internet connections in several ways:

- Link redundancy: If one interface goes down, the second interface automatically becomes the main connection.
- Load sharing: This ensures better throughput.
- Use a combination of link redundancy and load sharing.



This section describes the following dual internet connection scenarios:

- Scenario 1: Link redundancy and no load-sharing on page 600
- Scenario 2: Load-sharing and no link redundancy on page 602
- Scenario 3: Link redundancy and load-sharing on page 604

Scenario 1: Link redundancy and no load-sharing

Link redundancy ensures that if your Internet access is no longer available through a certain port, the FortiGate uses an alternate port to connect to the Internet.

In this scenario, two interfaces, WAN1 and WAN2, are connected to the Internet using two different ISPs. WAN1 is the primary connection. In the event of a failure of WAN1, WAN2 automatically becomes the connection to the Internet. For this configuration to function correctly, you must configure the following settings:

- Link health monitor on page 600: To determine when the primary interface (WAN1) is down and when the connection returns.
- Routing on page 601: Configure a default route for each interface.
- Security policies on page 602: Configure security policies to allow traffic through each interface to the internal network.

Link health monitor

Adding a link health monitor is required for routing failover traffic. A link health monitor confirms the device interface connectivity by probing a gateway or server at regular intervals to ensure it is online and working. When the server is not accessible, that interface is marked as down.

Set the `interval` (how often to send a ping) and `failtime` (how many lost pings are considered a failure). A smaller interval value and smaller number of lost pings results in faster detection, but creates more traffic on your network.

The link health monitor supports both IPv4 and IPv6, and various other protocols including ping, tcp-echo, udp-echo, http, and twamp.

To add a link health monitor (IPv4) using the CLI:

```
config system link-monitor
    edit <link-monitor-name>
```

```
        set addr-mode ipv4
        set srcintf <interface-name>
        set server <server-IP-address>
        set protocol {ping tcp-echo udp-echo http twamp}
        set gateway-ip <gateway-IP-address>
        set interval <seconds>
        set failtime <retry-attempts>
        set recoverytime <number-of-successful-responses>
        set status enable
    next
end
```

Command	Description
`update-cascade-interface {enable | disable}`	This option is used in conjunction with fail-detect and fail-alert options in interface settings to cascade the link failure down to another interface. See the Bring other interfaces down when link monitor fails KB article for details.
`update-static-route {enable | disable}`	When the link fails, all static routes associated with the interface will be removed.

Routing

You must configure a default route for each interface and indicate your preferred route as follows:

- Specify different distances for the two routes. The lower of the two distance values is declared active and placed in the routing table.

  Or

- Specify the same distance for the two routes, but give a higher priority to the route you prefer by defining a lower value. Both routes will be added to the routing table, but the route with a higher priority will be chosen as the best route

In the following example, we will use the first method to configure different distances for the two routes. You might not be able to connect to the backup WAN interface because the FortiGate does not route traffic out of the backup interface. The FortiGate performs a reverse path look-up to prevent spoofed traffic. If an entry cannot be found in the routing table that sends the return traffic out through the same interface, the incoming traffic is dropped.

To configure the routing of the two interfaces using the GUI:

1. Go to *Network* > *Static Routes*, and click *Create New*.
2. Enter the following information:

Destination	For an IPv4 route, enter a subnet of `0.0.0.0/0.0.0.0`. For an IPv6 route, enter a subnet of `::/0`.
Interface	Select the primary connection. For example, `wan1`.

Gateway Address	Enter the gateway address.
Administrative Distance	Leave as the default of `10`.

3. Click *OK*.
4. Repeat the above steps to set *Interface* to `wan2` and *Administrative Distance* to `20`.

To configure the routing of the two interfaces using the CLI:

```
config router {static | static6}
    edit 1
        set dst 0.0.0.0 0.0.0.0
        set device wan1
        set gateway <gateway_address>
        set distance 10
    next
    edit 2
        set dst 0.0.0.0 0.0.0.0
        set device wan2
        set gateway <gateway_address>
        set distance 20
    next
end
```

Security policies

When you create security policies, you need to configure duplicate policies to ensure that after traffic fails over WAN1, regular traffic is allowed to pass through WAN2, as it did with WAN1. This ensures that failover occurs with minimal effect to users.

Scenario 2: Load-sharing and no link redundancy

Load sharing may be accomplished in a few of the following ways of the many possible ways:

- By defining a preferred route with a lower distance, and specifying policy routes to route certain traffic to the secondary interface.
- By defining routes with same distance values but different priorities, and specifying policy routes to route certain traffic to the secondary interface.
- By defining routes with same distance values and priorities, and use equal-cost multi-path (ECMP) routing to equally distribute traffic between the WAN interfaces.

In our example, we will use the first option for our configuration. In this scenario, because link redundancy is not required, you do not have to configure a link monitor.

Traffic behaviour without a link monitor is as follows:

- If the remote gateway is down but the primary WAN interface of a FortiGate is still up, the FortiGate will continue to route traffic to the primary WAN. This results in traffic interruptions.
- If the primary WAN interface of a FortiGate is down due to physical link issues, the FortiGate will remove routes to it and the secondary WAN routes will become active. Traffic will failover to the secondary WAN.

Routing

Configure routing as you did in Scenario 1: Link redundancy and no load-sharing on page 600 above.

Policy routes

By configuring policy routes, you can redirect specific traffic to the secondary WAN interface. This works in this case because policy routes are checked before static routes. Therefore, even though the static route for the secondary WAN is not in the routing table, traffic can still be routed using the policy route.

In this example, we will create a policy route to route traffic from one address group to the secondary WAN interface.

To configure a policy route from the GUI:

1. Go to *Network* > *Policy Routes*, and click *Create New*.
2. Enter the following information:

Incoming interface	Define the source of the traffic. For example, `internal`.
Source Address	If we prefer to route traffic only from a group of addresses, define an address or address group, and add here.
Destination Address	Because we want to route all traffic from the address group here, we do not specify a destination address.
Protocol	Specify any protocol.
Action	Forward traffic.
Outgoing interface	Select the secondary WAN as the outbound interface. For example, `wan2`.
Gateway address	Input the gateway address for your secondary WAN. Because its default route has a higher distance value and is not added to the routing table, the gateway address must be added here.

3. Click OK.

To configure a policy route from the CLI:

```
config router policy
    edit 1
        set input-device "internal"
        set srcaddr "Laptops"
```

```
        set gateway <gateway_address>
        set output-device "wan2"
    next
end
```

Security policies

Your security policies should allow all traffic from `internal` to WAN1. Because link redundancy is not needed, you do not need to duplicate all WAN1 policies to WAN2. You will only need to define policies used in your policy route.

Scenario 3: Link redundancy and load-sharing

In this scenario, both the links are available to distribute Internet traffic with the primary WAN being preferred more. Should one of the interfaces fail, the FortiGate will continue to send traffic over the other active interface. The configuration is a combination of both the link redundancy and the load-sharing scenarios. The main difference is that the configured routes have equal distance values, with the route with a higher priority being preferred more. This ensures both routes are active in the routing table, but the route with a higher priority will be the best route.

Link health monitor

Link monitor must be configured for both the primary and the secondary WAN interfaces. This ensures that if the primary or the secondary WAN fails, the corresponding route is removed from the routing table and traffic re-routed to the other WAN interface.

For configuration details, see sample configurations in Scenario 1: Link redundancy and no load-sharing on page 600.

Routing

Both WAN interfaces must have default routes with the same distance. However, preference is given to the primary WAN by giving it a higher priority.

To configure the routing of the two interfaces using the CLI:

```
config router {static | static6}
    edit 1
        set dst 0.0.0.0 0.0.0.0
        set device wan1
        set gateway <gateway_address>
        set distance 10
        set priority 1
    next
    edit 2
        set dst 0.0.0.0 0.0.0.0
        set device wan2
        set gateway <gateway_address>
        set distance 10
```

```
        set priority 10
    next
end
```

Policy routes

The policy routes configuration is very similar to that of the policy routes in Scenario 2: Load-sharing and no link redundancy on page 602, except that the gateway address should not be specified. When a policy route is matched and the gateway address is not specified, the FortiGate looks at the routing table to obtain the gateway. In case the secondary WAN fails, traffic may hit the policy route. Because there is no gateway specified and the route to the secondary WAN is removed by the link monitor, the policy route will by bypassed and traffic will continue through the primary WAN. This ensures that the policy route is not active when the link is down.

Security policies

When you create security policies, you need to configure duplicate policies to ensure that after traffic fails over WAN1, regular traffic is allowed to pass through WAN2, as it was with WAN1. This ensures that failover occurs with minimal effect to users.

Dynamic routing

Dynamic routing protocols attempt to build a map of the network topology to identify the best routes to reach different destinations. Instead of manually defining static routes, which is not scalable, dynamic routing typically involves defining neighbors and peer routers that share their network topology and routing updates with each other. Protocols like distance vector, link state, and path vector are used by popular routing protocols. FortiGate supports RIP, OSPF, BGP, and IS-IS, which are interoperable with other vendors. When different dynamic routing protocols are used, the administrative distance of each protocol helps the FortiGate decide which route to pick.

Go to *System > Feature Visibility* and enable *Advanced Routing* to configure dynamic routing options in the GUI. See Feature visibility on page 3952 for more information.

Starting in 7.6.5, FortiGate models with 2 GB or 4 GB of RAM have optimized memory, and routing daemons are started only when routing configurations are detected. Routing daemons can be identified by following names:

- nsm
- bgpd
- pdmd
- pimd
- ospfd
- ospf6d
- isisd
- ripd
- ripngd

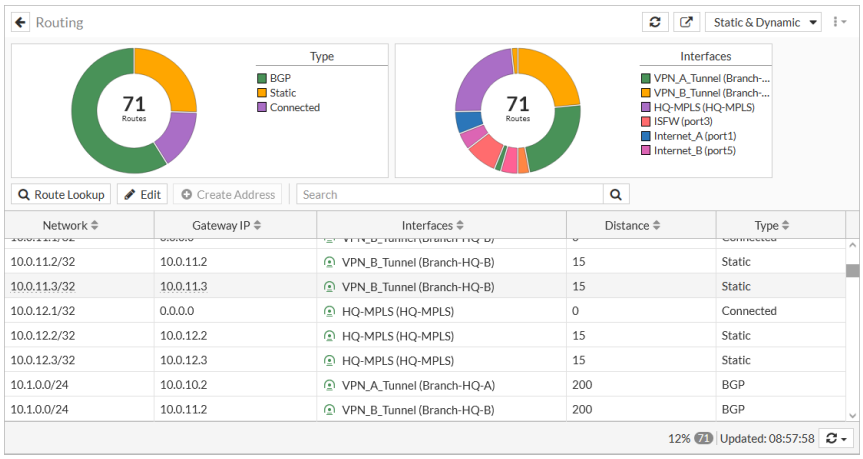
- pim6d

When the routing daemons are running, you can list the processes with commands such as `diagnose sys top` and `diagnose sys top-mem`.

This section includes:

- RIP on page 607
- OSPF on page 628
- BGP on page 646
- BFD on page 720
- Routing objects on page 731

To view the routing table and perform route look-ups in the GUI, go to *Dashboard* > *Network Monitor* > *Routing*.



To view the routing table in the CLI:

```
# get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*   0.0.0.0/0 [5/0] via 192.168.0.1, wan1
C    10.10.10.0/24 is directly connected, internal
C    169.254.2.1/32 is directly connected, Dialup-test
C    172.31.0.0/30 is directly connected, toKVM-MPLS
C    172.31.0.1/32 is directly connected, toKVM-MPLS
C    192.168.0.0/24 is directly connected, wan1
O    192.168.2.0/24 [110/101] via 10.10.10.11, internal, 00:00:26
S    192.168.20.0/24 [10/0] via 172.31.0.2, toKVM-MPLS
      [10/0] via 10.10.10.11, internal
```

RIP

Routing Information Protocol (RIP) is a distance-vector routing protocol that is intended for small and relatively homogeneous networks. It works well when there are minimal redundant paths and limited hop counts. FortiGate supports RIP version 1 (RFC 1058), RIP version 2 (RFC 2453), and RIPng (RFC 2080).

Basic configuration

To configure the FortiGate to participate in RIP using the most basic configurations in the GUI:

1. Go to *Network* > *RIP*.
2. Set the *Version*.
3. Add the networks that the FortiGate will advertise in and that will participate in RIP.
4. If the interface settings, such as passive interface, authentication, or enabling send/receive updates, must be edited, add the interfaces to the *Interface* table.
5. Click *Apply*.

To configure the FortiGate to participate in RIP using the most basic configurations in the CLI:

```
config router rip
    config network
        edit 1
            set prefix <subnet> <netmask>
        next
    end
    config interface
        edit <interface>
            set receive-version 2
            set send-version 2
        next
    end
end
```

Default route injection

Enabling *Inject default route* (`default-information-originate`) advertises a default route into the FortiGate's RIP network.

To enable/disable default route injection in the GUI:

1. Go to *Network* > *RIP*.
2. Expand the *Advanced Options*.
3. Enable/disable *Inject Default Route*.
4. Click *OK*.

To enable/disable default route injection in the CLI:

```
config router rip
    set default-information-originate {enable | disable}
end
```

Default metric

The default metric setting sets the default metric for all redistributed routes. If the default metric is set to five, and static routes are redistributed, then static routes have a metric of five. This value can be overridden by setting a specific metric value for a protocol. For example, the static route metric can be set to two, overriding the default metric.

```
config router rip
    set default-metric 5
    config redistribute "static"
        set status enable
        set metric 2
    end
end
```

The default metric is five, but redistributed static routes have a metric of two. So, the default metric is overridden and the metric for redistributed static routes is two.

Timers

RIP uses the update, timeout, and garbage timers to regulate its performance. The default timer settings are effective in most configurations. When customizing the settings, you must ensure that the new settings are compatible with your local routers and access servers.

Go to *Network* > *RIP* and expand the *Advanced Options* to configure the timers in the GUI, or use the CLI:

```
config router rip
    set timeout-timer <seconds>
    set update-timer <seconds>
    set garbage-timer <seconds>
end
```

Update timer

The update timer sets the interval between routing updates. The default value is 30 seconds. Randomness is added to help prevent network congestion due to multiple routers trying to update their neighbors simultaneously. The update timer must be at least three times shorter than the timeout timer.

If there is significant RIP traffic on the network, you can increase the update timer to send fewer updates. You must apply the same increase to all routers on the network to avoid timeouts that degrade your network speed.

Timeout timer

The timeout timer is the maximum amount of time that a reachable route is kept in the routing table since its last update. The default value is 180 seconds. If an update for the route is received before the timeout period elapses, then the timer is reset. The timeout timer should be at least three times longer than the update timer.

If routers are not responding to updates in time, increasing the timeout timer can help. A longer timeout timer results in longer update periods, and the FortiGate could wait a considerable amount of time for all of the timers to expire on an unresponsive route.

Garbage timer

The garbage timer is the amount of time that the FortiGate advertises a route as unreachable before deleting the route from the routing table. The default value is 120 seconds.

If the timer is short, older routes are removed from the routing table more quickly, resulting in a smaller routing table. This can be useful for large networks, or if the network changes frequently.

Authentication and key chain

RIP version 1 (RIPv1) has no authentication. RIP version 2 (RIPv2) uses text passwords or authentication keys to ensure that the routing information exchanged between routers is reliable. For authentication to work, both the sending and receiving routers must be set to use authentication and must be configured with the same password or keys. An authentication key that uses authentication key chains is more secure than a text password because the intervals when the key is valid can be configured.

A key chain is a list of one or more authentication keys that each have send and receive lifetimes. Keys are used to authenticate routing packets only during the keys specified lifetimes. The FortiGate migrates from one key to the next according to the scheduled lifetimes. The sending and receiving routers should have synchronized system dates and times to ensure that both ends are using the same keys at the same times. You can overlap the key lifetimes to make sure that a key is always available, even if there is some difference in the system times.

To configure a text password in the GUI:

1. Go to *Network* > *RIP*.
2. In the *Interfaces* table, click *Create New*, or edit an existing interface.
3. Enable *Authentication* and select *Text* or *MD5*.
4. Click *Change*, and enter the password.
5. Configure the remaining settings as needed.
6. Click *OK*.
7. Click *Apply*.

To configure a text password in the CLI:

```
config router rip
    config interface
        edit <interface>
            set auth-mode {text | md5}
```

```
            set auth-string **********
        next
    end
end
```

To configure a key chain with two sequentially valid keys and use it in a RIP interface:

```
config router key-chain
    edit rip_key
        config key
            edit 1
                set accept-lifetime 09:00:00 23 02 2020 09:00:00 17 03 2020
                set send-lifetime 09:00:00 23 02 2020 09:00:00 17 03 2020
                set key-string **********
            next
            edit 2
                set accept-lifetime 09:01:00 17 03 2020 09:00:00 1 04 2020
                set send-lifetime 09:01:00 17 03 2020 09:00:00 1 04 2020
                set key-string **********
            next
        end
    next
end
```

```
config router rip
    config interface
        edit port1
            set auth-keychain "rip_key"
        next
    end
end
```

Passive RIP interfaces

By default, an active RIP interface keeps the FortiGate routing table current by periodically asking neighbors for routes and sending out route updates. This can generate a significant amount of extra traffic in a large network.

A passive RIP interface listens to updates from other routers, but does not send out route updates. This can reduce network traffic when there are redundant routers in the network that would always send out essentially the same updates.

This example shows how to configure a passive RIPv2 interface on port1 using MD5 authentication.

To configure a passive RIP interface in the GUI:

1. Go to *Network* > *RIP*.
2. In the *Interfaces* table, click *Create New*.
3. Set *Interface* to the required interface.
4. Enable *Passive*.
5. Enable *Authentication* and set it to *MD5*.

6. Click *Change* and enter a password.
7. Set *Receive Version* to *2*.
8. Click *OK*.

To configure a passive RIP interface in the CLI:

```
config router rip
    set passive-interface "port1"
    config interface
        edit "port1"
            set auth-mode md5
            set auth-string **********
            set receive-version 2
            set send-version 2
        next
    end
end
```

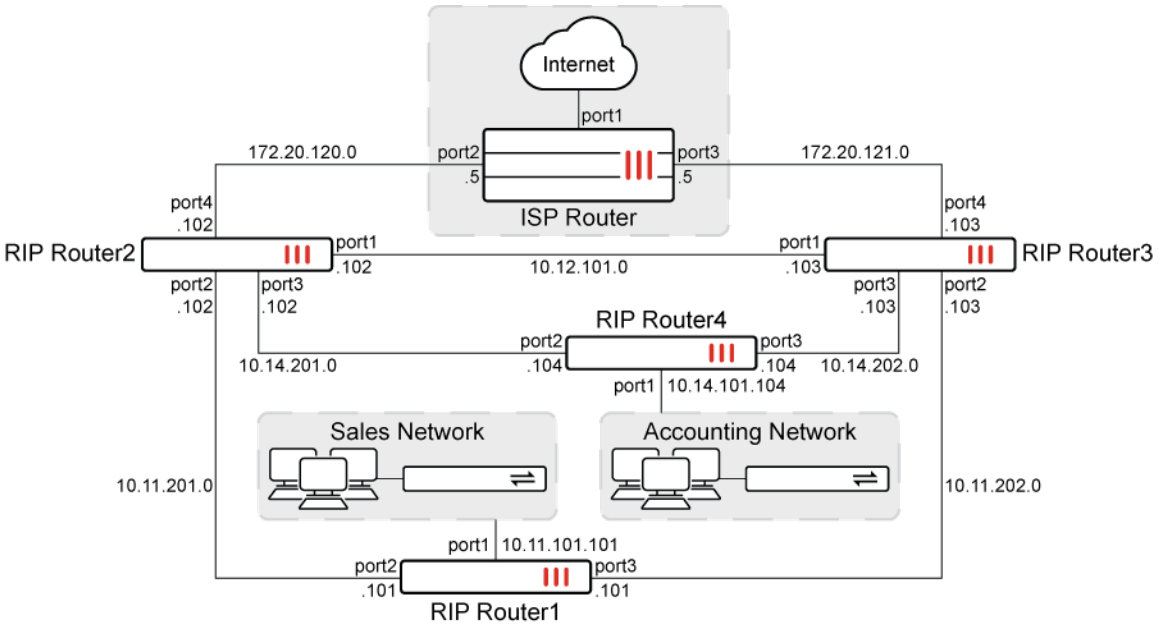
RIP and IPv6

RIP next generation (RIPng) is an extension of RIPv2 that includes support for IPv6. See Basic RIPng example on page 625 and IPv6 tunneling on page 923 for more information.

Basic RIP example

In this example, a medium-sized network is configured using RIPv2.

- Two core routers, RIP Router2 and RIP Router3, connect to the ISP router for two redundant paths to the internet.
- Two other routers, RIP Router1 and RIP Router4, connect to the two core routers and to different local networks.
- The ISP router is using RIP for its connections to the core routers, and redistributes its default route to the network - that is, default route injection is enabled.
- The ISP router uses NAT and has a static route to the internet. None of the other routers use NAT or static routes.



All of the FortiGate routers are configured as shown, using netmask 255.255.255.0. Firewall policies have been configured to allow the required traffic to flow across the interfaces.

Router	Interface	Interface name	IP address
Router1	port1	LoSales	10.11.101.101
	port2	vd12link0	10.11.201.101
	port3	vd13link0	10.11.202.101
Router2	port1	vd23link0	10.12.101.102
	port2	vd12link1	10.11.201.102
	port3	vd42link1	10.14.201.102
	port4	vdr2link1	172.20.120.102
Router3	port1	vd23link1	10.12.101.103
	port2	vd13link1	10.11.202.103
	port3	vd43link1	10.14.202.103
	port4	vdr3link1	172.20.121.103
Router4	port1	LoAccounting	10.14.101.104
	port2	vd42link0	10.14.201.104
	port3	vd43link0	10.14.202.104
ISP Router	port1	port1	To internet
	port2	vdr2link0	172.20.120.5
	port3	vdr3link0	172.20.121.5

After configuring each router, you can check the status of the connections by viewing the RIP database, RIP interfaces, and routing table. See Verifying the configuration on page 618.

After the network is configured, you can test it to ensure that when network events occur, such as a downed link, routing updates are triggered and converge as expected. See Testing the configuration and routing changes on page 622.

ISP router

To configure the ISP Router in the GUI:

1. Go to *Network* > *RIP*.
2. Set the *Version* to *2*.
3. Under *Networks*, add two networks:
   - *172.20.120.0/255.255.255.0*
   - *172.20.121.0/255.255.255.0*
4. Add the interfaces:
   a. In the *Interfaces* table, click *Create New*.
   b. Set *Interface* to *port2*.
   c. Leave the remaining settings as their default values.
   d. Click *OK*.
   e. Repeat these steps for *port3*.
5. Under *Advanced Options*, enable *Inject Default Route*.
   This setting allows the ISP router to share its default 0.0.0.0 routes with other routers in the RIP network.
6. Click *Apply*.

To configure the ISP Router in the CLI:

```
config router rip
    set default-information-originate enable
    config network
        edit 1
            set prefix 172.20.121.0 255.255.255.0
        next
        edit 2
            set prefix 172.20.120.0 255.255.255.0
        next
    end
    config interface
        edit "port2"
            set receive-version 2
            set send-version 2
        next
        edit "port3"
            set receive-version 2
            set send-version 2
        next
```

```
    end
end
```

Router2 and Router3

Router2 and Router3 RIP configurations have different IP addresses, but are otherwise the same.

To configure Router2 and Router3 in the GUI:

1. Go to *Network* > *RIP*.
2. Set the *Version* to *2*.
3. Under *Networks*, add the IP addresses for each port:

Router2	10.12.101.0/255.255.255.0
	10.11.201.0/255.255.255.0
	10.14.201.0/255.255.255.0
	172.20.120.0/255.255.255.0
Router3	10.12.101.0/255.255.255.0
	10.11.202.0/255.255.255.0
	10.14.202.0/255.255.255.0
	172.20.121.0/255.255.255.0

4. Add the interfaces:
   a. In the *Interfaces* table, click *Create New*.
   b. Set *Interface* to *port1*.
   c. Leave the remaining settings as their default values.
   d. Click *OK*.
   e. Repeat these steps for *port2*, *port3*, and *port4*.
5. Click *Apply*.

To configure Router2 in the CLI:

```
config router rip
    config network
        edit 1
            set prefix 10.12.101.0 255.255.255.0
        next
        edit 2
            set prefix 10.11.201.0 255.255.255.0
        next
        edit 3
            set prefix 10.14.201.0 255.255.255.0
        next
        edit 4
```

```
            set prefix 172.20.120.0 255.255.255.0
        next
    end
    config interface
        edit "port1"
            set receive-version 2
            set send-version 2
        next
        edit "port2"
            set receive-version 2
            set send-version 2
        next
        edit "port3"
            set receive-version 2
            set send-version 2
        next
        edit "port4"
            set receive-version 2
            set send-version 2
        next
    end
end
```

To configure Router3 in the CLI:

```
config router rip
    config network
        edit 1
            set prefix 10.12.101.0 255.255.255.0
        next
        edit 2
            set prefix 10.11.202.0 255.255.255.0
        next
        edit 3
            set prefix 10.14.202.0 255.255.255.0
        next
        edit 4
            set prefix 172.20.121.0 255.255.255.0
        next
    end
    config interface
        edit "port1"
            set receive-version 2
            set send-version 2
        next
        edit "port2"
            set receive-version 2
            set send-version 2
        next
        edit "port3"
            set receive-version 2
            set send-version 2
```

```
        next
        edit "port4"
            set receive-version 2
            set send-version 2
        next
    end
end
```

Router1 and Router4

Router1 and Router4 RIP configurations have different IP addresses, but are otherwise the same.

To configure Router1 and Router4 in the GUI:

1. Go to *Network* > *RIP*.
2. Set the *Version* to *2*.
3. Under *Networks*, add the IP addresses for each port:

Router1	10.11.101.0/255.255.255.0
	10.11.201.0/255.255.255.0
	10.11.202.0/255.255.255.0
Router4	10.14.101.0/255.255.255.0
	10.14.201.0/255.255.255.0
	10.14.202.0/255.255.255.0

4. Add the interfaces:
   a. In the *Interfaces* table, click *Create New*.
   b. Set *Interface* to *port1*.
   c. For port1 only, enable *Passive*.
   d. Leave the remaining settings as their default values.
   e. Click *OK*.
   f. Repeat these steps for *port2* and *port3*, making sure that *Passive* is disabled.
5. Click *Apply*.

To configure Router1 in the CLI:

```
config router rip
    config network
        edit 1
            set prefix 10.11.101.0 255.255.255.0
        next
        edit 2
            set prefix 10.11.201.0 255.255.255.0
        next
        edit 3
```

```
            set prefix 10.11.202.0 255.255.255.0
        next
    end
    set passive-interface "port1"
    config interface
        edit "port1"
            set receive-version 2
            set send-version 2
        next
        edit "port2"
            set receive-version 2
            set send-version 2
        next
        edit "port3"
            set receive-version 2
            set send-version 2
        next
    end
end
```

To configure Router4 in the CLI:

```
config router rip
    config network
        edit 1
            set prefix 10.14.101.0 255.255.255.0
        next
        edit 2
            set prefix 10.14.201.0 255.255.255.0
        next
        edit 3
            set prefix 10.14.202.0 255.255.255.0
        next
    end
    set passive-interface "port1"
    config interface
        edit "port1"
            set receive-version 2
            set send-version 2
        next
        edit "port2"
            set receive-version 2
            set send-version 2
        next
        edit "port3"
            set receive-version 2
            set send-version 2
        next
    end
end
```

Verifying the configuration

The interface's names are shown in the debugs. The same commands should also be run on the other routers.

To verify the configuration after the ISP router, Router2, and Route3 have been configured:

This verification can be done after the ISP router, Router2, and Router3 have been configured. Only Router2's debugs are shown.

1. Check the RIP interface information:

```
# get router info rip interface
Router2 is up, line protocol is up
  RIP is not enabled on this interface
ssl.Router2 is up, line protocol is up
  RIP is not enabled on this interface
vdr2link1 is up, line protocol is up
  Routing Protocol: RIP
    Receive RIPv2 packets only
    Send RIPv2 packets only
    Passive interface: Disabled
    Split horizon: Enabled with Poisoned Reversed
    IP interface address:
      172.20.120.102/24
vd12link1 is up, line protocol is up
  Routing Protocol: RIP
    Receive RIPv2 packets only
    Send RIPv2 packets only
    Passive interface: Disabled
    Split horizon: Enabled with Poisoned Reversed
    IP interface address:
      10.11.201.102/24
vd42link1 is up, line protocol is up
  Routing Protocol: RIP
    Receive RIPv2 packets only
    Send RIPv2 packets only
    Passive interface: Disabled
    Split horizon: Enabled with Poisoned Reversed
    IP interface address:
      10.14.201.102/24
vd23link0 is up, line protocol is up
  Routing Protocol: RIP
    Receive RIPv2 packets only
    Send RIPv2 packets only
    Passive interface: Disabled
    Split horizon: Enabled with Poisoned Reversed
    IP interface address:
      10.12.101.102/24
```

RIP starts exchanging routes as soon as the networks are added to the Router2 and Router3 configurations because the RIP interfaces are active by default, and start sending and receiving RIP updates when a matching interface on the subnet is found. The interface configuration allows the interface settings to be fine tuned, in this case to specify only RIPv2 support.

2. Check the RIP database:

```
# get router info rip database
Codes: R - RIP, Rc - RIP connected, Rs - RIP static, K - Kernel,
       C - Connected, S - Static, O - OSPF, I - IS-IS, B - BGP
   Network            Next Hop         Metric From            If     Time
R  0.0.0.0/0          172.20.120.5          2 172.20.120.5    vdr2link1 02:55
Rc 10.11.201.0/24                           1                 vd12link1
R  10.11.202.0/24     10.12.101.103         2 10.12.101.103   vd23link0 02:33
Rc 10.12.101.0/24                           1                 vd23link0
Rc 10.14.201.0/24                           1                 vd42link1
R  10.14.202.0/24     10.12.101.103         2 10.12.101.103   vd23link0 02:33
Rc 172.20.120.0/24                          1                 vdr2link1
R  172.20.121.0/24    10.12.101.103         2 10.12.101.103   vd23link0 02:33
```

3. Check the routing table:

```
# get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default
Routing table for VRF=0
R*      0.0.0.0/0 [120/2] via 172.20.120.5, vdr2link1, 13:37:23
C       10.11.201.0/24 is directly connected, vd12link1
R       10.11.202.0/24 [120/2] via 10.12.101.103, vd23link0, 14:10:01
C       10.12.101.0/24 is directly connected, vd23link0
C       10.14.201.0/24 is directly connected, vd42link1
R       10.14.202.0/24 [120/2] via 10.12.101.103, vd23link0, 14:10:01
C       172.20.120.0/24 is directly connected, vdr2link1
R       172.20.121.0/24 [120/2] via 10.12.101.103, vd23link0, 13:20:36
```

Router2 has learned the default gateway from the ISP router, and has learned of other networks from Router3.

4. If firewall policies are correctly configured, the outside network can be reached:

```
# execute ping-options source 10.11.201.102
# execute ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: icmp_seq=0 ttl=115 time=4.5 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=115 time=4.2 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=115 time=4.2 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=115 time=4.2 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=115 time=4.1 ms
--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 4.1/4.2/4.5 ms
```

```
# execute traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 32 hops max, 3 probe packets per hop, 84 byte packets
```

```
 1  172.20.120.5  0.101 ms  0.030 ms  0.014 ms
 2  172.16.151.1  0.169 ms  0.144 ms  0.131 ms
 3  * * *
```

To verify the configuration after Router1 and Router4 have also been configured:

This verification can be done after Router1 and Router4 have been configured. Only Router1's debugs are shown.

1. Check the RIP interface information:

```
# get router info rip interface
Router1 is up, line protocol is up
  RIP is not enabled on this interface
ssl.Router1 is up, line protocol is up
  RIP is not enabled on this interface
vd12link0 is up, line protocol is up
  Routing Protocol: RIP
    Receive RIPv2 packets only
    Send RIPv2 packets only
    Passive interface: Disabled
    Split horizon: Enabled with Poisoned Reversed
    IP interface address:
      10.11.201.101/24
vd13link0 is up, line protocol is up
  Routing Protocol: RIP
    Receive RIPv2 packets only
    Send RIPv2 packets only
    Passive interface: Disabled
    Split horizon: Enabled with Poisoned Reversed
    IP interface address:
      10.11.202.101/24
LoSales is up, line protocol is up
  Routing Protocol: RIP
    Receive RIPv2 packets only
    Send RIPv2 packets only
    Passive interface: Enabled
    Split horizon: Enabled with Poisoned Reversed
    IP interface address:
      10.11.101.101/24
      127.0.0.1/8
```

2. Check the RIP database:

```
# get router info rip database
Codes: R - RIP, Rc - RIP connected, Rs - RIP static, K - Kernel,
       C - Connected, S - Static, O - OSPF, I - IS-IS, B - BGP
   Network            Next Hop         Metric From            If     Time
R  0.0.0.0/0          10.11.202.103         3 10.11.202.103   vd13link0 02:35
Rc 10.11.101.0/24                           1                 LoSales
Rc 10.11.201.0/24                           1                 vd12link0
Rc 10.11.202.0/24                           1                 vd13link0
```

```
R  10.12.101.0/24     10.11.202.103         2 10.11.202.103   vd13link0 02:35
R  10.14.101.0/24     10.11.202.103         3 10.11.202.103   vd13link0 02:35
R  10.14.201.0/24     10.11.201.102         2 10.11.201.102   vd12link0 02:30
R  10.14.202.0/24     10.11.202.103         2 10.11.202.103   vd13link0 02:35
R  172.20.120.0/24    10.11.201.102         2 10.11.201.102   vd12link0 02:30
R  172.20.121.0/24    10.11.202.103         2 10.11.202.103   vd13link0 02:35
```

3. Check the routing table:

```
# get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default
Routing table for VRF=0
R*      0.0.0.0/0 [120/3] via 10.11.202.103, vd13link0, 00:09:42
C       10.11.101.0/24 is directly connected, LoSales
C       10.11.201.0/24 is directly connected, vd12link0
C       10.11.202.0/24 is directly connected, vd13link0
R       10.12.101.0/24 [120/2] via 10.11.202.103, vd13link0, 00:09:42
R       10.14.101.0/24 [120/3] via 10.11.202.103, vd13link0, 00:09:42
R       10.14.201.0/24 [120/2] via 10.11.201.102, vd12link0, 00:09:42
R       10.14.202.0/24 [120/2] via 10.11.202.103, vd13link0, 00:09:42
R       172.20.120.0/24 [120/2] via 10.11.201.102, vd12link0, 00:09:42
R       172.20.121.0/24 [120/2] via 10.11.202.103, vd13link0, 00:09:42
```

4. If firewall policies are correctly configured, the accounting network and the internet are reachable from the sales network:

```
# execute ping-options source 10.11.101.101
# execute ping 10.14.101.104
PING 10.14.101.104 (10.14.101.104): 56 data bytes
64 bytes from 10.14.101.104: icmp_seq=0 ttl=254 time=0.1 ms
64 bytes from 10.14.101.104: icmp_seq=1 ttl=254 time=0.0 ms
64 bytes from 10.14.101.104: icmp_seq=2 ttl=254 time=0.0 ms
64 bytes from 10.14.101.104: icmp_seq=3 ttl=254 time=0.0 ms
64 bytes from 10.14.101.104: icmp_seq=4 ttl=254 time=0.0 ms
--- 10.14.101.104 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.0/0.0/0.1 ms
```

```
# execute traceroute 10.14.101.104
traceroute to 10.14.101.104 (10.14.101.104), 32 hops max, 3 probe packets per hop, 84 byte
packets
 1  10.11.202.103  0.079 ms  0.029 ms  0.013 ms
 2  10.14.101.104  0.043 ms  0.020 ms  0.010 ms
```

```
# execute ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: icmp_seq=0 ttl=114 time=4.3 ms
```

```
64 bytes from 8.8.8.8: icmp_seq=1 ttl=114 time=4.1 ms
--- 8.8.8.8 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 4.1/4.2/4.3 ms
```

```
# execute traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 32 hops max, 3 probe packets per hop, 84 byte packets
 1  10.11.202.103  0.094 ms  0.036 ms  0.030 ms
 2  172.20.121.5  0.216 ms  0.045 ms  0.038 ms
```

Testing the configuration and routing changes

After the network is configured, test it to ensure that when network events occur, such as a downed link, routing updates are triggered and converge as expected.

In the following examples, we disable certain links to simulate network outages, then verify that routing and connectivity is restored after the updates have converged.

Example 1 - ISP router port3 interface goes down

In this example, a link outage occurs on port3 of the ISP router. Consequently, all routers must use Router2, and not Router3, to reach the internet. Note the RIP database before and after the link failure, and the time taken for the route updates to propagate and return to a functioning state.

Router4's debugs are shown.

Before:

```
# get router info rip database
Codes: R - RIP, Rc - RIP connected, Rs - RIP static, K - Kernel,
       C - Connected, S - Static, O - OSPF, I - IS-IS, B - BGP
   Network            Next Hop         Metric From            If     Time
R  0.0.0.0/0          10.14.202.103         3 10.14.202.103   vd43link0 02:31
R  10.11.101.0/24     10.14.202.103         3 10.14.202.103   vd43link0 02:31
R  10.11.201.0/24     10.14.201.102         2 10.14.201.102   vd42link0 02:47
R  10.11.202.0/24     10.14.202.103         2 10.14.202.103   vd43link0 02:31
R  10.12.101.0/24     10.14.202.103         2 10.14.202.103   vd43link0 02:31
Rc 10.14.101.0/24                           1                 LoAccounting
Rc 10.14.201.0/24                           1                 vd42link0
Rc 10.14.202.0/24                           1                 vd43link0
R  172.20.120.0/24    10.14.201.102         2 10.14.201.102   vd42link0 02:47
R  172.20.121.0/24    10.14.202.103         2 10.14.202.103   vd43link0 02:31
```

```
# get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default
Routing table for VRF=0
R*      0.0.0.0/0 [120/3] via 10.14.202.103, vd43link0, 02:45:15
```

```
R       10.11.101.0/24 [120/3] via 10.14.202.103, vd43link0, 02:44:49
R       10.11.201.0/24 [120/2] via 10.14.201.102, vd42link0, 02:45:15
R       10.11.202.0/24 [120/2] via 10.14.202.103, vd43link0, 02:45:15
R       10.12.101.0/24 [120/2] via 10.14.202.103, vd43link0, 02:45:15
C       10.14.101.0/24 is directly connected, LoAccounting
C       10.14.201.0/24 is directly connected, vd42link0
C       10.14.202.0/24 is directly connected, vd43link0
R       172.20.120.0/24 [120/2] via 10.14.201.102, vd42link0, 02:45:15
R       172.20.121.0/24 [120/2] via 10.14.202.103, vd43link0, 02:45:15
```

```
# execute traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 32 hops max, 3 probe packets per hop, 84 byte packets
 1  10.14.202.103  0.187 ms  0.054 ms  0.030 ms
 2  172.20.121.5  0.117 ms  0.062 ms  0.040 ms
 3  * * *
```

After:

- You might see different routes, and the routes might change, while convergence is occurring. During convergence, the metric for your default route increases to 16.

```
# get router info rip database
Codes: R - RIP, Rc - RIP connected, Rs - RIP static, K - Kernel,
       C - Connected, S - Static, O - OSPF, I - IS-IS, B - BGP
   Network            Next Hop         Metric From            If     Time
R  0.0.0.0/0          10.14.202.103        16 10.14.202.103   vd43link0 01:50
```

- After convergence is complete, the RIP database will look similar to the following:

```
# get router info rip database
Codes: R - RIP, Rc - RIP connected, Rs - RIP static, K - Kernel,
       C - Connected, S - Static, O - OSPF, I - IS-IS, B - BGP
   Network            Next Hop         Metric From            If     Time
R  0.0.0.0/0          10.14.201.102         3 10.14.201.102   vd42link0 02:53
R  10.11.101.0/24     10.14.202.103         3 10.14.202.103   vd43link0 03:00
R  10.11.201.0/24     10.14.201.102         2 10.14.201.102   vd42link0 02:53
R  10.11.202.0/24     10.14.202.103         2 10.14.202.103   vd43link0 03:00
R  10.12.101.0/24     10.14.202.103         2 10.14.202.103   vd43link0 03:00
Rc 10.14.101.0/24                           1                 LoAccounting
Rc 10.14.201.0/24                           1                 vd42link0
Rc 10.14.202.0/24                           1                 vd43link0
R  172.20.120.0/24    10.14.201.102         2 10.14.201.102   vd42link0 02:53
```

- The default router should point to Router2, with the same number of hops:

```
# get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default
Routing table for VRF=0
```

```
R*      0.0.0.0/0 [120/3] via 10.14.201.102, vd42link0, 00:05:24
R       10.11.101.0/24 [120/3] via 10.14.202.103, vd43link0, 02:58:13
R       10.11.201.0/24 [120/2] via 10.14.201.102, vd42link0, 02:58:39
R       10.11.202.0/24 [120/2] via 10.14.202.103, vd43link0, 02:58:39
R       10.12.101.0/24 [120/2] via 10.14.202.103, vd43link0, 02:58:39
C       10.14.101.0/24 is directly connected, LoAccounting
C       10.14.201.0/24 is directly connected, vd42link0
C       10.14.202.0/24 is directly connected, vd43link0
R       172.20.120.0/24 [120/2] via 10.14.201.102, vd42link0, 02:58:39
```

```
# execute traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 32 hops max, 3 probe packets per hop, 84 byte packets
 1  10.14.201.102  0.167 ms  0.063 ms  0.029 ms
 2  172.20.120.5  0.117 ms  0.073 ms  0.041 ms
 3  172.16.151.1  0.303 ms  0.273 ms  0.253 ms
```

Example 2- Additional link failures on Router2

In addition to the link failure on the ISP router in example, port1 and port3 on Router2 have also failed. This means that Router4 must go through Router3, Router1, Router2, then the ISP router to reach the internet. Note that, for a period of time, some routes' metrics increase to 16. If no better routes are found for these networks, then they eventually disappear.

After the convergence completes, the RIP database and routing table on Router4 should resemble the following:

```
# get router info rip database
Codes: R - RIP, Rc - RIP connected, Rs - RIP static, K - Kernel,
       C - Connected, S - Static, O - OSPF, I - IS-IS, B - BGP
   Network            Next Hop         Metric From            If     Time
R  0.0.0.0/0          10.14.202.103         5 10.14.202.103   vd43link0 02:54
R  10.11.101.0/24     10.14.202.103         3 10.14.202.103   vd43link0 02:54
R  10.11.201.0/24     10.14.202.103         3 10.14.202.103   vd43link0 02:54
R  10.11.202.0/24     10.14.202.103         2 10.14.202.103   vd43link0 02:54
Rc 10.14.101.0/24                           1                 LoAccounting
Rc 10.14.202.0/24                           1                 vd43link0
R  172.20.120.0/24    10.14.202.103         4 10.14.202.103   vd43link0 02:54
```

```
# get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default
Routing table for VRF=0
R*      0.0.0.0/0 [120/5] via 10.14.202.103, vd43link0, 00:03:54
R       10.11.101.0/24 [120/3] via 10.14.202.103, vd43link0, 03:10:12
R       10.11.201.0/24 [120/3] via 10.14.202.103, vd43link0, 00:03:54
R       10.11.202.0/24 [120/2] via 10.14.202.103, vd43link0, 03:10:38
C       10.14.101.0/24 is directly connected, LoAccounting
C       10.14.202.0/24 is directly connected, vd43link0
R       172.20.120.0/24 [120/4] via 10.14.202.103, vd43link0, 00:03:54
```

Reaching the internet on the default gateway now requires five hops from Router4:

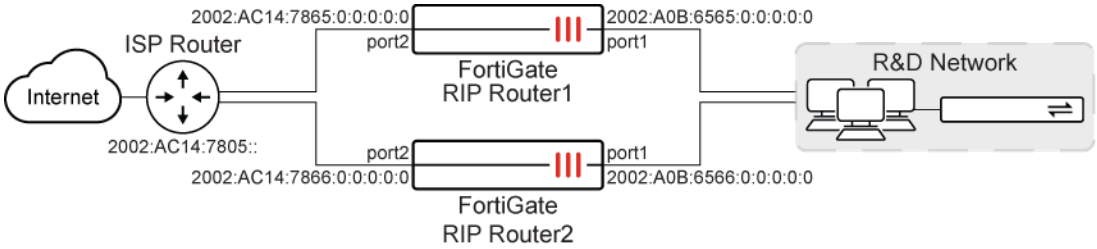
```
# execute traceroute 8.8.8.8
traceroute to 8.8.8.8 (8.8.8.8), 32 hops max, 3 probe packets per hop, 84 byte packets
 1  10.14.202.103  0.087 ms  0.026 ms  0.012 ms
 2  10.11.202.101  0.045 ms  0.024 ms  0.025 ms
 3  10.11.201.102  0.048 ms  0.024 ms  0.015 ms
 4  172.20.120.5  0.050 ms  0.028 ms  0.019 ms
 5  * * *
```

Basic RIPng example

In this example, a small network is configured with RIP next generation (RIPng). Two FortiGates are connected to the internal network and the ISP, providing some redundancy to help ensure that the internal network can always reach the internet.

The FortiGates are running in NAT mode with VDOMs disabled, and firewall policies have already been configured to allow traffic to flow across the interfaces.

All of the internal computers and other network devices support IPv6 addressing and are running RIPng (where applicable), so no static routing is required. Internal network devices only need to know the FortiGate's internal interface network addresses.



Router	Interface (alias)	IPv6 address
Router1	port1 (internal)	2002:A0B:6565:0:0:0:0:0
	port2 (ISP)	2002:AC14:7865:0:0:0:0:0
Router2	port1 (internal)	2002:A0B:6566:0:0:0:0:0
	port2 (ISP)	2002:AC14:7866:0:0:0:0:0

On each FortiGate, the interfaces are configured first, and then RIPng. No redistribution or authentication is configured.

In the RIPng configuration, only the interface names are required. The ISP router and the other FortiGate are configured as neighbors. Declaring the neighbors reduces the discovery traffic when the routers start. There is no specific command to include a subnet in the RIP broadcast, and RIPng can only be configured using the CLI.

To configure Router1:

1. Configure the interfaces:

```
config system interface
    edit port1
        set allowaccess ping https ssh
        set type physical
        set description "Internal RnD network"
        set alias "internal"
        config ipv6
            set ip6-address 2002:a0b:6565::/0
        end
    next
    edit port2
        set allowaccess ping https ssh
        set type physical
        set description "ISP and Internet"
        set alias "ISP"
        config ipv6
            set ip6-address 2002:ac14:7865::/0
        end
    next
end
```

2. Configure RIPng:

```
config router ripng
    config neighbor
        edit 1
            set ip6 2002:a0b:6566::
            set interface port1
        next
        edit 2
            set ip6 2002:ac14:7805::
            set interface port2
        next
    end
    config interface
        edit port1
        next
        edit port2
        next
    end
end
```

To configure Router2:

1. Configure the interfaces:

```
config system interface
    edit port1
```

```
        set allowaccess ping https ssh
        set type physical
        set description "Internal RnD network"
        set alias "internal"
        config ipv6
            set ip6-address 2002:a0b:6566::/0
        end
    next
    edit port2
        set allowaccess ping https ssh
        set type physical
        set description "ISP and Internet"
        set alias "ISP"
        config ipv6
            set ip6-address 2002:ac14:7866::/0
        end
    next
end
```

2. Configure RIPng:

```
config router ripng
    config neighbor
        edit 1
            set ip6 2002:a0b:6565::
            set interface port1
        next
        edit 2
            set ip6 2002:ac14:7805::
            set interface port2
        next
    end
    config interface
        edit port1
        next
        edit port2
        next
    end
end
```

Testing the configuration

The following commands can be used to check the RIPng information on the FortiGates, and can help track down issues:

To view the local scope IPv6 addresses used as next-hops by RIPng on the FortiGate:

```
# diagnose ipv6 address list
```

To view IPv6 addresses that are installed in the routing table:

```
# diagnose ipv6 route list
```

To view the IPv6 routing table:

```
# get router info6 routing-table
```

This information is similar to the `diagnose ipv6 route list` command, but it is presented in an easier to read format.

To view the brief output on the RIP information for the interface listed:

```
# get router info6 rip interface external
```

This includes information such as, if the interface is up or down, what routing protocol is being used, and whether passive interface or split horizon is enabled.

OSPF

Open Shortest Path First (OSPF) is a link state routing protocol that is commonly used in large enterprise networks with L3 switches, routers, and firewalls from multiple vendors. It can quickly detect link failures, and converge network traffic without networking loops. It also has features to control which routes are propagated, allowing for smaller routing tables, and provides better load balancing on external links when compared to other routing protocols.

To configure OSPF in the GUI, go to *Network* > *OSPF*:

Option	Description
Router ID	A unique ID to identify your router in the network, typically in the format x.x.x.x.
Areas	The areas that the router is part of. For each are area, define the *Area ID*, *Type*, and *Authentication* method.
Networks	The networks that OSPF is enabled in, and the area that they belong to.
Interfaces	OSPF interfaces for transmitting and receiving packets. Configure interface properties, such as *Network Type*, *Cost*, *Hello interval*, and others.
Advanced Options	Settings for *Inject Default Route*, *Passive Interfaces*, and *Redistribute*. Redistribution can be enabled by protocol and the metric for each protocol can be configured.

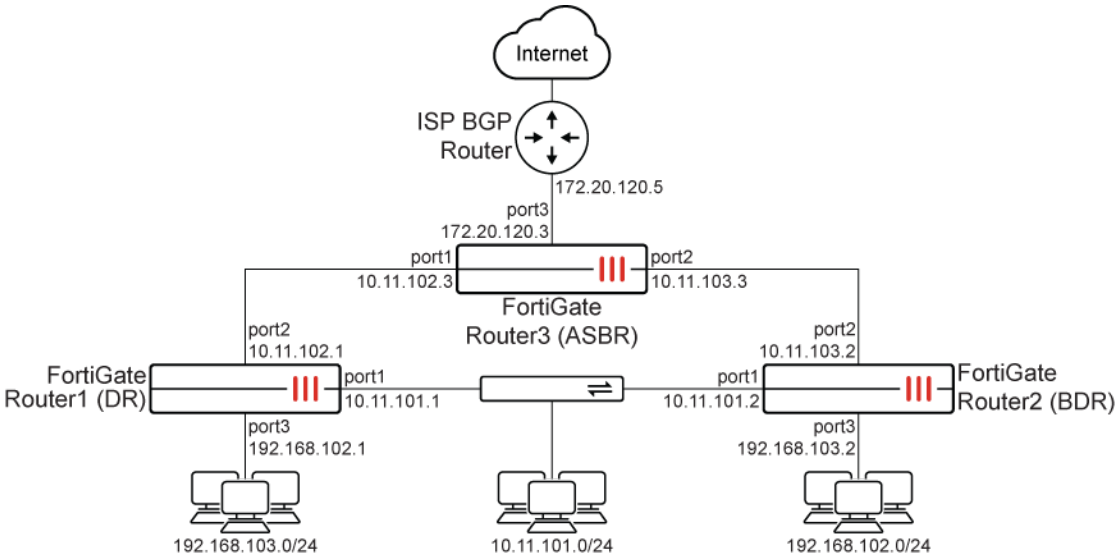
This section includes the following topics:

- Basic OSPF example on page 629
- OSPFv3 neighbor authentication on page 640
- OSPF graceful restart upon a topology change on page 642
- OSPF link detection customization on page 646

Basic OSPF example

In this example, three FortiGate devices are configured in an OSPF network.

- Router1 is the Designated Router (DR). It has the highest priority and the lowest IP address, to ensure that it becomes the DR.
- Router2 is the Backup Designated Router (BDR). It has a high priority to ensure that it becomes the BDR.
- Router3 is the Autonomous System Border Router (ASBR). It routes all traffic to the ISP BGP router for internet access. It redistributes routes from BGP and advertises a default route to its neighbors. It can allow different types of routes, learned outside of OSPF, to be used in OSPF. Different metrics can be assigned to these routes to make them more or less preferred than regular OSPF routes. Route maps could be used to further control what prefixes are advertised or received from the ISP.



FortiGate	Interface	IP address
Router1 (DR)	port1	10.11.101.1
	port2	10.11.102.1
	port3	192.168.102.1
Router2 (BDR)	port1	10.11.101.2
	port2	10.11.103.2
	port3	192.168.103.2
Router3 (ASBR)	port1	10.11.102.3
	port2	10.11.103.3
	port3	172.20.120.3

- Firewall policies are already configured to allow unfiltered traffic in both directions between all of the connected interfaces.

- The interfaces are already configured, and NAT is only used for connections to public networks. The costs for all of the interfaces is left at 0.
- The OSPF network belongs to Area 0, and is not connected to any other OSPF networks. All of the routers are part of the backbone 0.0.0.0 area, so no inter-area communications are needed.
- Router3 redistributes BGP routes into the OSPF AS and peers with the ISP BGP Router over eBGP. For information about configuring BGP, see BGP on page 646.
- The advertised networks - 10.11.101.0, 10.11.102.0, and 10.11.103.0 - are summarized by 10.11.0.0/16. Additional networks are advertised individually by the /24 subnet.

Router1

To configure Router1 in the GUI:

1. Go to *Network* > *OSPF*.
2. Set *Router ID* to *10.11.101.1*.
3. In the *Areas* table, click *Create New* and set the following:

Area ID	0.0.0.0
Type	Regular
Authentication	None

4. Click *OK*.
5. In the *Networks* table, click *Create New* and set the following:

Area	0.0.0.0
IP/Netmask	10.11.0.0 255.255.0.0

6. Click *OK*.
7. In the *Networks* table, click *Create New* again and set the following:

Area	0.0.0.0
IP/Netmask	192.168.102.0 255.255.255.0

8. Click *OK*.
9. In the *Interfaces* table, click *Create New* and set the following:

Name	Router1-Internal-DR
Interface	port1
Cost	0
Priority	255
Authentication	None
Timers	• Hello Interval: 10 • Dead Interval: 40

10. Click *OK*.

11. In the *Interfaces* table, click *Create New* again and set the following:

Name	Router1-External
Interface	port2
Cost	0
Authentication	None
Timers	• Hello Interval: 10 • Dead Interval: 40

12. Click *OK*.
13. Click *Apply*.

To configure Router1 in the CLI:

```
config router ospf
    set router-id 10.11.101.1
    config area
        edit 0.0.0.0
        next
    end
    config ospf-interface
        edit "Router1-Internal-DR"
            set interface "port1"
            set priority 255
            set dead-interval 40
            set hello-interval 10
        next
        edit "Router1-External"
            set interface "port2"
            set dead-interval 40
            set hello-interval 10
        next
    end
    config network
        edit 1
            set prefix 10.11.0.0 255.255.0.0
        next
        edit 2
            set prefix 192.168.102.0 255.255.255.0
        next
    end
end
```

Router2

To configure Router2 in the GUI:

1. Go to *Network* > *OSPF*.
2. Set *Router ID* to *10.11.101.2*.

3. In the *Areas* table, click *Create New* and set the following:

Area ID	0.0.0.0
Type	Regular
Authentication	None

4. Click *OK*.
5. In the *Networks* table, click *Create New* and set the following:

Area	0.0.0.0
IP/Netmask	10.11.0.0 255.255.0.0

6. Click *OK*.
7. In the *Networks* table, click *Create New* again and set the following:

Area	0.0.0.0
IP/Netmask	192.168.103.0 255.255.255.0

8. Click *OK*.
9. In the *Interfaces* table, click *Create New* and set the following:

Name	Router2-Internal
Interface	port1
Cost	0
Priority	250
Authentication	None
Timers	• Hello Interval: 10 • Dead Interval: 40

10. Click *OK*.
11. In the *Interfaces* table, click *Create New* again and set the following:

Name	Router2-External
Interface	port2
Cost	0
Authentication	None
Timers	• Hello Interval: 10 • Dead Interval: 40

12. Click *OK*.
13. Click *Apply*.

To configure Router2 in the CLI:

```
config router ospf
    set router-id 10.11.101.2
    config area
        edit 0.0.0.0
        next
    end
    config ospf-interface
        edit "Router2-Internal"
            set interface "port1"
            set priority 250
            set dead-interval 40
            set hello-interval 10
        next
        edit "Router2-External"
            set interface "port2"
            set dead-interval 40
            set hello-interval 10
        next
    end
    config network
        edit 1
            set prefix 10.11.0.0 255.255.0.0
        next
        edit 2
            set prefix 192.168.103.0 255.255.255.0
        next
    end
end
```

Router3

To configure Router3 in the GUI:

1. Go to *Network* > *OSPF*.
2. Set *Router ID* to *10.11.103.3*.
3. Under *Default Settings*, set *Inject default route* to *Regular Areas*.
   A default route must be present on Router3 to advertise it to other routers.
4. Enable *Redistribute BGP* and use the default settings.
5. In the *Areas* table, click *Create New* and set the following:

Area ID	0.0.0.0
Type	Regular
Authentication	None

6. Click *OK*.
7. In the *Networks* table, click *Create New* and set the following:

Area	0.0.0.0
IP/Netmask	10.11.0.0 255.255.0.0

8. Click *OK*.
9. In the *Interfaces* table, click *Create New* and set the following:

Name	Router3-Internal
Interface	port1
Cost	0
Authentication	None
Timers	• Hello Interval: 10 • Dead Interval: 40

10. Click *OK*.
11. In the *Interfaces* table, click *Create New* again and set the following:

Name	Router3-Internal2
Interface	port2
Cost	0
Authentication	None
Timers	• Hello Interval: 10 • Dead Interval: 40

12. Click *OK*.
13. Click *Apply*.

To configure Router3 in the CLI:

```
config router ospf
    set default-information-originate enable
    set router-id 10.11.103.3
    config area
        edit 0.0.0.0
        next
    end
    config ospf-interface
        edit "Router3-Internal"
            set interface "port1"
            set dead-interval 40
            set hello-interval 10
        next
        edit "Router3-Internal2"
            set interface "port2"
            set dead-interval 40
            set hello-interval 10
        next
```

```
    end
    config network
        edit 1
            set prefix 10.11.0.0 255.255.0.0
        next
    end
    config redistribute "bgp"
        set status enable
    end
end
```

To configure BGP on Router3 in the CLI:

```
config router bgp
    set as 64511
    set router-id 1.1.1.1
    config neighbor
        edit "172.20.120.5"
            set remote-as 64512
        next
    end
    config network
        edit 1
            set prefix 172.20.120.0 255.255.255.0
        next
    end
end
```

For more information on configuring BGP, see BGP on page 646.

Testing the configuration

Both the network connectivity and OSPF routing are tested. When a link goes down, routes should converge as expected.

Working state

- Router3:

```
Router3 # get router info ospf neighbor
OSPF process 0, VRF 0:
Neighbor ID     Pri   State           Dead Time   Address         Interface
10.11.101.1       1   Full/Backup     00:00:34    10.11.102.1     port1
10.11.101.2       1   Full/Backup     00:00:38    10.11.103.2     port2
```

```
Router3 # get router info ospf status
 Routing Process "ospf 0" with ID 10.11.103.3
 Process uptime is 18 hours 52 minutes
 Process bound to VRF default
 Conforms to RFC2328, and RFC1583Compatibility flag is disabled
```

```
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 Do not support Restarting
 This router is an ASBR (injecting external routing information)
 SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
 Refresh timer 10 secs
 Number of incomming current DD exchange neighbors 0/5
 Number of outgoing current DD exchange neighbors 0/5
 Number of external LSA 3. Checksum 0x021B78
 Number of opaque AS LSA 0. Checksum 0x000000
 Number of non-default external LSA 2
 External LSA database is unlimited.
 Number of LSA originated 16
 Number of LSA received 100
 Number of areas attached to this router: 1
    Area 0.0.0.0 (BACKBONE)
        Number of interfaces in this area is 2(2)
        Number of fully adjacent neighbors in this area is 2
        Area has no authentication
        SPF algorithm last executed 00:37:36.690 ago
        SPF algorithm executed 13 times
        Number of LSA 6. Checksum 0x03eafa
```

```
Router3 # get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default
Routing table for VRF=0
B*      0.0.0.0/0 [20/0] via 172.20.120.5, port3, 01:10:12
O       10.11.101.0/24 [110/2] via 10.11.103.2, port2, 00:39:34
                       [110/2] via 10.11.102.1, port1, 00:39:34
C       10.11.102.0/24 is directly connected, port1
C       10.11.103.0/24 is directly connected, port2
C       172.20.120.0/24 is directly connected, port3
O       192.168.102.0/24 [110/2] via 10.11.102.1, port1, 02:24:59
O       192.168.103.0/24 [110/2] via 10.11.103.2, port2, 02:14:32
B       192.168.160.0/24 [20/0] via 172.20.120.5, port3, 19:08:39
B       192.168.170.0/24 [20/0] via 172.20.120.5, port3, 01:10:12
```

- Router2:

```
Router2 # get router info ospf neighbor
OSPF process 0, VRF 0:
Neighbor ID     Pri   State           Dead Time   Address         Interface
10.11.101.1     255   Full/DR         00:00:35    10.11.101.1     port1
10.11.103.3       1   Full/DR         00:00:38    10.11.103.3     port3
```

```
Router2 # get router info ospf status
 Routing Process "ospf 0" with ID 10.11.101.2
 Process uptime is 2 hours 53 minutes
 Process bound to VRF default
 Conforms to RFC2328, and RFC1583Compatibility flag is disabled
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 Do not support Restarting
 SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
 Refresh timer 10 secs
 Number of incomming current DD exchange neighbors 0/5
 Number of outgoing current DD exchange neighbors 0/5
 Number of external LSA 3. Checksum 0x021979
 Number of opaque AS LSA 0. Checksum 0x000000
 Number of non-default external LSA 2
 External LSA database is unlimited.
 Number of LSA originated 5
 Number of LSA received 128
 Number of areas attached to this router: 1
    Area 0.0.0.0 (BACKBONE)
        Number of interfaces in this area is 3(3)
        Number of fully adjacent neighbors in this area is 2
        Area has no authentication
        SPF algorithm last executed 00:47:49.990 ago
        SPF algorithm executed 15 times
        Number of LSA 6. Checksum 0x03e8fb
```

```
Router2 # get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default
Routing table for VRF=0
O*E2    0.0.0.0/0 [110/10] via 10.11.103.3, port2, 01:03:58
C       10.11.101.0/24 is directly connected, port1
O       10.11.102.0/24 [110/2] via 10.11.103.3, port2, 00:49:01
                       [110/2] via 10.11.101.1, port1, 00:49:01
C       10.11.103.0/24 is directly connected, port2
O       192.168.102.0/24 [110/2] via 10.11.101.1, port1, 00:49:01
C       192.168.103.0/24 is directly connected, port3
O E2    192.168.160.0/24 [110/10] via 10.11.103.3, port2, 01:39:31
O E2    192.168.170.0/24 [110/10] via 10.11.103.3, port2, 01:19:39
```

The default route advertised by Router3 using `default-information-originate` is considered an OSPF E2 route. Other routes redistributed from BGP are also E2 routes.

- Router1:

```
Router1 # get router info ospf neighbor
OSPF process 0, VRF 0:
```

```
Neighbor ID     Pri   State           Dead Time   Address         Interface
10.11.101.2     250   Full/Backup     00:00:36    10.11.101.2     port1
10.11.103.3       1   Full/DR         00:00:37    10.11.102.3     port2
```

```
Router1 # get router info ospf status
 Routing Process "ospf 0" with ID 10.11.101.1
 Process uptime is 3 hours 7 minutes
 Process bound to VRF default
 Conforms to RFC2328, and RFC1583Compatibility flag is disabled
 Supports only single TOS(TOS0) routes
 Supports opaque LSA
 Do not support Restarting
 SPF schedule delay 5 secs, Hold time between two SPFs 10 secs
 Refresh timer 10 secs
 Number of incomming current DD exchange neighbors 0/5
 Number of outgoing current DD exchange neighbors 0/5
 Number of external LSA 3. Checksum 0x02157B
 Number of opaque AS LSA 0. Checksum 0x000000
 Number of non-default external LSA 2
 External LSA database is unlimited.
 Number of LSA originated 2
 Number of LSA received 63
 Number of areas attached to this router: 1
    Area 0.0.0.0 (BACKBONE)
        Number of interfaces in this area is 3(3)
        Number of fully adjacent neighbors in this area is 2
        Area has no authentication
        SPF algorithm last executed 00:54:08.160 ago
        SPF algorithm executed 11 times
        Number of LSA 6. Checksum 0x03e6fc
```

```
Router1 # get router info routing-table all
Routing table for VRF=0
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default
O*E2    0.0.0.0/0 [110/10] via 10.11.102.3, port2, 01:09:48
C       10.11.101.0/24 is directly connected, port1
C       10.11.102.0/24 is directly connected, port2
O       10.11.103.0/24 [110/2] via 10.11.102.3, port2, 00:54:49
                       [110/2] via 10.11.101.2, port1, 00:54:49
C       192.168.102.0/24 is directly connected, port3
O       192.168.103.0/24 [110/2] via 10.11.101.2, port1, 00:54:49
O E2    192.168.160.0/24 [110/10] via 10.11.102.3, port2, 01:45:21
O E2    192.168.170.0/24 [110/10] via 10.11.102.3, port2, 01:25:29
```

Link down state

If port1 is disconnected on Router3:

- Router3:

```
Router3 # get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default
Routing table for VRF=0
B*      0.0.0.0/0 [20/0] via 172.20.120.5, VLAN20, 01:29:25
O       10.11.101.0/24 [110/2] via 10.11.103.2, port2, 00:00:09
C       10.11.103.0/24 is directly connected, port2
C       172.20.120.0/24 is directly connected, port3
O       192.168.102.0/24 [110/3] via 10.11.103.2, port2, 00:00:09
O       192.168.103.0/24 [110/2] via 10.11.103.2, port2, 02:33:45
B       192.168.160.0/24 [20/0] via 172.20.120.5, port3, 19:27:52
B       192.168.170.0/24 [20/0] via 172.20.120.5, port3, 01:29:25
```

- Router2:

```
Router2 # get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default
Routing table for VRF=0
O*E2    0.0.0.0/0 [110/10] via 10.11.103.3, port2, 01:16:36
C       10.11.101.0/24 is directly connected, port1
O       10.11.102.0/24 [110/2] via 10.11.101.1, port1, 00:02:27
C       10.11.103.0/24 is directly connected, port2
O       192.168.102.0/24 [110/2] via 10.11.101.1, port1, 01:01:39
C       192.168.103.0/24 is directly connected, port3
O E2    192.168.160.0/24 [110/10] via 10.11.103.3, port2, 01:52:09
O E2    192.168.170.0/24 [110/10] via 10.11.103.3, port2, 01:32:17
```

- Router1:

```
Router1 # get router info routing-table all
Routing table for VRF=0
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default
O*E2    0.0.0.0/0 [110/10] via 10.11.101.2, port1, 00:05:14
C       10.11.101.0/24 is directly connected, port1
C       10.11.102.0/24 is directly connected, port2
O       10.11.103.0/24 [110/2] via 10.11.101.2, port1, 00:05:15
C       192.168.102.0/24 is directly connected, port3
```

```
O       192.168.103.0/24 [110/2] via 10.11.101.2, port1, 01:03:50
O E2    192.168.160.0/24 [110/10] via 10.11.101.2, port1, 00:05:14
O E2    192.168.170.0/24 [110/10] via 10.11.101.2, port1, 00:05:14
```

OSPFv3 neighbor authentication

OSPFv3 neighbor authentication is available for enhanced IPv6 security.

To configure an OSPF6 interface:

```
config router ospf6
    config ospf6-interface
        edit <name>
            set authentication {none | ah | esp | area}
            set key-rollover-interval <integer>
            set ipsec-auth-alg {md5 | sha1 | sha256 | sha384 | sha512}
            set ipsec-enc-alg {null | des | 3des | aes128 | aes192 | aes256}
            config ipsec-keys
                edit <spi>
                    set auth-key <string>
                    set enc-key <string>
                next
            end
        next
    end
end
```

To configure an OSPF6 virtual link:

```
config router ospf6
    config area
        edit <id>
            config virtual-link
                edit <name>
                    set authentication {none | ah | esp | area}
                    set key-rollover-interval <integer>
                    set ipsec-auth-alg {md5 | sha1 | sha256 | sha384 | sha512}
                    set ipsec-enc-alg {null | des | 3des | aes128 | aes192 | aes256}
                    config ipsec-keys
                        edit <spi>
                            set auth-key <string>
                            set enc-key <string>
                        next
                    end
                next
            end
        next
    end
end
```

To configure an OSPF6 area:

```
config router ospf6
    config area
        edit <id>
            set authentication {none | ah | esp}
            set key-rollover-interval <integer>
            set ipsec-auth-alg {md5 | sha1 | sha256 | sha384 | sha512}
            set ipsec-enc-alg {null | des | 3des | aes128 | aes192 | aes256}
            config ipsec-keys
                edit <spi>
                    set auth-key <string>
                    set enc-key <string>
                next
            end
        next
    end
end
```

CLI command descriptions

Command	Description
<id>	Area entry IP address.
authentication {none | ah | esp | area}	Authentication mode: • none: Disable authentication • ah: Authentication Header • esp: Encapsulating Security Payload • area: Use the routing area authentication configuration
key-rollover-interval <integer>	Enter an integer value (300 - 216000, default = 300).
ipsec-auth-alg {md5 | sha1 | sha256 | sha384 | sha512}	Authentication algorithm.
ipsec-enc-alg {null | des | 3des | aes128 | aes192 | aes256}	Encryption algorithm.
<spi>	Security Parameters Index.
auth-key <string>	Authentication key should be hexadecimal numbers. Key length for each algorithm: • MD5: 16 bytes • SHA1: 20 bytes • SHA256: 32 bytes • SHA384:48 bytes • SHA512:84 bytes If the key is shorter than the required length, it will be padded with zeroes.
enc-key <string>	Encryption key should be hexadecimal numbers. Key length for each algorithm:

Command	Description
	• DES: 8 bytes • 3DES: 24 bytes • AES128: 16 bytes • AES192: 24 bytes • AES256: 32 bytes If the key is shorter than the required length, it will be padded with zeroes.

OSPF graceful restart upon a topology change

In OSPF graceful restart mode, the `restart-on-topology-change` option can be used to keep restarting the router in graceful restart mode when a topology change is detected during a restart.

```
config router ospf
    set restart-on-topology-change {enable | disable}
end
```

OSPFv3 graceful restart mode upon a topology change can be used in OSPF6:

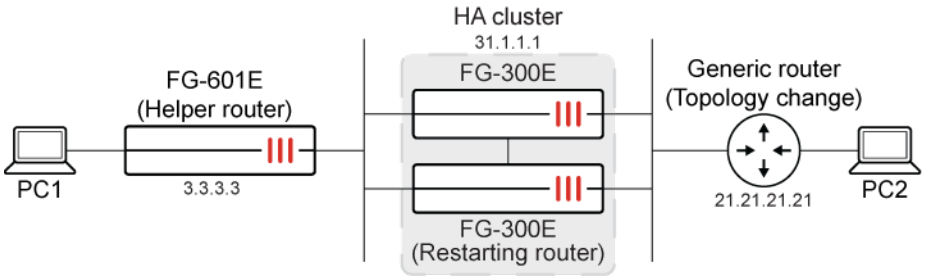
```
config router ospf6
    set restart-on-topology-change {enable | disable}
end
```

Example

In this example, a restarting router (one of the FG-300Es in the HA cluster) informs its neighbors using grace LSAs before restarting its OSPF process. When the helper router (the FG-601E) receives the grace LSAs, it enters helper mode to help with the graceful restart until the graceful period expires. It will act as though there are no changes on the restarting router (FG-300E). A generic router simulates a topology change during the restart event.

If `restart-on-topology-change` is enabled on the restarting router, it will not exit the graceful restart mode even when a topology change is detected.

If `restart-on-topology-change` is disabled on the restarting router, it will exit graceful restart mode when a topology change is detected.



To configure the restarting router:

```
config router ospf
    set router-id 31.1.1.1
    set restart-mode graceful-restart
    set restart-period 180
    set restart-on-topology-change enable
    config area
        edit 0.0.0.0
        next
    end
    config network
        edit 1
            set prefix 172.16.200.0 255.255.255.0
        next
        edit 2
            set prefix 31.1.1.1 255.255.255.255
        next
    end
end
```

To configure the restarting helper router:

```
config router ospf
    set router-id 3.3.3.3
    set restart-mode graceful-restart
    config area
        edit 0.0.0.0
        next
    end
    config network
        edit 1
            set prefix 172.16.200.0 255.255.255.0
        next
        edit 2
            set prefix 3.3.3.3 255.255.255.255
        next
    end
end
```

Testing the configuration

Topology change with continuing graceful restart enabled:

When `restart-on-topology-change` is enabled and there is a topology change during the HA OSPF graceful restart, the graceful restart will continue. The routes on the helper router (FG-601E) are still there and no traffic will drop.

```
# get router info ospf neighbor
OSPF process 0, VRF 0:
```

```
Neighbor ID     Pri   State           Dead Time   Address         Interface
31.1.1.1          1   Full/DR         00:14:47*   172.16.200.31   port1
```

```
# get router info routing-table ospf
Routing table for VRF=0
O       21.21.21.21/32 [110/300] via 172.16.200.31, port1, 00:09:55
O       31.1.1.1/32 [110/200] via 172.16.200.31, port1, 00:55:31
O       100.21.1.0/24 [110/200] via 172.16.200.31, port1, 00:12:31
```

```
# get router info ospf neighbor
OSPF process 0, VRF 0:
Neighbor ID     Pri   State           Dead Time   Address         Interface
31.1.1.1          1   Full/DR         00:14:47*   172.16.200.31   port1
```

```
# get router info routing-table ospf
Routing table for VRF=0
O       21.21.21.21/32 [110/300] via 172.16.200.31, port1, 00:10:07
O       31.1.1.1/32 [110/200] via 172.16.200.31, port1, 00:55:43
O       100.21.1.0/24 [110/200] via 172.16.200.31, port1, 00:12:43
```

```
# get router info ospf neighbor
OSPF process 0, VRF 0:
Neighbor ID     Pri   State           Dead Time   Address         Interface
31.1.1.1          1   Full/DR         00:14:38*   172.16.200.31   port1
```

```
# get router info routing-table ospf
Routing table for VRF=0
O       21.21.21.21/32 [110/300] via 172.16.200.31, port1, 00:10:17
O       31.1.1.1/32 [110/200] via 172.16.200.31, port1, 00:55:53
O       100.21.1.0/24 [110/200] via 172.16.200.31, port1, 00:12:53
```

```
# get router info ospf neighbor
OSPF process 0, VRF 0:
Neighbor ID     Pri   State           Dead Time   Address         Interface
31.1.1.1          1   Full/DR         00:00:38    172.16.200.31   port1
```

```
# get router info routing-table ospf
Routing table for VRF=0
O       21.21.21.21/32 [110/300] via 172.16.200.31, port1, 00:10:37
O       31.1.1.1/32 [110/200] via 172.16.200.31, port1, 00:56:13
O       100.21.1.0/24 [110/200] via 172.16.200.31, port1, 00:13:13
```

Topology change with continuing graceful restart disabled:

When `restart-on-topology-change` is disabled and there is a topology change during the HA OSPF graceful restart, the graceful restart will exit. The routes on the helper router (FG-601E) are lost and traffic will drop.

```
# get router info ospf neighbor
OSPF process 0, VRF 0:
Neighbor ID     Pri   State           Dead Time   Address         Interface
31.1.1.1          1   Full/DR         00:14:57*   172.16.200.31   port1
```

```
# get router info routing-table ospf
Routing table for VRF=0
O       21.21.21.21/32 [110/300] via 172.16.200.31, port1, 00:11:16
O       31.1.1.1/32 [110/200] via 172.16.200.31, port1, 00:56:52
O       100.21.1.0/24 [110/200] via 172.16.200.31, port1, 00:13:52
```

```
# get router info ospf neighbor
OSPF process 0, VRF 0:
Neighbor ID     Pri   State           Dead Time   Address         Interface
31.1.1.1          1   Full/DR         00:14:42*   172.16.200.31   port1
```

```
# get router info routing-table ospf
Routing table for VRF=0
O       21.21.21.21/32 [110/300] via 172.16.200.31, port1, 00:11:31
O       31.1.1.1/32 [110/200] via 172.16.200.31, port1, 00:57:07
O       100.21.1.0/24 [110/200] via 172.16.200.31, port1, 00:14:07
```

```
# get router info ospf neighbor
OSPF process 0, VRF 0:
Neighbor ID     Pri   State           Dead Time   Address         Interface
31.1.1.1          1   Full/DR         00:14:40*   172.16.200.31   port1
```

No routes are lost:

```
# get router info routing-table ospf
Routing table for VRF=0
O       31.1.1.1/32 [110/200] via 172.16.200.31, port1, 00:57:09
```

```
# get router info ospf neighbor
OSPF process 0, VRF 0:
Neighbor ID     Pri   State           Dead Time   Address         Interface
31.1.1.1          1   Full/DR         00:14:38*   172.16.200.31   port1
```

No routes are lost:

```
# get router info routing-table ospf
Routing table for VRF=0
O       31.1.1.1/32 [110/200] via 172.16.200.31, port1, 00:57:11
```

No routes are lost:

```
# get router info routing-table ospf
Routing table for VRF=0
O       21.21.21.21/32 [110/300] via 172.16.200.31, port1, 00:04:42
O       31.1.1.1/32 [110/200] via 172.16.200.31, port1, 01:01:59
O       100.21.1.0/24 [110/200] via 172.16.200.31, port1, 00:04:42
```

OSPF link detection customization

OSPF route detection and interface-down detection can be customized by customizing the Link State Advertisement (LSA) refresh intervals and enabling fast-link failover using the `config router ospf` command:

```
config router ospf
    set lsa-refresh-interval <integer>
    config ospf-interface
        edit <name>
            set interface <string>
            set linkdown-fast-failover {enable | disable}
        next
    end
end
```

Command	Description
`lsa-refresh-interval <integer>`	How often LSA refreshes for OSPF, in seconds (0 to 5, default = 5).
`linkdown-fast-failover {enable | disable}`	Enable/disable fast link failover. • `enable`: OSPF updates the link status from `up` to `down` and advertises the LSA update as soon as the underlying physical interface goes down. • `disable`: Disable the use of OSPF and use the kernal detection and notification instead. Used when the `ospf-interface` interface attribute is configured, and the type of the underlying interface is VLAN.

BGP

Border Gateway Protocol (BGP) is a standardized routing protocol that is used to route traffic across the internet. It exchanges routing information between Autonomous Systems (AS) on the internet and makes routing decisions based on path, network policies, and rule sets. BGP contains two distinct subsets: internal BGP (iBGP) and external BGP (eBGP). iBGP is intended for use within your own networks. eBGP is used to connect different networks together and is the main routing protocol for the internet backbone.

To configure BGP in the GUI, go to *Network* > *BGP*:

Option	Description
Local AS	The AS number for the local router.
Router ID	A unique ID to identify your router in the network, typically in the format x.x.x.x.
Neighbors	The neighbors that the FortiGate will be peering with. Configure the remote router's AS number, any other properties used for peering with the neighbor, and IPv4 and IPv6 filtering.

Option	Description
Neighbor Groups	The neighbor groups that share the same outbound policy configurations.
Neighbor Ranges	The source address range of BGP neighbors that will be automatically assigned to a neighbor group.
Networks	The IPv4 networks to be advertised to other BGP routers. Choose from: • *IP/Netmask*: Specify the IPv4 address and netmask. • *Firewall Address*: Select an existing firewall address or group with `allow-routing` enabled. See Using firewall addresses and groups for BGP network prefixes on page 703.
IPv6 Networks	The IPv6 networks to be advertised to other BGP routers.
IPv4 & IPv6 Redistribute	Enable redistribution by protocol. Specify either *All* routes, or *Filter* by route map.
Dampening	Enable route flap dampening to reduce the propagation of flapping routes.
Graceful Restart	Enable BGP graceful restart, which causes the adjacent routers to keep routes active while the BGP peering is restarted on the FortiGate. This is useful in HA instances when failover occurs.
Advanced Options	Various advanced settings, such as *Local Preference*, *Distance internal*, *Keepalive*, *Holdtime*, and others
Best Path Selection	Configure path selection attributes on this router.

This section includes the following topics:

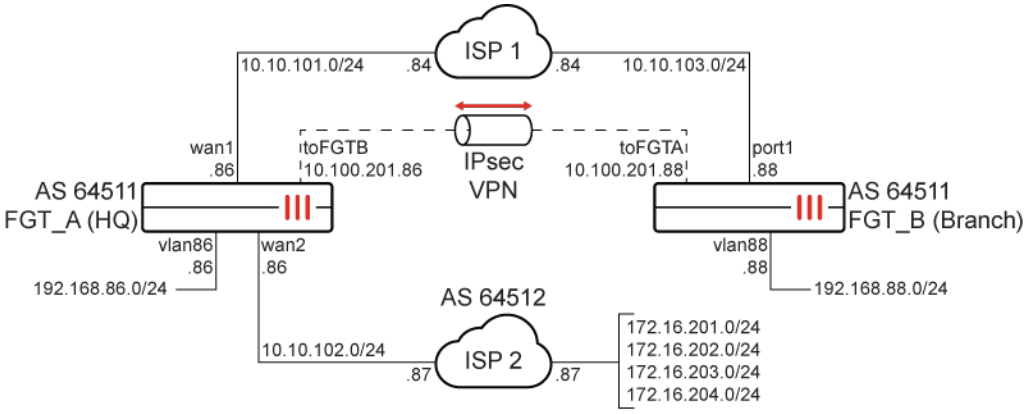
- Basic BGP example on page 648
- Route filtering with a distribution list on page 657
- Next hop recursive resolution using other BGP routes on page 661
- Next hop recursive resolution using ECMP routes on page 662
- Next hop recursive resolution using IPv6 prefix with on-link flag from route aggregation on page 663
- BGP conditional advertisement on page 666
- BGP error handling per RFC 7606 on page 681
- BGP next hop tag-match mode on page 682
- BGP next hop self-support for VPNv4 and VPNv6 route reflectors New on page 688
- BGP neighbor password on page 689
- BGP neighbor naming support New on page 690
- Defining a preferred source IP for local-out egress interfaces on BGP routes on page 693
- BGP multi-exit discriminator on page 695
- TCP Authentication Option advanced security measures on page 700
- Assigning multiple remote Autonomous Systems to a single BGP neighbor group on page 701
- Using firewall addresses and groups for BGP network prefixes on page 703
- Dynamic BGP learning of ISDB reputation IPs New on page 705
- Graceful BGP shutdown NEW on page 707
- Troubleshooting BGP on page 715

Basic BGP example

In this example, BGP is configured on two FortiGate devices. The FortiGates are geographically separated, and form iBGP peering over a VPN connection. FGT_A also forms eBGP peering with ISP2.

FGT_A learns routes from ISP2 and redistributes them to FGT_B while preventing any iBGP routes from being advertised.

The internal networks behind the FortiGates can communicate with each other, and the internal networks behind FGT_B can traverse FGT_A to reach networks that are advertised by ISP2.



- FGT_A and FGT_B have static routes to each other through ISP1. ISP1 does not participate in BGP.
- The IPsec VPN tunnel between FGT_A and FGT_B is configured with wildcard 0.0.0.0/0 networks for phase2 local and remote selectors. The VPN interfaces have IP addresses already configured and are used for peering between FGT_A and FGT_B.
- FGT_A is configure to peer with ISP2 on 10.10.102.86.
- The firewall policies between FGT_A and FGT_B are not NATed. The firewall policies egressing on wan2 are NATed.

Configuring iBGP peering

To configure FGT_A to establish iBGP peering with FGT_B in the GUI:

1. Go to *Network* > *BGP*.
2. Set *Local AS* to *64511*
3. Set *Router ID* to *1.1.1.1*.
4. In the *Neighbors* table, click *Create New* and set the following:

IP	10.100.201.88
Remote AS	64511

5. Click *OK*.
6. Under *Networks*, set *IP/Netmask* to *192.168.86.0/24*.
7. Click *Apply*.
8. In the CLI, set the interface used as the source IP address of the TCP connection (where the BGP session, TCP/179, is connecting from) for the neighbor (`update-source`) to `toFGTB`.

To configure FGT_A to establish iBGP peering with FGT_B in the CLI:

```
config router bgp
    set as 64511
    set router-id 1.1.1.1
    config neighbor
        edit "10.100.201.88"
            set remote-as 64511
            set update-source "toFGTB"
        next
    end
    config network
        edit 1
            set prefix 192.168.86.0 255.255.255.0
        next
    end
end
```

To configure FGT_B to establish iBGP peering with FGT_A in the GUI:

1. Go to *Network* > *BGP*.
2. Set *Local AS* to *64511*
3. Set *Router ID* to *2.2.2.2*.
4. In the *Neighbors* table, click *Create New* and set the following:

IP	10.100.201.86
Remote AS	64511

5. Click *OK*.
6. Under *Networks*, set *IP/Netmask* to *192.168.88.0/24*.
7. Click *Apply*.
8. In the CLI, set the interface used as the source IP address of the TCP connection (where the BGP session, TCP/179, is connecting from) for the neighbor (`update-source`) to `toFGTA`.

To configure FGT_B to establish iBGP peering with FGT_A in the CLI:

```
config router bgp
    set as 64511
    set router-id 2.2.2.2
    config neighbor
        edit "10.100.201.86"
            set remote-as 64511
            set update-source "toFGTA"
        next
    end
    config network
        edit 1
            set prefix 192.168.88.0 255.255.255.0
        next
```

```
    end
end
```

To check the FGT_A and FGT_B peering:

1. Check the BGP neighbors:

```
# get router info bgp neighbors
```

2. Check the networks learned from neighbors:

```
# get router info bgp network
```

3. Check that the routes are added to the routing table:

```
# get router info routing-table all
```

To see the neighborship status, network, and routing table command outputs for the completed example, see Troubleshooting and debugging on page 652.

Configuring eBGP peering

By establishing eBGP peering with ISP2, learned routes will have a distance of 20 and will automatically be propagated to iBGP peers. iBGP peers do not change the next hop when they advertise a route. To make FGT_B receive a route with FGT_A as the next hop, and not ISP 2's network, *Next hop self* (`next-hop-self`) is enabled for routes advertised to FGT_B.

Additionally, to peer with another router that is multiple hops away, enable `ebg-enforce-multihop` in the neighbor configuration.

In this example, the iBGP routes are automatically advertised to the eBGP neighbor, so a route map is created to deny iBGP routes from being advertised to ISP 2. Prefixes from ISP 2 are advertised to FGT_A and FGT_B, but no prefixes are advertised from FGT_A to ISP 2.

To configure FGT_A to establish eBGP peering with ISP 2 in the GUI:

1. Configure a route map to prevent advertisement of iBGP routes to ISP 2:
   a. Go to *Network* > *Routing Objects* and click *Create New* > *Route Map*.
   b. Set *Name* to *exclude1*.
   c. In the *Rules* table, click *Create New*.
   d. Set *Action* to *Deny*.
   e. Under *Other Rule Variables*, enable *Match origin* and set it to *IGP*.
   f. Click *OK*.
   g. Click *OK*.
2. Update the BGP configuration:
   a. Go to *Network* > *BGP*.
   b. In the *Neighbors* table, click *Create New* and set the following:

IP	10.10.102.87

Remote AS	64512
Route map out	exclude1

c. Click *OK*.
d. In the *Neighbors* table, edit the previously created entry, *10.100.201.88*.
e. Under *IPv4 Filtering*, select *Next hop self*.
f. Click *OK*.
g. Click *Apply*.

To configure FGT_A to establish eBGP peering with ISP 2 in the CLI:

1. Configure a route map to prevent advertisement of iBGP routes to ISP 2:

```
config router route-map
    edit "exclude1"
        config rule
            edit 1
                set action deny
                set match-origin igp
            next
        end
    next
end
```

2. Update the BGP configuration:

```
config router bgp
    config neighbor
        edit "10.10.102.87"
            set remote-as 64512
            set route-map-out "exclude1"
        next
        edit "10.100.201.88"
            set next-hop-self enable
        next
    end
end
```

To see the neighborship status, network, and routing table command outputs for the completed example, see Troubleshooting and debugging on page 652.

Firewall policies

On FGT_A configure the following policies:

- Allow the internal subnet to the VPN interface. Do not enable NAT. Enable security profiles as required.
- Allow the VPN interface to the internal subnet. Do not enable NAT. Enable security profiles as required.
- Allow the internal subnet to wan2. Enable NAT and security profiles as required.
- Allow VPN traffic from toFGTA to wan2. Enable NAT and security profiles as required.

On FGT_B configure the following policies:

- Allow the internal subnet to the VPN interface. Do not enable NAT. Enable security profiles as required.
- Allow the VPN interface to the internal subnet. Do not enable NAT. Enable security profiles as required.

To verify that pinging from FGT_B to FGT_A is successful:

```
FGT_B # execute ping-options source 192.168.88.88
FGT_B # execute ping 192.168.86.86
PING 192.168.86.86 (192.168.86.86): 56 data bytes
64 bytes from 192.168.86.86: icmp_seq=0 ttl=255 time=0.5 ms
...
--- 192.168.86.86 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.3/0.3/0.5 ms
```

To verify that pinging from FGT_B to a subnet in ISP 2 is successful:

```
FGT_B # execute ping-options source 192.168.88.88
FGT_B # execute ping 172.16.201.87
PING 172.16.201.87 (172.16.201.87): 56 data bytes
64 bytes from 172.16.201.87: icmp_seq=0 ttl=254 time=0.6 ms
...
--- 172.16.201.87 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.4/0.4/0.6 ms

FGT_B # execute traceroute-options source 192.168.88.88
FGT_B # execute traceroute 172.16.201.87
traceroute to 172.16.201.87 (172.16.201.87), 32 hops max, 3 probe packets per hop, 84 byte packets
 1  10.100.201.86  0.315 ms  0.143 ms  0.110 ms
 2  172.16.201.87  0.258 ms  0.144 ms  0.222 ms
```

Troubleshooting and debugging

When troubleshooting issues, logically step through the debugs. For example, if peering cannot be established between FGT_A and FGT_B:

1. Verify the basic connectivity between the FGT_A wan1 interface and the FGT_B port1 interface.
2. Verify that the VPN between FGT_A and FGT_B is established.
3. Verify the connectivity between the VPN interfaces.
4. Check the neighborship status on each peer. Use the BGP state to help determine the possible issue, for example:

Idle state	The local FortiGate has not started the BGP process with the neighbor. This could be because the eBGP peer is multiple hops away, but multihop is not enabled.
Connect	The local FortiGate has started the BGP process, but has not initiated a TCP connection, possibly due to improper routing.
Active	The local FortiGate has initiated a TCP connection, but there is no response. This might indicate issues with the delivery or the response from the remote peer.

5. If there are issues establishing the TCP connection, use the command `diagnose sniffer packet any 'tcp and port 179'` to identify the problem at the packet level.

The following outputs show instances where all of the configurations are completed, peering has formed, and routes have been exchanged. The debug output during each configuration step might differ from these outputs. These debug outputs can be used to help identify what might be missing or misconfigured on your device.

To verify the status of the neighbors:

```
FGT_A # get router info bgp neighbors
VRF 0 neighbor table:
BGP neighbor is 10.10.102.87, remote AS 64512, local AS 64511, external link
  BGP version 4, remote router ID 192.168.2.87
  BGP state = Established, up for 01:54:37
  Last read 00:00:29, hold time is 180, keepalive interval is 60 seconds
  Configured hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received (old and new)
    Address family IPv4 Unicast: advertised and received
    Address family IPv6 Unicast: advertised and received
  Received 513 messages, 1 notifications, 0 in queue
  Sent 517 messages, 2 notifications, 0 in queue
  Route refresh request: received 0, sent 0
  Minimum time between advertisement runs is 30 seconds
 For address family: IPv4 Unicast
  BGP table version 5, neighbor version 0
  Index 3, Offset 0, Mask 0x8
  Community attribute sent to this neighbor (both)
  Outbound path policy configured
  Route map for outgoing advertisements is *exclude1root
  4 accepted prefixes, 4 prefixes in rib
  0 announced prefixes
 For address family: IPv6 Unicast
  BGP table version 1, neighbor version 0
  Index 3, Offset 0, Mask 0x8
  Community attribute sent to this neighbor (both)
  0 accepted prefixes, 0 prefixes in rib
  0 announced prefixes
 Connections established 4; dropped 3
Local host: 10.10.102.86, Local port: 20364
Foreign host: 10.10.102.87, Foreign port: 179
Nexthop: 10.10.102.86
Nexthop interface: wan2
Nexthop global: ::
Nexthop local: ::
BGP connection: non shared network
Last Reset: 01:54:42, due to BGP Notification sent
Notification Error Message: (CeaseUnspecified Error Subcode)
BGP neighbor is 10.100.201.88, remote AS 64511, local AS 64511, internal link
  BGP version 4, remote router ID 2.2.2.2
  BGP state = Established, up for 01:54:07
  Last read 00:00:11, hold time is 180, keepalive interval is 60 seconds
```

```
  Configured hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received (old and new)
    Address family IPv4 Unicast: advertised and received
    Address family IPv6 Unicast: advertised and received
  Received 527 messages, 3 notifications, 0 in queue
  Sent 543 messages, 8 notifications, 0 in queue
  Route refresh request: received 0, sent 0
  Minimum time between advertisement runs is 30 seconds
  Update source is toFGTB
 For address family: IPv4 Unicast
  BGP table version 5, neighbor version 4
  Index 1, Offset 0, Mask 0x2
  NEXT_HOP is always this router
  Community attribute sent to this neighbor (both)
  1 accepted prefixes, 1 prefixes in rib
  5 announced prefixes
 For address family: IPv6 Unicast
  BGP table version 1, neighbor version 1
  Index 1, Offset 0, Mask 0x2
  Community attribute sent to this neighbor (both)
  0 accepted prefixes, 0 prefixes in rib
  0 announced prefixes
 Connections established 7; dropped 6
Local host: 10.100.201.86, Local port: 179
Foreign host: 10.100.201.88, Foreign port: 6245
Nexthop: 10.100.201.86
Nexthop interface: toFGTB
Nexthop global: ::
Nexthop local: ::
BGP connection: non shared network
Last Reset: 01:54:12, due to BGP Notification received
Notification Error Message: (CeaseUnspecified Error Subcode)
```

```
FGT_B # get router info bgp neighbors
VRF 0 neighbor table:
BGP neighbor is 10.100.201.86, remote AS 64511, local AS 64511, internal link
  BGP version 4, remote router ID 1.1.1.1
  BGP state = Established, up for 01:56:04
  Last read 00:00:48, hold time is 180, keepalive interval is 60 seconds
  Configured hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
    Route refresh: advertised and received (old and new)
    Address family IPv4 Unicast: advertised and received
    Address family IPv6 Unicast: advertised and received
  Received 532 messages, 3 notifications, 0 in queue
  Sent 526 messages, 3 notifications, 0 in queue
  Route refresh request: received 0, sent 0
  Minimum time between advertisement runs is 30 seconds
  Update source is toFGTA
 For address family: IPv4 Unicast
  BGP table version 4, neighbor version 3
```

```
  Index 1, Offset 0, Mask 0x2
  Community attribute sent to this neighbor (both)
  5 accepted prefixes, 5 prefixes in rib
  1 announced prefixes
 For address family: IPv6 Unicast
  BGP table version 1, neighbor version 1
  Index 1, Offset 0, Mask 0x2
  Community attribute sent to this neighbor (both)
  0 accepted prefixes, 0 prefixes in rib
  0 announced prefixes
 Connections established 7; dropped 6
Local host: 10.100.201.88, Local port: 6245
Foreign host: 10.100.201.86, Foreign port: 179
Nexthop: 10.100.201.88
Nexthop interface: toFGTA
Nexthop global: ::
Nexthop local: ::
BGP connection: non shared network
Last Reset: 01:56:09, due to BGP Notification sent
Notification Error Message: (CeaseUnspecified Error Subcode)
```

`# get router info bgp neighbors <neighbor's IP>` can also be used to verify the status of a specific neighbor.

To verify the networks learned from neighbors or a specific network:

```
FGT_A # get router info bgp network
VRF 0 BGP table version is 5, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight RouteTag Path
*> 172.16.201.0/24  10.10.102.87             0             0        0 64512 i <-/1>
*> 172.16.202.0/24  10.10.102.87             0             0        0 64512 i <-/1>
*> 172.16.203.0/24  10.10.102.87             0             0        0 64512 i <-/1>
*> 172.16.204.0/24  10.10.102.87             0             0        0 64512 i <-/1>
*> 192.168.86.0     0.0.0.0                       100  32768        0 i <-/1>
*>i192.168.88.0     10.100.201.88            0    100      0        0 i <-/1>
Total number of prefixes 6
FGT_A # get router info bgp network 172.16.201.0
VRF 0 BGP routing table entry for 172.16.201.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
   10.100.201.88
  Original VRF 0
  64512
    10.10.102.87 from 10.10.102.87 (192.168.2.87)
      Origin IGP metric 0, localpref 100, valid, external, best
      Last update: Tue Dec 15 22:52:08 2020
```

```
FGT_A # get router info bgp network 192.168.88.0
VRF 0 BGP routing table entry for 192.168.88.0/24
```

```
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local
    10.100.201.88 from 10.100.201.88 (2.2.2.2)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Last update: Tue Dec 15 22:52:39 2020
```

```
FGT_B # get router info bgp network
VRF 0 BGP table version is 4, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network          Next Hop            Metric LocPrf Weight RouteTag Path
*>i172.16.201.0/24  10.100.201.86            0    100      0        0 64512 i <-/1>
*>i172.16.202.0/24  10.100.201.86            0    100      0        0 64512 i <-/1>
*>i172.16.203.0/24  10.100.201.86            0    100      0        0 64512 i <-/1>
*>i172.16.204.0/24  10.100.201.86            0    100      0        0 64512 i <-/1>
*>i192.168.86.0     10.100.201.86            0    100      0        0 i <-/1>
*> 192.168.88.0     0.0.0.0                       100  32768        0 i <-/1>
Total number of prefixes 6
```

To verify the routing tables on FGT_A and FGT_B:

```
FGT_A # get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default
Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 172.16.151.1, port1, [5/0]
                  [10/0] via 192.168.2.1, port2, [10/0]
C       10.10.101.0/24 is directly connected, wan1
C       10.10.102.0/24 is directly connected, wan2
S       10.10.103.0/24 [10/0] via 10.10.101.84, wan1
C       10.100.201.0/24 is directly connected, toFGTB
C       10.100.201.86/32 is directly connected, toFGTB
C       172.16.151.0/24 is directly connected, port1
B       172.16.201.0/24 [20/0] via 10.10.102.87, wan2, 02:09:50
B       172.16.202.0/24 [20/0] via 10.10.102.87, wan2, 02:09:50
B       172.16.203.0/24 [20/0] via 10.10.102.87, wan2, 02:09:50
B       172.16.204.0/24 [20/0] via 10.10.102.87, wan2, 02:09:50
C       192.168.2.0/24 is directly connected, port2
C       192.168.86.0/24 is directly connected, vlan86
B       192.168.88.0/24 [200/0] via 10.100.201.88, toFGTB, 02:09:19
```

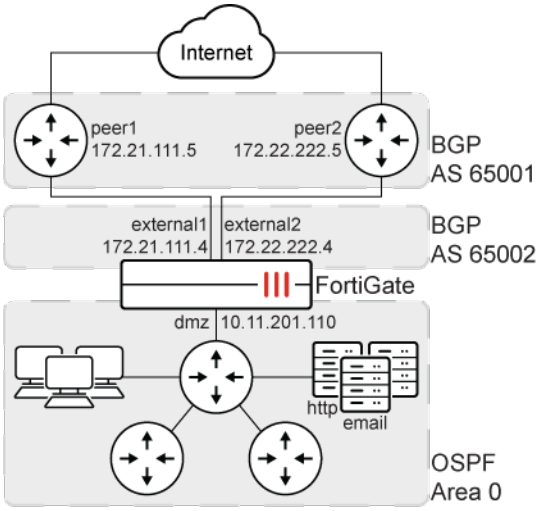
```
FGT_B # get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
```

```
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default
Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 10.10.103.84, port1
C       10.10.103.0/24 is directly connected, port1
C       10.100.201.0/24 is directly connected, toFGTA
C       10.100.201.88/32 is directly connected, toFGTA
B       172.16.201.0/24 [200/0] via 10.100.201.86, toFGTA, 02:11:36
B       172.16.202.0/24 [200/0] via 10.100.201.86, toFGTA, 02:11:36
B       172.16.203.0/24 [200/0] via 10.100.201.86, toFGTA, 02:11:36
B       172.16.204.0/24 [200/0] via 10.100.201.86, toFGTA, 02:11:36
B       192.168.86.0/24 [200/0] via 10.100.201.86, toFGTA, 02:11:36
C       192.168.88.0/24 is directly connected, vlan88
```

Route filtering with a distribution list

During BGP operations, routes can be propagated between BGP peers and redistributed from other routing protocols. In some situations, advertising routes from one peer to another might need to be prevented.

The Basic BGP example on page 648 explains using a route map to filter routes that are learned from iBGP to prevent them from propagating to an eBGP peer. In this example, a distribution list is used to prevent certain routes from one peer from being advertised to another peer.



- A company has its own web and email servers in an OSPF area, and needs to advertise routes to these resources to external peers. Users, routers, and other server all reside in the OSPF area.
- The FortiGate acts as the BGP border router, redistributing routes from the company's network to its BGP peers. It is connected to the OSPF area using its DMZ interface.
- Two ISP managed BGP peers in an AS (Peer 1 and Peer 2) are used to access the internet, and routes must not to be advertised from Peer 1 to Peer 2. The manufacturers of these routers, and information about other devices on the external BGP AS, are not known.
- Routes to the BGP peers are redistributed so that external locations can access the web and email servers in the OSPF area. The FortiGate device's external interfaces and the BGP peers are in different ASs, and

form eBGP peers.

- Other networking devices must be configured for BGP. The peer routers must be updated with the FortiGate device's BGP information, including IP addresses, AS number, and any specific capabilities that are used, such as IPv6, graceful restart, BFD, and so on.
- It is assumed that security policies have been configured to allow traffic between the networks and NAT is not used. To tighten security, only the required services should be allowed inbound to the various servers.
- In a real life scenario, public IP addresses would be used in place of private IP addresses.

Configuring BGP

In this example, Peer 1 routes are blocked from being advertised to Peer 2 using an access list. All incoming routes from Peer 1 are blocked when updates are sent to Peer 2.

Routes learned from OSPF are redistributed into BGP. EBGP multi path is enabled to load-balance traffic between the peers using ECMP. See Equal cost multi-path on page 595 for more information.

To configure BGP in the GUI:

1. Configure an access list to block Peer 1 routes:
   a. Go to *Network* > *Routing Objects* and click *Create New* > *Access List*.
   b. Set *Name* to *block_peer1*.
   c. In the *Rules* table, click *Create New*.
   d. Set *Action* to *Deny*.
   e. Enable *Exact Match* and specify the prefix *172.21.111.0 255.255.255.0*.
   f. Click *OK*.
   g. Click *OK*.
2. Configure BGP:
   a. Go to *Network* > *BGP*.
   b. Set *Local AS* to *65001*
   c. Set *Router ID* to *10.11.201.110*.
   d. In the *Neighbors* table, click *Create New* and set the following:

IP	172.21.111.5
Remote AS	65001

   e. Click *OK*.
   f. In the *Neighbors* table, click *Create New* again and set the following:

IP	172.22.222.5
Remote AS	65001
Distribute list out	Enable, and select the *block_peer1* access list.

   g. Click *OK*.
   h. Under *IPv4 Redistribute*, enable *OSPF* and select *ALL*.
   i. Expand *Best Path Selection* and enable *EBGP multi path*.
   j. Click *Apply*.

To configure BGP in the CLI:

1. Configure an access list to block Peer 1 routes:

```
config router access-list
    edit "block_peer1"
        config rule
            edit 1
                set action deny
                set prefix 172.21.111.0 255.255.255.0
                set exact-match enable
            next
        end
    next
end
```

2. Configure BGP:

```
config router bgp
    set as 65001
    set router-id 10.11.201.110
    set ebgp-multipath enable
    config neighbor
        edit "172.21.111.5"
            set remote-as 65001
        next
        edit "172.22.222.5"
            set distribute-list-out "block_peer1"
            set remote-as 65001
        next
    end
    config redistribute "ospf"
        set status enable
    end
end
```

Configuring OSPF

In this example, all of the traffic is within the one OSPF area, and there are other OSPF routers in the network. When adjacencies are formed, other routers receive the routes advertised from the FortiGate that are redistributed from BGP.

To configure OSPF in the GUI:

1. Go to *Network* > *OSPF*.
2. Set *Router ID* to *10.11.201.110*.
3. In the *Areas* table, click *Create New* and set the following:

Area ID	0.0.0.0
Type	Regular

Authentication	None

4. Click *OK*.
5. In the *Networks* table, click *Create New* and set the following:

Area	0.0.0.0
IP/Netmask	10.11.201.0 255.255.255.0

6. Click *OK*.
7. In the *Interfaces* table, click *Create New* and set the following:

Name	OSPF_dmz_network
Interface	dmz

8. Click *OK*.
9. Enable *Redistribute BGP* and set *Metric value* to *1*.
10. Click *Apply*.

To configure OSPF in the CLI:

```
config router ospf
    set router-id 10.11.201.110
    config area
        edit 0.0.0.0
        next
    end
    config ospf-interface
        edit "OSPF_dmz_network"
            set interface "dmz"
        next
    end
    config network
        edit 1
            set prefix 10.11.201.0 255.255.255.0
        next
    end
    config redistribute "bgp"
        set status enable
        set metric 1
    end
end
```

Testing the configuration

To test this configuration, run the standard connectivity checks, and also make sure that routes are being passed between protocols as expected. Use the following checklist to help verify that the FortiGate is configured successfully:

1. Check that the FortiGate has established peering with BGP Peer 1 and Peer 2:

```
# get router info bgp summary
```

```
# get router info bgp neighbors
```

2. Check that the FortiGate has formed adjacency with OSPF neighbors:

```
# get router info ospf status
```

```
# get router info ospf neighbors
```

3. Check the routing table on the FortiGate to make sure that routes from both OSPF and BGP are included:

```
# get router info routing-table all
```

4. Check devices in the OSPF network for internet connectivity and to confirm that routes redistributed from BGP are in their routing tables.
5. Check the routing table on Peer 2 to confirm that no routes from Peer 1 are included.
6. Check that the routes from the internal OSPF network are redistributed to Peer 1 and Peer 2.
7. Verify connectivity to the HTTP and email servers.

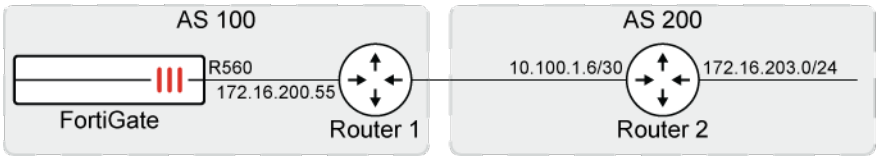
Next hop recursive resolution using other BGP routes

By default, BGP routes are not considered when a BGP next hop requires recursive resolution. They are considered when `recursive-next-hop` is enabled. Recursive resolution will resolve to one level.

To consider BGP routes for recursive resolution of next hops:

```
config router bgp
    set recursive-next-hop enable
end
```

Example



To see the change in the routing table when the option is enabled:

1. Check the BGP routing table:

```
# get router info routing-table bgp
Routing table for VRF=0
B       10.100.1.4/30 [200/0] via 10.100.1.14 (recursive is directly connected, R560),
00:02:06
```

2. Enable BGP routes for recursive resolution of next hops:

```
config router bgp
    set recursive-next-hop enable
end
```

3. Check the BGP routing table again:

```
# get router info routing-table bgp
Routing table for VRF=0
B       10.100.1.4/30 [200/0] via 10.100.1.14 (recursive is directly connected, R560),
00:02:15
B       172.16.203.0/24 [200/0] via 10.100.1.6 (recursive via 10.100.1.14, R560), 00:00:06
```

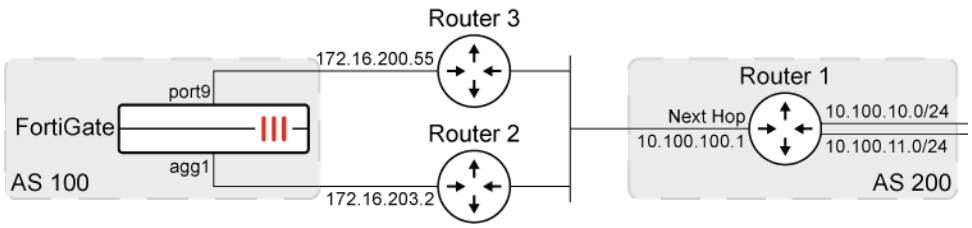
The second BGP route's next hop is now recursively resolved by another BGP route.

Next hop recursive resolution using ECMP routes

When there are multiple ECMP routes to a BGP next hop, all of them are considered for the next hop recursive resolution. This ensures that the outgoing traffic can be load balanced.

To support multipath, either EGBP or IGBP multipath must be enabled:

```
config router bgp
    set ebgp-multipath enable
    set ibgp-multipath enable
end
```



In this example, there are two static routes. The FortiGate has learned two BGP routes from Router 1 that have the same next hop at 10.100.100.1. The next hop is resolved by the two static routes.

To verify that the routes are added to the BGP routing table:

1. Check the two static routes:

```
# get router info routing-table static
Routing table for VRF=0
S       10.100.100.0/24 [10/0] via 172.16.200.55, port9
                        [10/0] via 172.16.203.2, agg1
```

2. Confirm that both routes are in the BGP routing table:

```
# get router info routing-table bgp
Routing table for VRF=0
B       10.100.10.0/24 [20/200] via 10.100.100.1 (recursive via 172.16.200.55, port9),
00:00:07
                                                 (recursive via 172.16.203.2, agg1), 00:00:07
B       10.100.11.0/24 [20/200] via 10.100.100.1 (recursive via 172.16.200.55, port9),
00:00:07
                                                 (recursive via 172.16.203.2, agg1), 00:00:07
```

Next hop recursive resolution using IPv6 prefix with on-link flag from route aggregation

When a FortiGate is acting as an IPv4 BGP neighbor and using stateful DHCPv6, it learns BGP routes with the IPv6 next hop belonging to an on-link prefix that is advertised through route aggregation (RA).

By default, the administrative distance for routes learned from the kernel is 255, and the routes do not interfere with the current route selection. To make the RA route usable by BGP, the distance must be set to less than 255 using the new `kernel-route-distance` command.

```
config router setting
    set kernel-route-distance <0-255>
end
```

If there are other user space routes with the same prefix, the best route is selected based on the distance.

To check the effect of changing the administrative distance:

1. Configure FGT_A:

```
config system interface
    edit "agg1"
        set vdom "root"
        set ip 172.16.203.1 255.255.255.0
        set allowaccess ping https http
        set type aggregate
        set member "port4"
        set alias "To_FGT_B_agg1"
        set lldp-transmission enable
        set snmp-index 40
        config ipv6
            set ip6-mode dhcp
            set ip6-allowaccess ping
        end
    next
end
```

2. Configure FGT_B (RA):

```
config system interface
    edit "agg2"
```

```
        set vdom "root"
        set ip 172.16.203.2 255.255.255.0
        set allowaccess ping https http
        set bfd disable
        set type aggregate
        set member "port4"
        set alias "To_FGT_A_agg1"
        set lldp-transmission enable
        set snmp-index 28
        config ipv6
            set ip6-address 2001:4::133/64
            set ip6-allowaccess ping
            set ip6-send-adv enable
            set ip6-manage-flag enable
            set ip6-other-flag enable
            set ip6-max-interval 10
            set ip6-min-interval 5
            config ip6-prefix-list
                edit 2001:4::/64
                next
            end
        end
    next
end
```

```
config system dhcp6 server
    edit 1
        set subnet 2001:4::/64
        set interface "agg2"
        config ip-range
            edit 1
                set start-ip 2001:4::11
                set end-ip 2001:4::20
            next
        end
    next
end
```

3. By default, the learned kernel route has a distance of 255 and does not interfere with the current route selection:

```
FGT_A (root)# get router info6 routing-table database
IPv6 Routing Table
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, B - BGP, V - BGP VPNv6
       > - selected route, * - FIB route, p - stale info
Timers: Uptime

Routing table for VRF=0
K    *  ::/0 via fe80::96f3:92ff:fe15:f7b, agg1, 00:00:20
```

```
C    *> ::1/128 via ::, root, 00:03:33
O       2000::1:1:1:1/128 [110/0] via ::, loopback1 inactive, 00:03:32, [1024/0]
C    *> 2000::1:1:1:1/128 via ::, loopback1, 00:03:33
O    *> 2000::2:2:2:2/128 [110/100] via fe80::96f3:92ff:fe15:f7b, agg1, 00:02:16, [1024/0]
O    *> 2000::3:3:3:3/128 [110/100] via fe80::6d5:90ff:fedb:e538, port1, 00:02:35, [1024/0]
O IA *> 2000::4:4:4:4/128 [110/1100] via fe80::96f3:92ff:fe15:f7b, agg1, 00:02:16, [1024/0]
C    *> 2000:10:100:1::/126 via ::, R150, 00:03:33
O       2000:10:100:1::4/126 [110/10000] via ::, R160, 00:03:32, [1024/0]
C    *> 2000:10:100:1::4/126 via ::, R160, 00:03:33
R    *> 2000:10:101:1::/64 [120/2] via fe80::3cd3:7cff:fed5:39, R150, 00:03:31, [1024/0]
S    *> 2000:10:101:2::/64 [10/0] via ::, Null, 00:03:33, [1024/0]
S    *> 2000:10:101:3::/64 [10/0] via ::, Null, 00:03:33, [1024/0]
S    *> 2000:10:102:1::/64 [10/0] via ::, Null, 00:03:33, [1024/0]
S    *> 2000:10:103:1::/64 [10/0] via ::, Null, 00:03:33, [1024/0]
S    *> 2000:10:104:1::/64 [10/0] via ::, Null, 00:03:33, [1024/0]
O    *> 2000:172:16::/48 [110/0] via ::, Null, 00:03:31, [1024/0]
O       2000:172:16:200::/64 [110/100] via ::, port1, 00:03:32, [1024/0]
C    *> 2000:172:16:200::/64 via ::, port1, 00:03:33
O E2 *> 2000:172:16:201::/64 [110/10] via fe80::96f3:92ff:fe15:f7b, agg1, 00:02:16, [1024/0]
O E2    2000:172:16:204::/64 [110/10] via fe80::96f3:92ff:fe15:f7b, agg1, 00:02:16, [1024/0]
S    *> 2000:172:16:204::/64 [10/0] via 2000:172:16:200::4, port1, 00:03:33, [1024/0]
      >                      [10/0] via 2000:172:16:203::2, agg1 inactive, 00:03:33, [1024/0]
     *>                      [10/0] via 2000:172:16:206::2, vlan100, 00:03:33, [1024/0]
C    *> 2000:172:16:206::/64 via ::, vlan100, 00:03:33
O       2000:172:16:207::/64 [110/10000] via ::, GRE_1, 00:03:32, [1024/0]
C    *> 2000:172:16:207::/64 via ::, GRE_1, 00:03:33
S    *> 2000:172:16:209::/64 [5/0] via to_FG_B_root tunnel ::172.16.206.2, 00:03:33, [1/0]
C    *> 2000:172:16:209::1/128 via ::, to_FG_B_root, 00:03:33
R    *> 2000:172:16:209::2/128 [120/2] via fe80::96f3:92ff:fe15:f7b, vlan100, 00:03:10,
[1024/0]
R    *> 2000:172:16:210::/64 [120/2] via fe80::96f3:92ff:fe15:f7b, vlan100, 00:03:10, [1024/0]
C    *> 2000:172:16:211::/64 via ::, sit_A_D, 00:03:33
B    *> 2000:172:27:1::/64 [200/0] via 2000::2:2:2:2 (recursive via fe80::96f3:92ff:fe15:f7b,
agg1), 00:01:25, [1024/0]
K    *  2001:4::/64 via ::, agg1, 00:00:20
O    *> 2001:4::/64 [110/100] via ::, agg1, 00:00:24, [1024/0]
```

```
FGT_A (root)# get router info6 routing-table kernel
No route available
```

4. Change the distance to 254:

```
FGT_A (root)# config router setting
    set kernel-route-distance 254
end
```

5. Now there is a kernel route:

```
FGT_A (root)# get router info6 routing-table kernel
Routing table for VRF=0
K*      ::/0 [254/1024] via fe80::96f3:92ff:fe15:f7b, agg1, 00:01:04
...
```

BGP conditional advertisement

BGP conditional advertisement allows the router to advertise a route only when certain conditions are met. Multiple conditions can be used together, with conditional route map entries treated as an AND operator, and IPv6 is supported.

Multiple conditions example

In this example, the FortiGate only advertises routes to its neighbor 2.2.2.2 if it learns multiple BGP routes defined in its conditional route map entry. All conditionals must be met.

To configure multiple conditions in BGP conditional advertisements:

1. Configure the IPv4 prefix list:

```
config router prefix-list
    edit "281"
        config rule
            edit 1
                set prefix 172.28.1.0 255.255.255.0
                unset ge
                unset le
            next
        end
    next
    edit "282"
        config rule
            edit 1
                set prefix 172.28.2.0 255.255.255.0
                unset ge
                unset le
            next
        end
    next
    edit "222"
        config rule
            edit 1
                set prefix 172.22.2.0 255.255.255.0
                unset ge
                unset le
            next
        end
    next
end
```

2. Configure the IPv4 route maps:

```
config router route-map
    edit "2814"
        config rule
            edit 1
                set match-ip-address "281"
            next
        end
    next
    edit "2224"
        config rule
            edit 1
                set match-ip-address "222"
            next
        end
    next
    edit "2824"
        config rule
            edit 1
                set match-ip-address "282"
            next
        end
    next
end
```

3. Configure the IPv6 prefix list:

```
config router prefix-list6
    edit "adv-2226"
        config rule
            edit 1
                set prefix6 2003:172:22:1::/64
                unset ge
                unset le
            next
        end
    next
    edit "list6-1"
        config rule
            edit 1
                set prefix6 2003:172:28:1::/64
                unset ge
                unset le
            next
        end
    next
    edit "list6-2"
        config rule
            edit 1
                set prefix6 2003:172:28:2::/64
                unset ge
                unset le
```

```
            next
        end
    next
end
```

4. Configure the IPv6 route maps:

```
config router route-map
    edit "map-2226"
        config rule
            edit 1
                set match-ip6-address "adv-2226"
            next
        end
    next
    edit "map-2816"
        config rule
            edit 1
                set match-ip6-address "list6-1"
            next
        end
    next
    edit "map-2826"
        config rule
            edit 1
                set match-ip6-address "list6-2"
            next
        end
    next
end
```

5. Configure the BGP settings:

```
config router bgp
    config neighbor
        edit "2.2.2.2"
            config conditional-advertise
                edit "2224"
                    set condition-routemap "2814" "2824"
                    set condition-type non-exist
                next
            end
        next
        edit "2003::2:2:2:2"
            config conditional-advertise6
                edit "map-2226"
                    set condition-routemap "map-2816" "map-2826"
                next
            end
            set route-reflector-client6 enable
        next
```

```
    end
end
```

To verify the IPv4 conditional advertisements:

```
# get router info bgp neighbors 2.2.2.2
...
  Conditional advertise-map:
        Adv-map 2224root 2814root, cond-state 0-1
                         2824root, cond-state 0-1
...
```

In this output, the condition is that the routes in route maps 2814 and 2824 do not exist. However, routes for 2814 and 2224 exist, so the conditions are not met.

To verify the IPv6 conditional advertisements:

```
# get router info6 bgp neighbors 2003::2:2:2:2
...
  Conditional advertise-map:
        Adv-map map-2226root map-2816root, cond-state 1-1
                             map-2826root, cond-state 1-0
...
```

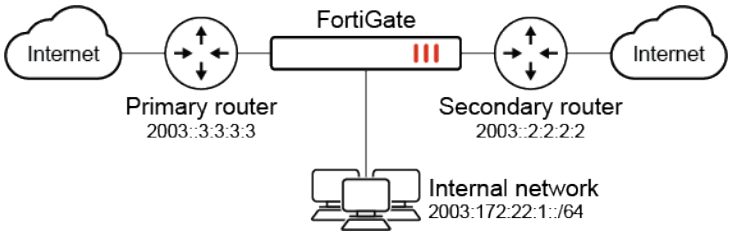
In this output, the condition is that the routes in route maps map-2816 and map-2826 exist. However, routes for map-2816 exist, but map-2826 does not, so the conditions are not met.

To view the conditional route maps:

```
# diagnose ip router command show-vrf root show running router bgp
...
 neighbor 2.2.2.2 advertise-map 2224root exist-map 2814root
 neighbor 2.2.2.2 advertise-map 2224root exist-map 2824root
... ...
 !
 address-family ipv6
 neighbor 2003::2:2:2:2 advertise-map map-2226root non-exist-map map-2816root
 neighbor 2003::2:2:2:2 advertise-map map-2226root non-exist-map map-2826root
!
```

IPv6 example 1

In this example, the FortiGate advertises its local network to the secondary router when the primary router is down. The FortiGate detects the primary router is down in the absence of a learned route.



- When the FortiGate learns route 2003:172:28:1::/64 from the primary router, it does not advertise its local route (2003:172:22:1::/64) to the secondary router.
- When the FortiGate does not learn route 2003:17:28:1::/64 from the primary router, advertises its local route (2003:172:22:1::/64) to the secondary router.
- The BGP conditional advertisement condition is set to be true if the condition route map (2003:172:28:1::/64) is not matched (`non-exist`).

To configure BGP conditional advertisement with IPv6:

1. Configure the IPv6 prefix lists:

```
config router prefix-list6
    edit "adv-222"
        config rule
            edit 1
                set prefix6 2003:172:22:1::/64
                unset ge
                unset le
            next
        end
    next
    edit "lrn-281"
        config rule
            edit 1
                set prefix6 2003:172:28:1::/64
                unset ge
                unset le
            next
        end
    next
end
```

2. Configure the route maps:

```
config router route-map
    edit "map-221"
        config rule
            edit 1
                set match-ip6-address "adv-222"
            next
        end
    next
    edit "map-281"
        config rule
```

```
            edit 1
                set match-ip6-address "lrn-281"
            next
        end
    next
end
```

3. Configure BGP:

```
config router bgp
    set as 65412
    set router-id 1.1.1.1
    set ibgp-multipath enable
    set network-import-check disable
    set graceful-restart enable
    config neighbor
        edit "2003::2:2:2:2"
            set soft-reconfiguration6 enable
            set remote-as 65412
            set update-source "loopback1"
            config conditional-advertise6
                edit "map-221"
                    set condition-routemap "map-281"
                    set condition-type non-exist
                next
            end
        next
        edit "2003::3:3:3:3"
            set soft-reconfiguration6 enable
            set remote-as 65412
            set update-source "loopback1"
        next
    end
end
```

In this configuration, if route map `map-281` does not exist, then the FortiGate advertises route map `map-221` to neighbor 2003::2:2:2:2.

4. Verify the routing table:

```
# get router info6 routing-table bgp
B       2003:172:28:1::/64 [200/0] via 2003::3:3:3:3 (recursive via ****::***:***:****:****,
port9), 01:23:45
B       2003:172:28:2::/64 [200/0] via 2003::3:3:3:3 (recursive via ****::***:***:****:****,
port9), 23:09:22
```

When the FortiGate learns 2003:172:28:1::/64, it will not advertise its local route 2003:172:22:1::/64 to neighbor 2003::2:2:2:2. If the FortiGate has not learned 2003:172:28:1::/64, it will advertise its local route 2003:172:22:1::/64 to neighbor 2003::2:2:2:2.

IPv6 example 2

With the same IPv6 prefix lists and route maps, when the FortiGate does learn 2003:172:28:1::/64, it advertises local route 2003:172:22:1::/64 to the secondary router. The BGP conditional advertisement condition is set to be true if the condition route map is matched (`exist`).

To configure BGP conditional advertisement with IPv6:

1. Configure BGP:

```
config router bgp
    config neighbor
        edit "2003::2:2:2:2"
            config conditional-advertise6
                edit "map-221"
                    set condition-routemap "map-281"
                    set condition-type exist
                next
            end
        next
    end
end
```

2. Verify the routing table:

```
# get router info6 routing-table bgp
B       2003:172:28:1::/64 [200/0] via 2003::3:3:3:3 (recursive via ****::***:***:****:****,
port9), 01:23:45
B       2003:172:28:2::/64 [200/0] via 2003::3:3:3:3 (recursive via ****::***:***:****:****,
port9), 23:09:22
```

When the FortiGate learns 2003:172:28:1::/64, it will advertise its local route 2003:172:22:1::/64 to neighbor 2003::2:2:2:2. If the FortiGate has not learned route 2003:172:28:1::/64, it will not advertise its local route 2003:172:22:1::/64 to neighbor 2003::2:2:2:2.

BGP conditional advertisements for IPv6 prefix when IPv4 prefix conditions are met and vice-versa

The FortiGate supports conditional advertisement of IPv4 and IPv6 route maps with `edit <advertise-routemap>` under `config conditional-advertise`, and supports configuring IPv4 and IPv6 route maps as conditions with the `condition-routemap` setting.

The FortiGate can cross-check conditions involving IPv4 and IPv6 route maps and perform conditional advertisements accordingly when those conditions are met. The global option, `cross-family-conditional-adv` in the BGP configuration settings allows this cross-checking to occur.

```
config router bgp
    set cross-family-conditional-adv {enable | disable}
    config conditional-advertise
        edit <advertise-routemap>
            set advertise-routemap <string>
            set condition-routemap <name1>, <name2>, ...
```

```
            set condition-type {exist | non-exist}
        next
    end
end
```

By default, the `cross-family-conditional-adv` setting is disabled. When disabled, the FortiGate will only check conditional route maps against the routing information base (RIB) of the IP address family (IPv4 or IPv6) that corresponds to the IP address family of the route map to be advertised conditionally.

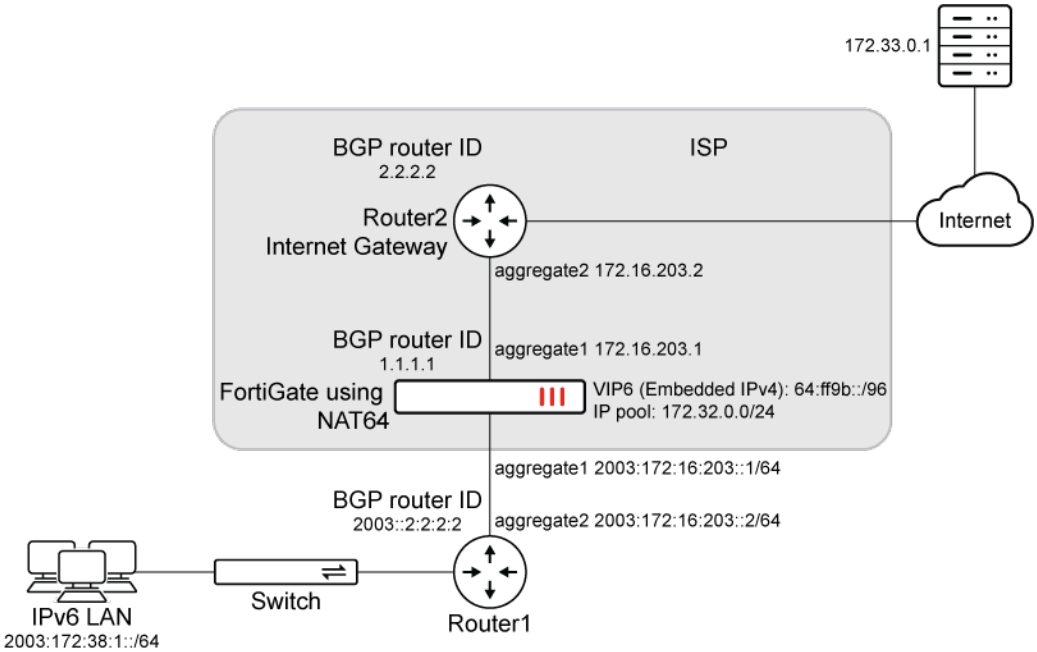
For example, for an IPv6 conditional advertisement, if IPv4 conditional route maps have been configured, then the FortiGate will not meet any of these conditions because IPv4 routes will not exist in the IPv6 RIB. The same behavior applies for an IPv4 conditional advertisement, namely, that the FortiGate will not meet any configured IPv6 conditions since these routes will not exist in the IPv4 RIB. If routes do not match a conditional route map, then the condition is considered non-existent.

IPv4 and IPv6 BGP conditional advertisements using advertising and conditional route maps of the same IP address family are already supported in previous versions of FortiOS.

NAT64 example

In this example, the FortiGate uses NAT64 where the LAN via Router1 uses IPv6 and where Router2 is the internet gateway using IPv4.

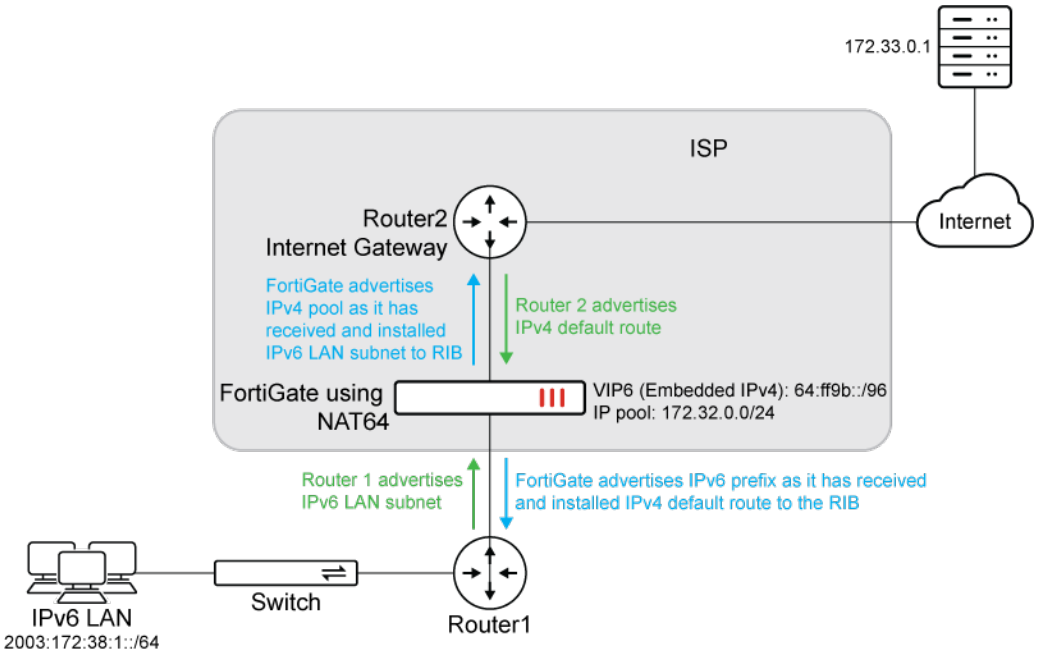
> This example assumes a pure NAT64 design with the following expectations:
> - IPv4 IP pools would be announced to Internet gateway when IPv6 client subnets learned from IPv6 route.
> - IPv6 virtual IP addresses (VIPs) (default = 64:ff9b::/96) would be announced to Router 1 when IPv4 default route learned from Internet gateway.



The administrator of the FortiGate has the following requirements, which are implemented using IPv4 and IPv6 conditional advertisements:

- The FortiGate needs to announce IPv4 pools for NAT translation towards the internet gateway only if the IPv6 prefix exists in the routing table.
- The FortiGate needs to advertise the IPv6 address towards the LAN only if the IPv4 default route exists on the FortiGate.

The below diagram details the flow of routing advertisements:



The prefixes defined in IPv4 route map 2814 and IPv6 route map `map-281` both exist, so the FortiGate advertises the route map prefix in route-map `2224` (172.22.2.0/255.255.255.0) to its BGP neighbor 2.2.2.2.

For IPv6 neighbor 2003::2:2:2:2, the prefixes defined in IPv4 route map 2874 and IPv6 route map `map-38` both do not exist, and the `condition-type` is set to `non-exist`, so the FortiGate advertises the route map prefix in route map `map-222` (2003:172:22:1::/64) to its BGP neighbor 2003::2:2:2:2.

When the global `cross-family-conditional-adv` enabled, this is the only time the FortiGate will cross-check the address family; otherwise, it only checks the corresponding conditional map and treats the cross-family addresses as non-existent.

To configure the conditional advertisement to BGP neighbor 2.2.2.2 and its conditional route maps:

1. Configure the IPv4 prefix lists:

```
config router prefix-list
    edit "281"
        config rule
            edit 1
                set prefix 172.28.1.0 255.255.255.0
                unset ge
                unset le
            next
```

```
        end
    next
    edit "222"
        config rule
            edit 1
                set prefix 172.22.2.0 255.255.255.0
                unset ge
                unset le
            next
        end
    next
end
```

2. Configure the IPv6 prefix list:

```
config router prefix-list6
    edit "list6-1"
        config rule
            edit 1
                set prefix6 2003:172:28:1::/64
                unset ge
                unset le
            next
        end
    next
end
```

3. Configure the route maps:

```
config router route-map
    edit "2814"
        config rule
            edit 1
                set match-ip-address "281"
            next
        end
    next
    edit "map-281"
        config rule
            edit 1
                set match-ip6-address "list6-1"
            next
        end
    next
    edit "2224"
        config rule
            edit 1
                set match-ip-address "222"
            next
        end
    next
end
```

To configure the conditional advertisement to BGP neighbor 2003::2.2.2.2 and its conditional route maps:

1. Configure the IPv4 prefix list:

```
config router prefix-list
    edit "287"
        config rule
            edit 1
                set prefix 172.28.7.0 255.255.255.0
                unset ge
                unset le
            next
        end
    next
end
```

2. Configure the IPv6 prefix lists:

```
config router prefix-list6
    edit "list6-38"
        config rule
            edit 1
                set prefix6 2003:172:38:1::/64
                unset ge
                unset le
            next
        end
    next
    edit "adv-222"
        config rule
            edit 1
                set prefix6 2003:172:22:1::/64
                unset ge
                unset le
            next
        end
    next
end
```

3. Configure the route maps:

```
config router route-map
    edit "2874"
        config rule
            edit 1
                set match-ip-address “287”
            next
        end
    next
    edit "map-38"
        config rule
            edit 1
```

```
                set match-ip6-address "list6-38"
            next
        end
    next
    edit "map-222"
        config rule
            edit 1
                set match-ip6-address "adv-222"
            next
        end
    next
end
```

To configure the BGP settings with address family cross-checking:

```
config router bgp
    set as 65412
    set router-id 1.1.1.1
    set ibgp-multipath enable
    set network-import-check disable
    set cluster-id 1.1.1.1
    set graceful-restart enable
    set cross-family-conditional-adv enable
    config neighbor
        edit "3.3.3.3"
            set activate6 disable
            set capability-graceful-restart enable
            set soft-reconfiguration enable
            set prefix-list-out "local-out"
            set remote-as 65412
            set route-map-out "as-prepend"
            set keep-alive-timer 30
            set holdtime-timer 90
            set update-source "loopback1"
            set route-reflector-client enable
        next
        edit "2.2.2.2"
            set advertisement-interval 5
            set activate6 disable
            set capability-graceful-restart enable
            set soft-reconfiguration enable
            set remote-as 65412
            set keep-alive-timer 34
            set holdtime-timer 90
            set update-source "loopback1"
            config conditional-advertise
                edit "2224"
                    set condition-routemap "2814" "map-281"
                next
            end
            set route-reflector-client enable
        next
```

```
        edit "2003::2:2:2:2"
            set advertisement-interval 5
            set activate disable
            set capability-graceful-restart6 enable
            set soft-reconfiguration enable
            set soft-reconfiguration6 enable
            set remote-as 65412
            set keep-alive-timer 30
            set holdtime-timer 90
            set update-source "loopback1"
            config conditional-advertise6
                edit "map-222"
                    set condition-routemap "map-38" "2874"
                    set condition-type non-exist
                next
            end
            set route-reflector-client6 enable
        next
        edit "2003::3:3:3:3"
            set advertisement-interval 5
            set activate disable
            set capability-graceful-restart6 enable
            set soft-reconfiguration6 enable
            set remote-as 65412
            set route-map-in6 "community-del777"
            set keep-alive-timer 30
            set holdtime-timer 90
            set update-source "loopback1"
        next
    end
    config network
        edit 1
            set prefix 172.27.1.0 255.255.255.0
        next
        edit 2
            set prefix 172.27.2.0 255.255.255.0
        next
        edit 3
            set prefix 172.22.2.0 255.255.255.0
        next
    end
    config network6
        edit 1
            set prefix6 2003:172:22:1::/64
        next
    end
end
```

To verify the BGP status and the BGP routing table for IPv4:

```
# get router info bgp summary
VRF 0 BGP router identifier 1.1.1.1, local AS number 65412
```

```
BGP table version is 2
6 BGP AS-PATH entries
2 BGP community entries

Neighbor   V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2.2.2.2    4      65412     100     148        2    0    0 00:42:22        3
3.3.3.3    4      65412      99      99        2    0    0 00:42:05        6
6.6.6.6    4         20       0       0        0    0    0    never Idle (Admin)
10.100.1.1 4         20     100     107        2    0    0 00:43:43        2
10.100.1.5 4         20      53      57        2    0    0 00:43:42        0

Total number of neighbors 5

Condition route map:
  2814, state 1, use 3
  map-281, state 1, use 3
```

To verify the BGP status and the BGP routing table for IPv6:

```
# get router info6 bgp summary
VRF 0 BGP router identifier 1.1.1.1, local AS number 65412
BGP table version is 3
6 BGP AS-PATH entries
2 BGP community entries

Neighbor      V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
6.6.6.6       4         20       0       0        0    0    0    never Idle (Admin)
10.100.1.1    4         20     100     108        3    0    0 00:43:51        0
10.100.1.5    4         20      53      57        3    0    0 00:43:50        0
2003::2:2:2:2 4      65412      98     118        3    0    0 00:42:25        1
2003::3:3:3:3 4      65412     102     100        2    0    0 00:42:20        3

Total number of neighbors 5

Condition route map:
  map-38, state 0, use 3
  2874, state 0, use 3
```

To verify the BGP routing table for IPv4 and confirm the conditional advertisement occurred:

```
# get router info routing-table bgp
Routing table for VRF=0
B       172.22.2.0/24 [200/0] via 1.1.1.1 (recursive via 172.16.203.1, agg2), 00:00:03, [1/0]
B       172.27.1.0/24 [200/0] via 1.1.1.1 (recursive via 172.16.203.1, agg2), 00:37:30, [1/0]
B       172.27.2.0/24 [200/0] via 1.1.1.1 (recursive via 172.16.203.1, agg2), 00:37:30, [1/0]
B       172.27.5.0/24 [200/0] via 1.1.1.1 (recursive via 172.16.203.1, agg2), 00:37:30, [1/0]
B       172.27.6.0/24 [200/0] via 1.1.1.1 (recursive via 172.16.203.1, agg2), 00:37:30, [1/0]
B       172.27.7.0/24 [200/0] via 1.1.1.1 (recursive via 172.16.203.1, agg2), 00:37:30, [1/0]
B       172.27.8.0/24 [200/0] via 1.1.1.1 (recursive via 172.16.203.1, agg2), 00:37:30, [1/0]
B       172.29.1.0/24 [200/0] via 1.1.1.1 (recursive via 172.16.203.1, agg2), 00:37:30, [1/0]
B       172.29.2.0/24 [200/0] via 1.1.1.1 (recursive via 172.16.203.1, agg2), 00:37:30, [1/0]
```

To verify the BGP routing table for IPv6 and confirm the conditional advertisement occurred:

```
# get router info6 routing-table bgp
Routing table for VRF=0
B       2003:172:22:1::/64 [200/0] via 2003::1:1:1:1 (recursive via 2003:172:16:203::1, agg2),
00:00:01, [1024/0]
B       2003:172:28:1::/64 [200/0] via 2003::3:3:3:3 (recursive via fe80::a5b:eff:feeb:ca45,
port1), 00:37:59, [1024/0]
B       2003:172:28:2::/64 [200/0] via 2003::3:3:3:3 (recursive via fe80::a5b:eff:feeb:ca45,
port1), 00:37:59, [1024/0]
```

Behavior when address family cross-checking is disabled

Using a similar BGP configuration with `cross-family-conditional-adv` disabled, note the following behavior based on the condition type.

When the condition type is set to exist:

```
config router bgp
       set cross-family-conditional-adv disable
       config neighbor
        edit "2.2.2.2"
            config conditional-advertise
                edit "222v4"
                    set condition-routemap "4-281" "6-281"
                    set condition-type exist
                next
            end
        next
    end
end
```

The FortiGate will only check the IPv4 RIB table to see if there is a matching IP address for each route map. Any IPv6 address under the route map will not get checked in the corresponding IPv6 RIB table, and the condition result will be non-existent. The `222v4` route map will not advertise to its neighbor because the result is non-existent, while the condition type is existent.

When the condition type is set to non-exist:

```
config router bgp
       set cross-family-conditional-adv disable
       config neighbor
        edit "2003::2:2:2:2"
            config conditional-advertise6
                edit "v6-222"
                    set condition-routemap "v6-238" "v4-287"
                    set condition-type non-exist
                next
            end
        next
```

```
    end
end
```

If the `v6-238` IPv6 prefix does not exist in the IPv6 RIB table, then the FortiGate will only check `v4-287` in the IPv6 RIB table. The FortiGate will not find it because it is an IPv4 address. Since the condition type is also `non-exist`, route `v6-222` will be advertised to its neighbor.

BGP error handling per RFC 7606

The FortiGate uses one of the three approaches to handle malformed attributes in BGP UPDATE messages, in order of decreasing severity:

1. Notification and Session reset
2. Treat-as-withdraw
3. Attribute discard

When a BGP UPDATE message contains multiple malformed attributes, the most severe approach that is triggered by one of the attributes is followed. See RFC 7606 for more information.

The following table lists the BGP attributes, and how FortiGate handles a malformed attribute in the UPDATE message:

BGP attribute	Handling
origin	Handled by the treat-as-withdraw approach.
AS path	Handled by the treat-as-withdraw approach.
AS 4 path	Handled by the attribute discard approach.
aggregator	Handled by the attribute discard approach.
aggregator 4	Handled by the attribute discard approach.
next-hop	Handled by the treat-as-withdraw approach.
multiple exit discriminator	Handled by the treat-as-withdraw approach.
local preference	Handled by the treat-as-withdraw approach.
atomic aggregate	Handled by the attribute discard approach.
community	Handled by the treat-as-withdraw approach.
extended community	Handled by the treat-as-withdraw approach.
originator	Handled by the treat-as-withdraw approach.
cluster	Handled by the treat-as-withdraw approach.
PMSI	Handled by the treat-as-withdraw approach.
MP reach	Handled by the notification message approach.
MP unreach	Handled by the notification message approach.
attribute set	Handled by the treat-as-withdraw approach.

BGP attribute	Handling
AIGP	Handled by the treat-as-withdraw approach.
Unknown	If the BGP flag does not indicate that this is an optional attribute, this malformed attribute is handled by the notification message approach.

BGP next hop tag-match mode

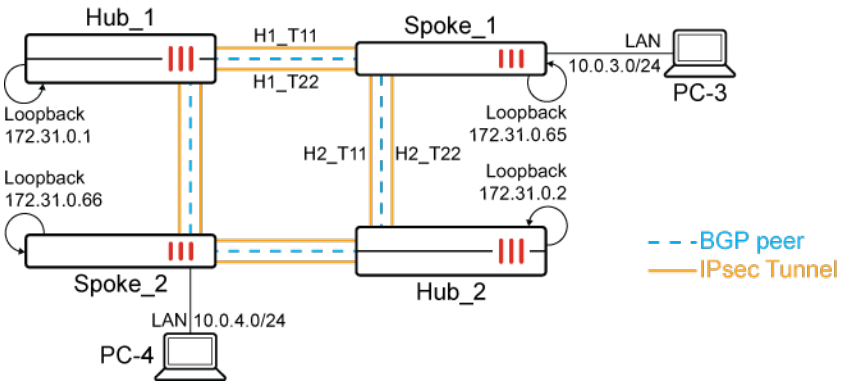
Tag-match mode can be configured to increase flexibility when controlling how BGP routes' next hops are resolved:

```
config router bgp
    set tag-resolve-mode {disable | preferred | merge}
end
```

Option	Description
Best-match (`disable`)	Resolve the BGP route's next hops with best-matched routes. This is the default setting.
Tag-match (`preferred`)	Resolve the BGP route's next hops with routes that have the same tag. If there are no results, resolve the next hops with best-matched routes.
Tag-and-best-match (`merge`)	Merge tag-match with best-match if they are using different routes, then let shortcuts hide their parents. The results exclude the next hops of tag-match whose interfaces have appeared in best-match.

In these examples:

- Each spoke has two IPsec tunnels to each hub, and one BGP peer on loopback interface to each hub (route-reflector).
- The loopbacks are exchanged with IKE between the spokes and hubs. They are installed as static routes that are used to provide reachability for establishing BGP neighbors.
- The summary BGP routes from the loopback IP address ranges that originated on the hubs are advertised to the spokes for resolving the BGP next hop s on the spokes.
- The spokes' PC LAN subnets are reflected by the hubs.
- Spoke_1 receives BGP routes (the LAN subnet and loopback IP summary) from Hub_1 with tag 1 and from Hub_2 with tag 2.
- SD-WAN is enabled on Spoke_1, and all of the tunnels are SD-WAN members.



Example 1: Connection between Hub and Spoke down

If the connections between Hub_1 and Spoke_2 are down, traffic from PC_3 to PC_4 can still go through Hub_1 because of the best-match resolving on Spoke_1, but packets will be dropped on Hub_1. When tag-match is enabled on Spoke_1, the spoke will resolve the PC_4 LAN route to Hub2, and traffic will be forwarded to Hub_2 and reach its destination.

To test the tag-match mode:

1. View the key routes on Spoke_1:

```
Spoke_1(root) # get router info routing-table all
C       10.0.3.0/24 is directly connected, port4
B       10.0.4.0/24 [200/0] via 172.31.0.66 [2] (recursive via H1_T11 tunnel 172.31.1.1),
20:09:52
              (recursive via H1_T22 tunnel 10.0.0.2), 20:09:52
              (recursive via H2_T11 tunnel 172.31.1.101), 20:09:52
              (recursive via H2_T22 tunnel 10.0.0.4), 20:09:52
B       172.31.0.0/25 [200/0] via 172.31.0.1 (recursive via H1_T11 tunnel 172.31.1.1),
23:25:37
              (recursive via H1_T22 tunnel 10.0.0.2), 23:25:37
            [200/0] via 172.31.0.2 (recursive via H2_T11 tunnel 172.31.1.101), 23:25:37
              (recursive via H2_T22 tunnel 10.0.0.4), 23:25:37
S       172.31.0.1/32 [15/0] via H1_T11 tunnel 172.31.1.1, [1/0]
            [15/0] via H1_T22 tunnel 10.0.0.2, [1/0]
S       172.31.0.2/32 [15/0] via H2_T11 tunnel 172.31.1.101, [1/0]
            [15/0] via H2_T22 tunnel 10.0.0.4, [1/0]
C       172.31.0.65/32 is directly connected, Loopback0
...
```

   172.31.0.0/25 is the loopback IP summary originated by both Hub_1 and Hub_2. The next hop of the PC_4 LAN route is resolved to Hub_1 (H1_T11, H1_T22) and Hub_2 (H2_T11, H2_T22) based on the loopback IP summary route.
2. When connections between Spoke_2 and Hub_1 fails due to the BGP neighbor, tunnels, or physical ports going down, the PC_4 LAN route can be still resolved to Hub_1 and Hub_2 because the loopback IP summary can still be received from both Hub_1 and Hub_2:

```
Spoke_1(root) # get router info routing-table all
C       10.0.3.0/24 is directly connected, port4
B       10.0.4.0/24 [200/0] via 172.31.0.66 (recursive via H1_T11 tunnel 172.31.1.1), 00:03:06
              (recursive via H1_T22 tunnel 10.0.0.2), 00:03:06
              (recursive via H2_T11 tunnel 172.31.1.101), 00:03:06
              (recursive via H2_T22 tunnel 10.0.0.4), 00:03:06
B       172.31.0.0/25 [200/0] via 172.31.0.1 (recursive via H1_T11 tunnel 172.31.1.1),
23:55:34
              (recursive via H1_T22 tunnel 10.0.0.2), 23:55:34
            [200/0] via 172.31.0.2 (recursive via H2_T11 tunnel 172.31.1.101), 23:55:34
              (recursive via H2_T22 tunnel 10.0.0.4), 23:55:34
...
```

3. If traffic sent from PC_3 to PC_4 goes through Hub_1, packets are dropped because there is no PC_4 LAN route on Hub_1:

```
Spoke_1 (root) # diagnose sniffer packet any 'host 10.0.4.2' 4
interfaces=[any]
filters=[host 10.0.4.2]
11.261264 port4 in 10.0.3.2 -> 10.0.4.2: icmp: echo request
11.261349 H1_T11 out 10.0.3.2 -> 10.0.4.2: icmp: echo request
12.260268 port4 in 10.0.3.2 -> 10.0.4.2: icmp: echo request
12.260291 H1_T11 out 10.0.3.2 -> 10.0.4.2: icmp: echo request

Hub_1 (root) #  diagnose sniffer packet any 'host 10.0.4.2' 4
interfaces=[any]
filters=[host 10.0.4.2]
6.966064 EDGE_T1 in 10.0.3.2 -> 10.0.4.2: icmp: echo request
7.965012 EDGE_T1 in 10.0.3.2 -> 10.0.4.2: icmp: echo request
```

4. If the tag-match mode is set to tag-match (`preferred`) on Spoke_1, then the PC_4 LAN route can only be resolved to Hub_2 because of tag-match checking:

```
Spoke_1(root) # get router info routing-table all
C       10.0.3.0/24 is directly connected, port4
B       10.0.4.0/24 [200/0] via 172.31.0.66 tag 2 (recursive via H2_T11 tunnel 172.31.1.101),
00:02:35
              (recursive via H2_T22 tunnel 10.0.0.4), 00:02:35
B       172.31.0.0/25 [200/0] via 172.31.0.1 tag 1 (recursive via H1_T11 tunnel 172.31.1.1),
03:18:41
              (recursive via H1_T22 tunnel 10.0.0.2), 03:18:41
            [200/0] via 172.31.0.2 tag 2 (recursive via H2_T11 tunnel 172.31.1.101), 03:18:41
              (recursive via H2_T22 tunnel 10.0.0.4), 03:18:41
...
```

```
Spoke_1 (root) # get router info routing-table details 10.0.4.0/24

Routing table for VRF=0
Routing entry for 10.0.4.0/24
  Known via "bgp", distance 200, metric 0, best
  Last update 00:01:11 ago
```

```
 * 172.31.0.66, tag 2 (recursive via H2_T11 tunnel 172.31.1.101), tag-match
             (recursive via H2_T22 tunnel 10.0.0.4), tag-match
```

5. If traffic is again sent from PC_3 to PC_4, it will go through Hub_2 and reach the destination:

```
Spoke_1 (root) # diagnose sniffer packet any 'host 10.0.4.2' 4
interfaces=[any]
filters=[host 10.0.4.2]
7.216948 port4 in 10.0.3.2 -> 10.0.4.2: icmp: echo request
7.217035 H2_T11 out 10.0.3.2 -> 10.0.4.2: icmp: echo request
7.217682 H2_T11 in 10.0.4.2 -> 10.0.3.2: icmp: echo reply
7.217729 port4 out 10.0.4.2 -> 10.0.3.2: icmp: echo reply
```

Example 2: SD-WAN failover when shortcut down

After the shortcut from Spoke_1 to Spoke_2 is established, Spoke_1 will only resolve the PC_4 LAN route to the shortcut, because of best-match resolving, prohibiting SD-WAN failover. When tag-and-best-match is enabled on Spoke_1, the spoke can resolve the PC_4 LAN route to the shortcut and to other alternative tunnels, allowing SD-WAN failover.

To test the tag-and-best-match mode:

1. Unset `tag-resolve-mode` and resume the connections between Spoke_2 and Hub_1. The routing table on Spoke_1 changes to the initial state:

```
Spoke_1(root) # get router info routing-table all
C       10.0.3.0/24 is directly connected, port4
B       10.0.4.0/24 [200/0] via 172.31.0.66 [2] (recursive via H1_T11 tunnel 172.31.1.1),
00:01:54
              (recursive via H1_T22 tunnel 10.0.0.2), 00:01:54
              (recursive via H2_T11 tunnel 172.31.1.101), 00:01:54
              (recursive via H2_T22 tunnel 10.0.0.4), 00:01:54
B       172.31.0.0/25 [200/0] via 172.31.0.1 (recursive via H1_T11 tunnel 172.31.1.1),
03:30:35
              (recursive via H1_T22 tunnel 10.0.0.2), 03:30:35
            [200/0] via 172.31.0.2 (recursive via H2_T11 tunnel 172.31.1.101), 03:30:35
              (recursive via H2_T22 tunnel 10.0.0.4), 03:30:35
S       172.31.0.1/32 [15/0] via H1_T11 tunnel 172.31.1.1, [1/0]
            [15/0] via H1_T22 tunnel 10.0.0.2, [1/0]
S       172.31.0.2/32 [15/0] via H2_T11 tunnel 172.31.1.101, [1/0]
            [15/0] via H2_T22 tunnel 10.0.0.4, [1/0]
C       172.31.0.65/32 is directly connected, Loopback0

...
```

2. Send traffic from PC_3 to PC_4.

   The shortcut from Spoke_1 o Spoke_2 is established.

   The PC_4 LAN route is only resolved to the shortcut because of best-match resolving. If the shortcut is out of SLA, then the traffic cannot switch over to another, alternative tunnel.

```
Spoke_1 (root) # get router info routing-table all
C       10.0.3.0/24 is directly connected, port4
B       10.0.4.0/24 [200/0] via 172.31.0.66 [2] (recursive via H1_T11_0 tunnel 10.0.0.40),
00:09:22
B       172.31.0.0/25 [200/0] via 172.31.0.1 (recursive via H1_T11 tunnel 172.31.1.1),
03:40:12
              (recursive via H1_T22 tunnel 10.0.0.2), 03:40:12
            [200/0] via 172.31.0.2 (recursive via H2_T11 tunnel 172.31.1.101), 03:40:12
              (recursive via H2_T22 tunnel 10.0.0.4), 03:40:12
S       172.31.0.1/32 [15/0] via H1_T11 tunnel 172.31.1.1, [1/0]
            [15/0] via H1_T22 tunnel 10.0.0.2, [1/0]
S       172.31.0.2/32 [15/0] via H2_T11 tunnel 172.31.1.101, [1/0]
            [15/0] via H2_T22 tunnel 10.0.0.4, [1/0]
C       172.31.0.65/32 is directly connected, Loopback0
S       172.31.0.66/32 [15/0] via H1_T11_0 tunnel 10.0.0.40, [1/0]
...
```

3. If the tag-match mode is set to tag-and-best-match (`merge`) on Spoke_1, then the PC_4 LAN route is resolved to the `H1_T11_0` shortcut based on best-match resolving, and to `H1_T11, H1_T22, H2_T11, H2_T22` based on tag-match resolving. It is then resolved to `H1_T11, H1_T22, H2_T11, H2_T22` after letting the shortcut hide its parent tunnel.

```
Spoke_1 (root) # get router info routing-table all
C       10.0.3.0/24 is directly connected, port4
B       10.0.4.0/24 [200/0] via 172.31.0.66 tag 1 (recursive via H1_T11_0 tunnel 10.0.0.40),
00:07:36
              (recursive via H1_T22 tunnel 10.0.0.2), 00:07:36
            [200/0] via 172.31.0.66 tag 2 (recursive via H1_T11_0 tunnel 10.0.0.40), 00:07:36
              (recursive via H2_T11 tunnel 172.31.1.101), 00:07:36
              (recursive via H2_T22 tunnel 10.0.0.4), 00:07:36
B       172.31.0.0/25 [200/0] via 172.31.0.1 tag 1 (recursive via H1_T11 tunnel 172.31.1.1),
03:48:26
              (recursive via H1_T22 tunnel 10.0.0.2), 03:48:26
            [200/0] via 172.31.0.2 tag 2 (recursive via H2_T11 tunnel 172.31.1.101), 03:48:26
              (recursive via H2_T22 tunnel 10.0.0.4), 03:48:26
S       172.31.0.1/32 [15/0] via H1_T11 tunnel 172.31.1.1, [1/0]
            [15/0] via H1_T22 tunnel 10.0.0.2, [1/0]
S       172.31.0.2/32 [15/0] via H2_T11 tunnel 172.31.1.101, [1/0]
            [15/0] via H2_T22 tunnel 10.0.0.4, [1/0]
C       172.31.0.65/32 is directly connected, Loopback0
S       172.31.0.66/32 [15/0] via H1_T11_0 tunnel 10.0.0.40, [1/0]

...
```

```
Spoke_1 (root) # get router info routing-table details 10.0.4.0/24

Routing table for VRF=0
Routing entry for 10.0.4.0/24
  Known via "bgp", distance 200, metric 0, best
  Last update 00:01:02 ago
  * 172.31.0.66, tag 1 (recursive via H1_T11_0 tunnel 10.0.0.42), best-match
              (recursive via H1_T22 tunnel 10.0.0.2), tag-match
```

```
  * 172.31.0.66, tag 2 (recursive via H1_T11_0 tunnel 10.0.0.42), best-match
              (recursive via H2_T11 tunnel 172.31.1.101), tag-match
              (recursive via H2_T22 tunnel 10.0.0.4), tag-match
```

4. If the H1_T11_0 shortcut goes out of SLA, traffic will switch to tunnel H1_T22 and shortcut H1_T22_0 is triggered. The PC_4 LAN route is resolved to H1_T11, H1_T22, H2_T11, H2_T22.

```
Spoke_1 (root) # get router info routing-table all
C       10.0.3.0/24 is directly connected, port4
B       10.0.4.0/24 [200/0] via 172.31.0.66 tag 1 (recursive via H1_T11_0 tunnel 10.0.0.40),
00:18:50
              (recursive via H1_T22_0 tunnel 10.0.0.41), 00:18:50
            [200/0] via 172.31.0.66 tag 2 (recursive via H1_T11_0 tunnel 10.0.0.40), 00:18:50
              (recursive via H1_T22_0 tunnel 10.0.0.41), 00:18:50
              (recursive via H2_T11 tunnel 172.31.1.101), 00:18:50
              (recursive via H2_T22 tunnel 10.0.0.4), 00:18:50
B       172.31.0.0/25 [200/0] via 172.31.0.1 tag 1 (recursive via H1_T11 tunnel 172.31.1.1),
03:59:40
              (recursive via H1_T22 tunnel 10.0.0.2), 03:59:40
            [200/0] via 172.31.0.2 tag 2 (recursive via H2_T11 tunnel 172.31.1.101), 03:59:40
              (recursive via H2_T22 tunnel 10.0.0.4), 03:59:40
S       172.31.0.1/32 [15/0] via H1_T11 tunnel 172.31.1.1, [1/0]
            [15/0] via H1_T22 tunnel 10.0.0.2, [1/0]
S       172.31.0.2/32 [15/0] via H2_T11 tunnel 172.31.1.101, [1/0]
            [15/0] via H2_T22 tunnel 10.0.0.4, [1/0]
C       172.31.0.65/32 is directly connected, Loopback0
S       172.31.0.66/32 [15/0] via H1_T11_0 tunnel 10.0.0.40, [1/0]
            [15/0] via H1_T22_0 tunnel 10.0.0.41, [1/0]
...
```

```
Spoke_1 (root) # get router info routing-table details 10.0.4.0/24

Routing table for VRF=0
Routing entry for 10.0.4.0/24
  Known via "bgp", distance 200, metric 0, best
  Last update 00:06:40 ago
  * 172.31.0.66, tag 1 (recursive via H1_T11_0 tunnel 10.0.0.42), best-match
              (recursive via H1_T22_0 tunnel 10.0.0.43), best-match
  * 172.31.0.66, tag 2 (recursive via H1_T11_0 tunnel 10.0.0.42), best-match
              (recursive via H1_T22_0 tunnel 10.0.0.43), best-match
              (recursive via H2_T11 tunnel 172.31.1.101), tag-match
              (recursive via H2_T22 tunnel 10.0.0.4), tag-match
```

```
Spoke_1(root) # diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
  Gen(22), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Member sub interface(4):
    1: seq_num(1), interface(H1_T11):
       1: H1_T11_0(93)
    3: seq_num(4), interface(H1_T22):
       1: H1_T22_0(94)
```

```
  Members(4):
    1: Seq_num(1 H1_T11), alive, sla(0x1), gid(0), cfg_order(0), cost(0), selected
    2: Seq_num(4 H1_T22_0), alive, sla(0x1), gid(0), cfg_order(3), cost(0), selected
    3: Seq_num(4 H1_T22), alive, sla(0x1), gid(0), cfg_order(3), cost(0), selected
    4: Seq_num(1 H1_T11_0), alive, sla(0x0), gid(0), cfg_order(0), cost(0), selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        10.0.0.0-10.255.255.255

Service(2): Address Mode(IPV4) flags=0x200 use-shortcut-sla
  Gen(10), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Members(2):
    1: Seq_num(6 H2_T11), alive, sla(0x1), gid(0), cfg_order(0), cost(0), selected
    2: Seq_num(9 H2_T22), alive, sla(0x1), gid(0), cfg_order(3), cost(0), selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        10.0.0.0-10.255.255.255
```

BGP next hop self-support for VPNv4 and VPNv6 route reflectors NEW

FortiGate now supports setting the BGP next hop attribute for VPNv4 and VPNv6 routes on route reflectors.

The `next-hop-self-rr-vpnv4` and `next-hop-self-rr-vpnv6` commands allow route reflectors to set the NEXT_HOP attribute to the local interface address when advertising VPNv4 or VPNv6 routes. This enhancement extends the existing `next-hop-self-rr behavior` that previously applied only to IPv4 and IPv6 unicast routes.

The feature can be configured at both the BGP neighbor and neighbor group levels, providing flexibility for large-scale MPLS VPN and route reflector deployments.

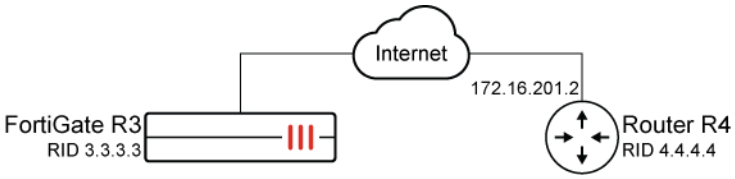
```
config router bgp
    config neighbor
        edit <ip>
            set next-hop-self-rr-vpnv4 {enable | disable}
            set next-hop-self-rr-vpnv6 {enable | disable}
        next
    end
    config neighbor-group
        edit <name>
            set next-hop-self-rr-vpnv4 {enable | disable}
            set next-hop-self-rr-vpnv6 {enable | disable}
        next
    end
end
```

Command	Description
`next-hop-self-rr-vpnv4 {enable | disable}`	Enable/disable setting of the next hop's address to interface's address for route-reflector VPNv4 routes.
`next-hop-self-rr-vpnv6 {enable | disable}`	Enable/disable setting of the next hop's address to interface's address for route-reflector VPNv6 routes.

BGP neighbor password

A BGP neighbor password is used for the neighbor range. Once a BGP group is configured, it uses a password to establish the neighborhood.

```
config router bgp
    config neighbor-group
        edit <name>
            set password <password>
        next
    end
end
```



To configure the BGP group:

1. Configure the R3 FortiGate settings:

```
config router bgp
    config neighbor-group
        edit "FGT"
            set soft-reconfiguration enable
            set remote-as 65050
            set local-as 65518
            set local-as-no-prepend enable
            set local-as-replace-as enable
            set route-map-in "del-comm"
            set keep-alive-timer 30
            set holdtime-timer 90
            set update-source "npu0_vlink0"
            set weight 1000
            set password ENC ********
        next
    end
    config neighbor-range
        edit 1
            set prefix 172.16.201.0 255.255.255.0
```

```
            set max-neighbor-num 10
            set neighbor-group "FGT"
        next
    end
end
```

2. Configure the R4 router settings:

```
config router bgp
    config neighbor
        edit "172.16.201.1"
            set soft-reconfiguration enable
            set remote-as 65518
            set password ********
        next
    end
end
```

BGP neighbor naming support NEW

BGP neighbors can now be assigned a name that appears alongside their IP address in the GUI and logs. This improves readability and simplifies peer identification in large or multi-tenant environments.

The `config neighbor` command includes a new option:

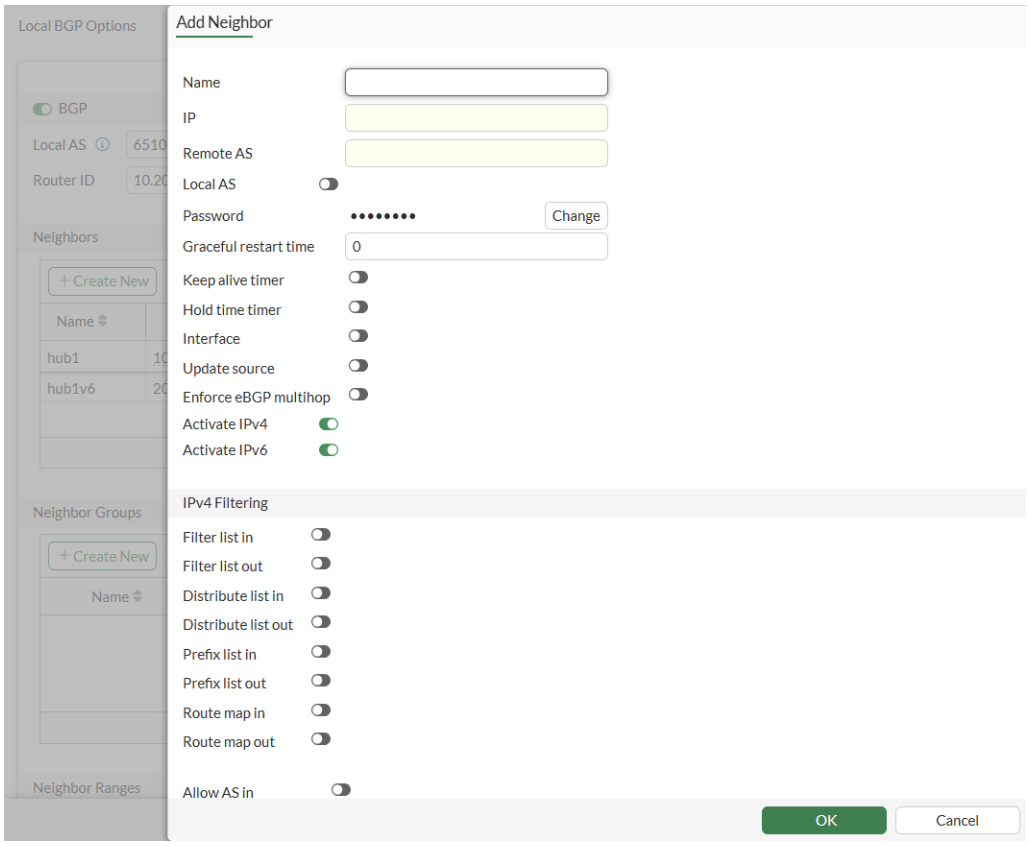
```
config neighbor
    edit <neighbor ID>
        set name <string>
        ...
    next
end
```

Option	Description
`name <string>`	Name of BGP neighbor.

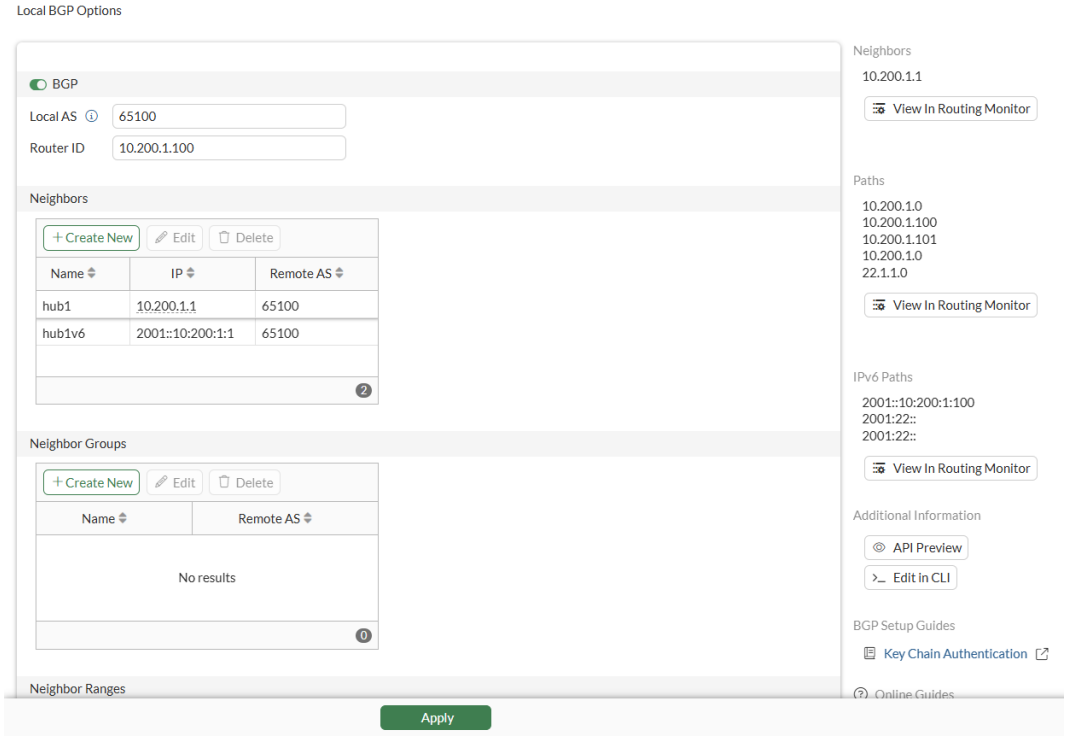
The GUI includes a new *Name* field.

To create a BGP neighbor with a name in the GUI:

1. Go to *Network* > *BGP*.
2. Under *Neighbors*, click *Create New*.
   The new *Name* field is available.



3. Enter a name for the BGP neighbor, set the remaining options as desired, and click *OK*.
   The name of the BGP neighbor (*hub1* and *hub1v6*) and its IP address is displayed in the GUI.



To create a BGP neighbor in the CLI:

```
config neighbor
    edit "10.200.1.1"
        set name "hub1"
        ...
    next
    edit "2001::10:200:1:1"
        set name "hub1v6"
        ...
    next
end
```

BGP neighbor names are included in some commands, such as:

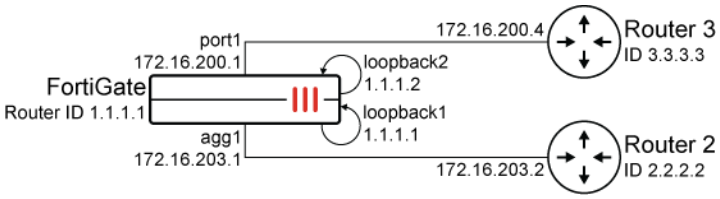
```
# diagnose ip router command show-vrf vd2 show run | grep "router bgp" -A50

... ...
 bgp ignore-unknown-optional-capability
 neighbor 10.200.1.1 name "hub1"
 neighbor 10.200.1.1 remote-as 65100
 neighbor 10.200.1.1 update-source vd2-lo1
 neighbor 10.200.1.1 advertisement-interval 1
 neighbor 10.200.1.1 timers connect 10
 neighbor 10.200.1.1 capability dynamic
 neighbor 10.200.1.1 soft-reconfiguration inbound
 no neighbor 10.200.1.1 additional-path
 neighbor 10.200.1.1 capability graceful-restart
 neighbor 10.200.1.1 route-map tagvd2 in
 neighbor 2001::10:200:1:1 name "hub1v6"
 neighbor 2001::10:200:1:1 remote-as 65100
 neighbor 2001::10:200:1:1 update-source vd2-lo1
 neighbor 2001::10:200:1:1 advertisement-interval 1
 neighbor 2001::10:200:1:1 timers connect 10
 neighbor 2001::10:200:1:1 capability dynamic
 no neighbor 2001::10:200:1:1 activate
 no neighbor 2001::10:200:1:1 additional-path
```

Log files also include the names of BGP neighbors, for example:

```
date=2025-08-14 time=11:20:27 eventtime=1755195627191151900 tz="-0700" logid="0103020300"
type="event" subtype="router" level="warning" vd="vd2" logdesc="BGP neighbor status changed"
msg="BGP: %BGP-5-ADJCHANGE: VRF 0 neighbor 10.200.1.1(\"hub1\") Down BGP Notification FSM-ERR"
date=2025-08-14 time=11:20:27 eventtime=1755195626192462532 tz="-0700" logid="0103020300"
type="event" subtype="router" level="warning" vd="vd2" logdesc="BGP neighbor status changed"
msg="BGP: %BGP-5-ADJCHANGE: VRF 0 neighbor 10.200.1.1(\"hub1\") Up "
date=2025-08-14 time=11:20:22 eventtime=1755195621411138773 tz="-0700" logid="0103020300"
type="event" subtype="router" level="warning" vd="vd2" logdesc="BGP neighbor status changed"
msg="BGP: %BGP-5-ADJCHANGE: VRF 0 neighbor 10.200.1.1(\"hub1\") Down BGP Notification FSM-ERR"
date=2025-08-14 time=11:20:22 eventtime=1755195621411132437 tz="-0700" logid="0103020304"
type="event" subtype="router" level="warning" vd="vd2" logdesc="Routing log warning" msg="BGP:
%BGP-3-NOTIFICATION: received from 10.200.1.1 6/2 (Cease/Administratively Shutdown.) 0 data-bytes
[]"
```

Defining a preferred source IP for local-out egress interfaces on BGP routes

The preferred source IP can be configured on BGP routes so that local-out traffic is sourced from that IP. In the following example, a route map is configured to set the preferred source IP so that the BGP route can support the preferred source.



To configure preferred source IPs for BGP routing:

1. Configure the route maps:

```
config router route-map
    edit "map1"
        config rule
            edit 1
                set set-ip-prefsrc 1.1.1.1
            next
        end
    next
    edit "map2"
        config rule
            edit 1
                set set-ip-prefsrc 1.1.1.2
            next
        end
    next
end
```

2. Configure the BGP settings:

```
config router bgp
    set as 65412
    set router-id 1.1.1.1
    set ibgp-multipath enable
    set cluster-id 1.1.1.1
    set graceful-restart enable
    config aggregate-address
        edit 1
            set prefix 172.28.0.0 255.255.0.0
            set as-set enable
            set summary-only enable
        next
    end
    config neighbor
```

```
        edit "3.3.3.3"
            set capability-graceful-restart enable
            set soft-reconfiguration enable
            set prefix-list-out "local-out"
            set remote-as 65412
            set route-map-in "map2"
            set route-map-out "as-prepend"
            set keep-alive-timer 30
            set holdtime-timer 90
            set update-source "loopback1"
            set route-reflector-client enable
        next
        edit "2.2.2.2"
            set advertisement-interval 5
            set activate6 disable
            set capability-graceful-restart enable
            set soft-reconfiguration enable
            set distribute-list-out "local-out-FGTB-deny"
            set remote-as 65412
            set route-map-in "map1"
            set route-map-out "as-rewrite"
            set keep-alive-timer 30
            set holdtime-timer 90
            set update-source "loopback1"
        next
    end
end
```

To verify the configuration:

1. Verify the BGP routing table for 172.25.1.0/24:

```
# get router info bgp network 172.25.1.0/24
VRF 0 BGP routing table entry for 172.25.1.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local
    2.2.2.2 (metric 10050) from 2.2.2.2 (2.2.2.2)
      Origin IGP metric 0, localpref 100, valid, internal, best, prefsrc 1.1.1.1
      Last update: Wed Jan 25 15:15:48 2023
```

2. Verify the BGP routing table for 172.28.5.0/24:

```
# get router info bgp network 172.28.5.0/24
VRF 0 BGP routing table entry for 172.28.5.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table, Advertisements suppressed by an
aggregate.)
  Not advertised to any peer
  Original VRF 0
  65050, (Received from a RR-client)
    3.3.3.3 (metric 11000) from 3.3.3.3 (3.3.3.3)
```

```
      Origin IGP metric 0, localpref 100, valid, internal, best, prefsrc 1.1.1.2
      Last update: Wed Jan 25 15:15:48 2023
```

3. Verify the kernel routing table for 172.28.5.0/24:

```
# get router info kernel | grep -B 2 172.28.5.0/24
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0->172.28.1.0/24 pref=1.1.1.2
gwy=172.16.200.4 dev=9(port1)
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0->172.28.2.0/24 pref=1.1.1.2
gwy=172.16.200.4 dev=9(port1)
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0->172.28.5.0/24 pref=1.1.1.2
gwy=172.16.200.4 dev=9(port1)
```

4. Verify the kernel routing table for 172.25.1.0/24:

```
# get router info kernel | grep -A 2 172.25.1.0/24
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0->172.25.1.0/24 pref=1.1.1.1
gwy=172.16.203.2 dev=33(agg1)
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0->172.26.1.0/24 pref=1.1.1.1
gwy=172.16.203.2 dev=33(agg1)
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0->172.26.2.0/24 pref=1.1.1.1
gwy=172.16.203.2 dev=33(agg1)
```

   The FortiGate learns routes from router 3.3.3.3 and prefers the source IP of 1.1.1.2. It learns routes from router 2.2.2.2 and prefers source IP of 1.1.1.1.

5. Run a sniffer trace after some traffic passes.
   a. When trying to reach a destination in the 172.25.1.0/0 subnet through router 2.2.2.2:

```
# diagnose sniffer packet any "icmp" 4
interfaces=[any]
filters=[icmp]
9.244334 agg1 out 1.1.1.1 -> 172.25.1.2: icmp: echo request
9.244337 port12 out 1.1.1.1 -> 172.25.1.2: icmp: echo request
10.244355 agg1 out 1.1.1.1 -> 172.25.1.2: icmp: echo request
10.244357 port12 out 1.1.1.1 -> 172.25.1.2: icmp: echo request
```

   b. When trying to reach a destination in the 172.28.5.0/24 subnet through router 3.3.3.3:

```
# diagnose sniffer packet any "icmp" 4
interfaces=[any]
filters=[icmp]
2.434035 port1 out 1.1.1.2 -> 172.28.5.2: icmp: echo request
3.434059 port1 out 1.1.1.2 -> 172.28.5.2: icmp: echo request
```

   Traffic destined for the 172.25.1.0/24 subnet uses 1.1.1.1 as source. Traffic destined for the 172.28.5.0/24 subnet uses 1.1.1.2 as source.

BGP multi-exit discriminator

Border Gateway Protocol (BGP) is the routing protocol that governs how internet traffic is efficiently routed between autonomous systems (AS). BGP uses path attributes for its best path calculation to a network.

Multi-Exit Discriminator (MED) is a BGP path attribute that discriminates among multiple exit or entry points to the same neighboring AS. MED is also known as Optional Non-Transitive path attribute. The lower the MED value, the more preferred the path is to the receiving router.

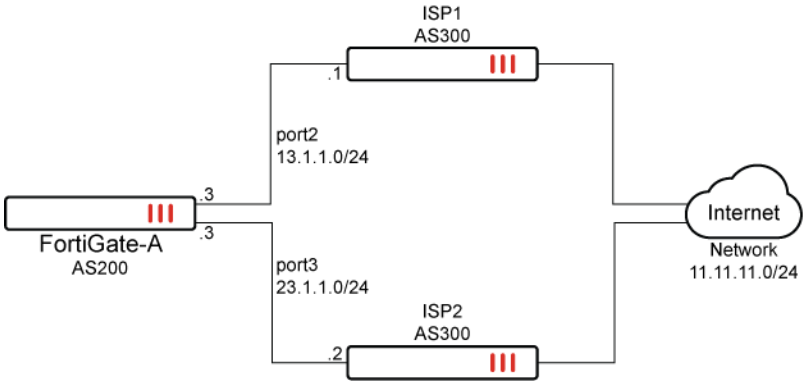
MED is typically utilized when an AS has multiple exit points to another AS. In such cases, the AS may want to influence incoming traffic by advertising different MED values for the same route.

Common use case

A common use case for MED is in the context of a dual-homed AS. In a dual-homed AS scenario, the FortiGate connects to two different ISPs belonging to the same AS for redundancy and load balancing. By manipulating the MED value, the AS can direct traffic to one ISP over the other. For instance, if one link is congested, the AS can advertise a higher MED value for that link, making the other link more suitable to incoming traffic.

MED is a BGP path attribute that discriminates among multiple exit or entry points to the same neighboring AS. If an administrator has two or more eBGP peering to different AS then the local preference can be used to influence the routing decision.

The following example discusses the use of MED in dual-homed AS network and the configuration of MED Path Attribute on FortiOS.



Characteristics of the topology include the following:

- FortiGate-A has two internet service providers: ISP1 and ISP2.
- FortiGate-A belongs to AS 200.
- ISP1 and ISP2 both belong to AS 300.
- FortiGate-A will establish eBGP peering relationships with ISP1 and ISP2.
- Network 11.11.11.0/24 resides on the Internet and is reachable by both ISP1 and ISP2.
- eBGP multipath is enabled on FortiGate-A if FortiGate-A needs to perform equal cost load-balancing of traffic between both ISP1 and ISP2 to reach to 11.11.11.0/24.
- The network 11.11.11.0/24 on the Internet is being advertised by both ISP1 and ISP2 to FortiGate-A through eBGP.

For this example, the traffic originating from behind the FortiGate-A should prefer ISP1 rather than ISP2 to reach 11.11.11.0/24, and needs to use the BGP MED Path Attribute. This is done by configuring `set set-metric` in a route map configuration and ensuring the MED value of ISP1 is less than that of ISP2.

To configure MED in a dual-homed AS network using the CLI:

1. Configure eBGP peering on FortiGate-A by specifying the BGP neighbors:

```
config router bgp
    set as 200
    set router-id 2.2.2.2
    config neighbor
        edit "13.1.1.1"
            set remote-as 300
        next
        edit "23.1.1.2"
            set remote-as 300
        next
    end
end
```

2. Configure eBGP on ISP1 and ISP2, and advertise the 11.11.11.0/24 network:

```
config router bgp
    set as 300
    set router-id 3.3.3.3
    config neighbor
        edit "13.1.1.3"
            set remote-as 200
        next
    end
    config network
        edit 1
            set prefix 11.11.11.0 255.255.255.0
        next
    end
end
```

3. Verify the eBGP neighbors on Fortigate-A with ISP1 and ISP2:

```
# get router info bgp summary
VRF 0 BGP router identifier 2.2.2.2, local AS number 200
BGP table version is 6
1 BGP AS-PATH entries
0 BGP community entries

Neighbor V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
13.1.1.1 4        300      48      50        0    0    0 00:25:45        1
23.1.1.2 4        300      56      60        6    0    0 00:26:30        1
```

4. Configure the prefix list to filter and select the 11.11.11.0/24 subnet inside the prefix list:
   a. Configure ISP1:

```
config router prefix-list
    edit "ISP1_Prefix_List"
        config rule
            edit 1
```

```
                set prefix 11.11.11.0 255.255.255.0
            next
        end
    next
end
```

b. Configure ISP2:

```
config router prefix-list
    edit "ISP2_Prefix_List"
        config rule
            edit 1
                set prefix 11.11.11.0 255.255.255.0
            next
        end
    next
end
```

5. Configure route maps and configure the MED value:

a. Configure ISP1:

```
config router route-map
    edit "MED_Route_MAP"
        config rule
            edit 1
                set match-ip-address "ISP1_Prefix_List"
                set set-metric 300
            next
        end
    next
end
```

b. Configure ISP2:

```
config router route-map
    edit "MED_Route_MAP_2"
        config rule
            edit 1
                set match-ip-address "ISP2_Prefix_List"
                set set-metric 400
            next
        end
    next
end
```

6. Apply the route map in the outbound direction:

Applying the configuration in the outbound direction ensures that the MED is changed to the respective value for the route 11.11.11.0/24 specified in the prefix list.

a. Configure the ISP1 BGP:

```
config router bgp
    set as 300
    set router-id 3.3.3.3
    config neighbor
        edit "13.1.1.3"
            set remote-as 200
            set route-map-out "MED_Route_MAP"
        next
    end
end
```

b. Configure the ISP2 BGP:

```
config router bgp
    set as 300
    set router-id 4.4.4.4
    config neighbor
        edit "23.1.1.3"
            set remote-as 200
            set route-map-out "MED_Route_MAP_2"
        next
    end
end
```

7. Verify that ISP1 is selected as the best path to reach the 11.11.11.0/24 network:

The neighbor that is considered the best, valid route is marked with a *>.

```
# get router info bgp network
VRF 0 BGP table version is 6, local router ID is 2.2.2.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric LocPrf Weight RouteTag Path
*> 11.11.11.0/24    13.1.1.1               300             0        0 300 i <-/1>
*                   23.1.1.2               400             0        0 300 i <-/->

Total number of prefixes 1
```

8. Verify the routing table of FortiGate-A:

The routing table only contains the best and valid paths.

```
# get router info routing-table details
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
```

```
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
B       11.11.11.0/24 [20/300] via 13.1.1.1, port2, 00:37:47
C       13.1.1.0/24 is directly connected, port2
C       23.1.1.0/24 is directly connected, port3
```

TCP Authentication Option advanced security measures

BGP incorporates TCP Authentication Option (TCP-AO) advanced security measures, which supports stronger algorithms, such as AES-128 CMAC and HMAC-SHA1. This integration bolsters the security of and enhances the reliability of BGP connections and contributes to the overall security of the internet.

To set the algorithm to AES-128 CMAC:

```
config router key-chain
    edit <name>
        config key
            edit <id>
                set algorithm cmac-aes128
            next
        end
    next
end
```

To select the key-chain with the TCP-AO in a neighbor or neighbor group:

```
config router bgp
    config {neighbor | neighbor-group}
        edit <ip>
            set auth-options <key-chain>
        end
    next
end
```

To debug the TCP authentication options:

```
diagnose sys tcp-auth-options
```

Example

In this example, the router BGP neighbor is configured to use the AES-128 CMAC algorithm.

To configure the router BGP to use the AES-128 CMAC algorithm:

1. Configure the router key-chain to use the AES-128 CMAC algorithm:

```
config router key-chain
    edit "11"
        config key
            edit "1"
                set accept-lifetime 01:01:01 01 01 2021 2147483646
                set send-lifetime 01:01:01 01 01 2021 2147483646
                set key-string **********
                set algorithm cmac-aes128
            next
        end
    next
end
```

2. Apply the key-chain to the BGP neighbor or neighbor group:
   The key-chain is applied to the BGP neighbor with IP address 2.2.2.2.

```
config router bgp
    set as 65412
    config neighbor
        edit "2.2.2.2"
            set auth-options "11"
        next
    end
end
```

3. Verify that the router BGP is using the algorithm.
   The command output shows that BGP neighbor 2.2.2.2 is using the AES-128 CMAC algorithm.

```
# diagnose sys tcp-auth-options

VFID=0 send-id=1 recv-id=1 flags=0x784 keylen=6
alg=2(aes128) addr=2.2.2.2
send-begin: Fri Jan  1 01:01:01 2021
send-end: Wed Jan 19 04:15:07 2089
recv-begin: Fri Jan  1 01:01:01 2021
recv-end: Wed Jan 19 04:15:07 2089
```

Assigning multiple remote Autonomous Systems to a single BGP neighbor group

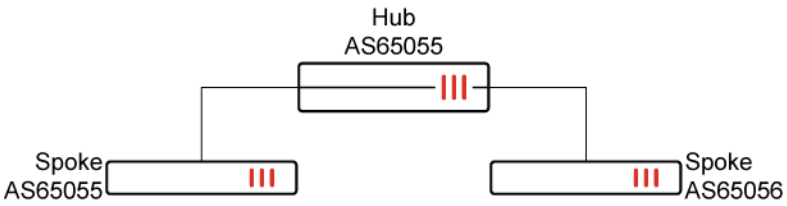
FortiOS allows the assignment of multiple remote Autonomous Systems (AS) to a single BGP neighbor group using the `remote-as-filter` command. This is achieved through the utilization of AS path lists. This enhancement offers increased flexibility and efficiency in managing BGP configurations, especially in intricate network environments.

```
config router bgp
    set as <local AS>
    config neighbor-group
        edit <name>
            set remote-as-filter <BGP filter for remote AS>
        next
    end
end
```

The `remote-as` and `remote-as-filter` commands are mutually exclusive in the BGP neighbor group. Unset the pre-existing value to use the other command. For example, to change from using `remote-as` to `remote-as-filter`, implement `unset remote-as`.

Example

The following example demonstrates assigning the remote AS of two spokes (AS65505 and AS65006) to a single BGP neighbor group of a hub (AS65505).



To assign multiple remote AS to a single BGP neighbor group in the local hub:

1. Configure the AS path list:

```
config router aspath-list
    edit "aslist1"
        config rule
            edit 1
                set action permit
                set regexp "^6550[1-6]$"
            next
        end
    next
end
```

2. Configure the BGP neighbor group of the hub to point to the AS path list:

```
config router bgp
    set as 65505
    config neighbor-group
        edit "gr1"
            set remote-as-filter "aslist1"
        next
    end
end
```

3. Configure the BGP neighbor range:

```
config router bgp
    config neighbor-range
        edit 1
            set prefix 10.10.100.0 255.255.255.0
            set neighbor-group "gr1"
        next
    end
end
```

4. Verify that multiple remote AS have been assigned to the BGP neighbor group:

```
# get router info bgp summary

VRF 0 BGP router identifier 11.11.11.11, local AS number 65505
BGP table version is 4
2 BGP AS-PATH entries
0 BGP community entries
Next peer check timer due in 55 seconds

Neighbor    V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.10.100.2 4      65505    8240    8229        4    0    0 5d00h03m        2
10.10.100.3 4      65506    8229    8253        4    0    0 4d23h59m        2
Total number of neighbors 2
```

Using firewall addresses and groups for BGP network prefixes

Firewall addresses (`ipmask` and `interface-subnet` types) and groups can be used for BGP prefixes.

```
config firewall address
    edit <address>
        set allow-routing {enable | disable}
    next
end
config firewall addrgrp
    edit <group>
        set allow-routing {enable | disable}
    next
end
config router bgp
    config network
        edit <id>
            set prefix-name <address or group>
        next
    end
end
```

Command	Description
allow-routing {enable | disable}	Enable/disable use of this address/group in routing configurations.

Command	Description
prefix-name <address or group>	Name of firewall address or address group.

To configure firewall addresses and groups and use them in BGP network prefixes:

1. Configure firewall addresses and groups with allow-routing enabled:

```
config firewall address
    edit "222"
        set allow-routing enable
        set subnet 22.2.3.0 255.255.255.0
    next
    edit "333"
        set allow-routing enable
        set subnet 33.1.1.0 255.255.255.0
    next
    edit "555"
        set type interface-subnet
        set allow-routing enable
        set subnet 66.6.2.0 255.255.255.0
        set interface "port6"
    next
    edit "vlan55 address"
        set type interface-subnet
        set allow-routing enable
        set subnet 55.2.2.0 255.255.255.0
        set interface "vlan55"
    next
end
```

```
config firewall addrgrp
    edit "group1"
        set allow-routing enable
        set member "222" "555"
    next
end
```

2. Configure BGP network prefixes

```
config router bgp
    ...
    config network
        edit 2
            set prefix-name "333"
        next
        edit 20
            set network-import-check disable
            set prefix-name "group1"
        next
        edit 66
```

```
            set prefix-name "vlan55 address"
        next
    end
    ...
end
```

Dynamic BGP learning of ISDB reputation IPs NEW

Dynamic BGP-based learning for Botnet and Spam ISDB categories allows FortiGate to automatically receive and advertise these IPs for more responsive, threat-aware routing.

```
config router bgp
    config network
        edit <id>
            set internet-service-name {Botnet-C&C.Server | Spam-Spamming.Server}
        next
    end
end
```

To test using the Botnet-C&C.Server internet service:

1. Configure BGP:

```
config router bgp
    set as 65412
    set router-id 2.2.2.2
    set network-import-check disable
    config neighbor
        edit "3.3.3.3"
            set advertisement-interval 5
            set capability-graceful-restart enable
            set ebgp-enforce-multihop enable
            set next-hop-self enable
            set soft-reconfiguration enable
            set remote-as 65412
            set route-map-out "as-prepend"
            set keep-alive-timer 30
            set holdtime-timer 90
            set update-source "loopback1"
        next
    end
    config network
        edit 1
            set prefix 172.28.1.0 255.255.255.0
        next
        edit 2
            set prefix 172.28.2.0 255.255.255.0
        next
        edit 3
            set prefix 172.25.1.0 255.255.255.0
```

```
        next
        edit 4
            set internet-service-name "Botnet-C&C.Server"
        next
    end
    config network6
        edit 1
            set prefix6 2000:172:27:1::/64
        next
    end
end
```

2. Check the internet service ID:

```
# diagnose internet-service id 3080383
...
222.158.197.138-222.158.197.138 country(392) region(65535) city(65535) blocklist(0x1)
reputation(1), popularity(1) domain(0) botnet(7630624) proto(6) port(80)
222.165.194.68-222.165.194.68 country(360) region(65535) city(65535) blocklist(0x1) reputation
(1), popularity(1) domain(0) botnet(7630624) proto(6) port(50486)
222.165.205.154-222.165.205.154 country(360) region(65535) city(65535) blocklist(0x1)
reputation(1), popularity(1) domain(0) botnet(7630624) proto(6) port(8089)
222.173.92.154-222.173.92.154 country(156) region(65535) city(65535) blocklist(0x1) reputation
(1), popularity(1) domain(0) botnet(7630624) proto(6) port(32075)
222.191.243.187-222.191.243.187 country(156) region(65535) city(65535) blocklist(0x1)
reputation(1), popularity(1) domain(0) botnet(7630624) proto(6) port(45730)
222.211.72.29-222.211.72.29 country(156) region(65535) city(65535) blocklist(0x1) reputation
(1), popularity(1) domain(0) botnet(7630020) proto(6) port(80)
222.217.68.17-222.217.68.17 country(156) region(65535) city(65535) blocklist(0x1) reputation
(1), popularity(1) domain(0) botnet(7630624) proto(6) port(35165)
222.252.23.5-222.252.23.5 country(704) region(65535) city(65535) blocklist(0x1) reputation(1),
popularity(1) domain(0) botnet(7630624) proto(6) port(8080)
223.100.166.3-223.100.166.3 country(156) region(65535) city(65535) blocklist(0x1) reputation
(1), popularity(1) domain(0) botnet(7630624) proto(6) port(49247)
223.165.243.209-223.165.243.209 country(410) region(65535) city(65535) blocklist(0x1)
reputation(1), popularity(1) domain(0) botnet(7630624) proto(6) port(47205)
```

3. On the BGP peer, check the networks/subnets of this internet service:

```
# get router info routing-table bgp
...
B       221.194.47.218/32 [200/0] via 2.2.2.2 (recursive via 172.16.200.2, port1), 00:00:15,
[1/0]
B       221.229.204.124/32 [200/0] via 2.2.2.2 (recursive via 172.16.200.2, port1), 00:00:15,
[1/0]
B       222.129.33.141/32 [200/0] via 2.2.2.2 (recursive via 172.16.200.2, port1), 00:00:15,
[1/0]
B       222.129.35.9/32 [200/0] via 2.2.2.2 (recursive via 172.16.200.2, port1), 00:00:15,
[1/0]
B       222.158.197.138/32 [200/0] via 2.2.2.2 (recursive via 172.16.200.2, port1), 00:00:15,
[1/0]
B       222.165.194.68/32 [200/0] via 2.2.2.2 (recursive via 172.16.200.2, port1), 00:00:15,
```

```
[1/0]
B       222.165.205.154/32 [200/0] via 2.2.2.2 (recursive via 172.16.200.2, port1), 00:00:15,
[1/0]
B       222.173.92.154/32 [200/0] via 2.2.2.2 (recursive via 172.16.200.2, port1), 00:00:15,
[1/0]
B       222.191.243.187/32 [200/0] via 2.2.2.2 (recursive via 172.16.200.2, port1), 00:00:15,
[1/0]
B       222.211.72.29/32 [200/0] via 2.2.2.2 (recursive via 172.16.200.2, port1), 00:00:15,
[1/0]
B       222.217.68.17/32 [200/0] via 2.2.2.2 (recursive via 172.16.200.2, port1), 00:00:15,
[1/0]
B       222.252.23.5/32 [200/0] via 2.2.2.2 (recursive via 172.16.200.2, port1), 00:00:15,
[1/0]
B       223.100.166.3/32 [200/0] via 2.2.2.2 (recursive via 172.16.200.2, port1), 00:00:15,
[1/0]
B       223.165.243.209/32 [200/0] via 2.2.2.2 (recursive via 172.16.200.2, port1), 00:00:14,
[1/0]
```

Graceful BGP shutdown NEW

In this enhancement, FortiGate adds support for graceful BGP shutdown (RFC 8326) to allow controlled and orderly shutdown of BGP sessions. This ensures peers have time to adjust to the change and lower the local preferences of the routes associated with the graceful shutdown community and prefix.

CLI syntax

The following commands can be used to configure graceful BGP shutdown for the initiator and receiver. These settings will persist after a reboot.

```
config router bgp
    config neighbor
        edit <IP address>
            set shutdown {enable | disable | graceful | graceful-soft}
            set graceful-shutdown-community <community>
            set graceful-shutdown-local-preference <local preference>
            set graceful-shutdown-delay <delay>
        next
    end
end
```

Command	Description
`shutdown {enable | disable | graceful | graceful-soft}`	• `graceful`: on the graceful BGP shutdown initiator, put the device into graceful shutdown mode. BGP sessions with peers will be closed once delay is reached. • `graceful-soft`: on the graceful BGP shutdown initiator, put the device into graceful-soft shutdown mode. BGP sessions will not be closed. Instead, after the delay is reached, the local-preference in peers will be reverted to original values.

Command	Description
	This setting should only be applied to the initiator.
`graceful-shutdown-community <community>`	As the initiator, specify the community in which to advertise the graceful shutdown. As the receiver, apply the `graceful-shutdown-community` to the graceful shutdown initiator's routes and perform action on the local preference of those routes. This setting is required for both the initiator and receiver.
`graceful-shutdown-local-preference <local preference>`	The receiver lowers the local preference of routes tagged by the `graceful-shutdown-community` to the defined <local preference>.
`graceful-shutdown-delay <delay>`	After the <delay> period in seconds, initiator performs the actions for either `graceful` shutdown or `graceful-soft` shutdown. If set to `0`, then FortiGate will stay in shutdown mode until manually canceled or stopped.

If administrators want to apply a one-time graceful BGP shutdown that does not persist through a reboot, use the following commands:

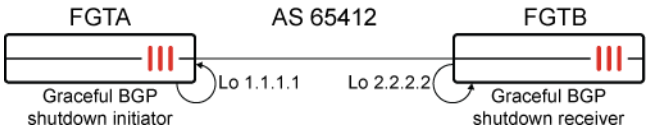
Command	Description
`execute router clear bgp all gshut [soft | cancel] [delay <60-36000>]`	On the initiator, initiate graceful shutdown on all routes. • `soft`: optionally perform a graceful-soft shutdown. • `cancel`: cancel a previously initiated shutdown. • `delay`: specify the delay timer in seconds for the shutdown. If no delay is specified, the timer is indefinite and shutdown does not occur.
`execute router clear bgp ip <IP/IPv6 address> gshut [soft | cancel] [delay <60-36000>]`	On the initiator, initiate graceful shutdown on specific routes based on the IP <IP/IPv6 address>.
`execute router clear bgp as <AS number> gshut [soft | cancel] [delay <60-36000>]`	On the initiator, initiate graceful shutdown on specific routes based on the AS <AS number>.
`execute router clear bgp group <group name> gshut [soft | cancel] [delay <60-36000>]`	On the initiator, initiate graceful shutdown on specific routes based on the group <group name>.

In addition, the default community `gshut` is added to:

- router.community-list.rule.match
- router.route-map.rule.set-community

Examples

In the following examples, two FortiGates have established BGP adjacencies.



FGTA needs to be shut down for maintenance. To pre-emptively inform its peer (FGTB), FGTA uses graceful BGP shutdown to announce the shutdown and apply the gshut community to its routes. This allows FGTB to set all routes learned from FGTA to community gshut and lower the local-preference for those routes.

Example 1: Apply graceful BGP shutdown using persistent configs

On FGTB, the graceful BGP shutdown receiver, apply the following BGP configuration:

```
config router bgp
    set as 65412
    set router-id 2.2.2.2
    set network-import-check disable
    config neighbor
        edit "1.1.1.1"
            set advertisement-interval 5
            set activate6 disable
            set capability-graceful-restart enable
            set ebgp-enforce-multihop enable
            set next-hop-self enable
            set soft-reconfiguration enable
            set remote-as 65412
            set keep-alive-timer 30
            set holdtime-timer 90
            set update-source "loopback1"
            set graceful-shutdown-community "gshut"
            set graceful-shutdown-local-preference 27
        next
    end
end
```

On FGTA, the graceful BGP shutdown initiator, apply the following BGP configuration:

```
config router bgp
    set as 65412
    set router-id 1.1.1.1
    set graceful-restart enable
    config neighbor
        edit "2.2.2.2"
            set advertisement-interval 5
            set activate6 disable
            set capability-graceful-restart enable
            set ebgp-enforce-multihop enable
            set next-hop-self enable
            set shutdown graceful
            set soft-reconfiguration enable
            set remote-as 65412
            set keep-alive-timer 30
            set holdtime-timer 90
```

```
            set update-source "loopback1"
            set graceful-shutdown-community "gshut"
            set graceful-shutdown-local-preference 221
            set graceful-shutdown-delay 60
        next
    end
end
```

Once graceful shutdown mode is enabled on FGTA, it will notify FGTB of the shutdown by advertising the gshut community.

On FGTA:

```
# get router info bgp summary

VRF 0 BGP router identifier 1.1.1.1, local AS number 65412
BGP table version is 5
1 BGP AS-PATH entries
1 BGP community entries

Neighbor V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2.2.2.2  4      65412     444     464        5    0    0 00:00:54        3 (In gshut)

Total number of neighbors 1
```

FGTB's learned routes will change community and apply the configured local-preference of 27:

```
# get router info bgp network 172.16.207.2/32
VRF 0 BGP routing table entry for 172.16.207.2/32
Paths: (1 available, best #1, table Default-IP-Routing-Table, graceful shutdown)
  Not advertised to any peer
  Original VRF 0
  Local
    1.1.1.1 from 1.1.1.1 (1.1.1.1)
      Origin incomplete distance 200 metric 0, localpref 27, valid, internal, best
      Community: gshut
      Advertised Path ID: 1
      Last update: Fri Sep 26 15:20:23 2025
```

After the configured `graceful-shutdown-delay` expires, FGTA will close the BGP session with FGTB.

On FGTA:

```
# get router info bgp summary

VRF 0 BGP router identifier 1.1.1.1, local AS number 65412
BGP table version is 6
1 BGP AS-PATH entries
1 BGP community entries

Neighbor V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2.2.2.2  4      65412     445     467        0    0    0    never Idle        (In gshut)
```

As FGTB in this example does not have any other paths, all learned routes to 172.16.207.2/32 from FGTA have disappeared.

On FGTB:

```
# get router info bgp network 172.16.207.2/32
% Network not in table
```

Example 2: Apply one-time graceful BGP shutdown using execute commands

On FGTA, use similar BGP settings as in Example 1:

```
config router bgp
    set as 65412
    set router-id 1.1.1.1
    set graceful-restart enable
    config neighbor
        edit "2.2.2.2"
            set graceful-shutdown-community "gshut"
            set graceful-shutdown-local-preference 221
            set graceful-shutdown-delay 0
        next
    end
end
```

`graceful-shutdown-delay` does not need to be configured.

On FGTB, use the same BGP settings as in Example 1:

```
config router bgp
    set as 65412
    set router-id 2.2.2.2
    set network-import-check disable
    config neighbor
        edit "1.1.1.1"
            set graceful-shutdown-community "gshut"
            set graceful-shutdown-local-preference 27
        next
    end
end
```

Initiate the one-time graceful shutdown from FGTA with a delay of 60 seconds:

```
# execute router clear bgp ip 2.2.2.2 gshut delay 60
```

Once graceful shutdown mode is enabled on FGTA, it will notify FGTB of the shutdown by advertising the gshut community.

On FGTA:

```
# get router info bgp summary

VRF 0 BGP router identifier 1.1.1.1, local AS number 65412
BGP table version is 5
```

```
1 BGP AS-PATH entries
1 BGP community entries

Neighbor V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2.2.2.2  4      65412     444     464        5    0    0 00:00:52        3 (In gshut)

Total number of neighbors 1
```

FGTB's learned routes will change community and apply the configured local-preference of 27 as in Example 1. After 60 seconds, BGP session will be terminated and routes from FGTA will be removed from FGTB.

Example 3: Apply graceful-soft BGP shutdown using persistent configs

Instead of completely shutting down BGP on the initiator, apply graceful-soft so that after the delay timeout, BGP reverts to normal behavior. BGP graceful shutdown receiver reverts the local-preferences of its routes.

On FGTB, the graceful BGP shutdown receiver, apply the same BGP configurations as Example 1.

On FGTA, the graceful-soft BGP shutdown initiator, apply the following BGP configurations:

```
config router bgp
    set as 65412
    set router-id 1.1.1.1
    set bestpath-cmp-routerid enable
    set ibgp-multipath enable
    set cluster-id 1.1.1.1
    set graceful-restart enable
    config neighbor
        edit "2.2.2.2"
            set advertisement-interval 5
            set activate6 disable
            set capability-graceful-restart enable
            set ebgp-enforce-multihop enable
            set next-hop-self enable
            set shutdown graceful-soft
            set soft-reconfiguration enable
            set remote-as 65412
            set keep-alive-timer 30
            set holdtime-timer 90
            set update-source "loopback1"
            set graceful-shutdown-community "gshut"
            set graceful-shutdown-local-preference 221
            set graceful-shutdown-delay 300
        next
    end
end
```

Once graceful shutdown mode is enabled on FGTA, it will notify FGTB of the shutdown by advertising the gshut community:

```
# get router info bgp summary
VRF 0 BGP router identifier 1.1.1.1, local AS number 65412
BGP table version is 12
```

```
1 BGP AS-PATH entries
1 BGP community entries
Neighbor V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2.2.2.2  4      65412     599     643       12    0    0 00:00:58        3 (In gshut)
Total number of neighbors 1
```

```
# get router info bgp neighbors | grep shut
 In graceful shut down , will go back to normal in 221 seconds
```

FGTB's learned routes will change community and apply the configured local-preference of 27:

```
# get router info bgp network 172.16.207.2/32
VRF 0 BGP routing table entry for 172.16.207.2/32
Paths: (1 available, best #1, table Default-IP-Routing-Table, graceful shutdown)
  Not advertised to any peer
  Original VRF 0
  Local
    1.1.1.1 from 1.1.1.1 (1.1.1.1)
      Origin incomplete distance 200 metric 0, localpref 27, valid, internal, best
      Community: gshut
      Advertised Path ID: 1
      Last update: Fri Sep 27 08:15:23 2025
```

After the delay time, both FortiGates will go back to normal mode:

On FGTA:

```
# get router info bgp summary
VRF 0 BGP router identifier 1.1.1.1, local AS number 65412
BGP table version is 12
1 BGP AS-PATH entries
1 BGP community entries
Neighbor V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2.2.2.2  4      65412     612     657       12    0    0 00:06:24        3
Total number of neighbors 1
```

```
# get router info bgp neighbors | grep shut
```

On FGTB:

```
# get router info bgp network 172.16.207.2/32
VRF 0 BGP routing table entry for 172.16.207.2/32
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local
    1.1.1.1 from 1.1.1.1 (1.1.1.1)
      Origin incomplete distance 200 metric 0, localpref 100, valid, internal, best
      Advertised Path ID: 1
      Last update: Sat Sep 27 08:17:37 2025
```

Example 4: Apply one-time graceful-soft BGP shutdown using execute command

Instead of using persistent configs like in Example 3, use the following execute command to perform a one-tine graceful-soft BGP shutdown:

```
# execute router clear bgp ip 2.2.2.2 gshut soft delay 100
```

After run this command on FGTA, FGTA will go into graceful shutdown mode, and FGTB will set `community` and `local_pref` value for all routes learned from FGTA.

After the delay time of 100 seconds, FGTA will go back to normal mode, and peer will restore the `community` and `local_pref` value for all routes learned from FGTA.

Example 5: Apply graceful BGP shutdown with indefinite delay using persistent configs on IPv6 routes

In this example, we apply graceful BGP shutdown mode indefinitely. IPv6 routes on the graceful BGP shutdown receiver will keep routes in the `graceful-shutdown-local-preference` indefinitely while in this mode.

On FGTB, apply the following configuration:

```
config router bgp
    set as 65412
    set router-id 2.2.2.2
    set network-import-check disable
    config neighbor
        edit "2000::1:1:1:1"
            set activate disable
            set soft-reconfiguration6 enable
            set remote-as 65412
            set keep-alive-timer 30
            set holdtime-timer 90
            set update-source "loopback1"
            set graceful-shutdown-community "gshut"
            set graceful-shutdown-local-preference 276
        next
    end
end
```

On FGTA, apply the following BGP configurations:

```
config router bgp
    set as 65412
    set router-id 1.1.1.1
    set bestpath-cmp-routerid enable
    set ibgp-multipath enable
    set cluster-id 1.1.1.1
    set graceful-restart enable
    config neighbor
        edit "2000::2:2:2:2"
            set advertisement-interval 5
            set activate disable
            set capability-graceful-restart6 enable
```

```
            set shutdown graceful
            set soft-reconfiguration6 enable
            set remote-as 65412
            set keep-alive-timer 30
            set holdtime-timer 90
            set update-source "loopback1"
            set graceful-shutdown-community "gshut"
            set graceful-shutdown-local-preference 2216
            set graceful-shutdown-delay 0
        next
    end
end
```

When `set shutdown graceful` is configured, FGTA is in graceful shutdown mode indefinitely:

On FGTA:

```
# get router info6 bgp summary
VRF 0 BGP router identifier 1.1.1.1, local AS number 65412
BGP table version is 10
1 BGP AS-PATH entries
1 BGP community entries
Neighbor      V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2000::2:2:2:2 4      65412    3235    3243       10    0    0 17:00:28        1 (In gshut)
Total number of neighbors 1
```

On FGTB:

```
# get router info6 bgp network 2000:172:16:211::/64
VRF 0 BGP routing table entry for 2000:172:16:211::/64
Paths: (1 available, best #1, table Default-IP-Routing-Table, graceful shutdown)
  Not advertised to any peer
  Original VRF 0
  Local
    2000::1:1:1:1 (metric 1000) from 2000::1:1:1:1 (1.1.1.1)
      Origin incomplete distance 200 metric 0, localpref 276, valid, internal, best
      Community: gshut
      Advertised Path ID: 1
      Last update: Fri Sep 26 15:41:04 2025
```

Troubleshooting BGP

There are some features in BGP that are used to deal with problems that may arise. Typically, the problems with a BGP network that has been configured involve routes going offline frequently. This is called route flap and causes problems for the routers using that route.

Clearing routing table entries

To see if a new route is being properly added to the routing table, you can clear all or some BGP neighbor connections (sessions) using the `execute router clear bgp` command.

For example, if you have 10 routes in the BGP routing table and you want to clear the specific route to IP address 10.10.10.1, enter the following CLI command:

```
# execute router clear bgp ip 10.10.10.1
```

To remove all routes for AS number 650001, enter the following CLI command:

```
# execute router clear bgp as 650001
```

Route flap

When routers or hardware along a route go offline and back online that is called a route flap. Flapping is the term that is used if these outages continue, especially if they occur frequently.

Route flap is a problem in BGP because each time a peer or a route goes down, all the peer routers that are connected to that out-of-service router advertise the change in their routing tables. This creates a lot of administration traffic on the network and the same traffic re-occurs when that router comes back online. If the problem is something like a faulty network cable that alternates online and offline every 10 seconds, there could easily be an overwhelming amount of routing updates sent out unnecessarily.

Another possible reason for route flap occurs with multiple FortiGate devices in HA mode. When an HA cluster fails over to the secondary unit, other routers on the network may see the HA cluster as being offline, resulting in route flap. While this doesn't occur often, or more than once at a time, it can still result in an interruption in traffic which is disruptive for network users. The easy solution for this problem is to increase the timers on the HA cluster, such as TTL timers, so they don't expire during the failover process. Also, configuring graceful restart on the HA cluster helps with a smooth failover.

The first method of dealing with route flap is to check your hardware. If a cable is loose or bad, it can easily be replaced and eliminate the problem. If an interface on the router is bad, either avoid using that interface or swap in a functioning router. If the power source is bad on a router, either replace the power supply or use a power conditioning backup power supply. These quick and easy fixes can save you from configuring more complex BGP options. However, if the route flap is from another source, configuring BGP to deal with the outages will ensure your network users uninterrupted service.

Some methods of dealing with route flap in BGP include:

- Holdtime timer on page 716
- Dampening on page 717
- Graceful restart on page 718
- BFD on page 719

Holdtime timer

The first step to troubleshooting a flapping route is the holdtime timer. This timer reduces how frequently a route going down will cause a routing update to be broadcast.

Once activated, the holdtime timer won't allow the FortiGate to accept any changes to that route for the duration of the timer. If the route flaps five times during the timer period, only the first outage will be recognized by the FortiGate. For the duration of the other outages, there won't be changes because the Fortigate is essentially treating this router as down. If the route is still flapping after the timer expires, it will start again.

If the route isn't flapping (for example, if it goes down, comes up, and stays back up) the timer will still count down and the route is ignored for the duration of the timer. In this situation, the route is seen as down longer

than it really is but there will be only the one set of route updates. This isn't a problem in normal operation because updates are not frequent.

The potential for a route to be treated as down when it's really up can be viewed as a robustness feature. Typically, you don't want most of your traffic being routed over an unreliable route. So if there's route flap going on, it's best to avoid that route if you can. This is enforced by the holdtime timer.

How to configure the holdtime timer

There are three different route flapping situations that can occur: the route goes up and down frequently, the route goes down and back up once over a long period of time, or the route goes down and stays down for a long period of time. These can all be handled using the holdtime timer.

For example, your network has two routes that you want to set the timer for. One is your main route (to 10.12.101.4) that all of your Internet traffic goes through, and it can't be down for long if it's down. The second is a low speed connection to a custom network that's used infrequently (to 10.13.101.4). The timer for the main route should be fairly short (for example, 60 seconds). The second route timer can be left at the default, since it's rarely used.

To configure the BGP holdtime timer:

```
config router bgp
    config neighbor
        edit 10.12.101.4
            set holdtime-timer 60
            set keepalive-timer 60
        next
        edit 10.13.101.4
            set holdtime-timer 180
            set keepalive-timer 60
        next
    end
end
```

Dampening

Dampening is a method that's used to limit the amount of network problems due to flapping routes. With dampening, the flapping still occurs but the peer routers pay less and less attention to that route as it flaps more often. One flap doesn't start dampening, but the second flap starts a timer where the router won't use that route because it is considered unstable. If the route flaps again before the timer expires, the timer continues to increase. There's a period of time called the reachability half-life, after which a route flap will be suppressed for only half the time. This half-life comes into effect when a route has been stable for a while but not long enough to clear all the dampening completely. For the flapping route to be included in the routing table again, the suppression time must expire.

If the route flapping was temporary, you can clear the flapping or dampening from the FortiGate device's cache by using one of the `execute router clear bgp` CLI commands:

```
# execute router clear bgp dampening {<ip_address> | <ip_address/netmask>}
```

or

```
# execute router clear bgp flap-statistics {<ip_address> | <ip_address/netmask>}
```

For example, to remove route flap dampening information for the 10.10.0.0/16 subnet, enter the following CLI command:

```
# execute router clear bgp dampening 10.10.0.0/16
```

To configure BGP route dampening:

```
config router bgp
    set dampening {enable | disable}
    set dampening6 {enable | disable}
    set dampening-max-suppress-time <minutes_integer>
    set dampening6-max-suppress-time <integer>
    set dampening-reachability-half-life <minutes_integer>
    set dampening6-reachability-half-life <integer>
    set dampening-reuse <reuse_integer>
    set dampening6-reuse <integer>
    set dampening-route-map <routemap-name_str>
    set dampening6-route-map <string>
    set dampening-suppress <limit_integer>
    set dampening6-suppress <integer>
    set dampening-unreachability-half-life <minutes_integer>
    set dampening6-unreachability-half-life <integer>
end
```

Graceful restart

BGP4 has the capability to gracefully restart.

In some situations, route flap is caused by routers that appear to be offline but the hardware portion of the router (control plane) can continue to function normally. One example of this is when some software is restarting or being upgraded but the hardware can still function normally.

Graceful restart is best used for these situations where routing won't be interrupted, but the router is unresponsive to routing update advertisements. Graceful restart doesn't have to be supported by all routers in a network, but the network will benefit when more routers support it.

FortiGate HA clusters can benefit from graceful restart. When a failover takes place, the HA cluster advertises that it is going offline, and will not appear as a route flap. It will also enable the new HA main unit to come online with an updated and usable routing table. If there is a flap, the HA cluster routing table will be out-of-date.

For example, the FortiGate is one of four BGP routers that send updates to each other. Any of those routers may support graceful starting. When a router plans to go offline, it sends a message to its neighbors stating how long it expects to be offline. This way, its neighboring routers don't remove it from their routing tables. However, if that router isn't back online when expected, the routers will mark it offline. This prevents routing flap and its associated problems.

FortiGate devices support both graceful restart of their own BGP routing software and neighboring BGP routers.

To configure BGP graceful restart:

```
config router bgp
    set graceful-restart {disable | enable}
    set graceful-restart-time <seconds_integer>
    set graceful-stalepath-time <seconds_integer>
    set graceful-update-delay <seconds_integer>
    config neighbor
        edit 10.12.101.4
            set capability-graceful-restart {enable | disable}
        next
    end
end
```

Before the restart, the router sends its peers a message to say it's restarting. The peers mark all the restarting router's routes as stale, but they continue to use the routes. The peers assume the router will restart, check its routes, and take care of them, if needed, after the restart is complete. The peers also know what services the restarting router can maintain during its restart. After the router completes the restart, the router sends its peers a message to say it's done restarting.

To restart the router:

```
# execute router restart
```

Scheduled time offline

Graceful restart is a means for a router to advertise that it is going to have a scheduled shutdown for a very short period of time. When neighboring routers receive this notice, they will not remove that router from their routing table until after a set time elapses. During that time, if the router comes back online, everything continues to function as normal. If that router remains offline longer than expected, then the neighboring routers will update their routing tables as they assume that the router will be offline for a long time.

The following example demonstrates if you want to configure graceful restart on the FortiGate where you expect the FortiGate to be offline for no more than two minutes, and after three minutes the BGP network should consider the FortiGate to be offline.

To configure graceful restart time settings:

```
config router bgp
    set graceful-restart enable
    set graceful-restart-time 120
    set graceful-stalepath-time 180
end
```

BFD

Bidirectional Forwarding Detection (BFD) is a protocol that you can use to quickly locate hardware failures in the network. Routers running BFD communicate with each other and if a timer runs out on a connection then that router is declared down. BFD then communicates this information to the routing protocol and the routing information is updated.

For more information about BFD, see BFD on page 720.

BGP path selection process

Sometimes the FortiGate may receive multiple BGP paths from neighbors and must decide which is the best path to take. The following criteria are used to determine the best path.

Consider only routes with no AS loops and a valid next hop, and then:

1. Prefer the highest weight (this attribute is local to the FortiGate).
2. Prefer the highest local preference (applicable within AS).
3. Prefer the route originated by the local router (next hop = 0.0.0.0).
4. Prefer the shortest AS path.
5. Prefer the lowest origin code (IGP > EGP > incomplete).
6. Prefer the lowest MED (exchanged between autonomous systems).
7. Prefer the EBGP path over IBGP path.
8. Prefer the path through the closest IGP neighbor.
9. Prefer the oldest route for EBGP paths.
10. Prefer the path with the lowest neighbor BGP router ID.
11. Prefer the path with the lowest neighbor IP address.

BFD

Bidirectional Forwarding Detection (BFD) is a protocol that you can use to quickly locate hardware failures in the network. Routers running BFD send packets to each other at a negotiated rate. If packets from a BFD-enabled router fail to arrive, that router is declared to be down. BFD communicates this information to the associated routing protocols and the routing information is updated. It helps detect one way device failure and is used for fast convergence of routing protocols.

BFD can run on an entire FortiGate, selected interfaces, or on a protocol, such as BGP, for all configured interfaces. The configuration hierarchy allows each lower level to override the BFD setting of the upper level. For example, if you enable BFD for an entire FortiGate, you can disable BFD for an interface or for BGP.

Echo mode and authentication are not supported for BFD on the FortiGate.

BFD can be enabled per device, VDOM, or interface. Once enabled, a BFD neighbor should be defined. Finally, enable BFD on a route or routing protocol.

To configure BFD for an entire FortiGate:

```
config system settings
    set bfd {enable | disable}
    set bfd-desired-min-tx <ms>
    set bfd-required-min-rx <ms>
    set bfd-detect-mult <multiplier>
    set bfd-dont-enforce-src-port {enable | disable}
end
```

To configure BFD for an interface:

```
config system interface
    edit <interface-name>
        set bfd {global | enable | disable}
        set bfd-desired-min-tx <ms>
        set bfd-required-min-rx <ms>
        set bfd-detect-mult <multiplier>
    next
end
```

To configure static BFD neighbors:

```
config router {bfd | bfd6}
    config neighbor
        edit <IP-address>
            set interface <interface-name>
        next
    end
end
```

See BFD on page 720.

To show BFD neighbors:

```
# get router {info | info6} bfd neighbor
```

To show BFD requests:

```
# get router {info | info6} bfd requests
```

BFD and static routes

BFD for static routes allows you to configure routing failover based on remote path failure detection. BFD removes a static route from the routing table if the FortiGate can't reach the route's destination and returns the route to the routing table if the route's destination is restored.

For example, you can add two static routes with BFD enabled. If one of the routes has a higher priority, all matching traffic uses that route. If BFD determines that the link to the gateway of the route with the higher priority is down, the higher priority route is removed from the routing table and all matching traffic uses the lower priority route. If the link to the gateway for the higher priority route comes back up, BFD adds the route back into the routing table and all matching traffic switches to use the higher priority route.

You can configure BFD for IPv4 and IPv6 static routes.

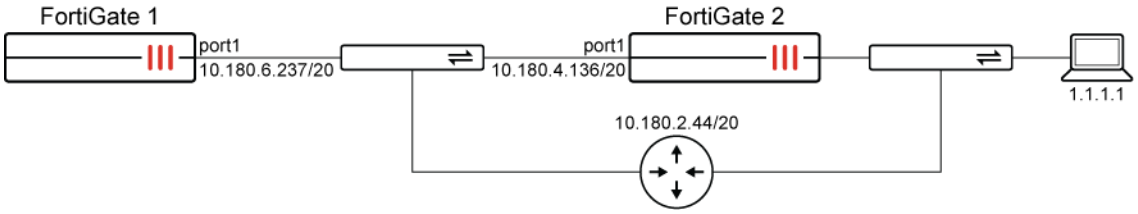
To configure BFD for static routes:

```
config router {static | static6}
    edit <sequence-number>
```

```
        set bfd {enable | disable}
        set device <gateway-out-interface>
    next
end
```

Example

The following example demonstrates the configuration of static routes between two FortiGates. There is a host behind FortiGate 2 with an IP address of 1.1.1.1. FortiGate 1 has multiple paths to reach the host.



To configure static routes:

1. Configure FortiGate 1:

```
config system interface
    edit "port1"
        set vdom "root"
        set ip 10.180.6.237 255.255.240.0
        set allowaccess ping
        set bfd enable
    next
end
config router bfd
    config neighbor
        edit 10.180.4.136
            set interface "port1"
        next
    end
end
```

2. Configure FortiGate 2:

```
config system interface
    edit "port1"
        set vdom "root"
        set ip 10.180.4.136 255.255.240.0
        set allowaccess ping
        set bfd enable
    next
end
config router bfd
    config neighbor
        edit 10.180.6.237
            set interface "port1"
```

```
        next
    end
end
```

> `config router bfd` is not strictly required. BFD will work without statically defining the neighbor.

3. Configure two static routes:

```
config router static
    edit 2
        set dst 1.1.1.1 255.255.255.255
        set gateway 10.180.4.136
        set device "port1"
        set bfd enable
    next
    edit 3
        set dst 1.1.1.1 255.255.255.255
        set gateway 10.180.2.44
        set distance 20
        set device "port1"
    next
end
```

4. Confirm that BFD neighborship is established:

```
# get router info bfd neighbor
OurAddress      NeighAddress    State       Interface       LDesc/RDesc
10.180.6.237    10.180.4.136    UP          port1           1/1
```

5. Review the active route in the routing table:

```
# get router info routing-table all
S       1.1.1.1/32 [10/0] via 10.180.4.136, port1
C       10.180.0.0/20 is directly connected, port1
```

> The route with the lower distance is preferred in the routing table.

If port1 on FortiGate 2 goes down or FortiGate 1 is unable to reach 10.180.4.126, the BFD neighborship will go down.

```
# get router info bfd neighbor
OurAddress      NeighAddress    State       Interface       LDesc/RDesc
10.180.6.237    10.180.4.136    DOWN        port1           1/1
```

With BFD neighborship down, the FortiGate is unable to reach 1.1.1.1/32 through gateway 10.180.4.136. The routing table will be updated so that the route through gateway 10.180.2.44 is active in the routing table.

```
# get router info routing-table all
S       1.1.1.1/32 [20/0] via 10.180.2.44, port1
C       10.180.0.0/20 is directly connected, port1
```

BFD removes a static route from the routing table if the FortiGate cannot reach the route's destination. The static route will be returned to the routing table is the route's destination is restored.

BFD and OSPF

You can configure BFD for Open Shortest Path First (OSPF) on a FortiGate. FortiGate supports BFD for OSPF for both IPv4 and IPv6. BFD must be configured globally and per interface.

To configure BFD for OSPF:

```
config router {ospf | ospf6}
    set bfd {enable | disable}
end
```

To enable BFD on a specific OSPF interface:

```
config router {ospf | ospf6}
    set bfd enable
    config {ospf-interface | ospf6-interface}
        edit <ID>
            set bfd {global | enable | disable}
            set area-id <IP address>
        next
    end
end
```

If BFD is configured when OSPF is not, no BFD packets will be sent. When both BFD and OSFP are configured, the neighbors for both will be the same. Use the following commands to confirm that the neighbor IP addresses match:

```
# get router info ospf neighbor
# get router info bfd neighbor
```

BFD and BGP

While BGP can detect route failures, BFD can be configured to detect these failures more quickly, which allows for faster responses and improved convergence. This can be balanced with the bandwidth BFD uses in its frequent route checking.

The `config router bgp` commands allow you to set the addresses of the neighbor units that are also running BFD. Both units must be configured with BFD in order to use it.

To configure BFD for BGP:

```
config router bgp
    config neighbor
        edit <neighbor-IP-address>
            set bfd {enable | disable}
```

```
        next
    end
end
```

BFD for Multihop paths

FortiGate BFD can support neighbors connected over multiple hops. When BFD is down, BGP sessions will be reset and will try to re-establish neighbor connection immediately. See BFD for multihop path for BGP on page 726 for more information.

To configure BFD for multihop paths:

```
config router {bfd | bfd6}
    config multihop-template
        edit <ID>
            set src <IP address/netmask>
            set dst <IP address/netmask>
            set bfd-desired-min-tx <integer>
            set bfd-required-min-rx <integer>
            set bfd-detect-mult <integer>
            set auth-mode {none | md5}
            set md5-key <password>
        next
    end
end
```

BFD enabled BGP graceful restart

You can add a BFD enabled BGP neighbor as a static BFD neighbor using the following command. This example shows how to add a BFD neighbor with IP address 172.20.121.23 that is on the network connected to port4:

```
config router bfd
    config neighbor
        edit 172.20.121.23
            set port4
        next
    end
end
```

The FGCP supports graceful restart of BFD enabled BGP neighbors. The `config router bfd` command is needed as the BGP auto-start timer is 5 seconds. After HA failover, BGP on the new primary unit has to wait for 5 seconds to connect to its neighbors, and then register BFD requests after establishing the connections. With static BFD neighbors, BFD requests and sessions can be created as soon as possible after the failover. The `get router info bfd requests` command shows the BFD peer requests.

A BFD session created for a static BFD neighbor/peer request will initialize its state as `INIT` instead of `DOWN` and its detection time as `bfd-required-min-rx * bfd-detect-mult milliseconds`.

When a BFD control packet with nonzero `your_discr` is received, if no session can be found to match the `your_discr`, instead of discarding the packet, other fields in the packet, such as addressing information, are used to choose one session that was just initialized, with zero as its remote discriminator.

When a BFD session in the up state receives a control packet with zero as `your_discr` and down as the state, the session will change its state to down, but will not notify this down event to BGP and/or other registered clients.

Troubleshooting BFD

You can troubleshoot BFD using the following commands:

```
# get router {info | info6} bfd neighbor
# get router {info | info6} bfd neighbor detail
# get router {info | info6} bfd requests
# diagnose sniffer packet any <filter> <sniffer count>
# diagnose debug application bfdd <debug level>
# diagnose debug enable
```

For example:

```
# get router info bfd neighbor detail

OurAddress NeighAddress State Interface LDesc/RDesc
10.56.242.195 10.56.242.129 UP port1 1/1
...
Registered protocols: OSPF BGP
```

The possible `Registered protocols` are:

Option	Description
static	static BFD
external	from routes other than existing routing protocol
BGP	from BGP protocol
OSPF	from OSPF protocol
pim-sm	from PIM sparse-mode protocol

BFD for multihop path for BGP

In BFD, a FortiGate can support neighbors connected over multiple hops. When BFD is down, BGP sessions are reset and will try to immediately re-establish neighbor connections. Previously, BFD was only supported when two routers or FortiGates were directly connected on the same network.

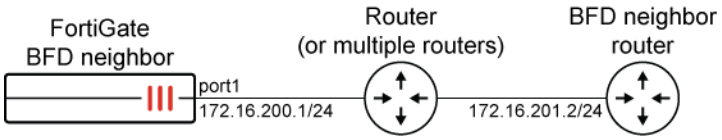
```
config router {bfd | bfd6}
    config multihop-template
        edit <ID>
            set src <class_IP/netmask>
```

```
            set dst <class_IP/netmask>
            set bfd-desired-min-tx <integer>
            set bfd-required-min-rx <integer>
            set bfd-detect-mult <integer>
            set auth-mode {none | md5}
            set md5-key <password>
        next
    end
end
```

src <class_IP/netmask>	Enter the source prefix.
dst <class_IP/netmask>	Enter the destination prefix.
bfd-desired-min-tx <integer>	Set the BFD desired minimal transmit interval, in milliseconds (100 - 30000, default = 250).
bfd-required-min-rx <integer>	Set the BFD required minimal transmit interval, in milliseconds (100 - 30000, default = 250).
bfd-detect-mult <integer>	Set the BFD detection multiplier (3 - 50, default = 3).
auth-mode {none | md5}	Set the authentication mode (none or meticulous MD5).
md5-key <password>	Enter the password.

Example

This example includes IPv4 and IPv6 BFD neighbor configurations. The BFD neighbor is also a BGP neighbor that is in a different AS.



To configure BFD with multihop BGP paths:

1. Enable BFD on all interfaces:

```
config system settings
    set bfd enable
end
```

2. Enable BFD on port1 and ignore the global configuration:

```
config system interface
    edit "port1"
        set bfd enable
    next
end
```

3. Configure the BGP neighbors:

```
config router bgp
    set as 65412
    set router-id 1.1.1.1
    config neighbor
        edit "172.16.201.2"
            set bfd enable
            set ebgp-enforce-multihop enable
            set soft-reconfiguration enable
            set remote-as 65050
        next
        edit "2000:172:16:201::2"
            set bfd enable
            set ebgp-enforce-multihop enable
            set soft-reconfiguration enable
            set remote-as 65050
        next
    end
end
```

4. Configure the IPv4 BFD:

```
config router bfd
    config multihop-template
        edit 1
            set src 172.16.200.0 255.255.255.0
            set dst 172.16.201.0 255.255.255.0
            set auth-mode md5
            set md5-key **********
        next
    end
end
```

5. Configure the IPv6 BFD:

```
config router bfd6
    config multihop-template
        edit 1
            set src 2000:172:16:200::/64
            set dst 2000:172:16:201::/64
        next
    end
end
```

Testing the connection

1. Verify the BFD status for IPv4 and IPv6:

```
# get router info bfd requests
BFD Peer Requests:
    client types(ct in 0x): 01=external 02=static
        04=ospf 08=bgp 10=pim-sm
src=172.16.200.1     dst=172.16.201.2     ct=08 ifi=9 type=SM
```

```
# get router info bfd neighbor
OurAddress      NeighAddress    State       Interface       LDesc/RDesc
172.16.200.1    172.16.201.2    UP          port1           5/3/M
```

```
# get router info6 bfd requests
BFD Peer Requests:
    client types(ct in 0x): 01=external 02=static
        04=ospf 08=bgp 10=pim-sm
src=2000:172:16:200::1
dst=2000:172:16:201::2
ct=08 ifi=9 type=SM
```

```
# get router info6 bfd neighbor
OurAddress: 2000:172:16:200::1
NeighAddress: 2000:172:16:201::2
State: UP Interface: port1 Desc: 6/4 Multi-hop
```

2. Verify the BGP status and the BGP routing table:

```
# get router info bgp summary
VRF 0 BGP router identifier 1.1.1.1, local AS number 65412
BGP table version is 11
3 BGP AS-PATH entries
0 BGP community entries

Neighbor           V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
172.16.201.2       4      65050     185     187       10    0    0 00:54:20        4
2000:172:16:201::2 4      65050     159     160       10    0    0 00:54:24        4

Total number of neighbors 2
```

```
# get router info routing-table bgp
Routing table for VRF=0
B       172.28.1.0/24 [20/0] via 172.16.201.2 (recursive via 172.16.200.4, port1), 00:54:32
B       172.28.2.0/24 [20/0] via 172.16.201.2 (recursive via 172.16.200.4, port1), 00:54:32
B       172.28.5.0/24 [20/0] via 172.16.201.2 (recursive via 172.16.200.4, port1), 00:54:32
B       172.28.6.0/24 [20/0] via 172.16.201.2 (recursive via 172.16.200.4, port1), 00:54:32
```

```
# get router info6 bgp summary
VRF 0 BGP router identifier 1.1.1.1, local AS number 65412
BGP table version is 8
3 BGP AS-PATH entries
0 BGP community entries

Neighbor           V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
172.16.201.2       4      65050     185     187        7    0    0 00:54:24        3
2000:172:16:201::2 4      65050     159     160        7    0    0 00:54:28        3

Total number of neighbors 2
```

```
# get router info6 routing-table bgp
Routing table for VRF=0
B       2000:172:28:1::/64 [20/0] via 2000:172:16:201::2 (recursive via 2000:172:16:200::4,
port1), 00:54:40
B       2000:172:28:2::/64 [20/0] via 2000:172:16:201::2 (recursive via 2000:172:16:200::4,
port1), 00:54:40
B       2000:172:28:3::/64 [20/0] via 2000:172:16:201::2 (recursive via 2000:172:16:200::4,
port1), 00:54:40
```

3. Simulate a disruption to the BFD connection. The BFD neighbor is lost:

```
# get router info bfd neighbor
OurAddress      NeighAddress    State       Interface       LDesc/RDesc
```

```
# get router info6 bfd neighbor
```

4. The BGP neighbor is reset, and the FortiGate attempts to re-establish a connection with the neighbor. The timers are reset once the neighbor connection is re-established:

```
# get router info bgp summary
VRF 0 BGP router identifier 1.1.1.1, local AS number 65412
BGP table version is 12
4 BGP AS-PATH entries
0 BGP community entries

Neighbor           V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
172.16.201.2       4      65050     189     192       11    0    0 00:00:11        4
2000:172:16:201::2 4      65050     165     167       12    0    0 00:00:08        4

Total number of neighbors 2
```

```
# get router info6 bgp summary
VRF 0 BGP router identifier 1.1.1.1, local AS number 65412
BGP table version is 10
4 BGP AS-PATH entries
0 BGP community entries

Neighbor           V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
172.16.201.2       4      65050     189     192        8    0    0 00:00:15        3
2000:172:16:201::2 4      65050     165     167        9    0    0 00:00:12        3

Total number of neighbors 2
```

5. The BGP routes are learned again, and there are new timers in the route tables:

```
# get router info routing-table bgp
Routing table for VRF=0
B       172.28.1.0/24 [20/0] via 172.16.201.2 (recursive via 172.16.200.4, port1), 00:00:15
B       172.28.2.0/24 [20/0] via 172.16.201.2 (recursive via 172.16.200.4, port1), 00:00:15
B       172.28.5.0/24 [20/0] via 172.16.201.2 (recursive via 172.16.200.4, port1), 00:00:15
B       172.28.6.0/24 [20/0] via 172.16.201.2 (recursive via 172.16.200.4, port1), 00:00:15
```

```
# get router info6 routing-table bgp
Routing table for VRF=0
B       2000:172:28:1::/64 [20/0] via 2000:172:16:201::2 (recursive via 2000:172:16:200::4,
port1), 00:00:13
B       2000:172:28:2::/64 [20/0] via 2000:172:16:201::2 (recursive via 2000:172:16:200::4,
port1), 00:00:13
B       2000:172:28:3::/64 [20/0] via 2000:172:16:201::2 (recursive via 2000:172:16:200::4,
port1), 00:00:13
```

Routing objects

The following objects can be configured from the *Network* > *Routing Objects* page:

- Route maps on page 731
- Access lists on page 734
- Prefix lists on page 737
- AS path lists on page 740
- Community lists on page 741

Route maps

Route maps are a powerful tool to apply custom actions to dynamic routing protocols based on specific conditions. They are used primarily in BGP to manipulate routes advertised by the FortiGate (`route-map-out`) or received routes from other BGP routers (`route-map-in`).

Route maps can be used in OSPF for conditional `default-information-originate`, filtering external routes, or matching specific routes for redistribution. Similarly, route maps can be used by RIP to match routes for redistribution.

A route map may have multiple rules that are processed in the ascending order of their rule ID numbers. The rule ID number determines the order of evaluation, where each rule has an action to permit or deny. If the action is not set, the default action is to permit.

When new are added to an existing route map, they may be shown at the end of the configuration list, regardless of their assigned rule ID. This visual arrangement does not impact functionality; FortiOS processes the rules according to their numerical IDs, and not their position in the route map configuration. For example, if a new rule with ID 15 is added to a route map that already contains rules with IDs 10, 20, 30, and 40, the new rule might appear at the bottom of the list in the configuration. Despite this placement, FortiOS will process the rules in the ascending order of Rule ID number: 10, 15, 20, 30, then 40.

To enhance readability and maintain an organized configuration, it is recommended to arrange the rules sequentially by their IDs. This can be achieved by deleting and re-adding the rules in the desired order during a maintenance window to avoid traffic disruption. Alternatively, using a text editor to reorder the rules before applying them to the FortiOS configuration can streamline this process.

The rules have criteria for matching a route based on various attributes, or setting attributes based on a matched route. For example, a route map can be used to match BGP routes with a certain community string, and then set an AS path to the matching route. This can be applied to a BGP neighbor by configuring the route map in the settings for that neighbor.

To configure a route map that matches criteria based on other routing objects:

```
config router route-map
    edit <name>
        config rule
            edit <id>
                set action {permit | deny}
                set match-as-path <string>
                set match-community <string>
                set match-ip-address <string>
                set match-ip6-address <string>
                set match-ip-nexthop <string>
                set match-ip6-nexthop <string>
            next
        end
    next
end
```

Option	Description
match-as-path <string>	Match a BGP AS path list.
match-community <string>	Match a BGP community list.
match-ip-address <string>	Match an IPv4 address permitted by access-list or prefix-list.
match-ip6-address <string>	Match an IPv6 address permitted by access-list6 or prefix-list6.
match-ip-nexthop <string>	Match a next hop IPv4 address passed by access-list or prefix-list.
match-ip6-nexthop <string>	Match a next hop IPv6 address passed by access-list6 or prefix-list6.

Route maps can be used by various routing protocols, such as RIP, OSPF, and BGP.

To use a route map with RIP:

```
config router rip
    config redistribute
        edit <name>
            set routemap <string>
        next
    end
end
```

To use a route map with OSPF:

```
config router ospf
    set default-information-route-map <string>
    set distribute-route-map-in <string>
    config redistribute <string>
        set routemap <string>
    end
end
```

Option	Description
`default-information-route-map <string>`	Enter the default information route map.
`distribute-route-map-in <string>`	Enter the route map to filter incoming external routes.
`redistribute <string>`	Configure the redistribute protocol.

To use a route map with BGP:

```
config router bgp
    config neighbor
        edit <ip>
            set route-map-in <string>
            set route-map-in6 <string>
            set route-map-in-vpnv4 <string>
            set route-map-out <string>
            set route-map-out-preferable <string>
            set route-map-out6 <string>
            set route-map-out6-preferable <string>
            set route-map-out-vpnv4 <string>
            set route-map-out-vpnv4-preferable <string>
        next
    end
    config network
        edit <id>
            set prefix <IP/netmask>
            set route-map <string>
        next
    end
    config redistribute <string>
        set route-map <string>
    end
end
```

Option	Description
`route-map-in <string>`	Enter the IPv4 inbound route map filter.
`route-map-in6 <string>`	Enter the IPv6 inbound route map filter.
`route-map-in-vpnv4 <string>`	Enter the VPNv4 inbound route map filter.
`route-map-out <string>`	Enter the IPv4 outbound route map filter.
`route-map-out-preferable <string>`	Enter the IPv4 outbound route map filter if the peer is preferred.
`route-map-out6 <string>`	Enter the IPv6 outbound route map filter.
`route-map-out6-preferable <string>`	Enter the IPv6 outbound route map filter if the peer is preferred.

Option	Description
`route-map-out-vpnv4 <string>`	Enter the VPNv4 outbound route map filter.
`route-map-out-vpnv4-preferable <string>`	Enter the VPNv4 outbound route map filter if the peer is preferred.
`route-map <string>`	Enter the route map to modify the generated route.
`redistribute <string>`	Configure the redistribute protocol.

To use a route map with BGP conditional advertisement:

```
config router bgp
    set as <AS_number>
    config neighbor
        edit <ip>
            set remote-as <AS_number>
            config conditional-advertise
                edit <advertise-routemap>
                    set condition-routemap <name1>, <name2>, ...
                    set condition-type {exist | non-exist}
                next
            end
        next
    end
end
```

Option	Description
`<advertise-routemap>`	Edit the advertising route map.
`condition-routemap <name1>, <name2>, ...`	Enter the list of conditional route maps.

Access lists

Access lists are simple lists used for filtering routes based on a prefix consisting of an IPv4 or IPv6 address and netmask.

To configure an IPv4 access list:

```
config router access-list
    edit <name>
        config rule
            edit <id>
                set action {permit | deny}
                set prefix <IPv4_address>
                set wildcard <wildcard_filter>
                set exact-match {enable | disable}
            next
```

```
        end
    next
end
```

To configure an IPv6 access list:

```
config router access-list6
    edit <name>
        config rule
            edit <id>
                set action {permit | deny}
                set prefix <IPv6_address>
                set exact-match {enable | disable}
            next
        end
    next
end
```

In RIP, an access list can be used in the `distribute-list` setting to filter received or advertised routes, or in an `offset-list` to offset the hop count metric for a specific prefix.

To use an access list in RIP:

```
config router rip
    config distribute-list
        edit <id>
            set direction {in | out}
            set listname <string>
        next
    end
    config offset-list
        edit <id>
            set direction {in | out}
            set access-list <string>
            set offset <integer>
        next
    end
end
```

Option	Description
`listname <string>`	Enter the distribute access or prefix list name.
`access-list <string>`	Enter the access list name.

In OSPF, an access list can be used in the `distribute-list-in` setting to act as a filter to prevent a certain route from being inserted into the routing table. An access list can also be used in the `distribute-list` to filter the routes that can be distributed from other protocols.

To use an access list in OSPF:

```
config router ospf
    set distribute-list-in <string>
    config distribute-list
        edit <id>
            set access-list <string>
            set protocol {connected | static | rip}
        next
    end
end
```

Option	Description
distribute-list-in <string>	Enter the filter for incoming routes.
access-list <string>	Enter the access list name.

In BGP, an access list can be used to filter updates from a neighbor or to a neighbor.

To use an access list in BGP:

```
config router bgp
    config neighbor
        edit <ip>
            set distribute-list-in <string>
            set distribute-list-in6 <string>
            set distribute-list-in-vpnv4 <string>
            set distribute-list-out <string>
            set distribute-list-out6 <string>
            set distribute-list-out-vpnv4 <string>
        next
    end
end
```

Option	Description
distribute-list-in <string>	Enter the filter for IPv4 updates from this neighbor.
distribute-list-in6 <string>	Enter the filter for IPv6 updates from this neighbor.
distribute-list-in-vpnv4 <string>	Enter the filter for VPNv4 updates from this neighbor.
distribute-list-out <string>	Enter the filter for IPv4 updates to this neighbor.
distribute-list-out6 <string>	Enter the filter for IPv6 updates to this neighbor.
distribute-list-out-vpnv4 <string>	Enter the filter for VPNv4 updates to this neighbor.

In a route map, an access list can be used to match IP addresses and next hops.

To use an access list in a route map:

```
config router route-map
    edit <name>
        config rule
            edit <id>
                set match-ip-address <string>
                set match-ip6-address <string>
                set match-ip-nexthop <string>
                set match-ip6-nexthop <string>
            next
        end
    next
end
```

Option	Description
`match-ip-address <string>`	Match an IPv4 address permitted by `access-list` or `prefix-list`.
`match-ip6-address <string>`	Match an IPv6 address permitted by `access-list6` or `prefix-list6`.
`match-ip-nexthop <string>`	Match a next hop IPv4 address passed by `access-list` or `prefix-list`.
`match-ip6-nexthop <string>`	Match a next hop IPv6 address passed by `access-list6` or `prefix-list6`.

Prefix lists

Similar to access lists, prefix lists are simple lists used for filtering routes based on a prefix consisting of an IPv4 or IPv6 address and netmask, but they use settings to specify the minimum (`ge`, greater than or equal) and maximum (`le`, less than or equal) prefix length to be matched. For example, a prefix of 10.0.0.0/8 with a `ge` of 16 will match anything in the 10.0.0.0/8 network with /16 or above; 10.10.0.0/16 will match, and 10.10.0.0/12 will not match.

To configure an IPv4 prefix list:

```
config router prefix-list
    edit "prefix-list1"
        config rule
            edit 1
                set action {permit | deny}
                set prefix <IPv4_address>
                set ge <integer>
                set le <integer>
            next
        end
    next
end
```

To configure an IPv6 prefix list:

```
config router prefix-list6
    edit "prefix-list-IPv6"
        config rule
            edit 1
                set action {permit | deny}
                set prefix6 <IPv6_address>
                set ge <integer>
                set le <integer>
            next
        end
    next
end
```

In RIP, an prefix list can be used in the `distribute-list` setting to filter received or advertised routes.

To use a prefix list in RIP:

```
config router rip
    config distribute-list
        edit <id>
            set listname <string>
        next
    end
end
```

Option	Description
`listname <string>`	Enter the distribute access or prefix list name.

In OSPF, a prefix list can be used in the `distribute-list-in` setting to act as a filter to prevent a certain route from being inserted into the routing table.

To use a prefix list in OSPF:

```
config router ospf
    set distribute-list-in <string>
end
```

Option	Description
`distribute-list-in <string>`	Enter the filter for incoming routes.

In BGP, a prefix list can be used to filter updates from a neighbor or to a neighbor.

To use a prefix list in BGP:

```
config router bgp
    config neighbor
```

```
        edit <ip>
            set prefix-list-in <string>
            set prefix-list-in6 <string>
            set prefix-list-in-vpnv4 <string>
            set prefix-list-out <string>
            set prefix-list-out6 <string>
            set prefix-list-out-vpnv4 <string>
        next
    end
end
```

Option	Description
`prefix-list-in <string>`	Enter the IPv4 inbound filter for updates from this neighbor.
`prefix-list-in6 <string>`	Enter the IPv6 inbound filter for updates from this neighbor.
`prefix-list-in-vpnv4 <string>`	Enter the inbound filter for VPNv4 updates from this neighbor.
`prefix-list-out <string>`	Enter the IPv4 outbound filter for updates to this neighbor.
`prefix-list-out6 <string>`	Enter the IPv6 outbound filter for updates to this neighbor.
`prefix-list-out-vpnv4 <string>`	Enter the outbound filter for VPNv4 updates to this neighbor.

In a route map, a prefix list can be used to match IP addresses and next hops.

To use a prefix list in a route map:

```
config router route-map
    edit <name>
        config rule
            edit <id>
                set match-ip-address <string>
                set match-ip6-address <string>
                set match-ip-nexthop <string>
                set match-ip6-nexthop <string>
            next
        end
    next
end
```

Option	Description
`match-ip-address <string>`	Match an IPv4 address permitted by `access-list` or `prefix-list`.
`match-ip6-address <string>`	Match an IPv6 address permitted by `access-list6` or `prefix-list6`.
`match-ip-nexthop <string>`	Match a next hop IPv4 address passed by `access-list` or `prefix-list`.
`match-ip6-nexthop <string>`	Match a next hop IPv6 address passed by `access-list6` or `prefix-list6`.

AS path lists

AS path lists use regular expressions to compare and match the AS_PATH attribute for a BGP route. They can be used to filter inbound or outbound routes from a BGP neighbor, or as matching criteria in a route map to match an AS_PATH in a BGP route.

To configure an AS path list:

```
config router aspath-list
    edit <name>
        config rule
            edit <id>
                set action {deny | permit}
                set regexp <string>
            next
        end
    next
end
```

To use an AS path list in BGP:

```
config router bgp
    config neighbor
        edit <ip>
            set filter-list-in <string>
            set filter-list-in6 <string>
            set filter-list-out <string>
            set filter-list-out6 <string>
        next
    end
end
```

Option	Description
`filter-list-in <string>`	Enter the BGP filter for IPv4 inbound routes.
`filter-list-in6 <string>`	Enter the BGP filter for IPv6 inbound routes.
`filter-list-out <string>`	Enter the BGP filter for IPv4 outbound routes.
`filter-list-out6 <string>`	Enter the BGP filter for IPv6 outbound routes.

To use an AS path list in a route map:

```
config router route-map
    edit <name>
        config rule
            edit <id>
                set match-as-path <string>
            next
        end
```

```
    next
end
```

Option	Description
match-as-path <string>	Match a BGP AS path list.

Community lists

Community lists provide a means to filter BGP routes using a community string. They can be applied in a route map to match routes that have the community string defined in the community list.

To configure a community list:

```
config router community-list
    edit <name>
        set type {standard | expanded}
        config rule
            edit <id>
                set action {deny | permit}
                set regexp <string>
                set match <string>
            next
        end
    next
end
```

To use a community list in a route map to match a BGP community:

```
config router route-map
    edit <name>
        config rule
            edit <id>
                set match-community <string>
            next
        end
    next
end
```

Option	Description
match-community <string>	Match a BGP community list.

In an SD-WAN deployment, a remote BGP router or spoke may communicate a preferred interface or path to route traffic using a community string. See Using BGP tags with SD-WAN rules on page 1434 and Controlling traffic with BGP route mapping and service rules on page 1440 for examples.

Multicast

The following topics include information about multicast:

- Multicast routing and PIM support on page 742
- Configuring multicast forwarding on page 748
- Using IPS inspection for multicast UDP traffic on page 754
- Including denied multicast sessions in the session table on page 757
- IPv6 multicast routing BSR support New on page 759

Multicast routing and PIM support

Multicasting (also called IP multicasting) consists of using a single multicast source to send data to many receivers. Multicasting can be used to send data to many receivers simultaneously while conserving bandwidth and reducing network traffic. Multicasting can be used for one-way delivery of media streams to multiple receivers and for one-way data transmission for news feeds, financial information, and so on. Many dynamic routing protocols such as RIPv2, OSPF, and EIGRP use multicasting to share hello packets and routing information.

A FortiGate can operate as a Protocol Independent Multicast (PIM) version 2 router. FortiGates support PIM sparse mode (RFC 4601) and PIM dense mode (RFC 3973), and can service multicast servers or receivers on the network segment to which a FortiGate interface is connected. Groups are supported in the PIM join/prune messages. Multicast routing is not supported in transparent mode.

To support PIM communications, the sending and receiving applications, and all connecting PIM routers in between, must be enabled with PIM version 2. PIM can use static routes, RIP, OSPF, or BGP to forward multicast packets to their destinations. To enable source-to-destination packet delivery, sparse mode or dense mode must be enabled on the PIM router interfaces. Sparse mode routers cannot send multicast messages to dense mode routers. If the FortiGate is located between a source and a PIM router, between two PIM routers, or is connected directly to a receiver, you must manually create a multicast policy to pass encapsulated (multicast) packets or decapsulated data (IP traffic) between the source and destination.

PIM domains

A PIM domain is a logical area comprising a number of contiguous networks. The domain contains at least one bootstrap router (BSR), and if sparse mode is enabled, a number of rendezvous points (RPs) and designated routers (DRs). When PIM is enabled, the FortiGate can perform any of these functions at any time as configured.

A PIM domain can be configured in the GUI by going to *Network* > *Multicast*, or in the CLI using `config router multicast`. Note that PIM version 2 must be enabled on all participating routers between the source and receivers. Use `config router multicast` to set the global operating parameters.

When PIM is enabled, the FortiGate allocates memory to manage mapping information. The FortiGate communicates with neighboring PIM routers to acquire mapping information and, if required, processes the multicast traffic associated with specific multicast groups.

Instead of sending multiple copies of generated IP traffic to more than one specific IP destination address, PIM-enabled routers encapsulate the data and use a Class D multicast group address (224.0.0.0 to 239.255.255.255) to forward multicast packets to multiple destinations. A single stream of data can be sent because one destination address is used. Client applications receive multicast data by requesting that the traffic destined for a certain multicast group address be delivered to them.

PIM support for virtual routing and forwarding (VRF)

PIM supports all VRFs (up to 511) and is aware of IPv4 and IPv6 multicast routing and forwarding over a single overlay, enhancing network scalability and flexibility compared to the previous VRF 0-only support. See Virtual routing and forwarding on page 791 for more information.

Per-VRF commands are included for multicast routing or it can be configured in *Network > Multicast*.

To configure VRF settings for multicast routing in the CLI:

```
config router {multicast | multicast6}
    config pim-sm-global-vrf
        edit <id>
            set bsr-candidate {enable | disable}
            set bsr-interface <string>
            set bsr-priority <integer>
            set bsr-hash <integer>
            set bsr-allow-quick-refresh {enable | disable}
            set cisco-crp-prefix {enable | disable}
            config rp-address
                edit <id>
                    set *ip6-address <IPv6 addr>
                    set group <string>
                next
            end
        next
    end
end
```

Command or option	Description
`config pim-sm-global-vrf`	Per-VRF PIM sparse-mode global settings.
`bsr-candidate {enable | disable}`	Enable/disable allowing this router to become a bootstrap router (BSR).
`bsr-interface <string>`	Interface to advertise as candidate BSR.
`bsr-priority <integer>`	BSR priority (0 - 255, default = 0).
`bsr-hash <integer>`	BSR hash length (0 - 32, default = 10).
`bsr-allow-quick-refresh {enable | disable}`	Enable/disable accept BSR quick refresh packets from neighbors.
`cisco-crp-prefix {enable | disable}`	Enable/disable making candidate RP compatible with old Cisco IOS: • `enable`: Do not allow sending group prefix of zero.

Command or option	Description
	• `disable`: Allow sending group prefix of zero.
`config rp-address`	Statically configure RP addresses.
`*ip6-address <IPv6 addr>`	RP router IPv6 address.
`group <string>`	Groups to use this RP.

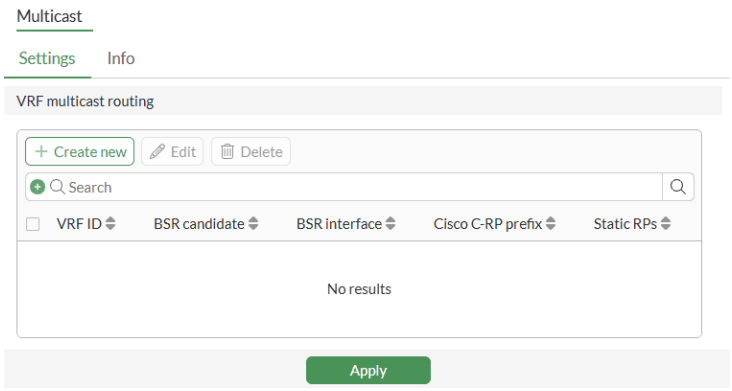
VRF support is also included in the following `diagnose`, `get`, and `execute` commands:

```
diagnose ip multicast mfc-add
diagnose ip multicast mfc-del
diagnose vpn {mr | mr6} add
diagnose vpn {mr | mr6} del
get router info multicast igmp groups
get router info multicast igmp groups-detail
get router info multicast pim sparse-mode bsr-info
get router info multicast pim sparse-mode next-hop
get router info multicast pim sparse-mode rp-mapping
get router info multicast pim sparse-mode table
get router info multicast table
get router info multicast table-count
get router info6 multicast mld groups
get router info6 multicast mld groups-detail
get router info6 multicast pim sparse-mode bsr-router
get router info6 multicast pim sparse-mode mroute
get router info6 multicast pim sparse-mode nexthop
get router info6 multicast pim sparse-mode rp mapping
get router info6 multicast pim sparse-mode rp-hash
execute mrouter clear multicast-routes
execute mrouter clear sparse-mode-bsr
execute mrouter clear sparse-routes
execute mrouter clear statistics
execute mrouter6 clear mld group
execute mrouter6 clear sparse-routes
```

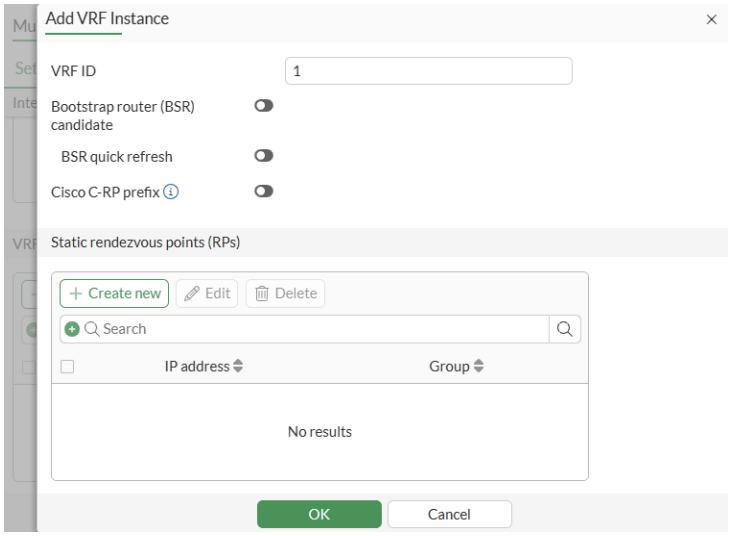
NPU offloading of VRF multicast traffic on a dynamic IPsec tunnel is not supported.

To configure VRF settings for multicast routing in the GUI:

1. Go to *Network* > *Multicast*.
2. Enable *Multicast routing* if you are configuring multicast for the first time in the GUI.
3. Under *VRF multicast routing*, click *Create new*.



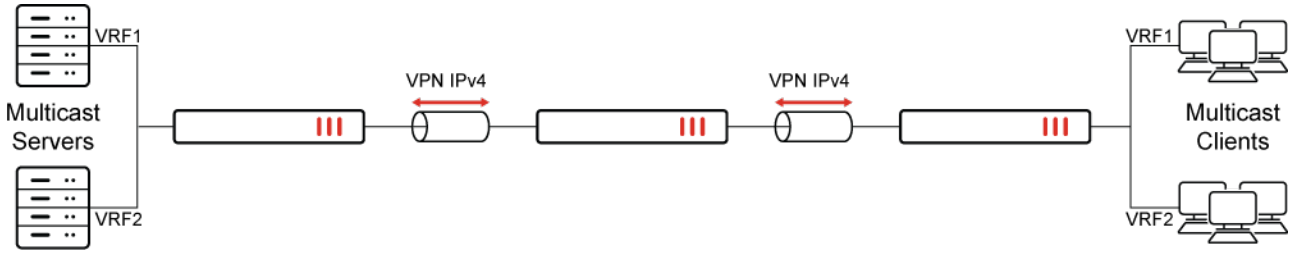
4. Configure the settings as needed.



5. Click *OK*.
6. Click *Apply*.

IPv4 example

This example uses the following topology:



In this example, the multicast server:

- Sends out multicast traffic 225.1.1.1 from 22.1.1.22 in VRF1.
- Sends out multicast traffic 225.1.1.2 from 22.1.1.55 in VRF2.

To verify VRF in IPv4 multicast routing:

1. Review the sniffer information:
   - The VRF1 client can receive 225.1.1.1 and cannot receive 225.1.1.2:

```
24.872130 vd33-vlan33 in 22.1.1.22 -> 225.1.1.1: icmp: echo request
25.872117 vd3-vlan33 out 22.1.1.22 -> 225.1.1.1: icmp: echo request
25.872123 vd33-vlan33 in 22.1.1.22 -> 225.1.1.1: icmp: echo request
26.872131 vd3-vlan33 out 22.1.1.22 -> 225.1.1.1: icmp: echo request
26.872137 vd33-vlan33 in 22.1.1.22 -> 225.1.1.1: icmp: echo request
```

   - The VRF2 client can receive 225.1.1.2 and cannot receive 225.1.1.1:

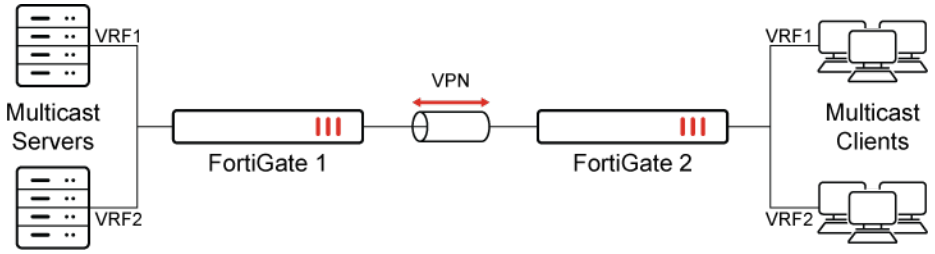
```
4.320988 vd3-vlan331 out 22.1.1.55 -> 225.1.1.2: icmp: echo request
4.320996 vd4-vlan331 in 22.1.1.55 -> 225.1.1.2: icmp: echo request
5.320703 vd3-vlan331 out 22.1.1.55 -> 225.1.1.2: icmp: echo request
5.320717 vd4-vlan331 in 22.1.1.55 -> 225.1.1.2: icmp: echo request
6.320671 vd3-vlan331 out 22.1.1.55 -> 225.1.1.2: icmp: echo request
6.320678 vd4-vlan331 in 22.1.1.55 -> 225.1.1.2: icmp: echo request
```

2. Review the group information:

```
# get router info multicast igmp groups
IGMP Connected Group Membership
VRF Group Address    Interface            Uptime   Expires          Last Reporter
1   225.1.1.1        vd3-vlan33           00:15:16 stopped(static)  0.0.0.0
2   225.1.1.2        vd3-vlan331          00:14:49 stopped(static)  0.0.0.0
```

IPv6 example NEW

This example uses the following topology:



In this example, the multicast server:

- Sends out multicast traffic ff0e::10 from 2000:172:16:205::11 in VRF1.
- Sends out multicast traffic ff0e::20 from 2000:172:16:205::22 in VRF2.

To configure IPv6 multicast-flow:

```
config router multicast6-flow
    edit "flow6a"
        config flows
            edit 1
                set group-addr ff0e::10
```

```
            next
        end
    next
    edit "flow6b"
        config flows
            edit 1
                set group-addr ff0e::20
            next
        end
    next
end
```

To configure PIM-SM global settings for each VRF:

```
config router multicast6
    set multicast-routing enable
    config interface
        edit "vd3-1"
        next
        edit "vd3-vlan33"
        next
        edit "vd3-vlan331"
        next
    end
    config pim-sm-global-vrf
        edit 2
            config rp-address
                edit 1
                    set ip6-address 2011::12
                next
            end
        next
        edit 1
            config rp-address
                edit 1
                    set ip6-address 2011::11
                next
            end
        next
    end
end
```

For brevity, only the commands relevant to this feature are included.

To verify VRF in IPv6 multicast routing:

1. Review the sniffer information:
   - The VRF1 client can receive ff0e::10 and cannot receive ff0e::20:

```
0.800190 npu0_vlink0 out 2000:172:16:205::11.33372 -> ff0e::10.5001: udp 46 [flowlabel
0xa620f]
0.800197 vd33-vlan33 in 2000:172:16:205::11.33372 -> ff0e::10.5001: udp 46 [flowlabel
0xa620f]
```

   - The VRF2 client can receive ff0e::20 and cannot receive ff0e::10:

```
0.666623 vd3-vlan331 out 2000:172:16:205::22.33372 -> ff0e::20.5001: udp 46 [flowlabel
0xa620f
0.666630 vd3-vlan331 out 2000:172:16:205::22.33372 -> ff0e::20.5001: udp 46 [flowlabel
0xa620f
```

2. Review the group information:

```
# get router info6 multicast mld groups
MLD Connected Group Membership
VRF Group Address                          Interface            Uptime   Expires  Last
Reporter
1   ff0e::10                               vd3-vlan33           00:21:05 now(static) ::
2   ff0e::20                               vd3-vlan331          00:21:05 now(static) ::
```

Configuring multicast forwarding

There is sometimes confusion between the terms forwarding and routing. These two functions should not take place at the same time. Multicast forwarding should be enabled when the FortiGate is in NAT mode and you want to forward multicast packets between multicast routers and receivers. However, this function should not be enabled when the FortiGate itself is operating as a multicast router, or has an applicable routing protocol that uses multicast.

Multicast forwarding is not supported on enhanced MAC VLAN interfaces. To use multicast with enhanced MAC VLAN interfaces, use PIM (Multicast routing and PIM support on page 742).

There are two steps to configure multicast forwarding:

1. Enabling multicast forwarding on page 748
2. Configuring multicast policies on page 749

Enabling multicast forwarding

Multicast forwarding is enabled by default. If a FortiGate is operating in transparent mode, adding a multicast policy enables multicast forwarding. In NAT mode you must use the `multicast-forward` setting to enable or disable multicast forwarding.

Multicast forwarding in NAT mode

When `multicast-forward` is enabled, the FortiGate forwards any multicast IP packets in which the TTL is 2 or higher to all interfaces and VLAN interfaces, except the receiving interface. The TTL in the IP header will be reduced by 1. Even though the multicast packets are forwarded to all interfaces, you must add multicast policies to allow multicast packets through the FortiGate.

To enable multicast forwarding in NAT mode:

```
config system settings
    set multicast-forward enable
end
```

Prevent the TTL for forwarded packets from being changed

You can use the `multicast-ttl-notchange` option so that the FortiGate does not increase the TTL value for forwarded multicast packets. Use this option only if packets are expiring before reaching the multicast router.

To prevent the TTL for forwarded packets from being changed:

```
config system settings
    set multicast-ttl-notchange enable
end
```

Disable multicast traffic from passing through the FortiGate without a policy check in transparent mode

In transparent mode, the FortiGate does not forward frames with multicast destination addresses. The FortiGate should not interfere with the multicast traffic used by routing protocols, streaming media, or other multicast communication. To avoid any issues during transmission, you can disable `multicast-skip-policy` and configure multicast security policies.

To disable multicast traffic from passing through the FortiGate without a policy check in transparent mode:

```
config system settings
    set multicast-skip-policy disable
end
```

Configuring multicast policies

Multicast packets require multicast policies to allow packets to pass from one interface to another. Similar to firewall policies, in a multicast policy you specify the source and destination interfaces, and the allowed address ranges for the source and destination addresses of the packets. You can also use multicast policies to configure source NAT and destination NAT for multicast packets.

Keep the following in mind when configuring multicast policies:

- The matched forwarded (outgoing) IP multicast source IP address is changed to the configured IP address.
- The `snat` setting is optional. Use it when SNAT is needed.

IPv4 and IPv6 multicast policies can be configured in the GUI. Go to *System > Feature Visibility*, and enable *Multicast Policy* and *IPv6*.

Sample basic policy

In this basic policy, multicast packets received on an interface are flooded unconditionally to all interfaces on the forwarding domain, except the incoming interface.

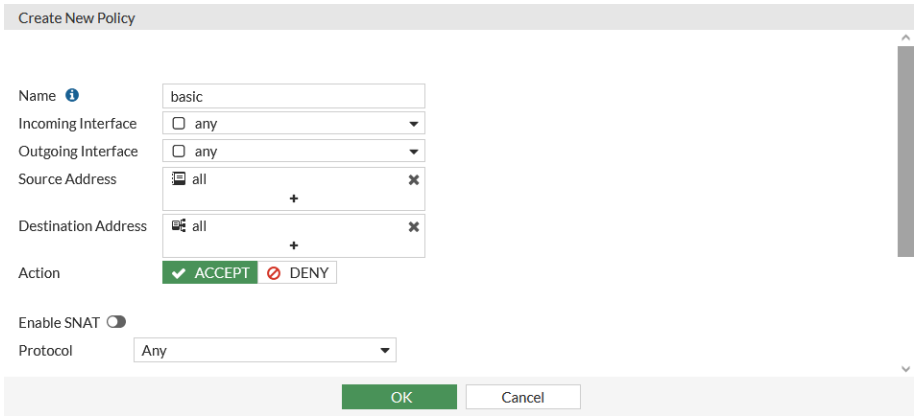
The destination address (`dstaddr`) is a multicast address object. The `all` option corresponds to all multicast addresses in the range 224.0.0.0-239.255.255.255.

To configure the multicast policy in the CLI:

```
config firewall multicast-policy
    edit 1
        set name "basic"
        set srcintf "any"
        set dstintf "any"
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

To configure the multicast policy in the GUI:

1. Go to *Policy & Objects > Multicast Policy* an click *Create New*.
2. Enter the required information:



Name	basic
Incoming Interface	any
Outgoing Interface	any

Source Address	all
Destination Address	all

3. Click *OK*.

Sample policy with specific source and destination interfaces

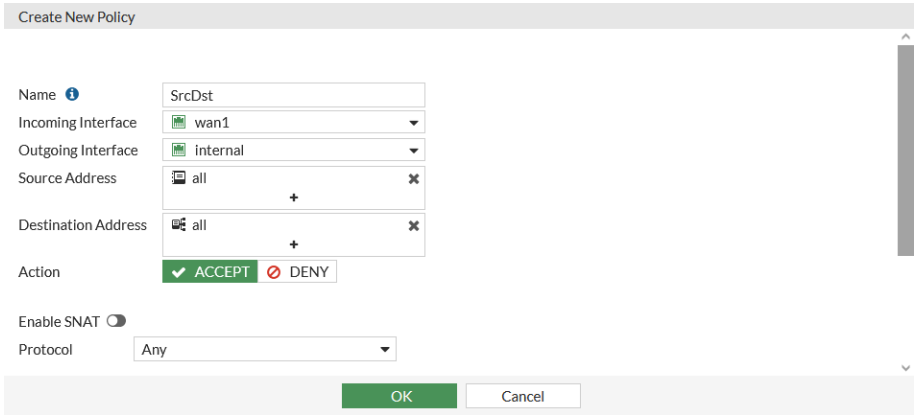
This multicast policy only applies to the source port `wan1` and the destination port `internal`.

To configure the multicast policy in the CLI:

```
config firewall multicast-policy
    edit 1
        set name "SrcDst"
        set srcintf "wan1"
        set dstintf "internal"
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

To configure the multicast policy in the GUI:

1. Go to *Policy & Objects* > *Multicast Policy* an click *Create New*.
2. Enter the required information:



Name	SrcDst
Incoming Interface	wan1
Outgoing Interface	internal
Source Address	all
Destination Address	all

3. Click *OK*.

Sample policy with specific source address object

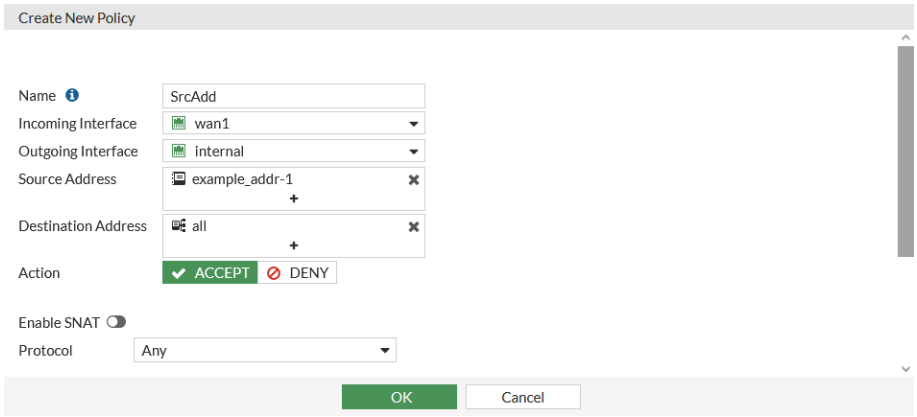
In this policy, packets are allowed to flow from `wan1` to `internal`, and sourced by the address 172.20.120.129, which is represented by the `example_addr-1` address object.

To configure the multicast policy in the CLI:

```
config firewall multicast-policy
    edit 1
        set name "SrcAdd"
        set srcintf "wan1"
        set dstintf "internal"
        set srcaddr "example_addr-1"
        set dstaddr "all"
    next
end
```

To configure the multicast policy in the GUI:

1. Go to *Policy & Objects* > *Multicast Policy* an click *Create New*.
2. Enter the required information:



Name	SrcAdd
Incoming Interface	wan1
Outgoing Interface	internal
Source Address	example_addr-1
Destination Address	all

3. Click *OK*.

Sample detailed policy

This policy accepts multicast packets that are sent from a PC with IP address 192.168.5.18 to destination address range 239.168.4.0-255. The policy allows the multicast packets to enter the `internal` interface and

then exit the `external` interface. When the packets leave the external interface, their source address is translated to 192.168.18.10.

```
config firewall address
    edit "192.168.5.18"
        set subnet 192.168.5.18 255.255.255.255
    next
end
```

```
config firewall multicast-address
    edit "239.168.4.0"
        set start-ip 239.168.4.0
        set end-ip 239.168.4.255
    next
end
```

```
config firewall multicast-policy
    edit 1
        set srcintf "internal"
        set dstintf "external"
        set srcaddr "192.168.5.18"
        set dstaddr "239.168.4.0"
        set snat enable
        set snat-ip 192.168.18.10
    next
end
```

To configure multicast policies in the GUI, enable *Multicast Policy* in *System > Feature Visibility*.

Using multi-VDOM mode

When using multi-VDOM mode, it is important to avoid causing a multicast network loop by creating an all-to-all multicast policy. By default, on models that support NPU virtual links, changing the `vdom-mode` to `multi-vdom` will create a pair of npu0_vlink0 and npu0_vlink1 interfaces in the same root VDOM. By virtue of the all-to-all multicast policy and the fact the npu0_vlink interfaces are virtually connected, it forms a multicast network loop.

Therefore, when using multi-VDOM mode:

1. Ensure there is no existing all-to-all multicast policy before changing to multi-VDOM mode.
2. If an all-to-all multicast policy must be defined, ensure that no two connected interfaces (such as npu0_vlink0 and npu0_vlink1) belong in the same VDOM.

This configuration will result in a multicast loop:

```
config system global
    set vdom-mode multi-vdom
end
config firewall multicast-policy
    edit 1
```

```
        set logtraffic enable
        set srcintf "any"
        set dstintf "any"
        set srcaddr "all"
        set dstaddr "all"
    next
end
show system interface
config system interface
    edit "npu0_vlink0"
        set vdom "root"
        set type physical
    next
    edit "npu0_vlink1"
        set vdom "root"
        set type physical
    next
end
```

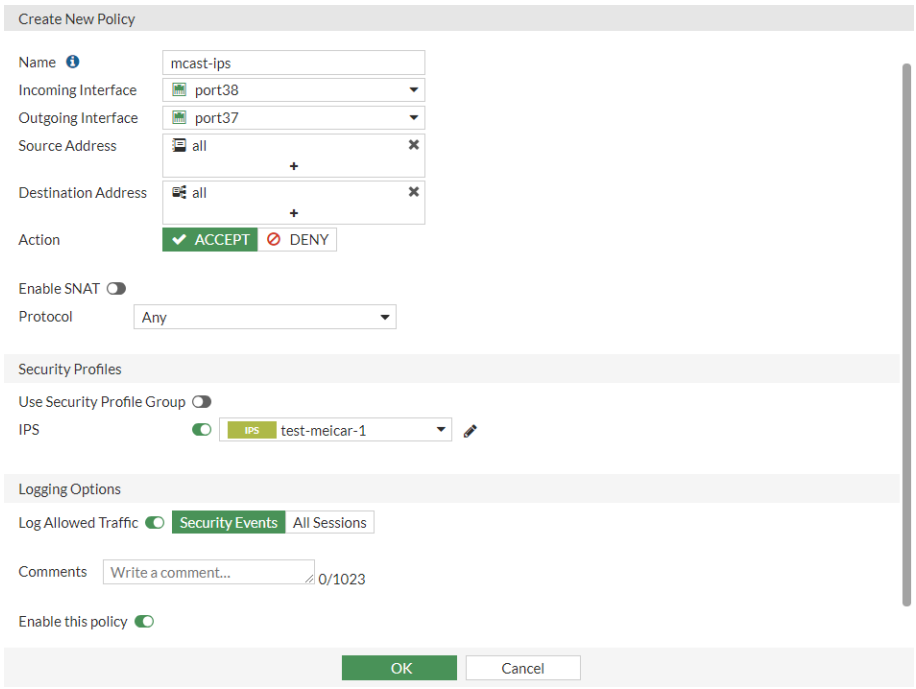
Using IPS inspection for multicast UDP traffic

IPS inspection can be applied for multicast UDP traffic in multicast firewall policies.

```
config firewall {multicast-policy | multicast-policy6}
    edit <id>
        set utm-status {enable | disable}
        set ips-sensor <name>
        set logtraffic {all | utm | disable}
    next
end
```

IPv4 and IPv6 multicast policies can be configured in the GUI. Go to *System > Feature Visibility*, and enable *Multicast Policy* and *IPv6*.

The multicast policy dialog page (*Policy & Objects > Multicast Policy*) includes a *Security Profiles* section where you can enable *IPS* and apply an IPS profile.



Example

In this example, an IPv4 multicast policy is configured with IPS inspection enabled. Multicast UDP traffic that contains IPS attacks is detected and blocked. A custom IPS signature is created with an infected EICAR pattern for the UDP protocol.

To use IPS inspection for multicast UDP traffic:

1. Configure the IPS custom signature:

```
config ips custom
    edit "meicar"
        set signature "F-SBID( --name \"meicar\"; --attack_id 9999; --protocol udp; --severity
medium; --default_action clear_session; --pattern \"$EICAR-STANDARD-ANTIVIRUS-TEST-FILE\";)"
        set protocol UDP
        set log disable
        set action block
    next
end
```

2. Configure the IPS sensor:

```
config ips sensor
    edit "test-meicar-1"
        config entries
            edit 1
                set rule 9999
                set status enable
```

```
                set action block
            next
        end
    next
end
```

3. Configure the multicast policy:

```
config firewall multicast-policy
    edit 1
        set srcintf "port38"
        set dstintf "port37"
        set srcaddr "all"
        set dstaddr "all"
        set utm-status enable
        set ips-sensor "test-meicar-1"
    next
end
```

4. Add the server to the multicast group 239.1.1.10 and join it using a terminal:

```
fosqa@ips_pc5:~$ iperf -s -u -B 239.1.1.10 -i 1
------------------------------------------------------------
Server listening on UDP port 5001
Binding to local address 239.1.1.10
Joining multicast group  239.1.1.10
Receiving 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 239.1.1.10 port 5001 connected with 10.1.100.11 port 52972
```

5. From a terminal on the client, send multicast UDP traffic with the EICAR file:

```
root@PC01:~# iperf -c 239.1.1.10 -u -T 3 -t 20 -i 1 -F eicar
------------------------------------------------------------
Client connecting to 239.1.1.10, UDP port 5001
Sending 1470 byte datagrams, IPG target: 11215.21 us (kalman adjust)
Setting multicast TTL to 3
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  4] local 10.1.100.11 port 33383 connected with 239.1.1.10 port 5001
[ ID] Interval       Transfer     Bandwidth
[  4]  0.0- 0.0 sec  1.44 KBytes  1.03 Mbits/sec
[  4] Sent 1 datagrams
```

The traffic will be blocked, and the server will not be able to receive the packets.

6. Verify that the traffic is blocked.
   a. Verify the IPS event log:

```
# execute log filter category 4
# execute log display
1 logs found.
```

```
1 logs returned.

1: date=2023-11-01 time=17:01:43 eventtime=1698883303178500916 tz="-0700"
logid="0419016384" type="utm" subtype="ips" eventtype="signature" level="alert" vd="vd1"
severity="medium" srcip=10.1.100.11 srccountry="Reserved" dstip=239.1.1.10
dstcountry="Reserved" srcintf="port38" srcintfrole="undefined" dstintf="port37"
dstintfrole="undefined" sessionid=18 action="dropped" proto=17 service="udp/5001"
policyid=1 poluuid="09bdd086-78e2-51ee-1d61-0955f9046b53" policytype="multicast-policy"
attack="meicar" srcport=52673 dstport=5001 direction="outgoing" attackid=9999
profile="test-meicar-1" incidentserialno=245366798 msg="custom: meicar" crscore=10
craction=16384 crlevel="medium"
```

b. Verify the IPS traffic log:

```
# execute log filter category 0
# execute log display
1 logs found.
1 logs returned.

1: date=2023-11-01 time=17:04:39 eventtime=1698883474200006380 tz="-0700"
logid="0002000012" type="traffic" subtype="multicast" level="notice" vd="vd1"
srcip=10.1.100.11 srcport=52673 srcintf="port38" srcintfrole="undefined" dstip=239.1.1.10
dstport=5001 dstintf="port37" dstintfrole="undefined" srccountry="Reserved"
dstcountry="Reserved" sessionid=18 proto=17 action="accept" policyid=1
policytype="multicast-policy" poluuid="09bdd086-78e2-51ee-1d61-0955f9046b53"
policyname="mcast-ips" service="udp/5001" trandisp="noop" duration=180 sentbyte=2996
rcvdbyte=0 sentpkt=2 rcvdpkt=0 appcat="unscanned" utmref=0-266
```

c. Verify the multicast session list:

```
# diagnose sys mcast-session list

session info: id=19 vf=1 proto=17 10.1.100.11.56538->239.1.1.10.5001
used=2 path=1 duration=2 expire=177 indev=10
state=00000000:
session-npu-info: ipid/vlifid=0/0 vlanid/vtag_in=0/0 in_npuid=0 tae_index=0 qid=0 fwd_
map=0x00000000
path: log ndr policy=1, outdev=9, tos=0xff
Total 1 sessions
```

Including denied multicast sessions in the session table

Sessions can be created for denied multicast traffic, enabling subsequent packets to be directly matched and dropped, reducing CPU usage and improving performance.

To configure denied multicast session inclusion:

```
config system setting
    set ses-denied-multicast-traffic {disable | enable}
end
```

Value	Description
disable	Do not add denied multicast sessions to the session table (default).
enable	Include denied multicast sessions in the session table.

Example

In this example, denied multicast sessions are included in the session table of the VDOM. A deny multicast policy is created and a packet is then sent that hits the policy. Checking the multicast session list shows that a denied multicast session is created.

To configure and test including denied multicast sessions:

1. Enable including denied multicast sessions:

```
config system setting
    set ses-denied-multicast-traffic enable
end
```

2. Create a deny multicast policy in the multicast policy table:

```
config firewall multicast-policy
    edit 1
        set name "Deny_Multicast_Policy"
        set srcintf "port1"
        set dstintf "port3"
        set srcaddr "172-16-200-0"
        set dstaddr "230-0-0-1"
        set action deny
        set logtraffic all
        set auto-asic-offload disable
    next
end
```

3. Send packets to hit the deny multicast policy then check the multicast session list. The second session shown is the denied multicast session:

```
# diagnose sys mcast-session list

session info: id=259 vf=1 proto=17 172.16.200.55.34896->230.0.0.10.7878
used=2 path=1 duration=8 expire=174 indev=9 pkts=4 bytes=2160
state=00000000:
session-npu-info: ipid/vlifid=0/0 vlanid/vtag_in=0/0 in_npuid=0 tae_index=0 qid=0 fwd_
map=0x00000000
path: log npu-deny policy=2, outdev=11, tos=0xff

session info: id=260 vf=1 proto=17 172.16.200.55.33488->230.0.0.1.7878
used=2 path=0 duration=6 expire=177 indev=9 pkts=5 bytes=2700
state=00000200: deny
session-npu-info: ipid/vlifid=0/0 vlanid/vtag_in=0/0 in_npuid=0 tae_index=0 qid=0 fwd_
```

```
map=0x00000000
Total 2 sessions
```

IPv6 multicast routing BSR support NEW

FortiGate supports BSR (Bootstrap Router) for IPv6 multicast, enabling dynamic RP discovery and aligning IPv6 multicast capabilities with existing IPv4 PIM-SM deployments. BSR is a mechanism in PIM-SM that automates the discovery and advertisement of Rendezvous Points (RPs), simplifying configuration and improving scalability for large-scale multicast environments.

Commands to support multicast flows for IPv6 to be used for static-group in PIM6 interface:

```
config router multicast6-flow
    edit <name>
        config flows
            edit <id>
                set *group-addr <IPv6 addr>
            next
        end
    next
end
```

Command or option	Description
`config router multicast6-flow`	Configure IPv6 multicast-flow.
`config flows`	IPv6 multicast-flow entries.
`id`	Flow ID.
`*group-addr <IPv6 addr>`	Multicast group IP address.

Commands to support PIM6 RP configurations for PIM6 interface:

```
config router multicast6
    config interface
        edit <name>
            set rp-candidate {enable | disable}
            set static-group <string>
            set rp-candidate-group <string>
            set rp-candidate-priority <value>
            set rp-candidate-interval <integer>
        next
    end
end
```

Command or option	Description
`config interface`	Protocol Independent Multicast (PIM) interfaces.

Command or option	Description
`rp-candidate {enable | disable}`	Enable/disable compete to become RP in elections: • `enable`: Compete for RP elections. • `disable`: Do not compete for RP elections.
`static-group <string>`	Statically set IPv6 multicast groups to forward out.
`rp-candidate-group <string>`	Multicast groups managed by this RP.
`rp-candidate-priority <value>`	Router's priority as RP (0 - 255, default = 192).
`rp-candidate-interval <integer>`	RP candidate advertisement interval (1 - 16383 sec, default = 60).

Commands to support PIM6 BSR configurations without VRF:

```
config router multicast6
    config pim-sm-global
        set bsr-candidate {enable | disable}
        set bsr-interface <string>
        set bsr-priority <integer>
        set bsr-hash <integer>
        set bsr-allow-quick-refresh {enable | disable}
        set cisco-crp-prefix {enable | disable}
        set cisco-ignore-rp-set-priority {enable | disable}
        set spt-threshold {enable | disable}
        set spt-threshould-group <string>
        config rp-address
            edit <id>
                set *ip6-address <IPv6 addr>
                set group <string>
            next
        end
    end
end
```

Command or option	Description
`config pim-sm-global`	PIM sparse-mode global settings.
`bsr-candidate {enable | disable}`	Enable/disable allowing this router to become a bootstrap router (BSR).
`bsr-interface <string>`	Interface to advertise as candidate BSR.
`bsr-priority <integer>`	BSR priority (0 - 255, default = 0).
`bsr-hash <integer>`	BSR hash length (0 - 32, default = 10).
`bsr-allow-quick-refresh {enable | disable}`	Enable/disable accept BSR quick refresh packets from neighbors.
`cisco-crp-prefix {enable | disable}`	Enable/disable making candidate RP compatible with old Cisco IOS: • `enable`: Do not allow sending group prefix of zero.

Command or option	Description
	• `disable`: Allow sending group prefix of zero.
`cisco-ignore-rp-set-priority {enable | disable}`	Use only hash for RP selection (compatibility with old Cisco IOS): • `enable`: Ignore RP-SET priority value. • `disable`: Do not ignore RP-SET priority value.
`spt-threshold {enable | disable}`	Enable/disable switching to source specific tree: • `enable`:Switch to Source tree when available. • `disable`: Do not switch to Source tree when available.
`spt-threshould-group <string>`	Groups allowed to switch to source tree.
`config rp-address`	Statically configure RP addresses.
`*ip6-address <IPv6 addr>`	RP router IPv6 address.
`group <string>`	Groups to use this RP.

Commands to support PIM6 BSR configurations per VRF:

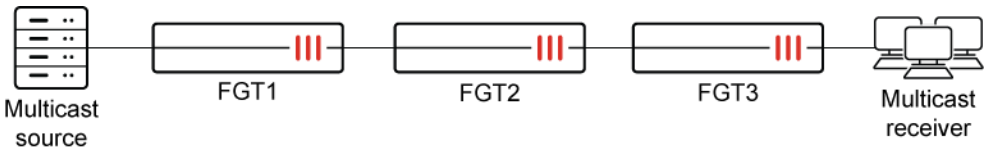
```
config router multicast6
    config pim-sm-global-vrf
        edit <id>
            set bsr-candidate {enable | disable}
            set bsr-interface <string>
            set bsr-priority <integer>
            set bsr-hash <integer>
            set bsr-allow-quick-refresh {enable | disable}
            set cisco-crp-prefix {enable | disable}
            config rp-address
                edit <id>
                    set *ip6-address <IPv6 addr>
                    set group <string>
                next
            end
        next
    end
end
```

Command or option	Description
`config pim-sm-global-vrf`	Per-VRF PIM sparse-mode global settings.
`bsr-candidate {enable | disable}`	Enable/disable allowing this router to become a bootstrap router (BSR).
`bsr-interface <string>`	Interface to advertise as candidate BSR.
`bsr-priority <integer>`	BSR priority (0 - 255, default = 0).
`bsr-hash <integer>`	BSR hash length (0 - 32, default = 10).
`bsr-allow-quick-refresh {enable | disable}`	Enable/disable accept BSR quick refresh packets

Command or option	Description
	from neighbors.
`cisco-crp-prefix {enable | disable}`	Enable/disable making candidate RP compatible with old Cisco IOS: • `enable`: Do not allow sending group prefix of zero. • `disable`: Allow sending group prefix of zero.
`config rp-address`	Statically configure RP addresses.
`*ip6-address <IPv6 addr>`	RP router IPv6 address.
`group <string>`	Groups to use this RP.

Example

This example about large-scale deployments shows how BSR (Bootstrap Router) can be used to dynamically learn the RP info and avoid mis-configurations.



In this example, three FortiGates are configured to support IPv6 multicast routing using PIM-SM. FGT1 and FGT2 are configured as RP and BSR candidates for IPv6 multicast groups, while port4 on FGT3 is connected to a multicast receiver:

- FGT1:

```
config router access-list6
    edit "list6"
        config rule
            edit 1
                set prefix6 ff05::10/128
            next
        end
    next
    edit "list11"
        config rule
            edit 1
                set prefix6 ff05::11/128
            next
        end
    next
end
```

```
config router multicast6
    set multicast-routing enable
    config interface
        edit "port1"
            set rp-candidate enable
```

```
            set rp-candidate-group "list11"
        next
        edit "port2"
            set rp-candidate enable
            set rp-candidate-group "list6"
            set rp-candidate-priority 200
        next
        edit "fortilink"
        next
    end
    config pim-sm-global
        set bsr-candidate enable
        set bsr-interface "port1"
    end
end
```

- FGT2

```
config router access-list6
    edit "list6"
        config rule
            edit 1
                set prefix6 ff05::10/128
            next
        end
    next
end
```

```
config router multicast6
    set multicast-routing enable
    config interface
        edit "port1"
            set rp-candidate enable
            set rp-candidate-group "list6"
        next
        edit "npu0_vlink0"
        next
    end
    config pim-sm-global
        set bsr-candidate enable
        set bsr-interface "port1"
    end
end
```

- FGT3

```
config router multicast6
    set multicast-routing enable
    config interface
        edit "npu0_vlink1"
        next
        edit "port4"
```

```
        next
    end
end
```

Verification of BSR Election and RP Mapping in IPv6 PIM Sparse Mode Across FortiGates:

- FGT1:

```
# get router info6 multicast pim sparse-mode bsr-router
PIMv2 Bootstrap information for VRF 0
  BSR address: 2000:172:16:200::4
  Uptime:      02:02:12, BSR Priority: 0, Hash mask length: 126
  Expires:     00:01:12
  Role: Candidate BSR
  State: Candidate BSR

  Candidate RP: 2000:172:16:205::1(port2)
    Advertisement interval 60 seconds
    Next Cand_RP_advertisement in 00:00:15

  Candidate RP: 2000:172:16:200::1(port1)
    Advertisement interval 60 seconds
    Next Cand_RP_advertisement in 00:00:02
```

```
# get router info6 multicast pim sparse-mode rp mapping
PIM Group-to-RP Mappings for VRF 0
Group(s): ff05::10/128
  RP: 2000:172:16:200::4
    Info source: 2000:172:16:200::4, via bootstrap, priority 192
         Uptime: 00:54:11, expires: 00:02:19
  RP: 2000:172:16:205::1
    Info source: 2000:172:16:200::4, via bootstrap, priority 200
         Uptime: 00:55:18, expires: 00:02:19
Group(s): ff05::11/128
  RP: 2000:172:16:200::1
    Info source: 2000:172:16:200::4, via bootstrap, priority 192
         Uptime: 00:55:18, expires: 00:02:19
```

```
# get router info6 multicast pim sparse-mode rp-hash ff05::11
    VRF 0 RP: 2000:172:16:200::1
    Info source: 2000:172:16:200::4, via bootstrap
```

```
# get router info6 multicast pim sparse-mode rp-hash ff05::10
    VRF 0 RP: 2000:172:16:200::4
    Info source: 2000:172:16:200::4, via bootstrap
```

```
# get router info6 multicast pim sparse-mode nexthop
Flags: N = New, R = RP, S = Source, U = Unreachable
VRF Destination          Type  Nexthop   Nexthop              Nexthop  Nexthop Metric Pref
Refcnt
                               Num     Addr                 Ifindex  Name
_____________________________________________________________________________________
```

```
  0 2000:172:16:200::1   .R..  1         2000:172:16:200::1   7               0      0     1
  0 2000:172:16:200::4   .RS.  1         2000:172:16:200::4   7               0      0     3
  0 2000:172:16:205::11  ..S.  1         2000:172:16:205::11  8               0      0     1
```

```
# get router info6 multicast pim sparse-mode mroute
IP Multicast Routing Table

(*,*,RP) Entries: 0
(*,G) Entries: 1
(S,G) Entries: 1
(S,G,rpt) Entries: 0
FCR Entries: 0

VRF 0 (2000:172:16:205::11, ff05::10)
RPF nbr: ::
RPF idx: None
SPT bit: 1
Upstream State: JOINED
 Local:
 Joined:
     port1
 Asserted:
 Outgoing:
     port1

VRF 0 (*, ff05::11)
RP: 2000:172:16:200::1
RPF nbr: ::
RPF idx: None
Upstream State: JOINED
 Local:
 Joined:
     port1
 Asserted:
FCR:
```

- FGT2

```
# get router info6 multicast pim sparse-mode bsr-router
PIMv2 Bootstrap information for VRF 0
This system is the Bootstrap Router (BSR)
  BSR address: 2000:172:16:200::4
  Uptime:      00:55:37, BSR Priority: 0, Hash mask length: 126
  Next bootstrap message in 00:00:28
  Role: Candidate BSR
  State: Elected BSR

  Candidate RP: 2000:172:16:200::4(port1)
    Advertisement interval 60 seconds
    Next Cand_RP_advertisement in 00:00:25
```

```
# get router info6 multicast pim sparse-mode rp mapping
PIM Group-to-RP Mappings for VRF 0
This system is the Bootstrap Router (v2)
Group(s): ff05::10/128
  RP: 2000:172:16:200::4
    Info source: 2000:172:16:200::4, via bootstrap, priority 192
         Uptime: 00:55:51, expires: 00:01:39
  RP: 2000:172:16:205::1
    Info source: 2000:172:16:205::1, via bootstrap, priority 200
         Uptime: 00:57:36, expires: 00:01:54
Group(s): ff05::11/128
  RP: 2000:172:16:200::1
    Info source: 2000:172:16:200::1, via bootstrap, priority 192
         Uptime: 00:57:49, expires: 00:01:41
```

- FGT3

```
# get router info6 multicast pim sparse-mode bsr-router
PIMv2 Bootstrap information for VRF 0
  BSR address: 2000:172:16:200::4
  Uptime:      02:04:31, BSR Priority: 0, Hash mask length: 126
  Expires:     00:01:53
  Role: Non-candidate BSR
  State: Accept Preferred
```

```
# get router info6 multicast pim sparse-mode rp mapping
PIM Group-to-RP Mappings for VRF 0
Group(s): ff05::10/128
  RP: 2000:172:16:200::4
    Info source: 2000:172:16:200::4, via bootstrap, priority 192
         Uptime: 00:56:21, expires: 00:02:09
  RP: 2000:172:16:205::1
    Info source: 2000:172:16:200::4, via bootstrap, priority 200
         Uptime: 00:57:28, expires: 00:02:09
Group(s): ff05::11/128
  RP: 2000:172:16:200::1
    Info source: 2000:172:16:200::4, via bootstrap, priority 192
         Uptime: 00:57:28, expires: 00:02:09
```

FortiExtender

Two configuration modes are available on FortiGate for FortiExtender integration: WAN extension mode and LAN extension mode.

For information about configuring FortiExtender, see the FortiExtender Admin Guide (FGT-Managed) and Admin Guide (Standalone).

WAN extension mode

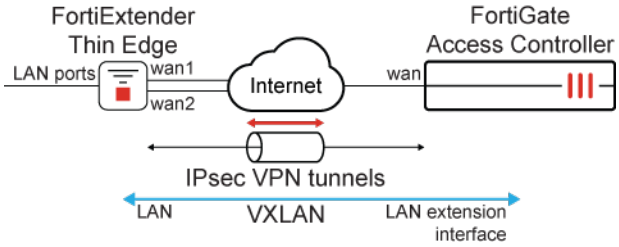
In WAN extension mode, the FortiExtender works as an extended WAN interface in IP pass-through mode. The FortiGate manages FortiExtender over the CAPWAP protocol in IP pass-through mode, and is integrated into FortiOS as a manageable interface.

Sample configurations in WAN extension mode could include connecting a FortiExtender to two FortiGates in HA active-passive mode, or connecting two FortiExtenders to two FortiGates in HA active-active mode to provide dual active redundancy for wireless WAN access.

For more information, see FortiExtender and FortiGate integration in the FortiExtender (Managed) Administration Guide.

LAN extension mode

The LAN extension configuration mode allows FortiExtender to provide remote thin edge connectivity back to the FortiGate over a backhaul connection. A FortiExtender deployed at a remote location will discover the FortiGate access controller (AC) and form an IPsec tunnel (or multiple tunnels when multiple links exist on the FortiExtender) back to the FortiGate. A VXLAN is established over the IPsec tunnels to create an L2 network between the FortiGate and the network behind the remote FortiExtender.



For more information, see FortiExtender as FortiGate LAN extension in the FortiExtender (Managed) Administration Guide.

Maximum FortiExtender devices supported per mode

FortiOS supports a maximum number of FortiExtender devices in WAN or LAN extension mode; see FortiExtender and FortiGate integration for details.

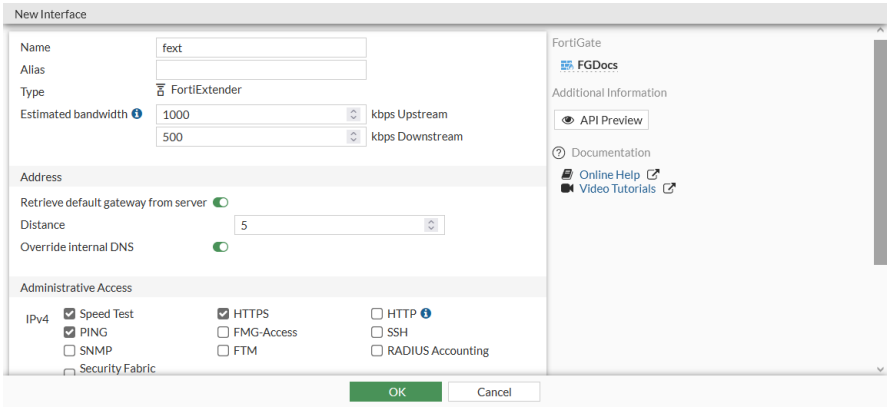
Adding a FortiExtender

To add a FortiExtender to the FortiGate, create a virtual FortiExtender interface, then add a FortiExtender and assign the interface to the modem. Like other interface types, the FortiExtender interface can be used in static

routes, SD-WAN (see Manage dual FortiExtender devices), policies, and other functions.

To create a virtual FortiExtender interface in the GUI:

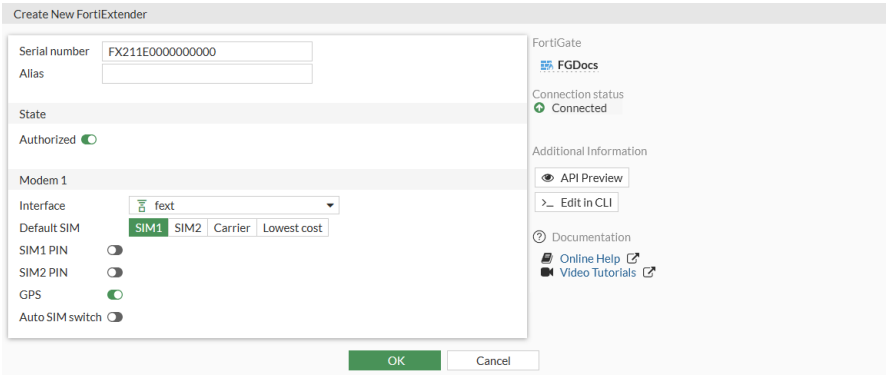
1. Go to *Network* > *Interfaces* and click *Create New* > *FortiExtender*.
2. Enter a name for the interface.
3. Configure the remaining settings as needed. See Interface settings on page 225 for more details.



4. Click *OK*.

To add a FortiExtender in the GUI:

1. Go to *Network* > *FortiExtender* and click *Create New* > *Extenders*.
2. Enter your FortiExtender's serial number in the *Serial number* field.
3. Optionally, set an *Alias* for the FortiExtender.
4. In the *State* section, enable *Authorized*.
5. Set *Interface* to the FortiExtender interface.
6. Configure the remaining setting as required. See the FortiExtender Administration Guide (FGT-Managed) for more information.



7. Click *OK*.
8. In the extenders list, right-click on the FortiExtender and select *Diagnostics and Tools* to review the modem and SIM status, and other details about the FortiExtender.

To create a virtual FortiExtender interface in the CLI:

```
config system interface
    edit "fext"
        set vdom "root"
        set mode dhcp
        set allowaccess ping https speed-test
        set type fext-wan
        set role wan
        set snmp-index 18
        set estimated-upstream-bandwidth 1000
        set estimated-downstream-bandwidth 500
    next
end
```

To configure the FortiExtender in the CLI:

```
config extension-controller extender
    edit "FX211E"
        set authorized enable
        set extension-type wan-extension
        set profile <profile>
        config wan-extension
            set modem1-extension "fext"
        end
    next
end
```

The `device-id` is automatically configured by the FortiGate.

To verify the modem settings in the CLI:

```
get extender modem-status FX211E0000000000 1
Modem 0:
    physical_port:       2-1.2
    manufacture:         Sierra Wireless, Incorporated
    product:             Sierra Wireless, Incorporated
    ....
```

Cellular modems

The following topics include information about configuring and using cellular modems:

- Automatic LTE connection establishment on page 770
- Direct IP support for LTE/4G on page 770
- Cellular interface support for IPv6 on page 773

- Active SIM card switching on page 776
- Airplane mode and LTE/BLE on page 784
- Upgrade LTE modem firmware directly from FortiGuard on page 786

Automatic LTE connection establishment

Establishing an LTE connection is now automated. When you insert a SIM card into FortiGate, FortiOS can obtain the Mobile Country Code (MCC) and Mobile Network Code (MNC) from the service provider's radio tower. FortiOS then uses the codes to look up the appropriate APN for the SIM card in a predefined table and automatically creates a wireless profile. Manual configuration is no longer needed, which simplifies the process of establishing an LTE connection.

Direct IP support for LTE/4G

Direct IP is a public IP address that is assigned to a computing device, which allows the device to directly access the internet.

When an LTE modem is enabled in FortiOS, a DHCP interface is created. As a result, the FortiGate can acquire direct IP (which includes IP, DNS, and gateway) from the LTE network carrier.

Since some LTE modems require users to input the access point name (APN) for the LTE network, the LTE modem configuration allows you to set the APN.

LTE modems can only be enabled by using the CLI.

DHCP relay packet flow through the modem interface can be enabled from the CLI using the `dhcp-relay` command.

```
config system lte-modem
    set dhcp-relay enable
end
```

To enable direct IP support using the CLI:

1. Enable the LTE modem:

```
config system lte-modem
  set status enable
end
```

2. Check that the LTE interface was created:

```
config system interface
  edit "wwan"
    set vdom "root"
    set mode dhcp
```

```
    set status down
    set distance 1
    set type physical
    set snmp-index 23
  next
end
```

Shortly after the LTE modem joins its carrier network, `wwan` is enabled and granted direct IP:

```
config system interface
    edit wwan
        get
name                : wwan
....
ip                  : 100.112.75.43 255.255.255.248
....
status              : up
....
defaultgw           : enable
DHCP Gateway        : 100.112.75.41
Lease Expires       : Thu Feb 21 19:33:27 2019
dns-server-override : enable
Acquired DNS1       : 184.151.118.254
Acquired DNS2       : 70.28.245.227
....
```

PCs can reach the internet via the following firewall policy:

```
config firewall policy
  edit 5
    set name "LTE"
    set srcintf "port9"
    set dstintf "wwan"
    set srcaddr "all"
    set dstaddr "all"
    set action accept
    set schedule "always"
    set service "ALL"
    set utm-status enable
    set fsso disable
    set nat enable
  next
end
```

Sample LTE interface

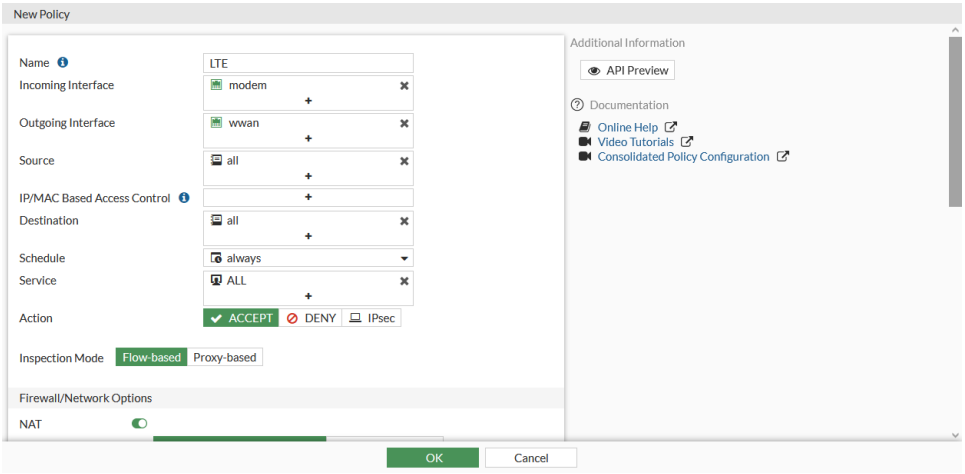
When an LTE modem is enabled, you can view the LTE interface in the GUI and check the acquired IP, DNS, and gateway.

To view the LTE interface in the GUI:

1. Go to *Network* > *Interfaces*.
2. Double-click the LTE interface (*wwan*) to view the properties.
3. Look in the *Address* section to see the *Obtained IP/Netmask*, *Acquired DNS*, and *Default Gateway*.
4. Click *Return*.

To configure the firewall policy that uses the LTE interface:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Edit the LTE policy.
3. In the *Outgoing Interface* field, select the interface (`wwan` in this example).
4. Configure the rest of the policy as needed.



5. Click *OK*.

Limitations

- Most LTE modems have a preset APN in their SIM card. Therefore, the APN does not need to be set in the FortiOS configuration. In cases where the internet cannot be accessed, consult with your carrier and set the APN in the LTE modem configuration (for example, inet.bell.ca):

```
config system lte-modem
  set status enable
  set apn "inet.bell.ca"
end
```

- Some models, such as the FortiGate 30E-3G4G, have built-in LTE modems. In this scenario, the LTE modem is enabled by default. The firewall policy via the LTE interface is also created by default. Once you plug in a SIM card, your network devices can connect to the internet.

Sample FortiGate 30E-3G4G default configuration:

```
config system lte-modem
  set status enable
  set extra-init ''
  set manual-handover disable
  set force-wireless-profile 0
  set authtype none
  set apn ''
  set modem-port 255
  set network-type auto
  set auto-connect disable
  set gpsd-enabled disable
  set data-usage-tracking disable
  set gps-port 255
end
```

```
config firewall policy
....
  edit 3
    set srcintf "internal"
    set dstintf "wwan"
    set srcaddr "all"
    set dstaddr "all"
    set action accept
    set schedule "always"
    set service "ALL"
    set nat enable
  next
end
```

Cellular interface support for IPv6

The cellular interface of FG-40F-3G4G devices supports IPv6.

```
config system lte-modem
    set pdptype {IPv4 | IPv6 | IPv4v6}
end
```

`pdptype`	Specify the Packet Data Protocol (PDP) for the cellular interface: • `IPv4`: use only IPv4. • `IPv6`: use only IPv6. • `IPv4v6`: use both IPv4 and IPv6 (default).

Example

In this example, PDP type is set to IPv4v6 in the wireless profile.

To use IPv4v6:

1. On FortiGate-40F-3G4G, use the `execute lte-modem wireless-profile` command to create or modify a wireless profile with `pdptype` set to `IPv4v6`. See the 3G4G LTE Modem Operator's Manual for details.
2. List all profiles.
   In the following example, `PDP_Type` is set to `3` to indicate support for both IPv4 and IPv6.

```
# execute lte-modem wireless-profile list
ID   Type  Name               APN                  PDP_Type  Authen  Username
 1   0                           ota.bell.ca             3                0
 2   0        Bell               ota.bell.ca             3                0


Profile Type:
    0 ==> QMI_WDS_PROFILE_TYPE_3GPP

Profile PDP type:
    0 ==> QMI_WDS_PDP_TYPE_IPV4
    1 ==> QMI_WDS_PDP_TYPE_PPP
    2 ==> QMI_WDS_PDP_TYPE_IPV6
    3 ==> QMI_WDS_PDP_TYPE_IPV4_OR_IPV6
Authentication:
    0 ==> QMI_WDS_AUTHENTICATION_NONE
    1 ==> QMI_WDS_AUTHENTICATION_PAP
    2 ==> QMI_WDS_AUTHENTICATION_CHAP
    3 ==> QMI_WDS_AUTHENTICATION_PAP|QMI_WDS_AUTHENTICATION_CHAP
```

3. Apply the correct profile.
   In the following example, profile 2 is selected. The `apn` setting must also match the `apn` setting in the selected profile.

```
config sys lte-modem
    set pdptype ipv4v6
    set force-wireless-profile 2
    set apn ota.bell.ca
end
```

4. Wait for the profile to take effect, and then check the data session information:

```
# diagose sys lte-modem data-session-info
LTE Modem data session information:
Interface name:        wwan
IPV4 connection:       QMI_WDS_CONNECTION_STATUS_CONNECTED
IPV6 connection:       QMI_WDS_CONNECTION_STATUS_CONNECTED
Profile ID:            2
Data profile name:     Bell
Profile type:          QMI_WDS_PROFILE_TYPE_3GPP
PDP context type:      QMI_WDS_PDP_TYPE_IPV4_OR_IPV6
APN name:              ota.bell.ca
-------------------------------------
IP:                    10.34.139.21
IP gateway:            10.34.139.22
```

```
IP netmask:            255.255.255.252
Primary DNS:           161.216.153.1
Secondary DNS:         161.216.157.1
MTU:                   1500
Link protocol:         QMI_WDA_LINK_LAYER_PROTOCOL_RAW_IP
------------------------------------
IPv6:                  2605:b100:93b:cf64:bd33:e6ba:b2ef:5e58
IPv6 prefix len:       64
IPv6 gateway:          2605:b100:93b:cf64:60c8:e41d:be4b:eaf5
IPv6 GW prefix len:    64
IPv6 PRI DNS:          2605:b100:880:9::1
IPv6 SEC DNS:          2605:b100:680:9::1
MTU:                   1500
Link protocol:         QMI_WDA_LINK_LAYER_PROTOCOL_RAW_IP
Auto connect:          QMI_WDS_AUTOCONNECT_DISABLED
Network type:          Unknown WDS Bearer Tech
Network type(last):    Unknown WDS Bearer Tech
```

5. Verify IPv4.

   In the following example, an IPv4 address is assigned to the wwan interface, and an IPv4 route is automatically added.

```
# diagnose ip address list
IP=192.168.2.111->192.168.2.111/255.255.255.0 index=5 devname=wan
IP=127.0.0.1->127.0.0.1/255.0.0.0 index=13 devname=root
IP=169.254.1.1->169.254.1.1/255.255.255.0 index=17 devname=fortilink
IP=192.168.1.99->192.168.1.99/255.255.255.0 index=18 devname=lan
IP=127.0.0.1->127.0.0.1/255.0.0.0 index=19 devname=vsys_ha
IP=127.0.0.1->127.0.0.1/255.0.0.0 index=21 devname=vsys_fgfm
IP=10.34.139.21->10.34.139.21/255.255.255.255 index=23 devname=wwan

FortiGate-40F-3G4G # get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       V - BGP VPNv4
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 10.34.139.22, wwan, [1/0]
C       10.34.139.21/32 is directly connected, wwan
```

6. Verify IPv6.

   In the following example, an IPv6 address is assigned to the wwan interface, and an IPv6 route is automatically added.

```
# diagnose ipv6  address list
dev=13 devname=root flag=P scope=254 prefix=128 addr=::1 preferred=4294967295 valid=4294967295
cstamp=2861 tstamp=2861
dev=19 devname=vsys_ha flag=P scope=254 prefix=128 addr=::1 preferred=4294967295
```

```
valid=4294967295 cstamp=5231 tstamp=5231
dev=21 devname=vsys_fgfm flag=P scope=254 prefix=128 addr=::1 preferred=4294967295
valid=4294967295 cstamp=5875 tstamp=5875
dev=23 devname=wwan flag=P scope=0 prefix=64 addr=2605:b100:93b:cf64:bd33:e6ba:b2ef:5e58
preferred=4294967295 valid=4294967295 cstamp=102181 tstamp=102181
dev=23 devname=wwan flag=P scope=253 prefix=64 addr=fe80::8049:4eff:fefc:ea5e
preferred=4294967295 valid=4294967295 cstamp=102181 tstamp=102181

FortiGate-40F-3G4G # get router info6 routing-table database
IPv6 Routing Table
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, B - BGP, V - BGP VPNv6
       > - selected route, * - FIB route, p - stale info
Timers: Uptime

Routing table for VRF=0
S    *> ::/0 [10/0] via 2605:b100:93b:cf64:60c8:e41d:be4b:eaf5, wwan, 00:09:20, [1024/0]
C    *> ::1/128 via ::, root, 00:24:50
C    *> 2605:b100:93b:cf64::/64 via ::, wwan, 00:09:20
```

Active SIM card switching

FortiGates with a cellular modem and dual SIM card can switch in real time from the active SIM card to the passive SIM card when any of the following issues arise with the active SIM card:

- Ping link monitor fails. The SIM switch time depends on the link monitor parameters set.
- An active SIM card cannot be detected. The SIM switch time is about 20 seconds after the SIM card is no longer detected.
- A modem disconnection is detected, and a specified interval has elapsed. The SIM switch time occurs after the specified interval.
- The LTE modem traffic exceeds the specified data plan limit for the configured billing period.

SIM card switching events are captured in the FortiGate event log.

In most cases, SIM cards come with the wireless carrier's APN, which is automatically retrieved at the first connection of the LTE modem. For these cases, you can use SIM cards for different wireless carriers in SIM slot 1 and slot 2.

When one or both SIM cards require their APN settings to be configured on the FortiGate, then both SIM cards should be for the same wireless carrier because `config system lte-modem` currently only supports a single `set apn < apn >` setting.

The following command and options can be used to configure this feature:

```
config system lte-modem
    set data-usage-tracking {enable | disable}
    config sim-switch
```

```
        set by-sim-state {enable | disable}
        set by-connection-state {enable | disable}
        set by-link-monitor {enable | disable}
        set by-data-plan {enable | disable}
        set link-monitor <link-monitor-name>
        set sim-switch-log-alert-interval <interval>
        set sim-switch-log-alert-threshold <threshold>
        set modem-disconnection-time <integer>
    end
    config data-plan
        edit <id>
            set target-sim-slot {SIM-slot-1 | SIM-slot-2}
            set data-limit <integer>
            set data-limit-alert <integer>
            set billing-period {monthly | weekly | daily}
            set billing-date <integer>
            set billing-weekday {sunday | monday | tuesday | wednesday | thursday | friday |
saturday}
            set billing-hour <integer>
            set overage {enable | disable}
            set iccid <string>
            set delay-switch-time <time>
        next
    end
end
```

`data-usage-tracking {enable | disable}`	Enable tracking of data usage for the LTE modem: • `enable`: track data usage. • `disable`: do not track data usage. Must be enabled to configure SIM card switching based on data plan overage.
`config sim-switch`	
`by-sim-state {enable | disable}`	Enable switching based on active SIM card state: • `enable`: switch to the passive SIM card whenever FortiGate cannot detect the active SIM card, such as when the active SIM card is ejected. • `disable`: do not switch SIM cards based on state.
`by-connection-state {enable | disable}`	Enable switching based on the connection state of the active SIM card: • `enable`: switch to the passive SIM card whenever FortiGate detects a modem signal loss after the `modem-disconnection-time` expires. • `disable`: do not switch SIM cards based on the connection state.
`by-link-monitor {enable | disable}`	Enable switching when a configured link monitor fails: • `enable`: switch to the passive SIM card when a link monitor configured with `link-monitor-name` fails. • `disable`: do not switch SIM cards based on the failure of a configured link monitor.
`by-data-plan {enable | disable}`	Enable switching of SIM cards on the LTE modem based on data plan limits: • `enable`: allow SIM card switching when `data-limit` is exceeded.

	• `disable`: do not switch SIM cards when `data-limit` is exceeded.
`link-monitor <link-monitor-name>`	Specify the name of the link monitor to use with `by-link-monitor`.
`sim-switch-log-alert-interval <interval>`	Identify what number of constant SIM card switch events will trigger an event log after the threshold in `sim-switch-log-alert-threshold` is met.
`sim-switch-log-alert-threshold`	Specify how many minutes to wait before creating an event log when the number of SIM card switches defined in `sim-switch-log-alert-interval` is met.
`modem-disconnection-time <integer>`	Specify how many seconds to wait before switching over to the passive SIM card when `by-connection-state` is enabled and a modem signal loss is detected.
`config data-plan`	
`target-sim-slot {sim-slot-1 | sim-slot-2}`	Specify which SIM slot to configure.
`data-limit <integer>`	Specify the data limit for the SIM slot, in MB (0 - 100000, 0 = unlimited data).
`data-limit-alert <integer>`	Specify at what percentage of used `data-limit` to trigger a log entry (1 - 99).
`billing-period {month | week | day}`	Specify the billing period.
`billing-date <integer>`	When `billing-period` is set to `monthly`, specify what day of the month the bill is issued (1 - 31).
`billing-weekday {sunday | monday | tuesday | wednesday | thursday | friday | saturday}`	When `billing-period` is set to `weekly` specify what day of the week the bill is issued.
`billing-hour <integer>`	When `billing-period` is set to `daily` specify what hour of the day the bill is issued (0 - 23).
`overage {enable | disable}`	Disable data usage from exceeding the configured data limit: • `enable`: allow data usage to exceed the amount specified in `data-limit`. • `disable`: do not allow data usage to exceed the amount specified in `data-limit`. When disabled, SIM cards are switched before the data limit is exceeded. Must be disabled to allow SIM card switching.
`iccid <string>`	Specify the Integrated Circuit Card Identification Number (ICCID) for the SIM card in 19 to 20 digits.
`delay-switch-time <integer:integer>`	Delay SIM card switch to a specified UTC time in format HH:MM.

Example 1

In this example, automatic SIM card switching is disabled. When disabled, the SIM card only works in the default slot1, but you can manually switch the SIM card to slot2. Event logs include details about the SIM card switch.

To manually switch a SIM card:

1. Disable automatic SIM card switching:

```
config system lte-modem
    config sim-switch
        set by-sim-state disable
        set by-connection-state disable
        set by-link-monitor disable
        set sim-slot 1
    end
end
```

2. Manually switch the SIM card from slot1 to slot2, and run the following command:

```
# execute lte-modem sim-switch
```

The SIM card switch may take a few seconds. You can run `diagnose system lte-modem sim-info` to check the results.

The following log is generated after unplugging an active SIM card:

```
7: date=2023-05-02 time=10:41:05 eventtime=1683049264795418820 tz="-0700" logid="0100046518"
type="event" subtype="system" level="information" vd="root" logdesc="LTE modem active SIM card
switch event" msg="LTE modem active SIM card slot changed to 2 by user."
```

Example 2

In this example, automatic SIM card switching is enabled and configured to switch based on SIM state, connection state, or link monitor state, and it includes example event logs for each scenario.

To enable automatic SIM card switching by SIM state:

1. Enable automatic SIM card switching by SIM state:

```
config system lte-modem
    config sim-switch
        set by-sim-state enable
    end
end
```

With this configuration, the second SIM card becomes active when the active SIM card is no longer detected, for example, if the active SIM card is ejected. The following event logs are generated:

```
5: date=2023-04-28 time=17:27:27 eventtime=1682728046989682780 tz="-0700" logid="0100046513"
type="event" subtype="system" level="information" vd="root" logdesc="LTE modem data link
```

```
connection event" msg="LTE modem data link changed from QMI_WDS_CONNECTION_STATUS_DISCONNECTED
to QMI_WDS_CONNECTION_STATUS_CONNECTED"

6: date=2023-04-28 time=17:27:17 eventtime=1682728036493684280 tz="-0700" logid="0100046512"
type="event" subtype="system" level="information" vd="root" logdesc="LTE modem SIM card state
event" msg="LTE modem SIM card change from QMI_UIM_CARD_STATE_ABSENT to QMI_UIM_CARD_STATE_
PRESENT"

7: date=2023-04-28 time=17:27:12 eventtime=1682728032589776580 tz="-0700" logid="0100046513"
type="event" subtype="system" level="information" vd="root" logdesc="LTE modem data link
connection event" msg="LTE modem data link changed from QMI_WDS_CONNECTION_STATUS_CONNECTED to
QMI_WDS_CONNECTION_STATUS_DISCONNECTED"

8: date=2023-04-28 time=17:27:11 eventtime=1682728031245682560 tz="-0700" logid="0100046512"
type="event" subtype="system" level="information" vd="root" logdesc="LTE modem SIM card state
event" msg="LTE modem SIM card change from QMI_UIM_CARD_STATE_PRESENT to QMI_UIM_CARD_STATE_
ABSENT"
```

To enable automatic SIM card switching by connection state:

1. Enable automatic SIM card switching by connection state:

```
config system lte-modem
    config sim-switch
        set by-connection-state enable
        set modem-disconnection-time 30
        set sim-switch-log-alert-interval 15
        set sim-switch-log-alert-threshold 5
    end
end
```

With this configuration, the second SIM card becomes active when the modem cannot establish a connection with the carrier through the active SIM card. For example, a FortiGate is in a room with poor signal quality. With this configuration, the SIM card switch is triggered after the modem is detected as disconnected for 30 seconds, and the following event log is generated:

```
56: date=2023-05-01 time=11:14:56 eventtime=1682964896356933480 tz="-0700" logid="0100046519"
type="event" subtype="system" level="notice" vd="root" logdesc="LTE modem active SIM card
switched: modem disconnection detected" msg="LTE modem active SIM card slot changed to 2, due
to modem connection down."

66: date=2023-05-01 time=11:14:13 eventtime=1682964852964869400 tz="-0700" logid="0100046519"
type="event" subtype="system" level="notice" vd="root" logdesc="LTE modem active SIM card
switched: modem disconnection detected" msg="LTE modem active SIM card slot changed to 1, due
to modem connection down."
```

When poor signal quality causes SIM cards to frequently switch back and forth, and the flapping rate occurs more than five times within the configured 15 minute time period, an event log is triggered to record the flapping severity:

```
65: date=2023-05-01 time=11:14:13 eventtime=1682964853083194400 tz="-0700" logid="0100046521"
type="event" subtype="system" level="warning" vd="root" logdesc="LTE modem active SIM card
```

```
slot flipped back and forth in short time" msg="LTE modem switched SIM slot 8 times in last 15
minutes, which is greater than 5 times threshold."
```

To enable automatic SIM card switching based on link monitor:

1. Enable automatic SIM card switching by link monitor, and specify the link monitor:

```
config system lte-modem
    config sim-switch
        set by-link-monitor enable
        set link-monitor "modem"
        set sim-switch-log-alert-interval 15
        set sim-switch-log-alert-threshold 5
    end
    config system link-monitor
    edit "modem"
        set srcintf "wwan"
        set server "8.8.8.8"
        set interval 1000
        set probe-timeout 100
        set failtime 3
        set recoverytime 8
    next
end
```

   With this configuration, the second SIM card becomes active when the link monitor detects the active SIM card exceeds the SLA.

2. Check the link monitor status. In this example, the link monitor status is dead:

```
# diagnose system link-monitor status modem

Link Monitor: modem, Status: dead, Server num(1), cfg_version=7 HA state: local(dead), shared
(dead)
Flags=0x9 init log_downgateway, Create time: Fri Apr 28 16:34:56 2023
Source interface: wwan (19)
VRF: 0
Interval: 1000 ms
Service-detect: disable
Diffservcode: 000000
Class-ID: 0
  Peer: 8.8.8.8(8.8.8.8)
        Source IP(10.192.195.164)
        Route: 10.192.195.164->8.8.8.8/32, gwy(10.192.195.165)
        protocol: ping, state: dead
                Packet lost: 11.667%
                MOS: 4.353
                Number of out-of-sequence packets: 0
                Recovery times(5/8) Fail Times(1/3)
                Packet sent: 60, received: 56, Sequence(sent/rcvd/exp): 61/61/62
```

   The following event log is generated when the link-monitor status is dead:

```
15: date=2023-04-28 time=16:31:38 eventtime=1682724697936494139 tz="-0700" logid="0100046520"
type="event" subtype="system" level="notice" vd="root" logdesc="LTE modem active SIM card
switched: link monitor probe failure detected" msg="LTE modem active SIM card slot changed to
2, due to link monitor probe failures."

19: date=2023-04-28 time=16:31:13 eventtime=1682724673152506599 tz="-0700" logid="0100022932"
type="event" subtype="system" level="warning" vd="root" logdesc="Link monitor status warning"
name="modem" interface="wwan" probeproto="ping" msg="Link Monitor changed state from alive to
dead, protocol: ping."
```

Example 3

In this example, data tracking and SIM card switching by data plan are enabled for the LTE modem. Each SIM card for the LTE modem is configured with a data plan.

When traffic causes data usage to surpass the configured data limit for one SIM card, the LTE modem disconnects, and the wwan interface loses its IP address and gateway. The idle SIM card becomes active, as long as it has available data to be used. After the SIM card switch completes, the LTE modem reconnects, and the wwan interface gains its IP address and gateway again.

To configure SIM card switching by data plan overage:

1. Enable data tracking for the LTE modem:

```
config system lte-modem
    set data-usage-tracking enable
end
```

2. Enable SIM card switching by data plan for the LTE modem:

```
config system lte-modem
    config sim-switch
        set by-data-plan enable
    end
end
```

3. Configure a data plan for each SIM card on the LTE modem:

   In this example, SIM-slot-1 is configured with a data limit of 50 MB for a monthly bill issued on the 10th day of the month.

   SIM-slot-2 is configured is configured with a data limit of 60 MB for a monthly bill issued on the first day of the month.

   Data overage is disabled for both SIM card slots to allow the SIM cards to switch when the data limits are exceeded.

```
config system lte-modem
    config data-plan
        edit "1"
            set target-sim-slot SIM-slot-1
            set data-limit 50
            set billing-period monthly
```

```
            set overage disable
            set billing-date 10
        next
        edit "2"
            set target-sim-slot SIM-slot-2
            set data-limit 60
            set billing-period monthly
            set overage disable
            set billing-date 1
        next
    end
end
```

When the specified `data-limit` is exceeded while `overage` is disabled, the SIM card switch is triggered.

When `overage` is enabled, the specified `data-limit` can be exceeded, and a SIM card switch is not triggered.

Data usage is reset after the billing period passes.

4. Monitor data usage against the data limit:

```
# diagnose sys lte-modem data-usage
Estimated LTE Modem data usage in this billing cycle:
Active data plan:              1
Active SIM slot:               slot-1
Plan data limit:               60(MB)
Plan overage status:           disable
sim-switch.by-data-plan:       enable
Usage:                         67(MB)
Usage percentage:              111.67%
Current time:                  2023-07-20 16:16:38
Plan refresh time:             2023-08-05 01:00:00
============================================
Idle data plan:                2
Idle SIM slot:                 slot-2
Idle Plan data limit:          100(MB)
Idle Plan overage status:      disable
Idle Plan Usage:               78(MB)
Idle Plan Usage percentage:    78.00%
Idle Plan refresh time:        2023-08-10 01:00:00
```

5. After the SIM card switch completes, view the active SIM card:

```
# diagnose sys lte-modem sim-info
LTE Modem SIM card information:
Active Slot: Slot 2
SIM state: QMI_UIM_CARD_STATE_PRESENT
ICCID: 89302370323035043340
IMSI: 302370605258650
Country: Canada
```

```
Network: Fido
SIM PIN status: Verified
```

Airplane mode and LTE/BLE

Airplane mode is supported on FGR-70F-3G4G models to enable/disable radio frequency signals for the internal LTE modem and Bluetooth Low Energy (BLE) module:

```
config system global
    set airplane-mode {disable | enable}
end
```

By default airplane mode is disabled, and LTE and BLE radio frequency signals are transmitted. Airplane mode can be enabled with a CLI command followed by a reboot of the FortiGate. Once airplane mode is enabled, LTE and BLE radio frequency signals remain silent during normal operation of the FortiGate.

A specific BIOS version is required to ensure radio frequency signals remain silent for LTE and BLE modules when FortiGate is rebooted.

`airplane-mode {disable | enable}`	Enable or disable airplane mode on FGR-70F-3G4G models: • `disable`: disable airplane mode, which means radio frequency signals of the internal LTE modem and BLE module are enabled and transmitted. • `enable`: enable airplane mode, which means radio frequency signals of the internal LTE modem and BLE module are turned off.

Example

To disable airplane mode:

1. Disable airplane mode:

```
config system global
    ...
    set airplane-mode disable
    ...
end
```

   Radio frequency signals of the LTE modem and BLE module are turned on.
2. Use the following commands to verify the settings:

`execute usb-device list`	Check the status of the LTE modem.
`diagnose test application lted 5`	Check the signal strength of the LTE modem.

`diagnose sys lte-modem modem-details`	Get detailed information about the LTE modem.
`diagnose sys lte-modem data-session-info`	Get session information about the LTE modem.
`diagnose bluetooth test_bt_conn`	Check the status of the BLE mode.
`diagnose bluetooth status`	Check the bluetooth status of the BLE mode.

To enable airplane mode:

1. Enable airplane mode:

```
config system global
    ...
    set airplane-mode enable
    ...
Enabling airplane mode will turn off LTE modem and Bluetooth RF signals.
Do you want to continue? (y/n)y
end
```

2. Reboot the FortiGate.

```
execute reboot
This operation will reboot the system !
Do you want to continue? (y/n)y
```

The LTE modem and BLE module are disabled, and radio frequency signals are turned off.

3. Show the configuration to confirm that airplane mode is enabled.

```
show full-configuration
config system global
    ...
    set airplane-mode enable
    ...
end
```

4. Check the USB device list (`execute usb-device list`) to confirm that the modem is not displayed in the list.
5. Check the signal strength to confirm that the modem is not found.

```
# diagnose test application lted 5
Modem device not currently connected! Please try again later...
```

6. Check the modem details to confirm that airplane mode is enabled, and the modem is not detected.

```
# diagnose sys lte-modem modem-details
LTE Modem detailed information:
system.global.airplane-mode:      On
Modem detected:   No
```

7. Check the modem session information to confirm that the modem is not detected.

```
# diagnose sys lte-modem data-session-info
LTE Modem data session information:
Modem not detected!
```

8. Run a Bluetooth test to confirm that airplane mode is on and Bluetooth testing is not allowed.

```
# diagnose bluetooth test_bt_conn
It's on airplane mode now. Bluetooth testing is not allowed.
```

9. Check Bluetooth status to confirm that the BLE module is disabled.

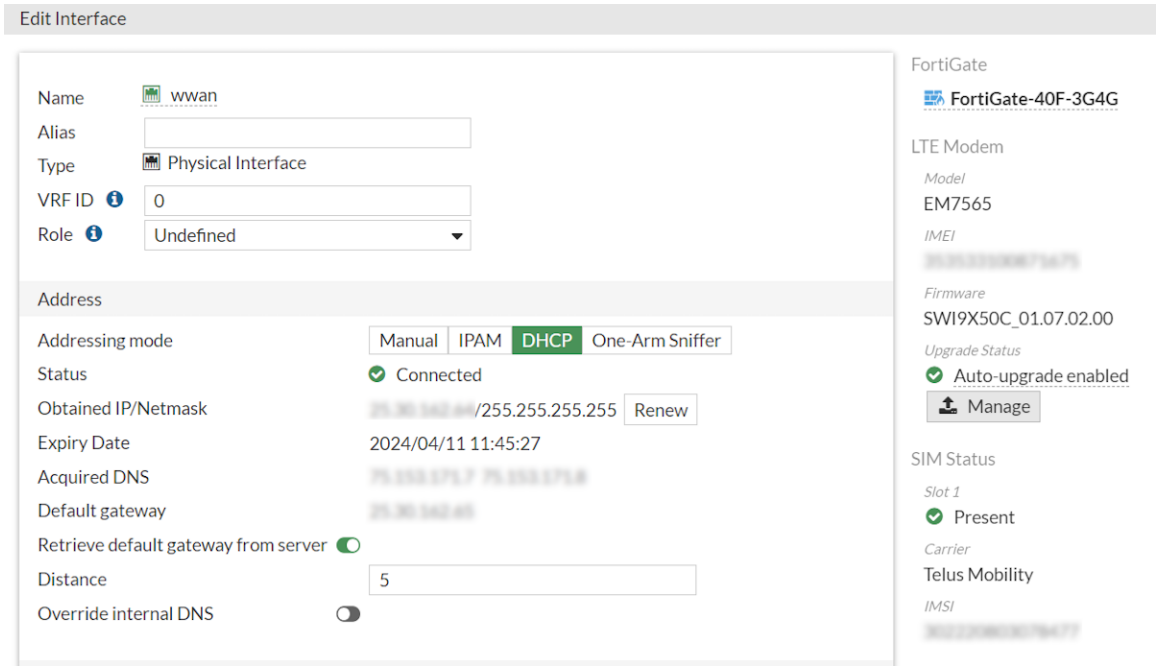
```
# diagnose bluetooth status
Bluetooth Status: RESET BOOTLOADER
Connect State(0):  BLE_MODE_DISABLED
```

Upgrade LTE modem firmware directly from FortiGuard

LTE modem firmware can be upgraded directly from the FortiGuard. This simplifies the process by eliminating the need for manual downloading and uploading, and offers users the flexibility to schedule the upgrade.

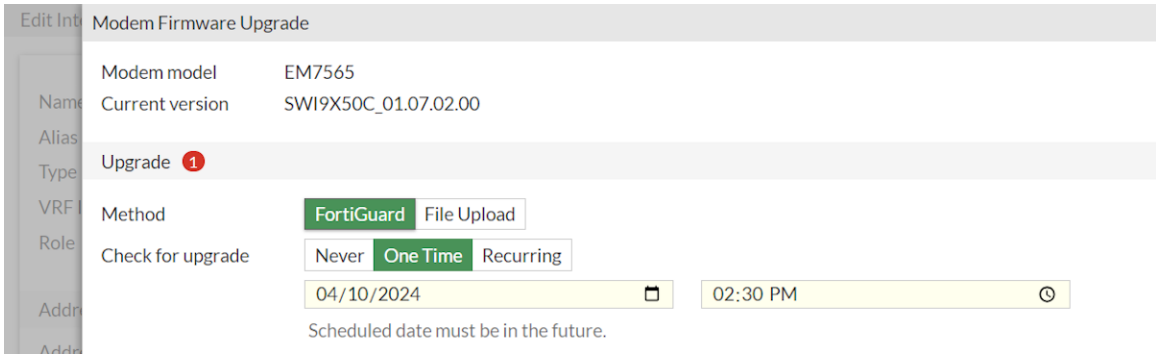
To schedule LTE modem firmware upgrades from FortiGuard in the GUI:

1. Go to *Network* > *Interfaces*.
2. Select the LTE modem interface and click *Edit*.

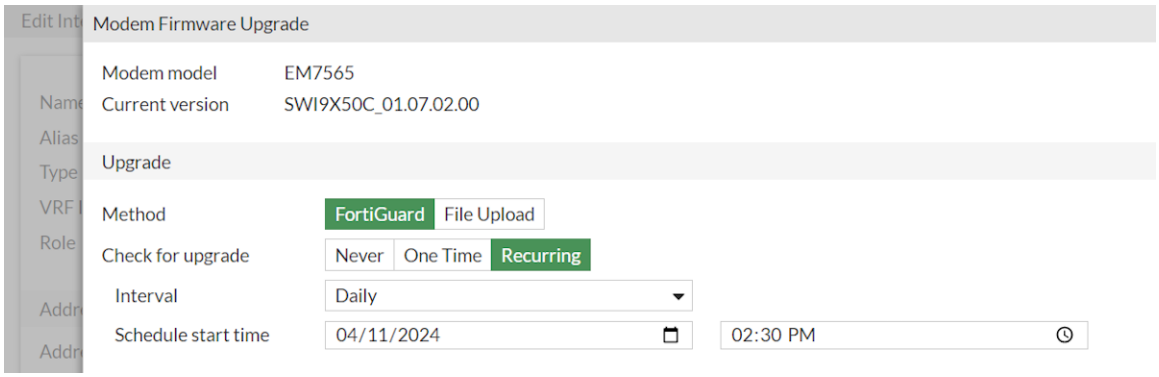


3. Click *Manage*.
4. Set the *Method* as *FortiGuard*.
5. Configure the upgrade schedule:

- *Never*: Disable LTE modem firmware upgrades from FortiGuard.
- *One Time*: Schedule a date and time to upgrade the firmware from FortiGuard.



- *Recurring*: Schedule a recurring upgrade to the firmware from FortiGuard based on a set *Interval*, start date, and start time.



6. Click *OK*.

To schedule a one time LTE modem firmware upgrade from FortiGuard in the CLI:

```
config system central-management
    set type fortiguard
    set ltefw-upgrade-time <YYYY-MM-DD HH:MM:SS>
end
```

To schedule recurring LTE modem firmware upgrades from FortiGuard in the CLI:

```
config system central-management
    set type fortiguard
    set ltefw-upgrade-time <YYYY-MM-DD HH:MM:SS>
    set ltefw-upgrade-frequency {everyHour | every12hour | everyDay | everyWeek}
end
```

To disable LTE modem firmware upgrades from FortiGuard in the CLI:

```
config system central-management
    set type fortiguard
```

```
    set allow-remote-lte-firmware-upgrade disable
end
```

LLDP reception

Device detection can scan LLDP as a source for device identification, but the FortiGate does not read or store the full information. Enabling LLDP reception allows the FortiGate to receive and store LLDP messages, learn about active neighbors, and makes the LLDP information available via the CLI, REST API, and SNMP.

You need to enable device detection (`device-identification`) at the interface level, and then `lldp-reception` can be enabled on three levels: globally, per VDOM, or per interface.

To configure device identification on an interface:

```
config system interface
    edit <port>
        set device-identification enable
    next
end
```

To configure LLDP reception globally:

```
config system global
    set lldp-reception enable
end
```

To configure LLDP reception per VDOM:

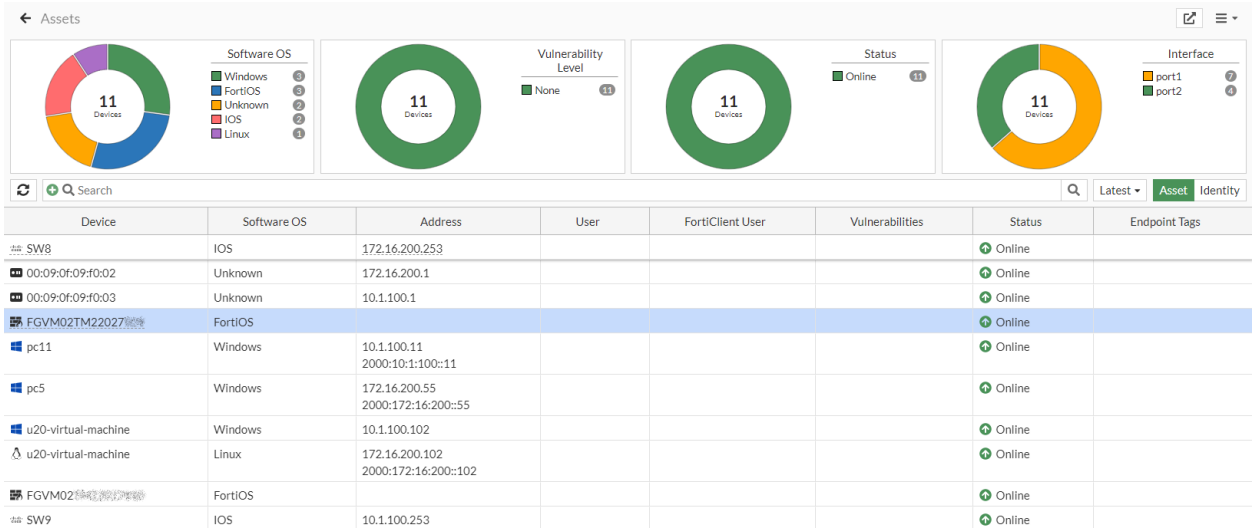
```
config system setting
    set lldp-reception enable
end
```

To configure LLDP reception per interface:

```
config system interface
    edit <port>
        set lldp-reception enable
    next
end
```

To view the LLDP information in the GUI:

1. Go to *Dashboard* > *Assets & Identities*.
2. Expand the *Assets* widget to full screen.



To view the received LLDP information in the CLI:

```
# diagnose user device list
hosts
vd root/0  00:0c:29:1c:03:ca  gen 4320  req 0
    created 515971s  gen 31  seen 28s  port1  gen 12
    hardware vendor 'Fortinet'  src lldp  id 4120  weight 255
    type 'Router'  src lldp  id 4120  weight 255
    family 'FortiGate'  src lldp  id 4120  weight 255
    os 'FortiOS'  src lldp  id 4120  weight 255
    hardware version 'VM64'  src lldp  id 4120  weight 255
    software version '7.4.0 Build 2360'  src lldp  id 4120  weight 255
    host 'FGVM02TM22027XXX'  src lldp
```

To view additional information about LLDP neighbors and ports:

```
# diagnose lldp rx neighbor {summary | details | clear}
```

```
# diagnose lldp rx port {details | summary | neighbor | filter}
```

```
# diagnose lldp rx port neighbor {summary | details}
```

Note that the port index in the output corresponds to the port index from the following command:

```
# diagnose netlink interface list port2 port3 | grep index
      if=port2 family=00 type=1 index=4 mtu=1500 link=0 master=0
      if=port3 family=00 type=1 index=5 mtu=1500 link=0 master=0
```

To view the received LLDP information in the REST API:

```
{
  "http_method":"GET",
  "results":[
```

```
    {
      "mac":"90:9c:9c:c9:c9:90",
      "chassis_id":"90:9C:9C:C9:C9:90",
      "port":19,
      "port_id":"port12",
      "port_desc":"port12",
      "system_name":"S124DN3W00000000",
      "system_desc":"FortiSwitch-124D v3.6.6,build0416,180515 (GA)",
      "ttl":120,
      "addresses":[
        {
          "type":"ipv4",
          "address":"192.168.1.99"
        }
      ]
    }
  ],
  "vdom":"root",
  "path":"network",
  "name":"lldp",
  "action":"neighbors",
  "status":"success",
  "serial":"FG201E4Q00000000",
  "version":"v7.4.0",
  "build":2360
}
```

```
{
  "http_method":"GET",
  "results":[
    {
      "name":"port1",
      "rx":320,
      "neighbors":1
    }
  ],
  "vdom":"root",
  "path":"network",
  "name":"lldp",
  "action":"ports",
  "mkey":"port1",
  "status":"success",
  "serial":"FG201E4Q00000000",
  "version":"v7.4.0",
  "build":2360
}
```

Virtual routing and forwarding

Virtual Routing and Forwarding (VRF) is used to divide the FortiGate's routing functionality (layer 3), including interfaces, routes, and forwarding tables, into separate units. Packets are only forwarded between interfaces that have the same VRF.

An exception applies to VRF 0. When traffic that is destined for a local IP (IP assigned to an interface) in another VRF comes into an interface in VRF 0, the packet is considered a local-in packet in VRF 0 and is allowed to pass.

VDOMs divide the FortiGate into two or more complete and independent virtual units that include all FortiGate functions. VDOMs can be used for routing segmentation, but that should not be the only reason to implement them when a less complex solution (VRFs) can be used. VDOMs also support administration boundaries, but VRFs do not.

Up to 512 VRFs can be configured per VDOM for any device, but only ten VDOMs can be configured by default on a FortiGate (more VDOMs can be configured on larger devices with additional licenses).

- Implementing VRF on page 791
- VRF routing support on page 792
- Route leaking between VRFs with BGP on page 802
- Route leaking between multiple VRFs on page 805
- VRF with IPv6 on page 816
- IBGP and EBGP support in VRF on page 820
- Support cross-VRF local-in and local-out traffic for local services on page 822
- Support specific VRF ID for local-out traffic on page 825

Implementing VRF

VRFs are always enabled and, by default, all routing is done in VRF 0. To use additional VRFs, assign a VRF ID to an interface. All routes relating to that interface are isolated to that VRF specific routing table. Interfaces in one VRF cannot reach interfaces in a different VRF.

If some traffic does have to pass between VRFs, route leaking can be used. See Route leaking between VRFs with BGP on page 802.

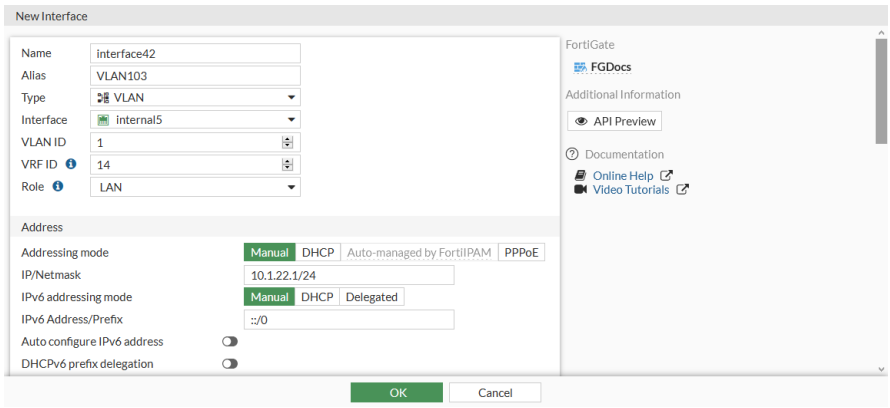
VRF 0 is a special VRF, by default, all routing is done in VRF 0, so all routes in different VRFs will all be included in VRF 0. VRF 0 cannot be used in the cross-VRF case.

For local-in/out traffic, all routes relating to one VRF are isolated from other VRFs so interfaces in one VRF cannot reach interfaces in a different VRF except VRF 0.

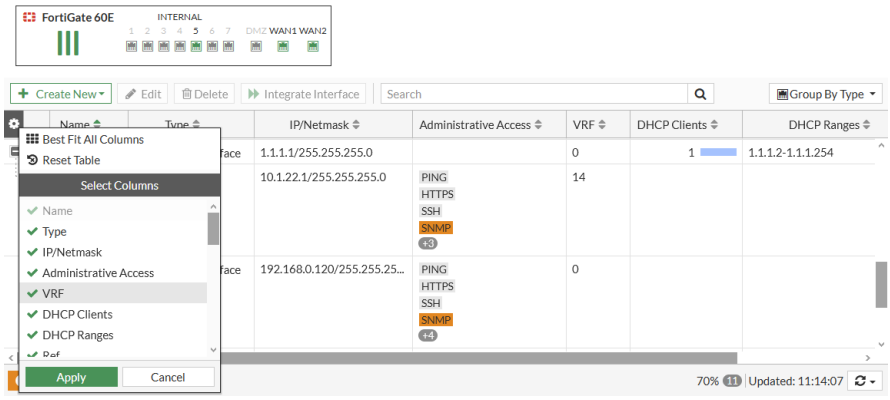
Enable *Advanced Routing* in *System > Feature Visibility* to configure VRFs.

To configure a VRF ID on an interface in the GUI:

1. Go to *Network* > *Interfaces* and click *Create New* > *Interface*.
2. Enter a value in the VRF ID field.
3. Configure the other settings as needed.



4. Click *OK*.
5. To add the VRF column in the interface table, click the gear icon, select *VRF*, and click *Apply*.

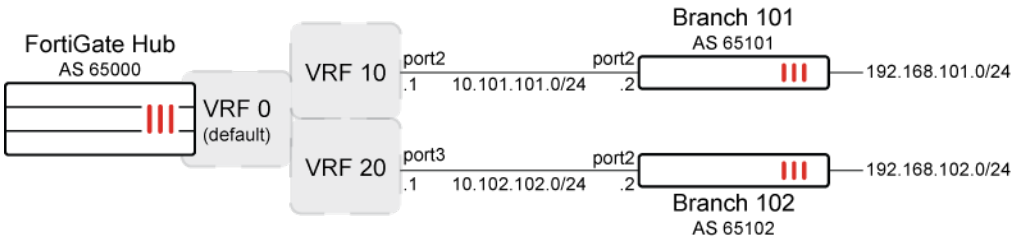


To configure a VRF ID on an interface in the CLI:

```
config system interface
    edit interface42
        ...
        set vrf 14
    next
end
```

VRF routing support

VRF supports static routing, OSPF, and BGP. Other routing protocols require using VDOMs.



BGP

The following BGP examples are provided:

- BGP example 1: update VRF neighbors on page 793
- BGP example 2: use overlapping subnets on page 796

BGP example 1: update VRF neighbors

In this example, BGP is used to update the VRF that it is neighbors with.

The hub is configured with two neighbors connected to two interfaces. The branches are configured to match the hub, with branch networks configured to redistribute into BGP.

Policies must be created on the hub and branches to allow traffic between them.

To configure the hub:

```
config router bgp
    set as 65000
    config neighbor
        edit "10.101.101.2"
            set soft-reconfiguration enable
            set interface "port2"
            set remote-as 65101
            set update-source "port2"
        next
        edit "10.102.102.2"
            set soft-reconfiguration enable
            set interface "port3"
            set remote-as 65102
            set update-source "port3"
        next
    end
end
```

To configure branch 101:

```
config router bgp
    set as 65101
    config neighbor
        edit "10.101.101.1"
```

```
            set soft-reconfiguration enable
            set interface "port2"
            set remote-as 65000
            set update-source "port2"
        next
    end
    config redistribute connected
        set status enable
    end
end
```

To configure branch 102:

```
config router bgp
    set as 65102
    config neighbor
        edit "10.102.102.1"
            set soft-reconfiguration enable
            set interface "port2"
            set remote-as 65000
            set update-source "port2"
        next
    end
    config redistribute connected
        set status enable
    end
end
```

To verify the BGP neighbors and check the routing table on the hub:

```
# get router info bgp summary
BGP router identifier 192.168.0.1, local AS number 65000
BGP table version is 2
2 BGP AS-PATH entries
0 BGP community entries

Neighbor               V       AS              MsgRcvd MsgSent TblVer  InQ     OutQ Up/Down    State/PfxRcd
10.101.101.2   4       65101           4               4               2               0       0       00:01:05
10.102.102.2   4       65102           3               3               1               0       0       00:00:30

Total number of neighbors 2
```

```
# get router info routing-table all
Routing table for VRF=0
Codes (…)
S*     0.0.0.0/0 [10/0] via 192.168.0.254, port1
C      10.101.101.0/24 is directly connected, port2
C      10.102.102.0/24 is directly connected, port3
C      192.168.0.0/24 is directly connected, port1
B      192.168.101.0/24 [20/0] via 10.101.101.2, port2, 00:01:25
B      192.168.102.0/24 [20/0] via 10.102.102.2, port3, 00:00:50
```

To configure VRF on the hub:

1. Put the interfaces into VRF:

```
config system interface
    edit port2
        set vrf 10
    next
    edit port3
        set vrf 20
    next
end
```

2. Restart the router to reconstruct the routing tables:

```
# execute router restart
```

3. Check the routing tables:

```
# get router info routing-table all
Routing table for VRF=0
Codes (…)
S*     0.0.0.0/0 [10/0] via 192.168.0.254, port1
C      192.168.0.0/24 is directly connected, port1

Routing table for VRF=10
C      10.101.101.0/24 is directly connected, port2
B      192.168.101.0/24 [20/0] via 10.101.101.2, port2, 00:02:25

Routing table for VRF=20
C      10.102.102.0/24 is directly connected, port3
B      192.168.102.0/24 [20/0] via 10.102.102.2, port2, 00:01:50
```

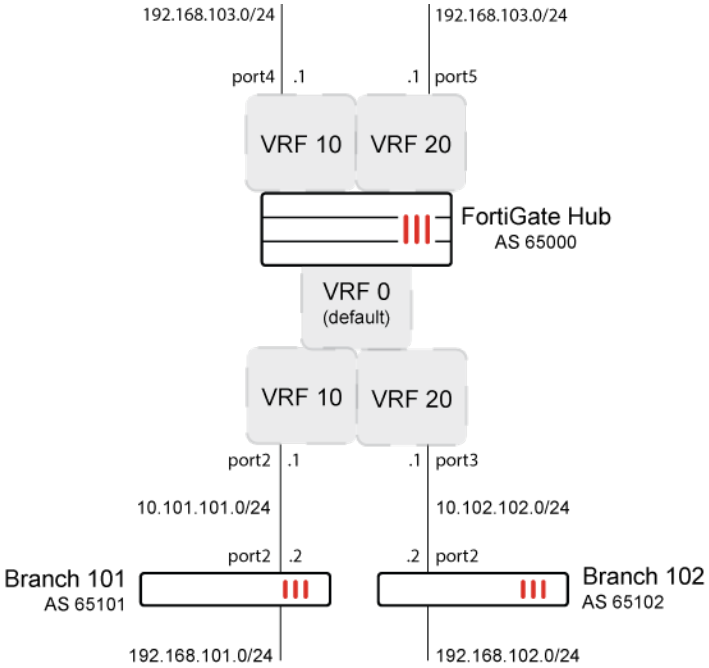
4. Check the BGP summary:

```
# get router info bgp summary

VRF 10 BGP router identifier 10.101.101.1, local AS number 65000
BGP table version is 1
2 BGP AS-PATH entries
0 BGP community entries

Neighbor               V       AS              MsgRcvd MsgSent TblVer  InQ     OutQ Up/Down    State/PfxRcd
10.101.101.2   4       65101           4               4               2               0       0       00:0

Total number of neighbors 1

VRF 10 BGP router identifier 10.101.101.1, local AS number 65000
BGP table version is 1
2 BGP AS-PATH entries
0 BGP community entries

Neighbor               V       AS              MsgRcvd MsgSent TblVer  InQ     OutQ Up/Down    State/PfxRcd
10.102.102.2   4       65102           3               3               1               0       0       00:0
```

```
Total number of neighbors 1
```

BGP example 2: use overlapping subnets

Building on BGP example 1: update VRF neighbors on page 793, example 2 includes two additional interfaces assigned to different VRFs: port4 belongs to VRF 10, and port5 belongs to VRF 20. The IP address and subnet configuration on port4 and port5 are identical, resulting in subnet overlap.



The hub needs to selectively advertise the overlapping prefix through BGP only to the peer within VRF 10. By default, the `config network` command of BGP does not advertise prefix on a per-VRF basis.

Thus to achieve selective route advertisement to a BGP peer belonging to specific VRF, route maps can be used.

To configure the hub:

1. Allow configuration of subnet overlap:

```
config system settings
    set allow-subnet-overlap enable
end
```

2. Configure IP addresses on port4 and port5:

```
config system interface
    edit "port4"
        set ip 192.168.103.1 255.255.255.0
    next
end
config system interface
```

```
    edit "port5"
        set ip 192.168.103.1 255.255.255.0
    next
end
```

3. Put port4 into VRF 10 and port5 into VRF 20:

```
config system interface
    edit "port4"
        set vrf 10
    next
end
config system interface
    edit "port5"
        set vrf 20
    next
end
```

4. Advertise the overlapped subnet inside BGP using the `config network` command.
   By default, prefix 192.168.103.0/24 is advertised to BGP peers of all VRFs.

```
config router bgp
    set as 65000
    config neighbor
        edit "10.101.101.2"
            set soft-reconfiguration enable
            set interface "port4"
            set remote-as 65101
            set update-source "port4"
        next
        edit "10.102.102.2"
            set soft-reconfiguration enable
            set interface "port3"
            set remote-as 65102
            set update-source "port3"
        next
    end
    config network
        edit 1
            set prefix 192.168.103.0 255.255.255.0
        next
    end
end
```

To verify the routing table before using route-map:

1. Verify that the routing table on Branch 101 displays only BGP routes:

```
# get router info routing-table bgp
Routing table for VRF=0
B       192.168.103.0/24 [20/0] via 10.102.102.1, port2, 00:17:04, [1/0]
```

2. Verify that the routing table on Branch 102 displays only BGP routes:

```
# get router info routing-table bgp
Routing table for VRF=0
B       192.168.103.0/24 [20/0] via 10.101.101.1, port2, 00:18:48
```

To use a route map to advertise a prefix for each VRF:

1. Configure `route-map` on hub to match VRF 10:

```
config router route-map
    edit "VRF_10"
        config rule
            edit 1
                set match-vrf 10
            next
        end
    next
end
```

2. Use `route-map` inside hub's BGP configuration (that is, inside the `config network` command) to selectively advertise a prefix 192.168.103.0/24 to BGP peers that belong to VRF 10 (that is, Branch 101):

```
config router bgp
    set as 65000
    config neighbor
        edit "10.101.101.2"
            set soft-reconfiguration enable
            set interface "port4"
            set remote-as 65101
            set update-source "port4"
        next
        edit "10.102.102.2"
            set soft-reconfiguration enable
            set interface "port3"
            set remote-as 65102
            set update-source "port3"
        next
    end
    config network
        edit 1
            set prefix 192.168.103.0 255.255.255.0
            set route-map "VRF_10"
        next
    end
end
```

To verify the routing table after using route-map:

1. Verify the routing table on Branch 101:

```
# get router info routing-table bgp
Routing table for VRF=0
B       192.168.103.0/24 [20/0] via 10.102.102.1, port2, 00:17:04, [1/0]
```

2. Check the routing table on Branch 102. The routes are not advertised to BGP peer belonging to VRF 20.

```
# get router info routing-table bgp
No route available
```

3. Verify advertised BGP routes on hub on per-neighbor basis:

```
# get router info bgp neighbors 10.101.101.2 advertised-routes
VRF 10 BGP table version is 1, local router ID is 192.168.103.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> 192.168.103.0    10.101.101.1                   100  32768        0 i <-/->

Total number of prefixes 1

# get router info bgp neighbors 10.102.102.2 advertised-routes
% No prefix for neighbor 10.102.102.2
```

OSPF

OSPF routes in VRFs work the same as BGP: the interface that OSPF is using is added to the VRF.

To configure the hub:

1. Configure OSPF:

```
config router ospf
    set router-id 1.1.1.1
    config area
        edit 0.0.0.0
        next
    end
    config ospf-interface
        edit Branch101
            set interface “port2”
            set dead-interval 40
            set hello-interval 10
        next
        edit Branch102
            set dead-interval 40
            set hello-interval 10
        next
    end
    config network
        edit 0
```

```
            set prefix 10.101.101.0 255.255.255.0
        next
        edit 0
            set prefix 10.102.102.0 255.255.255.0
        next
        edit 0
            set prefix 192.168.1.0 255.255.255.0
        next
    end
end
```

2. Put the interfaces into VRF:

```
config system interface
    edit port2
        set vrf 10
    next
    edit port3
        set vrf 20
    next
end
```

To configure branch 101:

```
config router ospf
    set router-id 101.101.101.101
    config area
        edit 0.0.0.0
        next
    end
    config ospf-interface
        edit HUB
            set interface port2
            set dead-interval 40
            set hello-interval 10
        next
    end
    config network
        edit 0
            set prefix 10.101.101.0 255.255.255.0
        next
        edit 0
            set prefix 192.168.101.0 255.255.255.0
        next
    end
end
```

To check the routing table and OSPF summary:

```
# get router info routing-table ospf
```

```
# get router info ospf interface
```

Static route

Static routes in VRFs work the same as BGP and OSPF because the interface that the static route is using is added to the VRF.

To add a VRF ID in a static route in the GUI:

1. Configure the interface:
   a. Go to *Network* > *Interfaces*.
   b. Click *Create New* > *Interface* or *Edit* an existing interface.
   c. Enter a value in the *VRF ID* field.
   d. Configure the other settings as needed.
   e. Click *OK*.
2. Add a static route to VRF. For example, using blackhole:
   a. Go to *Network* > *Static Routes*.
   b. Click *Create New* and select the type of static route.
   c. Enter a *Subnet*.
   d. In the *Interface* field, select *Blackhole*.
   e. In the *VRF ID* field, enter the ID created in step one.
   f. Click *OK*.

To add a VRF ID in a static route in the CLI:

1. Configure the interface:

```
config system interface
    edit port2
        set vrf 10
    next
end
```

2. Add a static route to the VRF. For example, using blackhole:

```
config router static
    edit 3
        set dst 0.0.0.0/0
        set blackhole enable
        set vrf 10
    next
end
```

A static route can also be added to the VRF when using an IPsec interface by enabling VPN ID with IPIP encapsulation. See SD-WAN segmentation over a single overlay on page 1495 for more information.

To add a static route to the VRF when using IPsec:

```
config vpn ipsec phase1-interface
    edit "vpn1"
        set interface "port2"
        set auto-discovery-receiver enable
        set encapsulation vpn-id-ipip
        set remote-gw 1.1.101.1
        set psksecret ******
    next
end
config router static
    edit 1
        set dst 10.32.0.0 255.224.0.0
        set device "vpn1"
        set vrf 10
    next
end
```

To check the routing table:

```
# get router info routing-table static
```

Route leaking between VRFs with BGP

Route leaking allows you to configure communication between VRFs. If route leaking is not configured, then the VRFs are isolated. This example shows route leaking with BGP using virtual inter-VDOM links.

In this example, a hub FortiGate forms BGP neighbors with two branches. It learns the networks 192.168.101.0/24 and 192.168.102.0/24 from the neighbors and separates them into VRF 10 and VRF 20.

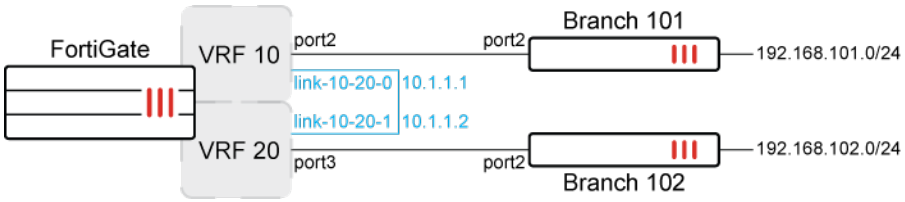
To leak the learned routes to each other, an inter-VDOM link (IVL) is formed. An IVL normally bridges two VDOMs, but in this case the links reside on the same VDOM and are used to bridge the two VRFs. NPU inter-VDOM links could also be used on models that support it to deliver better performance.

You can use the following command to enable NPU inter-VDOM links:

```
config system global
    set single-vdom-npuvlink enable
end
```

After entering this command, the NPU inter-VDOM links (npu0_vlink0, npu0_vlink1, and so on) are visible from the GUI and CLI and you can configure them to send traffic between VRFs.

VRF 10 has a leaked route to 192.168.102.0/24 on IVL *link-10-20-0*, and VRF 20 has a leaked route to 192.168.101.0/24 on IVL *link-10-20-1*,



To configure route leaking:

1. Allow interface subnets to use overlapping IP addresses:

```
config system settings
    set allow-subnet-overlap enable
end
```

2. Configure the inter-VDOM links:

```
config system vdom-link
    edit link-10-20-
    next
end
```

3. Configure the interface settings:

```
config system interface
    edit link-10-20-0
        set vdom "root"
        set vrf 10
        set ip 10.1.1.1/30
    next
    edit link-10-20-1
        set vdom "root"
        set vrf 20
        set ip 10.1.1.2/30
    next
end
```

4. Create the prefix lists:
   These objects define the subnet and mask that are leaked.

```
config router prefix-list
    edit VRF10_Route
        config rule
            edit 1
                set prefix 192.168.101.0 255.255.255.0
            next
        end
    next
    edit VRF20_Route
        config rule
            edit 1
                set prefix 192.168.102.0 255.255.255.0
```

```
            next
        end
    next
end
```

5. Create the route map:
   The route map can be used to group one or more prefix lists.

```
config router route-map
    edit "Leak_from_VRF10_to_VRF20"
        config rule
            edit 1
                set match-ip-address "VRF10_Route"
            next
        end
    next
    edit "Leak_from_VRF20_to_VRF10"
        config rule
            edit 1
                set match-ip-address "VRF20_Route"
            next
        end
    next
end
```

6. Configure the VRF leak in BGP, specifying a source VRF, destination VRF, an the route map to use:

```
config router bgp
    config vrf
        edit "10"
            config leak-target
                edit "20"
                    set route-map "Leak_from_VRF10_to_VRF20"
                    set interface "link-10-20-0"
                next
            end
        next
        edit "20"
            config leak-target
                edit "10"
                    set route-map "Leak_from_VRF20_to_VRF10"
                    set interface "link-10-20-1"
                next
            end
        next
    end
end
```

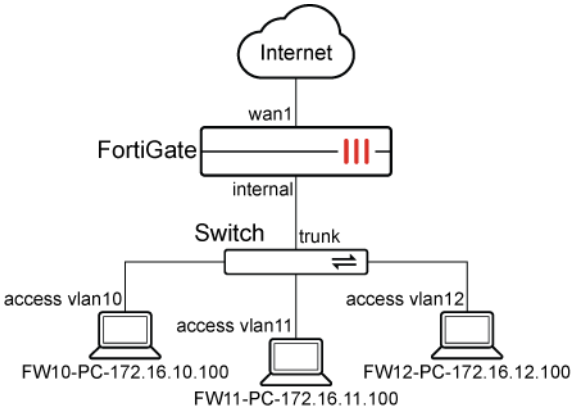
7. Create policies to allow traffic between the VRFs.
   Without a policy permitting traffic on the route between the VRFs, the VRFs are still isolated.

Route leaking between multiple VRFs

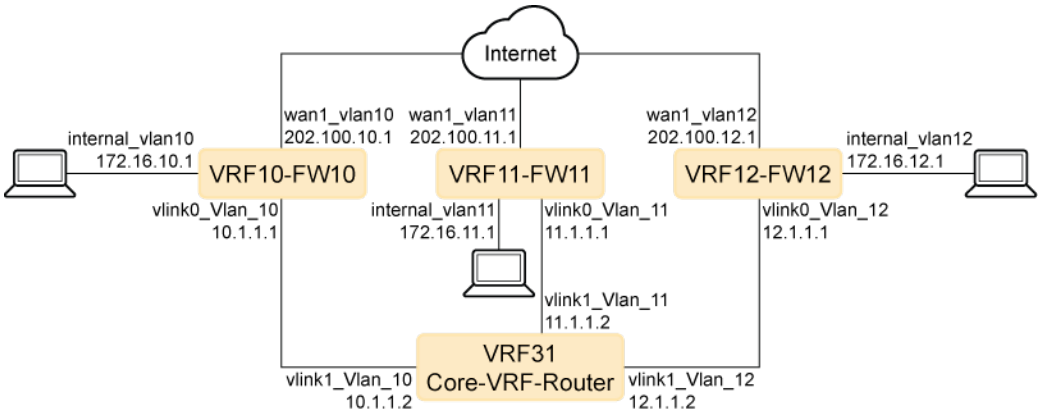
In this example, routing leaking between three VRFs in a star topology is configured. This allows the solution to be scaled to more VRFs without building full mesh, one-to-one connections between each pair of VRFs. VLAN subinterfaces are created on VDOM links to connect each VRF to the central VRF, allowing routes to be leaked from a VRF to the central VRF, and then to the other VRFs. Static routes are used for route leaking in this example.

For instructions on creating route leaking between two VRFs, see Route leaking between VRFs with BGP on page 802.

Physical topology:



Logical topology:



In this example, a specific route is leaked from each of the VRFs to each of the other VRFs. VLAN subinterfaces are created based on VDOM links to connect each VRF to the core VRF router.

NPU inter-VDOM links are used to accelerate traffic between VRFs. The setup could be configured by creating non-NPU VDOM links, but this is not recommended as these links are not offloaded to the NPU.

This example includes using the `single-vdom-npuvlink` option of the `config system global` command to enable NPU inter-VDOM links. You can also enable NPU inter-VDOM links by enabling multi-VDOM mode. Enabling multi-VDOM mode just to support VRF route leaking is not recommended.

To configure the FortiGate:

1. Enable NPU inter-VDOM links:

```
config system global
    set single-vdom-npuvlink enable
end
```

You can enter the following command to view the NPU inter-VDOM links that are now available:

```
# show system interface
config system interface
    ...
    edit "npu0_vlink0"
        set vdom "root"
        set type physical
    next
    edit "npu0_vlink1"
        set vdom "root"
        set type physical
    next
    ...
end
```

2. Allow interface subnets to use overlapping IP addresses:

```
config vdom
    edit root
        config system settings
            set allow-subnet-overlap enable
        end
```

3. Configure the inter-connecting VLAN subinterfaces between VRF based on VDOM-LINK:

```
config system interface
    edit "vlink0_Vlan_10"
        set vdom "root"
        set vrf 10
        set ip 10.1.1.1 255.255.255.252
        set allowaccess ping https ssh http
        set alias "vlink0_Vlan_10"
        set role lan
        set interface "npu0_vlink0"
        set vlanid 10
    next
    edit "vlink1_Vlan_10"
```

```
        set vdom "root"
        set vrf 31
        set ip 10.1.1.2 255.255.255.252
        set allowaccess ping https ssh http
        set alias "vlink1_Vlan_10"
        set role lan
        set interface "npu0_vlink1"
        set vlanid 10
    next
    edit "vlink0_Vlan_11"
        set vdom "root"
        set vrf 11
        set ip 11.1.1.1 255.255.255.252
        set allowaccess ping https ssh http
        set alias "vlink0_Vlan_11"
        set role lan
        set interface "npu0_vlink0"
        set vlanid 11
    next
    edit "vlink1_Vlan_11"
        set vdom "root"
        set vrf 31
        set ip 11.1.1.2 255.255.255.252
        set allowaccess ping https ssh http
        set alias "vlink1_Vlan_11"
        set role lan
        set interface "npu0_vlink1"
        set vlanid 11
    next
    edit "vlink0_Vlan_12"
        set vdom "root"
        set vrf 12
        set ip 12.1.1.1 255.255.255.252
        set allowaccess ping https ssh http
        set alias "vlink0_Vlan_12"
        set role lan
        set interface "npu0_vlink0"
        set vlanid 12
    next
    edit "vlink1_Vlan_12"
        set vdom "root"
        set vrf 31
        set ip 12.1.1.2 255.255.255.252
        set allowaccess ping https ssh http
        set alias "vlink1_Vlan_12"
        set role lan
        set interface "npu0_vlink1"
        set vlanid 12
    next
end
```

4. Configure a zone to allow intrazone traffic between VLANs in the central VRF:

```
config system zone
    edit "Core-VRF-Router"
        set intrazone allow
        set interface "vlink1_Vlan_10" "vlink1_Vlan_11" "vlink1_Vlan_12"
    next
end
```

5. Add allow policies for the VRF31 core router:

```
config firewall policy
    edit 0
        set name "any_to_core_vrf31"
        set srcintf "any"
        set dstintf "Core-VRF-Router"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic all
    next
    edit 0
        set name "core_vrf31_to_any"
        set srcintf "Core-VRF-Router"
        set dstintf "any"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic all
    next
end
```

6. Configure VRF10, VRF11, and VRF12 on the Internal and WAN VLAN sub-interfaces:

```
config system interface
    edit "Internal_VRF10"
        set vdom "root"
        set vrf 10
        set ip 172.16.10.1 255.255.255.0
        set allowaccess ping https ssh http
        set alias "Internal_VRF10"
        set role lan
        set interface "internal"
        set vlanid 10
    next
    edit "Internal_VRF11"
        set vdom "root"
        set vrf 11
        set ip 172.16.11.1 255.255.255.0
        set allowaccess ping https ssh http
```

```
        set alias "Internal_VRF11"
        set role lan
        set interface "internal"
        set vlanid 11
    next
    edit "Internal_VRF12"
        set vdom "root"
        set vrf 12
        set ip 172.16.12.1 255.255.255.0
        set allowaccess ping https ssh http
        set alias "Internal_VRF12"
        set role lan
        set interface "internal"
        set vlanid 12
    next
    edit "wan1_VRF10"
        set vdom "root"
        set vrf 10
        set ip 202.100.10.1 255.255.255.0
        set allowaccess ping
        set alias "wan1_VRF10"
        set role wan
        set interface "wan1"
        set vlanid 10
    next
    edit "wan1_VRF11"
        set vdom "root"
        set vrf 11
        set ip 202.100.11.1 255.255.255.0
        set allowaccess ping
        set alias "wan1_VRF11"
        set role wan
        set interface "wan1"
        set vlanid 11
    next
    edit "wan1_VRF12"
        set vdom "root"
        set vrf 12
        set ip 202.100.12.1 255.255.255.0
        set allowaccess ping
        set alias "wan1_VRF12"
        set role wan
        set interface "wan1"
        set vlanid 12
    next
end
```

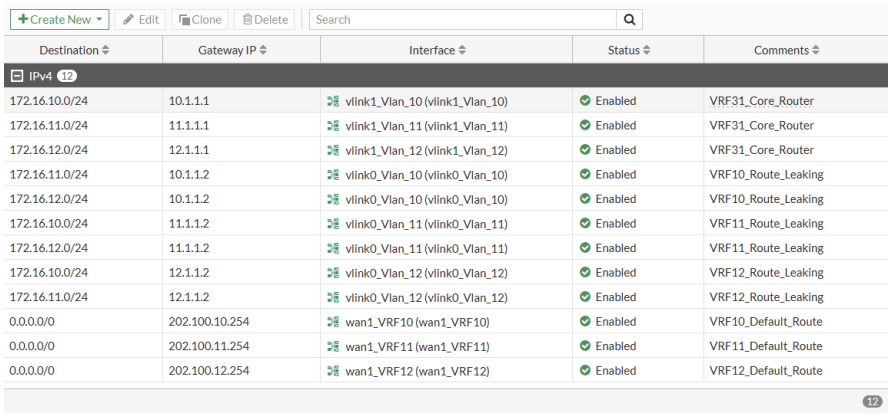
7. Configure static routing and route leaking between each VRF and Core-VRF-Router:

```
config router static
    edit 1
        set dst 172.16.10.0 255.255.255.0
```

```
        set gateway 10.1.1.1
        set device "vlink1_Vlan_10"
        set comment "VRF31_Core_Router"
    next
    edit 2
        set dst 172.16.11.0 255.255.255.0
        set gateway 11.1.1.1
        set device "vlink1_Vlan_11"
        set comment "VRF31_Core_Router"
    next
    edit 3
        set dst 172.16.12.0 255.255.255.0
        set gateway 12.1.1.1
        set device "vlink1_Vlan_12"
        set comment "VRF31_Core_Router"
    next
    edit 4
        set dst 172.16.11.0 255.255.255.0
        set gateway 10.1.1.2
        set device "vlink0_Vlan_10"
        set comment "VRF10_Route_Leaking"
    next
    edit 5
        set dst 172.16.12.0 255.255.255.0
        set gateway 10.1.1.2
        set device "vlink0_Vlan_10"
        set comment "VRF10_Route_Leaking"
    next
    edit 6
        set dst 172.16.10.0 255.255.255.0
        set gateway 11.1.1.2
        set device "vlink0_Vlan_11"
        set comment "VRF11_Route_Leaking"
    next
    edit 7
        set dst 172.16.12.0 255.255.255.0
        set gateway 11.1.1.2
        set device "vlink0_Vlan_11"
        set comment "VRF11_Route_Leaking"
    next
    edit 8
        set dst 172.16.10.0 255.255.255.0
        set gateway 12.1.1.2
        set device "vlink0_Vlan_12"
        set comment "VRF12_Route_Leaking"
    next
    edit 9
        set dst 172.16.11.0 255.255.255.0
        set gateway 12.1.1.2
        set device "vlink0_Vlan_12"
        set comment "VRF12_Route_Leaking"
```

```
    next
    edit 10
        set gateway 202.100.10.254
        set device "wan1_VRF10"
        set comment "VRF10_Default_Route"
    next
    edit 11
        set gateway 202.100.11.254
        set device "wan1_VRF11"
        set comment "VRF11_Default_Route"
    next
    edit 12
        set gateway 202.100.12.254
        set device "wan1_VRF12"
        set comment "VRF12_Default_Route"
    next
end
```

In the GUI, go to *Network* > *Static Routes* to view the static routes:



+Create New | Edit | Clone | Delete | Search

Destination	Gateway IP	Interface	Status	Comments
IPv4 12				
172.16.10.0/24	10.1.1.1	vlink1_Vlan_10 (vlink1_Vlan_10)	Enabled	VRF31_Core_Router
172.16.11.0/24	11.1.1.1	vlink1_Vlan_11 (vlink1_Vlan_11)	Enabled	VRF31_Core_Router
172.16.12.0/24	12.1.1.1	vlink1_Vlan_12 (vlink1_Vlan_12)	Enabled	VRF31_Core_Router
172.16.11.0/24	10.1.1.2	vlink0_Vlan_10 (vlink0_Vlan_10)	Enabled	VRF10_Route_Leaking
172.16.12.0/24	10.1.1.2	vlink0_Vlan_10 (vlink0_Vlan_10)	Enabled	VRF10_Route_Leaking
172.16.10.0/24	11.1.1.2	vlink0_Vlan_11 (vlink0_Vlan_11)	Enabled	VRF11_Route_Leaking
172.16.12.0/24	11.1.1.2	vlink0_Vlan_11 (vlink0_Vlan_11)	Enabled	VRF11_Route_Leaking
172.16.10.0/24	12.1.1.2	vlink0_Vlan_12 (vlink0_Vlan_12)	Enabled	VRF12_Route_Leaking
172.16.11.0/24	12.1.1.2	vlink0_Vlan_12 (vlink0_Vlan_12)	Enabled	VRF12_Route_Leaking
0.0.0.0/0	202.100.10.254	wan1_VRF10 (wan1_VRF10)	Enabled	VRF10_Default_Route
0.0.0.0/0	202.100.11.254	wan1_VRF11 (wan1_VRF11)	Enabled	VRF11_Default_Route
0.0.0.0/0	202.100.12.254	wan1_VRF12 (wan1_VRF12)	Enabled	VRF12_Default_Route

12

8. Configure firewall policies for VRF10, VRF11, and VRF12

```
config firewall policy
    edit 6
        set name "VRF10_to_Internet_Policy"
        set srcintf "Internal_VRF10"
        set dstintf "wan1_VRF10"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set nat enable
    next
    edit 7
        set name "VRF10_to_VRF_Leaking_Route"
        set srcintf "Internal_VRF10"
        set dstintf "vlink0_Vlan_10"
```

```
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic all
    next
    edit 8
        set name "VRF_Leaking_Route_to_VRF10"
        set srcintf "vlink0_Vlan_10"
        set dstintf "Internal_VRF10"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic all
    next
    edit 9
        set name "VRF11_to_Internet_Policy"
        set srcintf "Internal_VRF11"
        set dstintf "wan1_VRF11"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set nat enable
    next
    edit 10
        set name "VRF11_to_VRF_Leaking_Route"
        set srcintf "Internal_VRF11"
        set dstintf "vlink0_Vlan_11"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic all
    next
    edit 11
        set name "VRF_Leaking_Route_to_VRF11"
        set srcintf "vlink0_Vlan_11"
        set dstintf "Internal_VRF11"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic all
```

```
    next
    edit 12
        set name "VRF12_to_Internet_Policy"
        set srcintf "Internal_VRF12"
        set dstintf "wan1_VRF12"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set nat enable
    next
    edit 13
        set name "VRF12_to_VRF_Leaking_Route"
        set uuid 92bccf8e-b27b-51eb-3c56-6d5259af6299
        set srcintf "Internal_VRF12"
        set dstintf "vlink0_Vlan_12"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic all
    next
    edit 14
        set name "VRF_Leaking_Route_to_VRF12"
        set srcintf "vlink0_Vlan_12"
        set dstintf "Internal_VRF12"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic all
    next
end
```

In the GUI, go to *Policy & Objects* > *Firewall Policy* to view the policies.

To check the results:

1. On the FortiGate, check the routing table to see each VRF:

```
# get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default
```

```
Routing table for VRF=0
C       10.6.30.0/24 is directly connected, mgmt

Routing table for VRF=10
S*      0.0.0.0/0 [10/0] via 202.100.10.254, wan1_VRF10
C       10.1.1.0/30 is directly connected, vlink0_Vlan_10
C       172.16.10.0/24 is directly connected, Internal_VRF10
S       172.16.11.0/24 [10/0] via 10.1.1.2, vlink0_Vlan_10
S       172.16.12.0/24 [10/0] via 10.1.1.2, vlink0_Vlan_10
C       202.100.10.0/24 is directly connected, wan1_VRF10

Routing table for VRF=11
S*      0.0.0.0/0 [10/0] via 202.100.11.254, wan1_VRF11
C       11.1.1.0/30 is directly connected, vlink0_Vlan_11
S       172.16.10.0/24 [10/0] via 11.1.1.2, vlink0_Vlan_11
C       172.16.11.0/24 is directly connected, Internal_VRF11
S       172.16.12.0/24 [10/0] via 11.1.1.2, vlink0_Vlan_11
C       202.100.11.0/24 is directly connected, wan1_VRF11

Routing table for VRF=12
S*      0.0.0.0/0 [10/0] via 202.100.12.254, wan1_VRF12
C       12.1.1.0/30 is directly connected, vlink0_Vlan_12
S       172.16.10.0/24 [10/0] via 12.1.1.2, vlink0_Vlan_12
S       172.16.11.0/24 [10/0] via 12.1.1.2, vlink0_Vlan_12
C       172.16.12.0/24 is directly connected, Internal_VRF12
C       202.100.12.0/24 is directly connected, wan1_VRF12

Routing table for VRF=31
C       10.1.1.0/30 is directly connected, vlink1_Vlan_10
C       11.1.1.0/30 is directly connected, vlink1_Vlan_11
C       12.1.1.0/30 is directly connected, vlink1_Vlan_12
S       172.16.10.0/24 [10/0] via 10.1.1.1, vlink1_Vlan_10
S       172.16.11.0/24 [10/0] via 11.1.1.1, vlink1_Vlan_11
S       172.16.12.0/24 [10/0] via 12.1.1.1, vlink1_Vlan_12
```

2. From the FW10-PC:

```
# ifconfig ens32
ens32: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 172.16.10.100  netmask 255.255.255.0  broadcast 172.16.10.255
        inet6 fe80::dbed:c7fe:170e:e61c  prefixlen 64  scopeid 0x20<link>
        ether 00:0c:29:2a:3a:17  txqueuelen 1000  (Ethernet)
        RX packets 1632  bytes 160001 (156.2 KiB)
        RX errors 0  dropped 52  overruns 0  frame 0
        TX packets 2141  bytes 208103 (203.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```

```
# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         172.16.10.1     0.0.0.0         UG    100    0        0 ens32
```

```
172.16.10.0     0.0.0.0         255.255.255.0   U     100    0        0 ens32
192.168.122.0   0.0.0.0         255.255.255.0   U     0      0        0 virbr0
```

a. Ping a public IP address through VRF10:

```
# ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=113 time=4.33 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=113 time=4.17 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=113 time=4.04 ms
^C
--- 8.8.8.8 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2002ms
rtt min/avg/max/mdev = 4.049/4.188/4.336/0.117 ms
```

b. Ping the internet gateway through VRF10:

```
# ping 202.100.10.254
PING 202.100.10.254 (202.100.10.254) 56(84) bytes of data.
64 bytes from 202.100.10.254: icmp_seq=1 ttl=254 time=0.294 ms
64 bytes from 202.100.10.254: icmp_seq=2 ttl=254 time=0.225 ms
64 bytes from 202.100.10.254: icmp_seq=3 ttl=254 time=0.197 ms
^C
--- 202.100.10.254 ping statistics ---
3 packets transmitted, 3 received, 0% packet loss, time 2000ms
rtt min/avg/max/mdev = 0.197/0.238/0.294/0.044 ms
```

c. Ping the FW11-PC on VRF11 from VRF10:

```
# ping 172.16.11.100
PING 172.16.11.100 (172.16.11.100) 56(84) bytes of data.
64 bytes from 172.16.11.100: icmp_seq=1 ttl=61 time=0.401 ms
64 bytes from 172.16.11.100: icmp_seq=2 ttl=61 time=0.307 ms
64 bytes from 172.16.11.100: icmp_seq=3 ttl=61 time=0.254 ms
64 bytes from 172.16.11.100: icmp_seq=4 ttl=61 time=0.277 ms
64 bytes from 172.16.11.100: icmp_seq=5 ttl=61 time=0.262 ms
^C
--- 172.16.11.100 ping statistics ---
5 packets transmitted, 5 received, 0% packet loss, time 3999ms
rtt min/avg/max/mdev = 0.254/0.300/0.401/0.054 ms
```

3. On the FortiGate, sniff traffic between VRF10 and VRF11:

```
# diagnose sniffer packet any "icmp and host 172.16.11.100" 4 l 0
interfaces=[any]
filters=[icmp and host 172.16.11.100]
10.086656 Internal_VRF10 in 172.16.10.100 -> 172.16.11.100: icmp: echo request
10.086705 vlink0_Vlan_10 out 172.16.10.100 -> 172.16.11.100: icmp: echo request
10.086706 npu0_vlink0 out 172.16.10.100 -> 172.16.11.100: icmp: echo request

10.086711 vlink1_Vlan_10 in 172.16.10.100 -> 172.16.11.100: icmp: echo request
10.086739 vlink1_Vlan_11 out 172.16.10.100 -> 172.16.11.100: icmp: echo request
10.086740 npu0_vlink1 out 172.16.10.100 -> 172.16.11.100: icmp: echo request
```

```
10.086744 vlink0_Vlan_11 in 172.16.10.100 -> 172.16.11.100: icmp: echo request
10.086929 Internal_VRF11 out 172.16.10.100 -> 172.16.11.100: icmp: echo request
10.086930 internal out 172.16.10.100 -> 172.16.11.100: icmp: echo request

10.087053 Internal_VRF11 in 172.16.11.100 -> 172.16.10.100: icmp: echo reply
10.087061 vlink0_Vlan_11 out 172.16.11.100 -> 172.16.10.100: icmp: echo reply
10.087062 npu0_vlink0 out 172.16.11.100 -> 172.16.10.100: icmp: echo reply

10.087066 vlink1_Vlan_11 in 172.16.11.100 -> 172.16.10.100: icmp: echo reply
10.087071 vlink1_Vlan_10 out 172.16.11.100 -> 172.16.10.100: icmp: echo reply
10.087072 npu0_vlink1 out 172.16.11.100 -> 172.16.10.100: icmp: echo reply

10.087076 vlink0_Vlan_10 in 172.16.11.100 -> 172.16.10.100: icmp: echo reply
10.087176 Internal_VRF10 out 172.16.11.100 -> 172.16.10.100: icmp: echo reply
10.087177 internal out 172.16.11.100 -> 172.16.10.100: icmp: echo reply
^C
20 packets received by filter
0 packets dropped by kernel
```

VRF with IPv6

IPv6 routes support VRF. Static, connected, OSPF, and BGP routes can be isolated in different VRFs. BGP IPv6 routes can be leaked from one VRF to another.

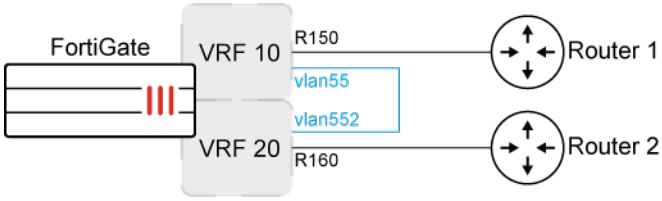
```
config router bgp
    config vrf6
        edit <origin vrf-id>
            config leak-target
                edit <target vrf-id>
                    set route-map <route-map>
                    set interface <interface>
                next
            end
        next
    end
end
```

The origin or target VRF ID is an integer value from 0 - 31.

```
config router static6
    edit <id>
        set vrf <vrf-id>
    next
end
```

Using a VRF leak on BGP

In this example, the route 2000:5:5:5::/64 learned from Router 1 is leaked to VRF 20 through the interface vlan552. Conversely, the route 2009:3:3:3::/64 learned from Router 2 is leaked to VRF 10 through interface vlan55.



To configure VRF leaking in BGP:

1. Configure the BGP neighbors:

```
config router bgp
    set as 65412
    config neighbor
        edit "2000:10:100:1::1"
            set activate disable
            set remote-as 20
            set update-source "R150"
        next
        edit "2000:10:100:1::5"
            set activate disable
            set soft-reconfiguration enable
            set interface "R160"
            set remote-as 20
        next
    end
end
```

2. Configure the VLAN interfaces:

```
config system interface
    edit "vlan55"
        set vdom "root"
        set vrf 10
        set ip 55.1.1.1 255.255.255.0
        set device-identification enable
        set role lan
        set snmp-index 51
        config ipv6
            set ip6-address 2000:55::1/64
        end
        set interface "npu0_vlink0"
        set vlanid 55
    next
    edit "vlan552"
        set vdom "root"
```

```
        set vrf 20
        set ip 55.1.1.2 255.255.255.0
        set device-identification enable
        set role lan
        set snmp-index 53
        config ipv6
            set ip6-address 2000:55::2/64
        end
        set interface "npu0_vlink1"
        set vlanid 55
    next
end
```

3. Configure the IPv6 prefixes:

```
config router prefix-list6
    edit "1"
        config rule
            edit 1
                set prefix6 2000:5:5:5::/64
                unset ge
                unset le
            next
        end
    next
    edit "2"
        config rule
            edit 1
                set prefix6 2009:3:3:3::/64
                unset ge
                unset le
            next
        end
    next
end
```

4. Configure the route maps:

```
config router route-map
    edit "from106"
        config rule
            edit 1
                set match-ip6-address "1"
            next
        end
    next
    edit "from206"
        config rule
            edit 1
                set match-ip6-address "2"
            next
        end
```

```
    next
end
```

5. Configure the IPv6 route leaking (leak route 2000:5:5:5::/64 learned from Router 1 to VRF 20, then leak route 2009:3:3:3::/64 learned from Router 2 to VRF 10):

```
config router bgp
    config vrf6
        edit "10"
            config leak-target
                edit "20"
                    set route-map "from106"
                    set interface "vlan55"
                next
            end
        next
        edit "20"
            config leak-target
                edit "10"
                    set route-map "from206"
                    set interface "vlan552"
                next
            end
        next
    end
end
```

To verify the VRF leaking:

1. Check the routing table before the leak:

```
# get router info6 routing-table bgp
Routing table for VRF=10
B       2000:5:5:5::/64 [20/0] via fe00::2000:0000:0000:00, R150, 00:19:45

Routing table for VRF=20
B       2008:3:3:3::/64 [20/0] via fe00::3000:0000:0000:00, R160, 00:18:49
B       2009:3:3:3::/64 [20/0] via fe00::3000:0000:0000:00, R160, 00:18:49
```

2. Check the routing table after the leak:

```
# get router info6 routing-table bgp
Routing table for VRF=10
B       2000:5:5:5::/64 [20/0] via fe00::2000:0000:0000:0, R150, 00:25:45
B       2009:3:3:3::/64 [20/0] via fe80::10:0000:0000:4245, vlan55, 00:00:17

Routing table for VRF=20
B       2000:5:5:5::/64 [20/0] via fe80::10:0000:0000:4244, vlan552, 00:00:16
B       2008:3:3:3::/64 [20/0] via fe00::3000:0000:0000:00, R160, 00:24:49
B       2009:3:3:3::/64 [20/0] via fe00::3000:0000:0000:00, R160, 00:24:49
```

Using VRF on a static route

In this example, a VRF is defined on static route 22 so that it will only appear in the VRF 20 routing table.

To configure the VRF on the static route:

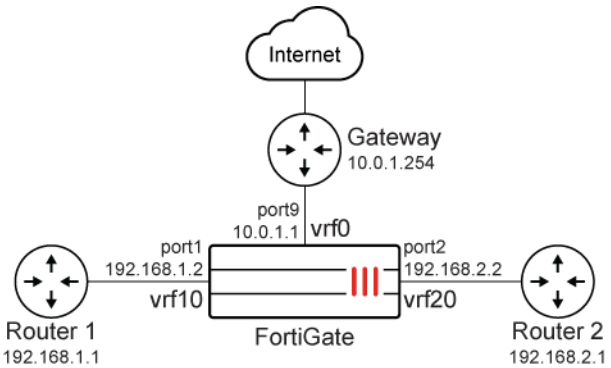
```
config router static6
    edit 22
        set dst 2010:2:2:2::/64
        set blackhole enable
        set vrf 20
    next
end
```

IBGP and EBGP support in VRF

Support is included for internal and external border gateway protocols (IBGP and EBGP) in virtual routing and forwarding (VRF).

FortiGate can establish neighbor connections with other FortiGates or routers, and the learned routes are put into different VRF tables according to the neighbor's settings.

This example uses the following topology:



- BGP routes learned from the Router1 neighbor are put into vrf10.
- BGP routes learned from the Router2 neighbor are put into vrf20.

To configure this example:

```
config system interface
    edit port1
        set vrf 10
    next
    edit port2
        set vrf 20
    next
end
```

```
config router bgp
    config neighbor
        edit "192.168.1.1"
            set update-source port1
        next
        edit "192.168.2.1"
            set interface port2
        next
    end
end
```

Results

Using the above topology:

- Both Router1 and Router2 establish OSPF and BGP neighbor with the FortiGate.
- Router1 advertises 10.10.1.0/24 into OSPF and 10.10.2.0/24 into BGP.
- Router2 advertises 20.20.1.0/24 into OSPF and 20.20.2.0/24 into BGP.

When port1 and port2 have not set VRF, all of the routing is in VRF=0:

```
# get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [5/0] via 10.0.1.254, port9
C       10.0.1.0/24 is directly connected, port9
O       10.10.1.0/24 [110/10] via 192.168.1.1, port1, 00:18:31
B       10.10.2.0/24 [20/200] via 192.168.1.1, port1, 00:01:31
O       20.20.1.0/22 [110/10] via 192.168.2.1, port2, 00:19:05
B       20.20.2.0/24 [20/200] via 192.168.2.1, port2, 00:01:31
C       192.168.1.0/24 is directly connected, port1
C       192.168.2.0/24 is directly connected, port2
```

After VRF is set for BGP, BGP routes are added to the VRF tables along with OSPF and connected routes:

```
# get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [5/0] via 10.0.1.254, port9
C       10.0.1.0/24 is directly connected, port9
```

```
Routing table for VRF=10
O       10.10.1.0/24 [110/10] via 192.168.1.1, port1, 00:18:31
B       10.10.2.0/24 [20/200] via 192.168.1.1, port1, 00:01:31
C       192.168.1.0/24 is directly connected, port1

Routing table for VRF=20
O       20.20.1.0/22 [110/10] via 192.168.2.1, port2, 00:19:05
B       20.20.2.0/24 [20/200] via 192.168.2.1, port2, 00:01:31
C       192.168.2.0/24 is directly connected, port2
```

BGP neighbor groups

This feature is also supported in the BGP neighbor groups. For example:

```
config router bgp
    config neighbor-group
        edit "FGT"
            set update-source "port1"
        next
    end
    config neighbor-range
        edit 1
            set prefix 172.16.201.0 255.255.255.0
            set neighbor-group "FGT"
        next
    end
end
```

Note that the `set interface` command is not supported.

Support cross-VRF local-in and local-out traffic for local services

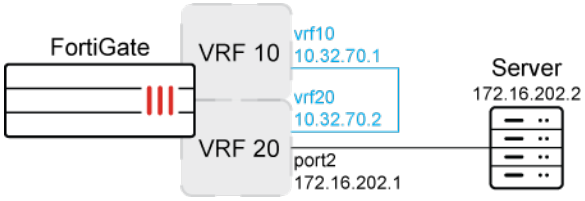
When local-out traffic such as SD-WAN health checks, SNMP, syslog, and so on are initiated from an interface on one VRF and then pass through interfaces on another VRF, the reply traffic will be successfully forwarded back to the original VRF.

VRF 0 is a special VRF. By default, all routing is done in VRF 0. So all routes in different VRFs, such as VRF 10 or VRF 20, will all be included in VRF 0. VRF 0 cannot be used in the cross-VRF case.

For local-in and local-out traffic, all routes relating to one VRF are isolated from other VRFs so interfaces in one VRF cannot reach interfaces in a different VRF, except for VRF 0.

Example

In this example, there is an NPU inter-VDOM link that is configured on the root VDOM. Two VLANs, vrf10 and vrf20, are created on either ends of the NPU inter-VDOM link, each belonging to a different VRF.



When pinging from the vrf10 interface in VRF 10 to the destination server 172.16.202.2, since there is a single static route for VRF 10 with a gateway of vrf20/10.32.70.2, traffic is sent to the next hop and subsequently routed through port12 to the server.

As seen in the sniffer trace, the ICMP replies are received on port12 in VRF 20, then pass through vrf20, and are ultimately forwarded back to vrf10 in VRF 10. The traffic flow demonstrates that local-out traffic sourced from one VRF passing through another VRF can return back to the original VRF.

To configure cross-VRF local-out traffic for local services:

1. Configure the interfaces:

```
config system interface
    edit "vrf10"
        set vdom "root"
        set vrf 10
        set ip 10.32.70.1 255.255.255.0
        set allowaccess ping
        set device-identification enable
        set role lan
        set snmp-index 35
        set interface "npu0_vlink0"
        set vlanid 22
    next
    edit "vrf20"
        set vdom "root"
        set vrf 20
        set ip 10.32.70.2 255.255.255.0
        set allowaccess ping
        set device-identification enable
        set role lan
        set snmp-index 36
        set interface "npu0_vlink1"
        set vlanid 22
    next
    edit "port12"
        set vdom "root"
        set vrf 20
        set ip 172.16.202.1 255.255.255.0
        set allowaccess ping https ssh snmp http telnet fgfm radius-acct probe-response fabric
```

```
ftm speed-test
        set type physical
        set alias "TO_FGT_D_port22"
        set snmp-index 14
        config ipv6
            set ip6-address 2003:172:16:202::1/64
            set ip6-allowaccess ping
        end
    next
end
```

2. Configure the firewall policy:

```
config firewall policy
    edit 1
        set srcintf "vrf20"
        set dstintf "port12"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
    next
end
```

3. Configure the static route:

```
config router static
    edit 2
        set gateway 10.32.70.2
        set distance 3
        set device "vrf10"
    next
end
```

To test the configuration:

1. Execute a ping from the vrf10 interface in VRF 10 to the destination server (172.16.202.2):

```
# execute ping-options interface vrf10
# execute ping 172.16.202.2
PING 172.16.202.2 (172.16.202.2): 56 data bytes
64 bytes from 172.16.202.2: icmp_seq=0 ttl=254 time=0.1 ms
64 bytes from 172.16.202.2: icmp_seq=1 ttl=254 time=0.0 ms

--- 172.16.202.2 ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.0/0.0/0.1 ms
```

2. Run a sniffer trace on 172.16.202.2 for ICMP:

```
# diagnose sniffer packet any "host 172.16.202.2 and icmp" 4
interfaces=[any]
filters=[host 172.16.202.2 and icmp]
3.393920 vrf10 out 10.32.70.1 -> 172.16.202.2: icmp: echo request
3.393922 npu0_vlink0 out 10.32.70.1 -> 172.16.202.2: icmp: echo request
3.393927 vrf20 in 10.32.70.1 -> 172.16.202.2: icmp: echo request
3.393943 port12 out 10.32.70.1 -> 172.16.202.2: icmp: echo request
3.393977 port12 in 172.16.202.2 -> 10.32.70.1: icmp: echo reply
3.393987 vrf20 out 172.16.202.2 -> 10.32.70.1: icmp: echo reply
3.393988 npu0_vlink1 out 172.16.202.2 -> 10.32.70.1: icmp: echo reply
3.393993 vrf10 in 172.16.202.2 -> 10.32.70.1: icmp: echo reply
4.393941 vrf10 out 10.32.70.1 -> 172.16.202.2: icmp: echo request
4.393942 npu0_vlink0 out 10.32.70.1 -> 172.16.202.2: icmp: echo request
4.393948 vrf20 in 10.32.70.1 -> 172.16.202.2: icmp: echo request
4.393957 port12 out 10.32.70.1 -> 172.16.202.2: icmp: echo request
4.393980 port12 in 172.16.202.2 -> 10.32.70.1: icmp: echo reply
4.393987 vrf20 out 172.16.202.2 -> 10.32.70.1: icmp: echo reply
4.393987 npu0_vlink1 out 172.16.202.2 -> 10.32.70.1: icmp: echo reply
4.393994 vrf10 in 172.16.202.2 -> 10.32.70.1: icmp: echo reply
```

Support specific VRF ID for local-out traffic

A Virtual Routing and Forwarding (VRF) instance can be specified for local-out traffic to provide traffic segregation, optimized routing, and enhanced policy enforcement.

The following configuration commands include the option to set a VRF instance number:

```
config system interface
    edit "port2"
        set dhcp-relay-vrf-select
    next
end
```

`dhcp-relay-vrf-select <integer>`	VRF ID used for connection to sever. Set VRF instance number (0 to 511, default = 0).

```
config system settings
  set dhcp-proxy-vrf-select <integer>
end
```

`dhcp-proxy-vrf-select <integer>`	VRF ID used for connection to sever. Set VRF instance number (0 to 511, default = 0).

```
config system dns
    set vrf-select <integer>
end

config system fortiguard
    set vrf-select <integer>
```

```
end

config system snmp community
  edit <id>
    config hosts
        edit <id>
          set vrf-select <integer>
        next
    config hosts6
        edit <id>
            set vrf-select <integer>
        next
    next
end

config system snmp user
    edit <name>
        set vrf-select <integer>
    next
end

config system email-server
    set vrf-select <integer>
end

config system vdom-dns
    set vrf-select <integer>
end

config system external-resource
    set vrf-select <integer>
end

config system fortindr
    set vrf-select <integer>
end

config system central-management
    set vrf-select <integer>
end

config system netflow|vdom-netflow
    config collectors
        edit <id>
            set vrf-select
        next
    next
end

config system ntp
    config ntpserver
        edit <id>
```

```
            set vrf-select <integer>
        next
    next
end

config system fortisandbox
    set vrf-select <integer>
end

config log syslogd setting
    set vrf-select <integer>
end

config log fortiguard setting
    set vrf-select <integer>
end

config log disk setting
    set vrf-select <integer>
end

config log tacacs+accounting setting
    set vrf-select <integer>
end

config log fortianalyzer|fortianalyzer2|fortianalyzer3|fortianalyzer-cloud setting
    set vrf-select <integer>
end

config system dns-database
    edit <name>
        set vrf-select <integer>
    end
end

config user external-identity-provider
    edit <name>
        set vrf-select <integer>
    end
end

config user fsso
    edit <name>
        set vrf-select <integer>
    end
end

config user ldap
    edit <name>
        set vrf-select <integer>
    end
end
```

```
config user radius
    edit <name>
        set vrf-select <integer>
        config accounting-server
            edit <id>
                set vrf-select <integer>
            end
        end
    end
end

config user tacacs+
    edit <name>
        set vrf-select <integer>
    end
end

config vpn certificate setting
    set vrf-select <integer>
end

config vpn kmip-server
    edit <name>
        set vrf-select <integer>
    end
end
```

`vrf-select <integer>`	VRF ID used for connection to sever. Set VRF instance number (0 to 511, default = 0).

The following execute commands now include the option to specify a VRF instance number:

```
# execute ping-options vrf <integer>
# execute ping6-options vrf <integer>
# execute traceroute-options vrf <integer>
```

`vrf <integer>`	VRF ID (0 to 511).

The following execute command now includes the option to specify a VRF instance number:

```
# execute tracert6 -v <integer>
```

`-v <integer>`	VRF ID (0 to 511).

The following diagnose commands now include the option to specify a VRF instance number:

```
# diagnose ip proute match <dst> <src> <iif> <proto> <dport> <sport> <vrf>
# diagnose ipv6 proute match <dst> <src> <iif> <proto> <dport> <sport> <vrf>
# diagnose test authserver radius-direct <server_name or IP> <port number (0 default port)> <udp |
tcp | tls> <secret> <pap | chap | mschap | mschap2> <vrf> <user> <password
```

`vrf <integer>`	VRF ID (0 to 511).

Example 1: RADIUS server and VRF ID

In this example, the local-out traffic flows through the VRF interface to its destination. The VRF server can be reached by port1 (VRF 22), but not port2 (VRF 0).

To configure a VRF ID for a RADIUS server:

1. Specify a VRF ID for the RADIUS server:

```
config user radius
    edit "FAC"
        set server "192.168.100.129"
        set secret ftntxxxxxx
        set password-renewal disable
        set vrf-select 22
    next
end
```

2. Get the static routing table:

```
# get router info routing-table static
Routing table for VRF=0
S*      0.0.0.0/0 [5/0] via 10.100.1.249, npu0_vlink0, [1/0]
S       192.168.100.0/24 [10/0] via 172.16.205.11, port2, [1/0]

Routing table for VRF=22
S*      0.0.0.0/0 [2/0] via 172.16.200.254, port1, [1/0]
```

Example 2: ping command and VRF ID

In this example, the local-out traffic flows through the VRF interface to its destination. IP address 3.3.3.3 can be reached by port1 (VRF 22).

To use a VRF ID with ping:

1. Get the static routing table:

```
# get router info routing-table static
Routing table for VRF=0
S*      0.0.0.0/0 [5/0] via 10.100.1.249, npu0_vlink0, [1/0]
S       3.3.3.0/24 [10/0] via 172.16.205.11, port2, [1/0]
S       192.168.100.0/24 [10/0] via 172.16.205.11, port2, [1/0]

Routing table for VRF=22
S*      0.0.0.0/0 [2/0] via 172.16.200.254, port1, [1/0]
```

2. Specify a VRF ID for the interface with ping access allowed:

```
config system interface
    edit "loop2"
        set vdom "root"
        set vrf 22
        set ip 1.1.1.2 255.255.255.255
        set allowaccess ping https http
        set type loopback
        set snmp-index 84
    next
end
```

3. Execute the ping command:

```
# execute ping-options vrf 22

# execute ping-options source 1.1.1.2

# execute ping 3.3.3.3
PING 3.3.3.3 (3.3.3.3): 56 data bytes
64 bytes from 3.3.3.3: icmp_seq=0 ttl=255 time=0.1 ms
```

Example 3: traceroute command and VRF ID

The VRF ID can be used with the `traceroute` command.

To use a VRF ID with traceroute:

1. Get the routing table details for IP address 3.3.3.3:

```
# get router info routing-table details 3.3.3.3

Routing table for VRF=0
Routing entry for 3.3.3.3/32
  Known via "ospf", distance 110, metric 10200, best
  Last update 05:53:57 ago
  * vrf 0 172.16.203.2, via agg1


Routing table for VRF=22
Routing entry for 3.3.3.3/32
  Known via "ospf", distance 110, metric 11000, best
  Last update 05:31:31 ago
  * vrf 22 172.16.200.4, via port1
  * vrf 22 172.16.200.40, via port1
```

2. Execute a traceroute for VRF 22:

```
# execute traceroute 3.3.3.3
VRF 22, traceroute to 3.3.3.3 (3.3.3.3), 32 hops max, 3 probe packets per hop, 84 byte packets
 1  3.3.3.3  0.135 ms  0.090 ms  0.073 ms
```

Example 4: diagnose command and VRF ID

Like the other examples in this topic, port1 is in VRF 22, and port2 is in VRF 0. This example demonstrates how the `diagnose ip proute match` command works with and without a specified VRF ID.

To use a VRF ID with a diagnose command:

1. Configure the router policy for port1 and port2:

```
config router policy
    edit 1
        set dst "23.23.23.23/255.255.255.255"
        set gateway 172.16.200.55
        set output-device "port1"
    next
    edit 2
        set dst "23.23.23.23/255.255.255.255"
        set gateway 172.16.205.11
        set output-device "port2"
    next
end
```

2. Run the `diagnose firewall proute list` command:

```
# diagnose firewall proute list
list route policy info(vf=root):

id=1(0x01) dscp_tag=0xfc 0xfc flags=0x0 tos=0x00 tos_mask=0x00 protocol=0 port=src(0->0):dst
(0->0) iif=0(any)
path(1): oif=7(port1) gwy=172.16.200.55
source wildcard(1): 0.0.0.0/0.0.0.0
destination wildcard(1): 23.23.23.23/255.255.255.255
hit_count=1 rule_last_used=2024-10-11 15:40:23

id=2(0x02) dscp_tag=0xfc 0xfc flags=0x0 tos=0x00 tos_mask=0x00 protocol=0 port=src(0->0):dst
(0->0) iif=0(any)
path(1): oif=8(port2) gwy=172.16.205.11
source wildcard(1): 0.0.0.0/0.0.0.0
destination wildcard(1): 23.23.23.23/255.255.255.255
hit_count=2 rule_last_used=2024-10-11 15:40:39
```

3. Run the `diagnose ip route match` command without a VRF ID:
   When the VRF is not specified, VRF 0 is used, and it will match port2.

```
# diagnose ip proute match 23.23.23.23 2.2.2.22 port4 6 23 23
dst=23.23.23.23 src=2.2.2.22 smac=00:00:00:00:00:00 iif=10 protocol=6 sport=23 dport=23 vrf=-1
id=00000002 type=Policy Route
seq-num=2 oif=8(port2)flags=0x0
```

4. Run the `diagnose ip route match` command with a VRF ID:
   When VRF 22 is specified, it will match policy route with port1, which is in VRF 22:

```
# diagnose ip proute match 23.23.23.23 2.2.2.22 port4 6 23 23 22
dst=23.23.23.23 src=2.2.2.22 smac=00:00:00:00:00:00 iif=10 protocol=6 sport=23 dport=23 vrf=22
id=00000001 type=Policy Route
seq-num=1 oif=7(port1)flags=0x0
```

NetFlow

NetFlow allows you to collect IP network traffic statistics for an interface, and then export those statistics for analysis. NetFlow samplers, that sample every packet, are configured per interface. Full NetFlow is supported through the information maintained in the firewall session.

To configure NetFlow:

```
config system netflow
    set active-flow-timeout <integer>
    set inactive-flow-timeout <integer>
    set template-tx-timeout <integer>
    set template-tx-counter <integer>
    set session-cache-size {min | default | max}
    config collectors
        edit <id>
            set collector-ip <IP address>
            set collector-port <port>
            set source-ip <IP address>
            set interface-select-method {auto | sdwan | specify}
            set interface <interface>
        next
    end
end
```

The `source-ip-interface` and `source-ip` commands are unavailable for NetFlow configurations when `ha-direct` is enabled. (See config system ha in the CLI Reference guide).

The `source-ip-interface` and `source-ip` commands are also mutually exclusive; they cannot be used at the same time, but one or the other can be used together with the *interface-select-method* command

`active-flow-timeout <integer>`	Timeout to report active flows, in seconds (60 - 3600, default = 1800).
`inactive-flow-timeout <integer>`	Timeout for periodic report of finished flows, in seconds (10 - 600, default = 15).
`template-tx-timeout <integer>`	Timeout for periodic template flowset transmission, in seconds (60 - 86400, default = 1800).

`template-tx-counter <integer>`	Counter of flowset records, before resending a template flowset record (10 - 6000, default = 20).
`session-cache-size {min | default | max}`	Maximum RAM usage allowed for NetFlow session cache. • `min`: Up to 0.5% of system RAM. • `default`: Up to 1% of system RAM. • `max`: Up to 2% of system RAM.
`collector-ip <ip>`	Collector IPv4 or IPv6 address.
`collector-port <port>`	NetFlow collector port number (0 - 65535).
`source-ip <ip>`	Source IPv4 or IPv6 address, for communication with the NetFlow agent.
`interface-select-method {auto | sdwan | specify}`	Routing of the NetFlow messages is determined by the selected method. If neither `source-ip-interface` nor `source-ip` is configured, then the source address of the message is the IP address of the interface selected by the interface select method. See Local out traffic on page 1468 for details.
`source-ip-interface <name>`	Utilize the IP address of the specified interface as the source when sending out the NetFlow messages. Routing of the messages does not change based on this setting. The `source-ip-interface` is unavailable for NetFlow configurations when FortiGate is in transparent VDOM mode.
`interface <interface>`	The outgoing interface to reach the server.

To configure NetFlow in a specific, non-management VDOM:

```
config vdom
    edit <vdom>
        config system vdom-netflow
            set vdom-netflow enable
            config collectors
                edit <id>
                    set collector-ip <IP address>
                    set collector-port <port>
                    set source-ip <IP address>
                    set interface-select-method {auto | sdwan | specify}
                    set interface <interface>
                next
            end
        end
    next
end
```

The `vdom-netflow` command is only available for non-management VDOMs. The management VDOM utilizes the global NetFlow settings.

To configure a NetFlow sampler on an interface:

```
config system interface
    edit <interface>
        set netflow-sampler {disable | tx | rx | both}
    next
end
```

disable	Disable the NetFlow protocol on this interface (default).
tx	Monitor transmitted traffic on this interface.
rx	Monitor received traffic on this interface.
both	Monitor transmitted/received traffic on this interface.

To configure a source interface for syslog:

1. Configure the interface:

```
config system interface
    edit "loopback"
        set vdom "vdom1"
        set ip 10.10.10.2 255.255.255.0
        set allowaccess ping
        set type loopback
    next
end
```

2. Configure the syslog device:

```
config log syslogd setting
    set status enable
    set server "172.16.200.55"
    set facility local6
    set source-ip-interface "loopback"
end
```

Verification and troubleshooting

If data are not seen on the NetFlow collector after it has been configured, use the following sniffer commands to verify if the FortiGate and the collector are communicating:

- By collector port:

```
# diagnose sniffer packet 'port <collector-port>'  6 0 a
```

- By collector IP address:

```
# diagnose sniffer packet 'host <collector-ip>' 6 0 a
```

NetFlow uses the sflow daemon. The current NetFlow configuration can be viewed using test level 3 or 4:

```
# diagnose test application sflowd 3
```

```
# diagnose test application sflowd 4
Netflow Cache Stats:
vdoms=1 Collectors=1 Cached_intf=2 Netflow_enabled_intf=1 Live_sessions=0 Session cache max
count:71950
```

NetFlow templates

NetFlow uses templates to capture and categorize the data that it collects. FortiOS supports the following NetFlow templates:

Name	Template ID	Description
STAT_OPTIONS	256	Statistics information about exporter
SAMPLER_OPTIONS	257	Sampling information about exporter
APP_ID_OPTIONS	258	Application information
IPV4	259	No NAT IPv4 traffic
IPV6	260	No NAT IPv6 traffic
ICMP4	261	No NAT ICMPv4 traffic
ICMP6	262	No NAT ICMPv6 traffic
IPV4_NAT	263	Source/Destination NAT IPv4 traffic
IPV4_AF_NAT	264	AF NAT IPv4 traffic (4->6)
IPV6_NAT	265	Source/Destination NAT IPv6 traffic
IPV6_AF_NAT	266	AF NAT IPv6 traffic (6->4)
ICMP4_NAT	267	Source/Destination NAT ICMPv4 traffic
ICMP4_AF_NAT	268	AF NAT ICMPv4 traffic (4->6)
ICMP6_NAT	269	Source/Destination NAT ICMPv6 traffic
ICMPv6_AF_NAT	270	AF NAT ICMPv6 traffic (6->4)

256 - STAT_OPTIONS

Description	Statistics information about exporter
Scope Field Count	1
Data Field Count	5
Option Scope Length	4

Option Length	28
Padding	0000

Scope fields

Field #	Field	Type	Length
1	System	System (1)	2

Data fields

Field #	Field	Type	Length
1	TOTAL_BYTES_EXP	TOTAL_BYTES_EXP (40)	8
2	TOTAL_PKTS_EXP	TOTAL_PKTS_EXP (41)	8
3	TOTAL_FLOWS_EXP	TOTAL_FLOWS_EXP (42)	8
4	FLOW_ACTIVE_TIMEOUT	FLOW_ACTIVE_TIMEOUT (36)	2
5	FLOW_INACTIVE_TIMEOUT	FLOW_INACTIVE_TIMEOUT (37)	2
6	SAMPLING_INTERVAL	SAMPLING_INTERVAL (34)	4
7	SAMPLING_ALGORITHM	SAMPLING_ALGORITHM (35)	1

257 - SAMPLER_OPTIONS

Description	Sampling information about exporter
Scope Field Count	1
Data Field Count	6
Option Scope Length	4
Option Length	24
Padding	0000

Scope fields

Field #	Field	Type	Length
1	System	System (1)	2

Data fields

Field #	Field	Type	Length
1	FLOW_SAMPLER_ID	FLOW_SAMPLER_ID (48)	1
2	FLOW_SAMPLER_MODE	FLOW_SAMPLER_MODE (49)	1
3	FLOW_SAMPLER_RANDOM_INTERVAL	FLOW_SAMPLER_RANDOM_INTERVAL (50)	4
4	SAMPLING_INTERVAL	SAMPLING_INTERVAL (34)	4
5	SAMPLING_ALGORITHM	SAMPLING_ALGORITHM (35)	1
6	FLOW_SAMPLER_NAME	FLOW_SAMPLER_NAME (84)	16

258 - APP_ID_OPTIONS

Description	Application information
Scope Field Count	1
Data Field Count	4
Option Scope Length	4
Option Length	16
Padding	0000

Scope fields

Field #	Field	Type	Length
1	System	System (1)	2

Data fields

Field #	Field	Type	Length
1	APPLICATION_ID	APPLICATION_ID (95)	9
2	APPLICATION_NAME	APPLICATION_NAME (96)	64
3	APPLICATION_DESC	APPLICATION_DESC (94)	64
4	applicationCategoryName	applicationCategoryName (372)	32

259 - IPV4

Description	No NAT IPv4 traffic
Data Field Count	22

Data fields

Field #	Field	Type	Length
1	BYTES	BYTES (1)	8
2	OUT_BYTES	OUT_BYTES (23)	8
3	PKTS	PKTS (2)	4
4	OUT_PKTS	OUT_PKTS (24)	4
5	FIRST_SWITCHED	FIRST_SWITCHED (22)	4
6	LAST_SWITCHED	LAST_SWITCHED (21)	4
7	L4_SRC_PORT	L4_SRC_PORT (7)	2
8	L4_DST_PORT	L4_DST_PORT (11)	2
9	INPUT_SNMP	INPUT_SNMP (10)	2
10	OUTPUT_SNMP	OUTPUT_SNMP (14)	2
11	PROTOCOL	PROTOCOL (4)	1
12	postIpDiffServCodePoint	postIpDiffServCodePoint (98)	1
13	IP_TOS	ipClassofService (5)	1
14	DST_TOS	postIpClassOfService (55)	1
15	APPLICATION_ID	APPLICATION_ID (95)	9
16	INTERNET_APPLICATION_ID	INTERNET_APPLICATION_ID (66)	4
17	FLOW_FLAGS	FLOW_FLAGS (65)	2
18	FORWARDING_STATUS	FORWARDING_STATUS (89)	1
19	flowEndReason	flowEndReason (136)	1
20	FLOW_SAMPLER_ID	FLOW_SAMPLER_ID (48)	1
21	IP_SRC_ADDR	IP_SRC_ADDR (8)	4
22	IP_DST_ADDR	IP_DST_ADDR (12)	4

260 - IPV6

Description	No NAT IPv6 traffic
Data Field Count	22

Data fields

Field #	Field	Type	Length
1	BYTES	BYTES (1)	8
2	OUT_BYTES	OUT_BYTES (23)	8
3	PKTS	PKTS (2)	4
4	OUT_PKTS	OUT_PKTS (24)	4
5	FIRST_SWITCHED	FIRST_SWITCHED (22)	4
6	LAST_SWITCHED	LAST_SWITCHED (21)	4
7	L4_SRC_PORT	L4_SRC_PORT (7)	2
8	L4_DST_PORT	L4_DST_PORT (11)	2
9	INPUT_SNMP	INPUT_SNMP (10)	2
10	OUTPUT_SNMP	OUTPUT_SNMP (14)	2
11	PROTOCOL	PROTOCOL (4)	1
12	postIpDiffServCodePoint	postIpDiffServCodePoint (98)	1
13	IP_TOS	ipClassofService (5)	1
14	DST_TOS	postIpClassOfService (55)	1
15	APPLICATION_ID	APPLICATION_ID (95)	9
16	INTERNET_APPLICATION_ID	INTERNET_APPLICATION_ID (66)	4
17	FLOW_FLAGS	FLOW_FLAGS (65)	2
18	FORWARDING_STATUS	FORWARDING_STATUS (89)	1
19	flowEndReason	flowEndReason (136)	1
20	FLOW_SAMPLER_ID	FLOW_SAMPLER_ID (48)	1
21	IPV6_SRC_ADDR	IPV6_SRC_ADDR (27)	16
22	IPV6_DST_ADDR	IPV6_DST_ADDR (28)	16

261 - ICMP4

Description	No NAT ICMPv4 traffic
Data Field Count	20

Data fields

Field #	Field	Type	Length
1	BYTES	BYTES (1)	8
2	OUT_BYTES	OUT_BYTES (23)	8
3	PKTS	PKTS (2)	4
4	OUT_PKTS	OUT_PKTS (24)	4
5	FIRST_SWITCHED	FIRST_SWITCHED (22)	4
6	LAST_SWITCHED	LAST_SWITCHED (21)	4
7	INPUT_SNMP	INPUT_SNMP (10)	2
8	OUTPUT_SNMP	OUTPUT_SNMP (14)	2
9	ICMP_TYPE	ICMP_TYPE (32)	2
10	PROTOCOL	PROTOCOL (4)	1
11	SRC_TOS	SRC_TOS (5)	1
12	DST_TOS	DST_TOS (55)	1
13	APPLICATION_ID	APPLICATION_ID (95)	9
14	INTERNET_APPLICATION_ID	INTERNET_APPLICATION_ID (66)	4
15	FLOW_FLAGS	FLOW_FLAGS (65)	2
16	FORWARDING_STATUS	FORWARDING_STATUS (89)	1
17	flowEndReason	flowEndReason (136)	1
18	FLOW_SAMPLER_ID	FLOW_SAMPLER_ID (48)	1
19	IP_SRC_ADDR	IP_SRC_ADDR (8)	4
20	IP_DST_ADDR	IP_DST_ADDR(12)	4

262 - ICMP6

Description	No NAT ICMPv6 traffic
Data Field Count	20

Data fields

Field #	Field	Type	Length
1	BYTES	BYTES (1)	8
2	OUT_BYTES	OUT_BYTES (23)	8
3	PKTS	PKTS (2)	4
4	OUT_PKTS	OUT_PKTS (24)	4
5	FIRST_SWITCHED	FIRST_SWITCHED (22)	4
6	LAST_SWITCHED	LAST_SWITCHED (21)	4
7	INPUT_SNMP	INPUT_SNMP (10)	2
8	OUTPUT_SNMP	OUTPUT_SNMP (14)	2
9	ICMP_TYPE	ICMP_TYPE (32)	2
10	PROTOCOL	PROTOCOL (4)	1
11	SRC_TOS	SRC_TOS (5)	1
12	DST_TOS	DST_TOS (55)	1
13	APPLICATION_ID	APPLICATION_ID (95)	9
14	INTERNET_APPLICATION_ID	INTERNET_APPLICATION_ID (66)	4
15	FLOW_FLAGS	FLOW_FLAGS (65)	2
16	FORWARDING_STATUS	FORWARDING_STATUS (89)	1
17	flowEndReason	flowEndReason (136)	1
18	FLOW_SAMPLER_ID	FLOW_SAMPLER_ID (48)	1
19	IPV6_SRC_ADDR	IPV6_SRC_ADDR (27)	16
20	IPV6_DST_ADDR	IPV6_DST_ADDR (28)	16

263 - IPV4_NAT

Description	Source/Destination NAT IPv4 traffic
Data Field Count	26

Data fields

Field #	Field	Type	Length
1	BYTES	BYTES (1)	8
2	OUT_BYTES	OUT_BYTES (23)	8

Field #	Field	Type	Length
3	PKTS	PKTS (2)	4
4	OUT_PKTS	OUT_PKTS (24)	4
5	FIRST_SWITCHED	FIRST_SWITCHED (22)	4
6	LAST_SWITCHED	LAST_SWITCHED (21)	4
7	L4_SRC_PORT	L4_SRC_PORT (7)	2
8	L4_DST_PORT	L4_DST_PORT (11)	2
9	INPUT_SNMP	INPUT_SNMP (10)	2
10	OUTPUT_SNMP	OUTPUT_SNMP (14)	2
11	PROTOCOL	PROTOCOL (4)	1
12	postIpDiffServCodePoint	postIpDiffServCodePoint (98)	1
13	IP_TOS	ipClassofService (5)	1
14	DST_TOS	postIpClassOfService (55)	1
15	APPLICATION_ID	APPLICATION_ID (95)	9
16	INTERNET_APPLICATION_ID	INTERNET_APPLICATION_ID(66)	4
17	FLOW_FLAGS	FLOW_FLAGS (65)	2
18	FORWARDING_STATUS	FORWARDING_STATUS (89)	1
19	flowEndReason	flowEndReason (136)	1
20	FLOW_SAMPLER_ID	FLOW_SAMPLER_ID (48)	1
21	IP_SRC_ADDR	IP_SRC_ADDR (8)	4
22	IP_DST_ADDR	IP_DST_ADDR (12)	4
23	postNATSourceIPv4Address	postNATSourceIPv4Address (225)	4
24	postNATDestinationIPv4Address	postNATDestinationIPv4Address (226)	4
25	postNAPTSourceTransportPort	postNAPTSourceTransportPort (227)	2
26	postNAPTDestinationTransportPort	postNAPTDestinationTransportPort (228)	2

264 - IPV4_AF_NAT

Description	AF NAT IPv4 traffic (4->6)
Data Field Count	26

Data fields

Field #	Field	Type	Length
1	BYTES	BYTES (1)	8
2	OUT_BYTES	OUT_BYTES (23)	8
3	PKTS	PKTS (2)	4
4	OUT_PKTS	OUT_PKTS (24)	4
5	FIRST_SWITCHED	FIRST_SWITCHED (22)	4
6	LAST_SWITCHED	LAST_SWITCHED (21)	4
7	L4_SRC_PORT	L4_SRC_PORT (7)	2
8	L4_DST_PORT	L4_DST_PORT (11)	2
9	INPUT_SNMP	INPUT_SNMP (10)	2
10	OUTPUT_SNMP	OUTPUT_SNMP (14)	2
11	PROTOCOL	PROTOCOL (4)	1
12	postIpDiffServCodePoint	postIpDiffServCodePoint (98)	1
13	SRC_TOS	SRC_TOS (5)	1
14	DST_TOS	DST_TOS (55)	1
15	APPLICATION_ID	APPLICATION_ID (95)	9
16	INTERNET_APPLICATION_ID	INTERNET_APPLICATION_ID (66)	4
17	FLOW_FLAGS	FLOW_FLAGS (65)	2
18	FORWARDING_STATUS	FORWARDING_STATUS (89)	1
19	flowEndReason	flowEndReason (136)	1
20	FLOW_SAMPLER_ID	FLOW_SAMPLER_ID (48)	1
21	IPV6_SRC_ADDR	IPV6_SRC_ADDR (27)	16
22	IPV6_DST_ADDR	IPV6_DST_ADDR (28)	16
23	postNATSourceIPv6Address	postNATSourceIPv6Address (281)	16
24	postNATDestinationIPv6Address	postNATDestinationIPv6Address (282)	16
25	postNAPTSourceTransportPort	postNAPTSourceTransportPort (227)	2
26	postNAPTDestinationTransportPort	postNAPTDestinationTransportPort (228)	2

265 - IPV6_NAT

Description	Source/Destination NAT IPv6 traffic
Data Field Count	26

Data fields

Field #	Field	Type	Length
1	BYTES	BYTES (1)	8
2	OUT_BYTES	OUT_BYTES (23)	8
3	PKTS	PKTS (2)	4
4	OUT_PKTS	OUT_PKTS (24)	4
5	FIRST_SWITCHED	FIRST_SWITCHED (22)	4
6	LAST_SWITCHED	LAST_SWITCHED (21)	4
7	L4_SRC_PORT	L4_SRC_PORT (7)	2
8	L4_DST_PORT	L4_DST_PORT (11)	2
9	INPUT_SNMP	INPUT_SNMP (10)	2
10	OUTPUT_SNMP	OUTPUT_SNMP (14)	2
11	PROTOCOL	PROTOCOL (4)	1
12	postIpDiffServCodePoint	postIpDiffServCodePoint (98)	1
13	SRC_TOS	SRC_TOS (5)	1
14	DST_TOS	DST_TOS (55)	1
15	APPLICATION_ID	APPLICATION_ID (95)	9
16	INTERNET_APPLICATION_ID	INTERNET_APPLICATION_ID (66)	4
17	FLOW_FLAGS	FLOW_FLAGS (65)	2
18	FORWARDING_STATUS	FORWARDING_STATUS (89)	1
19	flowEndReason	flowEndReason (136)	1
20	FLOW_SAMPLER_ID	FLOW_SAMPLER_ID (48)	1
21	IP_SRC_ADDR	IP_SRC_ADDR (8)	4
22	IP_DST_ADDR	IP_DST_ADDR (12)	4
23	postNATSourceIPv6Address	postNATSourceIPv6Address (281)	16
24	postNATDestinationIPv6Address	postNATDestinationIPv6Address (282)	16

Field #	Field	Type	Length
25	postNAPTSourceTransportPort	postNAPTSourceTransportPort (227)	2
26	postNAPTDestinationTransportPort	postNAPTDestinationTransportPort (228)	2

266 - IPV6_AF_NAT

Description	AF NAT IPv6 traffic (6->4)
Data Field Count	26

Data fields

Field #	Field	Type	Length
1	BYTES	BYTES (1)	8
2	OUT_BYTES	OUT_BYTES (23)	8
3	PKTS	PKTS (2)	4
4	OUT_PKTS	OUT_PKTS (24)	4
5	FIRST_SWITCHED	FIRST_SWITCHED (22)	4
6	LAST_SWITCHED	LAST_SWITCHED (21)	4
7	L4_SRC_PORT	L4_SRC_PORT (7)	2
8	L4_DST_PORT	L4_DST_PORT (11)	2
9	INPUT_SNMP	INPUT_SNMP (10)	2
10	OUTPUT_SNMP	OUTPUT_SNMP (14)	2
11	PROTOCOL	PROTOCOL (4)	1
12	postIpDiffServCodePoint	postIpDiffServCodePoint (98)	1
13	SRC_TOS	SRC_TOS (5)	1
14	DST_TOS	DST_TOS (55)	1
15	APPLICATION_ID	APPLICATION_ID (95)	9
16	INTERNET_APPLICATION_ID	INTERNET_APPLICATION_ID (66)	4
17	FLOW_FLAGS	FLOW_FLAGS (65)	2
18	FORWARDING_STATUS	FORWARDING_STATUS (89)	1
19	flowEndReason	flowEndReason (136)	1
20	FLOW_SAMPLER_ID	FLOW_SAMPLER_ID (48)	1

Field #	Field	Type	Length
21	IPV6_SRC_ADDR	IPV6_SRC_ADDR (27)	16
22	IPV6_DST_ADDR	IPV6_DST_ADDR (28)	16
23	postNATSourceIPv4Address	postNATSourceIPv4Address (225)	4
24	postNATDestinationIPv4Address	postNATDestinationIPv4Address (226)	4
25	postNAPTSourceTransportPort	postNAPTSourceTransportPort (227)	2
26	postNAPTDestinationTransportPort	postNAPTDestinationTransportPort (228)	2

267 - ICMPV4_NAT

Description	Source/Destination NAT ICMPv4 traffic
Data Field Count	24

Data fields

Field #	Field	Type	Length
1	BYTES	BYTES (1)	8
2	OUT_BYTES	OUT_BYTES (23)	8
3	PKTS	PKTS (2)	4
4	OUT_PKTS	OUT_PKTS (24)	4
5	FIRST_SWITCHED	FIRST_SWITCHED (22)	4
6	LAST_SWITCHED	LAST_SWITCHED (21)	4
7	INPUT_SNMP	INPUT_SNMP (10)	2
8	OUTPUT_SNMP	OUTPUT_SNMP (14)	2
9	ICMP_TYPE	ICMP_TYPE (32)	2
10	PROTOCOL	PROTOCOL (4)	1
11	SRC_TOS	SRC_TOS (5)	1
12	DST_TOS	DST_TOS (55)	1
13	APPLICATION_ID	APPLICATION_ID (95)	9
14	INTERNET_APPLICATION_ID	INTERNET_APPLICATION_ID (66)	4
15	FLOW_FLAGS	FLOW_FLAGS (65)	2
16	FORWARDING_STATUS	FORWARDING_STATUS (89)	1

Field #	Field	Type	Length
17	flowEndReason	flowEndReason (136)	1
18	FLOW_SAMPLER_ID	FLOW_SAMPLER_ID (48)	1
19	IP_SRC_ADDR	IP_SRC_ADDR (8)	4
20	IP_DST_ADDR	IP_DST_ADDR (12)	4
21	postNATSourceIPv4Address	postNATSourceIPv4Address (225)	4
22	postNATDestinationIPv4Address	postNATDestinationIPv4Address (226)	4
23	postNAPTSourceTransportPort	postNAPTSourceTransportPort (227)	2
24	postNAPTDestinationTransportPort	postNAPTDestinationTransportPort (228)	2

268 - ICMPV4_AF_NAT

Description	AF NAT ICMPv4 traffic (4->6)
Data Field Count	24

Data fields

Field #	Field	Type	Length
1	BYTES	BYTES (1)	8
2	OUT_BYTES	OUT_BYTES (23)	8
3	PKTS	PKTS (2)	4
4	OUT_PKTS	OUT_PKTS (24)	4
5	FIRST_SWITCHED	FIRST_SWITCHED (22)	4
6	LAST_SWITCHED	LAST_SWITCHED (21)	4
7	INPUT_SNMP	INPUT_SNMP (10)	2
8	OUTPUT_SNMP	OUTPUT_SNMP (14)	2
9	ICMP_TYPE	ICMP_TYPE (32)	2
10	PROTOCOL	PROTOCOL (4)	1
11	SRC_TOS	SRC_TOS (5)	1
12	DST_TOS	DST_TOS (55)	1
13	APPLICATION_ID	APPLICATION_ID (95)	9
14	INTERNET_APPLICATION_ID	INTERNET_APPLICATION_ID (66)	4

Field #	Field	Type	Length
15	FLOW_FLAGS	FLOW_FLAGS (65)	2
16	FORWARDING_STATUS	FORWARDING_STATUS (89)	1
17	flowEndReason	flowEndReason (136)	1
18	FLOW_SAMPLER_ID	FLOW_SAMPLER_ID (48)	1
19	IPV6_SRC_ADDR	IPV6_SRC_ADDR (27)	16
20	IPV6_DST_ADDR	IPV6_DST_ADDR (28)	16
21	postNATSourceIPv6Address	postNATSourceIPv6Address (281)	16
22	postNATDestinationIPv6Address	postNATDestinationIPv6Address (282)	16
23	postNAPTSourceTransportPort	postNAPTSourceTransportPort (227)	2
24	postNAPTDestinationTransportPort	postNAPTDestinationTransportPort (228)	2

269 - ICMPV6_NAT

Description	Source/Destination NAT ICMPv6 traffic
Data Field Count	24

Data fields

Field #	Field	Type	Length
1	BYTES	BYTES (1)	8
2	OUT_BYTES	OUT_BYTES (23)	8
3	PKTS	PKTS (2)	4
4	OUT_PKTS	OUT_PKTS (24)	4
5	FIRST_SWITCHED	FIRST_SWITCHED (22)	4
6	LAST_SWITCHED	LAST_SWITCHED (21)	4
7	INPUT_SNMP	INPUT_SNMP (10)	2
8	OUTPUT_SNMP	OUTPUT_SNMP (14)	2
9	ICMP_TYPE	ICMP_TYPE (32)	2
10	PROTOCOL	PROTOCOL (4)	1
11	SRC_TOS	SRC_TOS (5)	1
12	DST_TOS	DST_TOS (55)	1

Field #	Field	Type	Length
13	APPLICATION_ID	APPLICATION_ID (95)	9
14	INTERNET_APPLICATION_ID	INTERNET_APPLICATION_ID (66)	4
15	FLOW_FLAGS	FLOW_FLAGS (65)	2
16	FORWARDING_STATUS	FORWARDING_STATUS (89)	1
17	flowEndReason	flowEndReason (136)	1
18	FLOW_SAMPLER_ID	FLOW_SAMPLER_ID (48)	1
19	IP_SRC_ADDR	IP_SRC_ADDR (8)	4
20	IP_DST_ADDR	IP_DST_ADDR (12)	4
21	postNATSourceIPv6Address	postNATSourceIPv6Address (281)	16
22	postNATDestinationIPv6Address	postNATDestinationIPv6Address (282)	16
23	postNAPTSourceTransportPort	postNAPTSourceTransportPort (227)	2
24	postNAPTDestinationTransportPort	postNAPTDestinationTransportPort (228)	2

270 - ICMPV6_AF_NAT

Description	AF NAT ICMPv6 traffic (6->4)
Data Field Count	24

Data fields

Field #	Field	Type	Length
1	BYTES	BYTES (1)	8
2	OUT_BYTES	OUT_BYTES (23)	8
3	PKTS	PKTS (2)	4
4	OUT_PKTS	OUT_PKTS (24)	4
5	FIRST_SWITCHED	FIRST_SWITCHED (22)	4
6	LAST_SWITCHED	LAST_SWITCHED (21)	4
7	INPUT_SNMP	INPUT_SNMP (10)	2
8	OUTPUT_SNMP	OUTPUT_SNMP (14)	2
9	ICMP_TYPE	ICMP_TYPE (32)	2
10	PROTOCOL	PROTOCOL (4)	1

Field #	Field	Type	Length
11	SRC_TOS	SRC_TOS (5)	1
12	DST_TOS	DST_TOS (55)	1
13	APPLICATION_ID	APPLICATION_ID (95)	9
14	INTERNET_APPLICATION_ID	INTERNET_APPLICATION_ID (66)	4
15	FLOW_FLAGS	FLOW_FLAGS (65)	2
16	FORWARDING_STATUS	FORWARDING_STATUS (89)	1
17	flowEndReason	flowEndReason (136)	1
18	FLOW_SAMPLER_ID	FLOW_SAMPLER_ID (48)	1
19	IPV6_SRC_ADDR	IPV6_SRC_ADDR (27)	16
20	IPV6_DST_ADDR	IPV6_DST_ADDR (28)	16
21	postNATSourceIPv4Address	postNATSourceIPv4Address (225)	4
22	postNATDestinationIPv4Address	postNATDestinationIPv4Address (226)	4
23	postNAPTSourceTransportPort	postNAPTSourceTransportPort (227)	2
24	postNAPTDestinationTransportPort	postNAPTDestinationTransportPort (228)	2

NetFlow on FortiExtender and tunnel interfaces

NetFlow sampling is supported on FortiExtender and VPN tunnel interfaces.

VPN tunnel interfaces can be IPsec, IP in IP, or GRE tunnels. NetFlow sampling is supported on both NPU and non-NPU offloaded tunnels.

Examples

In the following examples, a FortiExtender and a VPN tunnel interface are configured with NetFlow sampling.

To configure a FortiExtender interface with NetFlow sampling:

1. Configure a FortiExtender interface with NetFlow sampling enabled for both transmitted and received traffic:

```
config system interface
    edit "fext-211"
        set vdom "root"
        set mode dhcp
        set type fext-wan
```

```
        set netflow-sampler both
        set role wan
        set snmp-index 8
        set macaddr 2a:4e:68:a3:f4:6a
    next
end
```

2. Check the NetFlow status and configuration:
   Device index 26 is the FortiExtender interface `fext-211`.

```
# diagnose test application sflowd 3
===== Netflow Vdom Configuration =====
Global collector:172.18.60.80:[2055] source ip: 0.0.0.0 active-timeout(seconds):60 inactive-
timeout(seconds):600
____ vdom: root, index=0, is master, collector: disabled (use global config) (mgmt vdom)
   |_ coll_ip:172.18.60.80[2055],src_ip:10.6.30.105,seq_num:300,pkts/time to next template:
18/29
   |_ exported: Bytes:3026268, Packets:11192, Sessions:290 Flows:482
   |____ interface:fext-211 sample_direction:both device_index:26 snmp_index:8
```

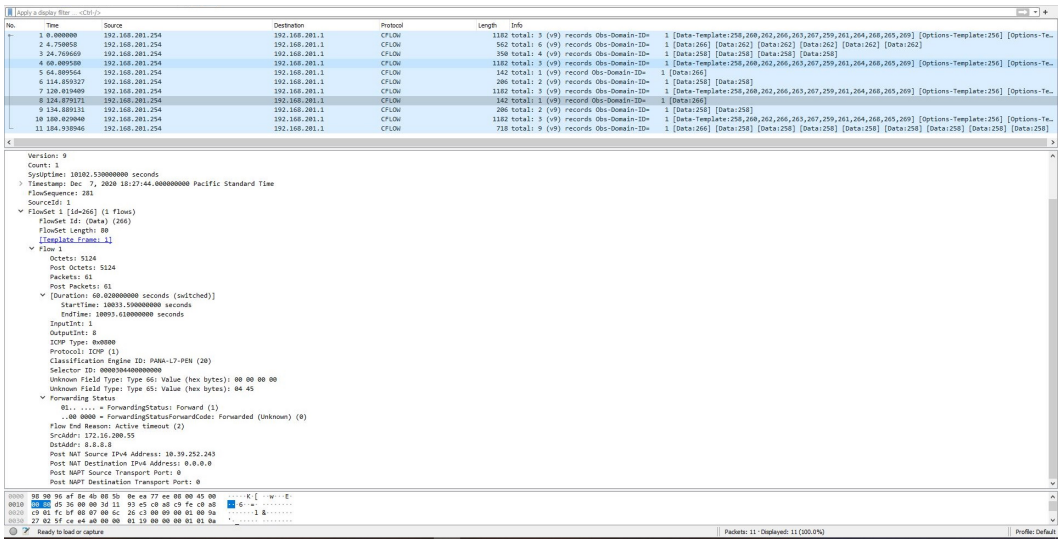
3. Check the network interface list:

```
# diagnose netlink interface list
...
if=fext-211 family=00 type=1 index=26 mtu=1500 link=0 master=0
ref=27 state=start present fw_flags=60000 flags=up broadcast run multicast
...
```

4. Check the session list for the FortiExtender interface and NetFlow flowset packet:

```
# diagnose sys session list
session info: proto=1 proto_state=00 duration=1732 expire=59 timeout=0 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty netflow-origin netflow-reply
statistic(bytes/packets/allow_err): org=145572/1733/1 reply=145572/1733/1 tuples=2
tx speed(Bps/kbps): 83/0 rx speed(Bps/kbps): 83/0
orgin->sink: org pre->post, reply pre->post dev=5->26/26->5 gwy=10.39.252.244/172.16.200.55
hook=post dir=org act=snat 172.16.200.55:61290->8.8.8.8:8(10.39.252.243:61290)
hook=pre dir=reply act=dnat 8.8.8.8:61290->10.39.252.243:0(172.16.200.55:61290)
misc=0 policy_id=1 auth_info=0 chk_client_info=0 vd=0
serial=00001298 tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id=00000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x040000
no_ofld_reason:  non-npu-intf
total session 1
```

5. The flowset packet can be captured on UDP port 2055 by a packet analyzer, such as Wireshark:



To configure a VPN tunnel interface with NetFlow sampling:

1. Configure a VPN interface with NetFlow sampling enabled for both transmitted and received traffic:

```
config system interface
    edit "A-to-B_vpn"
        set vdom "vdom1"
        set type tunnel
        set netflow-sampler both
        set snmp-index 42
        set interface "port3"
    next
end
```

2. Configure the VPN tunnel:

```
config vpn ipsec phase1-interface
    edit "A-to-B_vpn"
        set interface "port3"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set comments "VPN: A-to-B_vpn [Created by VPN wizard]"
        set wizard-type static-fortigate
        set remote-gw 10.2.2.2
        set psksecret ENC
    next
end
```

```
config vpn ipsec phase2-interface
    edit "A-to-B_vpn"
        set phase1name "A-to-B_vpn"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
```

```
        set comments "VPN: A-to-B_vpn [Created by VPN wizard]"
        set src-addr-type name
        set dst-addr-type name
        set src-name "A-to-B_vpn_local"
        set dst-name "A-to-B_vpn_remote"
    next
end
```

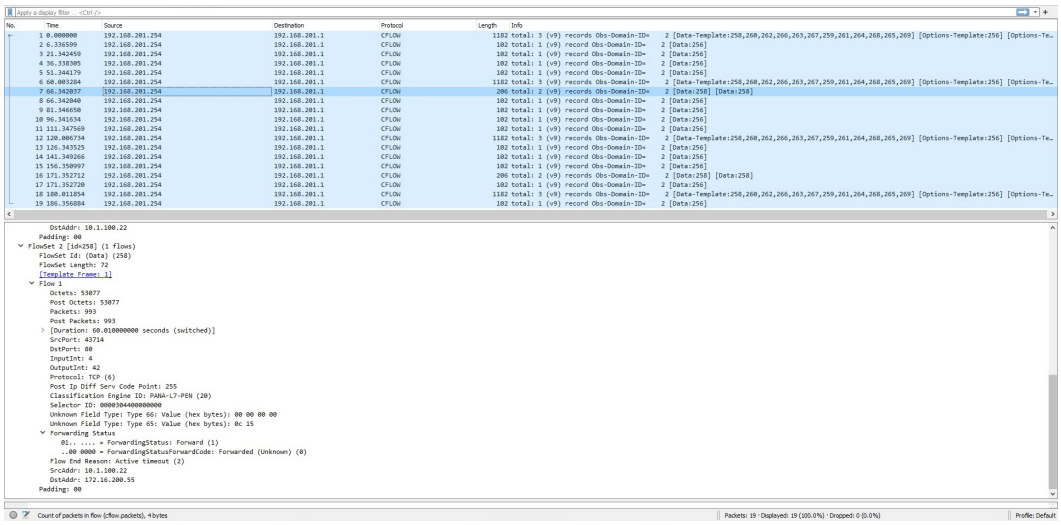
3. Check the NetFlow status and configuration:
   Device index 52 is the VPN interface `A-to-B_vpn`.

```
# diagnose test application sflowd 3
===== Netflow Vdom Configuration =====
Global collector:172.18.60.80:[2055] source ip: 0.0.0.0 active-timeout(seconds):60 inactive-
timeout(seconds):15
____ vdom: vdom1, index=1, is master, collector: disabled (use global config) (mgmt vdom)
   |_ coll_ip:172.18.60.80[2055],src_ip:10.1.100.1,seq_num:60,pkts/time to next template: 15/6
   |_ exported: Bytes:11795591, Packets:48160, Sessions:10 Flows:34
   |____ interface:A-to-B_vpn sample_direction:both device_index:52 snmp_index:42
```

4. Check the session list for the VPN interface and NetFlow flowset packet (unencapsulated traffic going through the VPN tunnel):

```
# diagnose sys session list
session info: proto=6 proto_state=01 duration=6 expire=3599 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty npu netflow-origin netflow-reply
statistic(bytes/packets/allow_err): org=6433/120/1 reply=884384/713/1 tuples=2
tx speed(Bps/kbps): 992/7 rx speed(Bps/kbps): 136479/1091
orgin->sink: org pre->post, reply pre->post dev=10->52/52->10 gwy=10.2.2.2/10.1.100.22
hook=pre dir=org act=noop 10.1.100.22:43714->172.16.200.55:80(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.200.55:80->10.1.100.22:43714(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
src_mac=00:0c:29:ac:ae:4f
misc=0 policy_id=5 auth_info=0 chk_client_info=0 vd=1
serial=00003b6c tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id=00000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x000001 no_offload
npu info: flag=0x82/0x00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0, vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
no_ofld_reason:  disabled-by-policy
total session 1
```

5. The flowset packet can be captured on UDP port 2055 by a packet analyzer, such as Wireshark:



Allow multiple NetFlow collectors

FortiOS can be configured with a maximum of six NetFlow collectors. This also applies to multi-VDOM environments where a maximum of six NetFlow collectors can be used globally or on a per-VDOMs basis. This feature enables up to a maximum of six unique parallel NetFlow streams or transmissions per NetFlow sample to six different NetFlow collectors. The NetFlow collector configuration can only be configured in the CLI.

The `netflow` command is global, and utilized by the management VDOM. The `vdom-netflow` command is only available for non-management VDOMs.

```
config system {netflow | vdom-netflow}
    config collectors
        edit <id>
            set collector-ip <IP address>
            set collector-port <port>
            set source-ip <IP address>
            set interface-select-method {auto | sdwan | specify}
            set interface <interface>
        next
    end
end
```

`collector-ip`	Enter the IPv4 or IPv6 address of the NetFlow collector that NetFlow agents added to interfaces in this VDOM send NetFlow datagrams to.
`collector-port`	Enter the UDP port number used for sending NetFlow datagrams; only configure if it is required by the NetFlow collector or network configuration (0 - 65535, default = 6343).
`source-ip`	Enter the source IPv4 or IPv6 address for the NetFlow agent.
`interface-select-method`	Specify how to select the outgoing interface to reach the server. • `auto`: Set the outgoing interface automatically.

	• sdwan: Set the outgoing interface by SD-WAN or policy routing rules. • specify: Set the outgoing interface manually.
interface <interface>	Enter the outgoing interface to reach the server.

If the interface-select-method is set to auto, the outgoing interface that is used to send the sampled NetFlow traffic to the NetFlow collector is decided by the routing table lookup.

Example 1: Multiple NetFlow collectors in a non-VDOM environment

In this example, six NetFlow collectors are configured in a non-VDOM environment with NetFlow sampling on the port1 interface.

To configure multiple NetFlow collectors:

1. Configure the NetFlow collectors:

```
config system netflow
   config collectors
   set active-flow-timeout 60
   set template-tx-timeout 60
        edit 1
            set collector-ip 172.16.200.155
            set collector-port 2055
            set source-ip 172.16.200.6
            set interface-select-method specify
            set interface "port1"
        next
        edit 2
            set collector-ip 10.1.100.59
            set collector-port 2056
            set source-ip 10.1.100.6
            set interface-select-method specify
            set interface "port2"
        next
        edit 3
            set collector-ip 172.18.60.80
            set collector-port 2057
            set interface-select-method specify
            set interface "port1"
        next
        edit 4
            set collector-ip "172.18.60.1"
            set collector-port 2058
        next
        edit 5
            set collector-ip "172.18.60.3"
```

```
            set collector-port 2059
        next
        edit 6
            set collector-ip "172.18.60.4"
            set collector-port 2060
        next
    end
end
```

2. Configure NetFlow sampling on port1:

```
config system interface
    edit port1
        set netflow-sampler both
    next
end
```

3. Verify the NetFlow diagnostics.
   a. Verify the NetFlow configuration status:

```
# diagnose test application sflowd 3

===== Netflow Vdom Configuration =====
Global collector(s) active-timeout(seconds):60 inactive-timeout(seconds):15
    Collector id:1: 172.16.200.155[2055] source IP:172.16.200.6
    Collector id:2: 10.1.100.59[2056] source IP:10.1.100.6
    Collector id:3: 172.18.60.80[2057] source IP:
    Collector id:4: 172.18.60.1[2058] source IP:
    Collector id:5: 172.18.60.3[2059] source IP:
    Collector id:6: 172.18.60.4[2060] source IP:
____ vdom: root, index=0, is master, collector: disabled (use global config) (mgmt vdom)
   |_ coll_ip:172.16.200.155:2056,src_ip:172.16.200.6
   |_ coll_ip:10.1.100.59:2057,src_ip:10.1.100.6
   |_ coll_ip:172.18.60.80:2058,src_ip:172.16.200.6
   |_ coll_ip:172.18.60.1:2058,src_ip:172.16.200.6
   |_ coll_ip:172.18.60.3:2059,src_ip:172.16.200.6
   |_ coll_ip:172.18.60.4:2060,src_ip:172.16.200.6
   |_ seq_num:13 pkts/time to next template: 16/29
   |_ exported: Bytes:2533746, Packets:3911, Sessions:70 Flows:70
   |_ active_intf: 1
   |____ interface:port1 sample_direction:both device_index:9 snmp_index:3
```

   b. Verify the sampled NetFlow traffic packet capture:

```
# diagnose sniffer packet any 'udp and port 2056 or 2057 or 2058' 4

filters=[udp and port 2056 or 2057 or 2058]
5.717060 port1 out 172.16.200.6.2472 -> 172.16.200.155.2055: udp 60
5.717068 port2 out 10.1.100.6.2472 -> 10.1.100.59.2056: udp 60
5.717075 port1 out 172.16.200.6.2472 -> 172.18.60.80.2057: udp 60
5.717078 port1 out 172.16.200.6.2472 -> 172.18.60.1.2058: udp 60
```

```
5.717081 port1 out 172.16.200.6.2472 -> 172.18.60.3.2059: udp 60
5.717085 port1 out 172.16.200.6.2472 -> 172.18.60.4.2060: udp 60
```

Example 2: Multiple NetFlow collectors in a multi-VDOM environment

In this example, six NetFlow collectors are configured in a multi-VDOM environment globally and per VDOM. NetFlow sampling is on the port1 and port4 interfaces.

Please note it is not mandatory to set up per-VDOM NetFlow collectors in a multi-vdom environment. However, if you don't enable per-VDOM collectors, the settings of the global NetFlow Collector will be used instead.

To configure multiple NetFlow collectors:

1. Configure the global NetFlow collectors:

```
config system netflow
   config collectors
   set active-flow-timeout 60
   set template-tx-timeout 60
        edit 1
            set collector-ip 172.16.200.155
            set collector-port 2055
            set source-ip 172.16.200.6
            set interface-select-method specify
            set interface "port1"
        next
        edit 2
            set collector-ip 10.1.100.59
            set collector-port 2056
            set source-ip 10.1.100.6
            set interface-select-method specify
            set interface "port2"
        next
        edit 3
            set collector-ip 172.18.60.80
            set collector-port 2057
            set interface-select-method specify
            set interface "port1"
        next
        edit 4
            set collector-ip "172.18.60.1"
            set collector-port 2058
        next
        edit 5
            set collector-ip "172.18.60.3"
            set collector-port 2059
```

```
        next
        edit 6
            set collector-ip "172.18.60.4"
            set collector-port 2060
        next
    end
end
```

2. Configure the per-VDOM NetFlow collectors:

```
config system vdom-netflow
    set vdom-netflow enable
    config collectors
        edit 1
            set collector-ip "172.10.100.101"
            set collector-port 2059
        next
        edit 2
            set collector-ip "172.10.100.102"
            set collector-port 2060
        next
        edit 3
            set collector-ip "172.10.100.103"
            set collector-port 2061
        next
        edit 4
            set collector-ip "172.10.100.104"
            set collector-port 2062
        next
        edit 5
            set collector-ip "172.10.100.105"
            set collector-port 2063
        next
        edit 6
            set collector-ip "172.10.100.106"
            set collector-port 2064
        next
    end
end
```

3. Configure NetFlow sampling on port1 and port4:

```
config system interface
    edit port1
        set netflow-sampler both
    next
    edit port4
        set netflow-sampler both
    next
end
```

In a multi-VDOM environment, ensure the interface selected for NetFlow sampling is in the same VDOM as the per-VDOM NetFlow collector. For global NetFlow collectors, the interface selected for NetFlow sampling should be in the management VDOM.

4. Verify the NetFlow diagnostics.
   a. Verify the NetFlow configuration status:

```
# diagnose test application sflowd 3

===== Netflow Vdom Configuration =====
Global collector(s) active-timeout(seconds):60 inactive-timeout(seconds):15
    Collector id:1: 172.16.200.155[2055] source IP:172.16.200.6
    Collector id:2: 10.1.100.59[2056] source IP:10.1.100.6
    Collector id:3: 172.18.60.80[2057] source IP:
    Collector id:4: 172.18.60.1[2058] source IP:
    Collector id:5: 172.18.60.3[2059] source IP:
    Collector id:6: 172.18.60.4[2060] source IP:
____ vdom: root, index=0, is master, collector: disabled (use global config) (mgmt vdom)
   |_ coll_ip:172.16.200.155:2056,src_ip:172.16.200.6
   |_ coll_ip:10.1.100.59:2057,src_ip:10.1.100.6
   |_ coll_ip:172.18.60.80:2058,src_ip:172.16.200.6
   |_ coll_ip:172.18.60.1:2058,src_ip:172.16.200.6
   |_ coll_ip:172.18.60.3:2059,src_ip:172.16.200.6
   |_ coll_ip:172.18.60.4:2060,src_ip:172.16.200.6
   |_ seq_num:13 pkts/time to next template: 16/29
   |_ exported: Bytes:2533746, Packets:3911, Sessions:70 Flows:70
   |_ active_intf: 1
   |____ interface:port1 sample_direction:both device_index:9 snmp_index:3
____ vdom: vdom1, index=1, is master, collector: enabled
   |_ coll_ip:172.10.100.101:2059,src_ip:20.1.100.111
   |_ coll_ip:172.10.100.102:2060,src_ip:20.1.100.111
   |_ coll_ip:172.10.100.103:2061,src_ip:20.1.100.111
   |_ coll_ip:172.10.100.104:2062,src_ip:20.1.100.111
   |_ coll_ip:172.10.100.105:2063,src_ip:20.1.100.111
   |_ coll_ip:172.10.100.106:2064,src_ip:20.1.100.111
   |_ seq_num:27 pkts/time to next template: 15/18
   |_ exported: Bytes:5040, Packets:60, Sessions:6 Flows:6
   |_ active_intf: 1
   |____ interface:port4 sample_direction:both device_index:12 snmp_index:6
```

   b. Verify the sampled NetFlow traffic packet capture:

```
# diagnose sniffer packet any 'udp and port 2059 or 2060 or 2061 or 2062 or 2063 or 2064'
4

filters=[udp and port 2059 or 2060 or 2061 or 2062 or 2063 or 2064]
7.005812 port4 out 20.1.100.111.2472 -> 172.10.100.101.2059: udp 60
7.005821 port4 out 20.1.100.111.2472 -> 172.10.100.102.2060: udp 60
7.005826 port4 out 20.1.100.111.2472 -> 172.10.100.103.2061: udp 60
7.005830 port4 out 20.1.100.111.2472 -> 172.10.100.104.2062: udp 60
```

```
7.005834 port4 out 20.1.100.111.2472 -> 172.10.100.105.2063: udp 60
7.005838 port4 out 20.1.100.111.2472 -> 172.10.100.106.2064: udp 60
```

Netflow sampling

Netflow sampling counts the number of packets or bytes an interface has sampled. If the packet count for a session surpasses the configured threshold for transmitted or received traffic on a NetFlow-enabled interface, a NetFlow report is exported. This helps reduce the load on the collector.

```
config system interface
    edit <name>
        set netflow-sampler {tx | rx | both}
        set netflow-sample-rate <integer>
        set netflow-sampler-id <integer>
    next
end
```

`netflow-sampler {tx | rx | both}`	Enable/disable NetFlow on this interface and set the data that NetFlow collects.
`netflow-sample-rate <integer>`	NetFlow sample rate. Sample one packet every configured number of packets (1 - 65535, default = 1, which means standard NetFlow where all packets are sampled).
`netflow-sampler-id <integer>`	NetFlow sampler ID.

All sessions that hit the interface with NetFlow sampling configured are still reported to the exporter daemon (sflowd), which keeps a tally of the sampled packets and bytes. If the session has more ingress, egress, or both, packets than the configured threshold (`netflow-sample-rate`), then a NetFlow report is exported. The Netflow report includes rounded-up numbers of packets and bytes divided by the sampling rate.

In this example, FortiGate is connect on port2 to a NetFlow collector, NetFlow sampling is configured on port2 with a sampling rate of 100. It is assumed that policies are already configured. Packets are sent that hit the policy. If the number of packets is less than the sampling, then no flowset is sent to the collector. If the number of packets is more than the sampling rate, then a flowset is sent to the collector.

To configure the interface and test sending different numbers of packets:

1. Connect the FortiGate port2 interface to the NetFlow collector.
2. Configure the interface:

```
config system interface
    edit "port2"
        set vdom "root"
        set ip 10.1.100.6 255.255.255.0
        set allowaccess https ssh snmp http telnet
        set type physical
        set netflow-sampler both
        set netflow-sample-rate 100
```

```
        set netflow-sampler-id 99
        set alias "To_vlan20"
        set snmp-index 2
        config ipv6
            set ip6-address 2000:10:1:100::6/64
            set ip6-allowaccess https ssh http telnet
        end
        set speed 1000auto
    next
end
```

3. Send 80 ICMP packets to pass through port2.

   Because this is less than the sample rate of 100, the FortiGate does not send any flowsets to the collector after the session timeout.

```
# diagnose sys session list

session info: proto=1 proto_state=00 duration=114 expire=26 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty f00 netflow-origin netflow-reply
statistic(bytes/packets/allow_err): org=6720/80/1 reply=6720/80/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=8->7/7->8 gwy=0.0.0.0/0.0.0.0
hook=post dir=org act=snat 10.1.100.41:20043->172.16.200.155:8(172.16.200.6:20043)
hook=pre dir=reply act=dnat 172.16.200.155:20043->172.16.200.6:0(10.1.100.41:20043)
misc=0 policy_id=44 pol_uuid_idx=8183 auth_info=0 chk_client_info=0 vd=0
serial=0001e4be tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
total session: 1
```

4. Send 105 ICMP packets to pass through port2.

   Because this is more than the sample rate of 100, the FortiGate sends one flowset packet to the collector after the session timeout.

```
# diagnose sys session list

session info: proto=1 proto_state=00 duration=110 expire=55 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty f00 netflow-origin netflow-reply
statistic(bytes/packets/allow_err): org=8820/105/1 reply=8820/105/1 tuples=2
tx speed(Bps/kbps): 78/0 rx speed(Bps/kbps): 78/0
orgin->sink: org pre->post, reply pre->post dev=8->7/7->8 gwy=0.0.0.0/0.0.0.0
```

```
hook=post dir=org act=snat 10.1.100.41:20072->172.16.200.155:8(172.16.200.6:20072)
hook=pre dir=reply act=dnat 172.16.200.155:20072->172.16.200.6:0(10.1.100.41:20072)
misc=0 policy_id=44 pol_uuid_idx=8183 auth_info=0 chk_client_info=0 vd=0
serial=0001e56f tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
total session: 1
```

5. Check the flowset:

```
Octets: 88
Post Octets: 88
Packets: 1
Post Packets: 1
......
SamplerID: 99
Srcaddr: 10.1.100.41
Dstaddr: 172.16.200.155
......
```

Because the sample rate is 100, the flowset means:

- 88 * 100 = 8800 Bytes
- 1 * 100 = 100 Packets

This is basically equal to the actual value of 8820 bytes / 105 packets. This shows that after configuring the NetFlow sampling rate, the received flowsets have been reduced, and the flowsets' values are consistent with the actual values, although not exactly the same.

Filter NetFlow sampling

Exclusion filters can be applied to NetFlow sampling based on criteria including source and destination IP addresses, source and destinations ports, and IP protocol. This enhances the relevance of collected data, streamlines data management processes, and reduces excess network traffic. Exclusion filters are defined globally, and up to 64 can be configured.

```
config system netflow
    config exclusion-filters
        edit <id>
            set source-ip <IP_address>
            set destination-ip <IP_address>
            set source-port <port>
            set destination-port <port>
            set protocol <protocol_ID>
        next
    end
end
```

`source-ip <IP_address>`	Session source address.

`destination-ip <IP_address>`	Session destination address.
`source-port <port>`	Session source port number or range.
`destination-port <port>`	Session destination port number or range.
`protocol <protocol_ID>`	Session IP protocol (0 - 255, default = 255, meaning any).

In this example, IPv4-IPv4 and IPv6-IPv4 exclusion filters are configured on a FortiGate that is connected to a NetFlow connector. Packets are sent that hit the filters, then the session lists are checked for the NetFlow flag and the sessions are checked on the collector.

To configure and test the NetFlow exclusion filters:

1. Create NetFlow exclusion filters:

```
config system netflow
    set active-flow-timeout 60
    set template-tx-timeout 60
    config exclusion-filters
        edit 44
            set source-ip 10.1.100.41
            set destination-ip 172.16.200.155
        next
        edit 66
            set source-ip 2000:10:1:100::41
            set destination-ip 2000:172:16:200::155
        next
        edit 64
            set source-ip 2000:10:1:100::41
            set destination-ip 65:ff9b::ac10:c837
        next
        edit 46
            set source-ip 10.1.100.41
            set destination-ip 10.1.100.101
        next
    end
    config collectors
        edit 1
            set collector-ip "10.1.100.59"
        next
    end
end
```

2. Send packets to generate sessions.
3. Check if the NetFlow flag is attached in the session/session6 list:
   - If there are two IPv4-IPv4 sessions, and only the first one matches filter ID 44:

```
# diagnose sys session list
session info: proto=1 proto_state=00 duration=17 expire=59 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
```

```
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty f00 netflow-origin netflow-reply
statistic(bytes/packets/allow_err): org=1512/18/1 reply=1512/18/1 tuples=2
tx speed(Bps/kbps): 84/0 rx speed(Bps/kbps): 84/0
orgin->sink: org pre->post, reply pre->post dev=8->7/7->8 gwy=0.0.0.0/0.0.0.0
hook=post dir=org act=snat 10.1.100.41:11165->172.16.200.155:8(172.16.200.6:11165)
hook=pre dir=reply act=dnat 172.16.200.155:11165->172.16.200.6:0(10.1.100.41:11165)
misc=0 policy_id=2 pol_uuid_idx=8173 auth_info=0 chk_client_info=0 vd=0
serial=000032e9 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy

session info: proto=1 proto_state=00 duration=16 expire=59 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty f00 netflow-origin netflow-reply
statistic(bytes/packets/allow_err): org=1428/17/1 reply=1428/17/1 tuples=2
tx speed(Bps/kbps): 86/0 rx speed(Bps/kbps): 86/0
orgin->sink: org pre->post, reply pre->post dev=8->7/7->8 gwy=0.0.0.0/0.0.0.0
hook=post dir=org act=snat 10.1.100.41:11166->172.16.200.55:8(172.16.200.6:11166)
hook=pre dir=reply act=dnat 172.16.200.55:11166->172.16.200.6:0(10.1.100.41:11166)
misc=0 policy_id=2 pol_uuid_idx=8173 auth_info=0 chk_client_info=0 vd=0
serial=000032ea tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
total session: 2
```

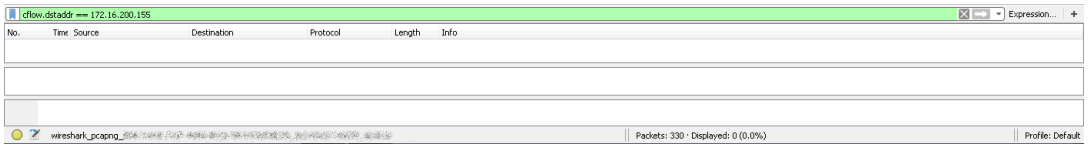
- If there are two IPv6-IPv4 (NAT64) sessions, and only the first one matches filter ID 64:

```
# diagnose sys session6 list
session6 info: proto=58 proto_state=00 duration=23 expire=59 timeout=0 refresh_dir=both
flags=00000000 sockport=0 socktype=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=log may_dirty netflow-origin netflow-reply
statistic(bytes/packets/allow_err): org=2392/23/0 reply=2392/23/0 tuples=2
tx speed(Bps/kbps): 102/0 rx speed(Bps/kbps): 102/0
orgin->sink: org pre->post, reply pre->post dev=8->43/43->8
hook=pre dir=org act=dnat 2000:10:1:100::41:11138->65:ff9b::ac10:c837:128
(65:ff9b::ac10:c837:11138)
hook=post dir=reply act=snat 65:ff9b::ac10:c837:11138->2000:10:1:100::41:129
(65:ff9b::ac10:c837:11138)
peer=172.16.201.8:1066->172.16.200.55:8 naf=1
```

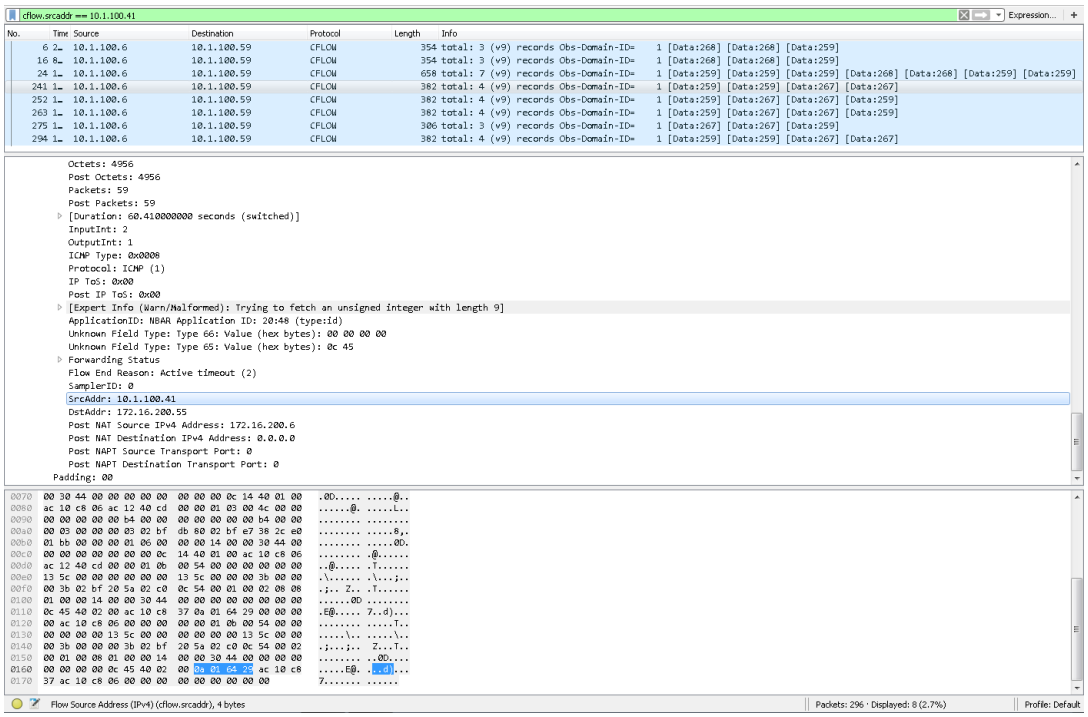
```
hook=pre dir=org act=noop 172.16.201.8:1066->172.16.200.55:8(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.200.55:1066->172.16.201.8:0(0.0.0.0:0)
misc=0 policy_id=4 pol_uuid_idx=8176 auth_info=0 chk_client_info=0 vd=0
serial=0000067e tos=ff/ff ips_view=9572 app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x040001 no_offload
no_ofld_reason:  disabled-by-policy non-npu-intf

session6 info: proto=58 proto_state=00 duration=25 expire=59 timeout=0 refresh_dir=both
flags=00000000 sockport=0 socktype=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=log may_dirty netflow-origin netflow-reply
statistic(bytes/packets/allow_err): org=2600/25/0 reply=2600/25/0 tuples=2
tx speed(Bps/kbps): 103/0 rx speed(Bps/kbps): 103/0
orgin->sink: org pre->post, reply pre->post dev=8->43/43->8
hook=pre dir=org act=dnat 2000:10:1:100::41:11137->65:ff9b::ac10:c89b:128
(65:ff9b::ac10:c89b:11137)
hook=post dir=reply act=snat 65:ff9b::ac10:c89b:11137->2000:10:1:100::41:129
(65:ff9b::ac10:c89b:11137)
peer=172.16.201.8:1065->172.16.200.155:8 naf=1
hook=pre dir=org act=noop 172.16.201.8:1065->172.16.200.155:8(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.200.155:1065->172.16.201.8:0(0.0.0.0:0)
misc=0 policy_id=4 pol_uuid_idx=8176 auth_info=0 chk_client_info=0 vd=0
serial=0000067d tos=ff/ff ips_view=9572 app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x040001 no_offload
no_ofld_reason:  disabled-by-policy non-npu-intf
total session6: 2
```

4. Check on the collector server. The FlowSets are received on collector for the session/session6 if they do not match the filters. If the sessions match the filter in the system NetFlow, then no FlowSets are received on the collector.
   - IPv4-IPv4 FlowSets:

     Do not receive FlowSets that match exclusion filter ID 44:

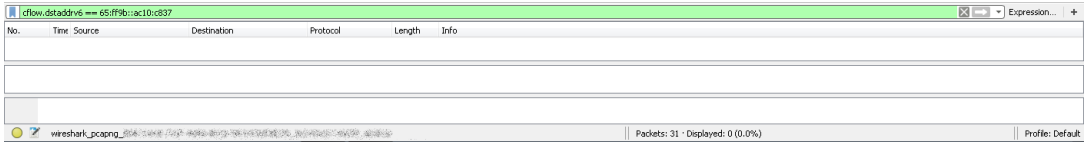
     

     Receive FlowSets that do not match any exclusion filters:

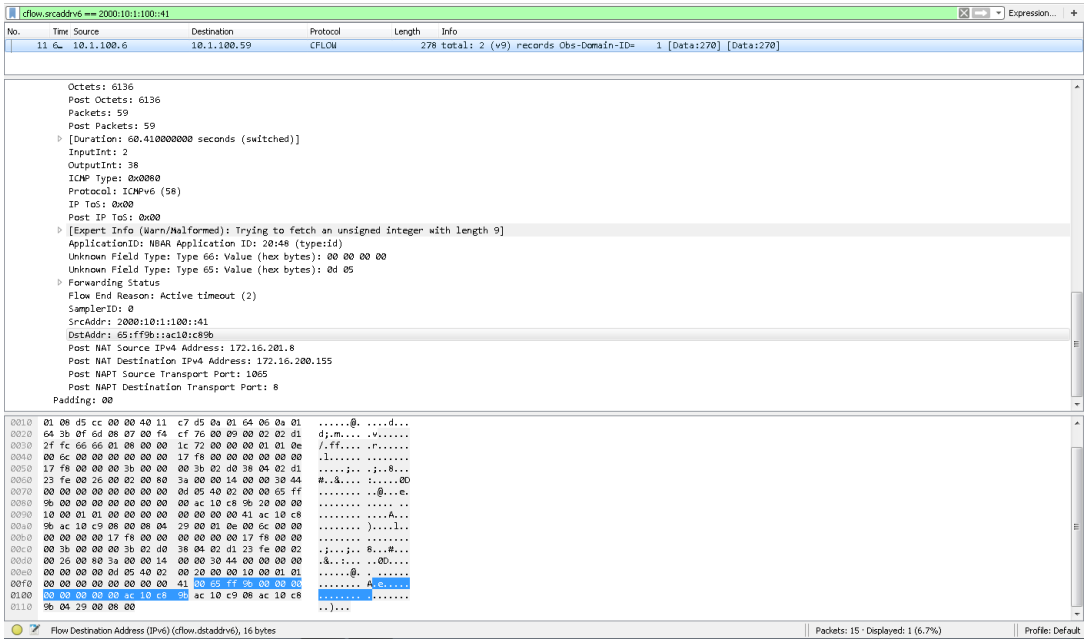


- IPv6-IPv4 FlowSets:

  Do not receive FlowSets that match exclusion filter ID 64:



  Receive FlowSets that do not match any exclusion filters:



sFlow

sFlow is a method of monitoring the traffic on your network to identify areas on the network that may impact performance and throughput. FortiGate supports sFlow v5. sFlow collector software is available from a number of third-party software vendors. For more information about sFlow, see www.sflow.org.

The packet information that the FortiGate's sFlow agent collects depends on the interface type:

- On an internal interface, when the interface receives packets from devices with private IP addresses, the collected information includes the private IP addresses.
- On an external, or WAN, interface, when the interface receives to route to or from the internet, the collected information includes the IP address of the WAN interface as the source or destination interface, depending on the direction of the traffic. It does not include IP addresses that are NATed on another interface.

sFlow datagrams contain the following information:

- Packet headers, such as MAC, IPv4, and TCP
- Sample process parameters, such as rate and pool
- Input and output ports
- Priority (802.1p and ToS)
- VLAN (802.1Q)
- Source prefixes, destination prefixes, and next hop addresses
- BGP source AS, source peer AS, destination peer AS, communities, and local preference
- User IDs (TACACS, RADIUS) for source and destination
- Interface statistics (RFC 1573, RFC 2233, and RFC 2358)

Configuring sFlow

sFlow can be configured globally, then on traffic VDOMs and individual interfaces. When configuring sFlow on a VDOM, the collector can be specified, or the collector that is configured globally can be used.

FortiOS can be configured with a maximum of three sFlow collectors. This also applies to multi-VDOM environments where a maximum of three sFlow collectors can be used globally and/or on a per-VDOMs basis. This enables up to a maximum of three unique parallel sFlow streams or transmissions per sFlow sample to three different sFlow collectors.

sFlow is supported on some interface types, such as physical, VLAN, and aggregate. It is not supported on virtual interfaces, such as VDOM link, IPsec, GRE, or SSL. When configuring sFlow on an interface, the rate that the agent samples traffic, the direction of that traffic, and the frequency that the agent sends sFlow datagrams to the sFlow collector can be specified. If sFlow is configured on the VDOM that the interface belongs to, the agent sends datagrams to the collector configured for the VDOM. Otherwise, the datagrams are sent to the collector that is configured globally.

Configuring sFlow for an interface disables NP offloading for all traffic on that interface.

To configure sFlow globally:

```
config system sflow
    config collectors
        edit <id>
            set collector-ip <ipv4_address>
            set collector-port <port>
            set source-ip <ipv4_address>
            set interface-select-method {auto | sdwan | specify}
            set interface <interface>
        next
    end
end
```

Command	Description
`collector-ip <ipv4_address>`	Enter the IPv4 address of the sFlow collector that sFlow agents added to interfaces in this VDOM send sFlow datagrams to (default = 0.0.0.0).
`collector-port <port>`	Enter the UDP port number used for sending sFlow datagrams; only configure if required by the sFlow collector or network configuration (0 - 65535, default = 6343).
`source-ip <ipv4_address>`	Enter the source IP address for the sFlow agent.
`interface-select-method {auto | sdwan | specify}`	Specify how to select the outgoing interface to reach the server (default = auto).
`interface <interface>`	Enter the outgoing interface to reach the server.

To configure sFlow for a VDOM:

```
config vdom
    edit <vdom>
        config system vdom-sflow
            set vdom-sflow {enable | disable}
            set collector-ip <ipv4_address>
            set collector-port <port>
            set source-ip <ipv4_address>
            set interface-select-method {auto | sdwan | specify}
            set interface <interface>
        end
    next
end
```

Command	Description
`vdom-sflow {enable | disable}`	Enable/disable the sFlow configuration for the current VDOM (default = disable).
`collector-ip <ipv4_address>`	Enter the IPv4 address of the sFlow collector that sFlow agents added to interface (default = 0.0.0.0).

Command	Description
	If this option is not configured, the global setting will be used.
`collector-port <port>`	Enter the UDP port number used for sending sFlow datagrams (0 - 65535, default = 6343). Only configured this option if required by the sFlow collector or your network configuration. If this option is not configured, the global setting will be used.
`source-ip <ipv4_address>`	Enter the source IPv4 address that the sFlow agent used to send datagrams to the collector (default = 0.0.0.0). If this option is not configured, the FortiGate uses the IP address of the interface that it sends the datagram through.
`interface-select-method {auto | sdwan | specify}`	Specify how the outgoing interface to reach the server is selected (default = auto).
`interface <interface>`	Enter the outgoing interface used to reach the server. This option is only available when `interface-select-method` is `specify`.

To configure sFlow on an interface:

```
config system interface
    edit <interface>
        set sflow-sampler {enable | disable}
        set sample-rate <integer>
        set polling-interval <integer>
        set sample-direction {tx | rx | both}
    next
end
```

Command	Description
`sflow-sampler {enable | disable}`	Enable/disable sFlow on this interface (default = disable).
`sample-rate <integer>`	Enter the average number of packets that the agent lets pass before taking a sample (10 - 99999, default = 2000). Setting a lower rate will sample a higher number of packets, increasing the accuracy or the sampling data, but also increasing the CPU and network bandwidth usage. The default value is recommended.
`polling-interval <integer>`	Enter the amount of time that the agent waits between sending datagrams to the collector, in seconds (1 - 255, default = 20). Setting a higher value lowers the amount of data that the agent sends across the network, but makes the collector's view of the network less current.
`sample-direction {tx | rx | both}`	Select the direction of the traffic that the agent collects (default = both).

Example 1: multiple sFlow collectors in a non-VDOM environment

In this example, three sFlow collectors are configured in a non-VDOM environment with sFlow sampling on the wan1 interface.

To configure multiple sFlow collectors:

1. Configure the sFlow collectors:

```
config system sflow
    config collectors
        edit 1
            set collector-ip 10.1.1.1
            set collector-port 6344
            set source-ip 0.0.0.0
            set interface-select-method auto
        next
        edit 2
            set collector-ip 10.1.1.2
            set collector-port 6345
            set source-ip 0.0.0.0
            set interface-select-method auto
        next
        edit 3
            set collector-ip 10.1.1.3
            set collector-port 6346
            set source-ip 0.0.0.0
            set interface-select-method auto
        next
    end
end
```

2. Configure sFlow sampling on wan1:

```
config system interface
    edit wan1
        set sflow-sampler enable
        set sample-rate 2000
        set polling-interval 20
        set sample-direction both
    next
end
```

3. Verify the sFlow diagnostics.
   a. Verify the sFlow configuration status:

```
# diagnose test application sflowd 1

global collector:10.1.1.1:[6344]
 global source ip: 0.0.0.0:[1399]
```

```
global collector:10.1.1.2:[6345]
 global source ip: 0.0.0.0:[1399]

global collector:10.1.1.3:[6346]
 global source ip: 0.0.0.0:[1399]
vdom: root, index=0, vdom sflow collector is disabled(use global sflow config), primary
(management vdom)
  intf:wan1 sample_rate:2000 polling_interval:20 sample_direction:both
```

b. Verify the sampled sFlow traffic packet capture:

```
# diagnose sniffer packet any 'port 1399' 4 0 l
interfaces=[any]
filters=[port 6344 or port 6345 or port 6346]
2023-11-14 15:44:41.658799 wan1 out 172.16.151.157.1399 -> 10.1.1.1.6344: udp 144
2023-11-14 15:44:41.658829 wan1 out 172.16.151.157.1399 -> 10.1.1.2.6345: udp 144
2023-11-14 15:44:41.658848 wan1 out 172.16.151.157.1399 -> 10.1.1.3.6346: udp 144
```

The outgoing interface that is used to send the sampled sFlow traffic to the sFlow collector is decided by the routing table lookup.

Example 2: multiple sFlow collectors in a multi-VDOM environment

In this example, three sFlow collectors are configured in a multi-VDOM environment globally and per VDOM. sFlow sampling is on the wan1 and dmz interfaces.

To configure multiple sFlow collectors:

1. Configure the global sFlow collectors:

```
config system sflow
    config collectors
        edit 1
            set collector-ip 10.1.1.1
            set collector-port 6344
            set source-ip 0.0.0.0
            set interface-select-method auto
        next
        edit 2
            set collector-ip 10.1.1.2
            set collector-port 6345
            set source-ip 0.0.0.0
            set interface-select-method auto
        next
        edit 3
            set collector-ip 10.1.1.3
            set collector-port 6346
            set source-ip 0.0.0.0
            set interface-select-method auto
```

```
        next
    end
end
```

2. Configure the per-VDOM sFlow collectors:

```
config vdom
    edit testvdom
        config system vdom-sflow
            set vdom-sflow enable
            config collectors
                edit 1
                    set collector-ip 10.1.1.4
                    set collector-port 6347
                    set source-ip 0.0.0.0
                    set interface-select-method auto
                next
                edit 2
                    set collector-ip 10.1.1.5
                    set collector-port 6348
                    set source-ip 0.0.0.0
                    set interface-select-method auto
                next
                edit 3
                    set collector-ip 10.1.1.6
                    set collector-port 6349
                    set source-ip 0.0.0.0
                    set interface-select-method auto
                next
            end
        end
    next
end
```

3. Configure sFlow sampling on wan1 and dmz:

```
config system interface
    edit wan1
        set vdom "root"
        set sflow-sampler enable
        set sample-rate 2000
        set polling-interval 20
        set sample-direction both
    next
    edit dmz
        set vdom "testvdom"
        set sflow-sampler enable
        set sample-rate 2000
        set polling-interval 20
        set sample-direction both
    next
end
```

4. Verify the sFlow diagnostics.
   a. Verify the sFlow configuration status:

```
# diagnose test application sflowd  1

global collector:10.1.1.1:[6344]
 global source ip: 0.0.0.0:[1399]

global collector:10.1.1.2:[6345]
 global source ip: 0.0.0.0:[1399]

global collector:10.1.1.3:[6346]
 global source ip: 0.0.0.0:[1399]
vdom: root, index=0, vdom sflow collector is disabled(use global sflow config), primary
(management vdom)
  intf:wan1 sample_rate:2000 polling_interval:20 sample_direction:both
vdom: testvdom, index=1, vdom sflow collector is enabled, primary
  collector:10.1.1.4:[6347] src:192.168.1.1:[1399]
  collector:10.1.1.5:[6348] src:192.168.1.1:[1399]
  collector:10.1.1.6:[6349] src:192.168.1.1:[1399]
  intf:dmz sample_rate:2000 polling_interval:20 sample_direction:both
```

   b. Verify the sampled sFlow traffic packet capture:

```
# sudo root diagnose sniffer packet any 'port 1399' 4 0 l
interfaces=[any]
filters=[port 1399]
2023-11-14 16:50:11.118807 wan1 out 172.16.151.157.1399 -> 10.1.1.1.6344: udp 144
2023-11-14 16:50:11.118838 wan1 out 172.16.151.157.1399 -> 10.1.1.2.6345: udp 144
2023-11-14 16:50:11.118865 wan1 out 172.16.151.157.1399 -> 10.1.1.3.6346: udp 144
2023-11-14 16:50:20.198784 dmz out 192.168.1.1.1399 -> 10.1.1.4.6347: udp 144
2023-11-14 16:50:20.198813 dmz out 192.168.1.1.1399 -> 10.1.1.5.6348: udp 144
2023-11-14 16:50:20.198832 dmz out 192.168.1.1.1399 -> 10.1.1.6.6349: udp 144
```

The outgoing interface that is used to send the sampled sFlow traffic to the sFlow collector is decided by the routing table lookup.

Link monitor

The link monitor is a mechanism that allows the FortiGate to probe the status of a detect server in order to determine the health of the link, next hop, or the path to the server. Ping, TCP echo, UDP echo, HTTP, and TWAMP protocols can be used for the probes. Typically, the detect server is set to a stable server several hops away. Multiple servers can also be configured with options to define the protocol and weights for each server.

The link monitor serves several purposes. In the most basic configuration, it can be used to detect failures and remove routes associated with the interface and gateway to prevent traffic from routing out the failed link. More granularity is added in 7.0 that allows only the routes specified in the link monitor to be removed from the

routing table. With this benefit, only traffic to specific routing destinations are removed, rather than all routing destinations.

Another enhancement starting in 7.0.1 is an option to toggle between enabling or disabling policy route updates when a link health monitor fails.

The link monitor can also monitor remote servers for HA failover. Using the HA built-in link monitor, it is only able to detect physical link failovers to trigger HA link failover. With the link monitor, remote servers can be used to monitor the health of the path to the server in order to trigger HA failover.

Finally, the link monitor can cascade the failure to other interfaces. When the `update-cascade-interface` option is enabled, the interface can be configured in conjunction with `fail-detect` enabled to trigger a link down event on other interfaces.

The following topics provide more information about the link monitor:

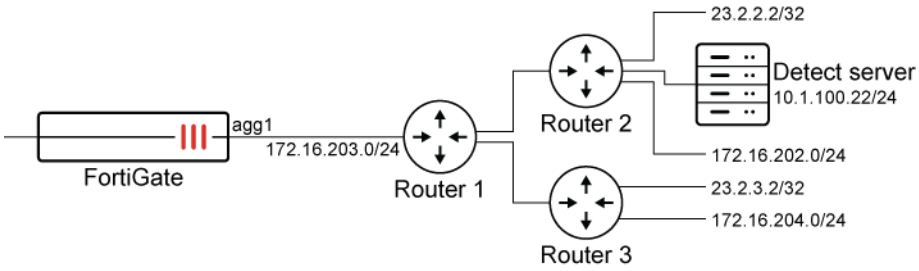
- Link monitor with route updates on page 874
- Enable or disable updating policy routes when link health monitor fails on page 875
- Add weight setting on each link health monitor server on page 878
- SLA link monitoring for dynamic IPsec and SSL VPN tunnels on page 881
- Dual internet connections on page 599
- HA remote link monitoring on page 884

Link monitor with route updates

When a link monitor fails, only the routes that are specified in the link monitor are removed from the routing table, instead of all the routes with the same interface and gateway. If no routes are specified, then all of the routes are removed. Only IPv4 routes are supported.

Example

In this example, the FortiGate has several routes to 23.2.2.2/32 and 172.16.202.2/24, and is monitoring the link *agg1* by pinging the server at 10.1.100.22. The link monitor uses the gateway 172.16.203.2.



When the link monitor fails, only the routes to the specified subnet using interface *agg1* and gateway 172.16.203.2 are removed.

To configure the link monitor:

```
config system link-monitor
    edit "22"
        set srcintf "agg1"
        set server "10.1.100.22"
        set gateway-ip 172.16.203.2
        set route "23.2.2.2/32" "172.16.202.0/24"
    next
end
```

To check the results:

1. When the link monitor is alive:

```
# get router info routing-table static
Routing table for VRF=0
S*      0.0.0.0/0 [5/0] via 10.100.1.249, port12
S       10.1.100.0/24 [10/0] via 172.16.203.2, agg1
S       23.2.2.2/32 [10/0] via 172.16.203.2, agg1
S       23.2.3.2/32 [10/0] via 172.16.203.2, agg1
S       172.16.201.0/24 [10/0] via 172.16.200.4, port9
S       172.16.202.0/24 [10/0] via 172.16.203.2, agg1
S       172.16.204.0/24 [10/0] via 172.16.200.4, port9
                        [10/0] via 172.16.203.2, agg1
                        [10/0] via 172.16.206.2, vlan100, [100/0]
```

2. When the link monitor is dead:

```
# get router info routing-table static
Routing table for VRF=0
S*      0.0.0.0/0 [5/0] via 10.100.1.249, port12
S       10.1.100.0/24 [10/0] via 172.16.203.2, agg1
S       23.2.3.2/32 [10/0] via 172.16.203.2, agg1
S       172.16.201.0/24 [10/0] via 172.16.200.4, port9
S       172.16.204.0/24 [10/0] via 172.16.200.4, port9
                        [10/0] via 172.16.203.2, agg1
                        [10/0] via 172.16.206.2, vlan100, [100/0]
```

Enable or disable updating policy routes when link health monitor fails

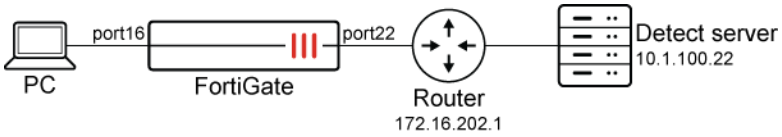
An option has been added to toggle between enabling or disabling policy route updates when a link health monitor fails. By disabling policy route updates, a link health monitor failure will not cause corresponding policy-based routes to be removed.

```
config system link-monitor
    edit <name>
        set update-policy-route {enable | disable}
```

```
    next
end
```

Example

In the following topology, the FortiGate is monitoring the detect server, 10.1.100.22. The FortiGate has a policy-based route to destination 172.16.205.10 using the same gateway (172.16.202.1) and interface (port22). By configuring `update-policy-route disable`, the policy-based route is not removed when the link health monitor detects a failure.



To disable updating policy routes when the link health monitor fails:

1. Configure the link health monitor:

```
config system link-monitor
    edit "test-1"
        set srcintf "port22"
        set server "10.1.100.22"
        set gateway-ip 172.16.202.1
        set failtime 3
        set update-policy-route disable
    next
end
```

2. Configure the policy route:

```
config router policy
    edit 1
        set input-device "port16"
        set dst "172.16.205.10/255.255.255.255"
        set gateway 172.16.202.1
        set output-device "port22"
        set tos 0x14
        set tos-mask 0xff
    next
end
```

3. When the health link monitor status is up, verify that the policy route is active.
   a. Verify the link health monitor status:

```
# diagnose sys link-monitor status
Link Monitor: test-1, Status: alive, Server num(1), HA state: local(alive), shared(alive)
Flags=0x1 init, Create time: Fri May 28 15:20:15 2021
Source interface: port22 (14)
Gateway: 172.16.202.1
```

```
Interval: 500 ms
Service-detect: disable
Diffservcode: 000000
Class-ID: 0
  Peer: 10.1.100.22(10.1.100.22)
        Source IP(172.16.202.2)
        Route: 172.16.202.2->10.1.100.22/32, gwy(172.16.202.1)
        protocol: ping, state: alive
                Latency(Min/Max/Avg): 0.374/0.625/0.510 ms
                Jitter(Min/Max/Avg): 0.008/0.182/0.074
                Packet lost: 0.000%
                Number of out-of-sequence packets: 0
                Fail Times(0/3)
                Packet sent: 7209, received: 3400, Sequence(sent/rcvd/exp): 7210/7210/7211
```

b. Verify the policy route list:

```
# diagnose firewall proute  list
list route policy info(vf=root):
id=1 dscp_tag=0xff 0xff flags=0x0 tos=0x14 tos_mask=0xff protocol=0 sport=0-0 iif=41
dport=0-65535 oif=14(port22) gwy=172.16.202.1
source wildcard(1): 0.0.0.0/0.0.0.0
destination wildcard(1): 172.16.205.10/255.255.255.255
hit_count=1 last_used=2021-05-27 23:04:33
```

4. When the health link monitor status is down, verify that the policy route is active:

a. Verify the link health monitor status:

```
# diagnose sys link-monitor status
Link Monitor: test-1, Status: die, Server num(1), HA state: local(die), shared(die)
Flags=0x9 init log_downgateway, Create time: Fri May 28 15:20:15 2021
Source interface: port22 (14)
Gateway: 172.16.202.1
Interval: 500 ms
Service-detect: disable
Diffservcode: 000000
Class-ID: 0
  Peer: 10.1.100.22(10.1.100.22)
        Source IP(172.16.202.2)
        Route: 172.16.202.2->10.1.100.22/32, gwy(172.16.202.1)
        protocol: ping, state: die
                Packet lost: 11.000%
                Number of out-of-sequence packets: 0
                Recovery times(0/5) Fail Times(0/3)
                Packet sent: 7293, received: 3471, Sequence(sent/rcvd/exp): 7294/7281/7282
```

b. Verify the policy route list:

```
# diagnose firewall proute list
list route policy info(vf=root):
id=1 dscp_tag=0xff 0xff flags=0x0 tos=0x14 tos_mask=0xff protocol=0 sport=0-0 iif=41
dport=0-65535 oif=14(port22) gwy=172.16.202.1
```

```
source wildcard(1): 0.0.0.0/0.0.0.0
destination wildcard(1): 172.16.205.10/255.255.255.255
hit_count=1 last_used=2021-05-27 23:04:33
```

If the `update-policy-route` setting is enabled, the link health monitor would be down and the policy-based route would be disabled:

```
# diagnose firewall proute list
list route policy info(vf=root):
id=1 dscp_tag=0xff 0xff flags=0x8 disable tos=0x14 tos_mask=0xff protocol=0 sport=0-0 iif=41
dport=0-65535 oif=14(port22) gwy=172.16.202.1
source wildcard(1): 0.0.0.0/0.0.0.0
destination wildcard(1): 172.16.205.10/255.255.255.255
hit_count=1 last_used=2021-05-27 23:04:33
```

Add weight setting on each link health monitor server

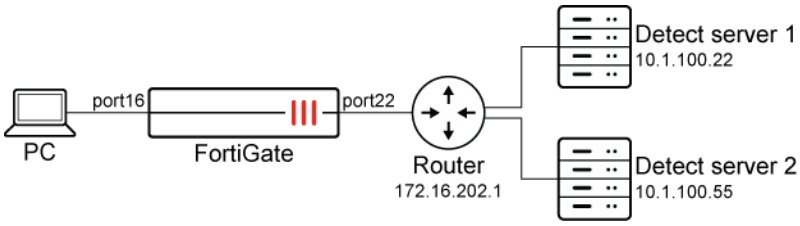
Prior to FortiOS 7.0.1, the link health monitor is determined to be dead when all servers are unreachable. Starting in 7.0.1, the link health monitor can configure multiple servers and allow each server to have its own weight setting. When the link health monitor is down, it will trigger static route updates and cascade interface updates if the weight of all dead servers exceeds the monitor's fail weight threshold.

```
config system link-monitor
    edit <name>
        set srcintf <interface>
        set server-config {default | individual}
        set fail-weight <integer>
        config server-list
            edit <id>
                set dst <address>
                set weight <integer>
            next
        end
    next
end
```

`server-config`	Set the server configuration mode: • default: all servers share the same attributes. • individual: some attributes can be specified for individual servers.
`fail-weight <integer>`	Threshold weight to trigger link failure alert (0 - 255, default = 0).
`server-list`	Configure the servers to be monitored by the link monitor.
`dst <address>`	Enter the IP address of the server to be monitored.
`weight <integer>`	Weight of the monitor to this destination (0 - 255, default = 0).

Examples

In the following topology, there are two detect servers that connect to the FortiGate through a router: server 1 (10.1.100.22) and server 2 (10.1.100.55).



Alive link health monitor

In this configuration, one server is dead and one server alive. The failed server weight is not over the threshold, so the link health monitor status is alive.

To configure the weight settings on the link health monitor:

1. Configure the link health monitor:

```
config system link-monitor
    edit "test-1"
        set srcintf "port22"
        set server-config individual
        set gateway-ip 172.16.202.1
        set failtime 3
        set fail-weight 40
        config server-list
            edit 1
                set dst "10.1.100.22"
                set weight 60
            next
            edit 2
                set dst "10.1.100.55"
                set weight 30
            next
        end
    next
end
```

2. Trigger server 2 to go down. The link monitor is still alive because the fail weight threshold has not been reached.
3. Verify the link health monitor status:

```
# diagnose sys link-monitor status test-1
Link Monitor: test-1, Status: alive, Server num(2), HA state: local(alive), shared(alive)
Flags=0x1 init, Create time: Fri Jun  4 17:23:29 2021
Source interface: port22 (14)
Gateway: 172.16.202.1
```

```
Interval: 500 ms
Service-detect: disable
Diffservcode: 000000
Class-ID: 0
Fail-weight (40): not activated
  Peer: 10.1.100.22(10.1.100.22)
        Source IP(172.16.202.2)
        Route: 172.16.202.2->10.1.100.22/32, gwy(172.16.202.1)
        protocol: ping, state: alive
                Latency(Min/Max/Avg): 0.417/0.585/0.530 ms
                Jitter(Min/Max/Avg): 0.007/0.159/0.057
                Packet lost: 0.000%
                Number of out-of-sequence packets: 0
                Fail Times(0/3)
                Packet sent: 239, received: 236, Sequence(sent/rcvd/exp): 240/240/241
  Peer: 10.1.100.55(10.1.100.55)
        Source IP(172.16.202.2)
        Route: 172.16.202.2->10.1.100.55/32, gwy(172.16.202.1)
        Fail weight 30 applied
        protocol: ping, state: dead
                Packet lost: 100.000%
                Number of out-of-sequence packets: 0
                Recovery times(0/5) Fail Times(1/3)
                Packet sent: 239, received: 3, Sequence(sent/rcvd/exp): 240/4/5
```

Dead link health monitor

In this configuration, one server is dead and one server alive. The failed server weight is over the threshold, so the link health monitor status is dead.

To configure the weight settings on the link health monitor:

1. Configure the link health monitor:

```
config system link-monitor
    edit "test-1"
        set srcintf "port22"
        set server-config individual
        set gateway-ip 172.16.202.1
        set failtime 3
        set fail-weight 40
        config server-list
            edit 1
                set dst "10.1.100.22"
                set weight 30
            next
            edit 2
                set dst "10.1.100.55"
                set weight 50
            next
        end
```

```
    next
end
```

2. Trigger server 2 to go down. The link monitor is dead because the fail weight threshold has been reached.
3. Verify the link health monitor status:

```
# diagnose sys link-monitor status  test-1
Link Monitor: test-1, Status: dead, Server num(2), HA state: local(dead), shared(dead)
Flags=0x9 init log_downgateway, Create time: Fri Jun  4 17:23:29 2021
Source interface: port22 (14)
Gateway: 172.16.202.1
Interval: 500 ms
Service-detect: disable
Diffservcode: 000000
Class-ID: 0
Fail-weight (40): activated
  Peer: 10.1.100.22(10.1.100.22)
        Source IP(172.16.202.2)
        Route: 172.16.202.2->10.1.100.22/32, gwy(172.16.202.1)
        protocol: ping, state: alive
                Latency(Min/Max/Avg): 0.393/0.610/0.520 ms
                Jitter(Min/Max/Avg): 0.009/0.200/0.095
                Packet lost: 0.000%
                Number of out-of-sequence packets: 0
                Fail Times(0/3)
                Packet sent: 680, received: 677, Sequence(sent/rcvd/exp): 681/681/682
  Peer: 10.1.100.55(10.1.100.55)
        Source IP(172.16.202.2)
        Route: 172.16.202.2->10.1.100.55/32, gwy(172.16.202.1)
        Fail weight 50 applied
        protocol: ping, state: dead
                Packet lost: 100.000%
                Number of out-of-sequence packets: 0
                Recovery times(0/5) Fail Times(1/3)
                Packet sent: 680, received: 3, Sequence(sent/rcvd/exp): 681/4/5
```

SLA link monitoring for dynamic IPsec and SSL VPN tunnels

The link health monitor settings can measure SLA information of dynamic VPN interfaces, which assign IP addresses to their clients during tunnel establishment. This includes SSL VPN tunnels, IPsec remote access, and IPsec site-to-site tunnels.

This feature currently only supports IPv4 and the ICMP monitoring protocol. In the IPsec tunnel settings, `net-device` must be disabled.

```
config system link-monitor
    edit <name>
```

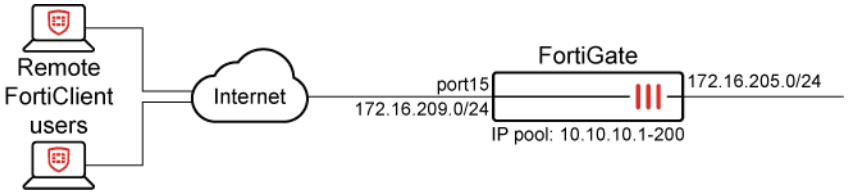
```
        set server-type {static | dynamic}
    next
end
```

To view the dial-up tunnel statistics:

```
# diagnose sys link-monitor tunnel {name | all} [<tunnel_name>]
```

Example

In this example, endpoint users dial up using FortiClient to create IPSec tunnels with the FortiGate and obtain IP addresses. The link monitor on the FortiGate's dynamic VPN interface detects the path quality to the endpoints.



To configure SLA link health monitoring in dynamic IPsec tunnels:

1. Configure the IPsec phase 1 interface:

```
config vpn ipsec phase1-interface
    edit "for_Branch"
        set type dynamic
        set interface "port15"
        set mode aggressive
        set peertype any
        set net-device disable
        set mode-cfg enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set dpd on-idle
        set dhgrp 5
        set xauthtype auto
        set authusrgrp "vpngroup"
        set assign-ip-from name
        set ipv4-netmask 255.255.255.0
        set dns-mode auto
        set ipv4-split-include "172.16.205.0"
        set ipv4-name "client_range"
        set save-password enable
        set psksecret **********
        set dpd-retryinterval 60
    next
end
```

2. Configure the IPsec phase 2 interface:

```
config vpn ipsec phase2-interface
    edit "for_Branch_p2"
        set phase1name "for_Branch"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
        set dhgrp 5
    next
end
```

3. Configure the dynamic interface:

```
config system interface
    edit "for_Branch"
        set vdom "root"
        set ip 10.10.10.254 255.255.255.255
        set type tunnel
        set remote-ip 10.10.10.253 255.255.255.0
        set snmp-index 100
        set interface "port15"
    next
end
```

4. Add the IPsec dial-up tunnel to the link health monitor:

```
config system link-monitor
    edit "1"
        set srcintf "for_Branch"
        set server-type dynamic
    next
end
```

5. Once endpoint users have connected using FortiClient, verify the tunnel information:

```
# get vpn ipsec tunnel summary
'for_Branch_0' 10.1.100.23:0  selectors(total,up): 1/1  rx(pkt,err): 21091/0  tx(pkt,err):
20741/0
'for_Branch_1' 10.1.100.13:0  selectors(total,up): 1/1  rx(pkt,err): 19991/0  tx(pkt,err):
20381/0
```

6. Verify the link health monitor status:

```
# diagnose sys link-monitor tunnel all
for_Branch_0 (1): state=alive, peer=10.10.10.1, create_time=2022-02-08 10:43:11, srcintf=for_
Branch, latency=0.162, jitter=0.018, pktloss=0.000%
for_Branch_1 (1): state=alive, peer=10.10.10.2, create_time=2022-02-08 10:49:24, srcintf=for_
Branch, latency=0.266, jitter=0.015, pktloss=0.000%
```

7. Manually add 200 ms latency on the path between the FortiGate and FortiClients.

8. Verify the link health monitor status again:

```
# diagnose sys link-monitor tunnel all
for_Branch_0 (1): state=alive, peer=10.10.10.1, create_time=2022-02-08 10:43:11, srcintf=for_
Branch, latency=200.177, jitter=0.021, pktloss=0.000%
for_Branch_1 (1): state=alive, peer=10.10.10.2, create_time=2022-02-08 10:49:24, srcintf=for_
Branch, latency=200.257, jitter=0.017, pktloss=0.000%
```

HA remote link monitoring

Remote link monitoring (also known as Remote IP Monitoring or PING server Monitoring) allows an FGCP cluster to detect failures beyond the directly connected network segment. By probing remote hosts, the primary unit can detect "soft" failures in the network path, such as a failed ISP gateway or a downstream router, and trigger an HA failover to a unit with a healthy path.

Prerequisites and constraints

- Cluster Mode: Units must be in an Active-Passive (A-P) FGCP cluster.
- Primary-Only Probing: In an FGCP cluster, only the primary unit performs active link monitoring.
- Routing Table: The secondary unit does not have an active routing table. While you can technically "ping" from a secondary unit, the ICMP requests are proxied through the primary unit via the HA heartbeat link.
- Multiple Servers: If a link monitor contains multiple servers, the monitor is only marked as failed if all configured servers stop responding.

General configuration steps

This section breaks down the configuration for this example into smaller procedures:

1. Configure the Link Monitor on page 884
2. Integrate monitoring into HA settings on page 885
3. Verify on page 886

Configure the Link Monitor

The link monitor defines the targets to be probed and the "penalty" (HA priority) to be applied if those targets become unreachable.

To configure the link monitor via the CLI:

```
config system link-monitor
    edit "WAN_Link_Monitor"
        set srcintf "port1"
        set server "8.8.8.8" "1.1.1.1"
        set protocol ping
        set ha-priority 10
```

```
        set interval 500
        set failtime 5
    next
end
```

Command	Description
srcintf	The Interface that receives the traffic to be monitored.
server	The IP address of the server(s) to be monitored.
protocol	The protocols used to monitor the server. Default is Ping
ha-priority	The penalty value added to the unit's health score upon failure. Default is 1.
interval	Detection interval in milliseconds (20 - 3600 * 1000 msec, default = 500).
failtime	Number of retry attempts before the server is considered down (1 - 3600, default = 5).

Integrate monitoring into HA settings

Once the monitor is defined, you must configure the HA to monitor the interface and set the threshold for failover.

To configure the HA failover logic:

```
config system ha
    set pingserver-monitor-interface "port1"
    set pingserver-failover-threshold 10
    set pingserver-flip-timeout 60
    set pingserver-secondary-force-reset disable
end
```

Command	Description
pingserver-monitor-interface	The Interfaces to check for remote IP monitoring.
pingserver-failover-threshold	Remote IP monitoring failover threshold (0 – 50, default = 0).
pingserver-flip-timeout	Time to wait in minutes before renegotiating after a remote IP monitoring failover. (6 - 2147483647, default = 60).
pingserver-secondary-force-reset	Enable/Disable force reset of secondary member after PING server failure. Default is enable

The Logic of failover

The FortiGate tracks a Global Penalty. When a link monitor fails, its ha-priority is added to this penalty. A failover is triggered only when the Global Penalty reaches or exceeds the pingserver-failover-threshold.

If...	Then...
ha-priority < threshold	No failover occurs.
ha-priority ≥ threshold	Failover is triggered immediately when the link is marked "dead."

Managing failback (the FLIP timer)

When a failover occurs, the cluster starts a FLIP Timer (defined by `pingserver-flip-timeout`). This prevents the cluster from rapidly switching back and forth if a link is flapping.

```
pingserver-secondary-force-reset
```

This setting dictates what happens when the FLIP timer expires:

- Enabled (Default): The penalty on the original primary is reset. If override is enabled, the original primary will attempt to take back the primary role. If the link is still down, it will fail over again once the timer expires, creating a FLIP-FLOP loop.
- Disabled: The penalty is not reset. The cluster remains on the new primary even if the link is restored on the original side. This is safer for unstable links but requires a manual reset or a failure on the new primary to switch back.

Verify

Verifying link status:

Run the following command to Check if the FortiGate considers the remote link "alive" or "dead":

```
# diagnose sys link-monitor status
```

Alive example:

```
Link Monitor: L_M_Port1, Status: alive, Server num(3), Flags=0x1 init, Create time: Thu March 12
10:45:46 2026
Source interface: port1 (11)
Interval: 500 ms
  Peer: 8.8.8.8(8.8.8.8)
        Source IP(192.168.1.1)
        Route: 192.168.1.1->8.8.8.8/32, gwy(192.168.1.2)
        protocol: ping, state: alive
        Latency(Min/Max/Avg): 1.921/2.103/1.973 ms
        Jitter(Min/Max/Avg): 0.003/0.142/0.034
        Packet lost: 0.000%
        Number of out-of-sequence packets: 0
                  Fail Times(0/5)
                  Packet sent: 4749, received: 4749, Sequence(sent/rcvd/exp): 4750/4750/4751
```

Dead example:

```
Link Monitor: L_M_Port1, Status: die, Server num(3), Flags=0x9 init, Create time: Thu March 12
10:45:46 2026
Source interface: port1 (11)
```

```
Interval: 500 ms
  Peer: 8.8.8.8(8.8.8.8)
        Source IP(192.168.1.1)
        Route: 192.168.1.1->8.8.8.8/32, gwy(192.168.1.2)
        protocol: ping, state: die
        Packet lost: 100.000%
        Number of out-of-sequence packets: 0
                  Recovery times(0/5) Fail Times(3/5)
                  Packet sent: 5862, received: 5563, Sequence(sent/rcvd/exp): 5863/5564/5565
```

IPv6

From an administrative point of view IPv6 works almost the same as IPv4 in FortiOS. The primary difference is the use of IPv6 format for addresses. See IPv6 overview on page 888 for more information.

By default, the IPv6 settings are not displayed in the GUI.

To enable IPv6 in the GUI:

1. Go to *System* > *Feature Visibility*.
2. Under *Core Features*, enable *IPv6*.
3. Click *Apply*.

Once enabled, you will be able to use IPv6 addresses as well as the IPv4 addressing for the following FortiGate firewall features which are not limited to:

- Interfaces
- DHCP Server
- DHCP Relay
- DNS
- Static Routes
- Firewall Policy
- NAT
- Addresses
- Virtual IPs
- IP Pools
- IPsec VPN
- GRE over IPsec

This section also contains the following topics:

- IPv6 overview on page 888
- IPv6 quick start on page 889
- Neighbor discovery proxy on page 892
- IPv6 address assignment on page 894
- NAT66, NAT46, NAT64, and DNS64 on page 909

- DHCPv6 relay on page 922
- IPv6 tunneling on page 923
- IPv6 Simple Network Management Protocol on page 935
- Dynamic routing in IPv6 on page 937
- IPv6 configuration examples on page 940

IPv6 overview

Internet Protocol version 6 (IPv6) is the latest version of the Internet Protocol (IP) and was developed to address the limitations of its predecessor, IPv4. The primary issue with IPv4 is its limited number of addresses, which are based on a 32-bit scheme and have a theoretical limit of 2 to the power of 32. In contrast, IPv6 uses a 128-bit address scheme, allowing for a much larger theoretical limit of 2 to the power of 128 addresses.

In simpler terms:

- IPv4 can support 4 294 967 296 addresses.
- IPv6 can support 340 282 366 920 938 463 463 374 607 431 768 211 456 addresses.

In addition to the expanded number of addresses, some of the other benefits of IPv6 include:

- More efficient routing due to reduction in the size of routing tables. This is achieved through hierarchical address allocation, which allows for more efficient routing of data packets.
- Reduced management requirements by supporting stateless auto-reconfiguration of hosts. This means that devices can automatically configure their network settings without the need for manual intervention.
- Improved methods to change Internet Service Providers. With IPv6, it is easier for users to switch between different ISPs without experiencing any service disruption.
- Better mobility support by providing seamless connection. This means that devices can move between different networks without losing their connection.
- Multi-homing. This allows a device to have multiple network connections, providing increased reliability and redundancy.
- Improved security with built-in support for IPsec. IPsec is a security protocol that provides authentication and encryption for data transmitted over a network.
- IPv6 offers scoped addresses with link-local, unique local, and global address spaces. This allows for more flexible addressing and improved network organization.

Address Type	Notation	Description	Example
Link-local Unicast	FE80::/10	Designed for use on a local link and are automatically configured on all interfaces. These addresses are not routable.	FE80::1
Unique Local Unicast	FC00::/7	Similar to IPv4 private addresses and can be used on your own network. They are not routable globally.	FC00::1 FD00::1
Global Unicast	2001::/3	Similar to IPv4 public addresses and can be used on the Internet. They are routable globally.	2001::1 3000::1

See Internet Protocol Version 6 Address Space for more information.

IPv6 quick start

This section provides an introduction to setting up a few basic IPv6 settings on the FortiGate. See Summary of steps on page 31 for more information about basic FortiGate administration.

> This chapter provides instructions for basic IPv6 configuration that should work in most cases, regardless of whether the device has an existing IPv4 configuration or is a new FortiGate device.

The topics covered in this section include:

- Configuring an interface on page 889
- Configuring the default route on page 890
- Configuring the DNS on page 890
- Configuring the address object on page 891
- Configuring the address group on page 891
- Configuring the firewall policy on page 892

Before starting, make sure to enable the IPv6 feature.

To enable IPv6 in the GUI:

1. Go to *System* > *Feature Visibility*.
2. Under *Core Features*, enable *IPv6*.
3. Click *Apply*. See IPv6 quick start example on page 940 for a sample configuration.

Configuring an interface

To configure an interface in the GUI:

1. Go to *Network* > *Interfaces*.
2. Select an interface and click *Edit*.
3. In the *Address* section, enter the *IPv6 Address/Prefix*.
4. In the *Administrative Access* section, select the IPv6 access options as needed (such as *PING*, *HTTPS*, and *SSH*).
5. Click *OK*.

To configure an interface in the CLI:

```
config system interface
    edit <interface name>
        config ipv6
            set ip6-address <IPv6 prefix>
            set ip6-allowaccess {ping | https | ssh | snmp | http | telnet | fgfm | fabric | scim
| probe-response}
        end
```

```
    next
end
```

Configuring the default route

Setting the default route enables basic routing to allow the FortiGate to return traffic to sources that are not directly connected. The gateway address should be your existing router or L3 switch that the FortiGate is connected to. Set the interface to be the interface the gateway is connected to.

To configure the default route in the GUI:

1. Go to *Network* > *Static Routes*.
2. Click *Create New* > *IPv6 Static Route*.
3. Leave the *Destination* prefix as ::/0. This is known as a default route, since it would match any IPv6 address.
4. Enter the *Gateway Address*.
5. Select an *Interface*.
6. Click *OK*.

To configure the default route in the CLI:

```
config router static6
    edit 0
        set gateway <IPv6 address>
        set device <interface name>
    next
end
```

Configuring the DNS

To configure a DNS domain list in the GUI:

1. Go to *Network* > *DNS*.
2. Under *IPv6 DNS Settings*, configure the primary and secondary DNS servers as needed.
3. Click *Apply*.

To configure a DNS domain list in the CLI:

```
config system dns
    set ip6-primary <IPv6 address>
    set ip6-secondary <IPv6 address>
end
```

Configuring the address object

Addresses define sources and destinations of network traffic and can be used in many functions such as firewall policies, ZTNA, and so on. When creating an IPv6 address object, several different types of addresses can be specified similar to IPv4 addresses. See Address types on page 2032 for more information.

To configure an IPv6 address in the GUI:

1. Go to *Policy & Objects* > *Addresses* and select *IPv6 Address*.
2. Select *Create new*.
3. Enter a *Name* for the address object.
4. In the *Type* field, select one of the types from the dropdown menu.
5. Configure the rest of the settings as required.
6. Click *OK*.

To configure an IPv6 address in the CLI:

```
config firewall address6
    edit <name>
        set type {ipprefix | iprange | fqdn | geography | dynamic | template | mac | route-tag}
    next
end
```

Configuring the address group

Address groups are designed for ease of use in the administration of the device. See Address group on page 2057 for more information.

To create an address group:

1. Go to *Policy & Objects* > *Addresses* and select *IPv6 Address Group*.
2. Go to *Create new*.
3. Enter a *Name* for the address object.
4. Select the *+* in the *Members* field. The *Select Entries* pane opens.
5. Select members of the group. It is possible to select more than one entry. Select the *x* icon in the field to remove an entry.
6. Enter any additional information in the *Comments* field.
7. Click *OK*.

To configure an address group in the CLI:

```
config firewall addrgrp6
    edit <name>
        set member <name>
    next
end
```

Configuring the firewall policy

A firewall policy must be in place for any traffic that passes through a FortiGate. See Firewall policy on page 1842 for more information.

To create a firewall policy in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Enter a *Name* and configure the following necessary settings:

Incoming Interface	Incoming (ingress) interface
Outgoing Interface	Outgoing (egress) interface
Source	Source IPv6 address name and address group names
Destination	Destination IPv6 address name and address group names
Schedule	Schedule name
Service	Service and service group names
Action	Policy action

To configure a firewall policy in the CLI:

```
config firewall policy
    edit <policyid>
        set srcintf <name>
        set dstintf <name>
        set action {accept | deny}
        set srcaddr6 <name>
        set dstaddr6 <name>
        set schedule <name>
        set service <name>
    next
end
```

See IPv6 quick start example on page 940 for a sample configuration.

Neighbor discovery proxy

This feature provides support for proxying the IPv6 Neighbor Discovery Protocol (NDP) to allow the following ICMP messages to be forwarded between upstream and downstream interfaces.

Message type	Function
Router Solicitation (RS)	Used by hosts to find any routers in a local segment and to request that they advertise their presence on the network.

Message type	Function
Router Advertisement (RA)	Used by an IPv6 router to advertise its presence on the network.
Neighbor Solicitation (NS)	Sent by a host to determine a remote host's link layer IPv6 address. Verifies the reachability of the neighbor or remote host in the Neighbor Discovery (ND) table
Neighbor Advertisement (NA)	Message used by the host to respond to an NS message. If an NS message is received by a remote host, it reciprocates with an NA message to the originating host. Additionally, this message is used by a host to announce a link layer address change.
Network Redirect	Message used by IPv6 routers to notify an originating host of a more optimal next-hop address for a specific destination. Only routers can send redirect messages. Redirect messages are exclusively processed by hosts.

Typically only one interface receives RA traffic, and the interface is automatically considered the upstream interface.

The Neighbor Discovery Protocol (NDP) is a layer 2 protocol that performs several tasks to improve the efficiency and consistency of data transmission across multiple networks and processes. NDP uses ICMPv6 messages to perform the following tasks:

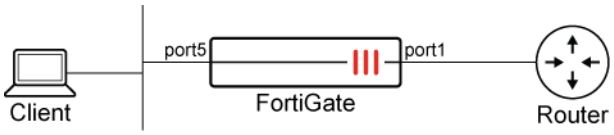
- Stateless auto-configuration: This enables the auto-configuration of IPv6 addresses without the need for a DHCP server. This means that each host on the network can automatically configure its unique IPv6 link-local address and global unicast address.
- Address Resolution: NDP performs a function similar to IPv4's Address Resolution Protocol (ARP), but instead of using ARP, it uses NDP to dynamically resolve IPv6 addresses to their corresponding MAC addresses.
- Neighbor Unreachability Detection (NUD): This function detects when a host is no longer reachable, allowing for more efficient routing and data transmission.
- Duplicate Address Detection (DAD): This function verifies that there is no duplication of unicast IPv6 addresses in the network, ensuring that each host has a unique address.

Configure ND proxy in the CLI using the following syntax:

```
config system nd-proxy
    set status {enable|disable}
    set member <interface> <interface> [<interface>...]
end
```

Option	Description
status	Enable/disable the use of neighbor discovery proxy.
member	List of interfaces using the neighbor discovery proxy.

In this example, the client is connected to a FortiGate device that is configured as an ND (Neighbor Discovery) proxy. Port1 is the upstream interface that receives Router Advertisement (RA) traffic, and port5 is the downstream interface that connects to the client. This setup allows the FortiGate device to facilitate communication between the client and the IPv6 router.



To configure ND Proxy on FortiGate:

1. Enable address auto-configuration on the upstream interface:

```
config system interface
    edit "port1"
        config ipv6
            set autoconf enable
        end
    next
end
```

2. Enable ND proxy on the interfaces:

```
config system nd-proxy
   set status enable
   set member "port1" "port5"
end
```

See RFC 4389 for more information on Neighbor Discovery Proxies (ND Proxy).

IPv6 address assignment

On the FortiGate, an interface can use the following methods to obtain an IPv6 address:

Method	Overview
IPv6 stateless address auto-configuration (SLAAC) on page 895	• Enables each network host to auto-configure a unique IPv6 address. • The lack of a state eliminates the need for a centralized server, thereby simplifying network management. • SLAAC does not provide DNS server addresses to hosts.
DHCPv6 stateful server on page 896	• Provides IPv6 addresses and additional information to hosts, such as a DNS server list and a domain name. • Offers more control to the administrator in assigning addresses, but requires extra configuration.
SLAAC with DHCPv6 stateless server on page 901	• Combines the benefits of SLAAC and DHCPv6. • Enables each host on the network to auto-configure a unique IPv6 address and allows them to obtain additional information, such as a DNS server list and a domain name.
IPv6 prefix delegation on page 904	• Enables internet service providers (ISPs) to provide organizations with a block of addresses that can be distributed throughout their network.

Deciding which method to employ should be based on the requirements of your system and your overall IT practice.

You can configure IPv6 using the CLI. To configure IPv6 using the GUI, ensure IPv6 is enabled by going to *System > Feature Visibility* and enabling *IPv6*.

IPv6 stateless address auto-configuration (SLAAC)

FortiGate can easily obtain an IPv6 address on any given interface using SLAAC (stateless address auto-configuration). SLAAC is designed only for IP assignments and does not provide DNS server addresses to hosts. See RFC 4862 for more information.

Use one of the following options to obtain a DNS server address:

- DHCPv6 stateful server on page 896
- SLAAC with DHCPv6 stateless server on page 901

In this example, the Enterprise Core FortiGate is connected to the First Floor FortiGate. The Enterprise Core FortiGate has SLAAC enabled, which allows the First Floor FortiGate to automatically obtain an IPv6 address using the auto-configuration IPv6 address option.

To enable IPv6 auto-configuration in the GUI:

1. Configure SLAAC on the Enterprise Core FortiGate:
   a. Go to *Network > Interfaces* and edit port5.
   b. Configure the following settings:

IPv6 addressing mode	Manual
IPv6 Address/Prefix	2001:db8:d0c:1::1/64
Stateless Address Auto-configuration (SLAAC)	Enable
IPv6 prefix list	Enable
IPv6 prefix	2001:db8:d0c:1::/64

   c. Click *OK*.
2. Configure the First Floor FortiGate to automatically obtain an IPv6 address:
   a. Go to *Network > Interfaces* and edit port5.
   b. Enable *Auto configure IPv6 address*. The First Floor FortiGate uses the prefix that it obtains from the Enterprise Core FortiGate interface, and automatically generates an IPv6 address.
3. Verify that the First Floor FortiGate automatically generated an IPv6 address:
   a. Go to *Network > Interfaces* and edit port5. The *IPv6 Address/Prefix* field is prepopulated with an IPv6 address.

To enable IPv6 auto-configuration in the CLI:

1. Configure SLAAC on the Enterprise Core FortiGate:

```
config system interface
    edit "port5"
        config ipv6
            set ip6-address 2001:db8:d0c:1::1/64
            set ip6-send-adv enable
            config ip6-prefix-list
                edit 2001:db8:d0c:1::/64
                next
            end
        end
    next
end
```

2. Configure the First Floor FortiGate to automatically obtain an IPv6 address:

```
config system interface
    edit "port5"
        config ipv6
            set autoconf enable
        end
    next
end
```

3. Verify that the First Floor FortiGate automatically generated an IPv6 address:

```
# diagnose ipv6 address list | grep port5
dev=4 devname=port5 flag= scope=0 prefix=64 addr=2001:db8:d0c:1:20c:29ff:fe4d:f83d
preferred=604419 valid=2591619 cstamp=976270 tstamp=979470
```

DHCPv6 stateful server

Similar to a DHCPv4 server, a DHCPv6 server is stateful. It can track client/server states, assign IP addresses to clients, and maintain full control over the process. In addition to assigning IP addresses, a DHCP server can also provide DNS server addresses. However, this IP address assignment method does not support failover protection. If the DHCPv6 server fails, hosts are unable to obtain an IPv6 address, and the network ceases to function. Furthermore, DHCPv6 does not provide gateway information. See RFC 3315 for more information.

DHCPv6 server/client can accommodate multiple DHCP options, including fqdn, hex (default), ip6, and string. Option 16 or Vendor Class Option is also supported, which allows IP pools and options to be assigned based on VCI matching for DHCPv6 server and client.

Example: basic configuration

In this example, the Enterprise Core FortiGate is connected to the First Floor FortiGate. The Enterprise Core FortiGate has a stateful DHCPv6 server configured that allows the First Floor FortiGate to automatically obtain an IPv6 address and DNS server address using the DHCP option.

To configure a DHCPv6 stateful server in the GUI:

1. Configure the Enterprise Core FortiGate with DHCPv6 stateful server:
   a. Go to *Network* > *Interfaces* and edit port5.
   b. Configure the following settings:

DHCPv6 Server	Enable
IPv6 subnet	2001:db8:d0c:1::/64
DNS service	Same as System DNS
Stateful server.	Enable
IP mode	IP range
Address range	2001:db8:d0c:1::a to 2001:db8:d0c:1::f

   c. Click *OK*.
2. Configure the First Floor FortiGate to obtain an IPv6 address using DHCP:
   a. Go to *Network* > *Interfaces* and edit port5.
   b. Set *IPv6 addressing mode* to *DHCP*.
   c. Click *OK*.
3. Verify that the First Floor FortiGate obtained an IPv6 address and DNS server address from the DHCPv6 server:
   a. Go to *Network* > *Interfaces* and edit port5. The *Obtained IP/Netmask* and *Acquired DNS* fields are populated with an IPv6 address.

To configure a DHCPv6 stateful server in the CLI:

1. Configure the Enterprise Core FortiGate with DHCPv6 stateful server:

```
config system dhcp6 server
    edit 1
        set dns-service default
        set subnet 2001:db8:d0c:1::/64
        set interface "port5"
        config ip-range
            edit 1
                set start-ip 2001:db8:d0c:1::a
                set end-ip 2001:db8:d0c:1::f
            next
        end
    next
end
```

2. Configure the First Floor FortiGate to obtain an IPv6 address using DHCP:

```
config system interface
    edit "port5"
        config ipv6
            set ip6-mode dhcp
        end
    next
end
```

3. Verify that the First Floor FortiGate obtained an IPv6 address and DNS server address from the DHCPv6 server:

```
# diagnose ipv6 address list | grep port5
dev=4 devname=port5 flag=P scope=0 prefix=128 addr=2001:db8:d0c:1::a preferred=4294967295
valid=4294967295 cstamp=1298969 tstamp=1298969ip6-address
# dia test application dnsproxy 3
worker idx: 0
VDOM: root, index=0, is primary, vdom dns is enabled, pip-0.0.0.0 dns_log=1
dns64 is disabled
DNS servers:
2001:db8:d0c:1::ff:53 vrf=0 tz=0 encrypt=none req=1 to=1 res=0 rt=0 ready=1 timer=0 probe=0
failure=1 last_failed=19812
```

Example: multiple DHCP options

The DHCPv6 server/client can accommodate multiple (more than three) DHCP options: fqdn, hex (default), ip6, and string.

To configure the options and IP range when the FortiGate is the DHCPv6 server:

```
config system dhcp6 server
    edit 1
        set dns-service default
        set subnet 2000:11:1:1::/64
        set interface "port3"
        config options
            edit 1
                set code 16
                set type string
                set value "vendor class option"
            next
            edit 2
                set code 15
                set type string
                set value "user class option"
            next
            edit 3
                set code 72
                set type fqdn
                set value "www.test.com"
            next
            edit 4
```

```
                set code 70
                set type ip6
                set ip6 2000:8:8:8::8
            next
            edit 5
                set code 96
                set type hex
                set value "0000013700084d53465420352e30"
            next
        end
        config ip-range
            edit 1
                set start-ip 2000:11:1:1::2
                set end-ip 2000:11:1:1::10
            next
        end
    next
    edit 2
        set subnet 2000:10:1:100::/64
        set interface "port2"
        config ip-range
            edit 1
                set start-ip 2000:10:1:100::11
                set end-ip 2000:10:1:100::11
            next
        end
    next
end
```

To configure the options and IP range when the FortiGate is a DHCPv6 client:

```
config system interface
    edit "port3"
        set vdom "vdom1"
        set allowaccess ping https ssh snmp http telnet
        set type physical
        set snmp-index 5
        config ipv6
            set ip6-mode dhcp
            config client-options
                edit 1
                    set code 16
                    set type hex
                    set value "0000013700084d53465420352e30"
                next
            end
            set ip6-allowaccess ping https ssh http telnet
        end
        set macaddr 00:09:0f:09:00:00
    next
end
```

Example: VCI matching in IP ranges

DHCPv6 supports option 16, also known as the Vendor Class Option. This allows IP pools and options assignment based on VCI matching for DHCPv6 server and client.

VCI matching in IP ranges

There are three cases:

1. If VCI matching is disabled in the IP range, then the DHCP client can get an IP address. This is the default.
2. If VCI matching is enabled and the VCI value matches the DHCP client, then the DHCP client can get an IP address.
3. If VCI matching is enabled and the VCI value does not match the DHCP client, then the DHCP client cannot get an IP address.

To enable VCI matching in an IP range:

```
config system dhcp6 server
    edit 2
        set subnet 2000:10:1:100::/64
        set interface "port2"
        config ip-range
            edit 1
                set start-ip 2000:10:1:100::11
                set end-ip 2000:10:1:100::11
                set vci-match enable
                set vci-string "PC1"
            next
        end
    next
end
```

VCI matching in DHCPv6 options

There are three cases:

1. If VCI matching is disabled in the option, then all options are sent back to the client. This is the default.
2. If VCI matching is enabled and the VCI value matches the DHCP client, then all options are sent back to the client.
3. If VCI matching is enabled and the VCI value does not matches the DHCP client, then the unmatched options cannot be sent back to the client.

To enable VCI matching in an option:

```
config system dhcp6 server
    edit 2
        set subnet 2000:10:1:100::/64
        set interface "port2"
        config options
            edit 1
```

```
                set code 16
                set type string
                set value "vendor class option"
                set vci-match enable
                set vci-string "PC1"
            next
        end
        config ip-range
            edit 1
                set start-ip 2000:10:1:100::11
                set end-ip 2000:10:1:100::11
            next
        end
    next
end
```

SLAAC with DHCPv6 stateless server

Using Stateless Address Auto Configuration (SLAAC) with a stateless DHCPv6 server provides a solution for obtaining other host configurations, such as DNS server addresses, while retaining the auto-configuration aspect of SLAAC. This approach also provides failover protection in the event that the DHCPv6 server fails. In addition to obtaining host configurations through the stateless DHCPv6 server, interfaces can also obtain gateway information through Router Advertisements (RAs). This allows for a robust and flexible IPv6 network configuration.

In this example, the Enterprise Core FortiGate is connected to the First Floor FortiGate. The Enterprise Core FortiGate has both SLAAC and stateless DHCPv6 server enabled. This allows the First Floor FortiGate to automatically obtain an IPv6 address using the *Auto configure IPv6 address* option and to acquire a DNS server address using the `dhcp6-information-request` option.

To enable IPv6 auto-configuration with DHCPv6 stateless server in the GUI:

1. Configure SLAAC on the Enterprise Core FortiGate:
   a. Go to *Network* > *Interfaces* and edit port5.
   b. Configure the following settings:

IPv6 addressing mode	Manual
IPv6 Address/Prefix	2001:db8:d0c:1::1/64
Stateless Address Auto-configuration (SLAAC)	Enable
IPv6 prefix list	Enable
IPv6 prefix	2001:db8:d0c:1::/64

   c. Click *OK*.

d. Input the following commands from the CLI:

```
config system interface
    edit "port5"
        config ipv6
            set ip6-other-flag enable
        end
    next
end
```

2. Configure DHCPv6 stateless server on the Enterprise Core FortiGate:
   a. Go to *Network* > *Interfaces* and edit port5.
   b. Configure the following settings:

DHCPv6 Server	Enable
DNS service	Same as System DNS
Stateful server	Disable

   c. Click *OK*.
3. Configure the First Floor FortiGate to automatically obtain an IPv6 address and DNS server address from the DHCPv6 server:
   a. Go to *Network* > *Interfaces* and edit port5.
   b. Enable *Auto configure IPv6 address*. The First Floor FortiGate uses the prefix obtained from the Enterprise Core FortiGate interface to automatically generate an IPv6 address.
   c. Input the following commands from the CLI:

```
config system interface
    edit "port5"
        config ipv6
            set dhcp6-information-request enable
        end
    next
end
```

   d. Click *OK*.
4. Verify that the First Floor FortiGate automatically generated an IPv6 address and obtained the DNS server address from the DHCPv6 server:
   a. Go to *Network* > *Interfaces* and edit port5. The IPv6 Address/Prefix field is populated with an IPv6 address
   b. Use the below CLI command to verify the DNS server address:

```
#dia test application dnsproxy 3
worker idx: 0
VDOM: root, index=0, is primary, vdom dns is enabled, pip-0.0.0.0 dns_log=1
dns64 is disabled
DNS servers:
2001:db8:d0c:1::1:53 vrf=0 tz=0 encrypt=none req=1 to=1 res=0 rt=0 ready=1 timer=0 probe=0
failure=1 last_failed=46738
...
```

To enable IPv6 auto-configuration with DHCPv6 stateless server in the CLI:

1. Configure SLAAC on the Enterprise Core FortiGate:

```
config system interface
    edit "port5"
        config ipv6
            set ip6-address 2001:db8:d0c:1::1/64
            set ip6-send-adv enable
            set ip6-other-flag enable
            config ip6-prefix-list
                edit 2001:db8:d0c:1::/64
            next
        end
    next
end
```

2. Configure DHCPv6 stateless server on the Enterprise Core FortiGate:

```
config system dhcp6 server
    edit 1
        set dns-service default
        set interface "port5"
    next
end
```

3. Configure the First Floor FortiGate to obtain an IPv6 address automatically:

```
config system interface
    edit "port5"
        config ipv6
            set autoconf enable
            set dhcp6-information-request enable
        end
    next
end
```

4. Verify that the First Floor FortiGate automatically generated an IPv6 address and obtained the DNS server address from the DHCPv6 server:

```
# diagnose ipv6 address list | grep port5
dev=4 devname=port5 flag= scope=0 prefix=64 addr=2001:db8:d0c:1:20c:29ff:fe4d:f83d
preferred=604681 valid=2591881 cstamp=1675487 tstamp=1772919
# dia test application dnsproxy 3
worker idx: 0
VDOM: root, index=0, is primary, vdom dns is enabled, pip-0.0.0.0 dns_log=1
dns64 is disabled
DNS servers:
2001:db8:d0c:1::1:53 vrf=0 tz=0 encrypt=none req=1 to=1 res=0 rt=0 ready=1 timer=0 probe=0
failure=1 last_failed=46738
…
```

IPv6 prefix delegation

IPv6 prefix delegation allows the dynamic assignment of an address prefix and DNS server address to an upstream interface. An upstream interface is typically the interface that is connected to an Internet Service Provider (ISP). This process also automates the assignment of prefixes to downstream interfaces. A downstream interface is any interface that is not an upstream interface and uses delegated addressing mode. Downstream interfaces can be configured to request specific IPv6 subnets from the upstream interface. Once a downstream interface receives the IPv6 address, other devices connected to the downstream interface can obtain an IPv6 address by using DHCPv6 or by configuring their own IP address using auto-configuration.

In this example, the Enterprise Core FortiGate is connected to a DHCPv6 server provided by the ISP through an upstream interface (port1). The Enterprise Core FortiGate is configured with a delegate interface (port5) to receive the IPv6 prefix and DNS server address from the upstream interface.

A downstream interface (port5) connects the First Floor FortiGate to the Enterprise Core FortiGate. The First Floor FortiGate interface (port5) is configured to receive the IPv6 address and DNS server address from the Enterprise Core FortiGate using DHCP addressing mode or auto-configuration.

Using the GUI or CLI to configure a downstream FortiGate to obtain the IPv6 and DNS server address from delegated interface using DHCP mode requires the following steps:

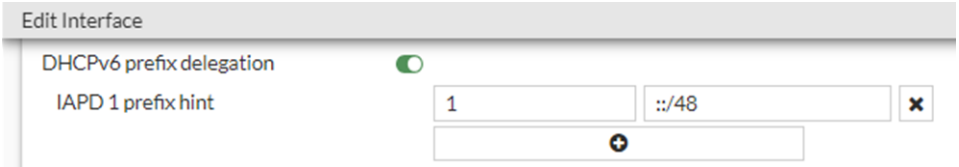
1. Configure the following items on the Enterprise Core FortiGate:
   - Upstream interface
   - Downstream interface
   - DHCPv6 server on the downstream interface.
2. Configure First Floor FortiGate to receive IPv6 prefix and DNS from the delegated interface.

Instead of configuring a DHCPv6 server on the downstream interface of the Enterprise Core FortiGate, you can configure SLAAC. See IPv6 prefix delegation with SLAAC on page 907.

GUI configuration

To configure the Enterprise Core FortiGate:

1. Configure the upstream interface on Enterprise Core FortiGate:
   a. Go to *Network* > *Interfaces* and edit port1.
   b. Enable *DHCPv6 prefix delegation*.
   c. Select the + in the IAPD prefix hint to open the ID and prefix field.
   d. Enter *1* for *ID* and *::/48* for *prefix* field. You can add two or more entries. Select the *x* icon in the field to remove an entry.



   e. Click *OK*.
2. Verify that the upstream interface obtained the prefix delegation, see Verify upstream interface obtained prefix delegation and DNS server address.
3. Configure the downstream interface on Enterprise Core FortiGate:
   a. Go to *Network* > *Interfaces* and edit port5.
   b. Set *IPv6 addressing mode* to *Delegated*.
   c. Enter *1* for *Identity association identifier* field.
   d. Set *IPv6 upstream* interface to *port1*.
   e. Click *OK*.
4. Verify that the downstream interface obtained an IPv6 address/prefix:
   a. Go to *Network* > *Interfaces* and edit port5. The *IPv6 Address/Prefix* field is prepopulated.
5. Configure the DHCPv6 server on the downstream interface:
   a. Go to *Network* > *Interfaces* and edit port5
   b. Enable *DHCPv6 Server*.
   c. Set *DNS service* to *Delegated*.
   d. From the *Upstream interface* dropdown list, select *port1*.
   e. Input the following commands from the CLI:

```
config system dhcp6 server
    edit 1
        set delegated-prefix-iaid 1
    next
end
```

   f. Enable *Stateful server*.
   g. Set *IP mode* to *Delegated*.
   h. Click *OK*.

To configure the First Floor FortiGate:

1. Configure the First Floor FortiGate interface using DHCP mode:
   a. Go to *Network* > *Interfaces* and edit the port5.
   b. Set *IPv6 addressing mode* to *DHCP*. This allows the First Floor FortiGate to obtain the IPv6 prefix and DNS from the delegated interface.
   c. Click *OK*.
2. Verify that the First Floor FortiGate obtained an IPv6 address and the DNS server address from the delegated interface:
   a. Go to *Network* > *Interfaces* and edit port5. The *Obtained IP/Netmask* and *Acquired DNS* fields are prepopulated with an IPv6 address.

CLI configuration

Using the CLI to configure a downstream FortiGate to obtain the IPv6 and DNS server address from delegated interface using DHCP mode requires the following steps:

To configure the Enterprise Core FortiGate:

1. Configure the upstream interface on the Enterprise Core FortiGate:

```
config system interface
    edit "port1"
        config ipv6
            set dhcp6-prefix-delegation enable
            config dhcp6-iapd-list
                edit 1
                    set prefix-hint ::/48
                next
            end
        end
    next
end
```

2. Verify that the upstream interface obtained a prefix delegation and DNS server address:

```
config system interface
     edit port1
        config ipv6
Enterprise Core FortiGate # get
ip6-mode            : static
…
dhcp6-prefix-delegation: enable
delegated-prefix iaid 1     : 2001:db8:d0c::/48
preferred-life-time         : 4294967295
valid-life-time     : 4294967295
delegated-DNS1      : 2001:db8::35
delegated-DNS2      : ::
…
dhcp6-iapd-list:
    == [ 1 ]
    iaid:     1       prefix-hint: ::/48           prefix-hint-plt: 604800           prefix-
hint-vlt: 2592001
```

3. Configure the downstream interface on the Enterprise Core FortiGate:

```
config system interface
    edit "port5"
        config ipv6
            set ip6-mode delegated
            set ip6-delegated-prefix-iaid 1
            set ip6-upstream-interface "port1"
        end
    next
end
```

4. Verify that the downstream interface obtained an IPv6 address/prefix:

```
config system interface
    edit "port5"
```

```
        config ipv6
Enterprise Core FortiGate # get
ip6-mode            : delegated
nd-mode             : basic
ip6-address         : 2001:db8:d0c::/48
…
ip6-delegated-prefix-iaid: 1
ip6-upstream-interface: port1
ip6-subnet          : ::/0
```

5. Configure the DHCPv6 server on the downstream interface:

```
config system dhcp6 server
    edit 1
        set dns-service delegated
        set interface "port5"
        set upstream-interface "port1"
        set delegated-prefix-iaid 1
        set ip-mode delegated
    next
end
```

To configure the First Floor FortiGate:

1. Configure the First Floor FortiGate interface to use DHCP mode:

```
config system interface
    edit "port5"
        config ipv6
            set ip6-mode dhcp
        end
    next
end
```

2. Verify that the First Floor FortiGate obtained an IPv6 address and DNS server address from the delegated interface:

```
# diagnose ipv6 address list | grep port5
dev=7 devname=port5 flag=P scope=0 prefix=128 addr=2001:db8:d0c::1 preferred=4294967295
valid=4294967295 cstamp=43208325 tstamp=43208325
# dia test application dnsproxy 3
worker idx: 0
VDOM: root, index=0, is primary, vdom dns is enabled, pip-0.0.0.0 dns_log=1
dns64 is disabled
DNS servers:
2001:db8::35:53 vrf=0 tz=0 encrypt=none req=3 to=2 res=0 rt=1046 ready=1 timer=0 probe=0
failure=2 last_failed=65131
```

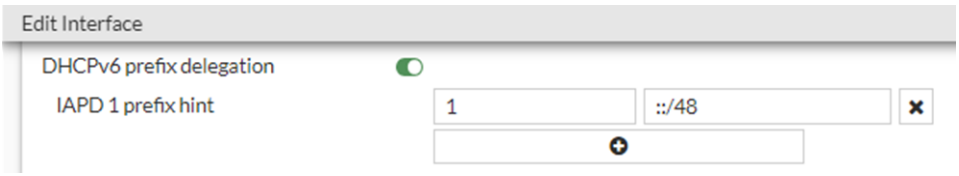
IPv6 prefix delegation with SLAAC

A downstream FortiGate can be configured to obtain the IPv6 address and DNS server address from a delegated interface using SLAAC instead of DHCPv6. Following is a summary of the configuration steps:

1. Configure the following items on the Enterprise Core FortiGate:
   - Upstream interface
   - Downstream interface
   - SLAAC on the downstream interface
2. Configure the First Floor FortiGate to receive an IPv6 prefix and DNS server address from the delegated interface.

To configure the Enterprise Core FortiGate:

1. Configure the upstream interface on Enterprise Core FortiGate:
   a. Go to *Network* > *Interfaces* and edit port1.
   b. Enable *DHCPv6 prefix delegation*.
   c. Select the *+* in the IAPD prefix hint to open the ID and prefix field.
   d. Enter *1* for *ID* and *::/48* for *prefix* field. You can add two or more entries. Select the *x* icon in the field to remove an entry.



   e. Click *OK*.
2. Verify that the upstream interface obtained the prefix delegation, see Verify upstream interface obtained prefix delegation and DNS server address.
3. Configure the downstream interface on Enterprise Core FortiGate:
   a. Go to *Network* > *Interfaces* and edit port5.
   b. Set *IPv6 addressing mode* to *Delegated*.
   c. Enter *1* for *Identity association identifier* field.
   d. Set *IPv6 upstream* interface to *port1*.
   e. Click *OK*.
4. Verify that the downstream interface obtained an IPv6 address/prefix:
   a. Go to *Network* > *Interfaces* and edit port5. The *IPv6 Address/Prefix* field is prepopulated.
5. Configure SLAAC on the downstream interface:

```
config system interface
    edit "port5"
        config ipv6
            set ip6-mode delegated
            set ip6-send-adv enable
            set ip6-delegated-prefix-iaid 1
            set ip6-upstream-interface "port1"
            config ip6-delegated-prefix-list
                edit 1
                    set upstream-interface "port1"
                    set delegated-prefix-iaid 1
                    set subnet 0:0:0:1::/64
                    set rdnss-service delegated
                next
```

```
            end
        end
    next
end
```

To configure the First Floor FortiGate:

1. Configure the First Floor FortiGate interface using auto-configure:

```
config system interface
    edit "port5"
        config ipv6
            set autoconf enable
        end
    next
end
```

2. Verify that the First Floor FortiGate automatically generated an IPv6 address and obtained the DNS server address from the delegated interface:

```
# diagnose ipv6 address list | grep port5
dev=4 devname=port5 flag= scope=0 prefix=64 addr=2000:db8:d0c:1:20c:29ff:fe4d:f847
preferred=4294967295 valid=4294967295 cstamp=17203697 tstamp=17225377
```

FortiGate can send DNS server addresses using Router Advertisement (RA), which allows any device that is capable of receiving DNS server addresses by using RA to obtain DNS server addresses.

Additionally, FortiGate can receive DNS server addresses through the use of SLAAC with a DHCPv6 stateless server, even though it is currently unable to receive DNS server addresses using RA due to RFC 4862 implementation. See SLAAC with DHCPv6 stateless server on page 901 for more information.

NAT66, NAT46, NAT64, and DNS64

NAT66, NAT46, NAT64, and DNS64 each offer their own distinct strategies and solutions to tackle the obstacles encountered during the transition from IPv4 to IPv6. This section provides a concise overview of these methods.

Method	Overview
NAT66	• NAT66 is used for translating an IPv6 source or destination address to a different IPv6 source or destination address. • Not a common method, since most IPv6 networks do not require NAT66.
NAT46	• NAT46 is used to translate IPv4 addresses to IPv6 addresses. • Enable a client on an IPv4 network to communicate transparently with a server on an IPv6 network.

Method	Overview
NAT64 and DNS64	• NAT64 is used to translate IPv6 addresses to IPv4 addresses. • Enable a client on an IPv6 network to communicate transparently with a server on an IPv4 network. • Typically used when networks are being transitioned from IPv4 to IPv6. • NAT64 is typically employed in tandem with DNS64. DNS64 is responsible for synthesizing AAAA records from A records.

Note that these are broad use cases and the specific use of each type of NAT can vary depending on the network configuration and requirement.

NAT66 policy

NAT66 is used for translating an IPv6 source or destination address to a different IPv6 source or destination address. NAT66 is not as common or as important as IPv4 NAT, as many IPv6 addresses do not need NAT66 as much as IPv4 NAT. However, NAT66 can be useful for a number of reasons. For example, you may have changed the IP addresses of some devices on your network but want traffic to still appear to be coming from their old addresses. You can use NAT66 to translate the source addresses of packets from the devices to their old source addresses.

In FortiOS, NAT66 options can be added to an IPv6 security policy. Configuring NAT66 is very similar to configuring NAT in an IPv4 security policy.

To configure NAT66:

1. Go to *Policy & Objects > Firewall Policy*.
2. Click *Create New*.
3. Configure the required policy parameters.
4. Enable *NAT* and select *Use Outgoing Interface Address*. For packets that match this policy, its source IP address is translated to the IP address of the outgoing interface.
5. Click *OK*.

Nat66 can also translate one IPv6 source address to another address that is not the same as the address of the existing interface. You can do this using IP pools.

To configure the IPv6 pool:

1. Go to *Policy & Objects > IP Pools* and navigate to the *IPv6 IP Pool* tab.
2. Click *Create new*.
3. Enter the following:

Name	test-ippool6-1
External IP Range	2000:172:16:101::1-2000:172:16:101::1

4. Click *OK*.

To use the IPv6 pool in the firewall policy:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Click *Create New* or edit an existing policy.
3. Configure the required policy parameters.
4. Enable *NAT* and select *Use Dynamic IP Pool*.
5. Click *OK*.

NAT66 destination address translation

NAT66 can also be used to translate destination addresses. This is done in an IPv6 policy by using IPv6 virtual IPs. For example, the destination address 2001:db8::dd can be mapped to 2001:db8::ee.

To configure the IPv6 VIP:

1. Go to *Policy & Objects* > *Virtual IPs* and navigate to the *Virtual IP* tab.
2. Click *Create new*.
3. Enter the following:

VIP type	IPv6
Name	example-vip6
External IP address/range	2001:db8::dd
Map to IPv6 address/range	2001:db8::ee

4. Click *OK*.

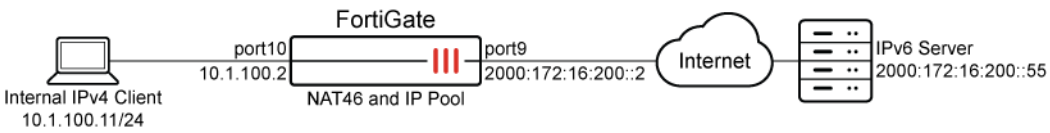
To use the IPv6 VIP in the firewall policy:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Click *Create New* or edit an existing policy.
3. Configure the required policy parameters.
4. In the *Destination* field, select *example-vip6* from the dropdown menu.
5. Click *OK*.

NAT46 policy

NAT46 refers to the mechanism that allows IPv4 addressed hosts to communicate with IPv6 hosts. Without such a mechanism, IPv4 environments cannot connect to IPv6 networks.

Sample topology



In this example, an IPv4 client tries to connect to an IPv6 server. A VIP is configured on FortiGate to map the server IPv6 IP address `2000:172:16:200:55` to an IPv4 address `10.1.100.55`. On the other side, an IPv6 IP pool is configured and the source address of packets from client are changed to the defined IPv6 address. In this setup, the client PC can access the server by using IP address `10.1.100.55`.

Sample configuration

To configure NAT46 in the GUI:

1. Enable IPv6:
   a. Go to *System* > *Feature Visibility*.
   b. In the *Core Features* section, enable *IPv6*.
   c. Click *Apply*.
2. Configure the VIP:
   a. Go to *Policy & Objects* > *Virtual IPs* and navigate to the *Virtual IP* tab.
   b. Click *Create new*.
   c. Enter the following:

Name	vip46_server
Interface	port2
Type	Static NAT
External IP address/range	10.1.100.55
Map to IPv6 address/range	2000:172:16:200::55

   d. Click *OK*.
3. Configure the IPv6 IP pool:
   a. Go to *Policy & Objects* > *IP Pools* and navigate to the *IPv6 IP Pool* tab.
   b. Click *Create new*.
   c. Enter the following:

Name	client_external
External IP address/range	2000:172:16:201::-2000:172:16:201::7
NAT46	Enable

   d. Click *OK*.
4. Configure the firewall policy:
   a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
   b. Enter the following:

Name	policy46-1
Incoming Interface	port10
Outgoing Interface	port9

Source	all
Destination	vip46_server
Schedule	always
Service	ALL
Action	ACCEPT
NAT	NAT46
IP Pool Configuration	client_external

c. Configure the other settings as needed.

d. Click *OK*.

To configure NAT46 in the CLI:

1. Enable IPv6:

```
config system global
    set gui-ipv6 enable
end
```

2. Configure the VIP:

```
config firewall vip
    edit "vip46_server"
        set extip 10.1.100.55
        set nat44 disable
        set nat46 enable
        set extintf "port2"
        set ipv6-mappedip 2000:172:16:200::55
    next
end
```

3. Configure the IPv6 IP pool:

```
config firewall ippool6
    edit "client_external"
        set startip 2000:172:16:201::
        set endip 2000:172:16:201::7
        set nat46 enable
    next
end
```

4. Configure the firewall policy:

```
config firewall policy
    edit 2
        set name "policy46-1"
        set srcintf "port10"
```

```
        set dstintf "port9"
        set action accept
        set nat46 enable
        set srcaddr "all"
        set dstaddr "vip46_server"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set auto-asic-offload disable
        set ippool enable
        set poolname6 "client_external"
    next
end
```

Sample troubleshooting

To trace the flow and troubleshoot:

```
# diagnose debug flow filter saddr 10.1.100.11
# diagnose debug flow show function-name enable
show function name
# diagnose debug flow show iprope enable
show trace messages about iprope
# diagnose debug flow trace start 5

id=20085 trace_id=1 func=print_pkt_detail line=5401 msg="vd-root:0 received a packet(proto=1,
10.1.100.11:27592->10.1.100.55:2048) from port10. type=8, code=0, id=27592, seq=1."
id=20085 trace_id=1 func=init_ip_session_common line=5561 msg="allocate a new session-000003b9"
id=20085 trace_id=1 func=iprope_dnat_check line=4948 msg="in-[port10], out-[]"
id=20085 trace_id=1 func=iprope_dnat_tree_check line=822 msg="len=1"
id=20085 trace_id=1 func=__iprope_check_one_dnat_policy line=4822 msg="checking gnum-100000
policy-1"
id=20085 trace_id=1 func=get_vip46_addr line=998 msg="find DNAT46: IP-2000:172:16:200::55, port-
27592"
id=20085 trace_id=1 func=__iprope_check_one_dnat_policy line=4904 msg="matched policy-1,
act=accept, vip=1, flag=100, sflag=2000000"
id=20085 trace_id=1 func=iprope_dnat_check line=4961 msg="result: skb_flags-02000000, vid-1, ret-
matched, act-accept, flag-00000100"
id=20085 trace_id=1 func=fw_pre_route_handler line=183 msg="VIP-10.1.100.55:27592, outdev-unkown"
id=20085 trace_id=1 func=__ip_session_run_tuple line=3220 msg="DNAT 10.1.100.55:8-
>10.1.100.55:27592"
id=20085 trace_id=1 func=vf_ip_route_input_common line=2594 msg="find a route: flag=80000000 gw-
10.1.100.55 via root"
id=20085 trace_id=1 func=ip4_nat_af_input line=601 msg="nat64 ipv4 received a packet proto=1"
id=20085 trace_id=1 func=__iprope_check line=2112 msg="gnum-100012, check-ffffffffa0024ebe"
id=20085 trace_id=1 func=__iprope_check_one_policy line=1873 msg="checked gnum-100012 policy-1,
ret-matched, act-accept"
id=20085 trace_id=1 func=__iprope_user_identity_check line=1677 msg="ret-matched"
```

```
id=20085 trace_id=1 func=get_new_addr46 line=1047 msg="find SNAT46: IP-2000:172:16:201::13(from
IPPOOL), port-27592"
id=20085 trace_id=1 func=__iprope_check_one_policy line=2083 msg="policy-1 is matched, act-accept"
id=20085 trace_id=1 func=__iprope_check line=2131 msg="gnum-100012 check result: ret-matched, act-
accept, flag-08050500, flag2-00200000"
id=20085 trace_id=1 func=iprope_policy_group_check line=4358 msg="after check: ret-matched, act-
accept, flag-08050500, flag2-00200000"
id=20085 trace_id=1 func=resolve_ip6_tuple line=4389 msg="allocate a new session-00000081"
```

NAT64 policy and DNS64 (DNS proxy)

NAT64 policy translates IPv6 addresses to IPv4 addresses so that a client on an IPv6 network can communicate transparently with a server on an IPv4 network.

NAT64 policy is usually implemented in combination with the DNS proxy called DNS64. DNS64 synthesizes AAAA records from A records and is used to synthesize IPv6 addresses for hosts that only have IPv4 addresses. DNS proxy and DNS64 are interchangeable terms.

Sample topology

In this example, a host on the internal IPv6 network communicates with `ControlPC.qa.fortinet.com` that only has IPv4 address on the Internet. Central NAT is disabled.

1. The host on the internal network does a DNS lookup for `ControlPC.qa.fortinet.com` by sending a DNS query for an AAAA record for `ControlPC.qa.fortinet.com`.
2. The DNS query is intercepted by the FortiGate DNS proxy. The DNS proxy performs an A-record query for `ControlPC.qa.fortinet.com` and gets back an RRSet containing a single A record with the IPv4 address `172.16.200.55`.
3. The DNS proxy then synthesizes an AAAA record. The IPv6 address in the AAAA record begins with the configured NAT64 prefix in the upper 96 bits and the received IPv4 address in the lower 32 bits. By default, the resulting IPv6 address is `64:ff9b::172.16.200.55`.
4. The host on the internal network receives the synthetic AAAA record and sends a packet to the destination address `64:ff9b::172.16.200.55`.
5. The packet is routed to the FortiGate internal interface (port10) where it is accepted by the NAT64 security policy.
6. The FortiGate translates the destination address of the packets from IPv6 address `64:ff9b::172.16.200.55` to IPv4 address `172.16.200.55` and translates the source address of the packets to `172.16.200.200` (or another address in the IP pool range) and forwards the packets out the port9 interface to the Internet.

Sample configuration

To configure a NAT64 policy with DNS64 in the GUI:

1. Enable IPv6 and DNS database:
   a. Go to *System > Feature Visibility*.
   b. In the *Core Features* section, enable *IPv6*.
   c. In the *Additional Features* section, enable *DNS Database*.
   d. Click *Apply*.
2. Enable DNS proxy on the IPv6 interface:
   a. Go to *Network > DNS Servers*.
   b. In the *DNS Service on Interface* table, click *Create New*.
   c. For *Interface*, select *port10*.
   d. For *Mode*, select *Forward to System DNS*.
   e. Click *OK*.
3. Configure the IPv6 DHCP server:
   a. Go to *Network > Interfaces* and edit port10.
   b. Enable *DHCPv6 Server* and enter the following:

IPv6 subnet	2001:db8:1::/64
DNS service	Specify
DNS server 1	2001:db8:1::10

   c. Click *OK*.
4. Configure the IPv6 VIP for the destination IPv6 addresses:
   These are all of the IPv6 addresses that the FortiGate DNS proxy synthesizes when an IPv6 device performs a DNS query that resolves to an IPv4 Address. In this example, the synthesized IPv6 address in the AAAA record begins with the configured NAT64 prefix in the upper 96 bits, so the VIP is for all the IPv6 addresses that begin with 64:ff9b.
   a. Go to *Policy & Objects > Virtual IPs* and navigate to the *IPv6 Virtual IP* tab.
   b. Click *Create new*.
   c. Enter the following:

Name	vip6
Eternal IP address/range	64:ff9b::-64:ff9b::ffff:ffff
Map to IPv4 address/range	Use Embedded

   d. Click *OK*.
5. Configure the IPv6 firewall address for the internal network:
   a. Click *Create New > Address*.
   b. Enter the following:

Category	IPv6 Address

Name	internal-net6
Type	IPv6 Subnet
IP/Netmask	2001:db8:1::/48

c. Click *OK*.

6. Configure the IP pool containing the IPv4 address that is used as the source address of the packets exiting port9:
   a. Go to *Policy & Objects* > *IP Pools* and navigate to the *IP Pool* tab.
   b. Click *Create new*.
   c. Enter the following:

Name	exit-pool4
Type	Overload
External IP address/range	172.16.200.200-172.16.200.207
NAT64	Enable

> *External IP address/range* must start and end on the boundaries of a valid subnet. For example, 172.16.200.0-172.16.200.7 and 172.16.200.16-172.16.200.31 are a valid subnets (/29 and /28 respectively).

   d. Click *OK*.
7. Configure the NAT64 policy:
   a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
   b. Enter the following:

Name	policy64-1
Incoming Interface	port10
Outgoing Interface	port9
Source	internal-net6
Destination	vip6
Schedule	always
Service	ALL
Action	ACCEPT
NAT	NAT64
IP Pool Configuration	exit-pool4

   c. Click *OK*.

To configure a NAT64 policy with DNS64 in the CLI:

1. Enable IPv6 and DNS database:

```
config system global
   set gui-ipv6 enable
end
```

```
config system settings
   set gui-dns-database enable
end
```

2. Enable DNS proxy on the IPv6 interface:

```
config system dns-server
   edit "port10"
       set mode forward-only
   next
end
```

3. Configure the IPv6 DHCP server:

```
config system dhcp6 server
   edit 1
       set subnet 2001:db8:1::/64
       set interface "port10"
       set dns-server1 2001:db8:1::10
   next
end
```

4. Configure the IPv6 VIP for the destination IPv6 addresses:

```
config firewall vip6
    edit "vip6"
        set extip 64:ff9b::-64:ff9b::ffff:ffff
        set embedded-ipv4-address enable
    next
end
```

5. Configure the IPv6 firewall address for the internal network:

```
config firewall address6
    edit "internal-net6"
        set ip6 2001:db8:1::/48
    next
end
```

6. Configure the IP pool containing the IPv4 address that is used as the source address of the packets exiting port9:

```
config firewall ippool
    edit "exit-pool4"
        set startip 172.16.200.200
```

```
        set endip 172.16.200.207
        set nat64 enable
    next
end
```

External IP address/range must start and end on the boundaries of a valid subnet. For example, 172.16.200.0-172.16.200.7 and 172.16.200.16-172.16.200.31 are a valid subnets (/29 and /28 respectively).

7. Configure the NAT64 policy:

```
config firewall policy
    edit 1
        set name "policy64-1"
        set srcintf "port10"
        set dstintf "port9"
        set action accept
        set nat64 enable
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 internal-net6
        set dstaddr6 vip6
        set schedule "always"
        set service "ALL"
        set ippool enable
        set poolname "exit-pool4"
    next
end
```

To enable DNS64 and related settings using the CLI:

Enabling DNS64 means that all IPv6 traffic received by the current VDOM can be subject to NAT64 if the source and destination address matches an NAT64 security policy.

By default, the setting `always-synthesize-aaaa-record` is enabled. If you disable this setting, the DNS proxy (DNS64) will attempt to find an AAAA records for queries to domain names and therefore resolve the host names to IPv6 addresses. If the DNS proxy cannot find an AAAA record, it synthesizes one by adding the NAT64 prefix to the A record.

```
config system dns64
    set status {enable | disable}
    set dns64-prefix <ipv6-prefix>
    set always-synthesize-aaaa-record {enable | disable}
end
```

By default, the `dns64-prefix` is `64:ff9b::/96`.

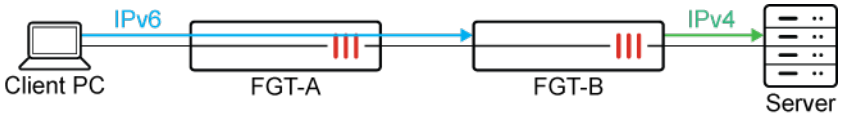
Port block allocation with NAT64

Port block allocation (PBA) support for NAT64 is supported on FortiGates with a hyperscale firewall license. This feature has been added to mainstream FortiOS to make it available to non-hyperscale customers, including customers running a VM version of FortiOS. Hyperscale firewall logging is designed for optimal performance and does not have the same detailed logging features as are available for non-hyperscale traffic.

```
config firewall ippool
    edit <name>
        set type port-block-allocation
        set nat64 enable
    next
end
```

Example

In this example, a NAT64 virtual IPv6 address and PBA IP pool are configured on FGT-B. IPv6 traffic from the client PC is able to access the IPv4 server.



The IPv6 addresses used in this example are for demonstrative purposes only and should not be used in your environment.

The 2001:db8::/32 prefix is a special IPv6 prefix designated for use in documentation examples. See RFC 3849 for more information.

To configure PBA for NAT64 on FGT-B:

1. Configure the IP pools and enable NAT 64:

```
(vdom1) config firewall ippool
    edit "ippool4-1072390-1"
        set type port-block-allocation
        set startip 172.16.164.164
        set endip 172.16.164.164
        set block-size 64
        set num-blocks-per-user 1
        set pba-timeout 60
        set nat64 enable
    next
    edit "ippool4-1072390-2"
        set type port-block-allocation
        set startip 172.16.164.165
        set endip 172.16.164.165
        set block-size 64
        set num-blocks-per-user 1
        set pba-timeout 60
```

```
        set nat64 enable
    next
end
```

2. Configure the virtual IP for IPv6:

```
(vdom1) config firewall vip6
    edit "vip64-1072390"
        set extip 64:ff9b::-64:ff9b::ffff:ffff
        set nat66 disable
        set nat64 enable
        set embedded-ipv4-address enable
    next
end
```

3. Configure the firewall policy:

```
(vdom1) config firewall policy
    edit 1072390
        set srcintf "port7"
        set dstintf "port1"
        set action accept
        set nat64 enable
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "vip64-1072390"
        set schedule "always"
        set service "ALL"
        set auto-asic-offload disable
        set ippool enable
        set poolname "ippool4-1072390-1" "ippool4-1072390-2"
    next
end
```

4. Send IPv6 packets from the client to access the IPv4 server.
5. Verify the NAT64 sessions:

```
(vdom1) # diagnose sys session6 stat
misc info:  session_count=128 setup_rate=0 exp_count=0 reflect_count=0 clash=0
      memory_tension_drop=0 ephemeral=0/0 removeable=0 extreme_low_mem=0
      npu_session_count=0
      nturbo_session_count=0
delete=0, flush=3, dev_down=0/0 ses_walkers=0
```

There are 128 sessions allocated to the two PBA IP pools.

6. Verify the PBA IP pools status:

```
(vdom1) # diagnose firewall ippool list
list ippool info:(vf=vdom1)
ippool ippool4-1072390-1: id=1, block-sz=64, num-block=1, fixed-port=no, use=5
      nat ip-range=172.16.164.164-172.16.164.164 start-port=5117, num-pba-per-ip=944
```

```
      clients=2, inuse-NAT-IPs=1
      total-PBAs=944, inuse-PBAs=1, expiring-PBAs=1, free-PBAs=99.89%
      allocate-PBA-times=2, reuse-PBA-times=0
ippool ippool4-1072390-2: id=2, block-sz=64, num-block=1, fixed-port=no, use=4
      nat ip-range=172.16.164.165-172.16.164.165 start-port=5117, num-pba-per-ip=944
      clients=1, inuse-NAT-IPs=1
      total-PBAs=944, inuse-PBAs=1, expiring-PBAs=0, free-PBAs=99.89%
      allocate-PBA-times=1, reuse-PBA-times=0
```

Each IP pool uses one IPv4 address and one block (64 ports) for SNAT.

7. Verify the PBAs in the IP pools in the current VDOM:

```
(vdom1) # diagnose firewall ippool list pba
user 2001:db8:d0c:1::1, 172.16.164.164, 5181-5244, idx=1, use=66
user 2001:db8:d0c:1::1, 172.16.164.165, 5117-5180, idx=0, use=66
```

This output includes the client IP, NAT IP, NAT port range, port block index, and a kernel reference counter.

8. Verify the NAT IPs in use in the current VDOM:

```
(vdom1) # diagnose firewall ippool list nat-ip
NAT-IP 172.16.164.164, pba=1, use=3
NAT-IP 172.16.164.165, pba=1, use=3
```

This output includes the number of PBAs allocated for the NAT IP and the number of PBAs in use.

9. Verify the number of PBAs assigned to the user IP and the number of PBAs being used:

```
(vdom1) # diagnose firewall ippool list user
User-IP 2001:db8:d0c:1::1, pba=1, use=3
User-IP 2001:db8:d0c:1::1, pba=1, use=3
```

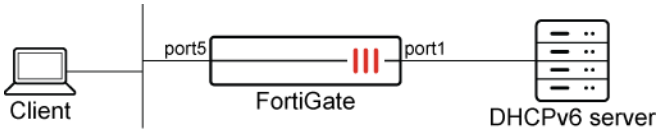
DHCPv6 relay

Similar to DHCPv4, DHCPv6 facilitates communication between networks by relaying queries and responses between a client and a DHCP server on separate networks. The FortiGate device serves as a DHCPv6 relay agent and forwards DHCPv6 messages between clients and servers. The relay agent receives DHCPv6 messages from clients and forwards them to the appropriate DHCPv6 server. In response, the server sends a message containing configuration information for the client, which the relay agent forwards to the client. This enables seamless information exchange between the two networks.

Configure DHCPv6 relay in the CLI using the following syntax:

```
config system interface
    edit <interface>
        config ipv6
            set dhcp6-relay-service {enable|disable}
            set dhcp6-relay-type regular
            set dhcp6-relay-ip <ip6-address>
        end
    next
end
```

In this example, a client connects to a FortiGate device that is configured to function as a DHCPv6 relay. Port1 on the FortiGate device connects to a DHCPv6 server, and port5 is configured as a DHCPv6 relay. The DHCPv6 server has an IP address of 2000:db8:d0c::a. This configuration enables the FortiGate device to facilitate communication between the client and the DHCPv6 server on separate networks.



To configure DHCPv6 relay on the FortiGate:

```
config system interface
    edit port5
        config ipv6
            set dhcp6-relay-service enable
            set dhcp6-relay-type regular
            set dhcp6-relay-ip 2000:db8:d0c::a
        end
    next
end
```

IPv6 tunneling

IPv6 tunneling involves tunneling IPv6 packets from an IPv6 network through an IPv4 network to another IPv6 network. This is different than NAT because once the packet reaches its final destination, the true originating address of the sender is still readable. The IPv6 packets are encapsulated within packets with IPv4 headers that carry their IPv6 payload through the IPv4 network. IPv6 tunneling is suitable in networks that have completely transitioned over to IPv6 but need an internet connection, which is still mostly IPv4 addresses.

Both IPv6 tunneling devices, whether they are a host or a network device, must be dual stack compatible. The tunneling process is as follows:

1. The tunnel entry node creates an encapsulating IPv4 header and transmits the encapsulated packet.
2. The tunnel exit node receives the encapsulated packet.
3. The IPv4 header is removed.
4. The IPv6 header is updated and the IPv6 packet is processed.

There are two types of tunnels in IPv6 tunneling, automatic and configured. Automatic tunnels are configured by using IPv4 address information embedded in an IPv6 address. The IPv6 address of the destination host includes information about which IPv4 address the packet should be tunneled to. Configured tunnels are manually configured, and they are used for IPv6 addresses that do not have any embedded IPv4 information. The IPv6 and IPv4 addresses of the tunnel endpoints must be specified.

Tunnel configurations

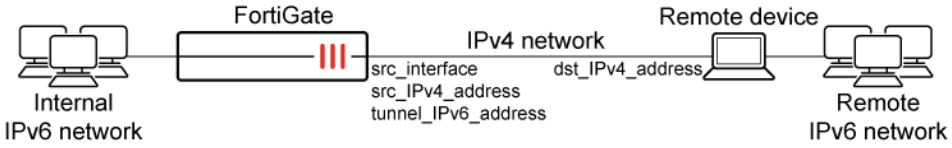
There are four tunneling configurations available depending on which segment of the path between the endpoints of the session the encapsulation takes place.

Type	Description
Network device-to-network device	Dual stack capable devices connected by an IPv4 infrastructure can tunnel IPv6 packets between themselves. The tunnel spans one segment of the path taken by the IPv6 packets.
Host-to-network device	Dual stack capable hosts can tunnel IPv6 packets to an intermediary IPv6 or IPv4 network device that is reachable through an IPv4 infrastructure. The tunnel spans the first segment of the path taken by the IPv6 packets.
Host-to-host	Dual stack capable hosts that are interconnected by an IPv4 infrastructure can tunnel IPv6 packets between themselves. The tunnel spans the entire path taken by the IPv6 packets.
Network device-to-host	Dual stack capable network devices can tunnel IPv6 packets to their final destination IPv6 or IPv4 host. The tunnel spans only the last segment of the path taken by the IPv6 packets.

Regardless of whether the tunnel starts at a host or a network device, the node that does the encapsulation needs to maintain soft state information, such as the maximum transmission unit (MTU), about each tunnel in order to process the IPv6 packets.

6in4 tunnel

The following tunnel configuration tunnels IPv6 traffic over an IPv4 network. An internal IPv6 interface can be configured under `config system interface`.



To configure an IPv6 tunnel over IPv4:

```
config system sit-tunnel
    edit <name>
        set source <src_IPv4_address>
        set destination <dst_IPv4_address>
        set interface <src_interface>
        set ip6 <tunnel_IPv6_address>
    next
end
```

4in6 tunnel

Conversely, the following tunnel configuration tunnels IPv4 traffic over an IPv6 network.

To configure an IPv4 tunnel over IPv6:

```
config system ipv6-tunnel
    edit <name>
        set source <src_IPv6_address>
        set destination <dst_IPv6_address>
        set interface <src_interface>
    next
end
```

The preceding configurations are not available in transparent mode.

This section includes:

- IPv6 IPsec VPN on page 925
- IPv6 GRE tunnels on page 926
- IPv6 tunnel inherits MTU based on physical interface on page 927
- Configuring IPv4 over IPv6 DS-Lite service on page 930

IPv6 IPsec VPN

This topic describes how to configure the IPv6 IPsec VPN feature on your FortiGate device.

You can configure IPv6 using the CLI. To configure IPv6 using GUI, ensure IPv6 is enabled by going to *System > Feature Visibility* and enabling IPv6.

Overview

FortiOS supports route-based IPv6 IPsec, but not policy-based. This section describes different ways IPv6 IPsec can be used:

IPv4 over IPv6	The VPN gateways have IPv6 addresses. The protected networks have IPv4 addresses. The phase 2 configurations at either end use IPv4 selectors. See Site-to-site IPv4 over IPv6 VPN example on page 956 for sample configuration.
IPv6 over IPv4	The VPN gateways have IPv4 addresses. The protected networks use IPv6 addresses. The phase 2 configurations at either end use IPv6 selectors. See Site-to-site IPv6 over IPv4 VPN example on page 964 for sample configuration.
IPv6 over IPv6	Both the VPN gateways and the protected networks use IPv6 addresses. The phase 2 configurations at either end use IPv6 selectors. See Site-to-site IPv6 over IPv6 VPN example on page 947 for sample configuration.

Configuring IPv6 IPsec VPNs

Configuration of an IPv6 IPsec VPN follows the same sequence as for an IPv4 route-based VPN:

Phase 1 and Phase 2 settings	The configuration is the same as for an IPv4 route-based VPN, except that `ip-version` is set to `6` and the `remote-gw6` keyword is used to specify an IPv6 remote gateway address. See Phase 1 configuration on page 2761 and Phase 2 configuration on page 2795 for more information.
Security policies	To complete the VPN configuration, you need a security policy in each direction to permit traffic between the protected network's port and the IPsec interface. You need IPv6 policies unless the VPN is IPv4 over IPv6. See VPN security policies on page 2840 for more information.
Routing	Appropriate routing is needed for both the IPsec packets and the encapsulated traffic within them: • You need a route, which could be the default route, to the remote VPN gateway via the appropriate interface. • You need a route to the remote protected network via the IPsec interface. • You need a blackhole route to the remote protected network to ensure that IPsec traffic doesn't match the default route when the IPsec tunnel is down. Routing is dependent on the method: • **IPv4 over IPv6**: The route to the remote VPN gateway is an IPv6 route. The route to the remote protected network is an IPv4 route. • **IPv6 over IPv4**: The route to the remote VPN gateways is an IPv4 route. The route to the remote protected network is an IPv6 route. • **IPv6 over IPv6**: Routes to both the remote VPN gateway and the remote protected network are IPv6 routes. You can create a new IPv6 static route from *Network > Static Routes*.

You can configure Phase 1 and Phase 2 settings from *VPN > IPsec Wizard*.

To configure Phase 1 and phase 2 settings:

1. Go to *VPN > IPsec Wizard*.
2. Enter a name and set *Template type* to *Custom*.
3. Click *Next*.
4. Under *Network*, set *IP Version* to *IPv6* .
5. Configure the rest of phase 1 and phase 2 settings as required and click *OK*.

IPv6 GRE tunnels

IPv6 addresses can be used at both ends of a GRE tunnel in the same way as with IPv4. See GRE over IPsec on page 2911 for more information.

The configuration is similar to a tunnel for IPv4. However, when you configure the specific tunnel, you need to set the `ip-version` option to `6`. This will enable IPv6-specific options for the tunnel.

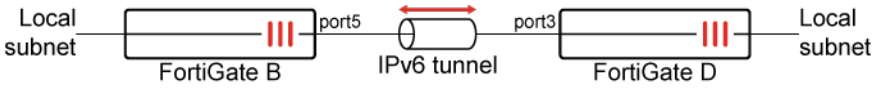
To configure a GRE tunnel:

```
config system gre-tunnel
    edit <name of tunnel>
        set ip-version 6
        set remote-gw6 <IPv6 address of the remote gateway>
        set local-gw6 <IPv6 address of the local gateway>
    next
end
```

IPv6 tunnel inherits MTU based on physical interface

The MTU of an IPv6 tunnel interface is calculated from the MTU of its parent interface minus headers.

Example



In this topology, FortiGate B and FortiGate D are connected over an IPv6 network. An IPv6 tunnel is formed, and IPv4 can be used over the IPv6 tunnel. The tunnel interface MTU is based on the physical interface MTU minus the IP and TCP headers (40 bytes). On FortiGate B's physical interface port5, the MTU is set to 1320. The IPv6 tunnel is based on port5, and its MTU value of 1280 is automatically calculated from the MTU value of its physical interface minus the header. The same is true for port3 on FortiGate D.

To verify the MTU for the IPv6 tunnel on FortiGate B:

1. Configure port5:

```
config system interface
    edit "port5"
        set vdom "root"
        set type physical
        set snmp-index 7
        config ipv6
            set ip6-address 2000:172:16:202::1/64
            set ip6-allowaccess ping
        end
        set mtu-override enable
        set mtu 1320
    next
end
```

2. Configure the IPv6 tunnel:

```
config system ipv6-tunnel
    edit "B_2_D"
        set source 2000:172:16:202::1
        set destination 2000:172:16:202::2
```

```
        set interface "port5"
    next
end
```

3. Configure the tunnel interface:

```
config system interface
    edit "B_2_D"
        set vdom "root"
        set ip 172.16.210.1 255.255.255.255
        set allowaccess ping https http
        set type tunnel
        set remote-ip 172.16.210.2 255.255.255.255
        set snmp-index 33
        config ipv6
            set ip6-address 2000:172:16:210::1/64
            set ip6-allowaccess ping
            config ip6-extra-addr
                edit fe80::2222/10
                next
            end
        end
        set interface "port5"
    next
end
```

4. Verify the interface lists:

```
# diagnose netlink interface list port5
if=port5 family=00 type=1 index=13 mtu=1320 link=0 master=0
ref=68 state=start present fw_flags=0 flags=up broadcast run multicast
Qdisc=mq hw_addr=**:**:**:**:**:** broadcast_addr=**:**:**:**:**:**
stat: rxp=1577 txp=1744 rxb=188890 txb=203948 rxe=0 txe=0 rxd=0 txd=0 mc=825 collision=0 @
time=1631647112
re: rxl=0 rxo=0 rxc=0 rxf=0 rxfi=0 rxm=0
te: txa=0 txc=0 txfi=0 txh=0 txw=0
misc rxc=0 txc=0
input_type=0 state=3 arp_entry=0 refcnt=68
```

```
# diagnose netlink interface list B_2_D
if=B_2_D family=00 type=769 index=41 mtu=1280 link=0 master=0
ref=25 state=start present fw_flags=0 flags=up p2p run noarp multicast
Qdisc=noqueue local=0.0.0.0 remote=0.0.0.0
stat: rxp=407 txp=417 rxb=66348 txb=65864 rxe=0 txe=61 rxd=0 txd=0 mc=0 collision=60 @
time=1631647126
re: rxl=0 rxo=0 rxc=0 rxf=0 rxfi=0 rxm=0
te: txa=0 txc=0 txfi=0 txh=0 txw=0
misc rxc=0 txc=0
input_type=0 state=3 arp_entry=0 refcnt=25
```

To verify the MTU for the IPv6 tunnel on FortiGate D:

1. Configure port3:

```
config system interface
    edit "port3"
        set vdom "root"
        set type physical
        set snmp-index 5
        config ipv6
            set ip6-address 2000:172:16:202::2/64
            set ip6-allowaccess ping
        end
        set mtu-override enable
        set mtu 1320
    next
end
```

2. Configure the IPv6 tunnel:

```
config system ipv6-tunnel
    edit "D_2_B"
        set source 2000:172:16:202::2
        set destination 2000:172:16:202::1
        set interface "port3"
    next
end
```

3. Configure the tunnel interface:

```
config system interface
    edit "D_2_B"
        set vdom "root"
        set ip 172.16.210.2 255.255.255.255
        set allowaccess ping https http
        set type tunnel
        set remote-ip 172.16.210.1 255.255.255.255
        set snmp-index 36
        config ipv6
            set ip6-address 2000:172:16:210::2/64
            set ip6-allowaccess ping
            config ip6-extra-addr
                edit fe80::4424/10
                next
            end
        end
        set interface "port3"
    next
end
```

4. Verify the interface lists:

```
# diagnose netlink interface list port3
```

```
# diagnose netlink interface list D_2_B
```

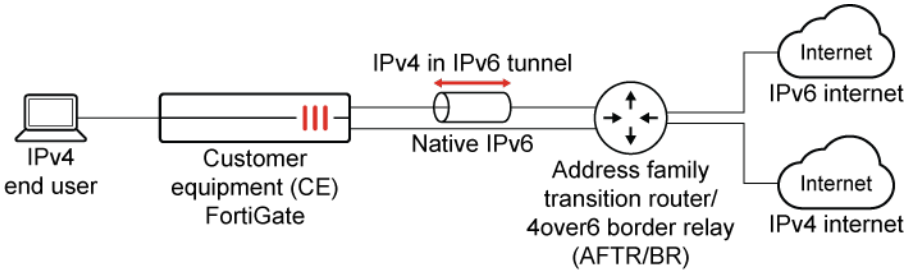
Configuring IPv4 over IPv6 DS-Lite service

IPv4 over IPv6 DS-Lite service can be configured on a virtual network enabler (VNE) tunnel. In addition, VNE tunnel fixed IP mode supports username and password authentication.

```
config system vne-tunnel
    set status enable
    set mode {map-e | fixed-ip | ds-lite}
    set ipv4-address <IPv4_address>
    set br <IPv6_address or FQDN>
    set http-username <string>
    set http-password <password>
end
```

`mode {map-e | fixed-ip | ds-lite}`	Set the VNE tunnel mode: • map-e: MAP-E • fixed-ip: fixed IP • ds-lite: DS-Lite
`ipv4-address <IPv4_address>`	Enter the tunnel IPv4 address and netmask. This setting is optional.
`br <IPv6_address or FQDN>`	Enter the IPv6 or FQDN of the border relay.
`http-username <string>`	Enter the HTTP authentication user name.
`http-password <password>`	Enter the HTTP authentication password.

DS-Lite allows applications using IPv4 to access the internet with IPv6. DS-Lite is supported by internet providers that do not have enough public IPv4 addresses for their customers, so DS-Lite is used for IPv6 internet connections. When a DS-Lite internet connections is used, the FortiGate encapsulates all data from IPv4 applications into IPv6 packets. The packets are then transmitted to the internet service provider using the IPv6 connection. Next, a dedicated server unpacks the IPv6 packets and forwards the IPv4 data to the actual destination on the internet.



DS-Lite example

In this example, DS-Lite VNE tunnel mode is used between the FortiGate and the BR.

To configure a DS-Lite tunnel between the FortiGate and the BR:

1. Configure the IPv6 interface:

```
config system interface
    edit "wan1"
        set vdom "root"
        set mode dhcp
        set allowaccess ping fgfm
        set type physical
        set role wan
        set snmp-index 1
        config ipv6
            set ip6-allowaccess ping
            set dhcp6-information-request enable
            set autoconf enable
            set unique-autoconf-addr enable
        end
    next
end
```

2. Configure the VNE tunnel:

```
config system vne-tunnel
    set status enable
    set interface "wan1"
    set ssl-certificate "Fortinet_Factory"
    set auto-asic-offload enable
    set ipv4-address 192.168.1.99 255.255.255.255
    set br "dgw.xxxxx.jp"
    set mode ds-lite
end
```

3. View the wan1 IPv6 configuration details:

```
config system interface
    edit "wan1"
        config ipv6
            get
                ip6-mode            : static
                nd-mode             : basic
                ip6-address         : 2001:f70:2880:xxxx:xxxx:xxxx:fe39:ccd2/64
                ip6-allowaccess     : ping
                icmp6-send-redirect : enable
                ra-send-mtu         : enable
                ip6-reachable-time  : 0
                ip6-retrans-time    : 0
                ip6-hop-limit       : 0
                dhcp6-information-request: enable
                cli-conn6-status    : 1
                vrrp-virtual-mac6   : disable
                vrip6_link_local    : ::
```

```
                ip6-dns-server-override: enable
                Acquired DNS1       : 2001:f70:2880:xxxx:xxxx:xxxx:fe40:9082
                Acquired DNS2       : ::
                ip6-extra-addr:
                ip6-send-adv        : disable
                autoconf            : enable
                prefix      : 2001:f70:2880:xxxx::/64
                preferred-life-time         : 942735360
                valid-life-time     : 1077411840
                unique-autoconf-addr: enable
                interface-identifier: ::
                dhcp6-relay-service : disable
        end
    next
end
```

4. Verify the IPv6 address list:

```
# diagnose ipv6 address list
dev=5 devname=wan1 flag= scope=0 prefix=64 addr=2001:f70:2880:xxxx:xxxx:xxxx:fe39:ccd2
preferred=11525 valid=13325 cstamp=6520 tstamp=6892
dev=5 devname=wan1 flag=P scope=253 prefix=64 addr=fe80::xxxx:xxxx:fe39:ccd2
preferred=4294967295 valid=4294967295 cstamp=6373 tstamp=6373
dev=18 devname=root flag=P scope=254 prefix=128 addr=::1 preferred=4294967295 valid=4294967295
cstamp=3531 tstamp=3531
dev=25 devname=vsys_ha flag=P scope=254 prefix=128 addr=::1 preferred=4294967295
valid=4294967295 cstamp=5604 tstamp=5604
dev=27 devname=vsys_fgfm flag=P scope=254 prefix=128 addr=::1 preferred=4294967295
valid=4294967295 cstamp=6377 tstamp=6377
```

5. Test the tunnel connection by pinging the Google public DNS IPv6 address:

```
# execute ping6 2001:4860:4860::8888
PING 2001:4860:4860::8888(2001:4860:4860::8888) 56 data bytes
64 bytes from 2001:4860:4860::8888: icmp_seq=1 ttl=114 time=6.89 ms
64 bytes from 2001:4860:4860::8888: icmp_seq=2 ttl=114 time=3.39 ms
64 bytes from 2001:4860:4860::8888: icmp_seq=3 ttl=114 time=3.46 ms
64 bytes from 2001:4860:4860::8888: icmp_seq=4 ttl=114 time=3.34 ms
64 bytes from 2001:4860:4860::8888: icmp_seq=5 ttl=114 time=3.39 ms
--- 2001:4860:4860::8888 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss, time 4079ms
rtt min/avg/max/mdev = 3.340/4.097/6.895/1.400 ms
```

Fixed IP mode example

In this example, fixed IP VNE tunnel mode with HTTP authentication is used between the FortiGate and the BR.

To configure a fixed IP mode with HTTP authentication between the FortiGate and the BR:

1. Configure the IPv6 interface:

```
config system interface
    edit "wan1"
        set vdom "root"
        set mode dhcp
        set allowaccess ping fgfm
        set type physical
        set role wan
        set snmp-index 1
        config ipv6
            set ip6-allowaccess ping
            set dhcp6-information-request enable
            set autoconf enable
        end
    next
end
```

2. Configure the VNE tunnel:

```
config system vne-tunnel
    set status enable
    set interface "wan1"
    set ipv4-address 120.51.xxx.xxx1 255.255.255.255
    set br "2001:f60:xxxx:xxxx::1"
    set update-url "https://ddnsweb1.ddns.xxxxxx.jp/cgi-bin/ddns_
api.cgi?d=xxxxxx.v4v6.xxxxx.jp&p=**********&a=[IP6]&u=xxxxxx.v4v6.xxxxx.jp"
    set mode fixed-ip
    set http-username "laptop-1"
    set http-password **********
end
```

3. Verify the wan1 IPv6 configuration details:

```
config system interface
    edit "wan1"
        config ipv6
            get
                ....
```

4. Verify the VNE daemon:

```
# diagnose test application vned 1
------------------------------------------------------------------------------
vdom: root/0, is master, devname=wan1 link=0 tun=vne.root mode=fixed-ip ssl_cert=Fortinet_
Factory
end user ipv6 perfix: 2001:f70:2880:xxxx::/64
interface ipv6 addr: 2001:f70:2880:xxxx:xxxx:xxxx:fe39:ccd2
config ipv4 perfix: 120.51.xxx.xxx1/255.255.255.255
config br: 2001:f60:xxxx:xxxx::1
```

```
HTTP username: laptop-1
update url: https://ddnsweb1.ddns.xxxxxx.jp/cgi-bin/ddns_
api.cgi?d=xxxxxx.v4v6.xxxxx.jp&p=**********&a=[IP6]&u=xxxxxx.v4v6.xxxxx.jp
host: ddnsweb1.ddns.xxxxxx.jp path: /cgi-bin/ddns_
api.cgi?d=xxxxxx.v4v6.xxxxx.jp&p=**********&a=[IP6]&u=xxxxxx.v4v6.xxxxx.jp port:443 ssl: 1
tunnel br: 2001:f60:xxxx:xxxx::1
tunnel ipv6 addr: 2001:f70:2880:xxxx:xxxx:xxxx:fe39:ccd2
tunnel ipv4 addr: 120.51.xxx.xxx1/255.255.255.255
update result: <H1>DDNS API</H1><HR><H2>* Query parameter check :
OK</H2>FQDN=xxxxxx.v4v6.xxxxx.jp<BR>Password=**********<BR>IPv6=2001:f70:2880:xxxx:xxxx:xxxx:f
e39:ccd2<BR>UID=xxxxxx.v4v6.xxxxx.jp<BR>Address=2001:f70:2880:xxxx:xxxx:xxxx:fe39:ccd2<BR><H
2>* routerinfo check : OK</H2><H2>* records check : OK</H2><H2>* routerinfo update :
OK</H2><H2>* records update : OK</H2><H2>* DDNS API update : Success [2022-01-18 18:37:58
1642498678]</H2>
Fixed IP rule client: state=succeed retries=0 interval=0 expiry=0 reply_code=0
fqdn=2001:f60:xxxx:xxxx::1 num=1 cur=0 ttl=4294967295 expiry=0
2001:f60:xxxx:xxxx::1
Fixed IP DDNS client: state=succeed retries=0 interval=10 expiry=0 reply_code=200
fqdn=ddnsweb1.ddns.xxxxxx.jp num=1 cur=0 ttl=6 expiry=0
2001:f61:0:2a::18
```

5. Test the tunnel connection by pinging the Google public DNS IPv4 and IPv6 addresses:

```
# execute ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: icmp_seq=0 ttl=119 time=3.7 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=119 time=3.6 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=119 time=3.6 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=119 time=3.6 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=119 time=3.5 ms
--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 3.5/3.6/3.7 ms
```

```
# execute ping6 2001:4860:4860::8888
PING 2001:4860:4860::8888(2001:4860:4860::8888) 56 data bytes
64 bytes from 2001:4860:4860::8888: icmp_seq=1 ttl=114 time=6.99 ms
64 bytes from 2001:4860:4860::8888: icmp_seq=2 ttl=114 time=3.61 ms
64 bytes from 2001:4860:4860::8888: icmp_seq=3 ttl=114 time=3.34 ms
64 bytes from 2001:4860:4860::8888: icmp_seq=4 ttl=114 time=3.27 ms
64 bytes from 2001:4860:4860::8888: icmp_seq=5 ttl=114 time=3.75 ms
--- 2001:4860:4860::8888 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss, time 4039ms
rtt min/avg/max/mdev = 3.276/4.195/6.992/1.409 ms
```

IPv6 Simple Network Management Protocol

Simple Network Management Protocol (SNMP) in IPv6 is similar to IPv4, with the main difference being the address format. Despite this, SNMP's principles and functionalities, including network management, device monitoring, and performance information gathering, remain consistent across both versions. See SNMP on page 3877 for more information.

SNMP for monitoring interface status example

In this example, SNMP manager (2001:db8:d0c:2::1) is configured to receive notifications when a FortiGate port either goes down or is brought up. Additionally, the SNMP manager has the capability to query the current status of the FortiGate port.

Please note that the IPv6 addresses used in this example are for illustrative purposes only and should not be used in your environment.

The 2001:db8::/32 prefix is a special IPv6 prefix designated for use in documentation examples. See RFC 3849 for more information.

To configure SNMP for monitoring interface status in the GUI:

1. Configure the Interface access:
   a. Go to *Network* > *Interfaces* and edit *port1*.
   b. In the *Administrative Access* options, enable *SNMP* under *IPv6*.
   c. Click *OK*.
2. Configure the SNMP agent:
   a. Go to *System* > *SNMP*.
   b. Enable *SNMP Agent*.
   c. Configure the following fields:

Description	Branch
Location	Burnaby
Contact Info	Jane Doe

   d. Click *Apply*.
3. Configure an SNMP v3 user:
   a. Go to *System* > *SNMP*.
   b. In the *SNMP v3* table, click *Create New*.
   c. Configure the following fields:

User Name	Interface_Status
Security Level	Authentication
Authentication Algorithm	SHA1

Password	*******
IPv6 Hosts > IP Address	2001:db8:d0c:2::1

d. Click *OK*.

e. Click *Apply*.

To configure SNMP for monitoring interface status in the CLI:

1. Configure the Interface access:

```
config system interface
    edit port1
        config ipv6
            append ip6-allowaccess snmp
        end
    next
end
```

2. Configure the SNMP agent:

```
config system snmp sysinfo
    set status enable
    set description Branch
    set contact-info "Jane Doe"
    set location Burnaby
end
```

3. Configure an SNMP v3 user:

```
config system snmp user
    edit "Interface_Status"
        set notify-hosts6 2001:db8:d0c:2::1
        set security-level auth-no-priv
        set auth-proto sha
        set auth-pwd ********************
    next
end
```

Verification

To verify the SNMP configuration:

1. Start the packet capture on interface port1 with the filter set to port 162. See Using the packet capture tool on page 1025 for more information.
2. Turn off one of the FortiGate interface statuses to down; in this case, port2.
3. Save the packet capture.

```
> Internet Protocol Version 6, Src: 2001:db8:d0c:2::f, Dst: 2001:db8:d0c:2::1
> User Datagram Protocol, Src Port: 162, Dst Port: 162
v Simple Network Management Protocol
    msgVersion: snmpv3 (3)
  > msgGlobalData
  > msgAuthoritativeEngineID: 800030440580003044040 85b0e9f05f0
    msgAuthoritativeEngineBoots: 1695743618
    msgAuthoritativeEngineTime: 189193
    msgUserName: Interface_Status
    msgAuthenticationParameters: 064b1a16a3db6b06f4d66d86
    msgPrivacyParameters: <MISSING>
  v msgData: plaintext (0)
    v plaintext
      > contextEngineID: 800030440580003044040 85b0e9f05f0
        contextName:
      v data: snmpV2-trap (7)
        v snmpV2-trap
            request-id: 753
            error-status: noError (0)
            error-index: 0
          v variable-bindings: 9 items
            > 1.3.6.1.2.1.1.3.0: 18925675
            > 1.3.6.1.6.3.1.1.4.1.0: 1.3.6.1.6.3.1.1.5.3 (iso.3.6.1.6.3.1.1.5.3)
            > 1.3.6.1.2.1.2.2.1.1.2: 2
            > 1.3.6.1.2.1.2.2.1.7.2: 1
            > 1.3.6.1.2.1.2.2.1.8.2: 2
            > 1.3.6.1.4.1.12356.100.1.1.1.0: "FGVM08TM22004645"
            > 1.3.6.1.2.1.1.5.0: "Root"
            > 1.3.6.1.2.1.31.1.1.1.1.2: "port2"
            > 1.3.6.1.2.1.2.2.1.2.2: <MISSING>
```

The SNMP v3 trap is observed to be transmitted from port1 to the SNMP manager. It's also noteworthy that the *msgAuthenticationParameters* are configured, signifying that authentication is active. However, the absence of *msgPrivacyParameters* suggests that encryption is not in place, a fact further corroborated by the plaintext nature of the *msgData*.

4. Verify that the SNMP manager has received the trap. See Important SNMP traps on page 3884 for an example of a trap.
5. Verify that the SNMP manager can successfully query and receive a response on the current status of the FortiGate ports.

```
# snmpwalk -v3 -u Interface_Status  -l authNoPriv -a SHA -A xxxxxxxx udp6:2001:db8:d0c:2::f
1.3.6.1.2.1.2.2.1.8
iso.3.6.1.2.1.2.2.1.8.1 = INTEGER: 1
iso.3.6.1.2.1.2.2.1.8.2 = INTEGER: 2
iso.3.6.1.2.1.2.2.1.8.3 = INTEGER: 1
iso.3.6.1.2.1.2.2.1.8.4 = INTEGER: 1
iso.3.6.1.2.1.2.2.1.8.5 = INTEGER: 1
iso.3.6.1.2.1.2.2.1.8.6 = INTEGER: 1
iso.3.6.1.2.1.2.2.1.8.7 = INTEGER: 1
iso.3.6.1.2.1.2.2.1.8.8 = INTEGER: 1
iso.3.6.1.2.1.2.2.1.8.9 = INTEGER: 1
iso.3.6.1.2.1.2.2.1.8.10 = INTEGER: 1
```

Dynamic routing in IPv6

The principles that govern dynamic routing in IPv6 are fundamentally the same as those in IPv4. However, it's crucial to understand that while IPv6 operates similarly to IPv4 in terms of routing, it utilizes a distinct routing table and process.

When a router receives a packet, the routing scheme of the packet determines how the traffic will be routed. If the packet is using the IPv4 scheme, the router will refer to the IPv4 routing table to determine the best path for the packet. This table contains all the necessary information about network paths within the IPv4 framework. See Dynamic routing on page 605 for more information.

Conversely, if the router receives a packet with an IPv6 address, it will consult the IPv6 routing table. This separate table contains all the relevant information for routing within the IPv6 framework.

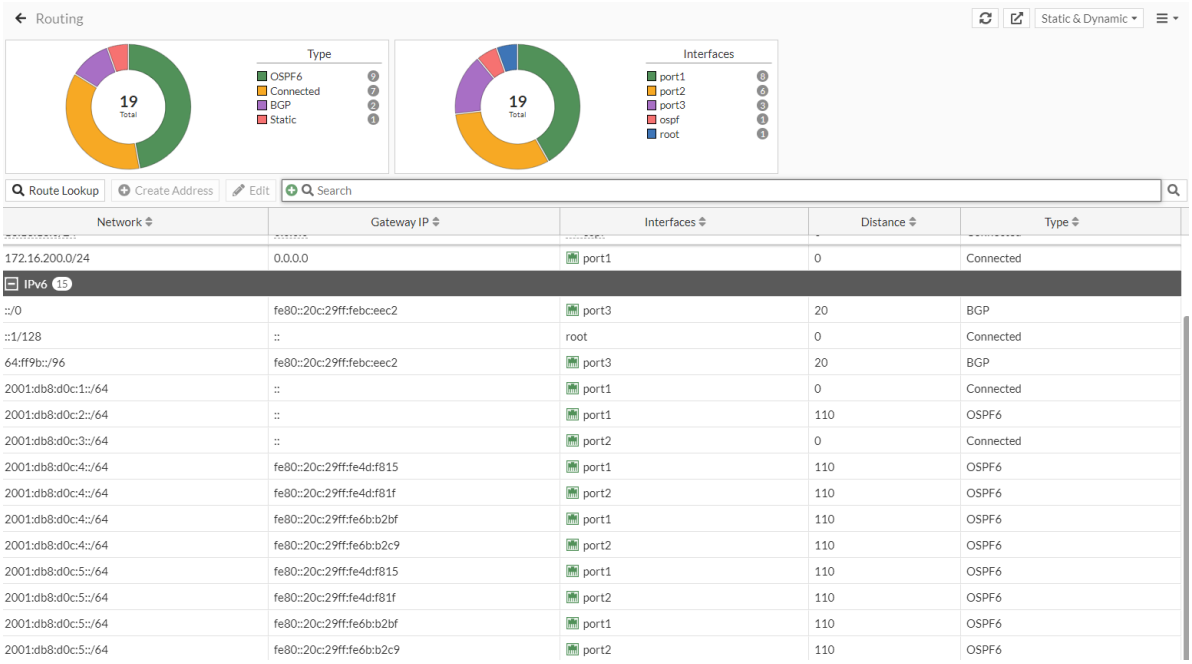
In essence, while both IPv4 and IPv6 use similar methods for routing packets, they each have their own dedicated processes and tables to ensure efficient and accurate routing.

Go to *System > Feature Visibility* and enable *Advanced Routing* to configure dynamic routing options in the GUI. See Feature visibility on page 3952 for more information.

This section includes:

- OSPFv3 and IPv6 on page 939
- BGP and IPv6 on page 939

To view the routing table and perform route look-ups in the GUI, go to *Dashboard > Network Monitor > Routing*.



To view the routing table in the CLI:

```
# get router info6 routing-table
IPv6 Routing Table
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, B - BGP, V - BGP VPNv6
       * - candidate default
```

```
Timers: Uptime

Routing table for VRF=0
B*      ::/0 [20/0] via fe80::20c:29ff:febc:eec2, port3, 02:45:56, [1024/0]
C       ::1/128 via ::, root, 03:45:04
B       64:ff9b::/96 [20/0] via fe80::20c:29ff:febc:eec2, port3, 02:45:56, [1024/0]
C       2001:db8:d0c:1::/64 via ::, port1, 00:33:21
O       2001:db8:d0c:2::/64 [110/2] via fe80::20c:29ff:fe4d:f81f, port1, 00:33:04, [1024/0]
                            [110/2] via fe80::20c:29ff:fe6b:b2c9, port2, 00:33:04, [1024/0]
C       2001:db8:d0c:3::/64 via ::, port2, 03:45:04
O       2001:db8:d0c:4::/64 [110/2] via fe80::20c:29ff:fe4d:f81f, port1, 00:33:04, [1024/0]
O       2001:db8:d0c:5::/64 [110/2] via fe80::20c:29ff:fe6b:b2c9, port2, 02:51:32, [1024/0]
C       2001:db8:d0c:6::/64 via ::, port3, 03:45:04
```

OSPFv3 and IPv6

OSPF version 3 (OSPFv3) includes support for IPv6 and can only be configured via the CLI. Unlike its predecessor, OSPFv2, which uses IPv4, OSPFv3 utilizes IPv6 addresses. However, the area numbers in OSPFv3 still adhere to the 32-bit numbering system of OSPFv2, as described in RFC 2740. Likewise, the router ID and area ID are in the same format as OSPFv2. See OSPF on page 628 for more information.

For IPv6, the main difference in OSPFv3 is that rather than using a network statement to enable OSPFv3 on an interface, you define OSPF6 (OSPF for IPv6) interfaces, which are bound to the interface and area. This configuration must be done in the CLI, as follows:

```
config router ospf6
    set router-id <id>
    config area
        edit <id>
        next
    end
    config ospf6-interface
        edit <name>
            set interface <string>
            set area-id <id>
        next
    end
end
```

Note that OSPFv3 neighbors use link-local IPv6 addresses, but with broadcast and point-to-point network types, and neighbors are automatically discovered. You only have to manually configure neighbors when using non-broadcast network types.

See Basic OSPFv3 example on page 973 for a sample configuration.

BGP and IPv6

Border Gateway Protocol (BGP) is a standardized routing protocol that is used to route traffic across the internet. See BGP on page 646 for more information.

FortiGate units support IPv6 over BGP using the same `config router bgp` command as IPv4, but different subcommands.

The main CLI keywords have IPv6 equivalents that are identified by the “6” on the end of the keyword, such as with `config network6` or `set allowas-in6`.

See config router bgp for a comprehensive list of commands.

See Basic IPv6 BGP example on page 981 for a sample configuration.

IPv6 configuration examples

The following topics provide instructions on different IPv6 configuration examples:

- IPv6 quick start example on page 940
- Site-to-site IPv6 over IPv6 VPN example on page 947
- Site-to-site IPv4 over IPv6 VPN example on page 956
- Site-to-site IPv6 over IPv4 VPN example on page 964
- Basic OSPFv3 example on page 973
- Basic IPv6 BGP example on page 981
- NPTv6 protocol for IPv6 address translation example on page 984
- Enhancing SIP reliability in 464XLAT environments on page 988

IPv6 quick start example

In this example, a host belonging to a specific range on the internal IPv6 network can communicate exclusively with the web server and FTP server.

Additionally, all internal clients can access the Internet.

Prerequisites

Before you begin to configure IPv6, go through the following steps:

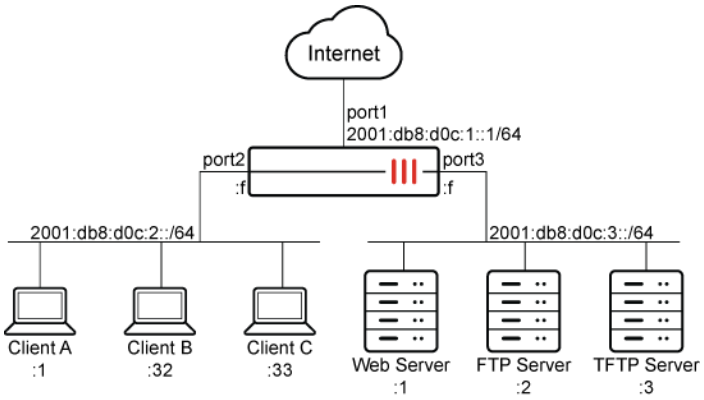
1. Obtain an IPv6 /48 global routing prefix, commonly known as a site prefix. To procure a 48-bit IPv6 site prefix for your organization simply liaise with your ISP.
2. Design a subnetting plan for your organization's IPv6 network using a 16-bit subnet ID, allowing for up to 65 535 subnets. The specific scheme will depend on the network's size, structure, and the organization's needs.

At this stage, the following installation and configuration conditions are assumed:

- You have administrative access to the GUI or CLI.
- The FortiGate unit is incorporated into your WAN or other networks, but for simplicity, only the standalone ForiGate configuration is displayed.

Topology

The following topology is used for this example:



- The company is assigned the site prefix of 2001:db8:d0c::/48 by their ISP.
- The IPv6 address for the Web Server is 2001:db8:d0c:3::1/64.
- The IPv6 address for the FTP Server is 2001:db8:d0c:3::2/64.
- The IPv6 address for the TFTP Server is 2001:db8:d0c:3::3/64.
- The range on the internal IPv6 network that can access both servers is from 2001:db8:d0c:2::1 to 2001:db8:d0c:2::32.
- The IPv6 address of port1 is 2001:db8:d0c:1::1/64.
- The IPv6 address of port2 is 2001:db8:d0c:2::f/64.
- The IPv6 address of port3 is 2001:db8:d0c:3::f/64.
- The IPv6 address of the default gateway is 2001:db8:d0c:1::f/64.

> ⚠ Please note that the IPv6 addresses used in this example are for illustrative purposes only and should not be used in your environment.
>
> The 2001:db8::/32 prefix is a special IPv6 prefix designated for use in documentation examples. See RFC 3849 for more information.

To configure the example in the GUI:

1. Configure the IPv6 address on port1, port2 and port3:
   a. Go to *Network* > *Interfaces* and edit port1.
   b. For *IPv6 addressing Mode*, select manual and enter the *IPv6 Address/Prefix*.

IPv6 Address/Prefix	2001:db8:d0c:1::1/64

   c. Click *OK*.
   d. Repeat steps a and b for port2.

IPv6 Address/Prefix	2001:db8:d0c:2::f/64

   e. Repeat steps a and b for port3.

IPv6 Address/Prefix	2001:db8:d0c:3::f/64

2. Configure the default route:

a. Go to *Network* > *Static Routes*.
b. Click *Create New* > *IPv6 Static Route*.
c. Configure the following settings:

Destination	::/0
Gateway Address	2001:db8:d0c:1::f
Interface	port1

d. Select *OK*.

3. Configure the IPv6 firewall address for the Web Server:
a. Go to *Policy & Objects* > *Addresses* and select *IPv6 Address*.
b. Select *Create new*.
c. Fill out the fields with the following information:

Name	Web_Server
Type	IPv6 Subnet
IPv6 Address	2001:db8:d0c:3::1/128

d. Select *OK*.

4. Configure the IPv6 firewall address for the FTP Server:
a. Go to *Policy & Objects* > *Addresses* and select *IPv6 Address*.
b. Select *Create new*.
c. Fill out the fields with the following information:

Name	FTP_Server
Type	IPv6 Subnet
IPv6 Address	2001:db8:d0c:3::2/128

d. Select *OK*.

5. Configure the IPv6 address group, which includes both the Web and FTP servers:
a. Go to *Policy & Objects* > *Addresses* and select *IPv6 Address Group*.
b. Fill out the fields with the following information:

Group name	Custom_Server
Members	Web_Server, FTP_Server

c. Select *OK*.

6. Configure the IPv6 firewall address for the Internal IPv6 network range which can access both the Web and FTP server:
a. Go to *Policy & Objects* > *Addresses* and select *IPv6 Address*.
b. Select *Create new*.
c. Fill out the fields with the following information:

Name	Internal_Custom_Range

Type	IPv6 Range
IP Range	2001:db8:d0c:2::1 - 2001:db8:d0c:2::32

d. Select *OK*.

7. Configure the IPv6 firewall policy to allow IPv6 traffic from *Internal_Custom_Range* to *Custom_Server*:
   a. Go to *Policy & Objects > Firewall Policy*.
   b. Click *Create New*.
   c. Name the policy and configure the following parameters:

Incoming Interface	port2
Outgoing Interface	port3
Source	Internal_Custom_Range
Destination	Custom_Server
Schedule	always
Service	FTP, HTTPS
Action	ACCEPT

   d. Click *OK*.
8. Configure the IPv6 firewall policy to allow IPv6 traffic from internal clients to the Internet:
   a. Go to *Policy & Objects > Firewall Policy*.
   b. Click *Create New*.
   c. Name the policy and configure the following parameters:

Incoming Interface	port2
Outgoing Interface	port1
Source	all
Destination	all
Schedule	always
Service	ALL
Action	ACCEPT

   d. Click *OK*.

To configure the example in the CLI:

1. Configure the IPv6 address on port1, port2, and port3:

```
config system interface
    edit "port1"
        config ipv6
            set ip6-address 2001:db8:d0c:1::1/64
        end
```

```
    next
    edit "port2"
        config ipv6
            set ip6-address 2001:db8:d0c:2::f/64
        end
    next
    edit "port3"
        config ipv6
            set ip6-address 2001:db8:d0c:3::f/64
        end
    next
end
```

2. Configure the default route:

```
config router static6
    edit 0
        set gateway 2001:db8:d0c:1::f
        set device "port1"
    next
end
```

3. Configure the IPv6 firewall address for the Web Server:

```
config firewall address6
    edit "Web_Server"
        set ip6 2001:db8:d0c:3::1/128
    next
end
```

4. Configure the IPv6 firewall address for the FTP Server:

```
config firewall address6
    edit "FTP_Server"
        set ip6 2001:db8:d0c:3::2/128
    next
end
```

5. Configure the IPv6 address group, which includes for the Web and FTP Servers:

```
config firewall addrgrp6
    edit "Custom_Server"
        set member "FTP_Server" "Web_Server"
    next
end
```

6. Configure the IPv6 firewall address for the Internal IPv6 network range which can access both the Web and FTP Server:

```
config firewall address6
    edit "Internal_Custom_Range"
        set type iprange
        set start-ip 2001:db8:d0c:2::1
```

```
        set end-ip 2001:db8:d0c:2::32
    next
end
```

7. Configure the IPv6 firewall policy to allow IPv6 traffic from *Internal_Custom_Range* to *Custom_Server*:

```
config firewall policy
    edit 1
        set name "IPv6_internal_to_server"
        set srcintf "port2"
        set dstintf "port3"
        set action accept
        set srcaddr6 "Internal_Custom_Range"
        set dstaddr6 "Custom_Server"
        set schedule "always"
        set service "FTP" "HTTPS"
        set utm-status enable
        set logtraffic all
    next
end
```

8. Configure the IPv6 firewall policy to allow IPv6 traffic from Internal clients to the Internet:

```
config firewall policy
    edit 1
        set name "IPv6_internal_to_internet"
        set srcintf "port2"
        set dstintf "port1"
        set action accept
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set logtraffic all
    next
end
```

Verification

The following commands can be used to verify that IPv6 traffic is entering and leaving the FortiGate as expected. See Debugging the packet flow on page 4968 for more information.

```
diagnose debug enable
diagnose debug flow trace start6 200
```

The output below indicates that hosts belonging to the *Internal_Custom_Range* can successfully reach both the *Web_Server* and *FTP_Server* defined in the *Custom_Server* address group.

However, they are unable to reach the TFTP server, as it is not included in the *Custom_Server* group. Furthermore, hosts with IPv6 addresses that do not belong to the *Internal_Custom_Range* are not able to access *Custom_Server*.

Host belonging to *Internal_Custom_Range* accessing *Web_Server*:

```
id=65308 trace_id=21 func=resolve_ip6_tuple_fast line=4962 msg="vd-root:0 received a packet
(proto=6, 2001:db8:d0c:2::1:55114->2001:db8:d0c:3::1:443) from port2."
id=65308 trace_id=21 func=resolve_ip6_tuple line=5102 msg="allocate a new session-0000006b"
id=65308 trace_id=21 func=ip6_route_input line=2186 msg="find a route: gw-:: via port3 err 0 flags
40000001"
id=65308 trace_id=21 func=fw6_forward_handler line=501 msg="Check policy between port2 -> port3"
id=65308 trace_id=21 func=fw6_forward_handler line=638 msg="Allowed by Policy-1:"
```

Host belonging to *Internal_Custom_Range* accessing *FTP_Server*:

```
id=65308 trace_id=6 func=resolve_ip6_tuple_fast line=4962 msg="vd-root:0 received a packet
(proto=6, 2001:db8:d0c:2::32:50982->2001:db8:d0c:3::2:21) from port2."
id=65308 trace_id=6 func=resolve_ip6_tuple line=5102 msg="allocate a new session-00000053"
id=65308 trace_id=6 func=ip6_route_input line=2186 msg="find a route: gw-:: via port3 err 0 flags
40000001"
id=65308 trace_id=6 func=fw6_forward_handler line=501 msg="Check policy between port2 -> port3"
id=65308 trace_id=6 func=fw6_forward_handler line=638 msg="Allowed by Policy-1:"
```

Host belonging to *Internal_Custom_Range* accessing TFTP Server:

```
id=65308 trace_id=17 func=resolve_ip6_tuple_fast line=4962 msg="vd-root:0 received a packet
(proto=17, 2001:db8:d0c:2::32:65316->2001:db8:d0c:3::3:69) from port2."
id=65308 trace_id=17 func=resolve_ip6_tuple line=5102 msg="allocate a new session-00000055"
id=65308 trace_id=17 func=ip6_route_input line=2186 msg="find a route: gw-:: via port3 err 0 flags
40000001"
id=65308 trace_id=17 func=fw6_forward_handler line=501 msg="Check policy between port2 -> port3"
id=65308 trace_id=17 func=fw6_forward_handler line=530 msg="Denied by forward policy check"
```

Host not belonging to *Internal_Custom_Range* accessing *FTP_Server*:

```
id=65308 trace_id=1 func=resolve_ip6_tuple_fast line=4962 msg="vd-root:0 received a packet
(proto=6, 2001:db8:d0c:2::33:52555->2001:db8:d0c:3::2:21) from port2."
id=65308 trace_id=1 func=resolve_ip6_tuple line=5102 msg="allocate a new session-0000004d"
id=65308 trace_id=1 func=ip6_route_input line=2186 msg="find a route: gw-:: via port3 err 0 flags
40000001"
id=65308 trace_id=1 func=fw6_forward_handler line=501 msg="Check policy between port2 -> port3"
id=65308 trace_id=1 func=fw6_forward_handler line=530 msg="Denied by forward policy check"
```

Internal clients accessing the Internet:

The output below indicates that internal clients can successfully reach the internet.

1. Go to *Log & Report > Forward Traffic*.
2. View the log details in the GUI, or download the log file:

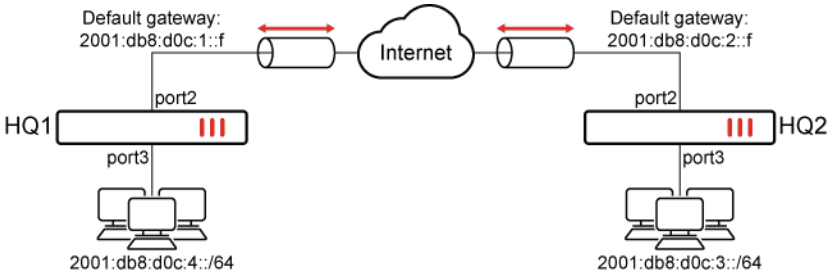
```
1: date=2023-05-10 time=13:22:54 eventtime=1683750174692262952 tz="-0700" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="root" srcip=2001:db8:d0c:2::1
```

```
srcport=64780 srcintf="port2" srcintfrole="undefined" dstip=64:ff9b::83fd:21c8 dstport=443
dstintf="port1" dstintfrole="undefined" sessionid=15723 proto=6 action="close" policyid=2
policytype="policy" poluuid="ea8a972e-d7e9-51ed-9b29-757f04e7194c" policyname="IPv6_internal_
to_internet" srccountry="Reserved" service="HTTPS" trandisp="noop" duration=3 sentbyte=47192
rcvdbyte=13199 sentpkt=49 rcvdpkt=48 appcat="unscanned"
2: date=2023-05-10 time=13:19:47 eventtime=1683749987902192921 tz="-0700" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="root" srcip=2001:db8:d0c:2::33
srcport=51246 srcintf="port2" srcintfrole="undefined" dstip=64:ff9b::349f:31c7 dstport=443
dstintf="port1" dstintfrole="undefined" sessionid=15126 proto=6 action="close" policyid=2
policytype="policy" poluuid="ea8a972e-d7e9-51ed-9b29-757f04e7194c" policyname="IPv6_internal_
to_internet" srccountry="Reserved" service="HTTPS" trandisp="noop" duration=59 sentbyte=5109
rcvdbyte=7726 sentpkt=13 rcvdpkt=11 appcat="unscanned"
```

Site-to-site IPv6 over IPv6 VPN example

In this example, clients on IPv6-addressed networks communicate securely over public IPv6 infrastructure.

The following topology is used for this example:



- Port2 connects to the public network and port3 connects to the local network.
- The IPv6 address for HQ1 port2 and port3 is 2001:db8:d0c:1::e and 2001:db8:d0c:4::e, respectively.
- The IPv6 address for HQ2 port2 and port3 is 2001:db8:d0c:2::e and 2001:db8:d0c:3::e, respectively.

Please note that the IPv6 addresses used in this example are for illustrative purposes only and should not be used in your environment.

The 2001:db8::/32 prefix is a special IPv6 prefix designated for use in documentation examples. See RFC 3849 for more information.

To configure the example in the GUI:

1. Configure the HQ1 FortiGate.
   a. Configure the IPv6 address on port2 and port3:
      i. Go to *Network* > *Interfaces* and edit port2.
      ii. Set *IPv6 addressing mode* to *Manual* and enter the *IPv6 Address/Prefix*.

IPv6 Address/Prefix	2001:db8:d0c:1::e/64

      iii. Click *OK*.
      iv. Repeat these steps for port3.

IPv6 Address/Prefix	2001:db8:d0c:4::e/64

b. Configure IPsec settings:
  i. Go to *VPN > IPsec Wizard* and enter a VPN name.
  ii. Set *Template type* to *Custom*.
  iii. Click *Next*.
  iv. Configure the following *Network* settings:

IP Version	IPv6
Remote Gateway	Static IP Address
IP Address	2001:db8:d0c:2::e
Interface	port2

  v. Configure the following *Authentication* settings:

Method	Pre-shared Key
Pre-shared Key	sample

  vi. Configure the following *New Phase 2* settings:

Local Address	IPv6 Subnet
Remote Address	IPv6 Subnet

c. Configure the IPv6 firewall policy to allow IPv6 traffic from port3 to the IPsec interface:
  i. Go to *Policy & Objects > Firewall Policy*.
  ii. Click *Create New*.
  iii. Name the policy and configure the following parameters:

Incoming Interface	port3
Outgoing Interface	to_HQ2
Source	all
Destination	all
Schedule	always
Service	ALL
Action	ACCEPT

  iv. Click *OK*.
d. Configure the IPv6 firewall policy to allow IPv6 traffic from the IPsec interface to port3:
  i. Go to *Policy & Objects > Firewall Policy*.
  ii. Click *Create New*.
  iii. Name the policy and configure the following parameters:

Incoming Interface	to_HQ2

Outgoing Interface	port3
Source	all
Destination	all
Schedule	always
Service	ALL
Action	ACCEPT

iv. Click *OK*.

e. Configure the static routes:

i. Go to *Network* > *Static Routes*.

ii. Click *Create New* > *IPv6 Static Route*.

iii. Configure the following settings for the default route to the remote VPN gateway:

Destination	::/0
Gateway Address	2001:db8:d0c:1::f
Interface	port2

iv. Select *OK*.

v. Repeat the first two steps and configure the following settings for the route to the remote protected network:

Destination	2001:db8:d0c:3::/64
Interface	to_HQ2

vi. Select *OK*.

vii. Repeat the first two steps and configure the following settings for the blackhole route:

Destination	2001:db8:d0c:3::/64
Interface	Blackhole
Administrative Distance	254

viii. Select *OK*.

2. Configure the HQ2 FortiGate:

a. Configure the IPv6 address on port2 and port3:

i. Go to *Network* > *Interfaces* and edit port2.

ii. Set *IPv6 addressing mode* to *Manual* and enter the *IPv6 Address/Prefix*.

IPv6 Address/Prefix	2001:db8:d0c:2::e/64

iii. Click *OK*.

iv. Repeat these steps for port3.

IPv6 Address/Prefix	2001:db8:d0c:3::e/64

b. Configure IPsec settings:

i. Go to *VPN > IPsec Wizard* and enter a VPN name.
ii. Set *Template type* to *Custom*.
iii. Click *Next*.
iv. Configure the following *Network* settings:

IP Version	IPv6
Remote Gateway	Static IP Address
IP Address	2001:db8:d0c:1::e
Interface	port2

v. Configure the following *Authentication* settings:

Method	Pre-shared Key
Pre-shared Key	sample

vi. Configure the following *New Phase 2* settings:

Local Address	IPv6 Subnet
Remote Address	IPv6 Subnet

c. Configure the IPv6 firewall policy to allow IPv6 traffic from port3 to the IPsec interface:
i. Go to *Policy & Objects > Firewall Policy*.
ii. Click *Create New*.
iii. Name the policy and configure the following parameters:

Incoming Interface	port3
Outgoing Interface	to_HQ1
Source	all
Destination	all
Schedule	always
Service	ALL
Action	ACCEPT

iv. Click *OK*.

d. Configure the IPv6 firewall policy to allow IPv6 traffic from the IPsec interface to port3:
i. Go to *Policy & Objects > Firewall Policy*.
ii. Click *Create New*.
iii. Name the policy and configure the following parameters:

Incoming Interface	to_HQ1
Outgoing Interface	port3
Source	all

Destination	all
Schedule	always
Service	ALL
Action	ACCEPT

iv. Click *OK*.

e. Configure the static routes:

i. Go to *Network* > *Static Routes*.

ii. Click *Create New* > *IPv6 Static Route*.

iii. Configure the following settings for the default route to the remote VPN gateway:

Destination	::/0
Gateway Address	2001:db8:d0c:2::f
Interface	port2

iv. Select *OK*.

v. Repeat the first two steps and configure the following settings for the route to the remote protected network:

Destination	2001:db8:d0c:4::/64
Interface	to_HQ1

vi. Select *OK*.

vii. Repeat the first two steps and configure the following settings for the blackhole route:

Destination	2001:db8:d0c:4::/64
Interface	Blackhole
Administrative Distance	254

viii. Select *OK*.

To configure the example in the CLI:

1. Configure the HQ1 FortiGate.

a. Configure the IPv6 address on port2 and port3:

```
config system interface
    edit port2
        config ipv6
            set ip6-address 2001:db8:d0c:1::e/64
        end
    next
    edit port3
        config ipv6
            set ip6-address 2001:db8:d0c:4::e/64
        end
```

```
    next
end
```

b. Configure IPsec settings:

```
config vpn ipsec phase1-interface
    edit "to_HQ2"
        set interface port2
        set ip-version 6
        set peertype any
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set remote-gw6 2001:db8:d0c:2::e
        set psksecret sample
    next
end
config vpn ipsec phase2-interface
    edit "to_HQ2"
        set phase1name "to_HQ2"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set src-addr-type subnet6
        set dst-addr-type subnet6
    next
end
```

c. Configure the IPv6 firewall policy to allow IPv6 traffic between port3 to the IPsec interface:

```
config firewall policy
    edit 1
        set srcintf "port3"
        set dstintf "to_HQ2"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic "all"
    next
    edit 2
        set srcintf "to_HQ2"
        set dstintf "port3"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic "all"
    next
end
```

d. Configure the static routes:

```
config router static6
    edit 1
        set gateway 2001:db8:d0c:1::f
        set device "port2"
    next
    edit 2
        set dst 2001:db8:d0c:3::/64
        set device "to_HQ2"
    next
    edit 3
        set dst 2001:db8:d0c:3::/64
        set blackhole enable
        set distance 254
    next
end
```

2. Configure the HQ2 FortiGate.
   a. Configure the IPv6 address on port2 and port3:

```
config system interface
    edit port2
        config ipv6
            set ip6-address 2001:db8:d0c:2::e/64
        end
    next
    edit port3
        config ipv6
            set ip6-address 2001:db8:d0c:3::e/64
        end
    next
end
```

   b. Configure IPsec settings:

```
config vpn ipsec phase1-interface
    edit "to_HQ1"
        set interface port2
        set ip-version 6
        set peertype any
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set remote-gw6 2001:db8:d0c:1::e
        set psksecret sample
    next
end
config vpn ipsec phase2-interface
    edit "to_HQ2"
        set phase1name "to_HQ1"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set src-addr-type subnet6
        set dst-addr-type subnet6
```

```
   next
end
```

c. Configure the IPv6 firewall policy to allow IPv6 traffic between port3 to the IPsec interface:

```
config firewall policy
    edit 1
        set srcintf "port3"
        set dstintf "to_HQ1"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic "all"
    next
    edit 2
        set srcintf "to_HQ1"
        set dstintf "port3"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic "all"
    next
end
```

d. Configure the static routes:

```
config router static6
    edit 1
        set gateway 2001:db8:d0c:2::f
        set device "port2"
    next
    edit 2
        set dst 2001:db8:d0c:4::/64
        set device "to_HQ1"
    next
    edit 3
        set dst 2001:db8:d0c:4::/64
        set blackhole enable
        set distance 254
    next
end
```

Verification

The following commands are useful to check IPsec phase1/phase2 interface status:

1. Run the `diagnose vpn ike gateway list` command on HQ1. The system should return the following:

```
vd: root/0
name: to_HQ2
version: 1
interface: port2 6
addr: 2001:db8:d0c:1::e:500 -> 2001:db8:d0c:2::e:500
tun_id: 10.0.0.1/::10.0.0.1
remote_location: 0.0.0.0
network-id: 0
created: 1537s ago
peer-id: 2001:db8:d0c:2::e
peer-id-auth: no
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 160 8d7231820bb5ffd9/6c840a4c11b57927
  direction: initiator
  status: established 1537-1537s ago = 0ms
  proposal: aes128-sha256
  key: 32d8521a77d98529-5fe4b67914d30f87
  lifetime/rekey: 86400/84562
  DPD sent/recv: 00000007/00000003
  peer-id: 2001:db8:d0c:2::e
```

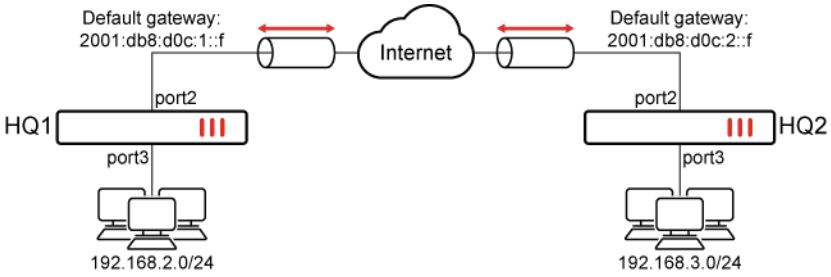
2. Run the `diagnose vpn tunnel list` command on HQ1. The system should return the following:

```
list all ipsec tunnel in vd 0
------------------------------------------------------
name=to_HQ2 ver=1 serial=1 2001:db8:d0c:1::e:0->2001:db8:d0c:2::e:0 tun_id=10.0.0.1 tun_
id6=::10.0.0.1 dst_mtu=1500 dpd-link=on weight=1
bound_if=6 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc  run_
state=0 role=primary accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=4 ilast=416 olast=416 ad=/0
stat: rxp=28 txp=51 rxb=76440 txb=274972
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=7
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=to_HQ2 proto=0 sa=1 ref=2 serial=1
  src: 0:::-ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff:0
  dst: 0:::-ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff:0
  SA:  ref=3 options=10202 type=00 soft=0 mtu=1422 expire=41332/0B replaywin=2048
       seqno=34 esn=0 replaywin_lastseq=0000001d qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42897/43200
  dec: spi=97511f0c esp=aes key=16 3b65a0121f54e07101d7b7a84b0ce243
       ah=sha1 key=20 c64cdcb40949573383c2c9f26d5af5d63776b1ce
  enc: spi=0f65cc64 esp=aes key=16 3de2f282167bac00d0a9dd942359cff3
       ah=sha1 key=20 e80e31d277f045053950e56db9eec5b6e529ea1a
  dec:pkts/bytes=56/152880, enc:pkts/bytes=99/357420
  npu_flag=00 npu_rgwy=2001:db8:d0c:2::e npu_lgwy=2001:db8:d0c:1::e npu_selid=0 dec_npuid=0
enc_npuid=0
run_tally=0
```

Site-to-site IPv4 over IPv6 VPN example

In this example, two private networks with IPv4 addressing communicate securely over IPv6 infrastructure.

The following topology is used for this example:



- Port2 connects to the IPv6 public network and port3 connects to the IPv4 local network.
- HQ1 port2 IPv6 address is 2001:db8:d0c:1::e and port3 IPv4 address is 192.168.2.1.
- HQ2 port2 IPv6 address is 2001:db8:d0c:2::e and port3 IPv4 address is 192.168.3.1.

Please note that the IPv6 addresses used in this example are for illustrative purposes only and should not be used in your environment.

The 2001:db8::/32 prefix is a special IPv6 prefix designated for use in documentation examples. See RFC 3849 for more information.

To configure the example in the GUI:

1. Configure the HQ1 FortiGate.
   a. Configure the IPv6 address on port2 and IPv4 address on port3:
      i. Go to *Network* > *Interfaces* and edit port2.
      ii. Set *IPv6 addressing mode* to *Manual* and enter the *IPv6 Address/Prefix*.

IPv6 Address/Prefix	2001:db8:d0c:1::e/64

      iii. Click *OK*.
      iv. Go to *Network* > *Interfaces* and edit port3.
      v. Set *Addressing mode* to *Manual* and enter the *IP/Netmask*.

IP/Netmask	192.168.2.1/24

   b. Configure IPsec settings:
      i. Go to *VPN* > *IPsec Wizard* and enter a VPN name.
      ii. Set *Template type* to *Custom*.
      iii. Click *Next*.
      iv. Configure the following *Network* settings:

IP Version	IPv6

Remote Gateway	Static IP Address
IP Address	2001:db8:d0c:2::e
Interface	port2

v. Configure the following *Authentication* settings:

Method	Pre-shared Key
Pre-shared Key	sample

vi. Configure the following *New Phase 2* settings:

Local Address	addr_subnet
Remote Address	addr_subnet

c. Configure the IPv4 firewall policy to allow IPv4 traffic from port3 to the IPsec interface:
   i. Go to *Policy & Objects > Firewall Policy*.
   ii. Click *Create New*.
   iii. Name the policy and configure the following parameters:

Incoming Interface	port3
Outgoing Interface	to_HQ2
Source	all
Destination	all
Schedule	always
Service	ALL
Action	ACCEPT

   iv. Click *OK*.
d. Configure the IPv4 firewall policy to allow IPv4 traffic from the IPsec interface to port3:
   i. Go to *Policy & Objects > Firewall Policy*.
   ii. Click *Create New*.
   iii. Name the policy and configure the following parameters:

Incoming Interface	to_HQ2
Outgoing Interface	port3
Source	all
Destination	all
Schedule	always
Service	ALL
Action	ACCEPT

   iv. Click *OK*.

e. Configure the static routes:

i. Go to *Network* > *Static Routes*.

ii. Click *Create New* > *IPv6 Static Route*.

iii. Configure the following settings for the default route to the remote VPN gateway:

Destination	0.0.0.0/0.0.0.0
Gateway Address	2001:db8:d0c:1::f
Interface	port2

iv. Select *OK*.

v. Repeat the first two steps for *IPv4 Static Route* and configure the following settings for the route to the remote protected network:

Destination	192.168.3.0/24
Interface	to_HQ2

vi. Select *OK*.

vii. Repeat the first two steps for *IPv4 Static Route* and configure the following settings for the blackhole route:

Destination	192.168.3.0/24
Interface	Blackhole
Administrative Distance	254

viii. Select *OK*.

2. Configure the HQ2 FortiGate:

a. Configure the IPv6 address on port2 and IPv4 address on port3:

i. Go to *Network* > *Interfaces* and edit port2.

ii. Set *IPv6 addressing mode* to *Manual* and enter the *IPv6 Address/Prefix*.

IPv6 Address/Prefix	2001:db8:d0c:2::e/64

iii. Click *OK*.

iv. Go to *Network* > *Interfaces* and edit port3.

v. Set *Addressing mode* to *Manual* and enter the *IP/Netmask*.

IP/Netmask	192.168.3.1/24

b. Configure IPsec settings:

i. Go to *VPN* > *IPsec Wizard* and enter a VPN name.

ii. Set *Template type* to *Custom*.

iii. Click *Next*.

iv. Configure the following *Network* settings:

IP Version	IPv6
Remote Gateway	Static IP Address

IP Address	2001:db8:d0c:1::e
Interface	port2

v. Configure the following *Authentication* settings:

Method	Pre-shared Key
Pre-shared Key	sample

vi. Configure the following *New Phase 2* settings:

Local Address	addr_subnet
Remote Address	addr_subnet

c. Configure the IPv4 firewall policy to allow IPv4 traffic from port3 to the IPsec interface:
   i. Go to *Policy & Objects > Firewall Policy*.
   ii. Click *Create New*.
   iii. Name the policy and configure the following parameters:

Incoming Interface	port3
Outgoing Interface	to_HQ1
Source	all
Destination	all
Schedule	always
Service	ALL
Action	ACCEPT

   iv. Click *OK*.
d. Configure the IPv4 firewall policy to allow IPv4 traffic from the IPsec interface to port3:
   i. Go to *Policy & Objects > Firewall Policy*.
   ii. Click *Create New*.
   iii. Name the policy and configure the following parameters:

Incoming Interface	to_HQ1
Outgoing Interface	port3
Source	all
Destination	all
Schedule	always
Service	ALL
Action	ACCEPT

   iv. Click *OK*.
e. Configure the static routes:

i. Go to *Network* > *Static Routes*.
ii. Click *Create New* > *IPv6 Static Route*.
iii. Configure the following settings for the default route to the remote VPN gateway:

Destination	0.0.0.0/0.0.0.0
Gateway Address	2001:db8:d0c:2::f
Interface	port2

iv. Select *OK*.
v. Repeat the first two steps for *IPv4 Static Route* and configure the following settings for the route to the remote protected network:

Destination	192.168.2.0/24
Interface	to_HQ1

vi. Select *OK*.
vii. Repeat the first two steps for *IPv4 Static Route* and configure the following settings for the blackhole route:

Destination	192.168.2.0/24
Interface	Blackhole
Administrative Distance	254

viii. Select *OK*.

To configure the example in the CLI:

1. Configure the HQ1 FortiGate.
   a. Configure the IPv6 address on port2 and IPv4 address on port3:

```
config system interface
    edit port2
        config ipv6
            set ip6-address 2001:db8:d0c:1::e/64
        end
    next
    edit port3
        set ip 192.168.2.1/24
    next
end
```

   b. Configure IPsec settings:

```
config vpn ipsec phase1-interface
    edit "to_HQ2"
        set interface port2
        set ip-version 6
        set peertype any
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
```

```
        set remote-gw6 2001:db8:d0c:2::e
        set psksecret sample
    next
end
config vpn ipsec phase2-interface
    edit "to_HQ2"
        set phase1name "to_HQ2"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set src-addr-type subnet
        set dst-addr-type subnet
    next
end
```

c. Configure the IPv4 firewall policy to allow IPv4 traffic between port3 to the IPsec interface:

```
config firewall policy
    edit 1
        set srcintf "port3"
        set dstintf "to_HQ2"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic "all"
    next
    edit 2
        set srcintf "to_HQ2"
        set dstintf "port3"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic "all"
    next
end
```

d. Configure the static routes:

```
config router static6
    edit 1
        set gateway 2001:db8:d0c:1::f
        set device "port2"
    next
end
config router static
    edit 1
        set dst 192.168.3.0 255.255.255.0
        set device "to_HQ2"
    next
    edit 2
```

```
        set dst 192.168.3.0 255.255.255.0
        set device blackhole
        set distance 254
    next
end
```

2. Configure the HQ2 FortiGate.
   a. Configure the IPv6 address on port2 and IPv4 address on port3:

```
config system interface
    edit port2
        config ipv6
            set ip6-address 2001:db8:d0c:2::e/64
        end
    next
    edit port3
        set ip 192.168.3.1/24
    next
end
```

   b. Configure IPsec settings:

```
config vpn ipsec phase1-interface
    edit "to_HQ1"
        set interface port2
        set ip-version 6
        set peertype any
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set remote-gw6 2001:db8:d0c:1::e
        set psksecret sample
    next
end
config vpn ipsec phase2-interface
    edit "to_HQ2"
        set phase1name "to_HQ1"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set src-addr-type subnet
        set dst-addr-type subnet
    next
end
```

   c. Configure the IPv4 firewall policy to allow IPv4 traffic between port3 to the IPsec interface:

```
config firewall policy
    edit 1
        set srcintf "port3"
        set dstintf "to_HQ1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
```

```
        set service "ALL"
        set logtraffic "all"
    next
    edit 2
        set srcintf "to_HQ1"
        set dstintf "port3"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic "all"
    next
end
```

d. Configure the static routes:

```
config router static6
    edit 1
        set gateway 2001:db8:d0c:2::f
        set device "port2"
    next
end
config router static
    edit 1
        set dst 192.168.2.0 255.255.255.0
        set device "to_HQ1"
    next
    edit 2
        set dst 192.168.2.0 255.255.255.0
        set device blackhole
        set distance 254
    next
end
```

Verification

The following commands are useful to check IPsec phase1/phase2 interface status:

1. Run the `diagnose vpn ike gateway list` command on HQ1. The system should return the following:

```
vd: root/0
name: to_HQ2
version: 1
interface: port2 6
addr: 2001:db8:d0c:1::e:500 -> 2001:db8:d0c:2::e:500
tun_id: 10.0.0.1/::10.0.0.1
remote_location: 0.0.0.0
network-id: 0
created: 7215s ago
peer-id: 2001:db8:d0c:2::e
peer-id-auth: no
```

```
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/2  established 1/2  time 0/5/10 ms

  id/spi: 160 8d7231820bb5ffd9/6c840a4c11b57927
  direction: initiator
  status: established 7215-7215s ago = 0ms
  proposal: aes128-sha256
  key: 32d8521a77d98529-5fe4b67914d30f87
  lifetime/rekey: 86400/78884
  DPD sent/recv: 00000007/00000003
  peer-id: 2001:db8:d0c:2::e
```

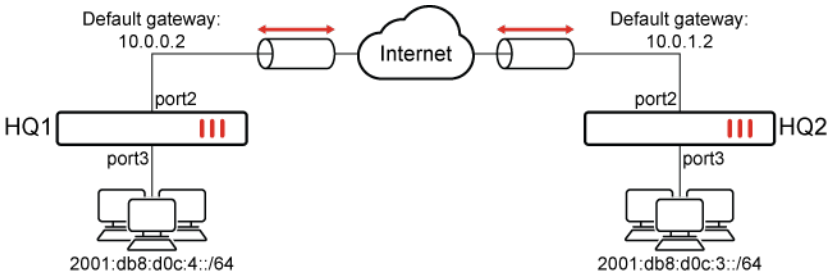
2. Run the `diagnose vpn tunnel list` command on HQ1. The system should return the following:

```
list all ipsec tunnel in vd 0
------------------------------------------------------
name=to_HQ2 ver=1 serial=1 2001:db8:d0c:1::e:0->2001:db8:d0c:2::e:0 tun_id=10.0.0.1 tun_
id6=::10.0.0.1 dst_mtu=1500 dpd-link=on weight=1
bound_if=6 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc  run_
state=0 role=primary accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=4 ilast=581 olast=581 ad=/0
stat: rxp=4 txp=4 rxb=26312 txb=26312
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=7
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=to_HQ2 proto=0 sa=1 ref=2 serial=2
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=10202 type=00 soft=0 mtu=1422 expire=42116/0B replaywin=2048
       seqno=5 esn=0 replaywin_lastseq=00000005 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42902/43200
  dec: spi=97511f0d esp=aes key=16 c7323977434f48604c37d7be423f7519
       ah=sha1 key=20 ee8f9b457cec9b3c2e614db058bb97896d7ef8d9
  enc: spi=0f65cc65 esp=aes key=16 8b78642018b02165d1ef29ad3d8215c8
       ah=sha1 key=20 f2adca47b0b3925a87e329a237f0fd521e0afd19
  dec:pkts/bytes=8/52624, enc:pkts/bytes=8/52984
  npu_flag=00 npu_rgwy=2001:db8:d0c:2::e npu_lgwy=2001:db8:d0c:1::e npu_selid=1 dec_npuid=0
enc_npuid=0
run_tally=0
```

Site-to-site IPv6 over IPv4 VPN example

In this example, IPv6-addressed networks communicate securely over IPv4 public infrastructure.

The following topology is used for this example:



- Port2 connects to the IPv4 public network and port3 connects to the IPv6 local network.
- HQ1 port2 IPv4 address is 10.0.0.1 and port3 IPv6 address is 2001:db8:d0c:4::e.
- HQ2 port2 IPv4 address is 10.0.1.1 and port3 IPv6 address is 2001:db8:d0c:3::e.

Please note that the IPv6 addresses used in this example are for illustrative purposes only and should not be used in your environment.

The 2001:db8::/32 prefix is a special IPv6 prefix designated for use in documentation examples. See RFC 3849 for more information.

To configure the example in the GUI:

1. Configure the HQ1 FortiGate.
   a. Configure the IPv4 address on port2 and IPv6 address on port3:
      i. Go to *Network* > *Interfaces* and edit port2.
      ii. Set *Addressing mode* to *Manual* and enter the *IP/Netmask*.

IP/Netmask	10.0.0.1/24

      iii. Click *OK*.
      iv. Go to *Network* > *Interfaces* and edit port3.
      v. Set *IPv6 addressing mode* to *Manual* and enter the *IPv6 Address/Prefix*.

IPv6 Address/Prefix	2001:db8:d0c:4::e/64

   b. Configure IPsec settings:
      i. Go to *VPN* > *IPsec Wizard* and enter a VPN name.
      ii. Set *Template type* to *Custom*.
      iii. Click *Next*.
      iv. Configure the following *Network* settings:

IP Version	IPv4
Remote Gateway	Static IP Address
IP Address	10.0.1.1
Interface	port2

      v. Configure the following *Authentication* settings:

Method	Pre-shared Key
Pre-shared Key	sample

vi. Configure the following *New Phase 2* settings:

Local Address	IPv6 Subnet
Remote Address	IPv6 Subnet

c. Configure the IPv6 firewall policy to allow IPv6 traffic from port3 to the IPsec interface:
   i. Go to *Policy & Objects > Firewall Policy*.
   ii. Click *Create New*.
   iii. Name the policy and configure the following parameters:

Incoming Interface	port3
Outgoing Interface	to_HQ2
Source	all
Destination	all
Schedule	always
Service	ALL
Action	ACCEPT

   iv. Click *OK*.
d. Configure the IPv6 firewall policy to allow IPv6 traffic from the IPsec interface to port3:
   i. Go to *Policy & Objects > Firewall Policy*.
   ii. Click *Create New*.
   iii. Name the policy and configure the following parameters:

Incoming Interface	to_HQ2
Outgoing Interface	port3
Source	all
Destination	all
Schedule	always
Service	ALL
Action	ACCEPT

   iv. Click *OK*.
e. Configure the static routes:
   i. Go to *Network > Static Routes*.
   ii. Click *Create New > IPv4 Static Route*.
   iii. Configure the following settings for the default route to the remote VPN gateway:

Destination	0.0.0.0/0.0.0.0
Gateway Address	10.0.0.2
Interface	port2

iv. Select *OK*.

v. Repeat the first two steps for *IPv6 Static Route* and configure the following settings for the route to the remote protected network:

Destination	2001:db8:d0c:3::/64
Interface	to_HQ2

vi. Select *OK*.

vii. Repeat the first two steps for *IPv6 Static Route* and configure the following settings for the blackhole route:

Destination	2001:db8:d0c:3::/64
Interface	Blackhole
Administrative Distance	254

viii. Select *OK*.

2. Configure the HQ2 FortiGate:

a. Configure the IPv4 address on port2 and IPv6 address on port3:

i. Go to *Network* > *Interfaces* and edit port2.

ii. Set *Addressing mode* to *Manual* and enter the *IP/Netmask*.

IP/Netmask	10.0.1.1/24

iii. Click *OK*.

iv. Go to *Network* > *Interfaces* and edit port3.

v. Set *IPv6 addressing mode* to *Manual* and enter the *IPv6 Address/Prefix*.

IPv6 Address/Prefix	2001:db8:d0c:3::e/64

b. Configure IPsec settings:

i. Go to *VPN* > *IPsec Wizard* and enter a VPN name.

ii. Set *Template type* to *Custom*.

iii. Click *Next*.

iv. Configure the following *Network* settings:

IP Version	IPv4
Remote Gateway	Static IP Address
IP Address	10.0.0.1
Interface	port2

v. Configure the following *Authentication* settings:

Method	Pre-shared Key
Pre-shared Key	sample

vi. Configure the following *New Phase 2* settings:

Local Address	IPv6 Subnet
Remote Address	IPv6 Subnet

c. Configure the IPv6 firewall policy to allow IPv6 traffic from port3 to the IPsec interface:
   i. Go to *Policy & Objects > Firewall Policy*.
   ii. Click *Create New*.
   iii. Name the policy and configure the following parameters:

Incoming Interface	port3
Outgoing Interface	to_HQ1
Source	all
Destination	all
Schedule	always
Service	ALL
Action	ACCEPT

   iv. Click *OK*.
d. Configure the IPv6 firewall policy to allow IPv6 traffic from the IPsec interface to port3:
   i. Go to *Policy & Objects > Firewall Policy*.
   ii. Click *Create New*.
   iii. Name the policy and configure the following parameters:

Incoming Interface	to_HQ1
Outgoing Interface	port3
Source	all
Destination	all
Schedule	always
Service	ALL
Action	ACCEPT

   iv. Click *OK*.
e. Configure the static routes:
   i. Go to *Network > Static Routes*.
   ii. Click *Create New > IPv4 Static Route*.
   iii. Configure the following settings for the default route to the remote VPN gateway:

Destination	0.0.0.0/0.0.0.0
Gateway Address	10.0.1.2
Interface	port2

iv. Select *OK*.

v. Repeat the first two steps for *IPv6 Static Route* and configure the following settings for the route to the remote protected network:

Destination	2001:db8:d0c:4::/64
Interface	to_HQ1

vi. Select *OK*.

vii. Repeat the first two steps for *IPv6 Static Route* and configure the following settings for the blackhole route:

Destination	2001:db8:d0c:4::/64
Interface	Blackhole
Administrative Distance	254

viii. Select *OK*.

To configure the example in the CLI:

1. Configure the HQ1 FortiGate.
   a. Configure the IPv6 address on port2 and port3:

```
config system interface
    edit port2
        set ip 10.0.0.1/24
    next
    edit port3
        config ipv6
            set ip6-address 2001:db8:d0c:4::e/64
        end
    next
end
```

   b. Configure IPsec settings:

```
config vpn ipsec phase1-interface
    edit "to_HQ2"
        set interface port2
        set ip-version 4
        set peertype any
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set remote-gw 10.0.1.1
        set psksecret sample
    next
end
```

```
config vpn ipsec phase2-interface
    edit "to_HQ2"
        set phase1name "to_HQ2"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set src-addr-type subnet6
        set dst-addr-type subnet6
    next
end
```

c. Configure the IPv6 firewall policy to allow IPv6 traffic between port3 to the IPsec interface:

```
config firewall policy
    edit 1
        set srcintf "port3"
        set dstintf "to_HQ2"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic "all"
    next
    edit 2
        set srcintf "to_HQ2"
        set dstintf "port3"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic "all"
    next
end
```

d. Configure the static routes:

```
config router static
    edit 1
        set gateway 10.0.0.2
        set device "port2"
    next
end
config router static6
    edit 1
        set dst 2001:db8:d0c:3::/64
        set device "to_HQ2"
    next
    edit 2
        set dst 2001:db8:d0c:3::/64
        set device blackhole
        set distance 254
```

```
   next
end
```

2. Configure the HQ2 FortiGate.
   a. Configure the IPv6 address on port2 and port3:

```
config system interface
    edit port2
        set ip 10.0.1.1/24
    next
    edit port3
        config ipv6
            set ip6-address 2001:db8:d0c:3::e/64
        end
    next
end
```

   b. Configure IPsec settings:

```
config vpn ipsec phase1-interface
    edit "to_HQ1"
        set interface port2
        set ip-version 4
        set peertype any
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set remote-gw 10.0.0.1
        set psksecret sample
    next
end
config vpn ipsec phase2-interface
    edit "to_HQ2"
        set phase1name "to_HQ1"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set src-addr-type subnet6
        set dst-addr-type subnet6
    next
end
```

   c. Configure the IPv6 firewall policy to allow IPv6 traffic between port3 to the IPsec interface:

```
config firewall policy
    edit 1
        set srcintf "port3"
        set dstintf "to_HQ1"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic "all"
    next
```

```
    edit 2
        set srcintf "to_HQ1"
        set dstintf "port3"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic "all"
    next
end
```

d. Configure the static routes:

```
config router static
    edit 1
        set gateway 10.0.1.2
        set device "port2"
    next
end
config router static6
    edit 1
        set dst 2001:db8:d0c:4::/64
        set device "to_HQ1"
    next
    edit 2
        set dst 2001:db8:d0c:4::/64
        set device blackhole
        set distance 254
    next
end
```

Verification

The following commands are useful to check IPsec phase1/phase2 interface status:

1. Run the `diagnose vpn ike gateway list` command on HQ1. The system should return the following:

```
vd: root/0
name: to_HQ2
version: 1
interface: port2 6
addr: 10.0.0.1:500 -> 10.0.0.2:500
tun_id: 10.0.0.2/::10.0.0.2
remote_location: 0.0.0.0
network-id: 0
created: 576319s ago
peer-id: 10.0.0.2
peer-id-auth: no
IKE SA: created 1/8  established 1/8  time 0/1127/9000 ms
IPsec SA: created 1/7  established 1/7  time 0/5/10 ms
```

```
  id/spi: 8 c04ab0ead989f579/267813e164d4ec22
  direction: initiator
  status: established 59710-59710s ago = 0ms
  proposal: aes128-sha256
  key: 034a0c3bf3deb551-8d647af9b6f76578
  lifetime/rekey: 86400/26389
  DPD sent/recv: 00000044/00000047
  peer-id: 10.0.0.2
```

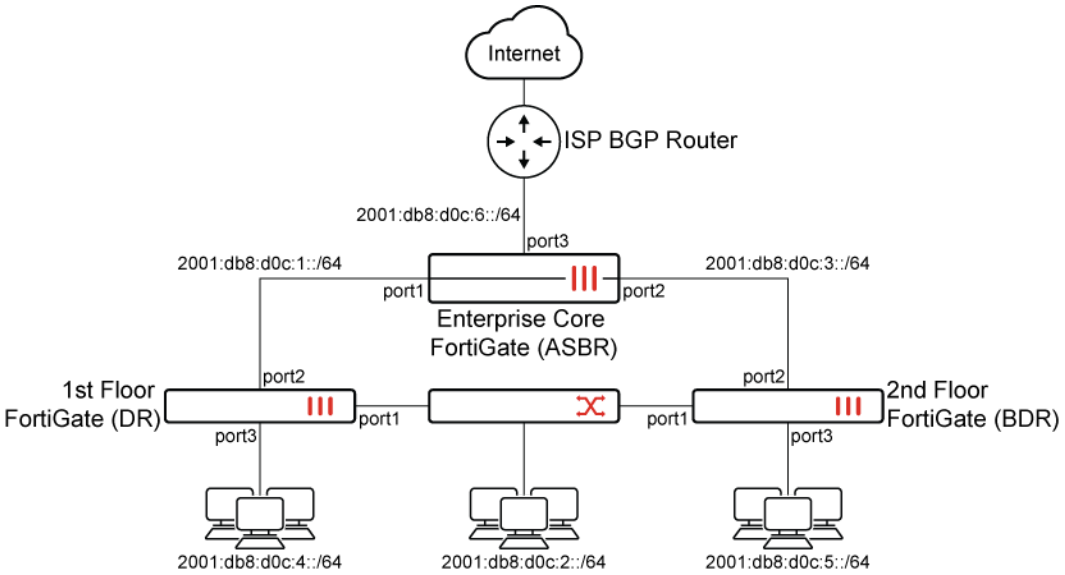
2. Run the `diagnose vpn tunnel list` command on HQ1. The system should return the following:

```
list all ipsec tunnel in vd 0
------------------------------------------------------
name=to_HQ2 ver=1 serial=1 10.0.0.1:0->10.0.0.2:0 tun_id=10.0.0.2 tun_id6=::10.0.0.2 dst_
mtu=1500 dpd-link=on weight=1
bound_if=6 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc  run_
state=0 role=primary accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=4 ilast=27652 olast=27652 ad=/0
stat: rxp=198 txp=192 rxb=15840 txb=15360
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=68
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=to_HQ2 proto=0 sa=1 ref=2 serial=1
  src: 0:::-ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff:0
  dst: 0:::-ffff:ffff:ffff:ffff:ffff:ffff:ffff:ffff:0
  SA:  ref=3 options=10202 type=00 soft=0 mtu=1438 expire=2709/0B replaywin=2048
       seqno=d esn=0 replaywin_lastseq=0000000c qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42933/43200
  dec: spi=24fe1f3a esp=aes key=16 de482993279020176bb2709052ef0656
       ah=sha1 key=20 b6fe007aa8e2c587762c4f9808321ae5e015dc0a
  enc: spi=5989a2d9 esp=aes key=16 438c8d60ae9ca8400138965ff90a1384
       ah=sha1 key=20 a931ee4518c365dae630431b25edfe6d930e8075
  dec:pkts/bytes=22/1760, enc:pkts/bytes=24/2784
  npu_flag=00 npu_rgwy=10.0.0.2 npu_lgwy=10.0.0.1 npu_selid=0 dec_npuid=0 enc_npuid=0
```

Basic OSPFv3 example

In this example, three FortiGate devices are configured in an OSPF network.

- 1st Floor FortiGate is the Designated Router (DR). It has the highest priority and the lowest IP address, to ensure that it becomes the DR.
- 2nd Floor FortiGate is the Backup Designated Router (BDR). It has a high priority to ensure that it becomes the BDR.
- Enterprise Core FortiGate is the Autonomous System Border Router (ASBR). It routes all traffic to the ISP BGP router for internet access. It redistributes routes from BGP and advertises a default route to its neighbors. It can allow different types of routes, learned outside of OSPF, to be used in OSPF. Different metrics can be assigned to these routes to make them more or less preferred than regular OSPF routes. Route maps could be used to further control what prefixes are advertised or received from the ISP.



Please note that the IPv6 addresses used in this example are for illustrative purposes only and should not be used in your environment.

The 2001:db8::/32 prefix is a special IPv6 prefix designated for use in documentation examples. See RFC 3849 for more information.

FortiGate	Interface	IP address
1st Floor FortiGate (DR)	loopback	1.1.1.1
	port1	2001:db8:d0c:2::1/64
	port2	2001:db8:d0c:1::2/64
	port3	2001:db8:d0c:4::1/64
2nd Floor FortiGate (BDR)	loopback	2.2.2.2
	port1	2001:db8:d0c:2::2/64
	port2	2001:db8:d0c:3::2/64
	port3	2001:db8:d0c:5::1/64
Enterprise Core FortiGate (ASBR)	loopback	13.13.13.13
	port1	2001:db8:d0c:1::1/64
	port2	2001:db8:d0c:3::1/64
	port3	2001:db8:d0c:6::1/64

- Firewall policies are already configured to allow unfiltered traffic in both directions between all of the connected interfaces.
- The interfaces are already configured. The cost for all of the interfaces is left at 0.
- The OSPF network belongs to Area 0, and is not connected to any other OSPF networks. All of the routers are part of the backbone 0.0.0.0 area, so no inter-area communications are needed.

- Enterprise Core FortiGate redistributes BGP routes into the OSPF AS and peers with the ISP BGP Router over eBGP. For information about configuring BGP, see Basic IPv6 BGP example on page 981.
- The ISP IPv6 address is 2001:db8:d0c:6::2/64.

1st Floor FortiGate

To configure 1st Floor FortiGate in the CLI:

```
config router ospf6
    set router-id 1.1.1.1
    config area
        edit 0.0.0.0
        next
    end
    config ospf6-interface
        edit "1st-Floor-FortiGate-Internal-DR"
            set interface "port1"
            set priority 255
            set dead-interval 40
            set hello-interval 10
        next
        edit "1st-Floor-FortiGate-External"
            set interface "port2"
            set dead-interval 40
            set hello-interval 10
        next
    end
    edit "1st-Floor-FortiGate-Internal"
        set interface "port3"
        set dead-interval 40
        set hello-interval 10
    next
end
```

2nd Floor FortiGate

To configure 2nd Floor FortiGate in the CLI:

```
config router ospf6
    set router-id 2.2.2.2
    config area
        edit 0.0.0.0
        next
    end
    config ospf6-interface
        edit "2nd-Floor-FortiGate-Internal"
            set interface "port1"
            set priority 250
            set dead-interval 40
            set hello-interval 10
```

```
        next
        edit "2nd-Floor-FortiGate-External"
            set interface "port2"
            set dead-interval 40
            set hello-interval 10
        next
    end
    edit "2nd-Floor-FortiGate-Internal1"
        set interface "port3"
        set dead-interval 40
        set hello-interval 10
    next
end
```

Enterprise Core FortiGate

To configure Enterprise Core FortiGate in the CLI:

```
config router ospf6
    set default-information-originate enable
    set router-id 13.13.13.13
    config area
        edit 0.0.0.0
        next
    end
    config ospf6-interface
        edit "Enterprise-Core-FortiGate-Internal"
            set interface "port1"
            set dead-interval 40
            set hello-interval 10
        next
        edit "Enterprise-Core-FortiGate-Internal2"
            set interface "port2"
            set dead-interval 40
            set hello-interval 10
        next
    end
    config redistribute "bgp"
        set status enable
    end
end
```

Testing and configuration

Both the network connectivity and OSPF routing are tested. When a link goes down, routes should converge as expected.

1. Working state
   - Enterprise Core FortiGate:

```
# get router info6 ospf neighbor
OSPFv3 Process (root)
Neighbor ID     Pri   State           Dead Time   Interface
1.1.1.1           1   Full/Backup     00:00:38    port1
2.2.2.2           1   Full/Backup     00:00:32    port2
```

```
# get router info6 ospf status
 Routing Process "OSPFv3 (root)" with ID 13.13.13.13
 Process uptime is 28 minutes
 Do not support Restarting
 This router is an ASBR (injecting external routing information)
 SPF schedule delay 5 secs, Hold time between SPFs 10 secs
 Minimum LSA interval 5 secs, Minimum LSA arrival 1 secs
 Number of incomming current DD exchange neighbors 0/5
 Number of outgoing current DD exchange neighbors 0/5
 Number of external LSA 0. Checksum Sum 0x0000
 Number of AS-Scoped Unknown LSA 0
 Number of LSA originated 14
 Number of LSA received 187
 Number of areas in this router is 1
    Area BACKBONE(0)
        Number of interfaces in this area is 2(2)
        SPF algorithm executed 36 times
        Number of LSA 9.  Checksum Sum 0x2DB91
        Number of Unknown LSA
```

```
# get router info6 routing-table
IPv6 Routing Table
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, B - BGP, V - BGP VPNv6
       * - candidate default

Timers: Uptime

Routing table for VRF=0
B*      ::/0 [20/0] via fe80::20c:29ff:febc:eec2, port3, 00:02:56, [1024/0]
C       ::1/128 via ::, root, 00:17:23
B       64:ff9b::/96 [20/0] via fe80::20c:29ff:febc:eec2, port3, 00:02:56, [1024/0]
C       2001:db8:d0c:1::/64 via ::, port1, 00:17:23
O       2001:db8:d0c:2::/64 [110/2] via fe80::20c:29ff:fe4d:f81f, port1, 00:16:36,
[1024/0]
                            [110/2] via fe80::20c:29ff:fe6b:b2c9, port2, 00:16:36,
[1024/0]
C       2001:db8:d0c:3::/64 via ::, port2, 00:17:23
O       2001:db8:d0c:4::/64 [110/2] via fe80::20c:29ff:fe4d:f81f, port1, 00:16:36,
[1024/0]
```

```
O       2001:db8:d0c:5::/64 [110/2] via fe80::20c:29ff:fe6b:b2c9, port2, 00:16:52,
[1024/0]
C       2001:db8:d0c:6::/64 via ::, port3, 00:17:23
```

- 2nd Floor FortiGate:

```
# get router info6 ospf neighbor
OSPFv3 Process (root)
Neighbor ID     Pri   State           Dead Time   Interface
1.1.1.1         255   Full/DR         00:00:35    port1
13.13.13.13       1   Full/DR         00:00:31    port2
```

```
# get router info6 ospf status
 Routing Process "OSPFv3 (root)" with ID 2.2.2.2
 Process is not up
 SPF schedule delay 5 secs, Hold time between SPFs 10 secs
 Minimum LSA interval 5 secs, Minimum LSA arrival 1 secs
 Number of incomming current DD exchange neighbors 0/5
 Number of outgoing current DD exchange neighbors 0/5
 Number of external LSA 0. Checksum Sum 0x0000
 Number of AS-Scoped Unknown LSA 0
 Number of LSA originated 19
 Number of LSA received 157
 Number of areas in this router is 1
    Area BACKBONE(0)
        Number of interfaces in this area is 2(2)
        SPF algorithm executed 32 times
        Number of LSA 9.  Checksum Sum 0x2D793
        Number of Unknown LSA
```

```
# get router info6 routing-table
IPv6 Routing Table
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, B - BGP
       * - candidate default

Timers: Uptime

O*E2    ::/0 [110/10] via fe80::20c:29ff:fefc:185e, port2, 00:00:37
C       ::1/128 via ::, root, 00:15:47
O E2    64:ff9b::/96 [110/10] via fe80::20c:29ff:fefc:185e, port2, 00:00:37
O       2001:db8:d0c:1::/64 [110/2] via fe80::20c:29ff:fe4d:f815, port1, 00:14:10
                            [110/2] via fe80::20c:29ff:fefc:185e, port2, 00:14:10
C       2001:db8:d0c:2::/64 via ::, port1, 00:15:47
C       2001:db8:d0c:3::/64 via ::, port2, 00:15:47
O       2001:db8:d0c:4::/64 [110/2] via fe80::20c:29ff:fe4d:f815, port1, 00:14:36
C       2001:db8:d0c:5::/64 via ::, port3, 00:15:47
C       fe80::/64 via ::, port8, 00:15:47
```

The default route advertised by Enterprise Core FortiGate using `default-information-originate` is considered an OSPF E2 route. Other routes redistributed from BGP are also E2 routes.

- 1st Floor FortiGate:

```
# get router info6 ospf neighbor
OSPFv3 Process (root)
Neighbor ID     Pri   State           Dead Time   Interface
2.2.2.2         250   Full/Backup     00:00:33    port1
13.13.13.13       1   Full/DR         00:00:31    port2
```

```
# get router info6 ospf status
 Routing Process "OSPFv3 (root)" with ID 1.1.1.1
 Process uptime is 38 minutes
 SPF schedule delay 5 secs, Hold time between SPFs 10 secs
 Minimum LSA interval 5 secs, Minimum LSA arrival 1 secs
 Number of incomming current DD exchange neighbors 0/5
 Number of outgoing current DD exchange neighbors 0/5
 Number of external LSA 0. Checksum Sum 0x0000
 Number of AS-Scoped Unknown LSA 0
 Number of LSA originated 21
 Number of LSA received 95
 Number of areas in this router is 1
    Area BACKBONE(0)
        Number of interfaces in this area is 2(2)
        SPF algorithm executed 30 times
        Number of LSA 9.  Checksum Sum 0x2D793
        Number of Unknown LSA 0
```

```
# get router info6 routing-table
IPv6 Routing Table
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, B - BGP
       * - candidate default

Timers: Uptime

Routing table for VRF=0
O*E2    ::/0 [110/10] via fe80::20c:29ff:fefc:1854, port2, 00:00:12
C       ::1/128 via ::, root, 00:15:10
O E2    64:ff9b::/96 [110/10] via fe80::20c:29ff:fefc:1854, port2, 00:00:12
C       2001:db8:d0c:1::/64 via ::, port2, 00:15:10
C       2001:db8:d0c:2::/64 via ::, port1, 00:15:10
O       2001:db8:d0c:3::/64 [110/2] via fe80::20c:29ff:fe6b:b2bf, port1, 00:13:45
                            [110/2] via fe80::20c:29ff:fefc:1854, port2, 00:13:45
C       2001:db8:d0c:4::/64 via ::, port3, 00:15:10
O       2001:db8:d0c:5::/64 [110/2] via fe80::20c:29ff:fe6b:b2bf, port1, 00:14:20
C       fe80::/64 via ::, port3, 00:15:10
```

2. Link down state

If port1 is disconnected on Enterprise Core FortiGate:

- Enterprise Core FortiGate:

```
# get router info6 routing-table
IPv6 Routing Table
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, B - BGP, V - BGP VPNv6
       * - candidate default

Timers: Uptime

Routing table for VRF=0
B*      ::/0 [20/0] via fe80::20c:29ff:febc:eec2, port3, 00:30:38, [1024/0]
C       ::1/128 via ::, root, 01:29:46
B       64:ff9b::/96 [20/0] via fe80::20c:29ff:febc:eec2, port3, 00:30:38, [1024/0]
O       2001:db8:d0c:1::/ 64 [110/3] via fe80::20c:29ff:fe6b:b2c9, port2, 00:00:01,
[1024/0]
O       2001:db8:d0c:2::/64 [110/2] via fe80::20c:29ff:fe6b:b2c9, port2, 00:02:57,
[1024/0]
C       2001:db8:d0c:3::/64 via ::, port2, 01:29:46
O       2001:db8:d0c:4::/64 [110/3] via fe80::20c:29ff:fe6b:b2c9, port2, 00:02:24,
[1024/0]
O       2001:db8:d0c:5::/64 [110/2] via fe80::20c:29ff:fe6b:b2c9, port2, 00:36:14,
[1024/0]
C       2001:db8:d0c:6::/64 via ::, port3, 01:29:46
```

- 2nd Floor FortiGate:

```
# get router info6 routing-table
IPv6 Routing Table
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, B - BGP
       * - candidate default

Timers: Uptime

O*E2    ::/0 [110/10] via fe80::20c:29ff:fefc:185e, port2, 00:28:38
C       ::1/128 via ::, root, 01:28:29
O E2    64:ff9b::/96 [110/10] via fe80::20c:29ff:fefc:185e, port2, 00:28:38F
O       2001:db8:d0c:1::/64 [110/2] via fe80::20c:29ff:fe4d:f815, port1, 00:00:27
C       2001:db8:d0c:2::/64 via ::, port1, 01:28:29
C       2001:db8:d0c:3::/64 via ::, port2, 01:28:29
O       2001:db8:d0c:4::/64 [110/2] via fe80::20c:29ff:fe4d:f815, port1, 00:34:12
C       2001:db8:d0c:5::/64 via ::, port3, 01:28:29
C       fe80::/64 via ::, port8, 01:28:29
```

- 1st Floor FortiGate:

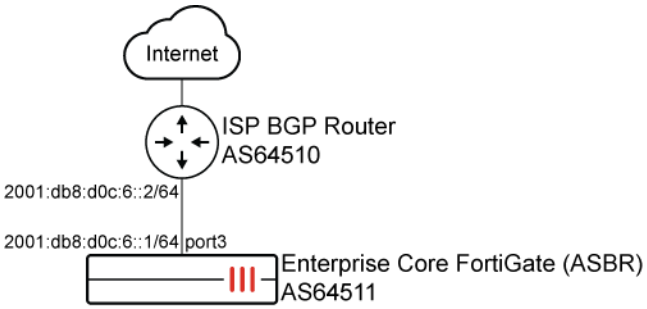
```
# get router info6 routing-table
IPv6 Routing Table
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, B - BGP
       * - candidate default

Timers: Uptime

Routing table for VRF=0
O*E2    ::/0 [110/10] via fe80::20c:29ff:fe6b:b2bf, port1, 00:00:55
C       ::1/128 via ::, root, 01:28:14
O E2    64:ff9b::/96 [110/10] via fe80::20c:29ff:fe6b:b2bf, port1, 00:00:55
C       2001:db8:d0c:1::/64 via ::, port2, 01:28:14
C       2001:db8:d0c:2::/64 via ::, port1, 01:28:14
O       2001:db8:d0c:3::/64 [110/2] via fe80::20c:29ff:fe6b:b2bf, port1, 00:00:56
C       2001:db8:d0c:4::/64 via ::, port3, 01:28:14
O       2001:db8:d0c:5::/64 [110/2] via fe80::20c:29ff:fe6b:b2bf, port1, 00:33:59
C       fe80::/64 via ::, port3, 01:28:14
```

Basic IPv6 BGP example

In this example, Enterprise Core FortiGate peers with the ISP BGP Router over eBGP to receive a default route.

Topology

The following topology is used for this example:



Please note that the IPv6 addresses used in this example are for illustrative purposes only and should not be used in your environment.

The 2001:db8::/32 prefix is a special IPv6 prefix designated for use in documentation examples. See RFC 3849 for more information.

Please note that the Autonomous System Numbers (ASN) used in this example are reserved for documentation use only and should not be used in your environment. See RFC 5398 for more information.

To configure BGP on the Enterprise Core FortiGate in the GUI:

1. Go to *Network* > *BGP*.
2. Set *Local AS* to *64511*.
3. Set *Router ID* to *13.13.13.13*.
4. In the *Neighbors* table, click *Create New* and set the following:

IP	2001:db8:d0c:6::2
Remote AS	64510

5. Click *OK*.
6. Under *IPv6 Networks*, set *IP/Netmask* to *2001:db8:d0c:6::/64*.
7. Click *Apply*.

To configure BGP on the Enterprise Core FortiGate in the CLI:

```
config router bgp
    set as 64511
    set router-id 13.13.13.13
    config neighbor
        edit "2001:db8:d0c:6::2"
            set remote-as 64510
        next
    end
    config network6
        edit 1
            set prefix6 2001:db8:d0c:6::/64
        next
    end
end
```

Testing the configuration

To verify the status of the neighbors:

```
# get router info6 bgp neighbors
VRF 0 neighbor table:
BGP neighbor is 2001:db8:d0c:6::2, remote AS 64510, local AS 64511, external link
  BGP version 4, remote router ID 1.1.1.2
  BGP state = Established, up for 02:43:35
  Last read 00:00:14, hold time is 180, keepalive interval is 60 seconds
  Configured hold time is 180, keepalive interval is 60 seconds
  Neighbor capabilities:
```

```
    Route refresh: advertised and received (old and new)
    Address family IPv4 Unicast: advertised and received
    Address family VPNv4 Unicast: advertised and received
    Address family IPv6 Unicast: advertised and received
    Address family L2VPN EVPN: advertised and received
  Received 263 messages, 0 notifications, 0 in queue
  Sent 260 messages, 1 notifications, 0 in queue
  Route refresh request: received 0, sent 0
  NLRI treated as withdraw: 0
  Minimum time between advertisement runs is 30 seconds

 For address family: IPv4 Unicast
  BGP table version 1, neighbor version 0
  Index 1, Offset 0, Mask 0x2
  Community attribute sent to this neighbor (both)
  0 accepted prefixes, 0 prefixes in rib
  0 announced prefixes

 For address family: VPNv4 Unicast
  BGP table version 1, neighbor version 0
  Index 1, Offset 0, Mask 0x2
  Community attribute sent to this neighbor (both)
  0 accepted prefixes, 0 prefixes in rib
  0 announced prefixes

 For address family: IPv6 Unicast
  BGP table version 3, neighbor version 2
  Index 1, Offset 0, Mask 0x2
  Community attribute sent to this neighbor (both)
  3 accepted prefixes, 3 prefixes in rib
  1 announced prefixes

 For address family: L2VPN EVPN
  BGP table version 1, neighbor version 1
  Index 1, Offset 0, Mask 0x2
  Community attribute sent to this neighbor (both)
  0 accepted prefixes, 0 prefixes in rib
  0 announced prefixes

 Connections established 2; dropped 1
Local host: 2001:db8:d0c:6::1, Local port: 179
Foreign host: 2001:db8:d0c:6::2, Foreign port: 16500
Egress interface: 9
Nexthop: 13.13.13.13
Nexthop interface: port3
Nexthop global: 2001:db8:d0c:6::1
Nexthop local: fe80::20c:29ff:fefc:1868
BGP connection: shared network
Last Reset: 02:43:42, due to BGP Notification sent
Notification Error Message: (CeaseUnspecified Error Subcode)
```

To verify the networks learned from neighbors or a specific network:

```
# get router info6 bgp network
VRF 0 BGP table version is 3, local router ID is 13.13.13.13
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*> ::/0             2001:db8:d0c:6::2(fe80::20c:29ff:febc:eec2)
                                        0                      0        0 64510 ? <-/1>
*> 64:ff9b::/96     2001:db8:d0c:6::2(fe80::20c:29ff:febc:eec2)
                                        0                      0        0 64510 ? <-/1>
*  2001:db8:d0c:6::/64
                    2001:db8:d0c:6::2(fe80::20c:29ff:febc:eec2)
                                        0                      0        0 64510 i <-/->
*>                                                  100  32768        0 i <-/1>

Total number of prefixes 3
```

To verify the routing table:

```
# get router info6 routing-table
IPv6 Routing Table
Codes: K - kernel route, C - connected, S - static, R - RIP, O - OSPF,
       IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, B - BGP, V - BGP VPNv6
       * - candidate default

Timers: Uptime

Routing table for VRF=0
B*      ::/0 [20/0] via fe80::20c:29ff:febc:eec2, port3, 02:45:56, [1024/0]
C       ::1/128 via ::, root, 03:45:04
B       64:ff9b::/96 [20/0] via fe80::20c:29ff:febc:eec2, port3, 02:45:56, [1024/0]
C       2001:db8:d0c:1::/64 via ::, port1, 00:33:21
O       2001:db8:d0c:2::/64 [110/2] via fe80::20c:29ff:fe4d:f81f, port1, 00:33:04, [1024/0]
                            [110/2] via fe80::20c:29ff:fe6b:b2c9, port2, 00:33:04, [1024/0]
C       2001:db8:d0c:3::/64 via ::, port2, 03:45:04
O       2001:db8:d0c:4::/64 [110/2] via fe80::20c:29ff:fe4d:f81f, port1, 00:33:04, [1024/0]
O       2001:db8:d0c:5::/64 [110/2] via fe80::20c:29ff:fe6b:b2c9, port2, 02:51:32, [1024/0]
C       2001:db8:d0c:6::/64 via ::, port3, 03:45:04
```

NPTv6 protocol for IPv6 address translation example

FortiOS adds partial support of the Network Prefix Translation (NPTv6) protocol in RFC6296 for IPv6 address translation, ensuring end-to-end connectivity, address independence, and 1:1 address mapping. It allows the use of private IPv6 addresses internally while translating them to globally routable IPv6 addresses when

communicating with external networks. It operates at the prefix level by translating the network, prefix portion of an IPv6 address while leaving the host information unchanged. This enhances network scalability and facilitates efficient IPv6 network management.

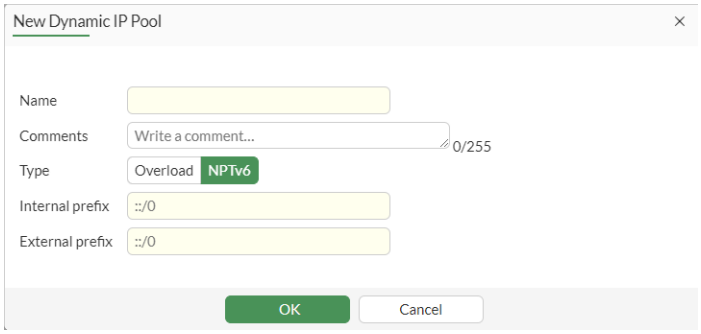
A new NPTv6 type IP pool is introduced which allows an internal prefix of an IPv6 address to be translated to an external prefix. This enables administrators the freedom to use any internal prefix, but efficiently translate these to the prefix provided by their provider that can be routed globally.

To configure NPTv6 in the CLI:

```
config firewall ippool6
    edit <IPv6 IP pool name>
        set type nptv6
        set internal-prefix <internal NPTv6 prefix length>
        set external-prefix <external NPTv6 prefix length>
    next
end
```

To configure NPTv6 in the GUI:

1. Go to *Policy & Objects* > *IP Pools*.
2. Go to the *IPv6 IP Pool* tab.
3. Click *Create new*.
4. Configure the IP pool:
   a. Enter a *Name*.
   b. Set the *Type* to *NPTv6*.
   c. Enter the *Internal prefix* and *External prefix*.
   d. Click *OK*.



How NPTv6 works

NPTv6 works on the principle that network translation is accomplished using a stateless approach, and that it is checksum-neutral at the transport layer. For instance, at the TCP layer, TCP checksums should remain unchanged even after the IP address has been translated.

In brief, TCP checksum is calculated based on a pseudo-header that includes the source and destination IP address. When the source address is translated, it needs to be done in a way that offsets the change to keep the checksum unchanged. This can be accomplished with the following rules:

- For an IPv6 with prefix length of 48-bit or shorter, a 16-bit adjustment can be made on the 4th word (4th hextat) of the IPv6 address which is the subnet field.
- For an IPv6 with prefix length greater than 48-bit, a 16-bit adjustment can be made on the 5th, 6th, 7th or 8th word of the IPv6 address.

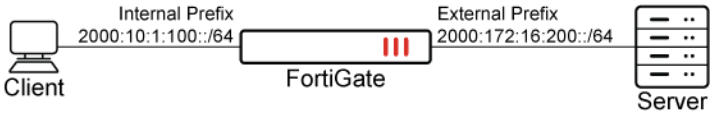
For example, to translate from an internal prefix of fc00:1::/48 to an external prefix of 2003:db8:1::/48, an "adjustment" of 0xce45 has to be made in the 5th word, by adding 0xce45 to the 16-bit subnet field.

Therefore, the internal address fc00:1::1/48 becomes 2003:db8:1:**ce45**::1/48.

For more information, see section RFC6296 Section 3 – NPTv6 Algorithmic Specification.

Example

In the following example, an NPTv6 pool will be created and applied to the firewall policy. The packet IPv6 address translation will then be verifeid.



To configure the NPTv6 IP pool:

1. Configure the NPTv6 IP pool and prefix length:

   The internal and external prefix must be the same length.

```
config firewall ippool6
    edit "NPTv6-ippool6-1"
        set type nptv6
        set internal-prefix 2000:10:1:100::/64
        set external-prefix 2000:172:16:200::/64
    next
end
```

2. Apply the IP pool in the firewall policy:

```
config firewall policy
    edit 2
        set name "NPTv6_policy6-1"
        set srcintf "port2"
        set dstintf "port1"
        set action accept
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set nat enable
        set ippool enable
        set poolname6 "NPTv6-ippool6-1"
```

```
    next
end
```

To verify the address translation:

1. Start an ICMP ping from 2000:10:1:100::41 to 2000:172:16:200::155.
2. Identify the 16-bit subnet field adjustment that is calculated according to the algorithm specified by RFC6296:

```
# diagnose firewall iprope6 list 100004
policy id: 2, group: 00100004, uuid_idx=8162
  action: accept, schedule: always
  cos_fwd=255 cos_rev=255
  flag (08050109): log redir nat master use_src pol_stats
  flag2(00004000): resolve_sso
  flag3(00000080): best-route
  shapers: / per_ip=
  sub_groups: av 00004e20 auth 00000000 split 00000000 misc 00000000
  app_list: 0 ips_view: 0
  vdom_id: 0
  zone_from(1): 8
  zone_to(1): 7
  address_src(1):
      all uuid_idx=8045
  address_dst(1):
      all uuid_idx=8045
  service(1):
      [0:0x0:0/(0,65535)->(0,65535)] helper:auto
  nat(0):
  nat_64(0):
  nptv6(1):
      2000:10:1:100::/64->2000:172:16:200::/64 adjust=FD88
```

3. Verify the session table. The translated IP address shows 2000:172:16:200:**fd88**::41 after applying the external prefix and the adjustment.

```
# diagnose sys session6 list
session6 info: proto=58 proto_state=00 duration=15 expire=45 timeout=0 refresh_dir=both
flags=00000000 sockport=0 socktype=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=log may_dirty npu
statistic(bytes/packets/allow_err): org=1096/2/0 reply=1096/2/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=8->7/7->8
hook=post dir=org act=snat 2000:10:1:100::41:15308->2000:172:16:200::155:128
(2000:172:16:200:fd88::41:15308)
hook=pre dir=reply act=dnat 2000:172:16:200::155:15308->2000:172:16:200:fd88::41:129
(2000:10:1:100::41:15308)
```

```
misc=0 policy_id=2 pol_uuid_idx=8162 auth_info=0 chk_client_info=0 vd=0
serial=00000b86 tos=ff/ff ips_view=0 app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000c00 ofld-O ofld-R
npu info: flag=0x81/0x81, offload=9/9, ips_offload=0/0, epid=128/129, ipid=129/128,
vlan=0x0000/0x0000
vlifid=129/128, vtag_in=0x0000/0x0000 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=4/7, ha_
divert=0/0
total session6: 1
```

4. Take a sniffer trace. The IP address also shows 2000:172:16:200:**fd88**::41.

```
# diagnose sniffer packet any icmp6 4
interfaces=[any]
filters=[icmp6]
6.712522 port2 in 2000:10:1:100::41 -> 2000:172:16:200::155: icmp6: echo request seq 1
[flowlabel 0xc6f01]
6.712555 port1 out 2000:172:16:200:fd88::41 -> 2000:172:16:200::155: icmp6: echo request seq 1
[flowlabel 0xc6f01]
6.712706 port1 in 2000:172:16:200::155 -> 2000:172:16:200:fd88::41: icmp6: echo reply seq 1
[flowlabel 0xa0b59]
6.712718 port2 out 2000:172:16:200::155 -> 2000:10:1:100::41: icmp6: echo reply seq 1
[flowlabel 0xa0b59]
7.721168 port2 in 2000:10:1:100::41 -> 2000:172:16:200::155: icmp6: echo request seq 2
[flowlabel 0xc6f01]
7.721181 port1 out 2000:172:16:200:fd88::41 -> 2000:172:16:200::155: icmp6: echo request seq 2
[flowlabel 0xc6f01]
7.721347 port1 in 2000:172:16:200::155 -> 2000:172:16:200:fd88::41: icmp6: echo reply seq 2
[flowlabel 0xa0b59]
7.721355 port2 out 2000:172:16:200::155 -> 2000:10:1:100::41: icmp6: echo reply seq 2
[flowlabel 0xa0b59]
^C
8 packets received by filter
0 packets dropped by kernel
```

The IPv6 prefix and subnet fields have been translated but the host field remains unchanged.

Enhancing SIP reliability in 464XLAT environments

FortiGate can now disable IP address translation within the SIP payload in 464XLAT environments as needed. This ensures SIP packets with IPv4 information reach user equipment without translation, which addresses an issue where the customer-side translator (CLAT) component does not revert the IPv6 address to IPv4 within the SIP header and body, which leads to RTP connection issues in 464XLAT environments. By preventing unnecessary translation, FortiGate ensures seamless communication and robust connectivity to improve the reliability of SIP-based services in complex network scenarios.

FortiGate only uses this feature as needed. A flag is added to each `voip_session_stream`, and FortiGate uses the flag and other information to identify 464XLAT environments. When a client initiates SIP traffic to a server, such as a `REGISTER` request to a server, and if the VOIP daemon detects an IPv6 layer address, but the SIP payload header contains an IPv4 address, the payload is identified as for a 464XLAT environment, and FortiGate stops NAT46 on the SIP payload.

Scope and limitations

Only one TCP connection is used between each SIP client and the SIP server, and the TCP connection remains up (in other words, doesn't tear down).

All SIP traffic between each SIP client and the SIP server must use the same TCP connection.

Because of the above requirements, SIP pinhole is not created.

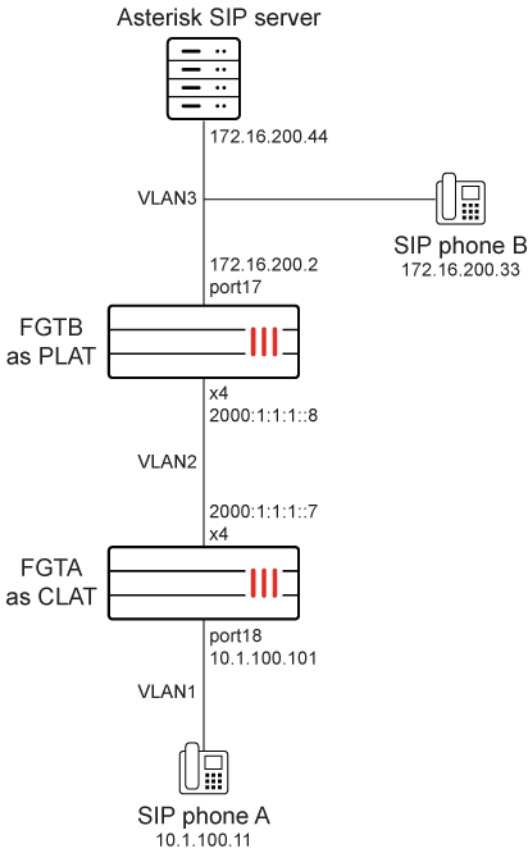
The SIP server must offer its own IP address. The twin case for hnt_464xlat is not supported.

Only SIP/TCP-5060 is supported, and `session-helper` must be configured:

```
config system session-helper
    edit 13
        set name sip
        set protocol 6
        set port 5060
    next
end
```

Example

The example uses the following topology:



1. The CLAT component run insides the user equipment (UE) and performs a one-to-one (IPv4 to IPv6) stateless translation. The CLAT component translates traffic with an IPv4 destination to IPv6. While the IPv4 address is converted to IPv6 in the IP header, the IP information within the payload, including the SIP header and body, remains unchanged (IPv4).
2. FortiGate as PLAT performs many-to-one (IPv6 to IPv4) stateful translation to translate the IP address within the IP header from IPv6 to IPv4. Furthermore, SIP-ALG modifies the IP address within the payload in the SIP header and body by substituting the IPv4 address with a publicly NATed IP address.
3. For traffic returning from SIP server, FortiGate stops NAT46 on the SIP payload and only translates the IPv4 addresses in the IP header into an IPv6 address. Thus, the IP address contained within the SIP header and body remains IPv4, which the user equipment can recognize, and RTP connections can be successfully established.
4. The packet reaches UE, and the CLAT component translates the IP address of the IP header from IPv6 to IPv4. The CLAT component does not function as an ALG and does not modify the IP address within the SIP header and body.
5. The UE receives the packet but cannot establish RTP connection with the IPv6 destination.

To check the SIP calls when phone A (10.1.100.11) and phone B (172.16.200.33) are registering to the SIP server (172.16.200.44):

1. View the SIP proxy SIP calls:

```
# diagnose sys sip-proxy calls
```

```
sip calls
  vdom 1 (vdom1) vrf 0 call 7fc9eb265000
    call-id: QFwxzc1FTIYZQJsQZW3-kKofMxJUcxaB
    txn 7fc9eb250e00 (REGISTER)
      cseq 49308 dir 0 state 5 status 200 expiry 903 HA 0
      i_session: 7fc9eb250000  r_session: 7fc9eb250000
      register: present
      from: sip:1001@172.16.200.44
      to: sip:1001@172.16.200.44
      src: [2000:1:1:1::1:6e0c]:49555
      dst: 172.16.200.44:5060
  vdom 1 (vdom1) vrf 0 call 7fc9eb265000
    call-id: QFwxzc1FTIYZQJsQZW3-kKofMxJUcxaB
    txn 7fc9eb250700 (REGISTER)
      cseq 49307 dir 0 state 7 status 401 expiry 23 HA 0
      i_session: 7fc9eb250000  r_session: 7fc9eb250000
      register: present
      from: sip:1001@172.16.200.44
      to: sip:1001@172.16.200.44
      src: [2000:1:1:1::1:6e0c]:49555
      dst: 172.16.200.44:5060
```

To check the SIP calls and session lists when phone A (10.1.100.1) is calling phone B (172.16.200.33):

1. View the SIP proxy SIP calls:

```
# diagnose sys sip-proxy calls

sip calls
  vdom 1 (vdom1) vrf 0 call 7fc9eb265100
    call-id: mNcOPU8ul76A1ihupFMNU78kePJYEfr3
    txn 7fc9eb251c00 (INVITE)
      cseq 102 dir 1 state 11 status 200 expiry 297 HA 0
      i_session: 7fc9eb250000  r_session: 7fc9eb250000
      register: not-present
      from: sip:1003@172.16.200.44
      to: sip:1001@172.16.200.44
      src: 172.16.200.44:5060
      dst: [2000:1:1:1::1:6e0c]:49555
  vdom 1 (vdom1) vrf 0 call 7fc9eb265100
    call-id: mNcOPU8ul76A1ihupFMNU78kePJYEfr3
    txn 7fc9eb250700 (INVITE)
      cseq 22155 dir 0 state 11 status 200 expiry 296 HA 0
      i_session: 7fc9eb250000  r_session: 7fc9eb250000
      register: not-present
      from: sip:1001@172.16.200.44
      to: sip:1003@172.16.200.44
      src: [2000:1:1:1::1:6e0c]:49555
      dst: 172.16.200.44:5060
  vdom 1 (vdom1) vrf 0 call 7fc9eb265100
    call-id: mNcOPU8ul76A1ihupFMNU78kePJYEfr3
```

```
    txn 7fc9eb251500 (INVITE)
      cseq 22154 dir 0 state 11 status 200 expiry 296 HA 0
      i_session: 7fc9eb250000  r_session: 7fc9eb250000
      register: not-present
      from: sip:1001@172.16.200.44
      to: sip:1003@172.16.200.44
      src: [2000:1:1:1::1:6e0c]:49555
      dst: 172.16.200.44:5060
  vdom 1 (vdom1) vrf 0 call 7fc9eb265000
    call-id: QFwxzc1FTIYZQJsQZW3-kKofMxJUcxaB
    txn 7fc9eb250e00 (REGISTER)
      cseq 49308 dir 0 state 5 status 200 expiry 871 HA 0
      i_session: 7fc9eb250000  r_session: 7fc9eb250000
      register: present
      from: sip:1001@172.16.200.44
      to: sip:1001@172.16.200.44
      src: [2000:1:1:1::1:6e0c]:49555
      dst: 172.16.200.44:5060
```

2. View the IPv6 session list:

```
# diagnose sys session6 list

...
orgin->sink: org pre->post, reply pre->post dev=34->43/43->34
hook=pre dir=org act=dnat 2000:1:1:1::1:6e0c:34143->2000:172:16:200::44:5060
(2000:172:16:200::44:5060)
hook=post dir=reply act=snat 2000:172:16:200::44:5060->2000:1:1:1::1:6e0c:34143
(2000:172:16:200::44:5060)
peer=172.16.200.6:34143->172.16.200.44:5060 naf=1
hook=pre dir=org act=noop 172.16.200.6:34143->172.16.200.44:5060(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.200.44:5060->172.16.200.6:34143(0.0.0.0:0)
...
orgin->sink: org pre->post, reply pre->post dev=34->43/43->0
hook=post dir=org act=noop 2000:1:1:1::1:6e0c:9117->2000:172:16:200::44:64446(:::0)
hook=pre dir=reply act=noop 2000:172:16:200::44:64446->2000:1:1:1::1:6e0c:9117(:::0)
peer=172.16.200.6:64444->172.16.200.44:18764 naf=1
hook=pre dir=org act=noop 172.16.200.6:64444->172.16.200.44:18764(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.200.44:18764->172.16.200.6:64444(0.0.0.0:0)
...
orgin->sink: org pre->post, reply pre->post dev=34->43/43->34
hook=pre dir=org act=dnat 2000:1:1:1::1:6e0c:9117->2000:172:16:200::33:64448
(2000:172:16:200::33:64448)
hook=post dir=reply act=snat 2000:172:16:200::33:64448->2000:1:1:1::1:6e0c:9117
(2000:172:16:200::33:64448)
peer=172.16.200.6:9117->172.16.200.33:64448 naf=1
hook=pre dir=org act=noop 172.16.200.6:9117->172.16.200.33:64448(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.200.33:64448->172.16.200.6:9117(0.0.0.0:0)
...
orgin->sink: org pre->post, reply pre->post dev=34->43/43->34
hook=post dir=org act=noop 2000:1:1:1::1:6e0c:9118->2000:172:16:200::44:64447(:::0)
hook=pre dir=reply act=noop 2000:172:16:200::44:64447->2000:1:1:1::1:6e0c:9118(:::0)
```

```
peer=172.16.200.6:64445->172.16.200.44:18765 naf=1
hook=pre dir=org act=noop 172.16.200.6:64445->172.16.200.44:18765(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.200.44:18765->172.16.200.6:64445(0.0.0.0:0)
...
orgin->sink: org pre->post, reply pre->post dev=34->43/43->34
hook=pre dir=org act=dnat 2000:1:1:1::1:6e0c:9118->2000:172:16:200::33:64449
(2000:172:16:200::33:64449)
hook=post dir=reply act=snat 2000:172:16:200::33:64449->2000:1:1:1::1:6e0c:9118
(2000:172:16:200::33:64449)
peer=172.16.200.6:9118->172.16.200.33:64449 naf=1
hook=pre dir=org act=noop 172.16.200.6:9118->172.16.200.33:64449(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.200.33:64449->172.16.200.6:9118(0.0.0.0:0)
...
```

3. View the IPv6 expectation session list:

```
# diagnose sys session6 list expectation

...
orgin->sink: org pre->post, reply pre->post dev=34->0/43->0
hook=post dir=org act=noop 2000:1:1:1::1:6e0c:0->2000:172:16:200::44:64448(:::0)
peer=172.16.200.6:64444->172.16.200.33:4000 naf=1
...
orgin->sink: org pre->post, reply pre->post dev=34->0/43->0
hook=post dir=org act=noop 2000:1:1:1::1:6e0c:0->2000:172:16:200::44:64448(:::0)
peer=172.16.200.6:64444->172.16.200.33:4000 naf=1
...
orgin->sink: org pre->post, reply pre->post dev=34->0/43->0
hook=post dir=org act=noop 2000:1:1:1::1:6e0c:0->2000:172:16:200::44:64449(:::0)
peer=172.16.200.6:64445->172.16.200.33:4001 naf=1
...
orgin->sink: org pre->post, reply pre->post dev=34->0/43->0
hook=post dir=org act=noop 2000:1:1:1::1:6e0c:0->2000:172:16:200::44:64449(:::0)
peer=172.16.200.6:64445->172.16.200.33:4001 naf=1
...
```

4. View the IPv4 session list:

```
# diagnose sys session list

...
orgin->sink: org pre->post, reply pre->post dev=43->23/23->43 gwy=172.16.200.33/172.16.200.6
hook=pre dir=org act=noop 172.16.200.6:9117->172.16.200.33:64448(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.200.33:64448->172.16.200.6:9117(0.0.0.0:0)
peer=2000:172:16:200::33:64448->2000:1:1:1::1:6e0c:9117 naf=2
hook=pre dir=org act=dnat 2000:1:1:1::1:6e0c:9117->2000:172:16:200::33:64448
(2000:172:16:200::33:64448)
hook=post dir=reply act=snat 2000:172:16:200::33:64448->2000:1:1:1::1:6e0c:9117
(2000:172:16:200::33:64448)
...
orgin->sink: org pre->post, reply pre->post dev=43->23/23->43 gwy=172.16.200.33/172.16.200.6
hook=pre dir=org act=noop 172.16.200.6:9118->172.16.200.33:64449(0.0.0.0:0)
```

```
hook=post dir=reply act=noop 172.16.200.33:64449->172.16.200.6:9118(0.0.0.0:0)
peer=2000:172:16:200::33:64449->2000:1:1:1::1:6e0c:9118 naf=2
hook=pre dir=org act=dnat 2000:1:1:1::1:6e0c:9118->2000:172:16:200::33:64449
(2000:172:16:200::33:64449)
hook=post dir=reply act=snat 2000:172:16:200::33:64449->2000:1:1:1::1:6e0c:9118
(2000:172:16:200::33:64449)
...
orgin->sink: org pre->post, reply pre->post dev=43->23/23->43 gwy=172.16.200.44/172.16.200.6
hook=pre dir=org act=noop 172.16.200.6:34143->172.16.200.44:5060(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.200.44:5060->172.16.200.6:34143(0.0.0.0:0)
peer=2000:172:16:200::44:5060->2000:1:1:1::1:6e0c:34143 naf=2
hook=pre dir=org act=dnat 2000:1:1:1::1:6e0c:34143->2000:172:16:200::44:5060
(2000:172:16:200::44:5060)
hook=post dir=reply act=snat 2000:172:16:200::44:5060->2000:1:1:1::1:6e0c:34143
(2000:172:16:200::44:5060)
...
orgin->sink: org pre->post, reply pre->post dev=43->23/23->0 gwy=172.16.200.44/0.0.0.0
hook=pre dir=org act=noop 172.16.200.6:64444->172.16.200.44:18764(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.200.44:18764->172.16.200.6:64444(0.0.0.0:0)
peer=2000:172:16:200::44:64446->2000:1:1:1::1:6e0c:9117 naf=2
hook=post dir=org act=noop 2000:1:1:1::1:6e0c:9117->2000:172:16:200::44:64446(:::0)
hook=pre dir=reply act=noop 2000:172:16:200::44:64446->2000:1:1:1::1:6e0c:9117(:::0)
...
orgin->sink: org pre->post, reply pre->post dev=43->23/23->43 gwy=172.16.200.44/172.16.200.6
hook=pre dir=org act=noop 172.16.200.6:64445->172.16.200.44:18765(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.200.44:18765->172.16.200.6:64445(0.0.0.0:0)
peer=2000:172:16:200::44:64447->2000:1:1:1::1:6e0c:9118 naf=2
hook=post dir=org act=noop 2000:1:1:1::1:6e0c:9118->2000:172:16:200::44:64447(:::0)
hook=pre dir=reply act=noop 2000:172:16:200::44:64447->2000:1:1:1::1:6e0c:9118(:::0)
...
```

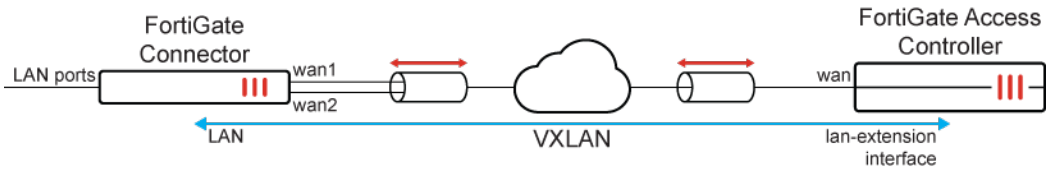
5. View the IPv4 expectation session list:

```
# diagnose sys session list expectation

...
orgin->sink: org pre->post, reply pre->post dev=43->0/23->0 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 172.16.200.6:0->172.16.200.33:4001(0.0.0.0:0)
peer=:::0->:::0 naf=2
...
orgin->sink: org pre->post, reply pre->post dev=43->0/23->0 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 172.16.200.6:0->172.16.200.33:4001(0.0.0.0:0)
peer=:::0->:::0 naf=2
...
orgin->sink: org pre->post, reply pre->post dev=43->0/23->0 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 172.16.200.6:0->172.16.200.33:4000(0.0.0.0:0)
peer=:::0->:::0 naf=2
...
orgin->sink: org pre->post, reply pre->post dev=43->0/23->0 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 172.16.200.6:0->172.16.200.33:4000(0.0.0.0:0)
peer=:::0->:::0 naf=2
...
```

FortiGate LAN extension

LAN extension mode allows a remote FortiGate to provide remote connectivity to a local FortiGate over a backhaul connection.

The remote FortiGate, called the FortiGate Connector, discovers the local FortiGate, called the FortiGate Controller, and forms one or more IPsec tunnels back to the FortiGate Controller. A VXLAN is established over the IPsec tunnels creating an L2 network between the FortiGate Controller and the network behind the FortiGate Connector.



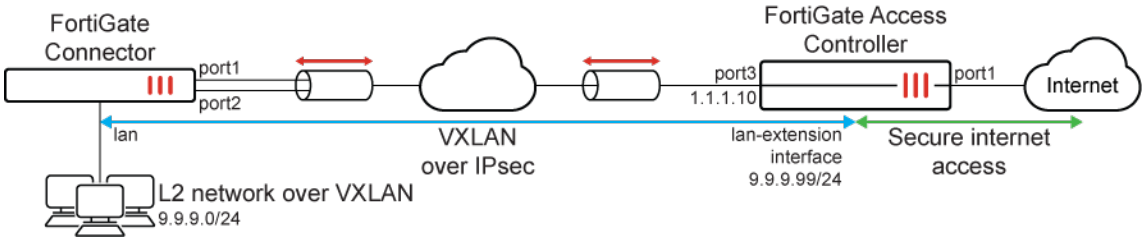
The following topics describe further details about the FortiGate LAN extension feature:

- Example CLI configuration on page 995
- Example GUI configuration on page 1002
- DHCP client mode for inter-VDOM links on page 1007
- FortiGate secure edge to FortiSASE on page 1008
- WiFi access point with internet connectivity on page 1012

Example CLI configuration

In this example, the Controller provides secure internet access to the remote network behind the Connector. The Controller has two WAN connections: an inbound backhaul connection and an outbound internet connection. The Connector has two wired WAN/uplink ports that are connected to the internet.

After the Connector discovers the Controller and is authorized by the Controller, the Controller pushes a FortiGate LAN extension profile to the Connector. The Connector uses the profile configurations to form two IPsec tunnels back to the Controller. Additional VXLAN aggregate interfaces are automatically configured to create an L2 network between the Connector LAN port and a virtual LAN extension interface on the Controller. Clients behind the Connector can then connect to the internet through the Controller that is securing the internet connection.



To discover and authorize the FortiGate Controller:

1. On the FortiGate Controller:
   a. For high-end models (1000 series and higher), enable the FortiExtender setting:

```
config system global
    set fortiextender enable
end
```

   > This command is configured by default on entry-level and mid-range models (900 series and lower).

   b. Enable IPAM and management of LAN extension interface addresses:

```
config system ipam
    set status enable
    set manage-lan-extension-addresses enable
end
```

   c. Enable security fabric connections on port3 to allow the Connector to connect over CAPWAP:

```
config system interface
    edit "port3"
        set vdom "root"
        set ip 1.1.1.10 255.255.255.0
        set allowaccess fabric ping
        set ip-managed-by-fortiipam disable
    next
end
```

   > IPAM is specifically disabled for this interface since a static IP address is desired for this topology.

2. On the FortiGate Connector:
   a. Enable VDOMs:

```
config system global
    set vdom-mode multi-vdom
end
```

   You will be logged out of the device when VDOM mode is enabled.

   b. For high-end models (1000 series and higher), enable the FortiExtender setting in the global VDOM:

```
config global
    config system global
        set fortiextender enable
    end
end
```

> This command is configured by default on entry-level and mid-range models (900 series and lower).

c. Create the lan-ext VDOM while setting the VDOM type to LAN extension (making the VDOM act as a FortiExtender in LAN extension mode), and add the Controller IP address:

```
config vdom
    edit lan-ext
        config system settings
            set vdom-type lan-extension
            set lan-extension-controller-addr "1.1.1.10"
            set ike-port 4500
        end
    next
end
```

d. Configure port1 and port2 to access the Controller:

```
config system interface
    edit "port1"
        set vdom "lan-ext"
        set ip 5.5.5.1 255.255.255.0
        set allowaccess ping fabric
        set type physical
        set lldp-reception enable
        set role wan
    next
    edit "port2"
        set vdom "lan-ext"
        set ip 6.6.6.1 255.255.255.0
        set allowaccess ping fabric
        set type physical
        set lldp-reception enable
        set role wan
    next
end
```

3. On the FortiGate Controller:

a. Extension controller configurations are automatically initialized:

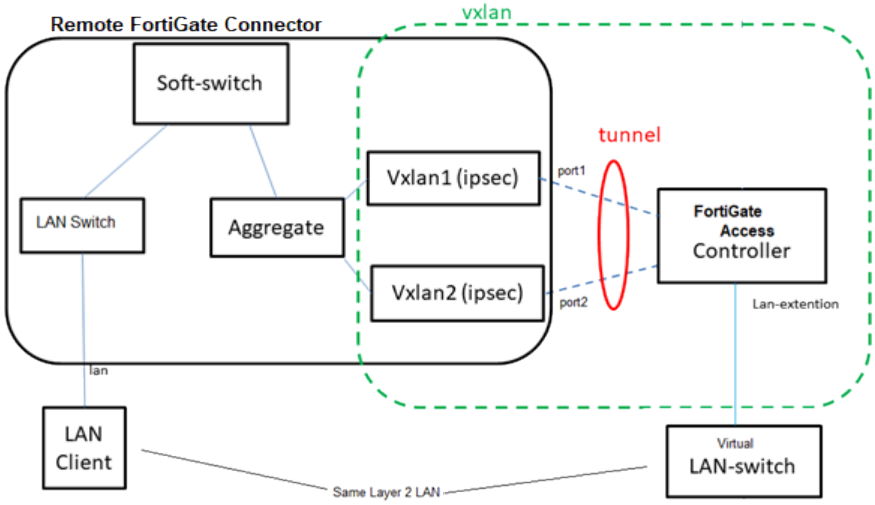
```
config extension-controller fortigate-profile
    edit "FGCONN-lanext-default"
        set id 0
        config lan-extension
            set ipsec-tunnel "fg-ipsec-XdSpij"
            set backhaul-interface "port3"
        end
    next
end
```

```
config extension-controller fortigate
    edit "FGT60F0000000001"
        set id "FG5H1E0000000001"
        set device-id 0
        set profile "FGCONN-lanext-default"
    next
end
```

b. Authorize the Connector:

```
config extension-controller fortigate
    edit "FGT60F0000000001"
        set authorized enable
    next
end
```

4. After the FortiGate Connector has been authorized, the Controller pushes the IPsec tunnel configuration to the Connector, forcing it to establish the tunnel and form the VXLAN mechanism.



The VXLANs are built on the IPsec tunnels between the Connector and Controller. The VXLAN interfaces are aggregated for load balancing and redundancy. A softswitch combines the aggregate interface with the local LAN ports, allowing the LAN ports to be part of the VXLAN. This combines the local LAN ports with the virtual LAN extension interface on the FortiGate Controller.

a. The Connector receives the IPsec configurations from the Controller, and automatically creates tunnels for each uplink:

```
config vpn ipsec phase1-interface
    edit "ul-port1"
        set interface "port1"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set localid "peerid-T4YLv2rp62SU6JhoCPIv02MzjLtS7P5HlxRER1Qpi6O9ZsAsbPSpvoiE"
        set dpd on-idle
        set comments "[FGCONN] Do NOT edit. Automatically generated by extension
```

```
controller."
        set remote-gw 1.1.1.10
        set psksecret ******
    next
    edit "ul-port2"
        set interface "port2"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set localid "peerid-T4YLv2rp62SU6JhoCPIv02MzjLtS7P5HlxRER1Qpi6O9ZsAsbPSpvoiE"
        set dpd on-idle
        set comments "[FGCONN] Do NOT edit. Automatically generated by extension
controller."
        set remote-gw 1.1.1.10
        set psksecret ******
    next
end
```

b. VXLAN interfaces are automatically configured and formed over each tunnel:

```
config system vxlan
    edit "vx-port1"
        set interface "ul-port1"
        set vni 1
        set dstport 9999
        set remote-ip "10.252.0.1"
    next
    edit "vx-port2"
        set interface "ul-port2"
        set vni 1
        set dstport 9999
        set remote-ip "10.252.0.1"
    next
end
```

c. An aggregate interface is automatically configured to load balance between the two VXLAN interfaces, using the source MAC and providing link redundancy:

```
config system interface
    edit "le-agg-link"
        set vdom "lan-ext"
        set type aggregate
        set member "vx-port1" "vx-port2"
        set snmp-index 35
        set lacp-mode static
        set algorithm Source-MAC
    next
end
```

d. The softswitch is automatically configured and bridges the aggregate interface and the local LAN to connect the LAN to the VXLAN bridged L2 network that goes to the FortiGate LAN extension interface:

```
config system switch-interface
    edit "le-switch"
        set vdom "lan-ext"
        set member "le-agg-link" "lan"
    next
end
```

To configure the LAN extension interface and firewall policy on the FortiGate Controller:

1. After the IPsec tunnel is setup and the VXLAN is created over the tunnel, the LAN extension interface is automatically created on the Controller:

```
config system interface
    edit "FGT60F0000000001"
        set vdom "root"
        set ip 192.168.0.254 255.255.255.0
        set allowaccess ping ssh
        set type lan-extension
        set role lan
        set snmp-index 27
        set ip-managed-by-fortiipam enable
        set interface "fg-ipsec-XdSpij"
    next
end
```

   Devices on the remote LAN network will use this IP address as their gateway.
2. Observe that with IPAM enabled on the Controller that the DHCP server settings have been automatically configured:

```
config system dhcp server
    edit 3
        set dns-service default
        set default-gateway 9.9.9.99
        set netmask 255.255.255.0
        set interface "FGT60F0000000001"
        config ip-range
            edit 1
                set start-ip 9.9.9.100
                set end-ip 9.9.9.254
            next
        end
        set dhcp-settings-from-fortiipam enable
        config exclude-range
            edit 1
                set start-ip 9.9.9.254
                set end-ip 9.9.9.254
            next
        end
    next
end
```

3. Configure the firewall policy to allow traffic from the LAN extension interface to the WAN interface (port1):

```
config firewall policy
    edit "2"
        set name "lan-ext"
        set srcintf "FGT60F0000000001"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

Optionally, security profiles and other settings can be configured.

The policy allows remote LAN clients to access the internet through the backhaul channel. Clients in the remote LAN behind the Connector receive an IP address over DHCP and access the internet securely through the Controller.

To verify the FortiGate LAN extension configuration:

1. Verify the IPsec tunnels' phase 1 and phase 2 negotiations on the Controller and Connector:

```
# diagnose ike vpn gateway list
# diagnose vpn tunnel list
```

2. Verify the VXLAN tunnel forwarding database list on the Controller and Connector:

```
# diagnose sys vxlan fdb list
```

3. Verify the DHCP server lease list on the Controller:

```
# execute dhcp lease-list
```

4. Verify the LAN extension session information on the Controller:

```
Controller-FGT # get extender session-info
Total 1 WS sessions, 0 AS sessions:
fg connector sessions:
FGT60F0000000001 : 1.1.1.10:5246 (dport 65535)lan-extension,  running,  install, data-enable,
refcnt 6, miss_echos -1, up-time 1554 secs, change 1
extender sessions:
```

In this example, the Connector is in a working state.

5. Verify the LAN extension status on the Connector:*

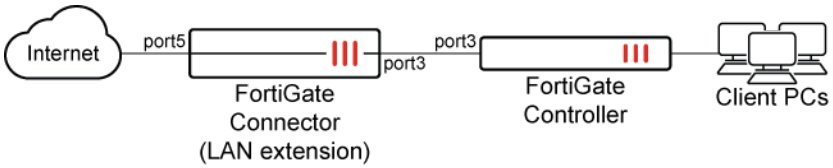
```
Connector-FGT (lan-ext) # get extender lanextension-vdom-status

Control-Channel:
        controller ip: 1.1.1.0
        controller port: 5246
        controller name: FG5H1E0000000001
```

```
        missed echo: 0
        up time(seconds): 29483
        status: EXTWS_RUN
Data-Channel:
uplink [0]: port1
        IPsec tunnel ul-port1
        VxLAN interface vx-port1
uplink [1]: port2
        IPsec tunnel ul-port2
        VxLAN interface vx-port2
downlink [0]: lan
```

In this example, the Connector is in a working state.

Example GUI configuration

In this example, the FortiGate Controller has CAPWAP access allowed on port3. The FortiGate Connector has its WAN port3 connected to the FortiGate Controller, and LAN port5 is connected to the client PCs.



To configure the FortiGate LAN extension:

1. On the FortiGate Controller, enable the FortiExtender setting. For high-end models (1000 series and higher) and VM models, enter:

```
config system global
    set fortiextender enable
end
```

> This command is configured by default on entry-level and mid-range models (900 series and lower).

2. On the FortiGate Controller, configure the port3 settings:
   a. Go to *Network* > *Interfaces* and edit *port3*.
   b. Set the *Addressing mode* to *IPAM*.
   c. In this example, IPAM is not enabled yet. Click *Enable IPAM*. The *IPAM Settings* pane opens.
   d. Set the *Status* to *Enabled*, enable *FortiExtender LAN extensions*, then click *OK*.
   e. In the *Administrative Access* > *IPv4* section, select *Security Fabric Connection* to enable CAPWAP on the interface.
   f. Enable *DHCP Server*.
   g. Click *OK*.
3. On the FortiGate Connector, enable VDOMs:

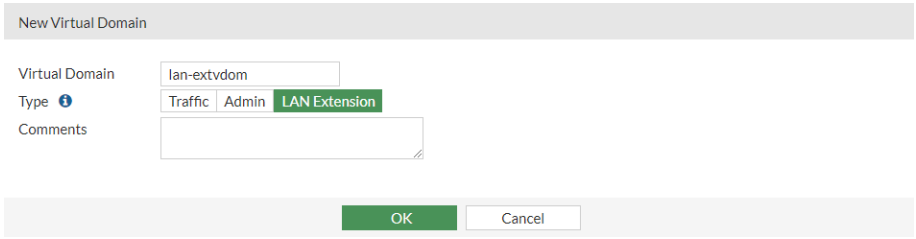
a. Go to *System* > *Settings* > *General*.
b. In the *System Operation Settings* sections, enable *Virtual Domains*.
c. Click *OK*. You will be logged out of the device when VDOM mode is enabled.

4. On the FortiGate Connector, enable the FortiExtender setting. For high-end models (1000 series and higher) and VM models, enter:

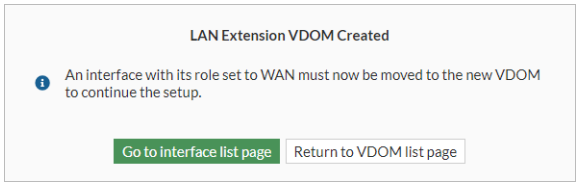
```
config system global
    set fortiextender enable
end
```

> This command is configured by default on entry-level and mid-range models (900 series and lower).

5. On the FortiGate Connector, configure the LAN extension VDOM:
   a. Go to *System* > *VDOM* and click *Create New*.
   b. Enter a name (*lan-extvdom*) and set the *Type* to *LAN Extension*.

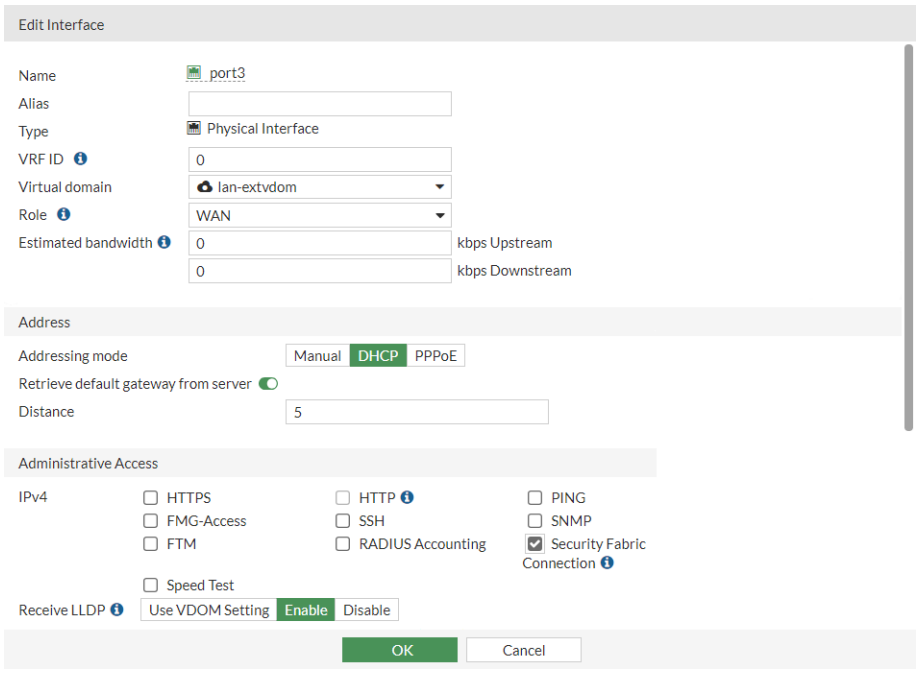


   c. Click *OK*. The *LAN Extension VDOM Created* prompt appears.



   d. Click *Go to interface list page* to assign a role (LAN or WAN) and the LAN extension VDOM.

6. On the FortiGate Connector, edit port3:
   a. Set the *Role* to *WAN*.
   b. Set the *Virtual domain* to *lan-extvdom*.

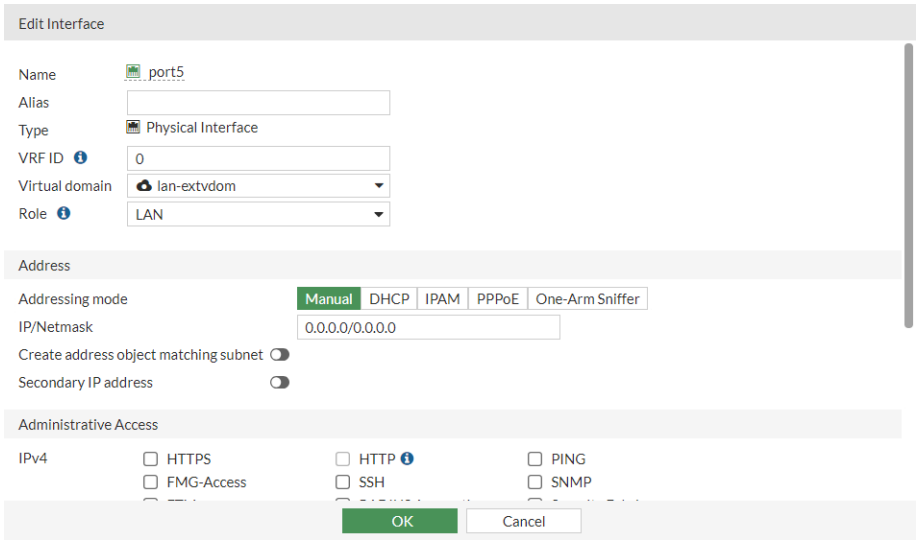


c. Click *OK*.

7. On the FortiGate Connector, edit port5:

a. Set the *Role* to *LAN*.

b. Set the *Virtual domain* to *lan-extvdom*.

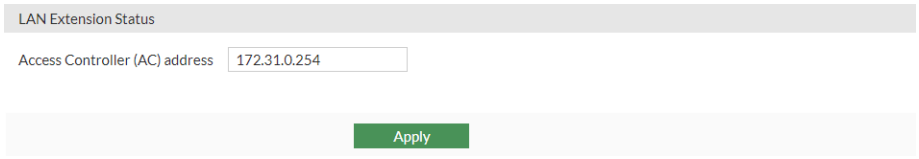


The addressing mode is set to *Manual* and the *IP/Netmask* set to 0.0.0.0/0.0.0.0 because port5 will become part of the *le-switch* software switch, which has its own IP address already assigned.

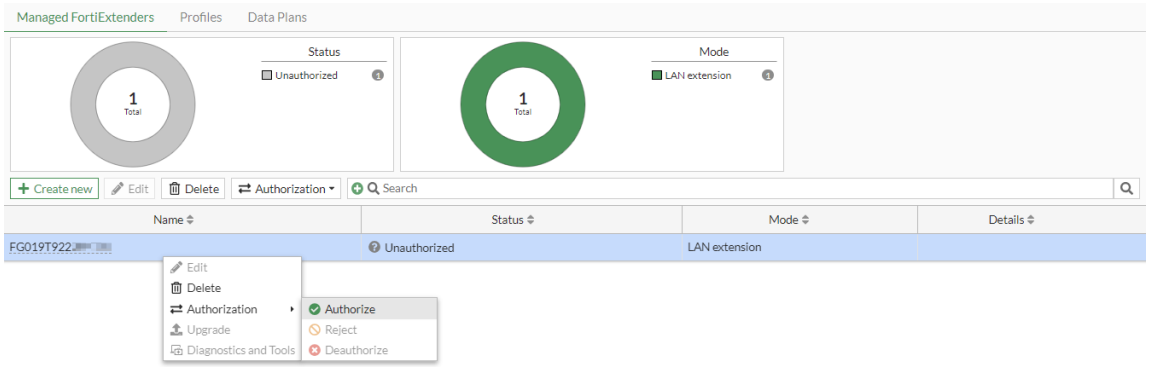
c. Click *OK*.

> Setting the *Role* to *LAN* will automatically add this interface to the *le-switch* LAN extension software switch, which forms an L2 network with the VXLAN.
>
> To add more LAN ports to *le-switch* automatically, set the *Role* to *LAN* for other desired LAN ports.

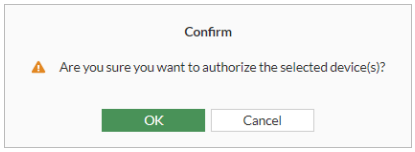
8. On the FortiGate Connector, select the LAN extension VDOM, and enter the IP address of the FortiGate controller:
   a. Go to *Network* > *LAN Extension*.
   b. Set the *Access Controller (AC) address* to the IP address of port3 on the FortiGate Controller. In this example, use *172.31.0.254*.



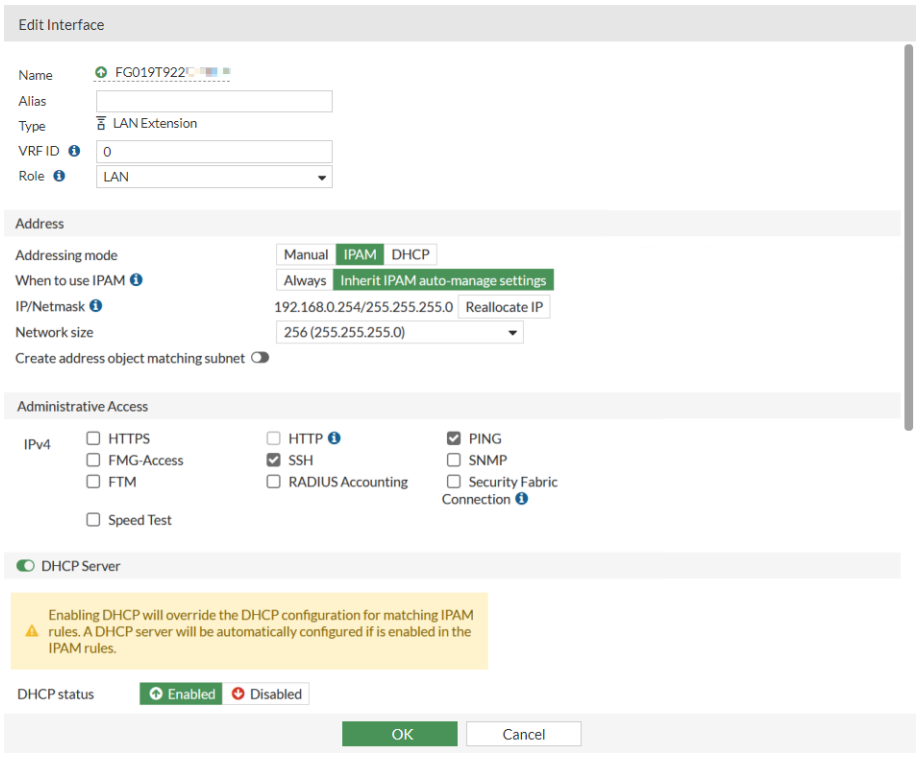
   c. Click *Apply*.
9. On the FortiGate Controller, enable the FortiExtender feature visibility in the GUI, and authorize the FortiGate connector:
   a. Go to *System* > *Feature Visibility*. In the *Additional Features* section, enable *FortiExtender* and click *Apply*.
   b. Go to *Network* > *FortiExtenders* and select the *Managed FortiExtenders* tab.
   c. Select the device, then right-click and select *Authorization* > *Authorize*.



   d. Click *OK* to authorize the device.



10. On the FortiGate Controller, configure the LAN extension interface:
    a. Go to *Network* > *Interfaces* and edit the LAN extension interface.
    b. Set the *Addressing mode* to *IPAM* and set *When to use IPAM* to *Inherit IPAM auto-manage settings* (default).
    c. Enable *DHCP Server*, and configure the settings as needed (see DHCP servers and relays on page 550 for more information).



d. Click *OK*.

11. On the FortiGate Controller, configure the default gateway if a static WAN IP configuration is used:

a. Go to *Network* > *Static Routes* and edit the default gateway settings to specify the correct internet gateway address and WAN interface.

b. Click *OK*.

12. On the FortiGate Controller, configure the firewall policy to allow traffic to pass:

a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.

b. Set the *Incoming Interface* to the LAN extension interface.

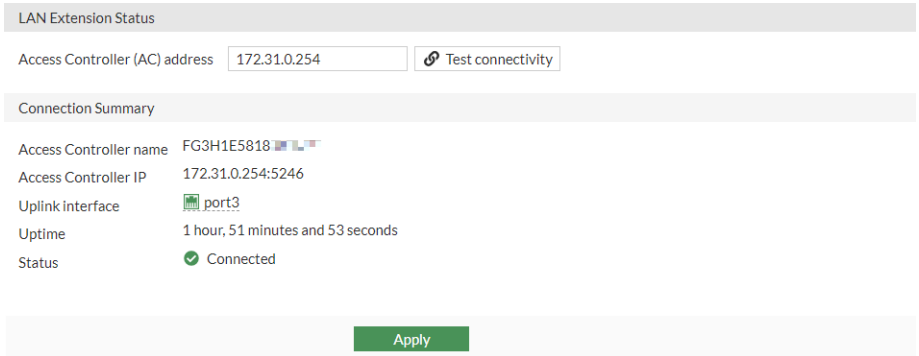
c. Configure the other settings as needed.

d. Click *OK*.

13. On the FortiGate Connector, verify that the LAN extension is connected:

a. Go to *Network* > *LAN Extension*.

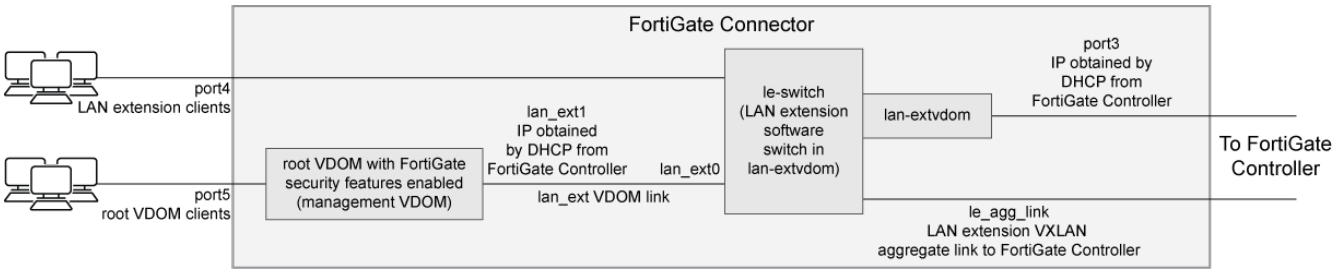
b. Verify that the *Status* is *Connected*.



DHCP client mode for inter-VDOM links

Continuing with the same configuration as Example GUI configuration on page 1002, a new VDOM named lan-extvdom was created on the FortiGate Connector and its type was set to LAN extension. This configuration allows the VDOM to function as a FortiExtender in LAN extension mode. However, it should be noted that this configuration results in the loss of FortiGate security features on that VDOM. For users who wish to utilize the security features of the FortiGate locally on the FortiGate Connector, another VDOM, such as the root VDOM, can be used.

Once the DHCP server is enabled on the FortiGate Controller (as shown in step 2 of Example GUI configuration on page 1002), an inter-VDOM link belonging to another VDOM (in this case, the root VDOM) can receive an IP address by DHCP from the FortiGate Controller.



In this topology, the DHCP clients on the FortiGate Connector interact with the different DHCP servers on the FortiGate Controller.

- The port3 IP address is obtained by DHCP from the FortiGate Controller DHCP server on the port3 connected interface.
- The lan_ext1 IP address is obtained by DHCP from the FortiGate Controller DHCP server on the LAN extension interface.

To configure DHCP client mode on the inter-VDOM link on the FortiGate Connector:

1. Add the VDOM link with an Ethernet type:

```
config system vdom-link
    edit "lan-extvdom"
        set type ethernet
    next
end
```

2. Configure the VDOM link interfaces:

```
config system interface
    edit "lan_ext0"
        set vdom "lan-extvdom"
        set role lan
    next
    edit "lan_ext1"
        set vdom "root"
        set mode dhcp
```

```
    next
end
```

Since lan_ext0 has its role set to `lan`, this interface is added to the le-switch software switch in the lan-extvdom VDOM. This software switch provides network connectivity to the LAN extension clients (in Example GUI configuration on page 1002) and the root VDOM clients (in this example) through the FortiGate Connector LAN extension VXLAN aggregate link.

3. Verify that the lan_ext1 interface obtained an IP address from the FortiGate Controller (the client IP address for the lan_ext1 VDOM link is from the same 192.168.0.0/24 subnet in step 10c of Example GUI configuration on page 1002):

```
Connector-FGT (lan-ext) # diagnose ip address list | grep lan_ext1
IP=192.168.0.1->192.168.0.1/255.255.255.0 index=30 devname=lan_ext1
```

FortiGate secure edge to FortiSASE

VDOM configuration for the FortiGate LAN extension has been simplified. When you configure the FortiGate LAN extension VDOM, FortiOS automatically configures a VDOM link between a traffic VDOM, which is by default the root VDOM, and the LAN extension VDOM.

After connecting to the FortiGate Controller, the following settings are automatically configured on the FortiGate Connector:

- VDOM link interface in the LAN extension VDOM is a part of the LAN extension software switch.
- VDOM link interface in the traffic VDOM is dynamically assigned an IP address obtained through the FortiGate Controller.

The traffic VDOM can be used to:

- Apply application steering to the local internet connection or to FortiGate Controller network (FortiSASE) using SD-WAN.
- Apply local security features for traffic egressing the local internet connection, such as antivirus, intrusion prevention security (IPS), application control, and web filtering, by creating a firewall policy with the destination interface configured as either the FortiGate WAN interface or an SD-WAN zone with the FortiGate WAN interface as a member.

Example

This example demonstrates how to configure the FortiGate Connector to connect to FortiSASE as the FortiGate Controller. In FortiSASE, the FortiGate Connector is more commonly known as the FortiGate secure edge.

To configure the FortiGate Connector using the CLI:

1. Enable multi-VDOM mode from the CLI:

```
config system global
    set vdom-mode multi-vdom
end
```

2. Verify that the FortiExtender setting is enabled in the global VDOM:

```
# config global
# show full system global | grep fortiextender -f
…
    set fortiextender enable
…
```

3. Create a new LAN extension VDOM with the LAN extension controller address as the FortiSASE domain name.

   See Connecting FortiGate to FortiSASE using GUI and CLI for details on how to find the FortiSASE domain name.

   In this example, the VDOM name is `ext`, and the FortiSASE domain name is `turbo-a1p0hv3p.edge.prod.fortisase.com`.

```
config vdom
    edit ext
        config system settings
            set vdom-type lan-extension
            set lan-extension-controller-addr turbo-a1p0hv3p.edge.prod.fortisase.com
            set ike-port 4500
        end
    next
end
```

4. Move interfaces from the root VDOM to the new LAN extension VDOM, and set the appropriate WAN and LAN roles.
   - Before moving an interface to a new VDOM, delete all references, such as firewall policies or firewall objects. See Finding object dependencies on page 4987.
   - If interfaces are already part of a hardware switch, remove them from the hardware switch to make them available for the new VDOM. See Hardware switch on page 280.

   In this example from the global VDOM, the *wan1* and *internal1* interfaces are moved to the LAN extension VDOM named *ext*, and their roles are set appropriately as *WAN* and *LAN*.

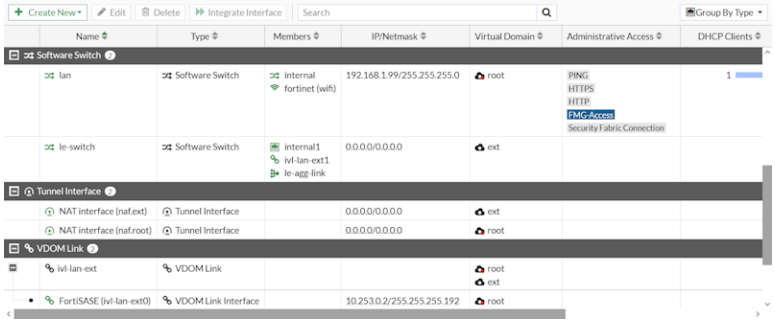
   a. From the GUI, go to the *Global* VDOM.
   b. Go to *Network* > *Interfaces* and edit the *wan1* interface:
      i. Set the *Role* to *WAN*.
      ii. Set the *Virtual domain* to *ext*.
      iii. Click *OK*.
   c. Go to *Network* > *Interfaces* and edit the *internal1* interface:
      i. Set the *Role* to *LAN*.
      ii. Set the *Virtual domain* to *ext*.
      iii. Click *OK*.
5. For the WAN interface within the LAN extension VDOM, edit the interface and ensure that Security Fabric connections are allowed:
   a. From the GUI, go to the *Global* VDOM.
   b. Go to *Network* > *Interfaces* and edit the *WAN1* interface.
   c. Under *Administrative Access*, ensure *PING* and *Security Fabric Connection* are selected.
   d. Click *OK*.

This configuration assumes that the WAN and LAN interfaces are already configured with static IP addresses or configured to use DHCP accordingly.

6. (Optional) If your LAN extension VDOM is not configured as the management VDOM, and you require a custom DNS server to resolve the FortiGate Controller hostname, then you must configure the VDOM DNS settings within the VDOM:

```
config vdom
    edit ext
        config system vdom-dns
            set vdom-dns enable
            set primary 1.2.3.4
            set secondary 2.3.4.5
        end
    next
end
```

7. In FortiSASE, authorize the FortiGate as a LAN extension in the *Edge Devices > FortiGates* page. See Authorizing a FortiGate.
8. In the LAN extension VDOM, in *Network > LAN Extension* observe that the *Connection Summary* shows values for the *Access Controller Name*, *Access Controller IP*, and *Connected* status. These all indicate the LAN extension VDOM established a successful connection with FortiSASE.
9. After the LAN extension VDOM connects to FortiSASE, observe from the *Global* VDOM under *Network > Interfaces*:
   - A VDOM link *ivl-lan-ext* is created.
   - The VDOM link interface in the LAN extension VDOM (*ivl-lan-ext1*) is part of the *le-switch* LAN extension software switch. Network connectivity to the FortiGate Controller (that is, to FortiSASE) is achieved through the software switch.
   - The VDOM link interface in the traffic (*root*) VDOM (*ivl-lan-ext0*) has obtained an IP address dynamically from the FortiGate Controller.



10. Within the *root* VDOM, create a firewall policy with *ivl-lan-ext0* as the destination and *lan* as the source within the traffic VDOM to allow local traffic in the IP address range for DHCP clients in the LAN subnet (LAN-DHCP-RANGE) from the FortiGate Connector to access the internet through the FortiGate Controller (FortiSASE).
    a. From the GUI, go to the *root* VDOM.
    b. Go to *Policy & Objects > Firewall Policy*.
    c. Click *Create New*.
    d. Create a new policy with the following settings:

Name	traffic-VDOM-to-FortiSASE
Incoming Interface	lan
Outgoing Interface	ivl-lan-ext0
Source	LAN-DHCP-RANGE
Destination	all
Schedule	always
Service	ALL
Action	ACCEPT
NAT	Enabled

e. Click *OK*.

a. Within the *root* VDOM, create a firewall policy with *wan2* (second ISP link on FortiGate with proper routing already set up) as the destination and *lan* as the source within the traffic VDOM to allow local traffic in the IP address range for static IP clients in the LAN subnet (LAN-STATIC-RANGE) from the FortiGate Connector to access the internet through the FortiGate Controller (FortiSASE). Security Profiles and SSL certificate inspection are also enabled on this policy.

i. From the GUI, go to the *root* VDOM.

ii. Go to *Policy & Objects > Firewall Policy*.

iii. Click *Create New*.

iv. Create a new policy with the following settings:

Name	traffic-VDOM-to-wan2
Incoming Interface	lan
Outgoing Interface	wan2
Source	LAN-STATIC-RANGE
Destination	all
Schedule	always
Service	ALL
Action	ACCEPT
NAT	Enabled
AntiVirus	g-default
Web filter	g-default
DNS filter	default
Application control	g-default
SSL inspection	certificate-inspection

v. Click *OK*.

WiFi access point with internet connectivity

A FortiGate LAN extension can be configured to allow clients that are connected to a WiFi access point on FortiExtender to have access to the internet. This example describes how to configure a WiFi access point with internet access for a managed FortiExtender. An overview of the configuration steps is provided followed by the details:

1. Create a LAN extension SSID for FortiExtender.
2. Configure a FortiExtender profile.
3. Apply the profile to FortiExtender and authorize the device.
4. Configure the LAN extension interface as a DHCP server.
5. Configure a firewall policy to allow internet access.
6. Verify settings.

Creating a LAN extension SSID for FortiExtender

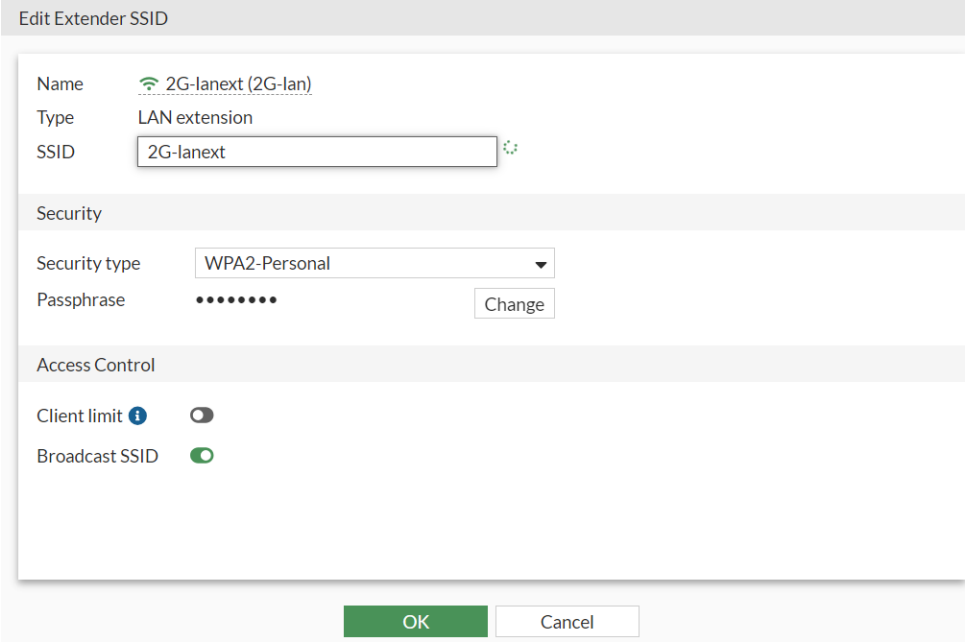
Use the *FortiExtender SSIDs* tab to create a LAN extension SSID for a managed FortiExtender device.

In this The SSID name in this example is *2G-lanext*.

To create a LAN extension SSID for FortiExtender in the GUI:

1. Go to *Network* > *FortiExtenders* > *FortiExtender SSIDs*, and click *Create New*.
2. Set the following options:

Type	Select *LAN extension*.
SSID	Enter a name, such as *2G-lanext*.
Security Type	Select a type of security.
Passphrase	Enter a passphrase.



3. Set the remaining options as desired, and click *OK*.

To create a LAN extension SSID for FortiExtender in the CLI:

```
config extension-controller extender-vap
    edit "2G-lan"
        set type lan-ext-vap
        set ssid "2G-lanext"
        set max-clients 0
        set broadcast-ssid enable
        set security WPA2-Personal
        set passphrase 12345678
    next
end
```

Configuring a FortiExtender profile

Use a FortiExtender profile to define the LAN extension settings, such as the radio band for the LAN extension SSID, to create a WiFi SSID.

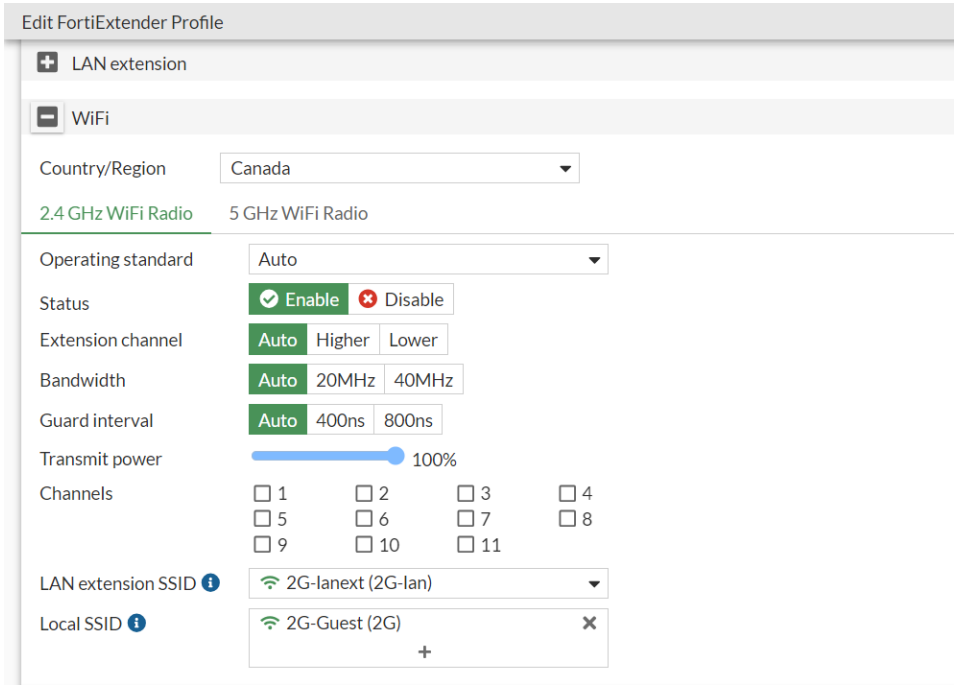
The FortiExtender profile in this example is named *FVA22F-lanext-default*, and the LAN extension named *2G-lanext* is selected to create a WiFi SSID named *2G-lanext (2G-lan)*.

To configure a FortiExtender profile in the GUI:

1. Go to *Network* > *FortiExtenders*, and create a new profile or double-click an existing profile to open it for editing.
2. Expand *WiFi* and set the following options:

2.4 GHz WiFi Radio	Click to display 2 GHz radio band options.

5 GHz WiFi Radio	Click to display 5 GHz radio band options.
LAN extension SSID	Select the LAN extension SSID, for example, *2G-lanext*.
Local SSID	Select an SSID.



3. Set the remaining options as desired, and click *OK*.

To configure a FortiExtender profile in the CLI:

```
config extension-controller extender-profile
    edit "FVA22F-lanext-default"
        set id 5
        set model FVA22F
        set extension lan-extension
        config cellular
            config sms-notification
            end
            config modem1
            end
            config modem2
            end
        end
        config lan-extension
            set ipsec-tunnel "fext-ipsec-g180"
            set backhaul-interface "lan"
            config backhaul
                edit "1"
                    set port wan
                    set role primary
                next
```

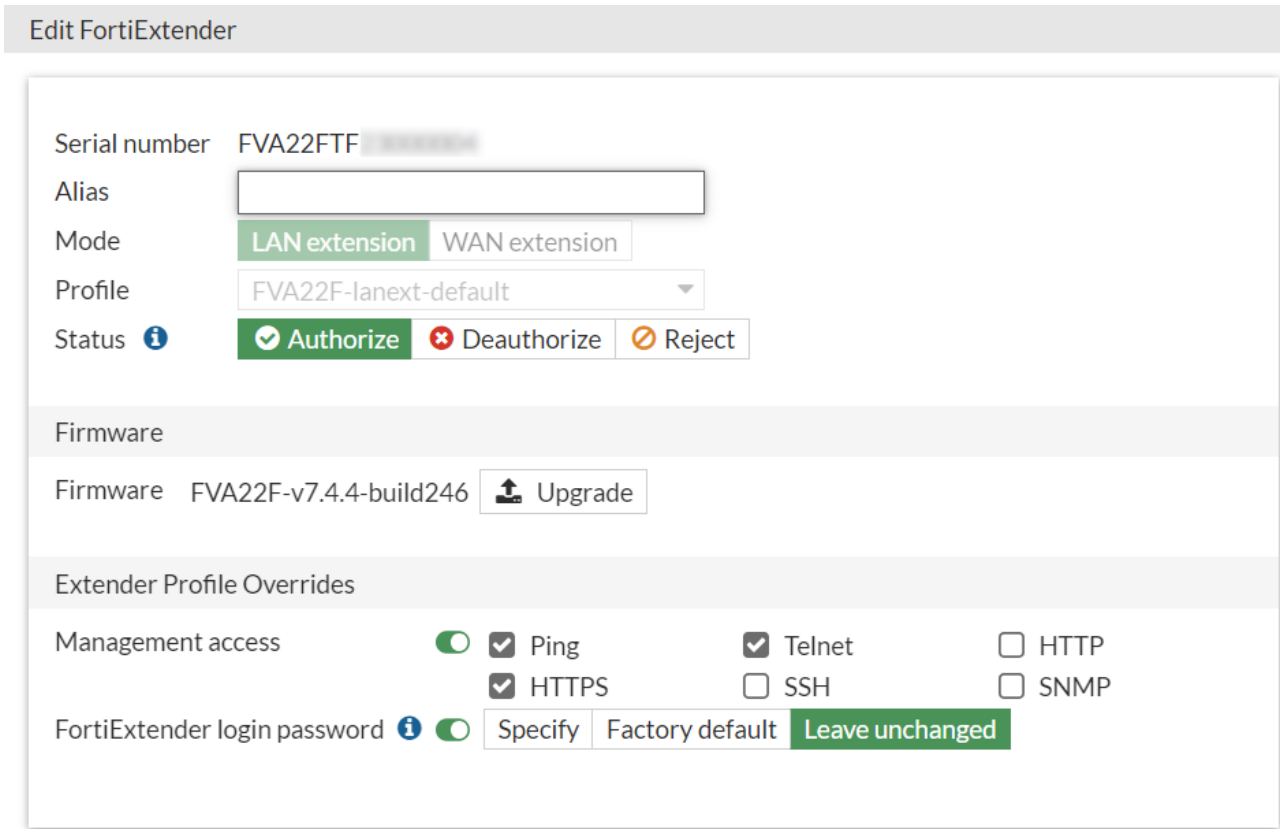
```
                edit "2"
                    set port lte1
                    set role secondary
                next
        config wifi
            set country CA
            config radio-1
                set mode AP
                set band 2.4GHz
                set status enable
                set operating-standard auto
                set lan-ext-vap "2G-lan"
                set local-vaps "2G"
            end
            config radio-2
                set mode AP
                set band 5GHz
                set status enable
                set operating-standard auto
                set local-vaps "5G"
            end
        end
    next
end
```

Apply the profile to FortiExtender and authorize the device

Associate the profile with FortiExtender and authorize the device. The profile settings are applied to the device during the authorization process.

To apply the profile and authorize FortiExtender in the GUI:

1. Go to *Network* > *FortiExtenders*, and double-click a FortiExtender device to open its settings.
2. In the *Profile* list, select the profile.
3. Click *Authorize*.



4. Set the remaining settings as desired, and click *OK*.

To apply the profile and authorize FortiExtender in the CLI:

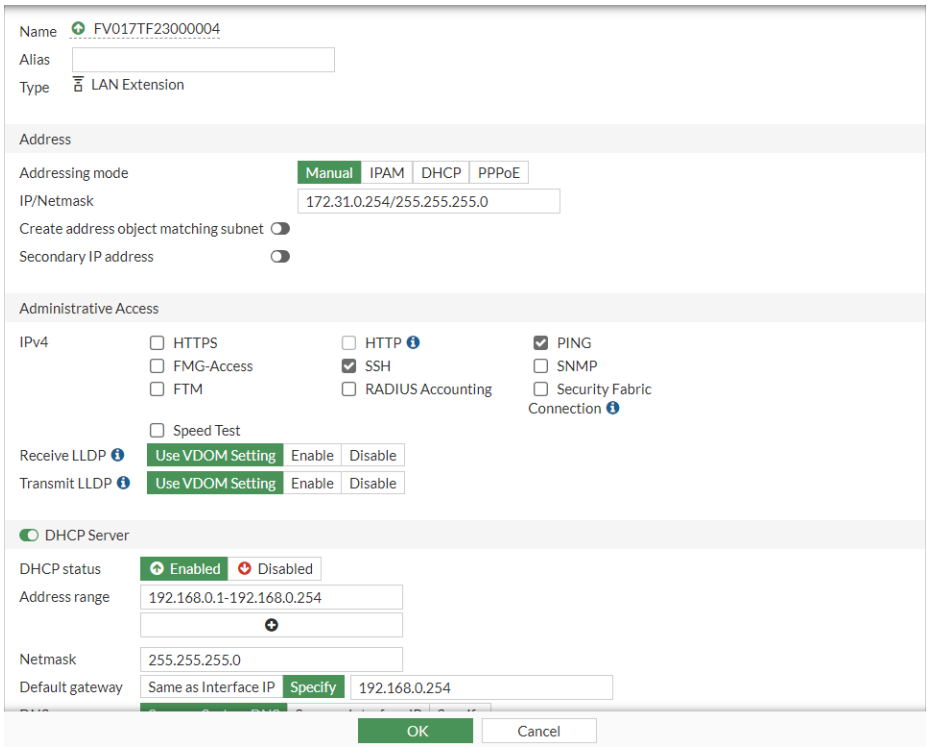
```
config extension-controller extender
    edit "FV017TF23000004"
        set id "FVA22FTF23000004"
        set authorized enable
        set device-id 0
        set extension-type lan-extension
        set profile "FVA22F-lanext-default"
    next
end
```

Configuring the LAN extension interface as a DHCP server

Configure the LAN extension interface as an DHCP server to assign IP addresses to WiFi clients.

To configure the LAN extension interface as a DHCP server in the GUI:

1. Go to *Network* > *Interfaces*, and double-click the LAN extension interface to open it for editing.
2. Enable *DHCP Server*.
3. Set *Address range*, *Netmask*, and *Default gateway*.



4. Set the remaining options as desired, and click *OK*.

To configure the LAN extension interface as a DHCP server in the CLI:

1. Assign an IP address to the LAN extension interface:

```
config system interface
    edit "FV017TF23000004"
        set vdom "root"
        set ip 172.31.0.254 255.255.255.0
        set allowaccess ping ssh
        set type lan-extension
        set role lan
        set snmp-index 27
        set ip-managed-by-fortiipam enable
        config ipv6
            set ip6-send-adv enable
            set ip6-other-flag enable
        end
        set interface "fext-ipsec-wiUx"
    next
end
```

2. Configure the DHCP server on the LAN extension interface:

```
config system dhcp server
    edit 3
        set dns-service default
        set default-gateway 172.31.0.254
```

```
        set netmask 255.255.255.0
        set interface "FV017TF23000004"
        config ip-range
            edit 1
                set start-ip 172.31.0.1
                set end-ip 172.31.0.254
            next
        end
        set dhcp-settings-from-fortiipam enable
        config exclude-range
            edit 1
                set start-ip 172.31.0.254
                set end-ip 172.31.0.254
            next
        end
    next
end
```

3. Confirm that the DHCP server can assign IP addresses to clients connecting to the FortiExtender Virtual Access Point (VAP).

   In this example, an iPhone connects to the FortiExtender 2.4GHz radio VAP named 2G-lan and receives an IP address of 172.31.0.3 from the LAN extension interface.

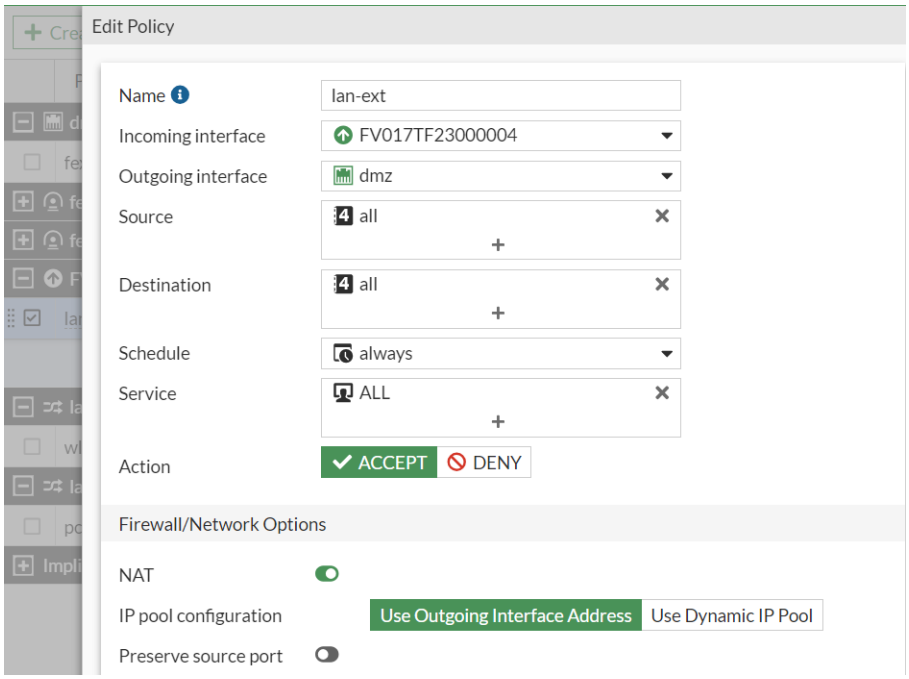
```
execute dhcp lease-list
FV017TF23000004
  IP            MAC-Address             Hostname            VCI                 SSID
     AP                   SERVER-ID           Expiry
  172.31.0.2    74:78:a6:8b:52:ff       FVA22FTF23000004    FortiExtenderVehicl
                         3                   Fri Apr 19 13:30:48 2024
  172.31.0.3    0a:ba:c9:5f:47:4d
```

Configuring a firewall policy

Configure a firewall policy with incoming interface set to the LAN extension interface to allow FortiExtender WiFi clients to reach the internet.

To configure a firewall policy to use the LAN extension interface in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy* and double-click the LAN extension policy to open it for editing.
2. Set *Incoming interface* to the LAN extension interface.
3. Set *Outgoing Interface*.



4. Set the remaining options as desired, and click *OK*.

To configure a firewall policy to use the LAN extension interface in the GUI:

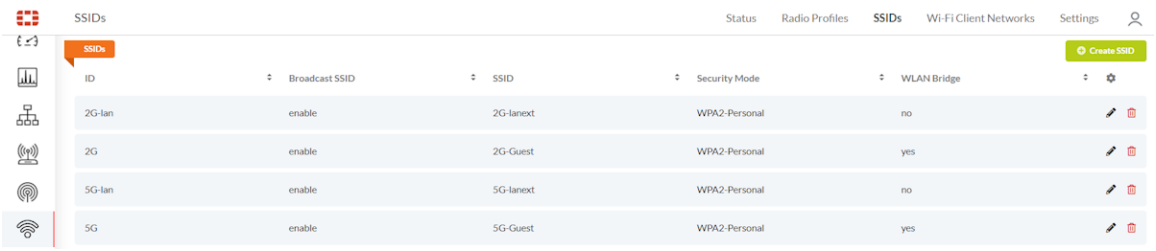
```
config firewall policy
...
    edit 4
        set name "lan-ext"
        set uuid 341c7010-270b-51ec-16b6-309891e3e880
        set srcintf "FV017TF23000004"
        set dstintf "dmz"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

Verifying FortiExtender received the configuration

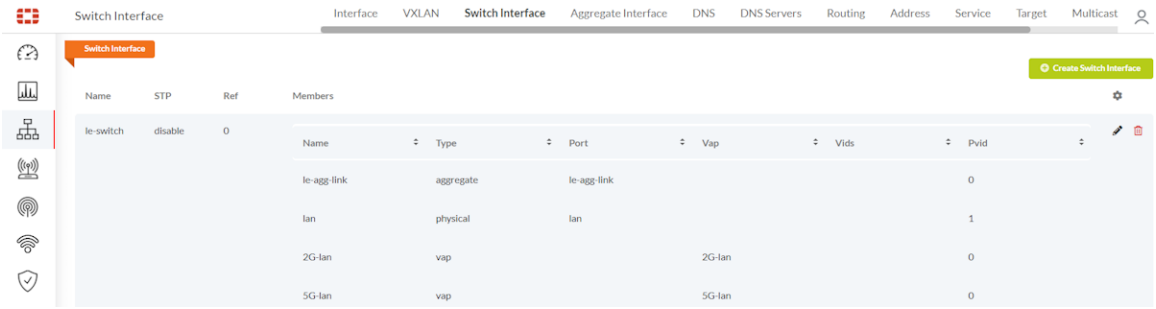
Verify that FortiExtender received the configuration from FortiGate.

To verify FortiExtender received the configuration:

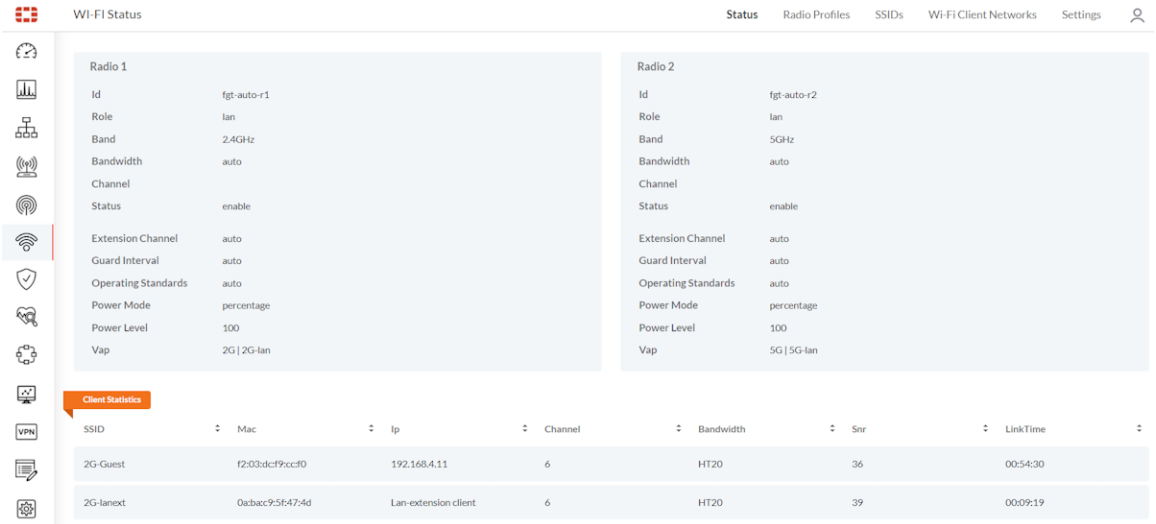
1. On FortiExtender GUI, go to *SSIDs*, and verify that the device received a LAN extension type of SSID. In this example, the LAN extension type of SSID is named *2G-lanext*.



2. Go to *Switch Interface*, and verify that the device received the WiFi interface for the LAN extension. In this example, the WiFi interface is named *2G-lan*.



3. Go to *Wi-Fi Status*, and verify that the WiFi client is connected to the LAN extension SSID. In this example, the client is connected to *2G-lanext*.



4. Confirm the FortiExtender WiFi client can reach the internet.

SCTP packets with zero checksum on the NP7 platform

The `sctp-csum-err` command provides flexibility to discard or permit IPv4 SCTP packets with zero checksum on the NP7 platform.

```
config system npu
    config fp-anomaly
        set sctp-csum-err {allow | drop | trap-to-host}
    end
end
```

`allow`	The NP7 platform will forward all SCTP packets with zero checksum.
`drop`	The NP7 platform will drop all SCTP packets with zero checksum. This is the default value.
`trap-to-host`	NP7 anomaly protection for SCTP will be disabled and SCTP packets with zero checksum will be forwarded. This option can also be used with a DoS policy to configure anomaly protection. If `policy-offload-level` is set to `dos-offload`, DoS policy anomaly protection is offloaded to the NP7 platform.

Industrial Connectivity

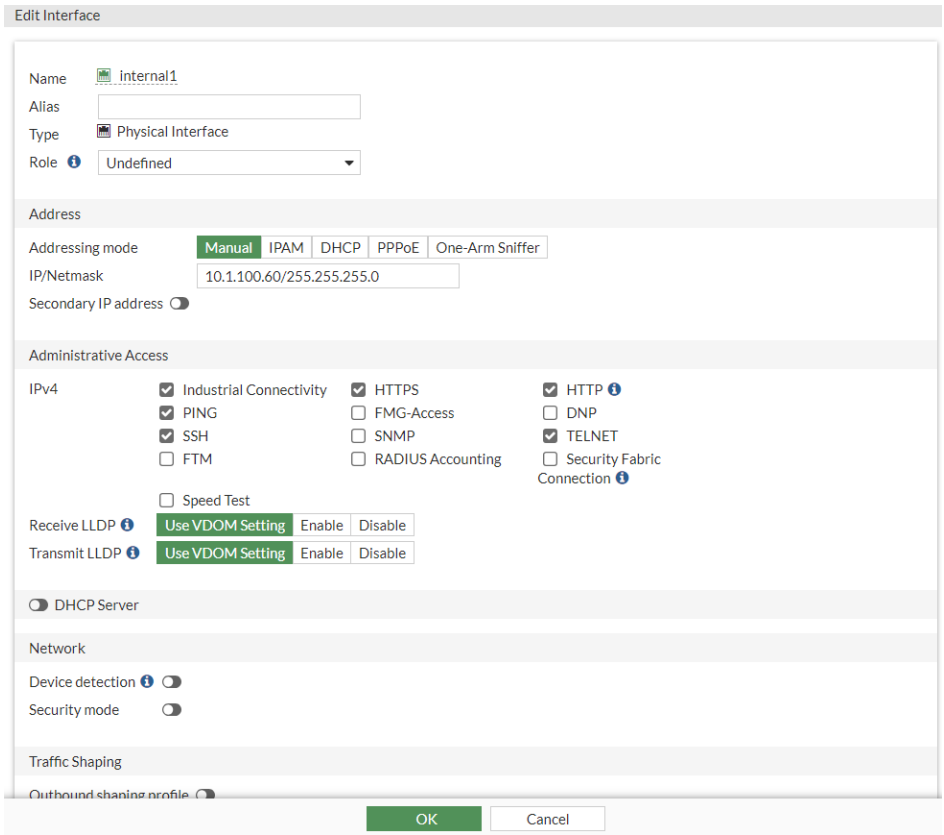
The industrial connectivity daemon (icond) and Industrial Connectivity service are available for FortiGate Rugged models equipped with a serial RS-232 (DB9/RJ45) interface to:

- Receive data in IEC 60870-5-101 serial protocol and convert it to IEC 60870-5-104 TCP/IP. See Sample configuration to convert IEC 60870-5-101 serial to IEC 60870-5-104 TCP/IP transport on page 1023.
- Receive data in Modbus serial (RTU/ASCII) protocol and convert it to Modbus TCP. See Sample configuration to convert Modbus serial to Modbus TCP on page 1024.

You can allow Industrial Connectivity access to an interface in the GUI and CLI.

To enable Industrial Connectivity for an interface in the GUI:

1. Go to *Network* > *Interfaces*.
2. Click *Create New* > *Interface*, or double-click an interface to open in for editing.
3. Set *Role* to *Undefined* or *WAN*
   Only internal and WAN interfaces support *Industrial Connectivity* administrative access.
4. In the *Administrative Access* section, select *Industrial Connectivity*.



5. Set the remaining options as desired, and click *OK*.

To enable Industrial Connectivity for an interface in the CLI:

```
config system interface
    edit <name>
        set allowaccess icond
        ...
    next
end
```

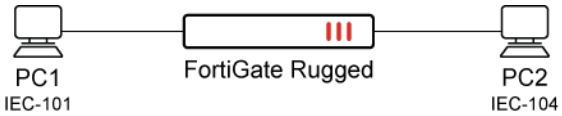
`allowaccess icond`	Specify what types of management protocols can access the interface: • `icond`: Industrial Connectivity service access to proxy traffic between serial port and TCP/IP.

Use the `config system icond` command to configure the Industrial Connectivity service provided by the Industrial Connectivity daemon (icond).

Sample configuration to convert IEC 60870-5-101 serial to IEC 60870-5-104 TCP/IP transport

After the Industrial Connectivity service is enabled and configured for an interface, supported FortiGate Rugged devices can receive data from Supervisory Control and Data Acquisition (SCADA) systems in serial format and convert it to TCP/IP formats.

In the following topology, PC1 uses the IEC 60870-5-101 (shortened to IEC-101) protocol to transmit data from SCADA systems to FortiGate Rugged, where the data is converted to the IEC 60870-5-104 (shortened to IEC-104) protocol, and sent to PC2.



The data is converted as follows:

- FortiGate Rugged transmits data over TCP port 502.
- Protocols IEC 60870-5-101 and IEC 60870-5-104 are both used to transmit the data.

  While IEC-101 is based on a serial transmission of data (for example, using RS-232 and FSK-based modems), IEC-104 is packet oriented and based on TCP/IP transmission.

To enable Industrial Connectivity for an interface in the CLI:

```
config system interface
    edit "internal1"
        set vdom "root"
        set ip 10.1.100.60 255.255.255.0
        set allowaccess ping https ssh http telnet icond
        set type physical
        set snmp-index 3
    next
end
```

To configure the Industrial Connectivity service in the CLI:

1. Configure the Industrial Connectivity service

   ```
   config system icond
       set status enable
       set type iec101-104
       set tty-device "serial0"
   end
   ```

2. Get the default status:

   ```
   get system icond
   status              : enable
   type                : iec101-104
   tty-device          : serial0
   ```

```
tty-baudrate        : 9600
tty-parity          : even
tty-databits        : 8
tty-stopbits        : 1
tty-flowcontrol     : none
iec101-mode         : unbalanced
iec101-laddr-size   : 1
iec101-laddr-local  : 1
iec101-laddr-remote : 2
iec101-use-ack-char : disable
iec101-keepalive    : enable
iec101-t0           : 500
iec101-trp          : 2500
iec104-t1           : 15
iec104-t2           : 10
iec104-t3           : 20
iec104-k            : 12
iec104-w            : 8
```

Sample configuration to convert Modbus serial to Modbus TCP

After the Industrial Connectivity service is enabled and configured for an interface, supported FortiGate Rugged devices can receive data from Modbus networks in serial format and convert it to TCP.

In the following topology, the Modbus controller uses Modbus serial to transmit data to FortiGate Rugged, where the data is converted to the Modbus TCP, and sent to the Modbus Remote Terminal Unit (RTU).

To enable Industrial Connectivity for an interface in the CLI:

```
config system interface
    edit "internal1"
        set vdom "root"
        set ip 10.1.100.60 255.255.255.0
        set allowaccess ping https http telnet icond
        set type physical
        set description "link to modbus server"
        set snmp-index 3
    next
end
```

To configure the Industrial Connectivity service in the CLI:

1. Configure the Industrial Connectivity service

```
config system icond
    set status enable
    set type modbus-serial-tcp
    set tty-device "serial0"
end
```

2. Get the default status:

```
get system icond
status              : enable
type                : modbus-serial-tcp
tty-device          : serial0
tty-baudrate        : 9600
tty-parity          : even
tty-databits        : 8
tty-stopbits        : 1
tty-flowcontrol     : none
modbus-serial-mode  : RTU/ASCII
modbus-serial-addr  : 1
modbus-serial-timeout-resp: 500
modbus-tcp-unit-id  : 255
```

Diagnostics

Administrators can use the *Diagnostics* page to access the following tools:

Capture packet	Captures packet streams in real-time to let you view header and payload information. See Using the packet capture tool on page 1025.
Debug flow	Traces packet flow through FortiOS to help you diagnose and debug issues. See Using the debug flow tool on page 1032.

See also Performing a sniffer trace or packet capture on page 4966 and Debugging the packet flow on page 4968.

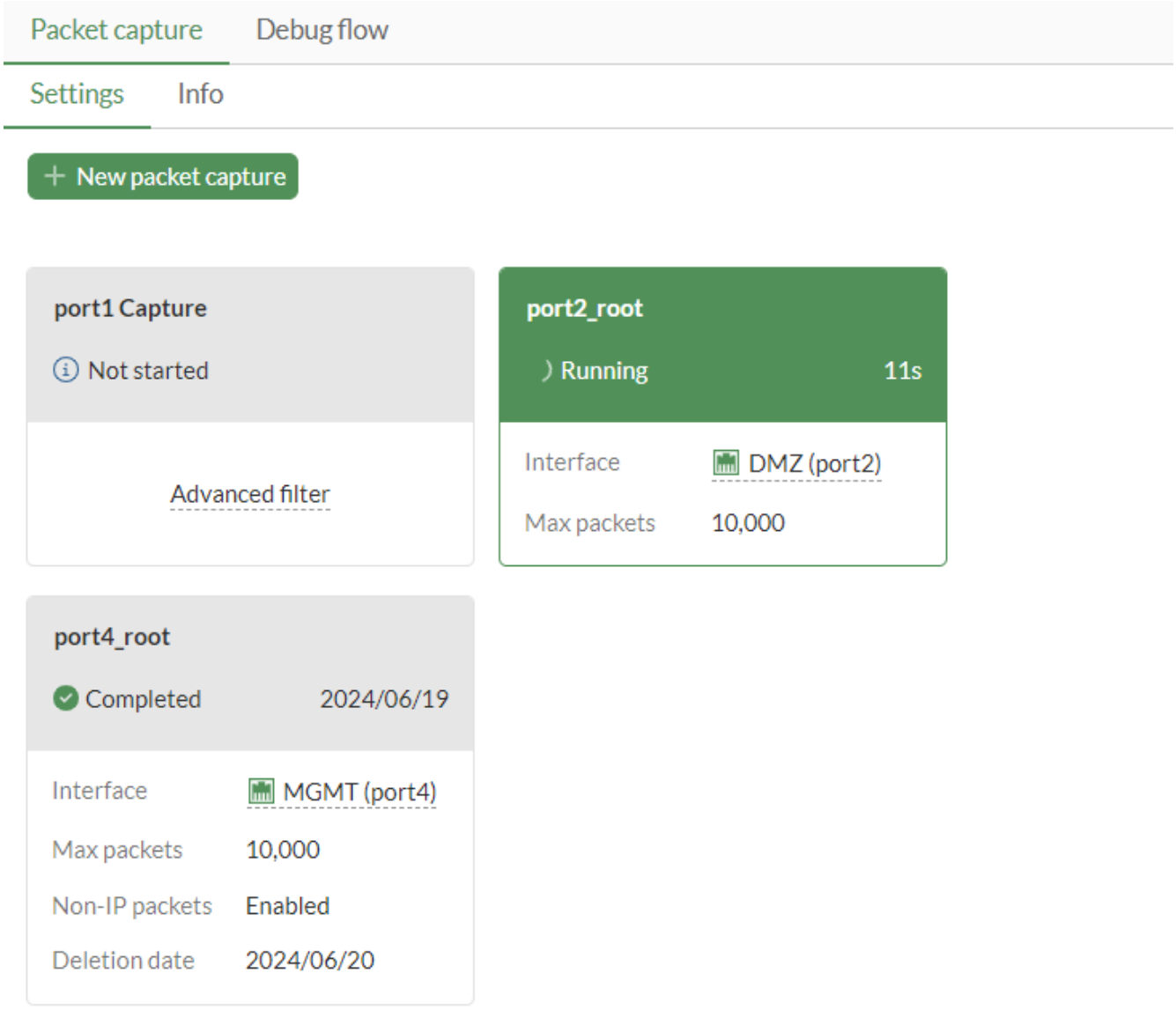
Using the packet capture tool

Administrators can use the packet capture tool to select a packet and view its header and payload information in real-time. Once completed, packets can be filtered by various fields or through the search bar. The capture can be saved as a PCAP file that you can use with a third-party application, such as Wireshark, for further analysis.

Packet capture criteria can be stored for the re-initiation of packet captures multiple times using the same parameters, such as interface, filters, and so on.

Packet capture criteria can be created and stored in order to re-initiate packet captures in the GUI with the same parameters. Capture cards in the *Network* > *Diagnostics* page are sorted in alphabetical order of the configured name and colored depending on state:

- Green: The packet capture is running.
- Gray: The packet capture has not started yet, has completed, or the capture files have been deleted.



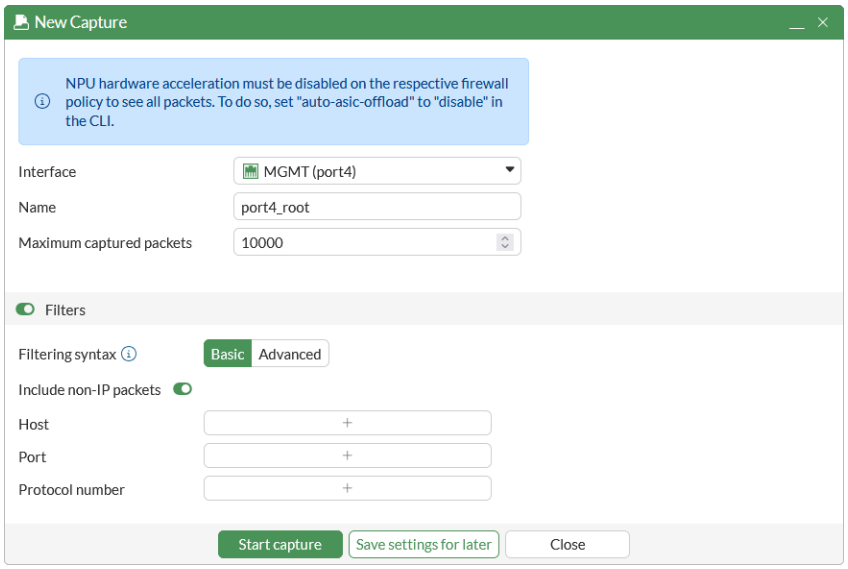
When creating the packet capture:

- The *Name* field must be a unique name for the packet capture criteria being configured.
- Enabling *Include non-IP packets* allows non-IP address packets to be captured when enabled. Supported non-IP address packet types include ARP, RARP, LLC, LLDP, VLAN, and LACPDU. When the packet capture is complete, non-IP address packets will include header information, however, unsupported types will display as *Unknown*.
- After configuring the packet capture criteria, you can choose to *Start capture*, *Save settings for later*, or *Close*. Starting a packet capture or saving the configured settings will both store criteria for future use.

System event logs are generated when a packet capture is started or stopped in the GUI.

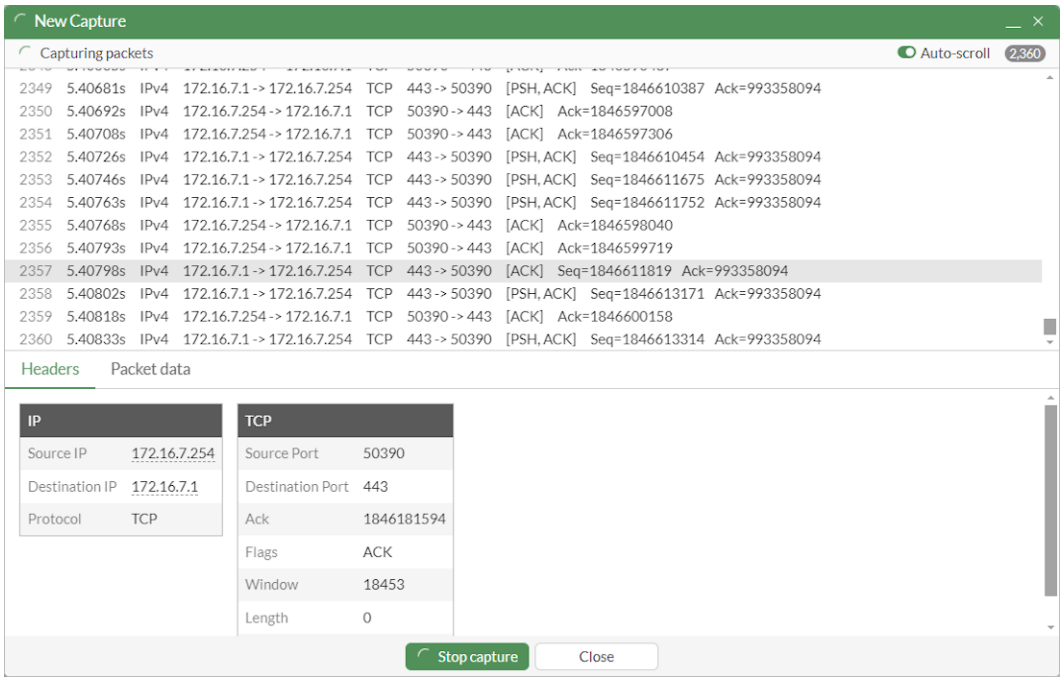
To use the packet capture tool in the GUI:

1. Go to *Network* > *Diagnostics* > *Packet Capture*.
2. Optionally, select an *Interface* (*any* is the default).
3. Optionally, enable *Filters* and select a *Filtering syntax*:
   a. *Basic*: enter criteria for the *Host*, *Port*, and *Protocol number*.



   b. *Advanced*: enter a string, such as *src host 172.16.200.254 and dst host 172.16.200.1 and dst port 443*.
4. Click *Start capture*. The capture is visible in real-time.
5. While the capture is running, select a packet, then click the *Headers* or *Packet data* tabs to view more information.

> When the packet capture is running, disable *Auto-scroll* to stop automatic scrolling behavior when new packets arrive.



6. When the capture is finished, click *Save as pcap*. The PCAP file is automatically downloaded.
7. Optionally, use the *Search* bar or the column headers to filter the results further.

Multiple packet captures

Multiple packet captures can be run simultaneously for when many packet captures are needed for one situation. For example, ingress and egress interfaces can be captured at the same time to compare traffic or the physical interface and VPN interface can be captured using different filters to see if packets are leaving the VPN.

The packet capture dialog can be docked and minimized to run in the background. The minimized dialog aligns with other CLI terminals that are minimized.

How many packet captures and the number of packets that can be captured depend on the device model.

Whether the device model has disk storage affects when packet captures are deleted. Without disk storage, packet captures are deleted 24 hours after completion or immediately after reboot. With disk storage, packet captures are deleted after 7 days.

To find the limit on the number of packet captures supported for a specific device model, use the Maximum Values Table, and search for the object `firewall.on-demand-sniffer`.

To run multiple packet captures at the same time:

1. Go to *Network* > *Diagnostics*.
2. Configure the first packet capture:
   a. Click *New packet capture*.
   b. Select the *Interface* and configure other settings as needed.

c. Click *Start capture*. The first packet capture begins.

3. Minimize the packet capture. The packet capture continues to run.
4. Configure the second packet capture:
   a. Click *New packet capture*.
   b. Select the *Interface* and configure other settings as needed.
   c. Click *Start capture*. The second packet capture begins.
5. When the captures are complete, expand the dialog and select *Save as pcap* for each packet capture.

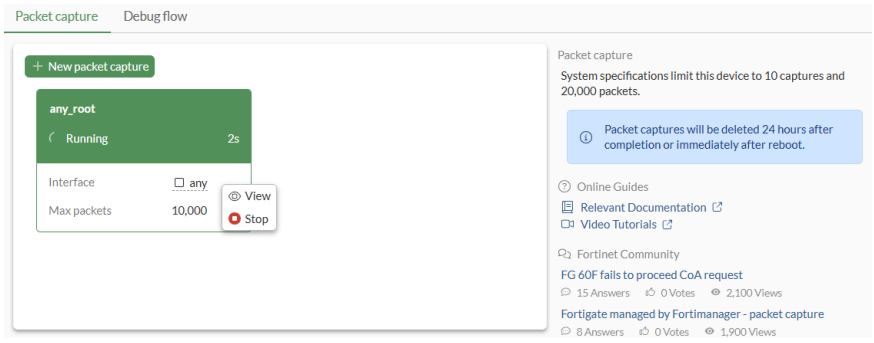
Persistent packet captures

If the browser is closed or refreshed, users can return at a later time to view, stop, restart, and download the packet capture.

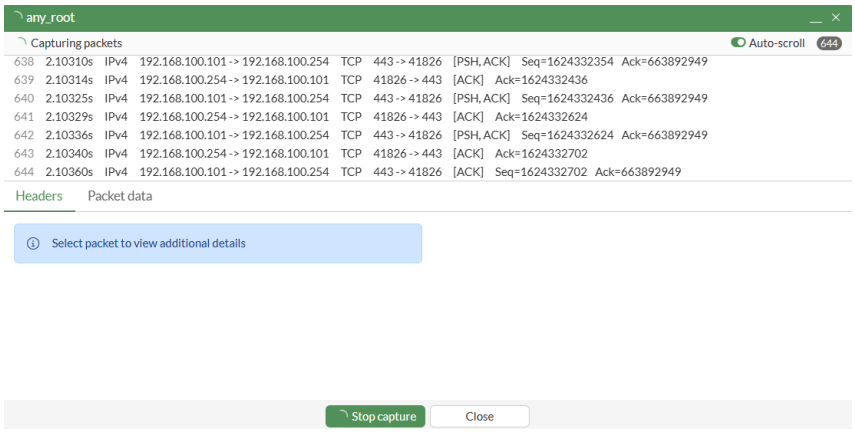
The limit of simultaneous packet captures and the number of packets per capture that can be saved is dependent on the FortiGate model and capabilities. If this limit is reached, new packet captures cannot be created. Go to *Network > Diagnostics* to view the limit values.

To interact with a packet capture in the GUI:

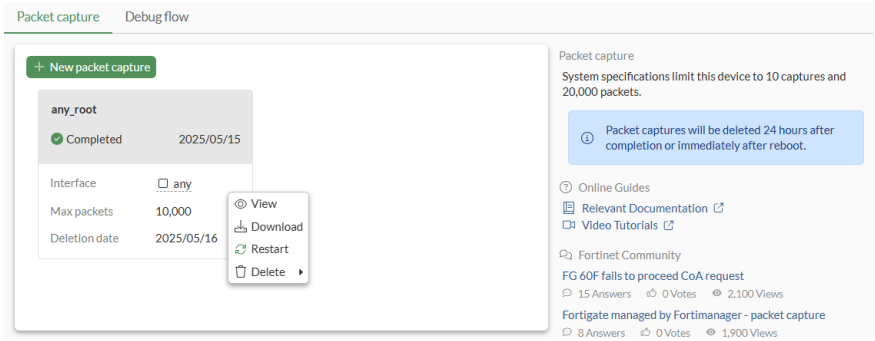
1. Go to *Network > Diagnostics*.
2. Click *New packet capture*.
3. Configure the packet capture fields.
4. Click *Start Capture*.
5. Stop the packet capture:
   - If you have closed the packet capture, select the packet capture and click *Stop*.



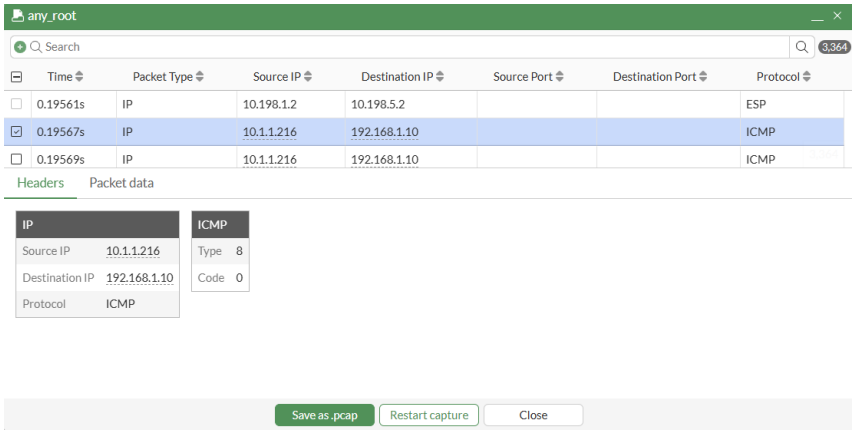
   - If you are viewing the packet capture, click *Stop capture*.



6. Download the packet capture:
   - If you have closed the packet capture, select the packet capture and click *Download*.

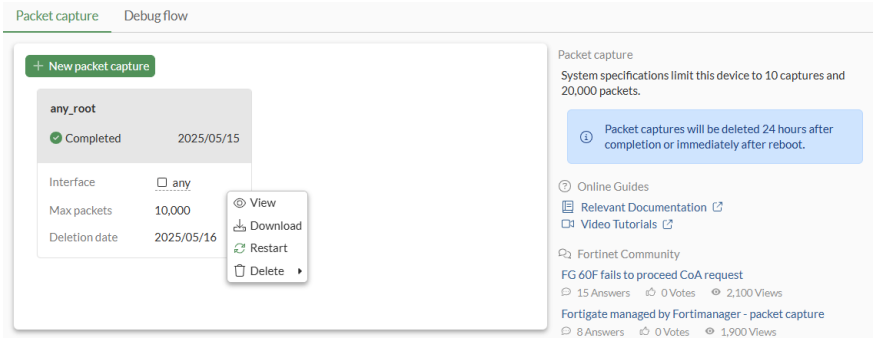


   - If you are viewing the packet capture, click *Save as .pcap*.

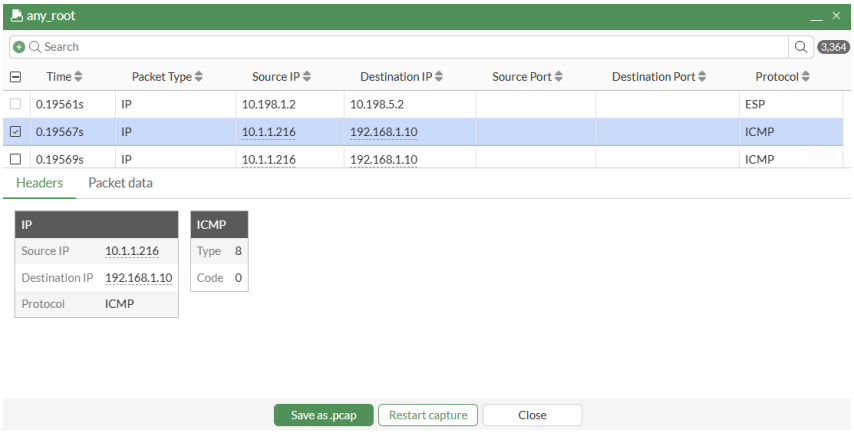


7. Restart the packet capture:

- If you have closed the packet capture, select the packet capture and click *Restart*.



- If you are viewing the packet capture, click *Restart capture*.



You will have the option to save the previous packet capture or discard it.

Controlling GUI packet captures in the CLI

GUI packet captures can be controlled in the CLI using the `on-demand-sniffer` commands.

To control GUI packet captures in the CLI:

1. Add a new firewall on-demand sniffer table to store the GUI packet capture settings and filters:

```
config firewall on-demand-sniffer
    edit "port1 Capture"
        set interface "port1"
        set max-packet-count 10000
        set advanced-filter "net 172.16.200.0/24 and port 443 and port 49257"
    next
end
```

2. Run packet capture commands:
   a. List all of the packet captures:

```
# diagnose on-demand-sniffer list
mkey: port1 Capture
```

```
interface: port1
status: not_started
start time:
end time:
```

b. Start a packet capture:

```
# diagnose on-demand-sniffer start "port1 Capture"
```

c. Stop a packet capture:

```
# diagnose on-demand-sniffer stop "port1 Capture"
```

d. Delete the result of a packet capture:

```
# diagnose on-demand-sniffer delete-results "port1 Capture"
```

To clean packet capture files:

```
# diagnose test application forticron 14
Running packet capture cleanup forcefully
```

For more information about running a packet capture in the CLI, see Performing a sniffer trace or packet capture on page 4966.

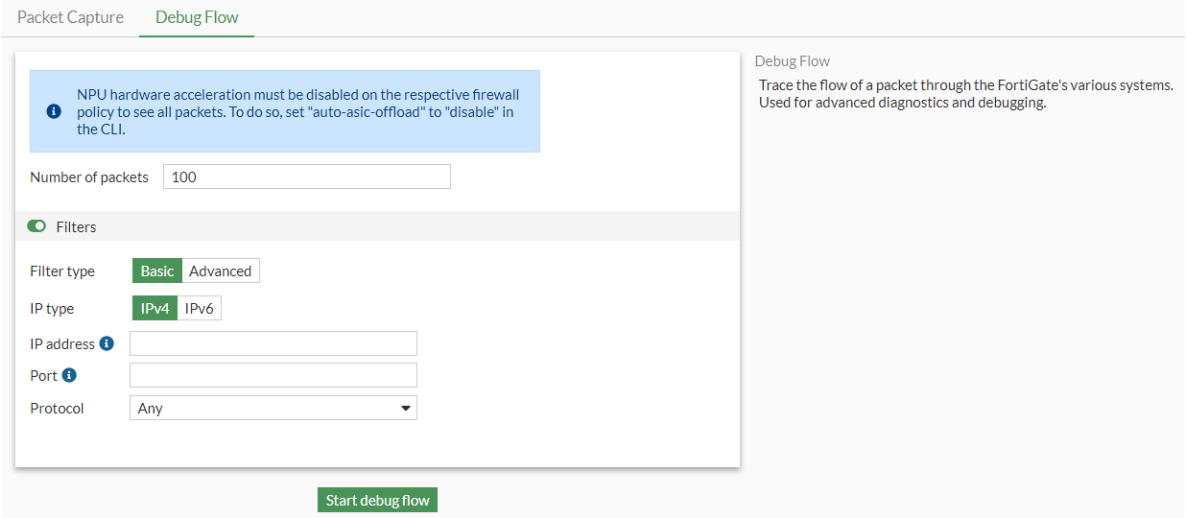
Using the debug flow tool

Administrators can use the debug flow tool to display debug flow output in real-time until it is stopped. The completed output can be filtered by time, message, or function. The output can be exported as a CSV file.

For information about using the debug flow tool in the CLI, see Debugging the packet flow on page 4968.

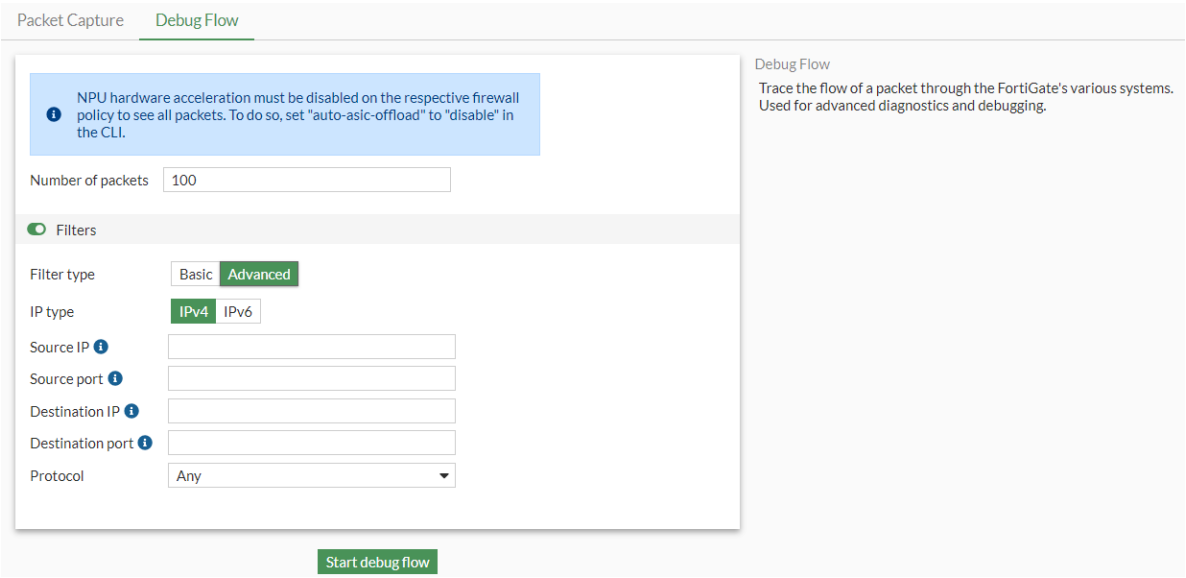
To run a debug flow:

1. Go to *Network* > *Diagnostics* > *Debug Flow*.
2. Optionally, enable *Filters* and select a *Filter type*:
   a. *Basic*: filter by *IP address*, *Port*, and *Protocol*, which is the equivalent of:
      - `# diagnose debug flow filter addr <addr/range>`
      - `# diagnose debug flow filter port <port/range>`
      - `# diagnose debug flow filter proto <protocol>`

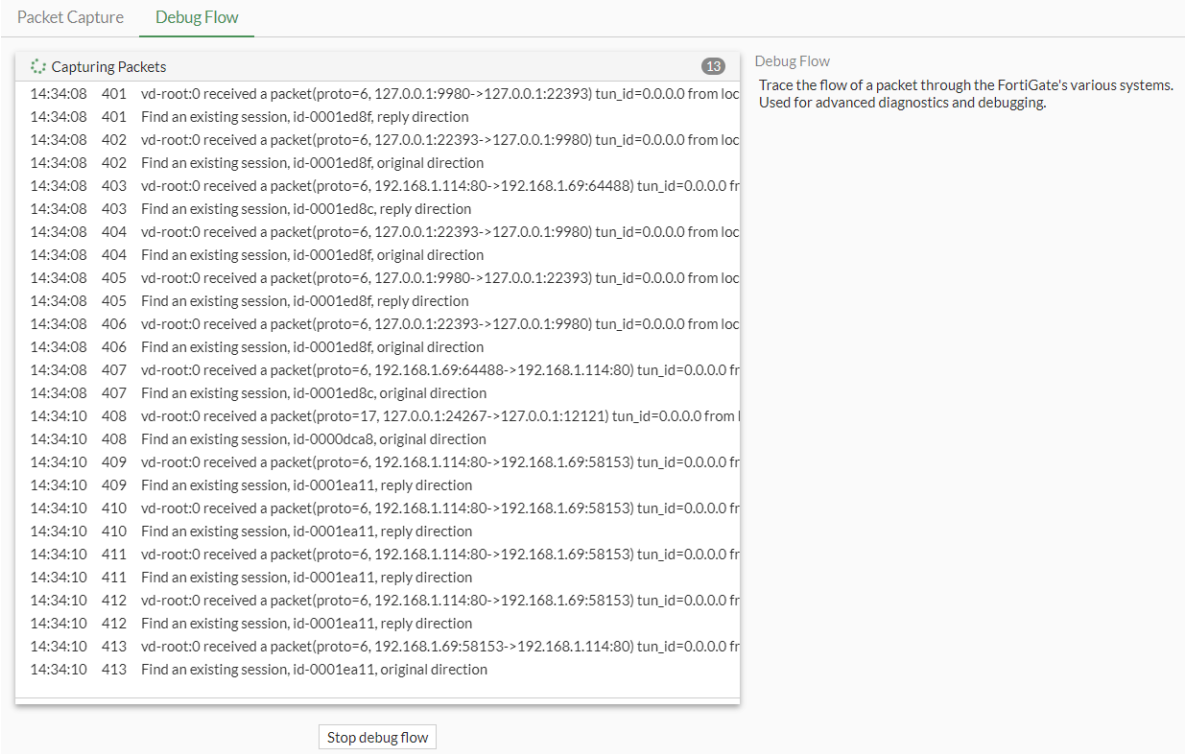


b. *Advanced*: filter by *Source IP*, *Source port*, *Destination IP*, *Destination port*, and *Protocol*, which is the equivalent of:

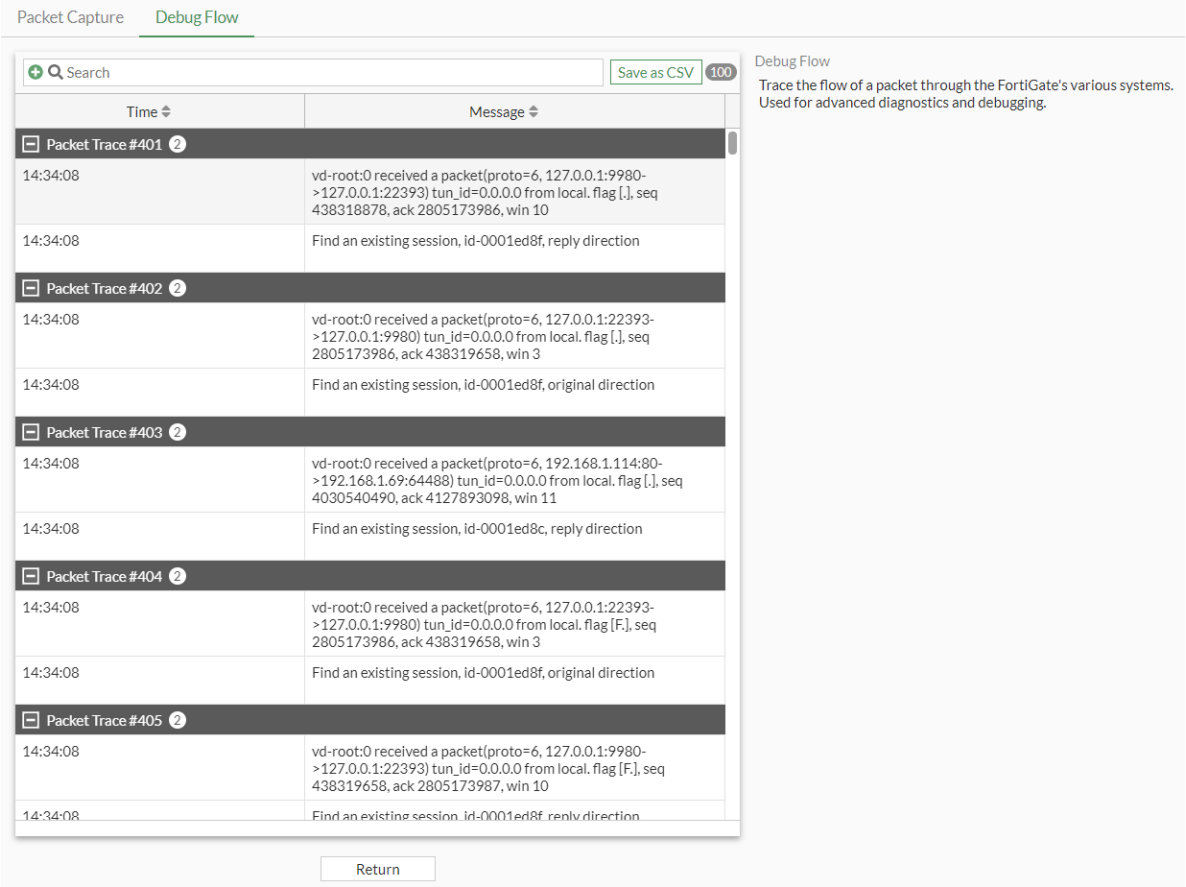
- `# diagnose debug flow filter saddr <addr/range>`
- `# diagnose debug flow filter sport <port/range>`
- `# diagnose debug flow filter daddr <addr/range>`
- `# diagnose debug flow filter dport <port/range>`
- `# diagnose debug flow filter proto <protocol>`



3. Click *Start debug flow*. The debug messages are visible in real-time.

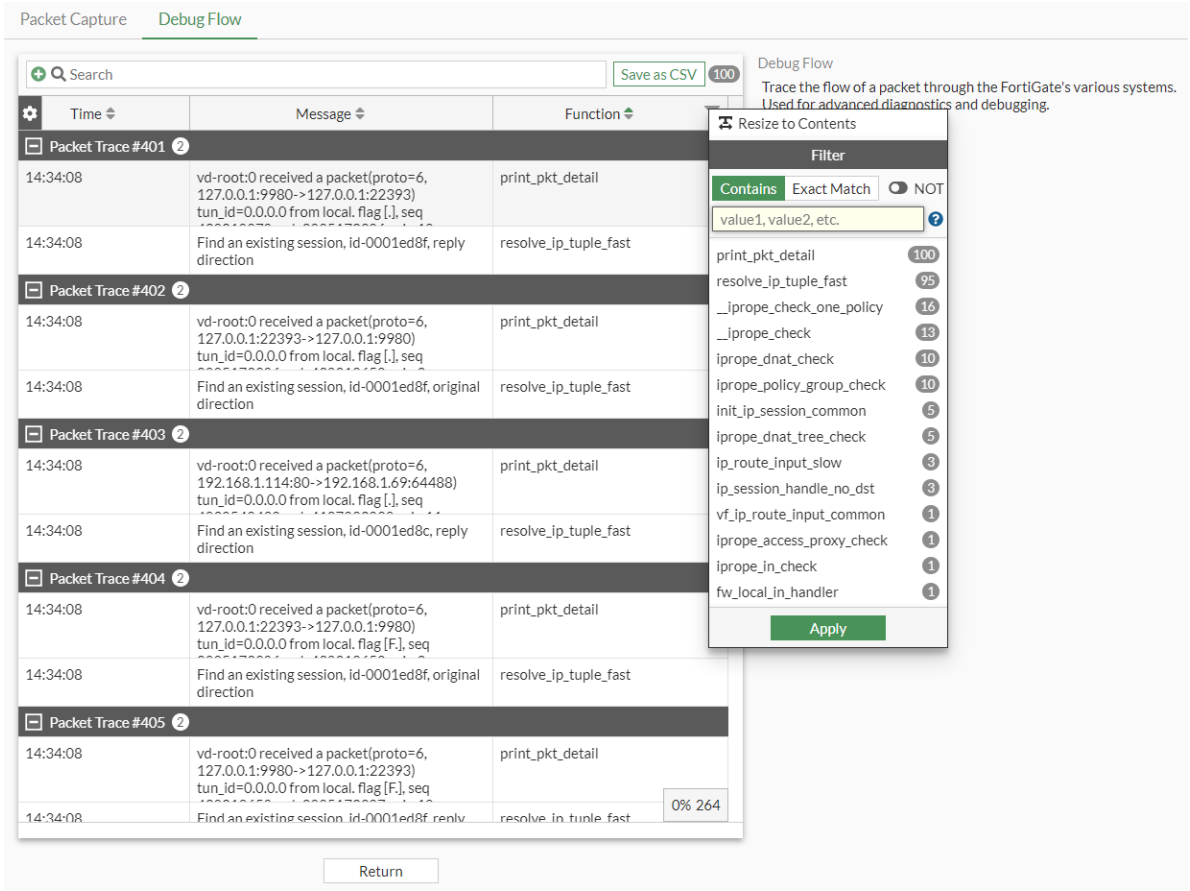


4. When the debug flow is finished (or you click *Stop debug flow*), click *Save as CSV*. The CSV file is automatically downloaded.



The current output can be filtered by *Time* and *Message*. The *Function* field can be added.

5. Hover over the table header and click the gear icon (*Configure Table*).
6. Select *Function* and click *Apply*. The *Function* column is displayed and can be used to filter the output for further analysis.


Packet Capture
Debug Flow
Search
Save as CSV
100
Time
Message
Function
Packet Trace #401 2
14:34:08
vd-root:0 received a packet(proto=6, 127.0.0.1:9980->127.0.0.1:22393) tun_id=0.0.0.0 from local. flag [.], seq
print_pkt_detail
14:34:08
Find an existing session, id-0001ed8f, reply direction
resolve_ip_tuple_fast
Packet Trace #402 2
14:34:08
vd-root:0 received a packet(proto=6, 127.0.0.1:22393->127.0.0.1:9980) tun_id=0.0.0.0 from local. flag [.], seq
print_pkt_detail
14:34:08
Find an existing session, id-0001ed8f, original direction
resolve_ip_tuple_fast
Packet Trace #403 2
14:34:08
vd-root:0 received a packet(proto=6, 192.168.1.114:80->192.168.1.69:64488) tun_id=0.0.0.0 from local. flag [.], seq
print_pkt_detail
14:34:08
Find an existing session, id-0001ed8c, reply direction
resolve_ip_tuple_fast
Packet Trace #404 2
14:34:08
vd-root:0 received a packet(proto=6, 127.0.0.1:22393->127.0.0.1:9980) tun_id=0.0.0.0 from local. flag [F.], seq
print_pkt_detail
14:34:08
Find an existing session, id-0001ed8f, original direction
resolve_ip_tuple_fast
Packet Trace #405 2
14:34:08
vd-root:0 received a packet(proto=6, 127.0.0.1:9980->127.0.0.1:22393) tun_id=0.0.0.0 from local. flag [F.], seq
print_pkt_detail
0% 264
Return
Debug Flow
Trace the flow of a packet through the FortiGate's various systems.
Used for advanced diagnostics and debugging.
Resize to Contents
Filter
Contains
Exact Match
NOT
value1, value2, etc.
print_pkt_detail 100
resolve_ip_tuple_fast 95
__iprope_check_one_policy 16
__iprope_check 13
iprope_dnat_check 10
iprope_policy_group_check 10
init_ip_session_common 5
iprope_dnat_tree_check 5
ip_route_input_slow 3
ip_session_handle_no_dst 3
vf_ip_route_input_common 1
iprope_access_proxy_check 1
iprope_in_check 1
fw_local_in_handler 1
Apply

SD-WAN

The following topics provide information about SD-WAN:

- SD-WAN overview on page 1037
- SD-WAN quick start on page 1041
- SD-WAN members and zones on page 1052
- SD-WAN performance SLA on page 1068
- SD-WAN rules on page 1105
- SD-WAN Underlay Bandwidth and Quality Monitoring service on page 1218
- Overlay on page 1277
- Routing on page 1426
- Security on page 1475
- Advanced configuration on page 1475
- Troubleshooting SD-WAN on page 1554
- SD-WAN cloud on-ramp on page 1567

SD-WAN overview

SD-WAN is a software-defined approach to managing Wide-Area Networks (WAN). It consolidates the physical transport connections, or underlays, and monitors and load-balances traffic across the links. VPN overlay networks can be built on top of the underlays to control traffic across different sites.

Health checks and SD-WAN rules define the expected performance and business priorities, allowing the FortiGate to automatically and intelligently route traffic based on the application, internet service, or health of a particular connection.

WAN security and intelligence can be extended into the LAN by incorporating wired and wireless networks under the same domain. FortiSwitch and FortiAP devices integrate seamlessly with the FortiGate to form the foundation of an SD-Branch.

Some of the key benefits of SD-WAN include:

- Reduced cost with transport independence across MPLS, 4G/5G LTE, and others.
- Reduced complexity with a single vendor and single-pane-of-glass management.
- Improve business application performance thanks to increased availability and agility.
- Optimized user experience and efficiency with SaaS and public cloud applications.

Starting in FortiOS 7.6.5, some FortiGate models with two WAN interfaces have a default SD-WAN configuration. See Default SD-WAN configuration on FortiGate models with two WAN ports on page 131 for more information.

SD-WAN components and design principles

SD-WAN can be broken down into three layers:

- Management and orchestration
- Control, data plane, and security
- Network access

The control, data plane, and security layer can only be deployed on a FortiGate. The other two layers can help to scale and enhance the solution. For large deployments, FortiManager and FortiAnalyzer provide the management and orchestration capabilities FortiSwitch and FortiAP provide the components to deploy an SD-Branch.

Layer	Functions	Devices	
Management and orchestration	• Unified management • Template based solution • Zero touch provisioning • Logging, monitoring, and analysis • Automated orchestration using the REST API	FortiManager	FortiAnalyzer

Control, data plane, and security	• Consolidation of underlays and overlays into SD-WAN zones
	  • Underlay and Overlay
	• Scalable VPN solutions using ADVPN
	  • Overlay
	• Static and dynamic routing definition
	  • Routing
	• NGFW firewalling
	  • Security
	• SD-WAN health-checks and monitoring
	  • SD-WAN
	• Application-aware steering and intelligence
	  • SD-WAN
		FortiGate	

Network access	• Wired and wireless network segmentation • Built-in network access control	FortiSwitch	FortiAP

Design principles

The Five-pillar approach, described in the SD-WAN / SD-Branch Architecture for MSSPs guide, is recommended when designing a secure SD-WAN solution.

Underlay

Determine the WAN links that will be used for the underlay network, such as your broadband link, MPLS, 4G/5G LTE connection, and others.

For each link, determine the bandwidth, quality and reliability (packet loss, latency, and jitter), and cost. Use this information to determine which link to prefer, what type of traffic to send across the each link, and to help you the baselines for health-checks.

Overlay

VPN overlays are needed when traffic must travel across multiple sites. These are usually site-to-site IPsec tunnels that interconnect branches, datacenters, and the cloud, forming a hub-and-spoke topology.

The management and maintenance of the tunnels should be considered when determining the overlay network requirements. Manual tunnel configuration might be sufficient in a small environment, but could become unmanageable as the environment size increases. ADVPN can be used to help scale the solution; see ADVPN on page 3235 for more information.

Routing

Traditional routing designs manipulate routes to steer traffic to different links. SD-WAN uses traditional routing to build the basic routing table to reach different destinations, but uses SD-WAN rules to steer traffic. This allows the steering to be based on criteria such as destination, internet service, application, route tag, and the health of the link. Routing in an SD-WAN solution is used to identify all possible routes across the underlays and overlays, which the FortiGate balances using ECMP.

In the most basic configuration, static gateways that are configured on an SD-WAN member interface automatically provide the basic routing needed for the FortiGate to balance traffic across the links. As the number of sites and destinations increases, manually maintaining routes to each destination becomes difficult. Using dynamic routing to advertise routes across overlay tunnels should be considered when you have many sites to interconnect.

Security

Security involves defining policies for access control and applying the appropriate protection using the FortiGate's NGFW features. Efficiently grouping SD-WAN members into SD-WAN zones must also be considered. Typically, underlays provide direct internet access and overlays provide remote internet or network access. Grouping the underlays together into one zone, and the overlays into one or more zones could be an effective method.

SD-WAN

The SD-WAN pillar is the intelligence that is applied to traffic steering decisions. It is comprised of four primary elements:

- **SD-WAN zones**

  SD-WAN is divided into zones. SD-WAN member interfaces are assigned to zones, and zones are used in policies as source and destination interfaces. You can define multiple zones to group SD-WAN interfaces together, allowing logical groupings for overlay and underlay interfaces. Routing can be configured per zone.

  See SD-WAN members and zones on page 1052.
- **SD-WAN members**

  Also called interfaces, SD-WAN members are the ports and interfaces that are used to run traffic. At least one interface must be configured for SD-WAN to function.

  See Underlay on page 1042.
- **Performance SLAs**

  Also called health-checks, performance SLAs are used to monitor member interface link quality, and to detect link failures. When the SLA falls below a configured threshold, the route can be removed, and traffic can be steered to different links in the SD-WAN rule.

  SLA health-checks use active or passive probing:

  - Active probing requires manually defining the server to be probed, and generates consistent probing traffic.
  - Passive probing uses active sessions that are passing through firewall policies used by the related SD-WAN interfaces to derive health measurements. It reduces the amount of configuration, and eliminates probing traffic. See Passive WAN health measurement on page 1081 for details.

  See SD-WAN performance SLA on page 1068.
- **SD-WAN rules**

  Also called services, SD-WAN rules control path selection. Specific traffic can be dynamically sent to the best link, or use a specific route.

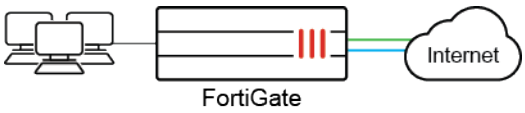
  Rules control the strategy that the FortiGate uses when selecting the outbound traffic interface, the SLAs that are monitored when selecting the outgoing interface, and the criteria for selecting the traffic that adheres to the rule. When no SD-WAN rules match the traffic, the implicit rule applies.

  See SD-WAN rules on page 1105.

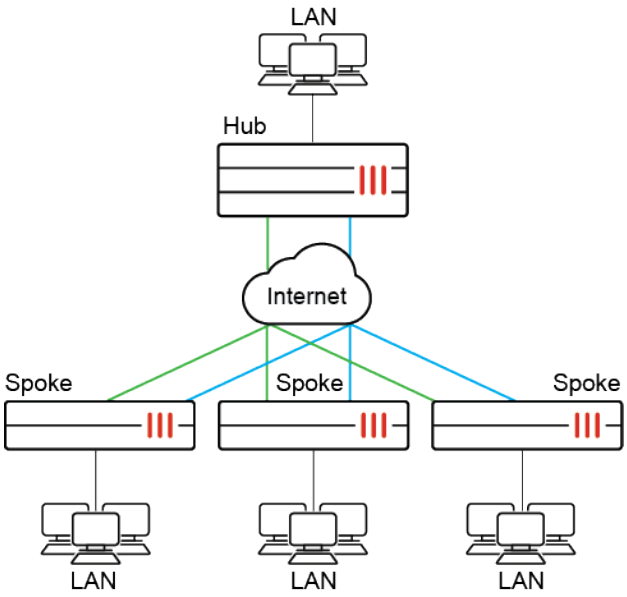
SD-WAN designs and architectures

The core functionalities of Fortinet's SD-WAN solution are built into the FortiGate. Whether the environment contains one FortiGate, or one hundred, you can use SD-WAN by enabling it on the individual FortiGates.

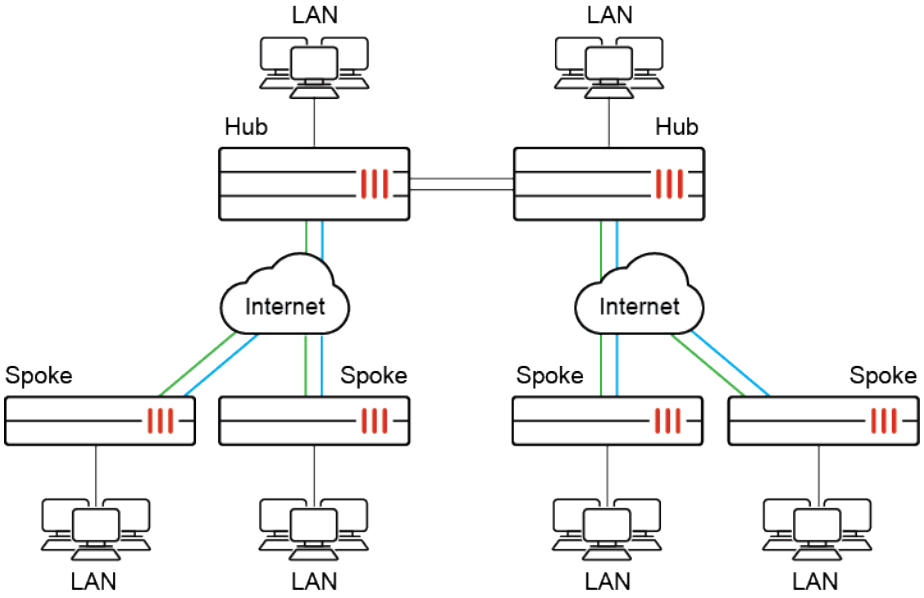
At a basic level, SD-WAN can be deployed on a single device in a single site environment:



At a more advanced level, SD-WAN can be deployed in a multi-site, hub and spoke environment:



At an enterprise or MSSP level, the network can include multiple hubs, possibly across multiple regions:

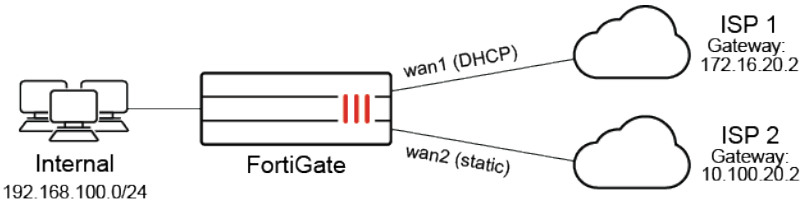


For more details, see the SD-WAN / SD-Branch Architecture for MSSPs guide.

SD-WAN quick start

This section provides an example of how to start using SD-WAN for load balancing and redundancy.

In this example, two ISP internet connections, wan1 (DHCP) and wan2 (static), use SD-WAN to balance traffic between them at 50% each.



1. Underlay on page 1042
2. Overlay on page 1043
3. Routing on page 1043
4. Security on page 1044
5. SD-WAN on page 1044
6. Results on page 1047
7. Configuring SD-WAN in the CLI on page 1049

Underlay

First, SD-WAN must be enabled and member interfaces must be selected and added to a zone. The selected FortiGate interfaces can be of any type (physical, aggregate, VLAN, IPsec, and others), but must be removed from any other configurations on the FortiGate.

In this step, two interfaces are configured and added to the default SD-WAN zone (virtual-wan-link) as SD-WAN member interfaces. This example uses a mix of static and dynamic IP addresses; your deployment could also use only one or the other.

Once the SD-WAN members are created and added to a zone, the zone can be used in firewall policies, and the whole SD-WAN can be used in static routes.

To configure SD-WAN members:

1. Configure the wan1 and wan2 interfaces. See Interface settings on page 225 for details.
   a. Set the wan1 interface *Addressing mode* to *DHCP* and *Distance* to *10*.

      By default, *Retrieve default gateway from server* (`defaultgw` in the CLI) is enabled for DHCP interfaces. This enables using the default gateway information that is retrieved from the DHCP server to create a default route through the DHCP interface with the default administrative distance.

      The default administrative distance for DHCP interfaces is *5*, and for static routes it is *10*. It is important to account for this when configuring your SD-WAN for 50/50 load balancing by setting the DHCP interface's distance to *10*.

   b. Set the wan2 interface *IP/Netmask* to *10.100.20.1 255.255.255.0*.
2. Go to *Network* > *SD-WAN*, select the *SD-WAN Zones* tab, and click *Create New* > *SD-WAN Member*.

Overlay

VPN overlays are utilized when traffic must travel securely between company sites, such as between a datacenter and a remote branch. In this example, the single site is concerned with only internet-destined traffic. Therefore there is no use for any VPN overlays.

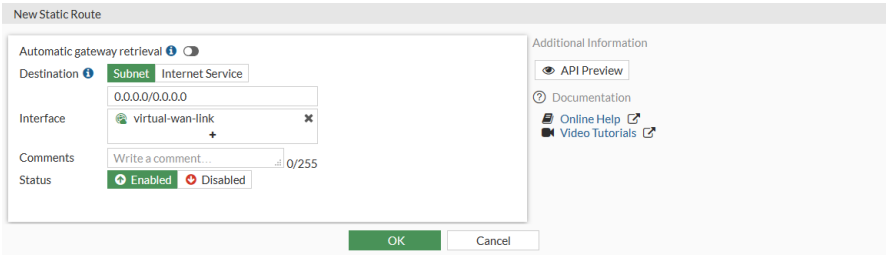
See SD-WAN cloud on-ramp on page 1567 for an example of adding an overlay to a single site SD-WAN implementation

Routing

You must configure a default route for the SD-WAN. The default gateways for each SD-WAN member interface do not need to be defined in the static routes table. FortiGate will decide what route or routes are preferred using Equal Cost Multi-Path (ECMP) based on distance and priority.

To create a static route for SD-WAN:

1. Go to *Network* > *Static Routes*.
2. Click *Create New*. The *New Static Route* page opens.
3. Set *Destination* to *Subnet*, and leave the IP address and subnet mask as *0.0.0.0/0.0.0.0*.
4. In the *Interface* field select an SD-WAN zone.



5. Ensure that *Status* is *Enabled*.
6. Click *OK*.

By default, a static route or the default route outgoing through an SD-WAN zone have an administrative distance of *1*.

To change the default distance in the CLI:

```
config router static
    edit <static-route-entry>
        set distance <AD>
    next
end
```

Security

SD-WAN zones can be used in policies as source and destination interfaces. Individual SD-WAN members cannot be used in policies.

You must configure a policy that allows traffic from your organization's internal network to the SD-WAN zone. Policies configured with the SD-WAN zone apply to all SD-WAN interface members in that zone.

To create a firewall policy for SD-WAN:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Click *Create New*. The *New Policy* page opens.
3. Configure the following:

Name	Enter a name for the policy.
Incoming Interface	*internal*
Outgoing Interface	*virtual-wan-link*
Source	*all*
Destination	*all*
Schedule	*always*
Service	*ALL*
Action	*ACCEPT*
Firewall / Network Options	Enable *NAT* and set *IP Pool Configuration* to *Use Outgoing Interface Address*.
Security Profiles	Apply profiles as required.
Logging Options	Enable *Log Allowed Traffic* and select *All Sessions*. This allows you to verify results later.

4. Enable the policy, then click *OK*.

SD-WAN

This section includes the following topics:

- Members and zones on page 1044
- Performance SLAs on page 1045
- SD-WAN rules on page 1046

Members and zones

First, SD-WAN must be enabled and member interfaces must be selected and added to a zone. The selected FortiGate interfaces can be of any type (physical, aggregate, VLAN, IPsec, and others), but must be removed

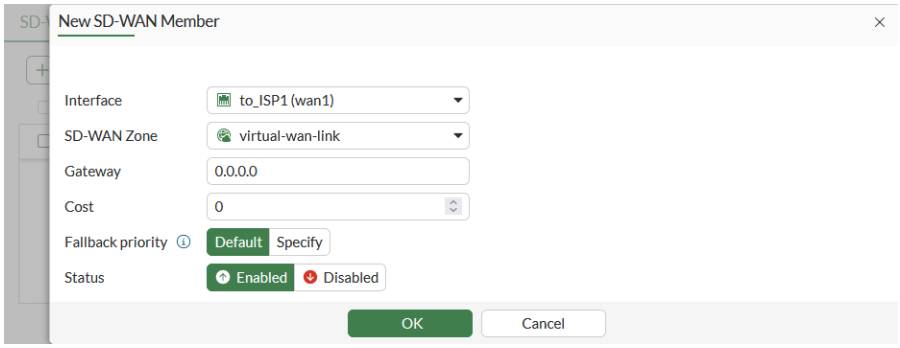
from any other configurations on the FortiGate.

In this step, two interfaces are configured and added to the default SD-WAN zone (virtual-wan-link) as SD-WAN member interfaces. This example uses a mix of static and dynamic IP addresses; your deployment could also use only one or the other.

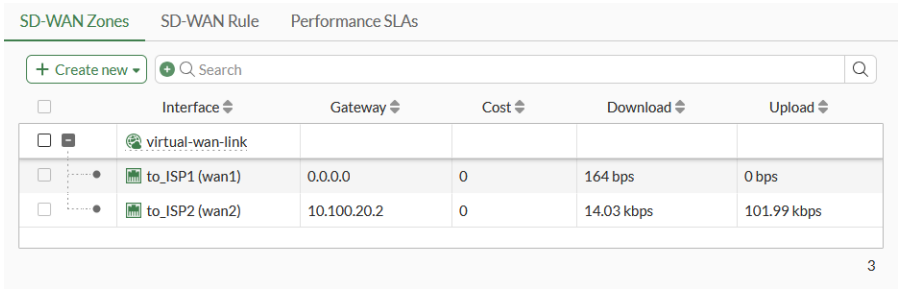
Once the SD-WAN members are created and added to a zone, the zone can be used in firewall policies, and the whole SD-WAN can be used in static routes.

To configure SD-WAN members:

1. Set the *Interface* to *wan1*.
2. Leave *SD-WAN Zone* as *virtual-wan-link*.
3. As wan1 uses DHCP, leave *Gateway* set to *0.0.0.0*.
   If IPv6 visibility is enabled in the GUI, an IPv6 gateway can also be added for each member. See Feature visibility on page 3952 for details.
4. Leave *Cost* as *0*.
   The *Cost* field is used by the Lowest Cost (SLA) strategy. The link with the lowest cost is chosen to pass traffic. The lowest possible *Cost* is *0*.

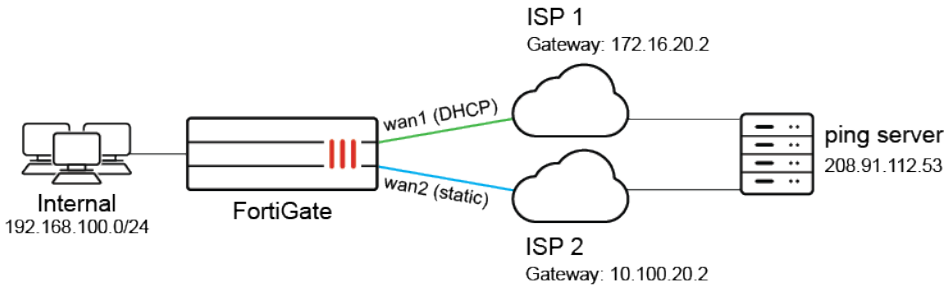


5. Set *Status* to *Enable*, and click *OK*.
6. Repeat the above steps for wan2, setting *Gateway* to the ISP's gateway: *10.100.20.2*.



Performance SLAs

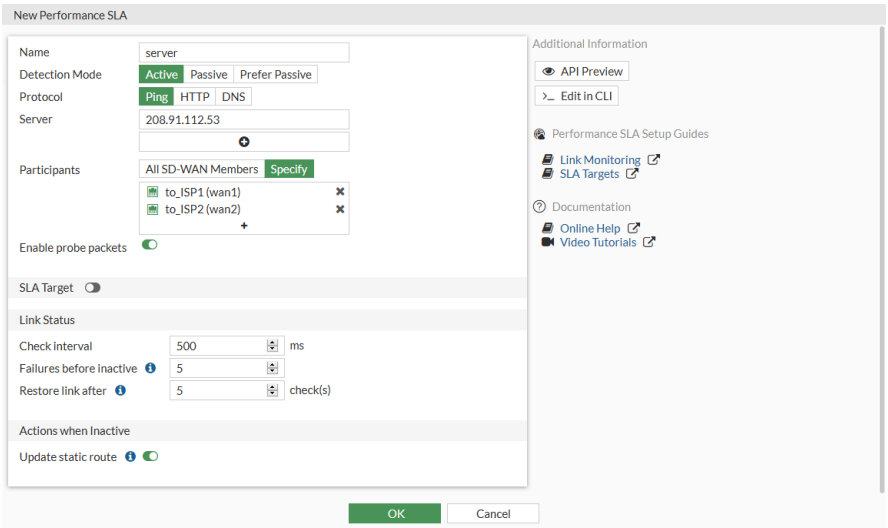
Performance SLA link monitoring measures the health of links that are connected to SD-WAN member interfaces by sending probing signals through each link to a server, and then measuring the link quality based on latency, jitter, and packet loss. If a link is broken, the routes on that link are removed and traffic is routed through other links. When the link is working again, the routes are re-enabled. This prevents traffic being sent to a broken link and lost.



In this example, the detection server IP address is 208.91.112.53. A performance SLA is created so that, if ping fails per the metrics defined, the routes to that interface are removed and traffic is detoured to the other interface. The ping protocol is used, but other protocols could also be selected as required.

To configure a performance SLA:

1. Go to *Network* > *SD-WAN*, select the *Performance SLAs* tab, and click *Create New*.
2. Enter a name for the SLA and set *Protocol* to *Ping*.
3. In the *Server* field, enter the detection server IP address (208.91.112.53 in this example).
4. In the *Participants* field, select *Specify* and add wan1 and wan2.



   SLA targets are not required for link monitoring.
5. Configure the required metrics in *Link Status*.
6. Ensure that *Update static route* is enabled. This disables static routes for the inactive interface and restores routes on recovery.
7. Click *OK*.

SD-WAN rules

SD-WAN rules define specific routing options to route traffic to an SD-WAN member.

If no routing rules are defined, the default *Implicit* rule is used. It can be configured to use one of five different load balancing algorithms. See Implicit rule on page 1114 for more details and examples.

This example shows four methods to equally balance traffic between the two WAN connections. Go to *Network > SD-WAN*, select the *SD-WAN Rules* tab, and edit the *sd-wan* rule to select the method that is appropriate for your requirements.

- *Source IP* (CLI command: `source-ip-based`):

  Select this option to balance traffic equally between the SD-WAN members according to a hash algorithm based on the source IP addresses.
- *Session* (`weight-based`):

  Select this option to balance traffic equally between the SD-WAN members by the session numbers ratio among its members. Use weight 50 for each of the 2 members.
- *Source-Destination IP* (`source-dest-ip-based`):

  Select this option to balance traffic equally between the SD-WAN members according to a hash algorithm based on the source and destination IP addresses.
- *Volume* (`measured-volume-based`):

  Select this option to balance traffic equally between the SD-WAN members according to the bandwidth ratio among its members.

Results

The following GUI pages show the function of the SD-WAN and can be used to confirm that it is setup and running correctly:

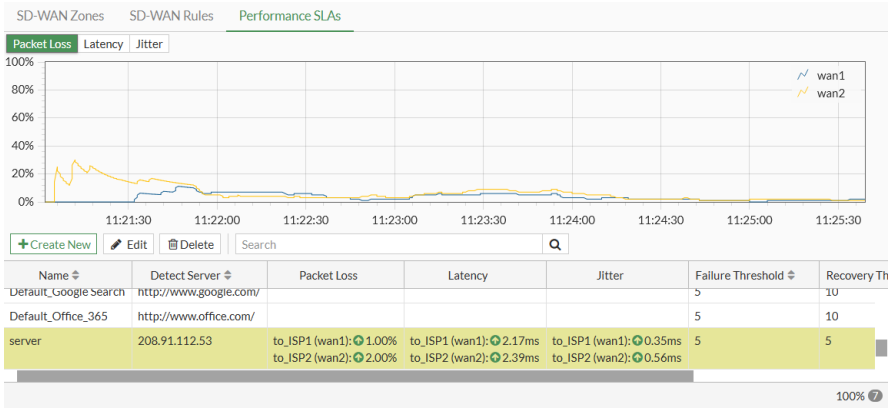
- Performance SLA on page 1047
- Routing table on page 1049
- Firewall policy on page 1049

Performance SLA

Go to *Network > SD-WAN*, select the *Performance SLAs* tab, and select the SLA from the table (*server* in this example) to view the packet loss, latency, and jitter on each SD-WAN member in the health check server.

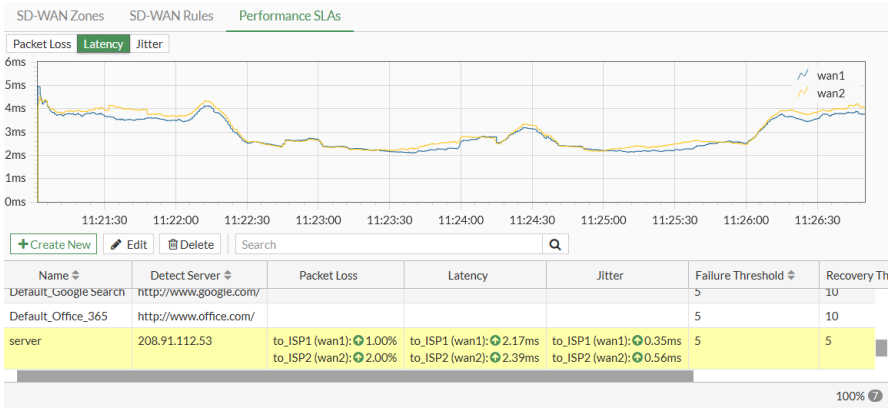
Packet loss

Select *Packet Loss* to see the percentage of packets lost for each member.



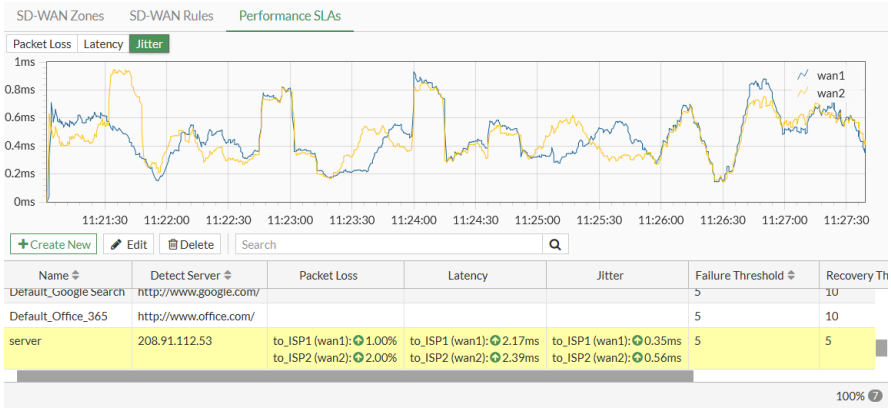
Latency

Select *Latency* to see the current latency, in milliseconds, for each member.



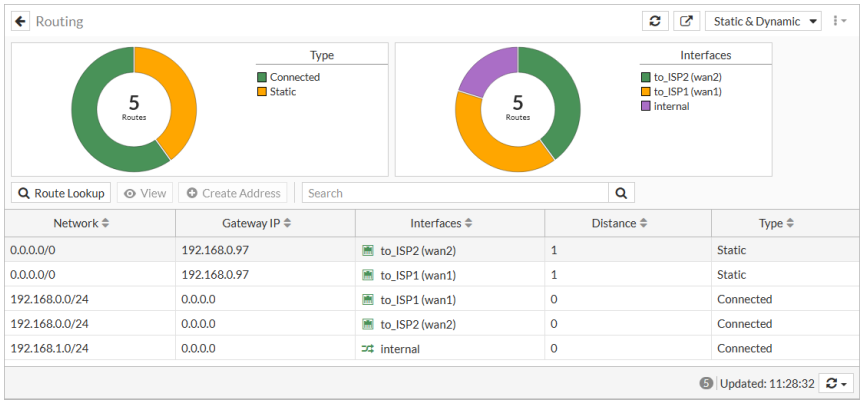
Jitter

Select *Jitter* to see the jitter, in milliseconds, for each member.



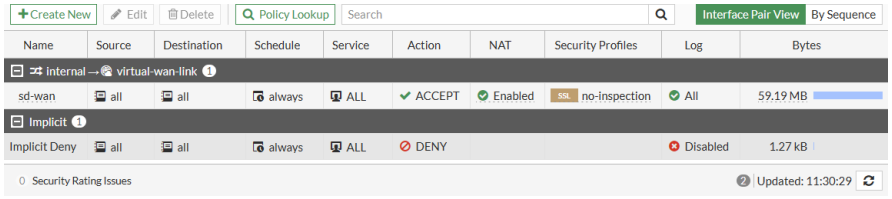
Routing table

Go to *Dashboard* > *Network Monitor* > *Routing* to review all static and dynamic routes. For more information about the widget, see Routing monitor on page 157.



Firewall policy

Go to *Policy & Objects* > *Firewall Policy* to review the SD-WAN policy.



Configuring SD-WAN in the CLI

This example can be entirely configured using the CLI.

To configure SD-WAN in the CLI:

1. Configure the wan1 and wan2 interfaces:

```
config system interface
    edit "wan1"
        set alias to_ISP1
        set mode dhcp
        set distance 10
    next
    edit "wan2"
        set alias to_ISP2
        set ip 10.100.20.1 255.255.255.0
    next
end
```

2. Enable SD-WAN and add the interfaces as members:

```
config system sdwan
    set status enable
    config members
        edit 1
            set interface "wan1"
        next
        edit 2
            set interface "wan2"
            set gateway 10.100.20.2
        next
    end
end
```

If no SD-WAN zone is specified, members are added to the default *virtual-wan-link* zone.

3. Create a static route for SD-WAN:

```
config router static
    edit 1
        set sdwan-zone "virtual-wan-link"
    next
end
```

4. Select the implicit SD-WAN algorithm:

```
config system sdwan
    set load-balance-mode {source-ip-based | weight-based | source-dest-ip-based | measured-
volume-based}
end
```

5. Create a firewall policy for SD-WAN:

```
config firewall policy
    edit <policy_id>
        set name <policy_name>
        set srcintf "internal"
        set dstintf "virtual-wan-link"
        set srcaddr all
        set dstaddr all
        set action accept
        set schedule always
        set service ALL
        set utm-status enable
        set ssl-ssh-profile <profile_name>
        set av-profile <profile_name>
        set webfilter-profile <profile_name>
        set dnsfilter-profile <profile_name>
        set emailfilter-profile <profile_name>
        set ips_sensor <sensor_name>
        set application-list <app_list>
```

```
        set voip-profile <profile_name>
        set logtraffic all
        set nat enable
        set status enable
    next
end
```

6. Configure a performance SLA:

```
config system sdwan
    config health-check
        edit "server"
            set server "208.91.112.53"
            set update-static-route enable
            set members 1 2
        next
    end
end
```

Results

To view the routing table:

```
# get router info routing-table all

Routing table for VRF=0
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

S*      0.0.0.0/0 [1/0] via 172.16.20.2, wan1
                  [1/0] via 10.100.20.2, wan2
C       10.100.20.0/24 is directly connected, wan2
C       172.16.20.2/24 is directly connected, wan1
C       192.168.0.0/24 is directly connected, internal
```

To diagnose the Performance SLA status:

```
FGT # diagnose sys sdwan health-check
Health Check(server):
Seq(1): state(alive), packet-loss(0.000%) latency(15.247), jitter(5.231) sla_map=0x0
Seq(2): state(alive), packet-loss(0.000%) latency(13.621), jitter(6.905) sla_map=0x0
```

SD-WAN members and zones

SD-WAN bundles interfaces together into zones. Interfaces are first configured as SD-WAN members. This does not change the interface, it just allows SD-WAN to reference the interface as a member. SD-WAN member interfaces can be any interface supported by FortiGates, such as physical ports, VLAN interfaces, LAGs, IPsec tunnels, GRE tunnels, IPIP tunnels, and FortiExtender interfaces. Once SD-WAN members are configured, they can be assigned to a zone. Zones are used in policies as source and destination interfaces, in static routes, and in SD-WAN rules.

Multiple zones can be used to group SD-WAN interfaces for logical scenarios, such as overlay and underlay interfaces. Using multiple zones in policies allows for more granular control over functions like resource access and UTM access. Individual SD-WAN member interfaces cannot be used directly in policies, but they can be moved between SD-WAN zones at any time. If a member interface requires a special SD-WAN consideration, it can be put into an SD-WAN zone by itself.

SD-WAN zones and members can be used in IPv4 and IPv6 static routes to make route configurations more flexible. SD-WAN zones and members can be used in SD-WAN rules to simplify the rule configuration. See Specify an SD-WAN zone in static routes and SD-WAN rules on page 1058 for more information.

SD-WAN zones can also be specified as interfaces in Local In policies, DoS policies, Multicast policies, TTL policies, and central SNAT maps. See Specify SD-WAN zones in some policies on page 1065 for more information.

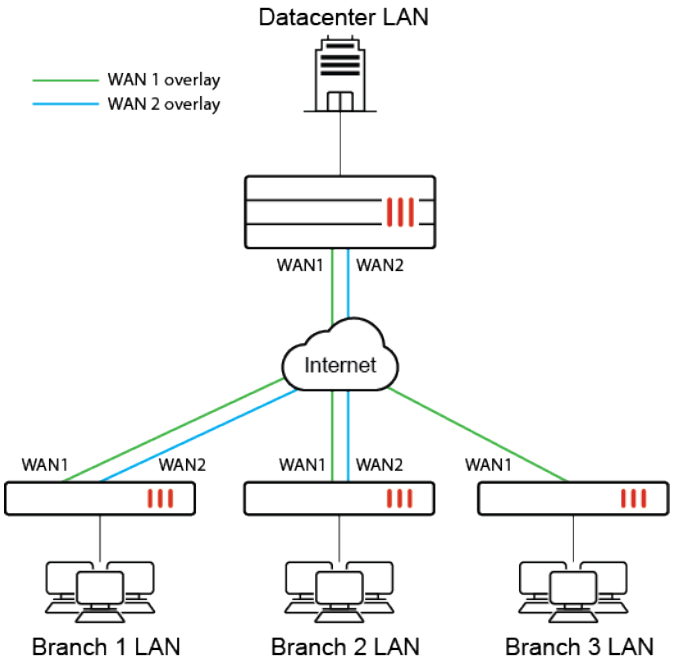
When the Security Fabric is configured, SD-WAN zones are included in the Security Fabric topology views.

Starting in FortiOS 7.6.5, some FortiGate models with two WAN interfaces have a default SD-WAN configuration. See Default SD-WAN configuration on FortiGate models with two WAN ports on page 131 for more information.

Topology

This topology is used in the following procedures:

- Configuring SD-WAN member interfaces
- Configuring SD-WAN zones
- Using SD-WAN zones



Configuring SD-WAN member interfaces

When configuring SD-WAN zones and members, it does not matter what order they are defined. In this example, the members are defined first, and they will be placed temporarily in the default zone called virtual-wan-link. A zone must be defined when creating a member, and the overlay and underlay zones will created in the next procedure. It is standard practice to create SD-WAN members for each underlay and overlay interface, as most SD-WAN implementations apply SD-WAN intelligence to both underlay and overlay networks.

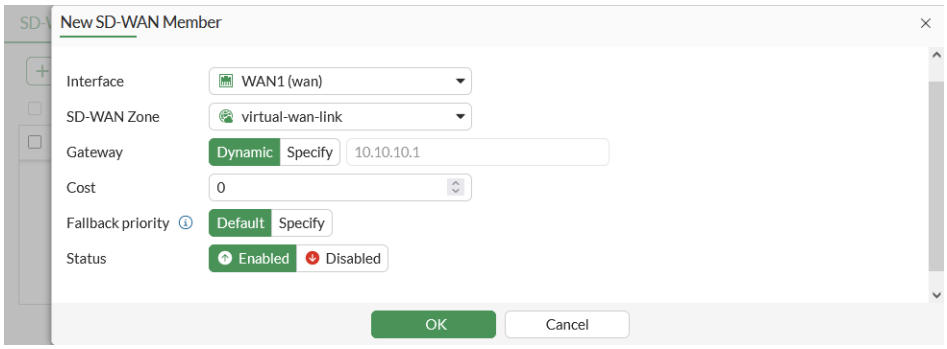
The following options can be configured for SD-WAN members:

GUI option	CLI option	Description
Interface	`interface`	Select the interface to use as an SD-WAN member. Optionally, select *None* in the GUI to not use an interface yet.
SD-WAN Zone	`zone`	Select the destination zone if it exists at the time of member creation. Otherwise, the default virtual-wan-link zone is applied. A new zone can be created within the GUI dropdown field.
Gateway/IPv6 Gateway	`gateway/gateway6`	Enter the default gateway for the interface. For interfaces that already have a default gateway, such as those configured using DHCP, this field is pre-populated in the GUI.

GUI option	CLI option	Description
Cost	`cost`	Enter the cost of the interface for services in SLA mode (0 - 4294967295, default = 0). A lower cost has a higher preference.
Fallback priority	`priority`	Enter the priority of the interface for IPv4 (1 - 65535, default = 1). The priority is used in the static route created for the SD-WAN member interface and in SD-WAN rules (including the implicit rule). When priority is used to determine the best route, the lower value takes precedence.
Status	`status`	Enable or disable the interface in SD-WAN.
n/a	`source/source6`	Set the source IP address used in the health check packet to the server.

To configure the SD-WAN members and add them to the default zone in the GUI:

1. Go to *Network* > *SD-WAN*, select the *SD-WAN Zones* tab, and click *Create New* > *SD-WAN Member*.
2. Set the *Interface* to *WAN1*.
3. Leave the *SD-WAN Zone* as *virtual-wan-link*.



4. Click *OK*.
5. Repeat these steps to create SD-WAN members for the WAN2, VPN1, and VPN2 interfaces.

To configure the SD-WAN members and add them to the default zone in the CLI:

```
config system sdwan
    config members
        edit 1
            set interface "WAN1"
            set zone "virtual-wan-link"
        next
        edit 2
            set interface "WAN2"
            set zone "virtual-wan-link"
```

```
        next
        edit 3
            set interface "VPN1"
            set zone "virtual-wan-link"
        next
        edit 4
            set interface "VPN2"
            set zone "virtual-wan-link"
        next
    end
end
```

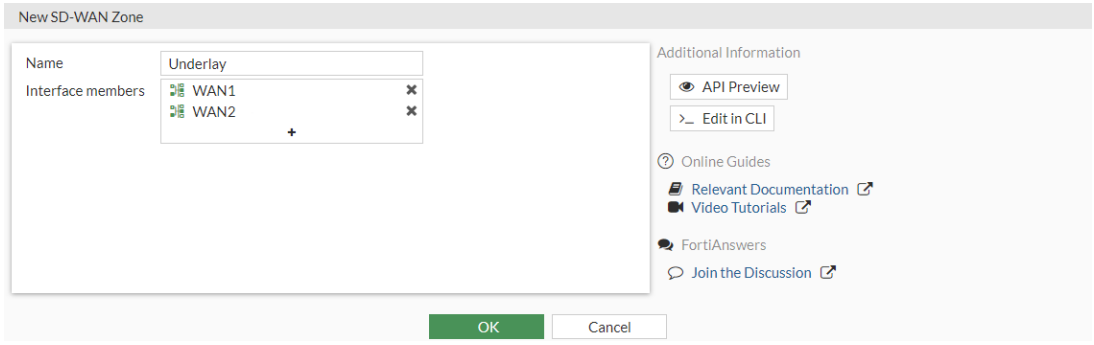
Configuring SD-WAN zones

While SD-WAN zones are primarily used to logically group interfaces that are often used for the same purpose (such as WAN1 and WAN2), sometimes an SD-WAN zone can have a single member. This is due to the constraint that SD-WAN members may not be referenced directly in policies; however, SD-WAN members can be referenced directly in SD-WAN rules.

In this example, two zones named Overlay and Underlay are configured, and the member interfaces are added to their respective zones.

To configure the SD-WAN zones in the GUI:

1. Go to *Network* > *SD-WAN* and select the *SD-WAN Zones* tab.
2. Click *Create New* > *SD-WAN Zone*.
3. Enter the *Name*, *Underlay*.
4. Set the *Interface members* to *WAN1* and *WAN2*.



5. Click *OK*.
6. Repeat these steps to configure the *Overlay* zone with members VPN1 and VPN2.

To configure the SD-WAN zones in the CLI:

1. Configure the SD-WAN zones:

```
config system sdwan
    config zone
```

```
        edit "Overlay"
        next
        edit "Underlay"
        next
    end
end
```

2. Add the member interfaces to their respective zones:

```
config system sdwan
    config members
        edit 1
            set interface WAN1
            set zone "Underlay"
        next
        edit 2
            set interface WAN2
            set zone "Underlay"
        next
        edit 3
            set interface VPN1
            set zone "Overlay"
        next
        edit 4
            set interface VPN2
            set zone "Overlay"
        next
    end
end
```

In the `config zone` settings, there is a `service-sla-tie-break` parameter that includes three options for the tie-break method used when multiple interfaces in a zone are eligible for traffic:

- `cfg-order`: members that meet the SLA are selected in the order they are configured (default).
- `fib-best-match`: members that meet the SLA are selected that match the longest prefix in the routing table.
- `input-device`: members that meet the SLA are selected by matching the input device.

See Overlay stickiness on page 1496 for more information.

Using SD-WAN zones

Once SD-WAN zones are defined, they can be used in firewall policies. This section covers three policy scenarios:

- Datacenter resource access
- Direct internet access
- Remote internet access

SD-WAN zones are a critical component of SD-WAN rules. See Fields for configuring WAN intelligence on page 1110 for more information.

Datacenter resource access

Datacenter resources are made available through the VPN branches or overlay. In this example, there are two SD-WAN members in the overlay zone that the branch FortiGate can use to route traffic to and from the datacenter resource. The overlay zone is used as the destination in the firewall policy.

To configure the firewall policy:

1. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
2. Configure the following settings:

Name	*DC_Access*
Incoming Interface	*LAN*
Outgoing Interface	*Overlay*
Source	*Branch_LAN*
Destination	*DC_LAN*
Action	*ACCEPT*

3. Configure the other settings as needed.
4. Click *OK*.

This firewall policy allows traffic to any interfaces included in the zone. The SD-WAN rules contain the intelligence used to select which members in the zone to use.

Direct internet access

Direct internet access (DIA) is how a branch may access resources contained on the public internet. This can be non-business resources (such as video streaming sites), or publically available business resources (such as vendor portals).

To configure the firewall policy:

1. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
2. Configure the following settings:

Name	*DIA*
Incoming Interface	*LAN*
Outgoing Interface	*Underlay*

Source	*Branch_LAN*
Destination	*all*
Action	*ACCEPT*

3. Configure the other settings as needed.
4. Click *OK*.

Remote internet access

Remote internet access (RIA) is the ability for a branch location to route public internet access requests across the overlay and out one of the hub's (or datacenter's) WAN interfaces. This option is effective when a branch has a WAN circuit with a local ISP and a second circuit that is private, such as MPLS. When the WAN circuit goes down, it is possible to send traffic through the hub using the MPLS overlay.

To configure the firewall policy:

1. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
2. Configure the following settings:

Name	*RIA*
Incoming Interface	*LAN*
Outgoing Interface	*Overlay*
Source	*Branch_LAN*
Destination	*all*
Action	*ACCEPT*

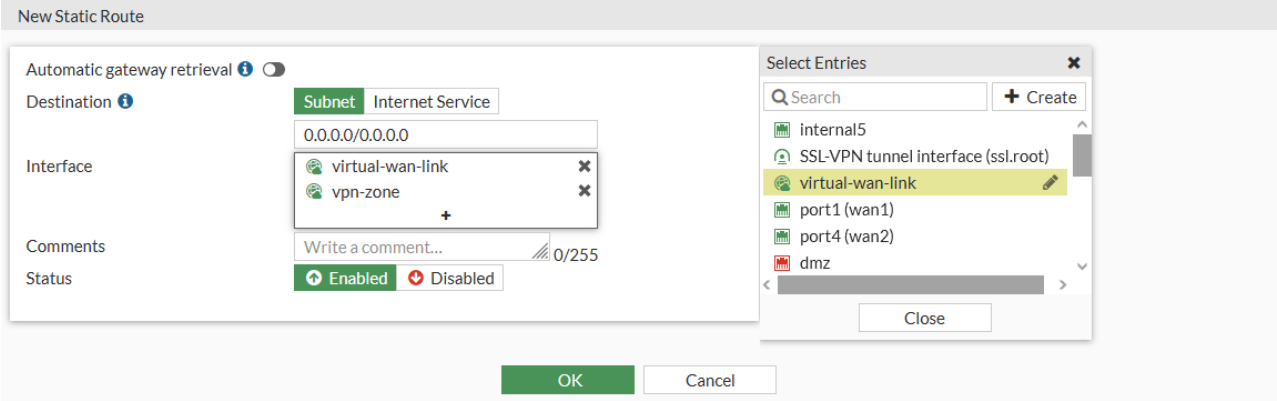
3. Configure the other settings as needed.
4. Click *OK*.

Specify an SD-WAN zone in static routes and SD-WAN rules

SD-WAN zones can be used in IPv4 and IPv6 static routes, and in SD-WAN service rules. This makes route configuration more flexible, and simplifies SD-WAN rule configuration.

To configure an SD-WAN zone in a static route in the GUI:

1. Go to *Network* > *Static Routes*
2. Edit an existing static route, or click *Create New* to create a new route.
3. Set *Interface* to one or more SD-WAN zones.



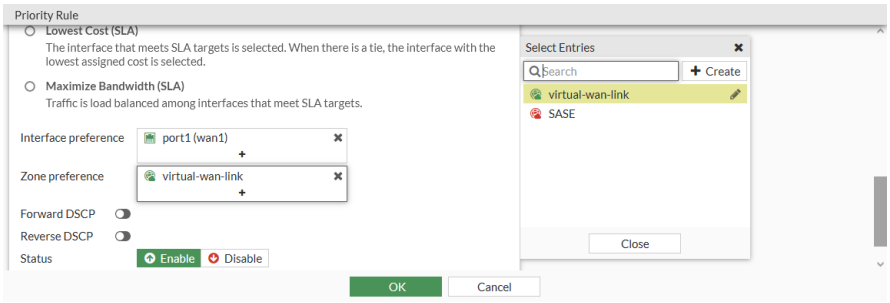
4. Configure the remaining settings are required.
5. Click *OK*.

To configure an SD-WAN zone in a static route in the CLI:

```
config router {static | static6}
    edit 1
        set sdwan-zone <zone> <zone> ...
    next
end
```

To configure an SD-WAN zone in an SD-WAN rule in the GUI:

1. Go to *Network* > *SD-WAN* and select the *SD-WAN Rules* tab
2. Edit an existing rule, or click *Create New* to create a new rule.
3. In the *Zone preference* field add one or more SD-WAN zones.



4. Configure the remaining settings are needed.
5. Click *OK*.

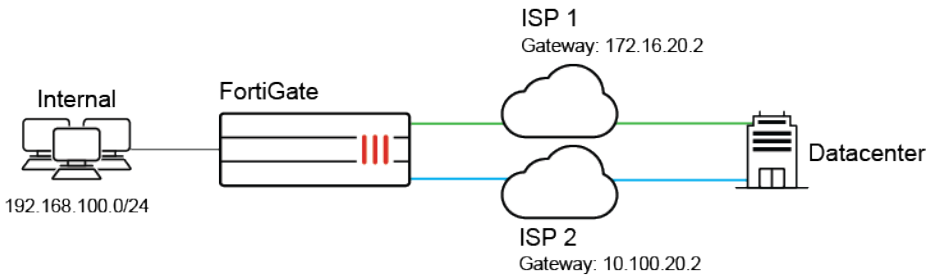
To configure an SD-WAN zone in an SD-WAN rule in the CLI:

```
config system sdwan
    config service
        edit 1
            set priority-zone <zone>
        next
```

```
    end
end
```

Examples

In these two examples, three SD-WAN members are created. Two members, port13 and port15, are in the default zone (*virtual-wan-link*), and the third member, to_FG_B_root, is in the *SASE* zone.



Example 1

In this example:

- Two service rules are created. Rule 1 uses the *virtual-wan-link* zone, and rule 2 uses the *SASE* zone.
- Two IPv4 static routes are created. The first route uses the *virtual-wan-link* zone, and the second route uses the *SASE* zone.

To configure the SD-WAN:

1. Assign port13 and port15 to the *virtual-wan-link* zone and to_FG_B_root to the *SASE* zone:

```
config system sdwan
    set status enable
    config members
        edit 1
            set interface "port13"
            set zone "virtual-wan-link"
            set gateway 10.100.1.1
        next
        edit 2
            set interface "port15"
            set zone "virtual-wan-link"
            set gateway 10.100.1.5
        next
        edit 3
            set interface "to_FG_B_root"
            set zone "SASE"
        next
    end
end
```

2. Create two service rules, one for each SD-WAN zone:

```
config system sdwan
    config service
        edit 1
            set dst "10.100.20.0"
            set priority-zone "virtual-wan-link"
        next
        edit 2
            set internet-service enable
            set internet-service-name "Fortinet-FortiGuard"
            set priority-zone "SASE"
        next
    end
end
```

3. Configure static routes for each of the SD-WAN zones:

```
config router static
    edit 1
        set distance 1
        set sdwan-zone "virtual-wan-link"
    next
    edit 2
        set dst 172.16.109.0 255.255.255.0
        set distance 1
        set sdwan-zone "SASE"
    next
end
```

To verify the results:

1. Check the service rule 1 diagnostics:

```
# diagnose sys sdwan service4 1

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
  Members(2):
    1: Seq_num(1 port13), alive, selected
    2: Seq_num(2 port15), alive, selected
  Dst address(1):
        10.100.20.0-10.100.20.255
```

Both members of the *virtual-wan-link* zone are selected. In manual mode, the interface members are selected based on the member configuration order. In SLA and priority mode, the order depends on the link status. If all of the link statuses pass, then the members are selected based on the member configuration order.

2. Check the service rule 2 diagnostics:

```
# diagnose sys sdwan service4 2

Service(2): Address Mode(IPV4) flags=0x200 use-shortcut-sla
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
```

```
  Members(1):
    1: Seq_num(3 to_FG_B_root), alive, selected
  Internet Service(1): Fortinet-FortiGuard(1245324,0,0,0)
```

The member of the *SASE* zone is selected.

3. Review the routing table:

```
# get router info routing-table static
Routing table for VRF=0
S*      0.0.0.0/0 [1/0] via 10.100.1.1, port13
                  [1/0] via 10.100.1.5, port15
S       172.16.109.0/24 [1/0] via 172.16.206.2, to_FG_B_root
```

The default gateway has the members from the *virtual-wan-link* zone, and the route to 172.16.10.9.0/24 has the single member from the *SASE* zone.

Example 2

In this example, two IPv6 static routes are created. The first route uses the *virtual-wan-link* zone, and the second route uses the *SASE* zone.

To configure the SD-WAN:

1. Configure port13 and port15 with IPv6 addresses and assign them to the *virtual-wan-link* zone, and assign to_FG_B_root to the *SASE* zone:

```
config system sdwan
    set status enable
    config members
        edit 1
            set interface "port13"
            set zone "virtual-wan-link"
            set gateway6 2004:10:100:1::1
            set source6 2004:10:100:1::2
        next
        edit 2
            set interface "port15"
            set zone "virtual-wan-link"
            set gateway6 2004:10:100:1::5
            set source6 2004:10:100:1::6
        next
        edit 3
            set interface "to_FG_B_root"
            set zone "SASE"
        next
    end
end
```

2. Configure IPv6 static routes for each of the SD-WAN zones:

```
config router static6
    edit 1
```

```
        set distance 1
        set sdwan-zone "virtual-wan-link"
    next
    edit 2
        set dst 2003:172:16:109::/64
        set distance 1
        set sdwan-zone "SASE"
    next
end
```

To verify the results:

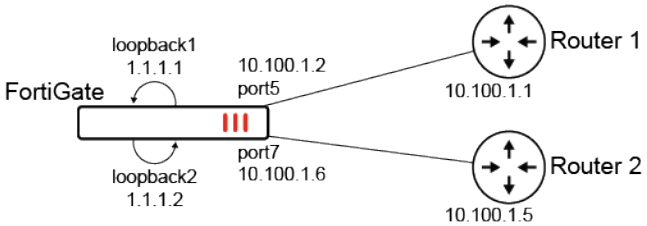
1. Review the routing table:

```
# get router info6 routing-table static
Routing table for VRF=0
S*      ::/0 [1/0] via 2004:10:100:1::1, port13, 00:20:51, [1024/0]
             [1/0] via 2004:10:100:1::5, port15, 00:20:51, [1024/0]
S       2003:172:16:109::/64 [1/0] via ::ac10:ce02, to_FG_B_root, 00:20:51, [1024/0]
S       2003:172:16:209::/64 [5/0] via ::ac10:ce02, to_FG_B_root, 14:40:14, [1024/0]
```

The IPv6 default route includes the members from the *virtual-wan-link* zone, and the route to 2003:172:16:109::/64 has the single member from the *SASE* zone.

Defining a preferred source IP for local-out egress interfaces on SD-WAN members

The preferred source IP can be configured on SD-WAN members so that local-out traffic is sourced from that IP. In the following example, two SD-WAN members (port5 and port7) will use loopback1 and loopback2 as sources instead of their physical interface address. A static route is created for destination 200.0.0.0/24 to use the virtual-wan-link. In turn, the FortiGate will create two ECMP routes to the member gateways and source the traffic from the loopback IPs.



To configure preferred source IPs for SD-WAN members:

1. Configure the SD-WAN members and other settings:

```
config system sdwan
    set status enable
    config zone
```

```
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "port5"
            set gateway 10.100.1.1
            set preferred-source 1.1.1.1
            set source 1.1.1.1
        next
        edit 2
            set interface "port7"
            set gateway 10.100.1.5
            set preferred-source 1.1.1.2
            set source 1.1.1.2
        next
    end
end
```

In the SD-WAN `config members` settings, configuring the `source` for the health check probes is still required. SD-WAN adds dedicated kernel routes (proto=17) for the health checks using the interface IP or source IP when specified. To view the kernel routes, use `diagnose ip route list`.

2. Configure the static route:

```
config router static
    edit 2000
        set dst 200.0.0.0 255.255.255.0
        set distance 1
        set sdwan-zone "virtual-wan-link"
    next
end
```

To verify the configuration:

1. Verify the kernel routing table for 200.0.0.0/24:

```
# get router info kernel | grep -A 2 200.0.0.0/24
tab=254 vf=0 scope=0 type=1 proto=11 prio=1 0.0.0.0/0.0.0.0/0->200.0.0.0/24 pref=0.0.0.0
        gwy=10.100.1.1 flag=14 hops=255 oif=13(port5) pref=1.1.1.1
        gwy=10.100.1.5 flag=14 hops=254 oif=15(port7) pref=1.1.1.2
```

2. Verify the routing table for 200.0.0.0/24:

```
# get router info routing-table details 200.0.0.0/24
Routing table for VRF=0
Routing entry for 200.0.0.0/24
  Known via "static", distance 1, metric 0, best
  * vrf 0 10.100.1.1, via port5, prefsrc 1.1.1.1
  * vrf 0 10.100.1.5, via port7, prefsrc 1.1.1.2
```

3. Run a sniffer trace after some traffic passes.
   a. When traffic leaves port5:

```
# diagnose sniffer packet any "host 200.0.0.1" 4
interfaces=[any]
filters=[host 200.0.0.1]
6.592488 port5 out 1.1.1.1 -> 200.0.0.1: icmp: echo request
7.592516 port5 out 1.1.1.1 -> 200.0.0.1: icmp: echo request
8.592532 port5 out 1.1.1.1 -> 200.0.0.1: icmp: echo request
```

   b. When traffic leaves port7:

```
# diagnose sniffer packet any "host 200.0.0.1" 4
interfaces=[any]
filters=[host 200.0.0.1]
75.664173 port7 out 1.1.1.2 -> 200.0.0.1: icmp: echo request
76.664194 port7 out 1.1.1.2 -> 200.0.0.1: icmp: echo request
```

   Traffic exiting each interface is sourced from the corresponding loopback IP.

Individual source IP commands, such as FortiManager source IP and FortiAnalyzer source IP, take precedence over these preferred-source configurations.

Specify SD-WAN zones in some policies

SD-WAN zones can be specified as interfaces in Local In policies, DoS policies, Multicast policies, TTL policies, and central SNAT maps. This simplifies policy management and improves operational efficiency.

```
config firewall local-in-policy
    edit <id>
        set intf <SD-WAN zone>
    next
end
config firewall DoS-policy
    edit <id>
        set interface <SD-WAN zone>
    next
end
config firewall interface-policy
    edit <id>
        set interface <SD-WAN zone>
    next
end
config firewall multicast-policy
    edit <id>
        set srcintf <SD-WAN zone>
        set dstintf <SD-WAN zone>
    next
end
```

```
config firewall ttl-policy
    edit <id>
        set srcintf <SD-WAN zone>
    next
end
config firewall central-snat-map
    edit <id>
        set srcintf <SD-WAN zone>
        set dstintf <SD-WAN zone>
    next
end
```

Example

To configure a use an SD-WAN zone as an interface in some policies:

1. Configure an SD-WAN zone:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
        edit "test"
        next
    end
    config members
        edit 1
            set interface "agg1"
            set zone "test"
            set gateway 172.16.203.2
        next
        edit 2
            set interface "vlan100"
            set zone "test"
            set gateway 172.16.206.2
        next
    end
end
```

2. Use that SD-WAN zone as an interface in policies:

```
config firewall local-in-policy
    edit 1
        set intf "test"
        set srcaddr "172.16.205.0"
        set dstaddr "all"
        set service "ALL"
        set schedule "always"
    next
end
```

```
config firewall DoS-policy
    edit 1
        set interface "test"
        set srcaddr "172.16.205.0"
        set dstaddr "all"
        set service "ALL"
        config anomaly
            edit "tcp_syn_flood"
                set threshold 2000
            next
            edit "tcp_port_scan"
                set threshold 1000
            next
            edit "tcp_src_session"
                set threshold 5000
            next
            ...
        end
    next
end
```

```
config firewall interface-policy
    edit 1
        set interface "test"
        set srcaddr "172.16.205.0"
        set dstaddr "all"
        set service "ALL"
    next
end
```

```
config firewall multicast-policy
    edit 1
        set srcintf "test"
        set dstintf "any"
        set srcaddr "172.16.205.0"
        set dstaddr "all"
    next
end
```

```
config firewall ttl-policy
    edit 1
        set srcintf "test"
        set srcaddr "172.16.205.0"
        set service "ALL"
        set schedule "always"
        set ttl 5
    next
end
```

```
config firewall central-snat-map
    edit 1
```

```
        set srcintf any
        set dstintf "test"
        set orig-addr "all"
        set dst-addr "172.16.205.0"
    next
end
```

SD-WAN performance SLA

Performance SLAs are used to measure the health of links that are connected to SD-WAN member interfaces by either sending probing signals through each link to a server, or using session information that is captured by firewall policies (see Passive WAN health measurement on page 1081 for information), and measuring the link quality based on latency, jitter, and packet loss. If a link fails all of the health checks, the routes on that link are removed from the SD-WAN link load balancing group, and traffic is routed through other links. When the link passes SLA, the routes are reestablished. This prevents traffic from being sent to a broken link and getting lost.

The following topics provide instructions on configuring performance SLA:

- Performance SLA overview on page 1068
- Link health monitor on page 1073
- Monitoring performance SLA on page 1076
- Passive WAN health measurement on page 1081
- Passive health-check measurement by internet service and application on page 1087
- Mean opinion score calculation and logging in performance SLA health checks on page 1092
- SD-WAN application monitor using FortiMonitor on page 1095
- Classifying SLA probes for traffic prioritization on page 1099

Performance SLA overview

Performance SLAs consist of three parts:

- Health checks
- SLA targets
- Link status

Health checks

A health check is defined by a probe mode, protocol, and server. These three options specify what resource is being evaluated and how the evaluation is done. Each health check should be configured specifically for that resource, so the probe mode, protocol and server should be tailored for the particular service. For example, the health check for a VoIP service will differ than one for a database replication service.

Performance SLA participants are the interfaces that will be evaluated for a given health check. They must be SD-WAN member interfaces, but do not have to belong to the same zone. When selecting participants, only

select participants that you expect the service communications to use. For example, a health check for a corporate resource might only use the overlay to access the service. Therefore, you would only add the VPN interfaces as participants.

There are five predefined performance SLA profiles for newly created VDOMs or factory reset FortiGate devices: DNS, FortiGuard, Gmail, Google Search, and Office 365. These performance SLA profiles provide Fortinet recommended settings for common services. To complete the performance SLA configuration, add the participants for the service. You can adjust the default settings to suit your needs.

Bandwidth limits and traffic prioritization can be enabled using SLA probe classification. The `class-id` command can be used to assign a class ID to SLA probes. Class IDs then guarantee that assigned bandwidth is honored when traffic congestion occurs. See Classifying SLA probes for traffic prioritization on page 1099.

Probe mode

The probe mode can be set to active, passive, or prefer passive.

In active mode, the FortiGate sends a packet of the type specified by the protocol setting towards the defined server. This allows you to evaluate the path to the destination server using the protocol that matches the service provided by the server. Active probing does add some overhead in the form of health check probes (and additional configurations to define the probe type and server), but it has the benefit of constantly measuring the performance of the path to the server. This can be beneficial when reviewing historical data.

In passive mode, session information captured by firewall policies is used to determine latency, jitter, and packet loss. This has the added benefit of not generating additional traffic, and does not require the performance SLA to define a specific server for measurement. Instead, the SD-WAN rule must define the traffic to evaluate, and the firewall policy permitting the traffic must have a setting enabled. See Passive WAN health measurement on page 1081 and Passive health-check measurement by internet service and application on page 1087 for more information.

Prefer passive mode is a combination of active and passive modes. Health is measured using traffic when there is traffic, and using probes when there is no traffic. A protocol and server must be configured.

Protocol

Health checks support a variety of protocols and protocol specific options. The most commonly used protocols (ping, HTTP, and DNS) can be configured in the GUI when creating a new performance SLA on the *Network* > *SD-WAN* > *Performance SLAs* page. The following protocols and options can be configured in the CLI using the `set protocol <option>` parameter:

`ping`	Use PING to test the link with the server.
`tcp-echo`	Use TCP echo to test the link with the server.
`udp-echo`	Use UDP echo to test the link with the server.
`http`	Use HTTP-GET to test the link with the server.
`https`	Use HTTPS-GET to test the link with the server.
`twamp`	Use TWAMP to test the link with the server.

`dns`	Use DNS query to test the link with the server. The FortiGate sends a DNS query for an A Record and the response matches the expected IP address.
`tcp-connect`	Use a full TCP connection to test the link with the server. The method to measure the quality of the TCP connection can be: • `half-open`: FortiGate sends SYN and gets SYN-ACK. The latency is based on the round trip between SYN and SYN-ACK (default). • `half-close`: FortiGate sends FIN and gets FIN-ACK. The latency is based on the round trip between FIN and FIN-ACK.
`ftp`	Use FTP to test the link with the server. The FTP mode can be: • `passive`: The FTP health-check initiates and establishes the data connection (default). • `port`: The FTP server initiates and establishes the data connection.

SD-WAN health checks can generate traffic that becomes quite high as deployments grow. Take this into consideration when setting DoS policy thresholds. For details on setting DoS policy thresholds, refer to DoS policy on page 1901.

The default FortiGuard, Google Search, and Office 365 performance SLA profiles use HTTPS.

To use UDP-echo and TCP-echo as health checks:

```
config system sdwan
    set status enable
    config health-check
        edit "h4_udp1"
            set protocol udp-echo
            set port 7
            set server <server>
        next
        edit "h4_tcp1"
            set protocol tcp-echo
            set port 7
            set server <server>
        next
        edit "h6_udp1"
            set addr-mode ipv6
            set server "2032::12"
            set protocol udp-echo
            set port 7
        next
    end
end
```

To use DNS as a health check, and define the IP address that the response must match:

```
config system sdwan
    set status enable
    config health-check
        edit "h4_dns1"
            set protocol dns
            set dns-request-domain "ip41.forti2.com"
            set dns-match-ip 1.1.1.1
        next
        edit "h6_dns1"
            set addr-mode ipv6
            set server "2000::15.1.1.4"
            set protocol dns
            set port 53
            set dns-request-domain "ip61.xxx.com"
        next
    end
end
```

To use TCP Open (SYN/SYN-ACK) and TCP Close (FIN/FIN-ACK) to verify connections:

```
config system sdwan
    set status enable
    config health-check
        edit "h4_tcpconnect1"
            set protocol tcp-connect
            set port 443
            set quality-measured-method {half-open | half-close}
            set server <server>
        next
        edit "h6_tcpconnect1"
            set addr-mode ipv6
            set server "2032::13"
            set protocol tcp-connect
            set port 444
            set quality-measured-method {half-open | half-close}
        next
    end
end
```

To use active or passive mode FTP to verify connections:

```
config system sdwan
    set status enable
    config health-check
        edit "h4_ftp1"
            set protocol ftp
            set port 21
            set user "root"
            set password ***********
```

```
            set ftp-mode {passive | port}
            set ftp-file "1.txt"
            set server <server>
        next
        edit "h6_ftp1"
            set addr-mode ipv6
            set server "2032::11"
            set protocol ftp
            set port 21
            set user "root"
            set password ***********
            set ftp-mode {passive | port}
            set ftp-file "2.txt"
        next
    end
end
```

Health check probe packets support DSCP markers for accurate link performance evaluation for high priority applications. This allows the probe packet to match the real traffic it is providing measurements for, including how that traffic is shaped by upstream devices based on the DSCP markers.

To mark health check packets with DSCP:

```
config system sdwan
    config health-check
        edit <name>
            set diffservcode <6-bits_binary, 000000-111111>
            set protocol <option>
        next
    end
end
```

Server

An IP address or FQDN can be defined as the server that the probe packets will be sent to. Up to two servers can be defined this way. When two servers are provided, both must fail in order for the health check to fail. This is to avoid a scenario where one remote server is down and causes a false positive that the link is down. The FortiGate can still use the interface associated with this health check to reach the remaining healthy server.

The purpose of the server is not simply to measure the health of the link, but rather the health of the path to a resource. It is highly recommended to use an IP address or FQDN that reflects the resource so the traffic path is considered.

A server can only be used in one performance SLA at any given time.

SLA targets

SLA targets are a set of constraints that are used in SD-WAN rules to control the paths that traffic takes. The constraints are:

- Latency threshold: latency for SLA to make a decision, in milliseconds (0 - 10000000, default = 5).
- Jitter threshold: jitter for SLA to make a decision, in milliseconds (0 - 10000000, default = 5).
- Packet loss threshold: packet loss for SLA to make a decision, in percentage (0 - 100, default = 0).

These settings should be specific to the service whose performance is being considered. You should attempt to configure the constraints to be just under the maximum values for the application or service to function well. For example, if your application requires less than 100 ms latency, then you should configure the SLA target to be 90 ms. Misconfiguring these settings will cause the performance SLA to lose value. If the values are too tight, then you may have traffic flipping between links before necessary. If the values are too loose, then performance may be impacted and the FortiGate will do nothing about it.

In the GUI, one SLA target can be configured, but additional targets can be configured in the CLI. Once a second target is configured in the CLI, additional targets can be configured from the GUI. Multiple SLA targets can be configured where a server provides multiple services that have different values for acceptable performance. For example, Google provides a DNS service and entertainment services (YouTube), so it is necessary to configure multiple SLA targets in this case since you can only configure a server in one performance SLA.

Link status

The *Link Status* section of the performance SLA configuration consists of three settings that determine the frequency that the link is evaluated, and the requirements to be considered valid or invalid:

- *Check interval*: the interval in which the FortiGate checks the interface, in milliseconds (20 - 3600000, default = 500). If the protocol type is set to HTTP or HTTPS, the check interval is automatically increased to 120,000.
- *Failures before inactive*: the number of failed status checks before the interface shows as inactive (1 - 3600, default =5). This setting helps prevent flapping, where the system continuously transfers traffic back and forth between links.
- *Restore link after*: the number of successful status checks before the interface shows as active (1 - 3600, default = 5). This setting also helps prevent flapping.

When a participant becomes inactive, the performance SLA causes the FortiGate to withdraw all static routes associated with that interface. If there are multiple static routes using the same interface, they will all be withdrawn when the link monitor is failing.

Link health monitor

Performance SLA link health monitoring measures the health of links that are connected to SD-WAN member interfaces by either sending probing signals through each link to a server, or using session information that is captured on firewall policies (see Passive WAN health measurement on page 1081 for information), and measuring the link quality based on latency, jitter, and packet loss. If a link fails all of the health checks, the routes on that link are removed from the SD-WAN link load balancing group, and traffic is routed through other links. When the link is working again the routes are reestablished. This prevents traffic being sent to a broken link and lost.

When an SD-WAN member has multiple health checks configured, all of the checks must fail for the routes on that link to be removed from the SD-WAN link load balancing group.

Two health check servers can be configured to ensure that, if there is a connectivity issue, the interface is at fault and not the server. A server can only be used in one health check.

The FortiGate uses the first server configured in the health check server list to perform the health check. If the first server is unavailable, then the second server is used. The second server continues to be used until it becomes unavailable, and then the FortiGate returns to the first server, if it is available. If both servers are unavailable, then the health check fails.

You can configure the protocol that is used for status checks, including: Ping, HTTP, HTTPS. DNS, TCP echo, UDP echo, two-way active measurement protocol (TWAMP), TCP connect, and FTP. In the GUI, only Ping, HTTP, and DNS are available.

You can view link quality measurements by going to *Network > SD-WAN* and selecting the *Performance SLAs* tab. The table shows the default health checks, the health checks that you configured, and information about each health check. The values shown in the *Packet Loss*, *Latency*, and *Jitter* columns are for the health check server that the FortiGate is currently using. The green up arrows indicate that the server is responding, and does not indicate if the health checks are being met. See Results on page 1047 for more information.

To configure a link health monitor in the GUI:

1. Go to *Network > SD-WAN*, select the *Performance SLAs* tab, and click *Create New*.
2. Set a *Name* for the SLA.
3. If enabled in Feature Visibility, set the *IP Version*. *IPv6* does not support all of the protocols.
4. Set the *Probe mode*:
   - *Active*: Send probes to determine link quality.
   - *Passive*: Use traffic to determine link quality. Enable passive health checks in policies to allow measurement.
   - *Prefer Passive*: Same as passive mode, but send probes when there is no traffic.
5. Set the *Protocol* that you need to use for status checks: *Ping*, *HTTP*, or *DNS*.
6. Set *Server* to the IP addresses of up to two servers that all of the SD-WAN members in the performance SLA can reach. If the *Protocol* is *DNS*, set the *DNS Server* to either the same as the system DNS, or specify the primary and secondary DNS servers.
7. Set *Participants* to *All SD-WAN Members*, or select *Specify* to choose specific SD-WAN members.
8. Set *Enable probe packets* to enable or disable sending probe packets.
9. Configure *SLA Target*:

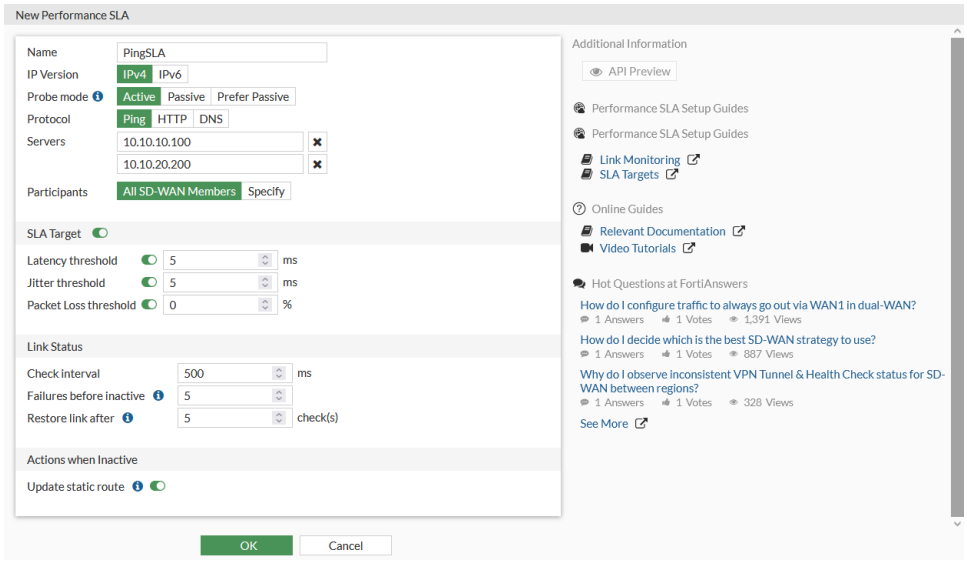
   If the health check is used in an SD-WAN rule that uses *Manual* or *Best Quality* strategies, enabling *SLA Target* is optional. If the health check is used in an SD-WAN rule that uses *Lowest Cost (SLA)* or *Maximum Bandwidth (SLA)* strategies, then *SLA Target* is enabled.

   When *SLA Target* is enabled, configure the following:
   - *Latency threshold*: Calculated based on last 30 probes (default = 5ms).
   - *Jitter threshold*: Calculated based on last 30 probes (default = 5ms).
   - *Packet Loss threshold*: Calculated based on last 100 probes (default = 0%).
10. In the *Link Status* section configure the following:
    - *Check interval*: the interval in which the FortiGate checks the interface, in milliseconds (20 - 3600000, default = 500).
    - *Failures before inactive*: The number of failed status checks before the interface shows as inactive (1 - 3600, default =5). This setting helps prevent flapping, where the system continuously transfers traffic back and forth between links
    - *Restore link after*: The number of successful status checks before the interface shows as active (1 - 3600, default = 5). This setting helps prevent flapping, where the system continuously transfers traffic

back and forth between links

11. In the *Actions when Inactive* section, enable *Update static route* to disable static routes for inactive interfaces and restore routes when interfaces recover.

> ⚠ When a performance SLA is inactive or out-of-SLA, SD-WAN will not send traffic through the link that the performance SLA is evaluating. Enabling *Update static route* applies the SD-WAN logic to every traffic flow, whether or not it is steered by SD-WAN. When you enable *Update static route*, you must carefully consider the consequence for overlay tunnels.



12. Click *OK*.

To configure a link health monitor in the CLI:

```
config system sdwan
    config health-check
        edit "PingSLA"
            set addr-mode {ipv4 | ipv6}
            set server <server1_IP_address> <server2_IP_address>
            set detect-mode {active | passive | prefer-passive | remote | agent-based}
            set protocol {ping | tcp-echo | udp-echo | http | https| twamp | dns | tcp-connect |
ftp}
            set ha-priority <integer>
            set probe-count <integer>
            set probe-packets {enable | disable}
            set probe-timeout <integer>
            set agent-probe-timeout <integer>
            set remote-probe-timeout <integer>
            set interval <integer>
            set failtime <integer>
            set recoverytime <integer>
            set diffservcode <binary>
            set update-static-route {enable | disable}
            set update-cascade-interface {enable | disable}
```

```
            set sla-fail-log-period <integer>
            set sla-pass-log-period <integer>
            set threshold-warning-packetloss <integer>
            set threshold-alert-packetloss <integer>
            set threshold-warning-latency <integer>
            set threshold-alert-latency <integer>
            set threshold-warning-jitter <integer>
            set threshold-alert-jitter <integer>
            set vrf <integer>
            set source <ip address>
            set members <member_number> ... <member_number>
            config sla
                edit 1
                    set link-cost-factor {latency jitter packet-loss}
                    set latency-threshold <integer>
                    set jitter-threshold <integer>
                    set packetloss-threshold <integer>
                next
            end
        next
    end
end
```

Additional settings are available for some of the protocols:

Protocol	Additional options

http, https	`port <port_number>`
	`http-get <url>`
	`http-agent <string>`
	`http-match <response_string>`

twamp	`port <port_number>`
	`security mode {none | authentication}`
	`password <password>`
	`packet-size <size>`

ftp	`ftp-mode {passive | port}` `ftp-file <path>`

For more examples see Protocol.

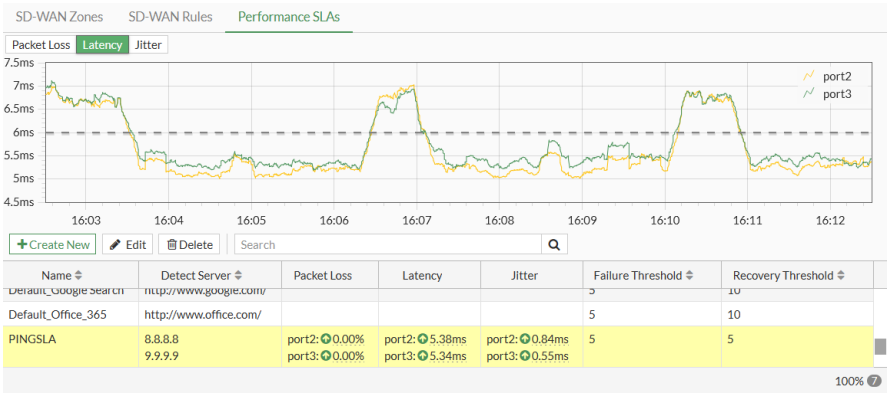
Monitoring performance SLA

SD-WAN diagnostics can be used to help maintain your SD-WAN solution.

Monitoring SD-WAN link quality status

Link quality plays a significant role in link selection for SD-WAN. Investigate any prolonged issues with packet loss, latency, or jitter to ensure that your network does not experience degraded performance or an outage.

You can monitor the link quality status of SD-WAN interface members by going to *Network > SD-WAN* and selecting the *Performance SLAs* tab.



The live charts show the packet loss, latency, or jitter for the selected health check. Hover the cursor over a line in the chart to see the specific value for that interface at that specific time.

The table shows information about each health check, including the configured servers, link quality data, and thresholds. The colored arrow indicates the status of the interface when the last status check was performed: green means that the interface was active, and red means that the interface was inactive. Hover the cursor over the arrow for additional information.

Monitoring system event logs

The features adds an SD-WAN daemon function to keep a short, 10 minute history of SLA that can be viewed in the CLI.

Performance SLA results related to interface selection, session failover, and other information, can be logged. These logs can then be used for long-term monitoring of traffic issues at remote sites, and for reports and views in FortiAnalyzer.

The time intervals that Performance SLA fail and pass logs are generated in can be configured.

To configure the fail and pass logs' generation time interval:

```
config system sdwan
    config health-check
        edit "PingSLA"
            set sla-fail-log-period 30
            set sla-pass-log-period 60
        next
    end
end
```

To view the 10 minute Performance SLA link status history:

```
FGDocs # diagnose sys sdwan sla-log PingSLA 1
Timestamp: Fri Sep  4 10:32:37 2020, vdom root, health-check PingSLA, interface: wan2, status: up,
latency: 4.455, jitter: 0.430, packet loss: 0.000%.
Timestamp: Fri Sep  4 10:32:37 2020, vdom root, health-check PingSLA, interface: wan2, status: up,
latency: 4.461, jitter: 0.436, packet loss: 0.000%.
Timestamp: Fri Sep  4 10:32:38 2020, vdom root, health-check PingSLA, interface: wan2, status: up,
latency: 4.488, jitter: 0.415, packet loss: 0.000%.
...
Timestamp: Fri Sep  4 10:42:36 2020, vdom root, health-check PingSLA, interface: wan2, status: up,
latency: 6.280, jitter: 0.302, packet loss: 0.000%.
Timestamp: Fri Sep  4 10:42:37 2020, vdom root, health-check PingSLA, interface: wan2, status: up,
latency: 6.261, jitter: 0.257, packet loss: 0.000%.
Timestamp: Fri Sep  4 10:42:37 2020, vdom root, health-check PingSLA, interface: wan2, status: up,
latency: 6.229, jitter: 0.245, packet loss: 0.000%.
```

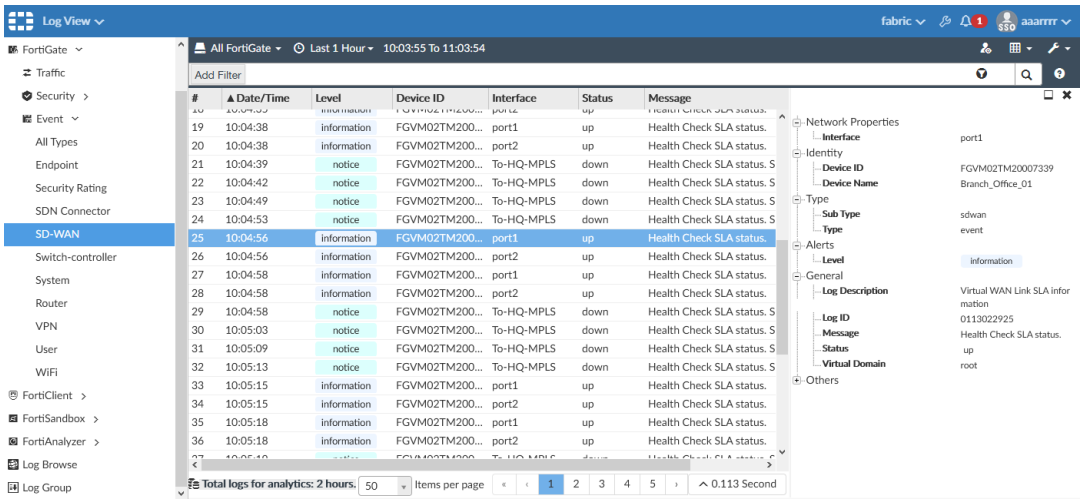
SLA pass logs

The FortiGate generates Performance SLA logs at the specified pass log interval (`sla-pass-log-period`) when SLA passes.

```
date="2021-04-15" time="10:04:56" id=6951431609690095758 bid=52507 dvid=1047 itime=1618506296
euid=3 epid=3 dsteuid=3 dstepid=3 logver=700000066 logid="0113022925" type="event" subtype="sdwan"
level="information" msg="Health Check SLA status." logdesc="SDWAN SLA information" status="up"
interface="port1" eventtime=1618506296222639301 tz="-0700" eventtype="SLA" jitter="0.277"
inbandwidthavailable="10.00Gbps" outbandwidthavailable="10.00Gbps"
bibandwidthavailable="20.00Gbps" packetloss="1.000%" latency="186.071" slamap="0x1"
healthcheck="BusinessCritical_CloudApps" slatargetid=1 outbandwidthused="40kbps"
inbandwidthused="24kbps" bibandwidthused="64kbps" devid="FGVM02TM20000000" vd="root"
devname="Branch_Office_01" csf="fabric"
```

```
date="2021-04-15" time="10:04:56" id=6951431609690095759 bid=52507 dvid=1047 itime=1618506296
euid=3 epid=3 dsteuid=3 dstepid=3 logver=700000066 logid="0113022925" type="event" subtype="sdwan"
level="information" msg="Health Check SLA status." logdesc="SDWAN SLA information" status="up"
interface="port2" eventtime=1618506296223163068 tz="-0700" eventtype="SLA" jitter="0.204"
inbandwidthavailable="10.00Gbps" outbandwidthavailable="10.00Gbps"
bibandwidthavailable="20.00Gbps" packetloss="0.000%" latency="185.939" slamap="0x1"
healthcheck="BusinessCritical_CloudApps" slatargetid=1 outbandwidthused="142kbps"
inbandwidthused="23kbps" bibandwidthused="165kbps" devid="FGVM02TM20000000" vd="root"
devname="Branch_Office_01" csf="fabric"
```

In the FortiAnalyzer GUI:



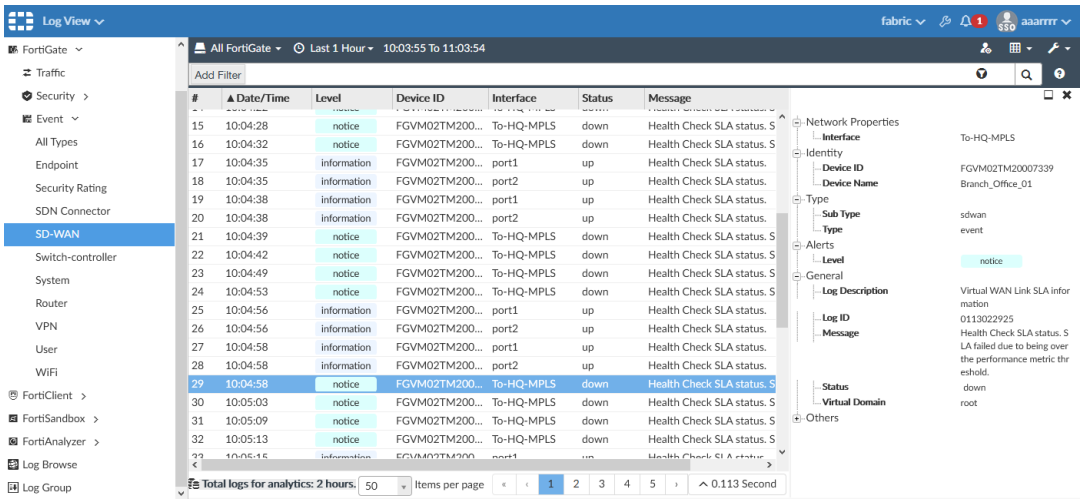
SLA fail logs

The FortiGate generates Performance SLA logs at the specified fail log interval (`sla-fail-log-period`) when SLA fails.

```
date="2021-04-15" time="10:04:59" id=6951431618280030243 bid=52507 dvid=1047 itime=1618506298
euid=3 epid=3 dsteuid=3 dstepid=3 logver=700000066 logid="0113022925" type="event" subtype="sdwan"
level="notice" msg="Health Check SLA status. SLA failed due to being over the performance metric
threshold." logdesc="SDWAN SLA information" status="down" interface="To-HQ-MPLS"
eventtime=1618506299718862835 tz="-0700" eventtype="SLA" jitter="0.000"
inbandwidthavailable="10.00Gbps" outbandwidthavailable="10.00Gbps"
bibandwidthavailable="20.00Gbps" packetloss="100.000%" latency="0.000" slamap="0x0"
healthcheck="BusinessCritical_CloudApps" slatargetid=1 metric="packetloss"
outbandwidthused="0kbps" inbandwidthused="0kbps" bibandwidthused="0kbps" devid="FGVM02TM20000000"
vd="root" devname="Branch_Office_01" csf="fabric"
```

```
date="2021-04-15" time="10:05:03" id=6951431639754866704 bid=52514 dvid=1046 itime=1618506303
euid=3 epid=3 dsteuid=3 dstepid=3 logver=700000066 logid="0113022925" type="event" subtype="sdwan"
level="notice" msg="Health Check SLA status. SLA failed due to being over the performance metric
threshold." logdesc="SDWAN SLA information" status="down" interface="To-HQ-MPLS"
eventtime=1618506304085863643 tz="-0700" eventtype="SLA" jitter="0.000"
inbandwidthavailable="10.00Gbps" outbandwidthavailable="10.00Gbps"
bibandwidthavailable="20.00Gbps" packetloss="100.000%" latency="0.000" slamap="0x0"
healthcheck="BusinessCritical_CloudApps" slatargetid=1 metric="packetloss"
outbandwidthused="6kbps" inbandwidthused="3kbps" bibandwidthused="9kbps" devid="FGVM02TM20000000"
vd="root" devname="Branch_Office_02" csf="fabric"
```

In the FortiAnalyzer GUI:



Monitoring using the REST API

SLA log and interface information can be monitored using the REST API. This feature is also used by FortiManager as part of its detailed SLA monitoring and drilldown features.

API call	URL
Interface log	`https://172.172.172.9/api/v2/monitor/virtual-wan/interface-log`
SLA log	`https://172.172.172.9/api/v2/monitor/virtual-wan/sla-log`
Health check log	`https://172.172.172.9/api/v2/monitor/virtual-wan/health-check`

A comprehensive list of API calls with sample output is available on the Fortinet Developer Network.

CLI diagnose commands:

```
# diagnose sys sdwan intf-sla-log port13
    Timestamp: Wed Jan 9 18:33:49 2019, used inbandwidth: 3208bps, used outbandwidth: 3453bps,
used bibandwidth: 6661bps, tx bytes: 947234bytes, rx bytes: 898622bytes.
    Timestamp: Wed Jan 9 18:33:59 2019, used inbandwidth: 3317bps, used outbandwidth: 3450bps,
used bibandwidth: 6767bps, tx bytes: 951284bytes, rx bytes: 902937bytes.
    Timestamp: Wed Jan 9 18:34:09 2019, used inbandwidth: 3302bps, used outbandwidth: 3389bps,
used bibandwidth: 6691bps, tx bytes: 956268bytes, rx bytes: 907114bytes.
    Timestamp: Wed Jan 9 18:34:19 2019, used inbandwidth: 3279bps, used outbandwidth: 3352bps,
used bibandwidth: 6631bps, tx bytes: 958920bytes, rx bytes: 910793bytes.
    Timestamp: Wed Jan 9 18:34:29 2019, used inbandwidth: 3233bps, used outbandwidth: 3371bps,
used bibandwidth: 6604bps, tx bytes: 964374bytes, rx bytes: 914854bytes.
    Timestamp: Wed Jan 9 18:34:39 2019, used inbandwidth: 3235bps, used outbandwidth: 3362bps,
used bibandwidth: 6597bps, tx bytes: 968250bytes, rx bytes: 918846bytes.
    Timestamp: Wed Jan 9 18:34:49 2019, used inbandwidth: 3165bps, used outbandwidth: 3362bps,
used bibandwidth: 6527bps, tx bytes: 972298bytes, rx bytes: 922724bytes.
    Timestamp: Wed Jan 9 18:34:59 2019, used inbandwidth: 3184bps, used outbandwidth: 3362bps,
used bibandwidth: 6546bps, tx bytes: 977282bytes, rx bytes: 927019bytes.
```

```
# diagnose sys sdwan sla-log ping 1 spoke11-p1_0
    Timestamp: Wed Mar  3 15:35:20 2021, vdom root, health-check ping, interface: spoke11-p1_0,
status: up, latency: 0.135, jitter: 0.029, packet loss: 0.000%.

# diagnose sys sdwan sla-log ping 2 spoke12-p1_0
    Timestamp: Wed Mar  3 15:36:08 2021, vdom root, health-check ping, interface: spoke12-p1_0,
status: up, latency: 0.095, jitter: 0.010, packet loss: 0.000%.
```

```
# diagnose sys sdwan health-check
    Health Check(ping):
    Seq(1 spoke11-p1): state(alive), packet-loss(0.000%) latency(0.156), jitter(0.043) sla_map=0x1
    Seq(1 spoke11-p1_0): state(alive), packet-loss(0.000%) latency(0.128), jitter(0.024) sla_
map=0x1
    Seq(2 spoke12-p1): state(alive), packet-loss(0.000%) latency(0.125), jitter(0.028) sla_map=0x1
    Seq(2 spoke12-p1_0): state(alive), packet-loss(0.000%) latency(0.093), jitter(0.008) sla_
map=0x1
```

Passive WAN health measurement

SD-WAN passive WAN health measurement determines the health check measurements (jitter, latency, and packet loss) using session information captured from the firewall policies that have *Passive Health Check* (`passive-wan-health-measurement`) enabled. Passive measurements analyze session information that is gathered from various TCP sessions can be viewed using the command `diagnose sys link-monitor-passive admin list by-interface`.

Using passive WAN health measurement reduces the amount of configuration required and decreases the traffic that is produced by health check monitor probes doing active measurements. Passive WAN health measurement analyzes real-life traffic; active WAN health measurement using a detection server might not reflect the real-life traffic.

By default, active WAN health measurement is enabled when a new health check is created. It can be changed to passive or prefer passive:

passive	Health is measured using live traffic passing through an SD-WAN link to determine link metrics (jitter, latency, and packet loss) of participating SD-WAN links. No link health monitor needs to be configured.
prefer-passive	Health is measured using live traffic when there is traffic passing though an SD-WAN link to determine link metrics (jitter, latency, and packet loss). If there is no live traffic flowing through an SD-WAN link for three continuous minutes, then the FortiGate sends out active probes to the configured health check server (`set server`) to calculate the link metrics. A link health monitor must be configured, see Link health monitor for details.

When `passive-wan-health-measurement` is enabled, `auto-asic-offload` will be disabled.

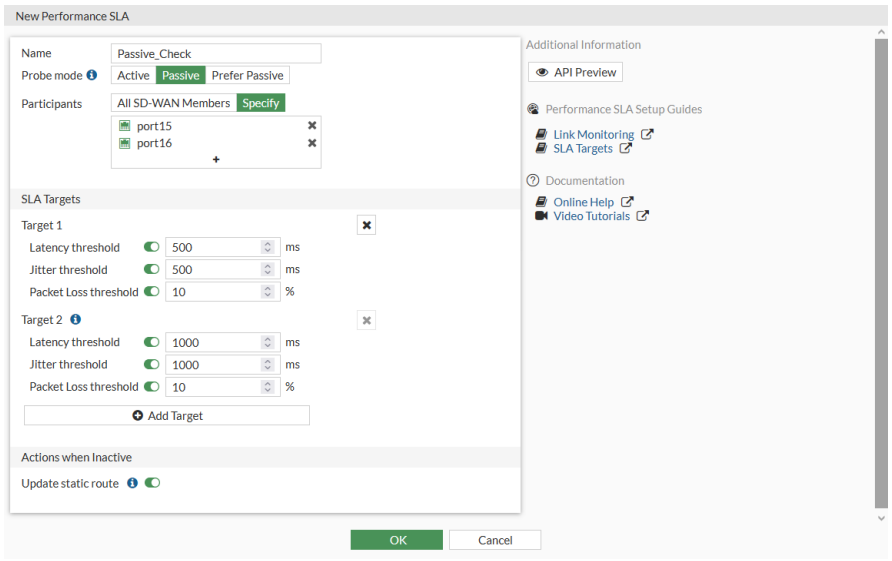
Example

In this example, the FortiGate is configured to load-balance between two WAN interfaces, port15 and port16. A health check is configured in passive mode, and SLA thresholds are set. Passive WAN health measurement is enabled on the SD-WAN policy.

Measurements are taken from YouTube traffic generated by the PC. When latency is introduced to the traffic on port15, the passive health check trigger threshold is exceeded and traffic is rerouted to port16.

To configure the SD-WAN in the GUI:

1. Create the SD-WAN zone:
   a. Go to *Network* > *SD-WAN* and select the *SD-WAN Zones* tab.
   b. Click *Create New* > *SD-WAN Zone*.
   c. Enter a name for the zone, such as *SD-WAN*.
   d. Click *OK*.
2. Create the SD-WAN members:
   a. Go to *Network* > *SD-WAN* and select the *SD-WAN Zones* tab.
   b. Click *Create New* > *SD-WAN Member*.
   c. Set *Interface* to *port15*, *SD-WAN Zone* to *SD-WAN*, and *Gateway* set to *172.16.209.2*.
   d. Click *OK*.
   e. Click *Create New* > *SD-WAN Member* again.
   f. Set *Interface* to *port16*, *SD-WAN Zone* to *SD-WAN*, and *Gateway* set to *172.16.210.2*.
   g. Click *OK*.
3. Create a performance SLA:
   a. Go to *Network* > *SD-WAN* and select the *Performance SLAs* tab.
   b. Edit an existing health check, or create a new one.
   c. Set *Probe mode* to *Passive*.
   d. Set *Participants* to *Specify* and add *port15* and *port16*.
   e. Configure two SLA targets. Note that the second SLA target must be configured in the CLI.



f. Configure the remaining settings as needed.

g. Click *OK*.

The SLA list shows the probe mode in the *Detect Server* column, if the probe mode is passive or prefer passive.

> Probe packets can only be disabled in the CLI and when the probe mode is not passive.

4. Create SD-WAN rules:

a. Go to *Network* > *SD-WAN*, select the *SD-WAN Rules* tab, and click *Create New*.

b. Configure the first rule:

Name	Background_Traffic
Source address	172.16.205.0
Application	Click in the field, and in the *Select Entries* pane search for *YouTube* and select all of the entries
Strategy	Maximize Bandwidth (SLA)
Interface preference	port15 and port16
Required SLA target	Passive_Check#2

c. Click *OK*.

d. Click *Create New* again and configure the second rule:

Name	Foreground_Traffic
Source address	172.16.205.0
Address	all
Protocol number	Specify - 1

Strategy	Lowest Cost (SLA)
Interface preference	port15 and port16
Required SLA target	Passive_Check#1

e. Click *OK*.

To configure the firewall policy in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
2. Configure the policy:

Name	SD-WAN-HC-policy
Incoming Interface	port5
Outgoing Interface	SD-WAN
Source	all
Destination	all
Schedule	always
Service	ALL
Action	ACCEPT
Passive Health Check	Enabled Passive health check can only be enabled in a policy when the outgoing interface is an SD-WAN zone.

3. Click *OK*.

To configure the SD-WAN in the CLI:

```
config system sdwan
    set status enable
    config zone
        edit "SD-WAN"
        next
    end
    config members
        edit 1
            set zone "SD-WAN"
            set interface "port15"
            set gateway 172.16.209.2
        next
        edit 2
            set zone "SD-WAN"
            set interface "port16"
            set gateway 172.16.210.2
        next
    end
```

```
    config health-check
        edit "Passive_Check"
            set detect-mode passive
            set members 1 2
            config sla
                edit 1
                    set latency-threshold 500
                    set jitter-threshold 500
                    set packetloss-threshold 10
                next
                edit 2
                    set latency-threshold 1000
                    set jitter-threshold 1000
                    set packetloss-threshold 10
                next
            end
        next
    end
    config service
        edit 1
            set name "Background_Traffic"
            set mode sla
            set load-balance enable
            set src "172.16.205.0"
            set internet-service enable
            set internet-service-app-ctrl 31077 33321 41598 31076 33104 23397 30201 16420 17396
38569 25564
            config sla
                edit "Passive_Check"
                    set id 2
                next
            end
            set priority-member 1 2
        next
        edit 2
            set name "Foreground_Traffic"
            set mode sla
            set src "172.16.205.0"
            set protocol 1
            set dst "all"
            config sla
                edit "Passive_Check"
                    set id 1
                next
            end
            set priority-member 1 2
        next
    end
end
```

To configure the firewall policy in the CLI:

```
config firewall policy
    edit 1
        set name "SD-WAN-HC-policy"
        set srcintf "port5"
        set dstintf "SD-WAN"
        set nat enable
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set passive-wan-health-measurement enable
        set auto-asic-offload disable
    next
end
```

Results

When both links pass the SLA:

```
# diagnose sys link-monitor-passive admin list by-interface
Interface port16 (28):
  Default(0x00000000): latency=10.0    15:46:36, jitter=5.0    15:46:37, pktloss=0.0  % 10:09:21

Interface port15 (27):
  Default(0x00000000): latency=60.0    15:46:36, jitter=0.0    15:46:37, pktloss=0.0  % 10:39:24
```

```
# diagnose sys sdwan health-check
Health Check(Passive_Check):
Seq(1 port15): state(alive), packet-loss(0.000%) latency(60.000), jitter(0.750) sla_map=0x3
Seq(2 port16): state(alive), packet-loss(0.000%) latency(10.000), jitter(5.000) sla_map=0x3
```

```
# diagnose sys sdwan service4 2

Service(2): Address Mode(IPV4) flags=0x200
  Gen(1), TOS(0x0/0x0), Protocol(1: 1->65535), Mode(sla), sla-compare-order
  Members(2):
    1: Seq_num(1 port15), alive, sla(0x1), gid(0), cfg_order(0), cost(0), selected
    2: Seq_num(2 port16), alive, sla(0x1), gid(0), cfg_order(1), cost(0), selected
  Src address(1):
        172.16.205.0-172.16.205.255

  Dst address(1):
        8.8.8.8-8.8.8.8
```

When the latency is increased to 610ms on port15, the SLA is broken and pings are sent on port16:

```
# diagnose sys sdwan health-check
Health Check(Passive_Check):
Seq(1 port15): state(alive), packet-loss(0.000%) latency(610.000), jitter(2.500) sla_map=0x3
Seq(2 port16): state(alive), packet-loss(0.000%) latency(50.000), jitter(21.000) sla_map=0x3
```

```
# diagnose sys sdwan service4 2

Service(2): Address Mode(IPV4) flags=0x200
  Gen(6), TOS(0x0/0x0), Protocol(1: 1->65535), Mode(sla), sla-compare-order
  Members(2):
    1: Seq_num(2 port16), alive, sla(0x1), gid(1), cfg_order(1), cost(0), selected
    2: Seq_num(1 port15), alive, sla(0x0), gid(2), cfg_order(0), cost(0), selected
  Src address(1):
        172.16.205.0-172.16.205.255

  Dst address(1):
        8.8.8.8-8.8.8.8
```

Passive measurement

Passive measurement allows SLA information per internet service/application to be differentiated and collected when internet services/applications are defined in an SD-WAN rule that uses passive or prefer passive SLA. The SLA metrics (jitter, latency, and packet loss) on each SD-WAN member in the rule are calculated based on the relevant internet services/applications SLA information. These metrics help analyze the performance of different applications using the same WAN link. See Passive health-check measurement by internet service and application on page 1087 for more information.

Passive health-check measurement by internet service and application

Active probing relies on checking the performance metrics of underlying infrastructure using layer 3 probes (ping) and layer 4 probes (tcp-echo, http, dns, and others) to provide limited information about an application's true performance.

Passive WAN health measurement uses passive probing to provide more realistic application performance information by collecting the performance metrics (jitter, latency, and packet loss) of live traffic that is passing through the firewall policies. See Passive WAN health measurement on page 1081.

Different applications can have different performance on the same WAN link, depending on the application's implementation. Passive measurement can be used to measure the performance of different internet services/applications that use the same WAN link.

The following is required:

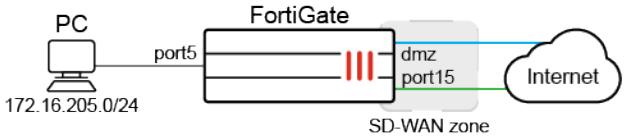
1. Firewall policy configuration:
   - Enable passive WAN health measurement (`set passive-wan-health-measurement enable`).
   - Disable hardware offloading (`set auto-asic-offload disable`).
   - Use an application control security profile to identify applications.
2. SD-WAN rule configuration:
   - Use passive or prefer passive performance SLA.
   - Use ISDB/application signatures or ISDB/application signature groups to identify applications.
   - Enable passive measurement (`set passive-measurement enable`).

If internet services or applications are defined in an SD-WAN rule with passive or prefer passive performance SLA , SLA information for each service or application will be differentiated and collected. SLA metrics (jitter, latency, and packet loss) on each SD-WAN member in the rule are then calculated based on the relevant internet service's or application's SLA information.

In this example, three SD-WAN rules are created:

- Rule 1: Best quality (latency) using passive SLA for the internet services Alibaba and Amazon.
- Rule 2: Best quality (latency) using passive SLA for the applications Netflix and YouTube.
- Rule 3: Best quality (latency) using passive SLA for all other traffic.

After passive application measurement is enabled for rules one and two, the SLA metric of rule one is the average latency of the internet services Alibaba and Amazon, and the SLA metric of rule two is the average latency of the applications Netflix and YouTube.



To configure the SD-WAN:

1. Configure the SD-WAN members:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "dmz"
            set gateway 172.16.208.2
        next
        edit 2
            set interface "port15"
            set gateway 172.16.209.2
        next
    end
end
```

2. Configure the passive mode health check:

```
config health-check
    edit "Passive_HC"
        set detect-mode passive
        set members 1 2
    next
end
```

3. Configure SD-WAN service rules:

```
config service
    edit 1
        set name "1"
        set mode priority
        set src "172.16.205.0"
        set internet-service enable
        set internet-service-name "Alibaba-Web" "Amazon-Web"
        set health-check "Passive_HC"
        set priority-members 1 2
        set passive-measurement enable    //Enable "passive application measurement", it is a
new command which is introduced in this project.
    next
    edit 2
        set name "2"
        set mode priority
        set src "172.16.205.0"
        set internet-service enable
        set internet-service-app-ctrl 18155 31077
        set health-check "Passive_HC"
        set priority-members 1 2
        set passive-measurement enable    ////Enable "passive application measurement"
    next
    edit 3
        set name "3"
        set mode priority
        set dst "all"
        set src "172.16.205.0"
        set health-check "Passive_HC"
        set priority-members 1 2
    next
end
```

4. Configure SD-WAN routes:

```
config router static
    edit 1
        set distance 1
        set sdwan-zone "virtual-wan-link"
    next
end
```

5. Configure the firewall policy with passive WAN health measurement enabled:

```
config firewall policy
    edit 1
        set uuid 972345c6-1595-51ec-66c5-d705d266f712
        set srcintf "port5"
        set dstintf "virtual-wan-link"
        set action accept
        set srcaddr "172.16.205.0"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set passive-wan-health-measurement enable
        set utm-status enable
        set ssl-ssh-profile "certificate-inspection"
        set application-list "g-default"
        set auto-asic-offload disable
    next
end
```

To verify the results:

1. On the PC, open the browser and visit the internet services and applications.
2. On the FortiGate, check the collected SLA information to confirm that each server or application on the SD-WAN members was measured individually:

```
# diagnose sys link-monitor-passive admin list by-interface

Interface dmz (5):
       Default(0x00000000): latency=3080.0  11:57:54, jitter=5.0     11:58:08, pktloss=0.0  %
NA
   Alibaba-Web(0x00690001): latency=30.0    11:30:06, jitter=25.0    11:29:13, pktloss=0.0  %
NA
       YouTube(0x00007965): latency=100.0   12:00:35, jitter=2.5     12:00:30, pktloss=0.0  %
NA
       Netflix(0x000046eb): latency=10.0    11:31:24, jitter=10.0    11:30:30, pktloss=0.0  %
NA
    Amazon-Web(0x00060001): latency=80.0    11:31:52, jitter=35.0    11:32:07, pktloss=0.0  %
NA

Interface port15 (27):
       Default(0x00000000): latency=100.0   12:00:42, jitter=0.0     12:00:42, pktloss=0.0  %
NA
    Amazon-Web(0x00060001): latency=30.0    11:56:05, jitter=0.0     11:55:21, pktloss=0.0  %
NA
   Alibaba-Web(0x00690001): latency=0.0     11:26:08, jitter=35.0    11:27:08, pktloss=0.0  %
NA
       YouTube(0x00007965): latency=100.0   11:33:34, jitter=0.0     11:33:50, pktloss=0.0  %
NA
       Netflix(0x000046eb): latency=0.0     11:26:29, jitter=0.0     11:29:03, pktloss=0.0  %
NA
```

The `Default(0x00000000)` applications are other, unidentified applications that do not have ISDB or application signatures configured in SD-WAN rules. The latency of default/application is taken into account in per SD-WAN rule calculations only if `passive-measurement` is disabled in any one of the SD-WAN rules.

3. Verify that the SLA metrics on the members are calculated as expected:

```
# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x600 use-shortcut-sla
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(priority), link-cost-factor(latency),
link-cost-threshold(10), health-check(Passive_HC)
  Members(2):
    1: Seq_num(2 port15), alive, latency: 15.000, selected           // Average latency of
"Alibaba-Web" and "Amazon-Web" on port15:     15.000 = (0.0+30.0)/2
    2: Seq_num(1 dmz), alive, latency: 55.000, selected              // Average latency of
"Alibaba-Web" and "Amazon-Web" on dmz:         55.000 = (30.0+80.0)/2
  Internet Service(2): Alibaba-Web(6881281,0,0,0) Amazon-Web(393217,0,0,0)
  Src address(1):
        172.16.205.0-172.16.205.255

Service(2): Address Mode(IPV4) flags=0x600 use-shortcut-sla
  Gen(2), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(priority), link-cost-factor(latency),
link-cost-threshold(10), heatlh-check(Passive_HC)
  Members(2):
    1: Seq_num(1 dmz), alive, latency: 55.000, selected               // Average latency of
"Netflix" and "YouTube" on dmz:        55.000 = (10.0+100.0)/2
    2: Seq_num(2 port15), alive, latency: 50.000, selected           // Average latency of
"Netflix" and "YouTube" on port15:     50.000 = (0.0+100.0)/2
  Internet Service(2): Netflix(4294837427,0,0,0 18155) YouTube(4294838283,0,0,0 31077)
  Src address(1):
        172.16.205.0-172.16.205.255

Service(3): Address Mode(IPV4) flags=0x200 use-shortcut-sla
  Gen(9), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(priority), link-cost-factor(latency),
link-cost-threshold(10), health-check(Passive_HC)
  Members(2):
    1: Seq_num(2 port15), alive, latency: 46.000, selected          // Average latency of all
TCP traffic on port15:       46   = (100.0+30.0+0.0+100.0+0.0)/5
    2: Seq_num(1 dmz), alive, latency: 660.000, selected           // Average latency of all
TCP traffic on dmz:           660 = (3080.0+30.0+100.0+10.0+80.0)/5
  Src address(1):
        172.16.205.0-172.16.205.255

  Dst address(1):
        0.0.0.0-255.255.255.255
```

The latency on each member interface per SD-WAN rule is the average of the latency of the application identified by respective SD-WAN rules.

The SLA metrics listed for each member interface per SD-WAN rule shown by the `diagnose sys sdwan service4` and `diagnose sys sdwan service6` commands are derived from the output of the SLA information for the applications shown in the output of the `diagnose sys link-monitor-passive admin list by-interface` command.

Until the applications are identified, their SLA metrics are not used to calculate SLA metrics for each member per SD-WAN rule. Applications are identified only when there is (or was) any application traffic passing through a member interface.

Mean opinion score calculation and logging in performance SLA health checks

The mean opinion score (MOS) is a method of measuring voice quality using a formula that takes latency, jitter, packet loss, and the codec into account to produce a score from one to five (1 - 5). The G.711, G.729, and G.722 codecs can be selected in the health check configurations, and an MOS threshold can be entered to indicate the minimum MOS score for the SLA to pass. The maximum MOS score will depend on which codec is used, since each codec has a theoretical maximum limit.

```
config system sdwan
    config health-check
          edit <name>
            set mos-codec {g711 | g729 | g722}
            config sla
                edit <id>
                    set link-cost-factor {latency jitter packet-loss mos}
                    set mos-threshold <value>
                next
            end
        next
    end
end
```

`mos-codec {g711 | g729 | g722}`	Set the VoIP codec to use for the MOS calculation (default = g711).
`link-cost-factor {latency jitter packet-loss mos}`	Set the criteria to base the link selection on.
`mos-threshold <value>`	Set the minimum MOS for the SLA to be marked as pass (1.0 - 5.0, default = 3.6).

To configure a health check to calculate the MOS:

```
config system sdwan
    set status enable
```

```
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "dmz"
            set gateway 172.16.208.2
        next
        edit 2
            set interface "port15"
            set gateway 172.16.209.2
        next
    end
    config health-check
        edit "Test_MOS"
            set server "2.2.2.2"
            set sla-fail-log-period 30
            set sla-pass-log-period 30
            set members 0
            set mos-codec g729
            config sla
                edit 1
                    set link-cost-factor mos
                    set mos-threshold "4.0"
                next
            end
        next
    end
end
```

To use an MOS SLA to steer traffic in an SD-WAN rule:

```
config system sdwan
    config service
        edit 1
            set name "MOS_traffic_steering"
            set mode sla
            set dst "HQ_LAN"
            set src "Branch_LAN"
            config sla
                edit "Test_MOS"
                    set id 1
                next
            end
            set priority-members 0
        next
    end
end
```

The MOS currently cannot be used to steer traffic when the `mode` is set to `priority`.

To verify the MOS calculation results:

1. Verify the health check diagnostics:

```
# diagnose sys sdwan health-check
Health Check(Test_MOS):
Seq(1 dmz): state(alive), packet-loss(0.000%) latency(0.114), jitter(0.026), mos(4.123),
bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x1
Seq(2 port15): state(alive), packet-loss(0.000%) latency(0.100), jitter(0.008), mos(4.123),
bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
```

```
# diagnose sys sdwan sla-log Test_MOS 1
Timestamp: Tue Jan  4 11:23:06 2022, vdom root, health-check Test_MOS, interface: dmz, status:
up, latency: 0.151, jitter: 0.040, packet loss: 0.000%, mos: 4.123.
Timestamp: Tue Jan  4 11:23:07 2022, vdom root, health-check Test_MOS, interface: dmz, status:
up, latency: 0.149, jitter: 0.041, packet loss: 0.000%, mos: 4.123.
```

```
# diagnose sys sdwan sla-log Test_MOS 2
Timestamp: Tue Jan  4 11:25:09 2022, vdom root, health-check Test_MOS, interface: port15,
status: up, latency: 0.097, jitter: 0.009, packet loss: 0.000%, mos: 4.123.
Timestamp: Tue Jan  4 11:25:10 2022, vdom root, health-check Test_MOS, interface: port15,
status: up, latency: 0.097, jitter: 0.008, packet loss: 0.000%, mos: 4.123.
```

2. Change the `mos-codec` to g722. The diagnostics will now display different MOS values:

```
# diagnose sys sdwan health-check
Health Check(Test_MOS):
Seq(1 dmz): state(alive), packet-loss(0.000%) latency(0.150), jitter(0.031), mos(4.453),
bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x1
Seq(2 port15): state(alive), packet-loss(0.000%) latency(0.104), jitter(0.008), mos(4.453),
bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
```

3. Increase the latency on the link in port15. The calculated MOS value will decrease accordingly. In this example, port15 is out of SLA since its MOS value is now less than the 4.0 minimum:

```
# diagnose sys sdwan  health-check
Health Check(Test_MOS):
Seq(1 dmz): state(alive), packet-loss(0.000%) latency(0.106), jitter(0.022), mos(4.453),
bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x1
Seq(2 port15): state(alive), packet-loss(0.000%) latency(300.119), jitter(0.012), mos(3.905),
bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x0
```

Sample logs

```
date=2022-01-04 time=11:57:54 eventtime=1641326274876828300 tz="-0800" logid="0113022933"
type="event" subtype="sdwan" level="notice" vd="root" logdesc="SDWAN SLA notification"
eventtype="SLA" healthcheck="Test_MOS" slatargetid=1 interface="port15" status="up"
latency="300.118" jitter="0.013" packetloss="0.000" mos="3.905" inbandwidthavailable="1000.00Mbps"
```

```
outbandwidthavailable="1000.00Mbps" bibandwidthavailable="2.00Gbps" inbandwidthused="0kbps"
outbandwidthused="0kbps" bibandwidthused="0kbps" slamap="0x0" metric="mos" msg="Health Check SLA
status. SLA failed due to being over the performance metric threshold."
```

```
date=2022-01-04 time=11:57:24 eventtime=1641326244286635920 tz="-0800" logid="0113022923"
type="event" subtype="sdwan" level="notice" vd="root" logdesc="SDWAN status" eventtype="Health
Check" healthcheck="Test_MOS" slatargetid=1 oldvalue="2" newvalue="1" msg="Number of pass member
changed."
```

```
date=2022-01-04 time=11:57:24 eventtime=1641326244286627260 tz="-0800" logid="0113022923"
type="event" subtype="sdwan" level="notice" vd="root" logdesc="SDWAN status" eventtype="Health
Check" healthcheck="Test_MOS" slatargetid=1 member="2" msg="Member status changed. Member out-of-
sla."
```

```
date=2022-01-04 time=11:57:02 eventtime=1641326222516756500 tz="-0800" logid="0113022925"
type="event" subtype="sdwan" level="information" vd="root" logdesc="SDWAN SLA information"
eventtype="SLA" healthcheck="Test_MOS" slatargetid=1 interface="port15" status="up"
latency="0.106" jitter="0.007" packetloss="0.000" mos="4.453" inbandwidthavailable="1000.00Mbps"
outbandwidthavailable="1000.00Mbps" bibandwidthavailable="2.00Gbps" inbandwidthused="0kbps"
outbandwidthused="0kbps" bibandwidthused="0kbps" slamap="0x1" msg="Health Check SLA status."
```

SD-WAN application monitor using FortiMonitor

The agent-based health check detection mode works with FortiMonitor to provide more accurate user level performance statistics. FortiMonitor acts as an agent and sends health check probes on behalf of the monitored FortiGate interface. FortiMonitor mimics a real user, and the probes return a more accurate application level performance. The SLA information collected from FortiMonitor is sent back to the FortiGate as the monitored interface's SLA information. These statistics can be used to gain a deeper insight into the SD-WAN traffic performance.

FortiGate can log statistics when using FortiMonitor to detect advanced SD-WAN application performance metrics. These logs may also be sent to FortiAnalyzer and FortiManager for review and reporting.

```
config system sdwan
    config health-check
        edit <name>
            set detect-mode agent-based
        next
    end
    config service
        edit <id>
            set agent-exclusive {enable | disable}
        next
    end
    set app-perf-log-period <time in seconds>
end
```

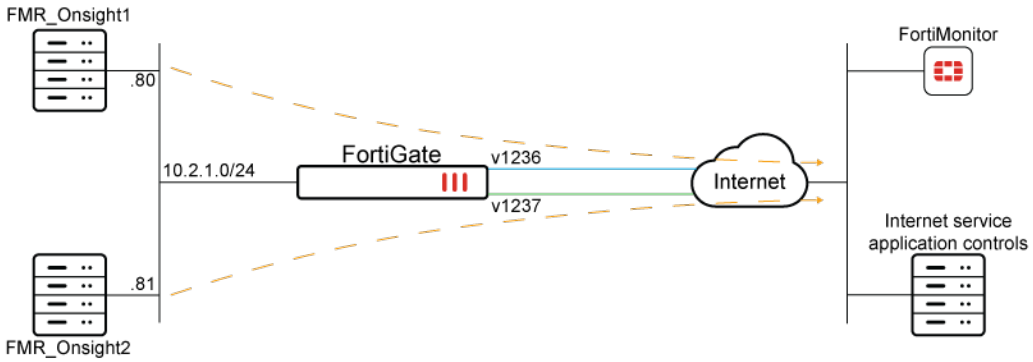
The following diagnostic commands can be used to view agent related metrics:

```
# diagnose sys link-monitor-passive agent <option>
```

`list`	List all the collected reports.
`list-app`	List the details of each application.
`flush`	Flush all the collected reports.
`flush-app`	Flush the details of all the applications.
`agent-oif-map`	List the agent and interface maps.

Example

In this example, routing is achieved through SD-WAN rules. The agent-based health check detection mode creates the FortiMonitor IP address and FortiGate SD-WAN interface map.



This example assumes that the FortiMonitor has already been added to the Security Fabric (see Configuring FortiMonitor on page 4384 for detailed instructions). The FortiMonitor OnSight (client) can be configured for two or more IP addresses, and each IP address is capable of sending application probes to user-specified applications.

Specific routing is implemented on the FortiGate to ensure each FortiMonitor client collects performance statistics for only one SD-WAN member interface. The FortiMonitor is configured to send application-specific probes to measure that application's performance on a given SD-WAN member. The FortiGate uses the FortiMonitor performance statistics to determine link quality based on application performance by mapping the health check. The link quality for a given application can then be used to steer the matching application traffic with greater accuracy.

To configure the FortiGate:

1. Configure the address objects for each FortiMonitor client:

```
config firewall address
    edit "FMR_OnSight1"
        set subnet 10.2.1.80 255.255.255.255
    next
    edit "MR_OnSight2"
        set subnet 10.2.1.81 255.255.255.255
    next
end
```

2. Set the logging frequency:

```
config system sdwan
    set status enable
    set app-perf-log-period 60
end
```

3. Configure the SD-WAN zone and members:

```
config system sdwan
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "v1236"
            set gateway 10.12.36.2
        next
        edit 2
            set interface "v1237"
            set gateway 10.12.37.20
        next
    end
end
```

4. Configure the SD-WAN rules to ensure each OnSight client uses only one SD-WAN member, and map the FortiMonitor IP to an SD-WAN member (interface):

```
config system sdwan
    config service
        edit 1
            set dst "all"
            set src "FMR_OnSight1"
            set priority-members 1
            set agent-exclusive enable
        next
        edit 2
            set dst "all"
            set src "FMR_OnSight2"
            set priority-members 2
            set agent-exclusive enable
        next
    end
end
```

5. Configure the SD-WAN health check:

```
config health-check
    edit "FMR"
        set detect-mode agent-based
        set members 1 2
        config sla
            edit 1
```

```
            next
        end
    next
end
```

To verify the SD-WAN member performance:

1. Verify the health check diagnostics:

```
# diagnose sys sdwan health-check
Health Check(FMR):
Seq(1 v1236): state(alive), packet-loss(0.000%) latency(183.214), jitter(0.124), mos(4.225),
bandwidth-up(999992), bandwidth-dw(999976), bandwidth-bi(1999968) sla_map=0x0
Seq(2 v1237): state(alive), packet-loss(0.000%) latency(182.946), jitter(0.100), mos(4.226),
bandwidth-up(999998), bandwidth-dw(999993), bandwidth-bi(1999991) sla_map=0x0
```

2. Verify the collected reports:

```
# diagnose sys link-monitor-passive agent list
       v1236( 23) | src=10.2.1.80 | latency=183.2   20:27:24 | jitter=0.1     20:27:24 |
pktloss=0.0  % 20:27:24
       v1237( 24) | src=10.2.1.81 | latency=182.9   20:27:24 | jitter=0.1     20:27:24 |
pktloss=0.0  % 20:27:24
```

3. Verify the details of each application:

```
# diagnose sys link-monitor-passive agent list-app
app_id=0x00000000, app=fortinet.com, dev=v1236(23)
        latency=183.2, jitter=0.1, pktloss=0.0,ntt=99.2,srt=384.8,app_err=0.0, 20:28:25
app_id=0x00000000, app=fortinet.com, dev=v1237(24)
        latency=183.1, jitter=0.5, pktloss=0.0,ntt=104.4,srt=377.8,app_err=0.0, 20:28:25
```

4. Verify the agent and interface maps:

```
# diagnose sys link-monitor-passive agent agent-oif-map
oif=v1236(23), src=10.2.1.80
oif=v1237(24), src=10.2.1.81
```

5. Review the SD-WAN logs:

6.
```
# execute log filter category event
# execute log filter field subtype sdwan
# execute log display

1: date=2023-01-27 time=16:32:15 eventtime=1674865935918381398 tz="-0800" logid="0113022937"
type="event" subtype="sdwan" level="information" vd="root" logdesc="Virtuan WAN Link
application performance metrics via FortiMonitor" eventtype="Application Performance Metrics"
app="fortinet.com" appid=0 interface="v1237" latency="200.2" jitter="0.6" packetloss="0.0"
serverresponsetime="827.7" networktransfertime="107.7" apperror="0.0" timestamp="01-28
00:31:59" msg="Application Performance Metrics via FortiMonitor"

2: date=2023-01-27 time=16:32:15 eventtime=1674865935918367770 tz="-0800" logid="0113022937"
```

```
type="event" subtype="sdwan" level="information" vd="root" logdesc="Virtuan WAN Link
application performance metrics via FortiMonitor" eventtype="Application Performance Metrics"
app="fortinet.com" appid=0 interface="v1236" latency="200.0" jitter="0.3" packetloss="0.0"
serverresponsetime="870.6" networktransfertime="130.4" apperror="0.0" timestamp="01-28
00:31:59" msg="Application Performance Metrics via FortiMonitor"

3: date=2023-01-27 time=16:31:15 eventtime=1674865875917685437 tz="-0800" logid="0113022937"
type="event" subtype="sdwan" level="information" vd="root" logdesc="Virtuan WAN Link
application performance metrics via FortiMonitor" eventtype="Application Performance Metrics"
app="fortinet.com" appid=0 interface="v1237" latency="200.5" jitter="0.7" packetloss="0.0"
serverresponsetime="1008.9" networktransfertime="129.8" apperror="0.0" timestamp="01-28
00:31:02" msg="Application Performance Metrics via FortiMonitor"

4: date=2023-01-27 time=16:31:15 eventtime=1674865875917672824 tz="-0800" logid="0113022937"
type="event" subtype="sdwan" level="information" vd="root" logdesc="Virtuan WAN Link
application performance metrics via FortiMonitor" eventtype="Application Performance Metrics"
app="fortinet.com" appid=0 interface="v1236" latency="200.3" jitter="0.8" packetloss="0.0"
serverresponsetime="825.4" networktransfertime="106.4" apperror="0.0" timestamp="01-28
00:31:02" msg="Application Performance Metrics via FortiMonitor"

5: date=2023-01-27 time=16:30:15 eventtime=1674865815912801725 tz="-0800" logid="0113022937"
type="event" subtype="sdwan" level="information" vd="root" logdesc="Virtuan WAN Link
application performance metrics via FortiMonitor" eventtype="Application Performance Metrics"
app="fortinet.com" appid=0 interface="v1237" latency="200.1" jitter="0.4" packetloss="0.0"
serverresponsetime="845.4" networktransfertime="116.0" apperror="0.0" timestamp="01-28
00:30:01" msg="Application Performance Metrics via FortiMonitor"

6: date=2023-01-27 time=16:30:15 eventtime=1674865815912786458 tz="-0800" logid="0113022937"
type="event" subtype="sdwan" level="information" vd="root" logdesc="Virtuan WAN Link
application performance metrics via FortiMonitor" eventtype="Application Performance Metrics"
app="fortinet.com" appid=0 interface="v1236" latency="200.0" jitter="0.3" packetloss="0.0"
serverresponsetime="1032.0" networktransfertime="138.9" apperror="0.0" timestamp="01-28
00:30:01" msg="Application Performance Metrics via FortiMonitor"
```

Classifying SLA probes for traffic prioritization

Traffic classification on SLA probes helps to ensure that they are prioritized in times of congestion. This prevents SD-WAN link flapping and unexpected routing behaviors, and stabilizes SD-WAN from unnecessary failovers.

SLA probes can be classified into a specific class ID so that SLA probes assigned to a class ID with higher priority are prioritized over other traffic. SLA probes are assigned using the `class-id` command:

```
config system sdwan
    config health-check
        edit <health-check name>
            set class-id <class name>
        next
    end
end
```

For more information on traffic shaping, see Traffic shaping on page 2108.

Example

In this example, SLA probes are assigned into different class ID. The interfaces dmz and vd1-01 both have outbandwidth of 1000000 Kbps (1 Gbps) configured. Three traffic shaping classes are defined:

Class ID	Name	Definition
2	sla_probe	High priority with a guaranteed 10% of bandwidth (100 Mbps)
3	default	Low priority with a guaranteed 80% of bandwidth (800 Mbps)
4	sla_probe_2	Medium priority with a guaranteed 10% of bandwidth (100 Mbps)

Under this scheme, when congestion occurs, traffic in each class will have their guaranteed bandwidth honored. If there is remaining bandwidth, higher priority traffic will get the bandwidth. On the SD-WAN health check, probes to server 2.2.2.2 are assigned to class 2 (sla_probe). This means it has a guaranteed bandwidth and has the highest priority to use unused bandwidth. This allows SD-WAN health check to function properly even during times of congestion.

To classify SLA probes for traffic prioritization:

1. Configure the firewall traffic class:

```
config firewall traffic-class
    edit 2
        set class-name "sla_probe"
    next
    edit 3
        set class-name "default"
    next
    edit 4
        set class-name "sla_probe_2"
    next
end
```

2. Configure the class ID priority and guaranteed bandwidth:

```
config firewall shaping-profile
    edit "profile-1"
        set default-class-id 3
        config shaping-entries
            edit 2
                set class-id 2
                set priority high
                set guaranteed-bandwidth-percentage 10
```

```
                set maximum-bandwidth-percentage 100
            next
            edit 3
                set class-id 3
                set priority low
                set guaranteed-bandwidth-percentage 80
                set maximum-bandwidth-percentage 100
            next
            edit 4
                set class-id 4
                set priority medium
                set guaranteed-bandwidth-percentage 10
                set maximum-bandwidth-percentage 100
            next
        end
    next
end
```

3. Configure the interfaces:

```
config system interface
    edit "dmz"
        set outbandwidth 1000000
        set egress-shaping-profile "profile-1"
        ...
    next
    edit "vd1-p1"
        set outbandwidth 1000000
        set egress-shaping-profile "profile-1"
        ...
    next
end
```

4. Configure the SD-WAN health check and assign the SLA probes into class 2:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "dmz"
            set gateway 172.16.208.2
        next
        edit 2
            set interface "vd1-p1"
        next
    end
    config health-check
        edit "1"
```

```
            set server "2.2.2.2"
            set members 1 2
            set class-id 2
            config sla
                edit 1
                next
            end
        next
    end
end
```

To verify the SLA probe assignment:

1. Verify the health check diagnostics:

```
# diagnose sys sdwan health-check
Health Check(1):
Seq(1 dmz): state(alive), packet-loss(0.000%) latency(0.247), jitter(0.022), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x1
Seq(2 vd1-p1): state(alive), packet-loss(0.000%) latency(0.247), jitter(0.018), mos(4.404),
bandwidth-up(999999), bandwidth-dw(1000000), bandwidth-bi(1999999) sla_map=0x1
```

2. Verify the SLA probes are assigned into class 2:

```
# diagnose netlink interface list dmz
if=dmz family=00 type=1 index=5 mtu=1500 link=0 master=0
ref=36 state=start present fw_flags=10018000 flags=up broadcast run multicast
Qdisc=mq hw_addr=e0:23:ff:9d:f9:9e broadcast_addr=ff:ff:ff:ff:ff:ff
egress traffic control:
        bandwidth=1000000(kbps) lock_hit=0 default_class=3 n_active_class=3
        class-id=3      allocated-bandwidth=800000(kbps)        guaranteed-bandwidth=800000
(kbps)
                            max-bandwidth=1000000(kbps)     current-bandwidth=1(kbps)
                            priority=low    forwarded_bytes=1446
                            dropped_packets=0       dropped_bytes=0
        class-id=4      allocated-bandwidth=100000(kbps)        guaranteed-bandwidth=100000
(kbps)
                            max-bandwidth=1000000(kbps)     current-bandwidth=0(kbps)
                            priority=medium         forwarded_bytes=0
                            dropped_packets=0       dropped_bytes=0
        class-id=2      allocated-bandwidth=100000(kbps)        guaranteed-bandwidth=100000
(kbps)
                            max-bandwidth=1000000(kbps)     current-bandwidth=1(kbps)
                            priority=high   forwarded_bytes=1404
                            dropped_packets=0       dropped_bytes=0
stat: rxp=19502 txp=14844 rxb=2233923 txb=802522 rxe=0 txe=0 rxd=0 txd=0 mc=0     collision=0
@ time=1675121675
re: rxl=0 rxo=0 rxc=0 rxf=0 rxfi=0 rxm=0
te: txa=0 txc=0 txfi=0 txh=0 txw=0
misc rxc=0 txc=0
input_type=0 state=3 arp_entry=0 refcnt=36
```

```
# diagnose netlink interface list vd1-p1
if=vd1-p1 family=00 type=768 index=99 mtu=1420 link=0 master=0
ref=20 state=start present fw_flags=10010000 flags=up p2p run noarp multicast
Qdisc=noqueue
egress traffic control:
        bandwidth=1000000(kbps) lock_hit=0 default_class=3 n_active_class=3
        class-id=3      allocated-bandwidth=800000(kbps)        guaranteed-bandwidth=800000
(kbps)
                            max-bandwidth=1000000(kbps)     current-bandwidth=0(kbps)
                            priority=low    forwarded_bytes=0
                            dropped_packets=0       dropped_bytes=0
        class-id=4      allocated-bandwidth=100000(kbps)        guaranteed-bandwidth=100000
(kbps)
                            max-bandwidth=1000000(kbps)     current-bandwidth=0(kbps)
                            priority=medium         forwarded_bytes=0
                            dropped_packets=0       dropped_bytes=0
        class-id=2      allocated-bandwidth=100000(kbps)        guaranteed-bandwidth=100000
(kbps)
                            max-bandwidth=1000000(kbps)     current-bandwidth=1(kbps)
                            priority=high   forwarded_bytes=1120
                            dropped_packets=0       dropped_bytes=0
stat: rxp=4097 txp=4586 rxb=540622 txb=221500 rxe=0 txe=19 rxd=0 txd=0 mc=0     collision=0 @
time=1675121742
re: rxl=0 rxo=0 rxc=0 rxf=0 rxfi=0 rxm=0
te: txa=0 txc=0 txfi=0 txh=0 txw=0
misc rxc=0 txc=0
input_type=0 state=3 arp_entry=0 refcnt=20
```

When verifying the class assignment, the counter value should increase.

The example also demonstrates assigning SLA probes to class 4 (sla_probe_2), in which case the probes get medium priority.

To assign the SLA probe to medium priority:

1. Assign SLA probes into class 4:

```
config sys sdwan
    config health-check
        edit 1
            set class-id 4
        next
    end
    set status disable
end
config sys sdwan
    set status enable
end
```

2. Verify the SLA probes are assigned into class 4.

```
# diagnose netlink interface list dmz
if=dmz family=00 type=1 index=5 mtu=1500 link=0 master=0
ref=34 state=start present fw_flags=10018000 flags=up broadcast run multicast
Qdisc=mq hw_addr=e0:23:ff:9d:f9:9e broadcast_addr=ff:ff:ff:ff:ff:ff
egress traffic control:
        bandwidth=1000000(kbps) lock_hit=0 default_class=3 n_active_class=3
        class-id=3      allocated-bandwidth=800000(kbps)        guaranteed-bandwidth=800000
(kbps)
                            max-bandwidth=1000000(kbps)     current-bandwidth=1(kbps)
                            priority=low    forwarded_bytes=24K
                            dropped_packets=0       dropped_bytes=0
        class-id=4      allocated-bandwidth=100000(kbps)        guaranteed-bandwidth=100000
(kbps)
                            max-bandwidth=1000000(kbps)     current-bandwidth=1(kbps)
                            priority=medium         forwarded_bytes=1674
                            dropped_packets=0       dropped_bytes=0
        class-id=2      allocated-bandwidth=100000(kbps)        guaranteed-bandwidth=100000
(kbps)
                            max-bandwidth=1000000(kbps)     current-bandwidth=0(kbps)
                            priority=high   forwarded_bytes=0
                            dropped_packets=0       dropped_bytes=0
stat: rxp=20818 txp=15874 rxb=2382789 txb=857674 rxe=0 txe=0 rxd=0 txd=0 mc=0 collision=0 @
time=1675122057
re: rxl=0 rxo=0 rxc=0 rxf=0 rxfi=0 rxm=0
te: txa=0 txc=0 txfi=0 txh=0 txw=0
misc rxc=0 txc=0
input_type=0 state=3 arp_entry=0 refcnt=34
```

```
# diagnose netlink interface list vd1-p1
if=vd1-p1 family=00 type=768 index=99 mtu=1420 link=0 master=0
ref=20 state=start present fw_flags=10010000 flags=up p2p run noarp multicast
Qdisc=noqueue
egress traffic control:
        bandwidth=1000000(kbps) lock_hit=0 default_class=3 n_active_class=3
        class-id=3      allocated-bandwidth=800000(kbps)        guaranteed-bandwidth=800000
(kbps)
                            max-bandwidth=1000000(kbps)     current-bandwidth=0(kbps)
                            priority=low    forwarded_bytes=0
                            dropped_packets=0       dropped_bytes=0
        class-id=4      allocated-bandwidth=100000(kbps)        guaranteed-bandwidth=100000
(kbps)
                            max-bandwidth=1000000(kbps)     current-bandwidth=1(kbps)
                            priority=medium         forwarded_bytes=1280
                            dropped_packets=0       dropped_bytes=0
        class-id=2      allocated-bandwidth=100000(kbps)        guaranteed-bandwidth=100000
(kbps)
                            max-bandwidth=1000000(kbps)     current-bandwidth=0(kbps)
                            priority=high   forwarded_bytes=0
                            dropped_packets=0       dropped_bytes=0
stat: rxp=4097 txp=4703 rxb=540622 txb=226180 rxe=0 txe=19 rxd=0 txd=0 mc=0 collision=0 @
time=1675122058
re: rxl=0 rxo=0 rxc=0 rxf=0 rxfi=0 rxm=0
```

```
te: txa=0 txc=0 txfi=0 txh=0 txw=0
misc rxc=0 txc=0
input_type=0 state=3 arp_entry=0 refcnt=20
```

SD-WAN rules

SD-WAN rules, which are sometimes called *service rules*, identify traffic of interest, and then route the traffic based on a strategy and the condition of the route or *link* between two devices. You can use many strategies to select the outgoing interface and many performance service level agreements (SLAs) to evaluate the link conditions.

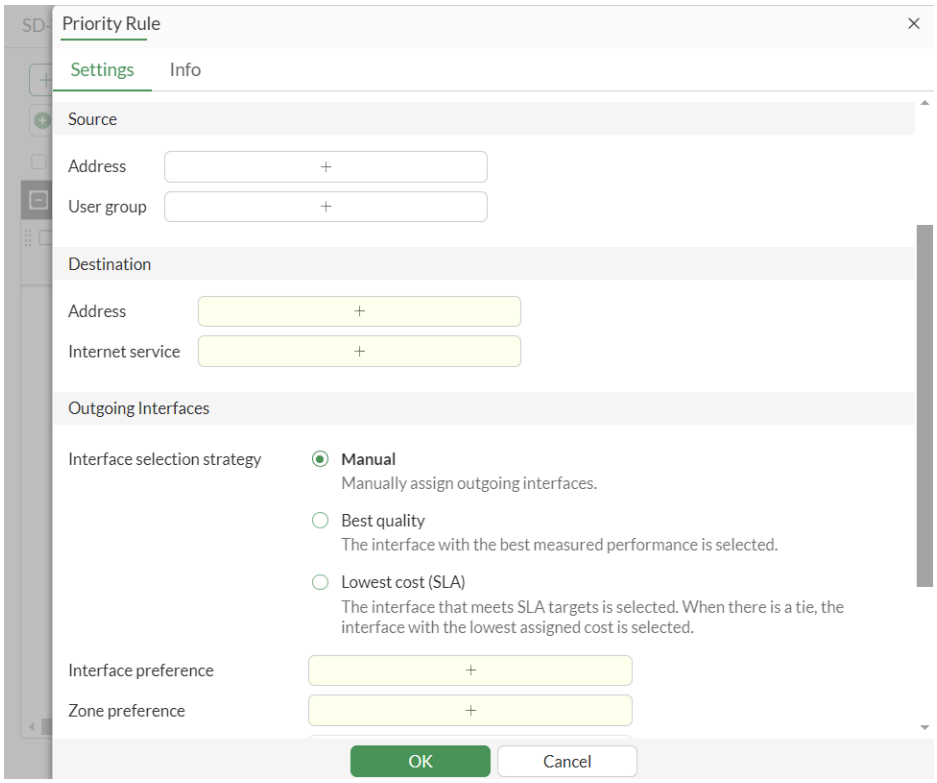
Use the following topics to learn about and create SD-WAN rules for your needs:

- SD-WAN rules overview on page 1105
- Implicit rule on page 1114
- Strategy on page 1118
- SDN dynamic connector addresses in SD-WAN rules on page 1143
- Application steering using SD-WAN rules on page 1145
- DSCP tag-based traffic steering in SD-WAN on page 1158
- ECMP support for the longest match in SD-WAN rule matching on page 1165
- Override quality comparisons in SD-WAN longest match rule matching on page 1168
- Internet service and application control steering on page 1171
- Use maximize bandwidth to load balance traffic between ADVPN shortcuts on page 1181
- Use SD-WAN rules to steer multicast traffic on page 1189
- Use SD-WAN rules for WAN link selection with load balancing on page 1209
- Ensuring SD-WAN prioritizes healthy SD-WAN members on page 1217

SD-WAN rules overview

SD-WAN rules control how sessions are distributed to SD-WAN members. You can configure SD-WAN rules from the GUI and CLI.

From the GUI, go to *Network* > *SD-WAN* > *SD-WAN Rules*. When creating a new SD-WAN rule, or editing an existing SD-WAN rule, use the *Source* and *Destination* sections to identify traffic, and use the *Outgoing interfaces* section to configure WAN intelligence for routing traffic.



From the CLI, use the following command to configure SD-WAN rules:

```
config system sdwan
    config service
        edit <ID>
        set comment <string>
    next
end
```

The following topics describe the fields used to configure SD-WAN rules:

- Fields for identifying traffic on page 1106
- Fields for configuring WAN intelligence on page 1110
- Additional fields for configuring WAN intelligence on page 1112

Fields for identifying traffic

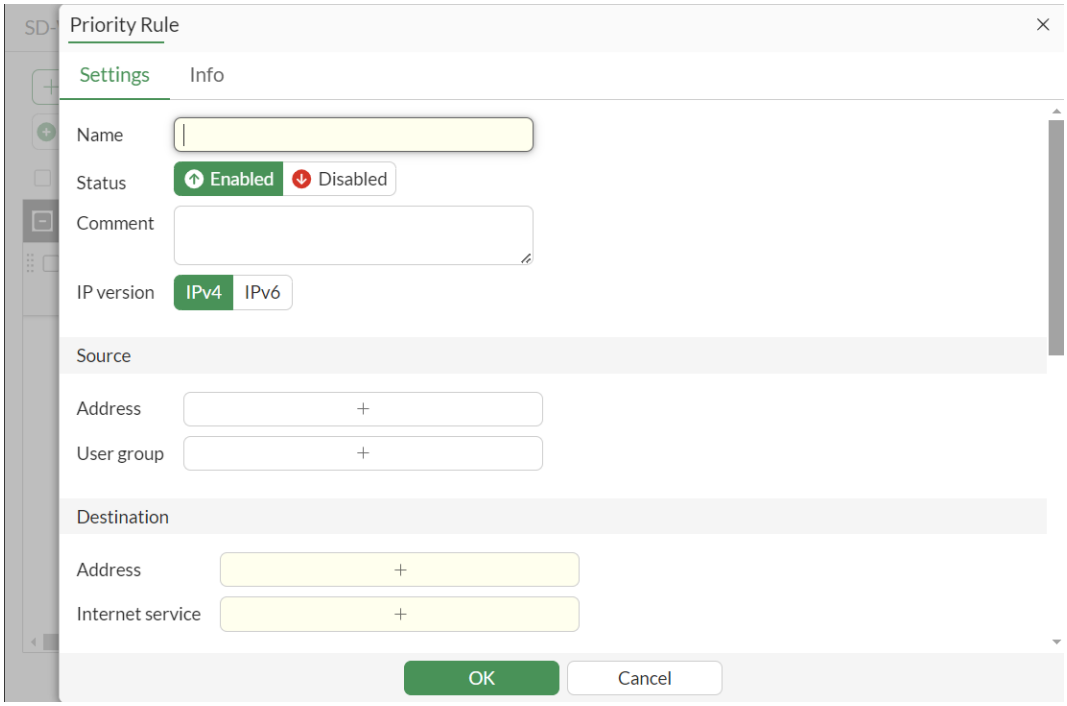
This topic describes the fields in an SD-WAN rule used for defining the traffic to which the rule applies. Some fields are available only in the CLI.

SD-WAN rules can identify traffic by a variety of means:

Address type	Source	Destination
IPv4/6	✓	✓
MAC	✓	✓

Address type	Source	Destination
Group	✓	✓
FABRIC_DEVICE dynamic address	✓	✓
Users	✓	✓
User groups	✓	✓
Application control (application aware routing)		✓
Internet service database (ISDB)		✓
BGP route tags		✓
Differentiated Services Code Point (DSCP) tags		✓

In the GUI, go to *Network* > *SD-WAN* > *SD-WAN Rules*. Click *Create New*, or double-click an existing rule to open it for editing. The *Source* and *Destination* sections are used to identify traffic for the rule:



In the CLI, edit the service definition ID number to identify traffic for the rule:

```
config system sdwan
    config service
        edit <ID>
        set comment <string>
        <CLI commands from the following tables>
        ...
    next
end
```

The following table describes the fields used for the name, ID, and IP version of the SD-WAN rule:

ID, Name, and IP version		
Field	**CLI**	**Description**

ID	`config system sdwan`
	`config service`
	`edit <ID>`
	`set comment <string>`
	`next`
	`end`
		ID is generated when the rule is created. You can only specify the ID from the CLI.

Name	`set name <string>`	The name does not need to relate to the traffic being matched, but it is good practice to have intuitive rule names.
IP version	`set addr-mode <ipv4 | ipv6>`	The addressing mode can be IPv4 or IPv6. To configure in the GUI, IPv6 must be enabled from *System* > *Feature Visibility* page.

The following table describes the fields used for source section of the SD-WAN rule:

Source		
Field	**CLI**	**Description**

Source address	`set src <object>`
	`set start-src-port <integer>`
	`set end-src-port <integer>`
	Use `set src-negate enable` to negate the address object.
		One or more address objects.
		Start source port number. CLI only.
		End source port number. CLI only.

User group	`set users <user object>` `set groups <group object>`	Individual users or user groups
Source interface	`set input-device <interface name>` Can be negated with `set input-device-negate enable`.	Select one or more source interfaces. CLI only.

The following table describes the fields used for the destination section of the SD-WAN rule:

Destination		
Field	**CLI**	**Description**

Address	`set dst <object>`
	`set protocol <integer>`
	`set start-port <integer>`
	`set end-port <integer>`
	Use `set dst-negate enable` to negate the address object.
		One or more address objects.
		One protocol and one port range can be combined with the address object.

Destination		
Field	**CLI**	**Description**
		If it is necessary for an SD-WAN rule to match multiple protocols or multiple port ranges, you can create a custom Internet Service.

Internet Service	`set internet-service enable`
	`set internet-service-custom <name_1> <name_2> ... <name_n>`
	`set internet-service-custom-group <name_1> <name_2> ... <name_n>`
	`set internet-service-name <name_1> <name_2> ... <name_n>`
	`set internet-service-group <name_1> <name_2> ... <name_n>`
		One or more internet services or service groups.

Application	`set internet-service-app-ctrl <id_1> <id_2> ... <id_n>`
	`set internet-service-app-ctrl-group <name_1> <name_2> ... <name_n>`
	`set internet-service-app-ctrl-category <id_1> <id_2> ... <id_n>`
		One or more applications or application groups.
		Can be used with internet services or service group.

Route tag (`route-tag`)	`set route-tag <integer>`	CLI only. This replaces the `dst` field (if previously configured) and matches a BGP route tag configured in a route map. See Using BGP tags with SD-WAN rules on page 1434.
TOS mask (`tos-mask`)	`set tos-mask <8-bit hex value>`	CLI only. In order to leverage type of service (TOS) matching or DSCP matching on the IP header, the SD-WAN rule must specify the bit mask of the byte holding the TOS value. For example, a TOS mask of 0xe0 (11100000) matches the upper 3 bits.
TOS (`tos`)	`set tos <8 bit hex value>`	CLI only. The value specified here is matched after the `tos-mask` is applied.

Destination		
Field	**CLI**	**Description**
		For example, the FortiGate receives DSCP values 110000 and 111011. (DSCP is the upper 6 bits of the TOS field – 11000000 and 11101100 respectively). Using the TOS value 0xe0 (11100000), only the second DSCP value is matched.

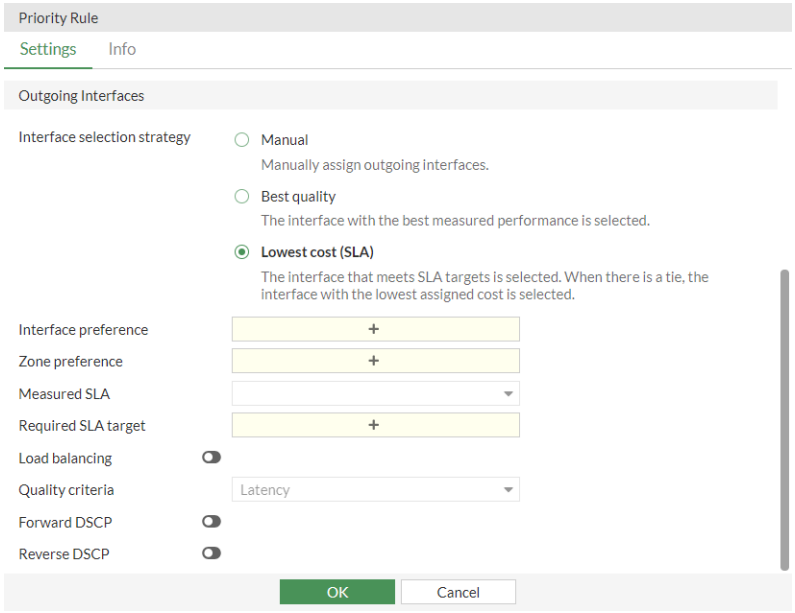
By default, individual applications and application groups cannot be selected in SD-WAN rules. To enable this functionality in the GUI, go to *System > Feature Visibility* and enable *Application Detection Based SD-WAN*. In the CLI, enter:

```
config system global
    set gui-app-detection-sdwan enable
end
```

Fields for configuring WAN intelligence

This topic describes the fields in an SD-WAN rule used for configuring WAN intelligence, which processes and routes traffic that matches the SD-WAN rule.

In the GUI, go to *Network > SD-WAN > SD-WAN Rules*. Click *Create New*, or double-click an existing rule to open it for editing. The *Outgoing Interfaces* section is used to configure WAN intelligence for the rule:



WAN intelligence is comprised of the following parts:

- Interface or zone preference on page 1111
- Strategy on page 1111

- Performance SLA on page 1112

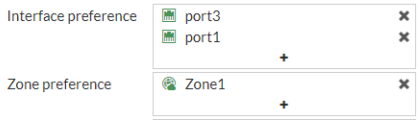
Interface or zone preference

By default, the configured order of interfaces and/or zones in a rule are used. Interfaces and zones that are selected first have precedence over interfaces selected second and so on.

You can specify both interfaces and zones. When a zone is specified in the *Zone preference* field, it is equivalent to selecting each of the contained interface members in the *Interface preference* section. Interface members in a zone have lower priority than interfaces configured in the *Interface preference* section.

For example:

- There are 3 interfaces: port1, port2 and port3.
  - Port2 is in Zone1
  - Port1 and port3 belong to the default *virtual-wan-link* zone.
- An SD-WAN rule is created with *Interface preference* set to *port3* and *port1*, and *Zone preference* set to *Zone1*.



The SD-WAN rule prefers the interfaces in the following order:

1. port3
2. port1
3. port2

You can configure the interface and zone preference in the CLI:

```
config system sdwan
    config service
        edit <ID>
            set priority-members <integer>
            set priority-zone <interface>
        next
    end
end
```

Strategy

Strategy dictates how the interface and/or zone order changes as link conditions change. You can use the following strategies:

- Automatic (`auto`): interfaces are assigned a priority based on quality. See Automatic strategy on page 1118.
- Manual (`manual`): interfaces are manually assigned a priority. See Manual strategy on page 1119.
- Best Quality (`priority`): interfaces are assigned a priority based on the `link-cost-factor` of the interface. See Best quality strategy on page 1123.
- Lowest cost (SLA) (`sla`): interfaces are assigned a priority based on selected SLA settings. See Lowest cost (SLA) strategy on page 1126.

Performance SLA

The best quality, lowest cost, and maximize bandwidth strategies are the most intelligent modes, and they leverage SLA health checks to provide meaningful metrics for a given link. FortiGate uses the metrics to make intelligent decisions to route traffic.

Automatic and manual strategies have pre-configured logic that do not leverage SLA health checks.

The goal of the performance SLA is to measure the quality of each SD-WAN member link. The following methods can be used to measure the quality of a link:

- Active measurement
  - Health-check traffic is sent to a server with a variety of protocols options.
  - The following SLA metrics are measured on this probe traffic:
    - Latency
    - Jitter
    - Packet loss
- Passive measurement
  - SLA metrics are measured on real or live traffic, reducing the amount of probe traffic that is sent and received.
  - There is the option (prefer passive) to initiate probe traffic when no live traffic is present.

Performance SLA is utilized by `auto`, *Lowest Cost (SLA)*, *Maximize Bandwidth (SLA)*, and *Best Quality* strategies. *Lowest Cost (SLA)* and *Maximize Bandwidth SLA* use SLA targets in a pass or fail style to evaluate whether a link is considered for traffic. *Best Quality* compares a specific metric of the SLA to pick the best result.

Therefore it is integral to select or create an SLA target(s) that relates to the traffic targeted by the rule. It does not make sense to evaluate a public resource, such as YouTube, when the rule matches Azure traffic.

See SD-WAN performance SLA on page 1068 for more details.

Additional fields for configuring WAN intelligence

This topic describes the fields in an SD-WAN rule used for configuring WAN intelligence for egress traffic:

- Forward and/or reverse differentiated services code point (DSCP) on page 1112
- Default and gateway options on page 1113

For information about accessing fields for configuring WAN intelligence, see Fields for configuring WAN intelligence on page 1110.

Forward and/or reverse differentiated services code point (DSCP)

The FortiGate differentiated services feature can be used to change the DSCP value for all packets accepted by a policy.

The packet's `DSCP` field for traffic initiating a session (forward) or for reply traffic (reverse) can be changed and enabled in each direction separately by configuring it in the firewall policy using the *Forward DSCP* and *Reverse DSCP* fields.

From the CLI:

```
config system sdwan
    config service
        edit <ID>
            ...
            set dscp-forward enable
            ...
        next
    end
end
```

Command	Description
`dscp-forward enable`	Enable use of forward DSCP tag.
`dscp-forward-tag 000000`	Forward traffic DSCP tag.
`dscp-reverse enable`	Enable use of reverse DSCP tag.
`dscp-reverse-tag 000000`	Reverse traffic DSCP tag.

Default and gateway options

Following are additional gateway options that can be set only in the CLI:

```
config system sdwan
    config service
        edit <ID>
            ...
            set default enable
            ...
        next
    end
end
```

Command	Description
`default [enable|disable]`	Enable or disable use of SD-WAN as default service.
`gateway [enable|disable]`	Enable or disable SD-WAN service gateway.

By default, these settings are set to `disable`.

These two commands help adjust FortiGate route selection by affecting how the FortiGate consults the Forward Information Base (FIB).

In order to decide whether an SD-WAN policy-route can be matched, FortiGate performs the following FIB lookups:

- FIB best match for the destination must return an SD-WAN member.
- FIB route to the destination must exist over the desired SD-WAN member.

When `set default enable` is used with `set gateway enable`, FortiGate bypasses the FIB checks, and instead routes any matching traffic of the SD-WAN rule to the chosen SD-WAN member using the member's configured gateway. SD-WAN members must have a gateway configured.

When `set default disable` is used with `set gateway enable`, FortiGate keeps the first rule in effect but causes the second rule to change to:

- FIB route to the gateway IP address must exist over any interface.

See also Fields for configuring WAN intelligence on page 1110.

Implicit rule

SD-WAN rules define specific policy routing options to route traffic to an SD-WAN member. When no explicit SD-WAN rules are defined, or if none of the rules are matched, then the default implicit rule is used.

In an SD-WAN configuration, the default route usually points to the SD-WAN interface, so each active member's gateway is added to the routing table's default route. FortiOS uses equal-cost multipath (ECMP) to balance traffic between the interfaces. One of five load balancing algorithms can be selected:

Source IP (`source-ip-based`)	Traffic is divided equally between the interfaces, including the SD-WAN interface. Sessions that start at the same source IP address use the same path. This is the default selection.
Sessions (`weight-based`)	The workload is distributing based on the number of sessions that are connected through the interface. The weight that you assign to each interface is used to calculate the percentage of the total sessions that are allowed to connect through an interface, and the sessions are distributed to the interfaces accordingly. The sessions with the same source and destination IP are forwarded to the same path if the device model and kernel version supports route cache. However, it is not guaranteed and the route cache could be refreshed in case network events take place. In most cases where route cache is not supported, the sessions with the same source and destination IP will be load balanced between SD-WAN member interfaces. An interface's weight value cannot be zero.
Spillover (`usage-based`)	The interface is used until the traffic bandwidth exceeds the ingress and egress thresholds that you set for that interface. Additional traffic is then sent through the next SD-WAN interface member.
Source-Destination IP (`source-dest-ip-based`)	Traffic is divided equally between the interfaces. Sessions that start at the same source IP address and go to the same destination IP address use the same path.
Volume (`measured-volume-based`)	The workload is distributing based on the number of packets that are going through the interface. The volume weight that you assign to each interface is used to calculate the percentage of the total bandwidth that is allowed to go through an interface, and the bandwidth is distributed to the interfaces accordingly. An interface's volume value cannot be zero.

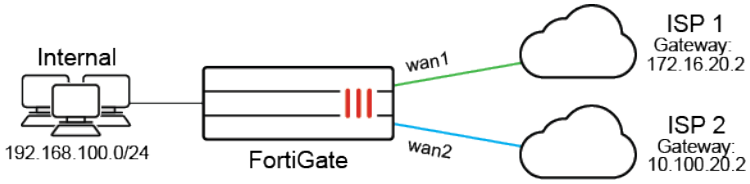
You cannot exclude an interface from participating in load balancing using the implicit rule. If the weight or volume were set to zero in a previous FortiOS version, the value is treated as a one.

When using dynamic routes for routing, sessions are distributed equally regardless of weight.

Interfaces with static routes can be excluded from ECMP if they are configured with a lower priority than other static routes.

Examples

The following four examples demonstrate how to use the implicit rules (load-balance mode).



If no SD-WAN zone is specified, members are added to the default *virtual-wan-link* zone.

Example 1

Outgoing traffic is equally balanced between wan1 and wan2, using *source-ip-based* or *source-dest-ip-based* mode.

Using the GUI:

1. On the FortiGate, enable SD-WAN and add wan1 and wan2 as SD-WAN members, then add a policy and static route. See SD-WAN quick start on page 1041 for details.
2. Go to *Network* > *SD-WAN* and select the *SD-WAN Rules* tab.
3. Edit the *sd-wan* rule (the last default rule).
4. For the *Load Balancing Algorithm*, select either *Source IP* or *Source-Destination IP*.
5. Click *OK*.

Using the CLI:

1. Enable SD-WAN and add wan1 and wan2 as SD-WAN members, then add a policy and static route. See SD-WAN quick start on page 1041 for details.
2. Set the load balancing algorithm:
   Source IP based:

```
config system sdwan
    set load-balance-mode source-ip-based
end
```

   Source-Destination IP based:

```
config system sdwan
    set load-balance-mode source-dest-ip-based
end
```

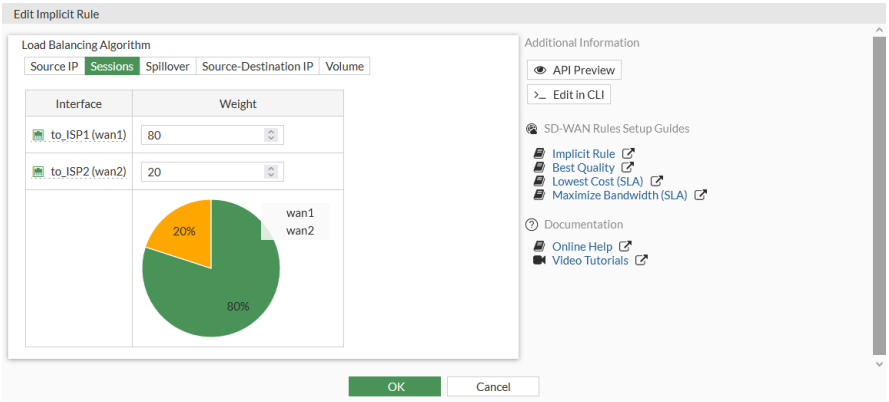
Example 2

Outgoing traffic is balanced between wan1 and wan2 with a customized ratio, using *weight-based* mode: wan1 runs 80% of the sessions, and wan2 runs 20% of the sessions.

Sessions with the same source and destination IP addresses (`src-ip` and `dst-ip`) will be forwarded to the same path, but will still be considered in later session ratio calculations.

Using the GUI:

1. Go to *Network* > *SD-WAN* and select the *SD-WAN Rules* tab.
2. Edit the *sd-wan* rule (the last default rule).
3. For the *Load Balancing Algorithm*, select *Sessions*.
4. Enter 80 in the *wan1* field, and 20 in the *wan2* field.



5. Click *OK*.

Using the CLI:

```
config system sdwan
    set load-balance-mode weight-based
    config members
        edit 1
            set interface "wan1"
            set weight 80
        next
        edit 2
            set interface "wan2"
            set weight 20
        next
    end
end
```

Example 3

Outgoing traffic is balanced between wan1 and wan2 with a customized ratio, using *measured-volume-based* mode: wan1 runs 80% of the volume, and wan2 runs 20% of the volume.

Using the GUI:

1. Go to *Network* > *SD-WAN* and select the *SD-WAN Rules* tab.
2. Edit the *sd-wan* rule (the last default rule).
3. For the *Load Balancing Algorithm*, select *Volume*.
4. Enter 80 in the *wan1* field, and 20 in the *wan2* field.
5. Click *OK*.

Using the CLI:

```
config system sdwan
    set load-balance-mode measured-volume-based
    config members
        edit 1
            set interface "wan1"
            set volume-ratio 80
        next
        edit 2
            set interface "wan2"
            set volume-ratio 20
        next
    end
end
```

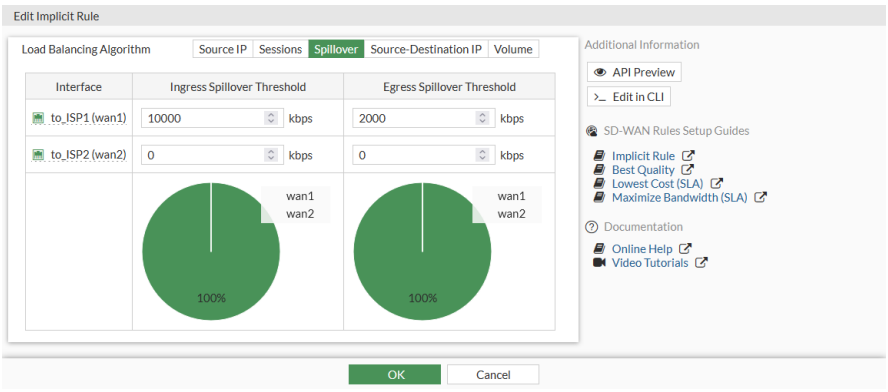
Example 4

Load balancing can be used to reduce costs when internet connections are charged at different rates. For example, if wan2 charges based on volume usage and wan1 charges a fixed monthly fee, we can use wan1 at its maximum bandwidth, and use wan2 for overflow.

In this example, wan1's bandwidth is 10Mbps down and 2Mbps up. Traffic will use wan1 until it reaches its spillover limit, then it will start to use wan2. Note that `auto-asic-offload` must be disabled in the firewall policy.

Using the GUI:

1. On the FortiGate, enable SD-WAN and add wan1 and wan2 as SD-WAN members, then add a policy and static route. See SD-WAN quick start on page 1041 for details.
2. Go to *Network* > *SD-WAN* and select the *SD-WAN Rules* tab.
3. Edit the *sd-wan* rule (the last default rule).
4. For the *Load Balancing Algorithm*, select *Spillover*.
5. Enter 10000 in the *wan1 Ingress Spillover Threshold* field, and 2000 in the *wan1 Egress Spillover Threshold* field.



6. Click *OK*.

Using the CLI:

```
config system sdwan
    set load-balance-mode usage-based
    config members
        edit 1
            set interface "wan1"
            set spillover-threshold 2000
            set ingress-spillover-threshold 10000
        next
    end
end
```

Strategy

The following strategies can be used:

- Automatic strategy on page 1118
- Manual strategy on page 1119
- Best quality strategy on page 1123
- Lowest cost (SLA) strategy on page 1126
- Hybrid strategy of priority and SLA modes on page 1133

Automatic strategy

The automatic strategy is a legacy rule that lets you select an outgoing interface based on its performance ranking compared to the other SD-WAN interfaces. This is achieved by applying a performance SLA to rank the interfaces, and then selecting the desired rank.

In this example, you have three SD-WAN interfaces to three different ISPs that all go to the public internet. WAN1 is your highest quality link and should be reserved for business critical traffic. WAN2 and WAN3 are redundant backup links. You noticed one non-critical application is taking up a lot of bandwidth and want to prioritize it to the lowest quality link at any given time.

To configure automatic SD-WAN rules from the CLI:

```
config system sdwan
    config members
        edit 1
            set interface "wan1"
        next
        edit 2
            set interface "wan2"
        next
        edit 3
            set interface "wan3"
        next
    end
    config health-check
        edit "non-critical application"
            set server "noncritical.application.com"
            set members 1 2 3
            config sla
                edit 1
                    set latency-threshold 250
                    set jitter-threshold 50
                    set packletloss-threshold 3
                next
            end
        next
    end
    config service
        edit 1
            set name "non-critical application"
            set mode auto
            set quality-link 3
            set dst "non-critical-app-address-object"
            set health-check "non-critical application"
        next
    end
end
```

⚠ The `auto` option is only available in the CLI. If you use the GUI to edit the rule, the `auto` option will be overwritten because you cannot select *auto* in the GUI.

Manual strategy

In manual mode, no health checks are used. As a result, the decision making closer resembles logic than intelligence. SD-WAN manual rules are similar to regular policy-based routes, but have the added features of application-aware routing and BGP-tag routing. A manual strategy rule is comprised of the following parts:

- Defining the interfaces to be used
- Ordering the interfaces based on preference, or load balancing traffic out of the specified interfaces using a load balancing algorithm

The maximize bandwidth (`load-balance`) strategy used prior to FortiOS 7.4.1 is now known as the load balancing strategy. This strategy can be configured under the manual mode and the lowest cost (SLA) strategies.

- When the load balancing strategy is configured under the manual mode strategy, SLA targets are not used.
- When the load balancing strategy is configured under the lowest cost (SLA) strategy, SLA targets are used.

To configure manual SD-WAN rules from the GUI:

1. Go to *Network* > *SD-WAN*.
2. Select the *SD-WAN Rules* tab, and click *Create New*.
3. Set the following options to create a manual rule:

Name	Type a name for the rule.
Source	(Optional) Specify a *Source address* and/or *User group*.
Destination	Specify the destination using an *Address* object or an *Internet Service* or an *Application*.
Zone preference	Specify one or more SD-WAN interfaces or zones. The order in which the interfaces or zones are specified determines their priority when the rule is matched.

4. Set the remaining options as desired, and click *OK* to create the rule.

To configure manual SD-WAN rules from the CLI:

```
config system sdwan
    config members
        edit 1
            set interface "wan1"
        next
        edit 2
            set interface "wan2"
        next
    end
    config service
        edit 1
            set name "manual"
            set mode manual
            set priority-members 2 1
            set dst "DC_net"
            set hold-down-time 60
        next
    end
end
```

- The command `set mode manual` will not appear in the configuration because it is the default mode.
- The command `set hold-down-time <integer>` is an optional command that controls how long to wait before switching back to the primary interface in the event of a failover.

Load balancing strategy without SLA targets

The load balancing strategy known as maximize bandwidth (`load-balance`) prior FortiOS 7.4.1 is now configured within manual mode SD-WAN rules to achieve load balancing but without the need to configure SLA targets.

By enabling load balancing mode (`set load-balance enable`) inside the manual SD-WAN rule, SD-WAN will start to load balance traffic out of all the specified interfaces based on the configured load balancing algorithm. There is no explicit need to configure SLA targets to achieve load balancing. The load balancing algorithm, or hash method, can be one of the following:

`round-robin`	All traffic is distributed to selected interfaces in equal portions and circular order. This is the default method, and the only option available when using the GUI.
`source-ip-based`	All traffic from a source IP is sent to the same interface.
`source-dest-ip-based`	All traffic from a source IP to a destination IP is sent to the same interface.
`inbandwidth`	All traffic is distributed to a selected interface with most available bandwidth for incoming traffic.
`outbandwidth`	All traffic is distributed to a selected interface with most available bandwidth for outgoing traffic.
`bibandwidth`	All traffic is distributed to a selected interface with most available bandwidth for both incoming and outgoing traffic.

When the `inbandwidth`, `outbandwidth`, or `bibandwidth` load balancing algorithm is used, the FortiGate will compare the bandwidth based on the configured upstream and downstream bandwidth values.

The interface speedtest can be used to populate the bandwidth values based on the speedtest results. See GUI speed test on page 1239 for details.

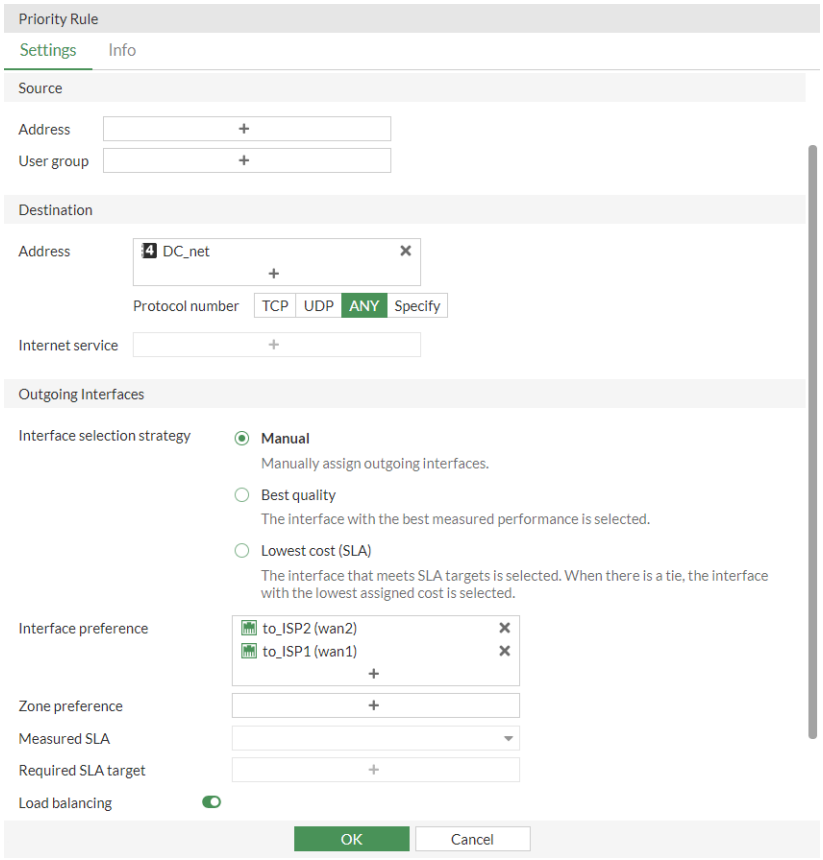
To manually configure the upstream and downstream bandwidth values:

```
config system interface
    edit <interface>
        set estimated-upstream-bandwidth <speed in kbps>
        set estimated-downstream-bandwidth <speed in kbps>
    next
end
```

To enable the load balancing strategy for manual mode in the GUI:

1. Go to *Network* > *SD-WAN*.
2. Select the *SD-WAN Rules* tab, and click *Create New*.

3. Set the *Interface selection strategy* to *Manual*.
4. Enable *Load balancing*.



5. Set the remaining options as desired, and click *OK* to create the rule.

To enable the load balancing strategy for manual mode in the CLI:

```
config system sdwan
    ...
    config service
        edit 1
            set name "manual"
            set mode manual
            set load-balance enable
            set hash-mode round-robin
            set priority-members 2 1
            set dst "DC_net"
            set hold-down-time 60
        next
    end
end
```

Best quality strategy

When using *Best Quality* mode, SD-WAN will choose the best link to forward traffic by comparing the *link-cost-factor*. A link-cost factor is a specific metric of participating link(s) (such as, latency, packet loss, and so on) evaluated against a target that you define (such as a health-check server), for example, the latency of WAN1 and WAN2 to your datacenter. Below is a list of link-cost factors available to you:

GUI	CLI	Description
Latency	`latency`	Select a link based on latency.
Jitter	`jitter`	Select a link based on jitter.
Packet Loss	`packet-loss`	Select a link based on packet loss.
Downstream	`inbandwidth`	Select a link based on available bandwidth of incoming traffic.
Upstream	`outbandwidth`	Select a link based on available bandwidth of outgoing traffic.
Bandwidth	`bibandwidth`	Select a link based on available bandwidth of bidirectional traffic.
Customized profile	`custom-profile-1`	Select link based on customized profile. If selected, set the following weights: • packet-loss-weight: Coefficient of packet-loss. • latency-weight: Coefficient of latency. • jitter-weight: Coefficient of jitter. • bandwidth-weight: Coefficient of reciprocal of available bidirectional bandwidth.

Although SD-WAN intelligence selects the best quality link according to the selected metric, by default a preference or advantage is given to the first configured SD-WAN member. This default is 10% and may be configured with the CLI command `set link-cost-threshold 10`.

Example of how `link-cost-threshold` works:

```
config system sdwan
    config members
        edit 1
            set interface "wan1"
        next
        edit 2
            set interface "wan2"
        next
    end
    config service
        edit 1
            set name "Best_Quality"
            set mode priority
            set priority-members 2 1
            set dst "DC_net"
            set health-check “DC_HealthCheck”
            set link-cost-factor latency
            set link-cost-threshold 10
```

```
        next
    end
end
```

In this example both WAN1 and WAN2 are assumed to have 200ms latency to the health-check server named `DC_HealthCheck`. Because WAN2 is specified before WAN1 in priority-members, SD-WAN parses the two interfaces metric as follows:

- WAN1: 200ms
- WAN2: 200ms / (1+10%) = ~182ms

As a result, WAN2 is selected because the latency is lower.

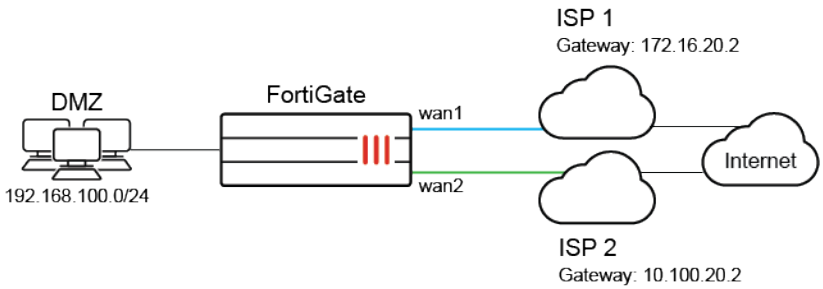
If the *Downstream* (`inbandwidth`), *Upstream* (`outbandwidth`), or *Bandwidth* (`bibandwidth`) quality criteria is used, the FortiGate uses the upstream and downstream bandwidth values configured on the member interfaces to calculate bandwidth.

The interface bandwidth configuration can be done manually, or the interface speedtest can be used to populate the bandwidth values based on the speedtest results. See GUI speed test on page 1239 for details.

To manually configure the upstream and downstream interface bandwidth values:

```
config system interface
    edit <interface>
        set estimated-upstream-bandwidth <speed in kbps>
        set estimated-downstream-bandwidth <speed in kbps>
    next
end
```

Example

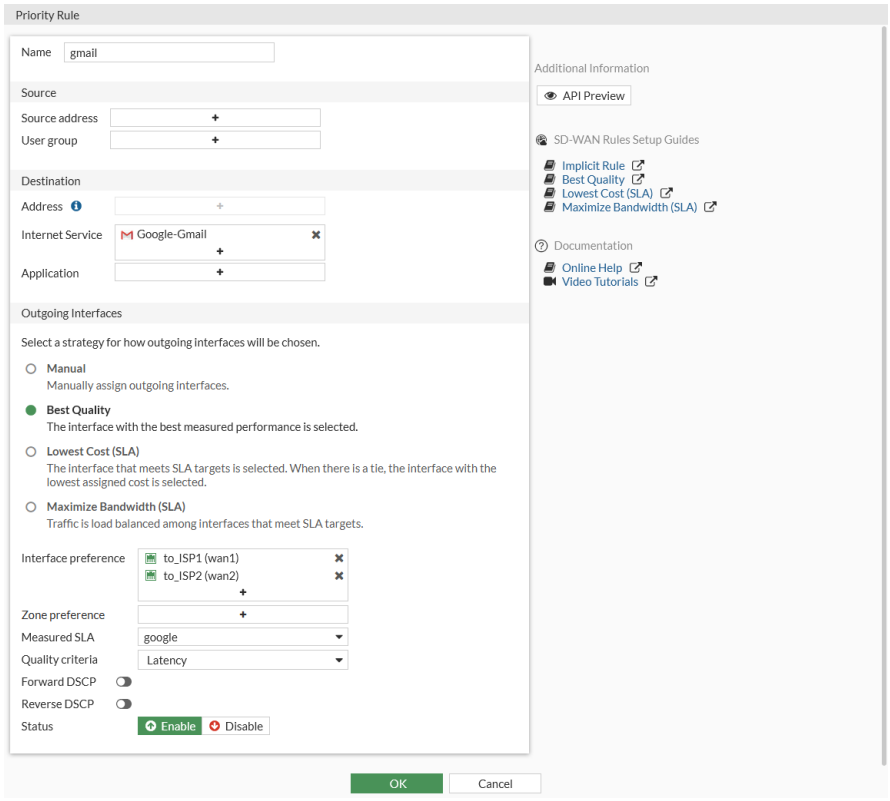


In this example, your wan1 and wan2 SD-WAN interfaces connect to two ISPs that both go to the public internet, and you want Gmail services to use the link with the least latency.

To configure an SD-WAN rule to use Best Quality:

1. On the FortiGate, add wan1 and wan2 as SD-WAN members, then add a policy and static route. See SD-WAN quick start on page 1041 for more details.
2. Go to *Network* > *SD-WAN*, select the *Performance SLAs* tab, and click *Create New*.
3. Enter a name for the performance SLA, such as *google*, and set the *Server* to *google.com*. See Health checks for more details.
4. Click *OK*.

5. Go to *Network* > *SD-WAN*, select the *SD-WAN Rules* tab, and click *Create New*.
6. Enter a name for the rule, such as *gmail*.
7. Configure the following settings:



Internet Service	Google-Gmail
Strategy	Best Quality
Interface preference	wan1 and wan2
Measured SLA	google
Quality criteria	Latency

8. Click *OK*.

To configure an SD-WAN rule to use priority:

```
config system sdwan
    config health-check
        edit "google"
            set server "google.com"
            set members 1 2
        next
    end
    config service
        edit 1
```

```
            set name "gmail"
            set mode priority
            set internet-service enable
            set internet-service-id 65646
            set health-check "google"
            set link-cost-factor latency
            set priority-members 1 2
        next
    end
end
```

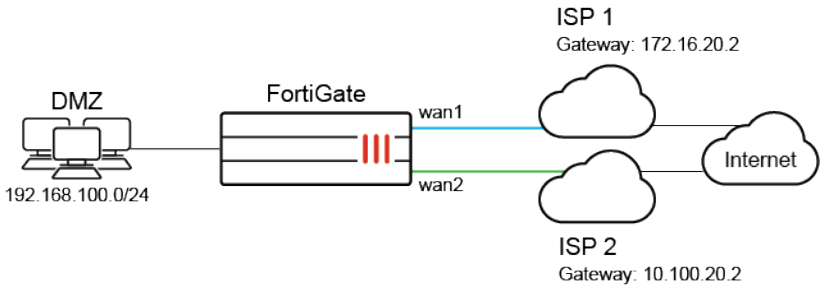
To diagnose the Performance SLA status:

```
FGT # diagnose sys sdwan health-check google
Health Check(google):
Seq(1): state(alive), packet-loss(0.000%) latency(14.563), jitter(4.334) sla_map=0x0
Seq(2): state(alive), packet-loss(0.000%) latency(12.633), jitter(6.265) sla_map=0x0

FGT # diagnose sys sdwan service4 1
Service(1):

    TOS(0x0/0x0), protocol(0: 1->65535), Mode(priority), link-cost-facotr(latency), link-cost-
threshold(10), health-check(google) Members:

        1: Seq_num(2), alive, latency: 12.633, selected
        2: Seq_num(1), alive, latency: 14.563, selected

    Internet Service: Google-Gmail(65646)
```

As wan2 has a smaller latency, SD-WAN will put Seq_num(2) on top of Seq_num(1) and wan2 will be used to forward Gmail traffic.

Lowest cost (SLA) strategy

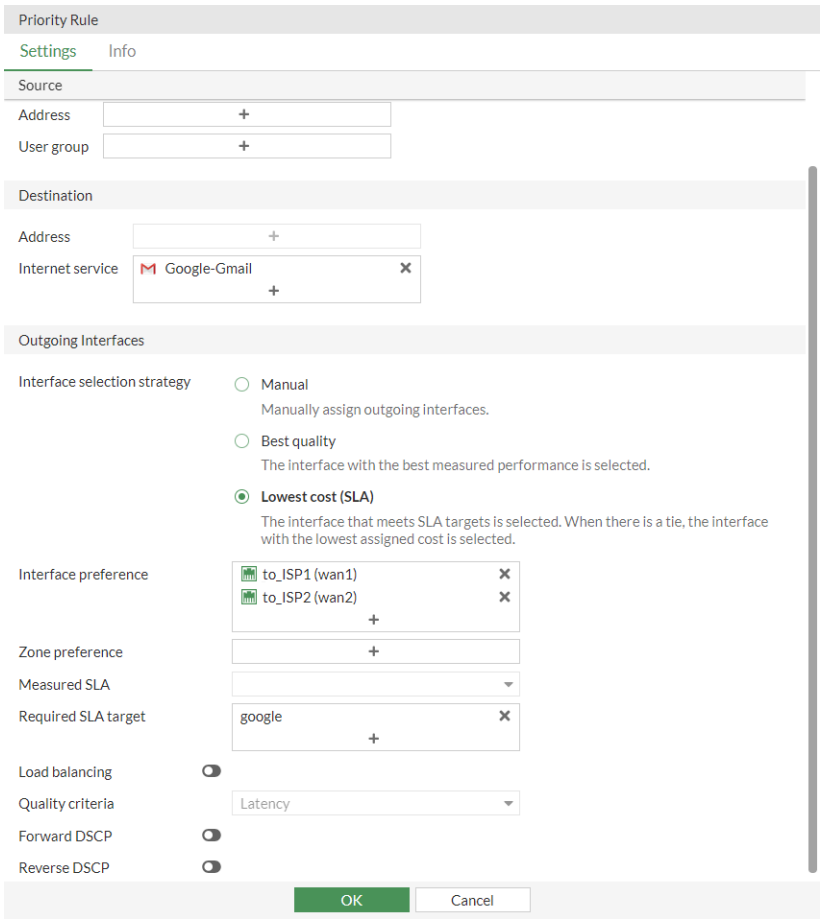
When using *Lowest Cost (SLA)* mode (`sla` in the CLI), SD-WAN will choose the lowest cost link that satisfies SLA to forward traffic. The lowest possible cost is `0`. If multiple eligible links have the same cost, the *Interface preference* order will be used to select a link.



In this example, your wan1 and wan2 SD-WAN interfaces connect to two ISPs that both go to the public internet. The cost of wan2 is less than that of wan1. You want to configure Gmail services to use the lowest cost interface, but the link quality must meet a standard of latency: 10ms, and jitter: 5ms.

To configure an SD-WAN rule to use Lowest Cost (SLA):

1. On the FortiGate, add wan1 and wan2 as SD-WAN members, then add a policy and static route. See SD-WAN quick start on page 1041 for details.
2. Go to *Network* > *SD-WAN*, select the *Performance SLAs* tab, and click *Create New*.
3. Enter a name for the performance SLA, such as *google*, and set the *Server* to *google.com*.
4. Enable *SLA Target*. Set the *Latency threshold* to *10 ms*, and the *Jitter threshold* to *5 ms*. See Health checks for more details.
5. Click *OK*.
6. Go to *Network* > *SD-WAN*, select the *SD-WAN Rules* tab, and click *Create New*.
7. Enter a name for the rule, such as *gmail*.
8. Configure the following settings:



Internet Service	Google-Gmail
Strategy	Lowest Cost (SLA)
Interface preference	wan1 and wan2
Required SLA target	google

9. Click *OK*.

To configure an SD-WAN rule to use SLA:

```
config system sdwan
    config members
        edit 1
            set interface "wan1"
            set cost 10
        next
        edit 2
            set interface "wan2"
            set cost 5
        next
    end
    config health-check
        edit "google"
            set server "google.com"
            set members 1 2
            config sla
                edit 1
                    set latency-threshold 10
                    set jitter-threshold 5
                next
            end
        next
    end
    config service
        edit 1
            set name "gmail"
            set mode sla
            set internet-service enable
            set internet-service-id 65646
            config sla
                edit "google"
                    set id 1
                next
            end
            set priority-members 1 2
        next
    end
end
```

If no SD-WAN zone is specified, members are added to the default *virtual-wan-link* zone.

The CLI command `set minimum-sla-meet-members` allows you to specify the number of links that must meet SLA for the rule to take effect. If the number of members is less than the minimum set with this command, the rule will not take effect.

To diagnose the performance SLA status:

```
FGT # diagnose sys sdwan health-check status
Health Check(google):
Seq(1): state(alive), packet-loss(0.000%) latency(14.563), jitter(4.334) sla_map=0x0
Seq(2): state(alive), packet-loss(0.000%) latency(12.633), jitter(6.265) sla_map=0x0

FGT # diagnose sys sdwan service4 1
Service(1): Address Mode(IPV4) flags=0x0

    TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla)
    Members:<<BR>>

        1: Seq_num(2), alive, sla(0x1), cfg_order(1), selected
        2: Seq_num(1), alive, sla(0x1), cfg_order(0), selected

    Internet Service: Google.Gmail(65646)
```

When both wan1 and wan2 meet the SLA requirements, Gmail traffic will only use wan2. If only wan1 meets the SLA requirements, Gmail traffic will only use wan1, even though it has a higher cost. If neither interface meets the requirements, wan2 will be used.

If both interface had the same cost and both met the SLA requirements, the first link configured in `set priority-members` would be used.

Load balancing strategy with SLA targets

SD-WAN rules can be configured to load balance traffic out of all the interfaces that satisfy the SLA target.

The load balancing strategy known as maximize bandwidth (`load-balance`) prior FortiOS 7.4.1 is now configured within *Lowest Cost (SLA)* mode (`sla`) SD-WAN rules.

By enabling load balancing mode (`set load-balance enable`) inside the lowest cost SD-WAN rule, SD-WAN will choose all of the links that satisfy the SLA target to forward traffic based on a load balancing algorithm. The load balancing algorithm, or hash method, can be one of the following:

`round-robin`	All traffic is distributed to selected interfaces in equal portions and circular order. This is the default method, and the only option available when using the GUI.
`source-ip-based`	All traffic from a source IP is sent to the same interface.
`source-dest-ip-based`	All traffic from a source IP to a destination IP is sent to the same interface.
`inbandwidth`	All traffic is distributed to a selected interface with most available bandwidth for incoming traffic.
`outbandwidth`	All traffic is distributed to a selected interface with most available bandwidth for outgoing traffic.
`bibandwidth`	All traffic is distributed to a selected interface with most available bandwidth for both incoming and outgoing traffic.

When the `inbandwidth`, `outbandwidth`, or `bibandwidth` load balancing algorithm is used, the FortiGate will compare the bandwidth based on the configured upstream and downstream bandwidth values.

The interface speedtest can be used to populate the bandwidth values based on the speedtest results. See GUI speed test on page 1239 for details.

To manually configure the upstream and downstream bandwidth values:

```
config system interface
    edit <interface>
        set estimated-upstream-bandwidth <speed in kbps>
        set estimated-downstream-bandwidth <speed in kbps>
    next
end
```

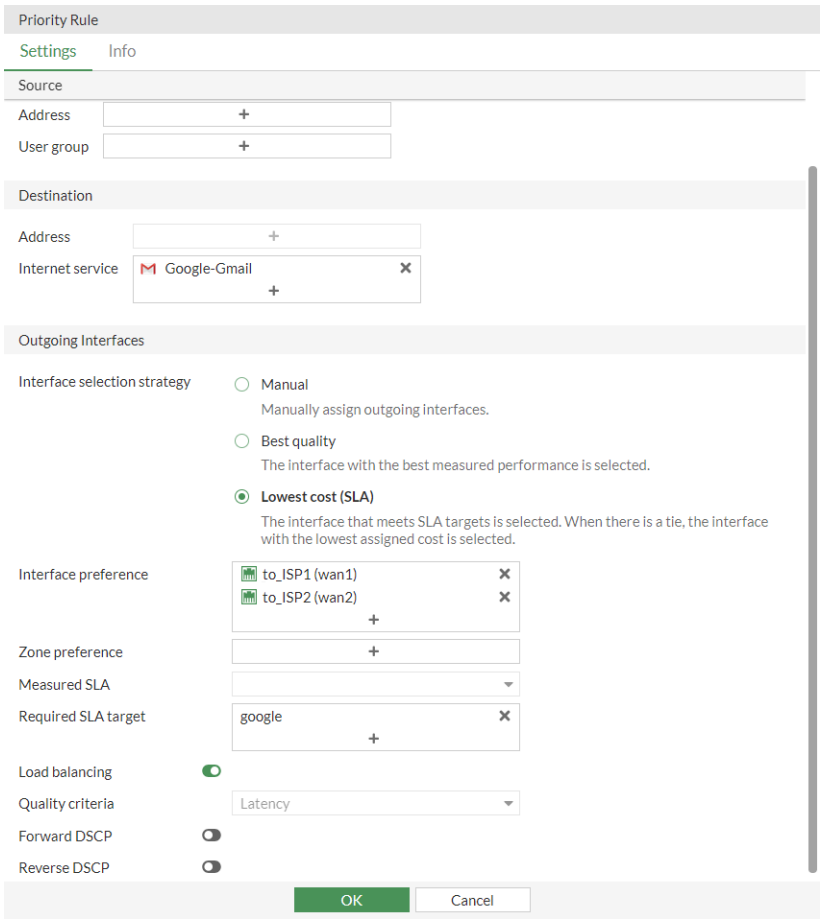
Example

Based on the same topology as the preceding example, your wan1 and wan2 SD-WAN interfaces connect to two ISPs that both go to the public internet. You want to configure Gmail services to use both of the interface, but the link quality must meet a standard of latency: 10ms, and jitter: 5ms. This can maximize the bandwidth usage by using load balancing.

To configure an SD-WAN rule to use load balancing with SLA targets in the GUI:

1. On the FortiGate, add wan1 and wan2 as SD-WAN members, then add a policy and static route. See SD-WAN quick start on page 1041 for details.
2. Go to *Network* > *SD-WAN*, select the *Performance SLAs* tab, and click *Create New*.
3. Enter a name for the performance SLA, such as *google*, and set the *Server* to *google.com*.
4. Enable *SLA Target*. Set the *Latency threshold* to *10 ms*, and the *Jitter threshold* to *5 ms*. See Health checks for more details.
5. Click *OK*.
6. Go to *Network* > *SD-WAN*, select the *SD-WAN Rules* tab, and click *Create New*.
7. Enter a name for the rule, such as *gmail*.

8. Configure the following settings:



Internet Service	Google-Gmail
Strategy	Lowest Cost (SLA)
Interface preference	wan1 and wan2
Required SLA target	google
Load balancing	Enable this setting

9. Click *OK*.

To configure an SD-WAN rule to use load balancing with SLA targets in the CLI:

```
config system sdwan
   config members
        edit 1
            set interface "wan1"
            set cost 10
        next
        edit 2
```

```
            set interface "wan2"
            set cost 5
        next
    end
    config health-check
        edit "google"
            set server "google.com"
            set members 1 2
            config sla
                edit 1
                    set latency-threshold 10
                    set jitter-threshold 5
                next
            end
        next
    end
   config service
        edit 1
            set name "gmail"
            set load-balance enable
            set mode sla
            set internet-service enable
            set internet-service-name "Google-Gmail"
            config sla
                edit "google"
                    set id 1
                next
            end
            set priority-members 1 2
        next
    end
end
```

To diagnose the performance SLA status:

```
FGT # diagnose sys sdwan health-check status
Health Check(google):
Seq(1): state(alive), packet-loss(0.000%) latency(14.563), jitter(4.334) sla_map=0x0
Seq(2): state(alive), packet-loss(0.000%) latency(12.633), jitter(6.265) sla_map=0x0

FGT # diagnose sys sdwan service4 1
Service(1): Address Mode(IPV4) flags=0x0

    TOS(0x0/0x0), Protocol(0: 1->65535), Mode(load-balance)
    Members:<<BR>>

        1: Seq_num(1), alive, sla(0x1), num of pass(1), selected
        2: Seq_num(2), alive, sla(0x1), num of pass(1), selected

    Internet Service: Google.Gmail(65646)
```

When both wan1 and wan2 meet the SLA requirements, Gmail traffic will use both wan1 and wan2. If only one of the interfaces meets the SLA requirements, Gmail traffic will only use that interface.

If neither interface meets the requirements but the `health-check` is still alive, then wan1 and wan2 tie. The traffic will try to balance between wan1 and wan2, using both interfaces to forward traffic.

The maximize bandwidth (`load-balance`) strategy used prior to FortiOS 7.4.1 is now known as the load balancing strategy. This strategy can be configured under the manual mode and the lowest cost (SLA) strategies.

- When the load balancing strategy is configured under the manual mode strategy, SLA targets are not used.
- When the load balancing strategy is configured under the lowest cost (SLA) strategy, SLA targets are used.

  The load balancing strategy functionality remains the same as the maximum bandwidth (SLA) strategy when it is configured inside the lowest cost (SLA) strategy: load balance traffic out of all the interfaces that satisfy the SLA targets. Interface cost is not considered when selecting the best path when the load balancing strategy is used.

Load balancing strategy

The maximize bandwidth (`load-balance`) strategy used prior to FortiOS 7.4.1 is now known as the load balancing strategy. This strategy can be configured under the manual mode and the lowest cost (SLA) strategies.

- When the load balancing strategy is configured under the manual mode strategy, SLA targets are not used (see Load balancing strategy without SLA targets for an example configuration).
- When the load balancing strategy is configured under the lowest cost (SLA) strategy, SLA targets are used (see Load balancing strategy with SLA targets for an example configuration).

The load balancing strategy functionality remains the same as the maximum bandwidth (SLA) strategy when it is configured inside the lowest cost (SLA) strategy: load balance traffic out of all the interfaces that satisfy the SLA targets. Interface cost is not considered when selecting the best path when the load balancing strategy is used.

Hybrid strategy of priority and SLA modes

A hybrid strategy for service rules is where priority mode and SLA mode work together for sla-mode service rules. See Best quality strategy on page 1123 and Lowest cost (SLA) strategy on page 1126 for more information about priority mode and SLA mode.

With the hybrid strategy, SD-WAN selects the best member based on both SLA value and link quality (latency, jitter, packet-loss). Additionally, health-check supports `custom-profile` as a `link-cost-factor`, allowing for a more customized SLA evaluation based on various link quality metrics.

In the hybrid mode, the sla-mode service rule selects the outgoing interface based on the following criteria:

1. Highest SLA Value: The SD-WAN member with the highest SLA value is chosen.
2. Tie-breaker 1: If multiple members have the same SLA value, the system compares members' cost to determine the better option.
3. Tie-breaker 2: If there is still a tie, members' quality based on specified link-cost-factor is considered and is used for ranking.
4. Tie-breaker 3: If there is still a tie, cfg_order is considered.

The following CLI commands are available:

```
config system sdwan
    config health-check
        edit <health-check name>
            set packet-loss-weight <weight>
            set latency-weight <weight>
            set jitter-weight <weight>
            set bandwidth-weight <weight>
            config sla
                edit 1
                    set link-cost-factor custom-profile-1
                    set custom-profile-threshold <threshold>
                next
            end
        next
    end
    config service
        edit 1
            set mode sla
            set tie-break priority
            set link-cost-factor custom-profile-1
            set link-cost-threshold <threshold>
            config sla
                edit <health-check name>
                    set id 1
                next
            end
        next
    end
end
```

Option	Description
`packet-loss-weight <weight>`	Coefficient of packet-loss in the formula of custom-profile-1 (0 - 10000000, default = 0).
`latency-weight <weight>`	Coefficient of latency in the formula of custom-profile-1 (0 - 10000000, default = 0).
`jitter-weight <weight>`	Coefficient of jitter in the formula of custom-profile-1 (0 - 10000000, default = 0).
`bandwidth-weight <weight>`	Coefficient of reciprocal of available bidirectional bandwidth in the formula of custom-profile-1 (0 - 10000000, default = 0).

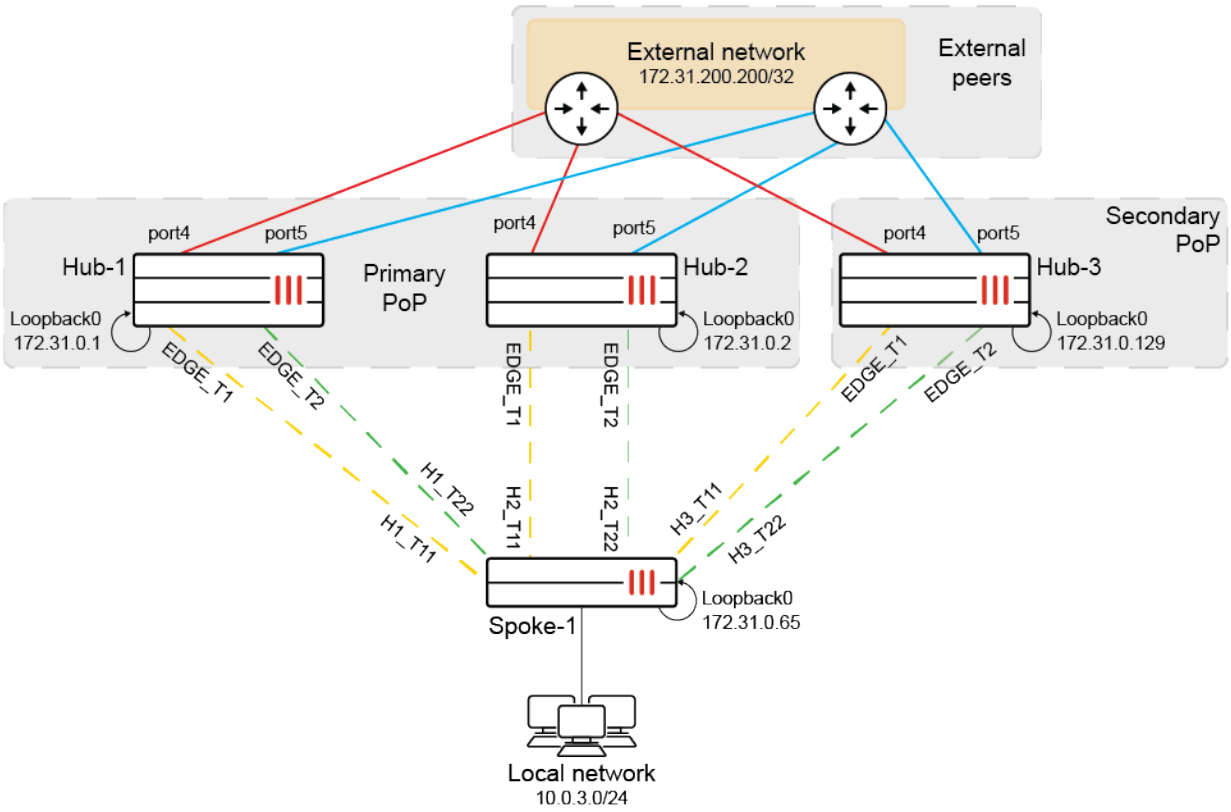
Option	Description
`link-cost-factor custom-profile-1`	Custom profile allowing for a more customized SLA evaluation based on various link quality metrics.
`custom-profile-threshold <threshold>`	Custom profile threshold for SLA to be marked as pass (0 - 10000000, default = 0).
`tie-break priority`	Select the best members that meet the SLA based on link-cost-factor. This command is only configurable for sla-mode service rule.
`link-cost-factor custom-profile-1`	Link cost factor. This command is only configurable when `tie-break` is set to `priority`.
`link-cost-threshold <threshold>`	Percentage threshold change of link cost values that will result in policy route regeneration (0 - 10000000, default = 10). This command is only configurable when `tie-break` is set to `priority`.

Example

This example shows how the sla-mode service rule selects the outgoing interface based on different criteria as members experience various degradation.

The configuration contains the following components:

- Two PoPs:
  - The primary PoP has two hubs (Hub-1 and Hub-2).
  - The secondary PoP has one hub (Hub-3).
- Spoke-1 has six overlays, with two overlay connections to each hub.
- Spoke-1 has three BGP neighbors, with one BGP neighbor for each hub.
  - All BGP neighbors are established on loopback IPs.
- Each hub has two paths to external peers.



1. Configure Spoke-1:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
        edit "PoP1"
            set minimum-sla-meet-members 2
        next
        edit "PoP2"
        next
    end
    config members
        edit 1
            set interface "H1_T11"
            set zone "PoP1"
        next
        edit 2
            set interface "H1_T22"
            set zone "PoP1"
        next
        edit 3
            set interface "H2_T11"
            set zone "PoP1"
        next
```

```
        edit 4
            set interface "H2_T22"
            set zone "PoP1"
        next
        edit 5
            set interface "H3_T11"
            set zone "PoP2"
        next
        edit 6
            set interface "H3_T22"
            set zone "PoP2"
        next
    end
    config health-check
        edit "Hubs"
            set server "172.31.100.100"
            set source 172.31.0.65
            set members 0
            set packet-loss-weight 1
            set latency-weight 1
            set jitter-weight 1
            config sla
                edit 1
                    set link-cost-factor latency packet-loss jitter custom-profile-1
                    set latency-threshold 100
                    set jitter-threshold 20
                    set packetloss-threshold 10
                    set custom-profile-threshold 60
                next
            end
        next
    end
    config service
        edit 1
            set mode sla
            set zone-mode enable
            set dst "all"
            set src "CORP_LAN"
            set tie-break priority
            set link-cost-factor packet-loss
            set link-cost-threshold 1
            config sla
                edit "Hubs"
                    set id 1
                next
            end
            set priority-zone "PoP1" "PoP2"
        next
    end
end
```

2. Check the initial SD-WAN status on Spoke-1:

```
# diagnose sys sdwan health-check
Health Check(Hubs):
Seq(1 H1_T11): state(alive), packet-loss(0.000%), latency(0.256), jitter(0.030), mos(4.404),
custom_profile(0.286), bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996), sla_
map=0x1
Seq(2 H1_T22): state(alive), packet-loss(0.000%), latency(0.215), jitter(0.010), mos(4.404),
custom_profile(0.225), bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997), sla_
map=0x1
Seq(3 H2_T11): state(alive), packet-loss(0.000%), latency(0.219), jitter(0.022), mos(4.404),
custom_profile(0.241), bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996), sla_
map=0x1
Seq(4 H2_T22): state(alive), packet-loss(0.000%), latency(0.208), jitter(0.013), mos(4.404),
custom_profile(0.221), bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997), sla_
map=0x1
Seq(5 H3_T11): state(alive), packet-loss(0.000%), latency(0.199), jitter(0.014), mos(4.404),
custom_profile(0.213), bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996), sla_
map=0x1
Seq(6 H3_T22): state(alive), packet-loss(0.000%), latency(0.191), jitter(0.010), mos(4.404),
custom_profile(0.200), bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997), sla_
map=0x1
```

Zone PoP1 is preferred over zone PoP2 because the number of in-sla members in PoP1 is four, which is more than the configured `minimum-sla-meet-members`.

In the preferred zone PoP1, all members have the same SLA value (0x01), cost (0), and link quality (packet-loss 0.000%). As a result, the first member in the cfg-order, H1_T11, is selected as the best outgoing interface.

```
# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x14200 use-shortcut-sla use-shortcut
 Tie break: priority
 Shortcut priority: 2
  Gen(4765), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), link-cost-
factor(packet-loss), link-cost-threshold(1), sla-compare-order
  Service role: standalone
  Members(6):
    1: Seq_num(1 H1_T11 PoP1  active), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
packet loss: 0.000%, selected
    2: Seq_num(2 H1_T22 PoP1  active), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
packet loss: 0.000%, selected
    3: Seq_num(3 H2_T11 PoP1  active), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
packet loss: 0.000%, selected
    4: Seq_num(4 H2_T22 PoP1  active), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
packet loss: 0.000%, selected
    5: Seq_num(5 H3_T11 PoP2 standby), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
packet loss: 0.000%, selected
    6: Seq_num(6 H3_T22 PoP2 standby), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
packet loss: 0.000%, selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255
```

Traffic matching service 1 selects H1_T11 as outgoing interface:

```
# diagnose sniffer packet any 'host 172.31.200.200' 4
interfaces=[any]
filters=[host 172.31.200.200]
2.863817 port4 in 10.0.3.2 -> 172.31.200.200: icmp: echo request
2.863926 H1_T11 out 10.0.3.2 -> 172.31.200.200: icmp: echo request
2.864236 H1_T11 in 172.31.200.200 -> 10.0.3.2: icmp: echo reply
2.864389 port4 out 172.31.200.200 -> 10.0.3.2: icmp: echo reply
3.862809 port4 in 10.0.3.2 -> 172.31.200.200: icmp: echo request
3.862836 H1_T11 out 10.0.3.2 -> 172.31.200.200: icmp: echo request
3.863040 H1_T11 in 172.31.200.200 -> 10.0.3.2: icmp: echo reply
3.863072 port4 out 172.31.200.200 -> 10.0.3.2: icmp: echo reply
```

3. Introduce packet-loss on some members, then check the SD-WAN status on Spoke-1:

```
# diagnose sys sdwan  health-check
Health Check(Hubs):
Seq(1 H1_T11): state(alive), packet-loss(8.000%), latency(0.259), jitter(0.026), mos(4.400),
custom_profile(8.285), bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997), sla_
map=0x1
Seq(2 H1_T22): state(alive), packet-loss(6.000%), latency(0.253), jitter(0.083), mos(4.401),
custom_profile(6.337), bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998), sla_
map=0x1
Seq(3 H2_T11): state(alive), packet-loss(5.000%), latency(0.228), jitter(0.028), mos(4.402),
custom_profile(5.256), bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997), sla_
map=0x1
Seq(4 H2_T22): state(alive), packet-loss(1.000%), latency(0.219), jitter(0.014), mos(4.404),
custom_profile(1.232), bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998), sla_
map=0x1
Seq(5 H3_T11): state(alive), packet-loss(0.000%), latency(0.229), jitter(0.068), mos(4.404),
custom_profile(0.297), bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997), sla_
map=0x1
Seq(6 H3_T22): state(alive), packet-loss(0.000%), latency(0.191), jitter(0.016), mos(4.404),
custom_profile(0.207), bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998), sla_
map=0x1
```

Zone PoP1 is preferred over zone PoP2 because the number of in-sla members in PoP1 is four, which is more than the configured `minimum-sla-meet-members`.

In the preferred zone PoP1, all members have the same SLA value (0x01) and cost (0). As a result, the members' quality based on packet-loss is considered, and the member H2_T22 with lowest packet-loss is selected as the best outgoing interface.

```
# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x14200 use-shortcut-sla use-shortcut
 Tie break: priority
 Shortcut priority: 2
  Gen(7067), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), link-cost-
factor(packet-loss), link-cost-threshold(1), sla-compare-order
  Service role: standalone
  Members(6):
```

```
    1: Seq_num(4 H2_T22 PoP1  active), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
packet loss: 1.000%, selected
    2: Seq_num(3 H2_T11 PoP1  active), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
packet loss: 5.000%, selected
    3: Seq_num(2 H1_T22 PoP1  active), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
packet loss: 6.000%, selected
    4: Seq_num(1 H1_T11 PoP1  active), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
packet loss: 8.000%, selected
    5: Seq_num(5 H3_T11 PoP2 standby), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
packet loss: 0.000%, selected
    6: Seq_num(6 H3_T22 PoP2 standby), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
packet loss: 0.000%, selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255
```

Traffic matching service 1 selects H2_T22 as outgoing interface:

```
# diagnose sniffer packet any 'host 172.31.200.200' 4
interfaces=[any]
filters=[host 172.31.200.200]
3.755271 port4 in 10.0.3.2 -> 172.31.200.200: icmp: echo request
3.755377 H2_T22 out 10.0.3.2 -> 172.31.200.200: icmp: echo request
3.755708 H2_T22 in 172.31.200.200 -> 10.0.3.2: icmp: echo reply
3.755759 port4 out 172.31.200.200 -> 10.0.3.2: icmp: echo reply
4.754278 port4 in 10.0.3.2 -> 172.31.200.200: icmp: echo request
4.754352 H2_T22 out 10.0.3.2 -> 172.31.200.200: icmp: echo request
4.754572 H2_T22 in 172.31.200.200 -> 10.0.3.2: icmp: echo reply
4.754617 port4 out 172.31.200.200 -> 10.0.3.2: icmp: echo reply
```

4. Introduce some latency on H2_T22, then check the SD-WAN status on Spoke-1:

   On H2_T22, custom_profile = (weight 1 * packet-loss 1.190) + (weight 1 * latency 60.202) + (weight 1 * jitter 0.010) = 61.402, which is above the threshold of 60, so H2_T22 is out of SLA.

```
# diagnose sys sdwan  health-check
Health Check(Hubs):
Seq(1 H1_T11): state(alive), packet-loss(8.333%), latency(0.257), jitter(0.034), mos(4.400),
custom_profile(8.623), bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997), sla_
map=0x1
Seq(2 H1_T22): state(alive), packet-loss(6.052%), latency(0.219), jitter(0.012), mos(4.401),
custom_profile(6.183), bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998), sla_
map=0x1
Seq(3 H2_T11): state(alive), packet-loss(5.271%), latency(0.214), jitter(0.012), mos(4.403),
custom_profile(3.797), bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997), sla_
map=0x1
Seq(4 H2_T22): state(alive), packet-loss(1.190%), latency(60.202), jitter(0.010), mos(4.372),
custom_profile(61.402), bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998),
sla_map=0x0
Seq(5 H3_T11): state(alive), packet-loss(0.000%), latency(0.224), jitter(0.074), mos(4.404),
custom_profile(0.298), bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997), sla_
map=0x1
```

```
Seq(6 H3_T22): state(alive), packet-loss(0.000%), latency(0.194), jitter(0.017), mos(4.404),
custom_profile(0.211), bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998), sla_
map=0x1
```

Zone PoP1 is preferred over zone PoP2 because the number of in-sla members in PoP1 is three, which is more than the configured `minimum-sla-meet-members`.

In the preferred zone PoP1, H2_T11, H1_T22 and H1_T11 have the same highest SLA value (0x01) and cost (0). As a result, the members' quality based on packet-loss is considered and the member H2_T11 with lowest packet-loss is selected as the best outgoing interface.

```
# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x14200 use-shortcut-sla use-shortcut
 Tie break: priority
 Shortcut priority: 2
  Gen(347), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), link-cost-
factor(packet-loss), link-cost-threshold(1), sla-compare-order
  Service role: standalone
  Members(6):
    1: Seq_num(3 H2_T11 PoP1  active), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
packet loss: 5.000%, selected
    2: Seq_num(2 H1_T22 PoP1  active), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
packet loss: 6.000%, selected
    3: Seq_num(1 H1_T11 PoP1  active), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
packet loss: 8.000%, selected
    4: Seq_num(5 H3_T11 PoP2 standby), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
packet loss: 0.000%, selected
    5: Seq_num(6 H3_T22 PoP2 standby), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
packet loss: 0.000%, selected
    6: Seq_num(4 H2_T22 PoP1 standby), alive, sla(0x0), gid(0), cfg_order(0), local cost(0),
packet loss: 1.000%, selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255
```

Traffic matching service 1 selects H2_T11 as outgoing interface:

```
# diagnose sniffer packet any 'host 172.31.200.200' 4
interfaces=[any]
filters=[host 172.31.200.200]
2.235538 port4 in 10.0.3.2 -> 172.31.200.200: icmp: echo request
2.235629 H2_T11 out 10.0.3.2 -> 172.31.200.200: icmp: echo request
2.235955 H2_T11 in 172.31.200.200 -> 10.0.3.2: icmp: echo reply
2.235990 port4 out 172.31.200.200 -> 10.0.3.2: icmp: echo reply
3.234544 port4 in 10.0.3.2 -> 172.31.200.200: icmp: echo request
3.234570 H2_T11 out 10.0.3.2 -> 172.31.200.200: icmp: echo request
3.234776 H2_T11 in 172.31.200.200 -> 10.0.3.2: icmp: echo reply
3.234811 port4 out 172.31.200.200 -> 10.0.3.2: icmp: echo reply
```

5. Introduce some latency on H2_T11 and H1_T22, then check the SD-WAN status on Spoke-1:
   H2_T11, H1_T22, and H2_T22 are out of SLA becasue custom_profile on them are above the threshold of 60.

```
# diagnose sys sdwan  health-check
Health Check(Hubs):
Seq(1 H1_T11): state(alive), packet-loss(9.000%), latency(0.281), jitter(0.023), mos(4.400),
custom_profile(9.305), bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997), sla_
map=0x1
Seq(2 H1_T22): state(alive), packet-loss(6.000%), latency(60.233), jitter(0.021), mos(4.369),
custom_profile(66.254), bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997),
sla_map=0x0
Seq(3 H2_T11): state(alive), packet-loss(4.000%), latency(60.214), jitter(0.016), mos(4.371),
custom_profile(64.231), bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997),
sla_map=0x0
Seq(4 H2_T22): state(alive), packet-loss(1.000%), latency(60.236), jitter(0.020), mos(4.372),
custom_profile(61.255), bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998),
sla_map=0x0
Seq(5 H3_T11): state(alive), packet-loss(0.000%), latency(0.217), jitter(0.015), mos(4.404),
custom_profile(0.232), bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997), sla_
map=0x1
Seq(6 H3_T22): state(alive), packet-loss(0.000%), latency(0.223), jitter(0.028), mos(4.404),
custom_profile(0.251), bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998), sla_
map=0x1
```

Zone PoP2 is preferred over zone PoP1 because the number of in-sla members in zone PoP1 is one, which is less than the configured minimum-sla-meet-members.

In the preferred zone PoP2, all members have the same SLA value (0x01), cost (0), and link quality (packet-loss 0.000%). As a result, the first member in cfg-order, H3_T11, is selected as the best outgoing interface.

```
# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x14200 use-shortcut-sla use-shortcut
 Tie break: priority
 Shortcut priority: 2
  Gen(956), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), link-cost-
factor(packet-loss), link-cost-threshold(1), sla-compare-order
  Service role: standalone
  Members(6):
    1: Seq_num(5 H3_T11 PoP2  active), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
packet loss: 0.000%, selected
    2: Seq_num(6 H3_T22 PoP2  active), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
packet loss: 0.000%, selected
    3: Seq_num(4 H2_T22 PoP1 standby), alive, sla(0x0), gid(0), cfg_order(0), local cost(0),
packet loss: 1.000%, selected
    4: Seq_num(3 H2_T11 PoP1 standby), alive, sla(0x0), gid(0), cfg_order(0), local cost(0),
packet loss: 4.000%, selected
    5: Seq_num(2 H1_T22 PoP1 standby), alive, sla(0x0), gid(0), cfg_order(0), local cost(0),
packet loss: 6.000%, selected
    6: Seq_num(1 H1_T11 PoP1 standby), alive, sla(0x0), gid(0), cfg_order(0), local cost(0),
packet loss: 9.000%, selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255
```

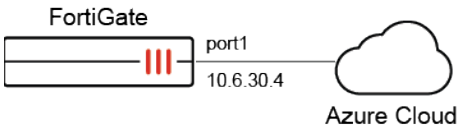
Traffic matching service 1 selects H3_T11 as outgoing interface:

```
# diagnose sniffer packet any 'host 172.31.200.200' 4
interfaces=[any]
filters=[host 172.31.200.200]
3.611593 port4 in 10.0.3.2 -> 172.31.200.200: icmp: echo request
3.611685 H3_T11 out 10.0.3.2 -> 172.31.200.200: icmp: echo request
3.611972 H3_T11 in 172.31.200.200 -> 10.0.3.2: icmp: echo reply
3.612009 port4 out 172.31.200.200 -> 10.0.3.2: icmp: echo reply
4.611686 port4 in 10.0.3.2 -> 172.31.200.200: icmp: echo request
4.611746 H3_T11 out 10.0.3.2 -> 172.31.200.200: icmp: echo request
4.611916 H3_T11 in 172.31.200.200 -> 10.0.3.2: icmp: echo reply
4.611948 port4 out 172.31.200.200 -> 10.0.3.2: icmp: echo reply
```

SDN dynamic connector addresses in SD-WAN rules

SDN dynamic connector addresses can be used in SD-WAN rules. FortiGate supports both public (AWS, Azure, GCP, OCI, AliCloud) and private (Kubernetes, VMware ESXi and NSX, OpenStack, ACI, Nuage) SDN connectors.

The configuration procedure for all of the supported SDN connector types is the same. This example uses an Azure public SDN connector.



There are four steps to create and use an SDN connector address in an SD-WAN rule:

1. Configure the FortiGate IP address and network gateway so that it can reach the Internet.
2. Create an Azure SDN connector.
3. Create a firewall address to associate with the configured SDN connector.
4. Use the firewall address in an SD-WAN service rule.

To create an Azure SDN connector:

1. Go to *Security Fabric* > *External Connectors*.
2. Click *Create New*.
3. In the *Public SDN* section, click *Microsoft Azure*.
4. Enter the following:

Name	azure1
Status	Enabled
Update Interval	Use Default
Server region	Global
Directory ID	942b80cd-1b14-42a1-8dcf-4b21dece61ba
Application ID	14dbd5c5-307e-4ea4-8133-68738141feb1

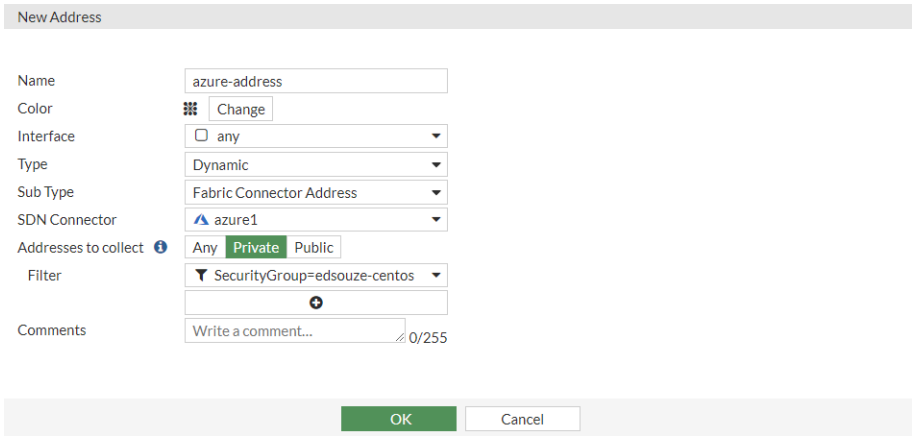
Client secret	xxxxxx
Resource path	disabled

5. Click *OK*.

To create a firewall address to associate with the configured SDN connector:

1. Go to *Policy & Objects* > *Addresses* and select *Address*.
2. Click *Create new*.
3. Enter the following:

Name	azure-address
Type	Dynamic
Sub Type	Fabric Connector Address
SDN Connector	azure1
Addresses to collect	Private
Filter	SecurityGroup=edsouza-centos
Interface	Any



4. Click *OK*.

To use the firewall address in an SD-WAN service rule:

1. Go to *Network* > *SD-WAN*, select the *SD-WAN Rules* tab, and click *Create New*.
2. Set the *Name* to *Azure1*.
3. For the *Destination Address* select *azure-address*.
4. Configure the remaining settings as needed. See SD-WAN rules on page 1105 for details.
5. Click *OK*.

Diagnostics

Use the following CLI commands to check the status of and troubleshoot the connector.

To see the status of the SDN connector:

```
# diagnose sys sdn status
SDN Connector          Type        Status          Updating        Last update
-------------------------------------------------------------------------------
azure1                 azure       connected       no              n/a
```

To debug the SDN connector to resolve the firewall address:

```
# diagnose debug application azd -1
    Debug messages will be on for 30 minutes.

    ...
    azd sdn connector azure1 start updating IP addresses
    azd checking firewall address object azure-address-1, vd 0
     IP address change, new list:
      10.18.0.4
      10.18.0.12
      ...
      ...
```

```
# diagnose sys sdwan service4

    Service(2): Address Mode(IPV4) flags=0x0
      TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
      Service role: standalone
      Member sub interface:
      Members:
        1: Seq_num(1), alive, selected
      Dst address:
            10.18.0.4 - 10.18.0.4
            10.18.0.12 - 10.18.0.12
             ... ...
             ... ...
             ... ...
```

Application steering using SD-WAN rules

This topic covers how to use application steering in a topology with multiple WAN links. The following examples illustrate how to use different strategies to perform application steering to accommodate different business needs:

- Application matching on page 1146
- Static application steering with a manual strategy on page 1147
- Dynamic application steering with lowest cost and best quality strategies on page 1149

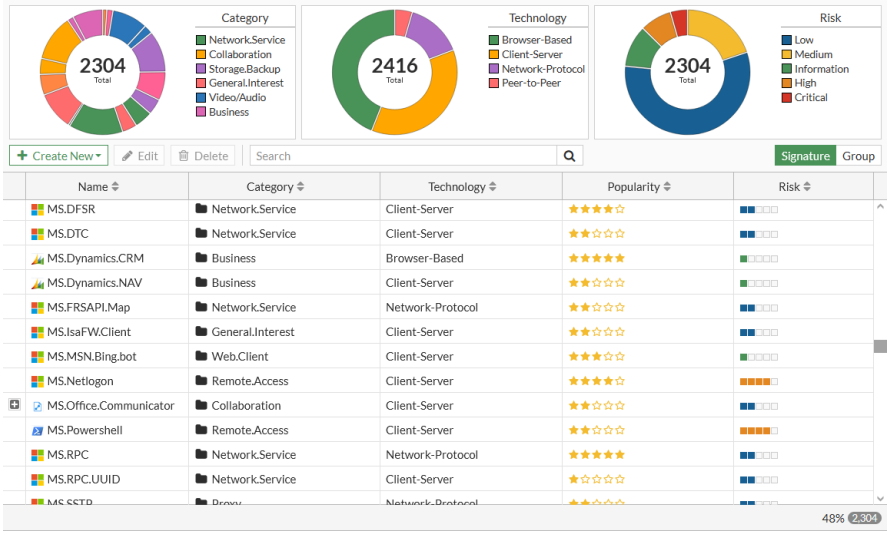
By default, individual applications and application groups cannot be selected in SD-WAN rules. To enable this functionality in the GUI, go to *System > Feature Visibility* and enable *Application Detection Based SD-WAN*. In the CLI, enter:

```
config system global
    set gui-app-detection-sdwan enable
end
```

For application based steering to work, application control must be enabled in a policy. See Application control on page 2428.

Application matching

To apply application steering, SD-WAN service rules match traffic based on the applications that are in the application signature database. To view the signatures, go to *Security Profiles* > *Application Signatures* and select *Signature*.



On the first session that passes through, the IPS engine processes the traffic in the application layer to match it to a signature in the application signature database. The first session does not match any SD-WAN rules because the signature has not been recognized yet. When the IPS engine recognizes the application, it records the 3-tuple IP address, protocol, and port in the application control Internet Service ID list. To view the application and corresponding 3-tuple:

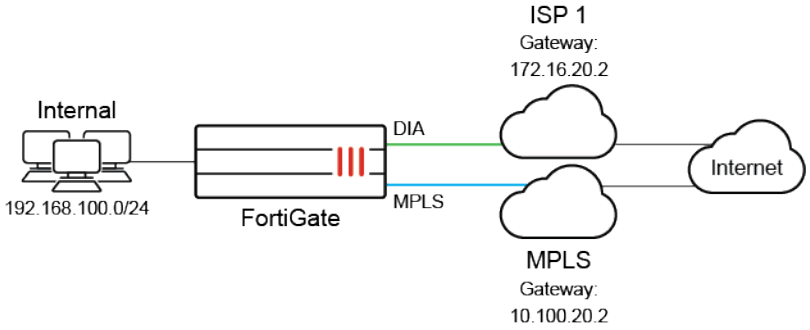
```
# diagnose sys sdwan internet-service-app-ctrl-list [app ID]
52.114.142.254
Microsoft.Teams(43541 4294837333): 52.114.142.254 6 443 Fri Jun 18 13:52:18 2021
```

The recognized application and 3-tuple stay in the application control list for future matches to occur. If there are no hits on the entry for eight hours, the entry is deleted.

For services with multiple IP addresses, traffic might not match the expected SD-WAN rule because the traffic is destined for an IP address that hat no previously been recognized by the FortiGate. The `diagnose sys sdwan internet-service-app-ctrl-list` command can be used to help troubleshoot such situations.

Static application steering with a manual strategy

This example covers a typical usage scenario where the SD-WAN has two members: MPLS and DIA. DIA is primarily used for direct internet access to internet applications, such as Office365, Google applications, Amazon, and Dropbox. MPLS is primarily used for SIP, and works as a backup when DIA is not working.



This example configures all SIP traffic to use MPLS while all other traffic uses DIA. If DIA is not working, the traffic will use MPLS.

By default, individual applications and application groups cannot be selected in SD-WAN rules. To enable this functionality in the GUI, go to *System > Feature Visibility* and enable *Application Detection Based SD-WAN*. In the CLI, enter:

```
config system global
    set gui-app-detection-sdwan enable
end
```

To configure an SD-WAN rule to use SIP and DIA in the GUI:

1. Add port1 (DIA) and port2 (MPLS) as SD-WAN members, and configure a static route. See Underlay on page 1042 for details.
2. Create a firewall policy with an *Application Control* profile configured. See Security on page 1044 for details.
3. Go to *Network > SD-WAN*, select the *SD-WAN Rules* tab, and click *Create New*.
4. Enter a name for the rule, such as *SIP*.
5. Click the *Application* field and select the applicable SIP applications from the *Select Entries* panel.
6. Under *Outgoing Interfaces*, select *Manual*.
7. For *Interface preference*, select *MPLS*.
8. Click *OK*.
9. Click *Create New* to create another rule.
10. Enter a name for the rule, such as *Internet*.
11. Click the *Address* field and select *all* from the panel.
12. Under *Outgoing Interfaces*, select *Manual*.
13. For *Interface preference*, select *DIA*.
14. Click *OK*.

To configure the firewall policy using the CLI:

```
config firewall policy
    edit 1
        set name "1"
        set srcintf "dmz"
        set dstintf "virtual-wan-link"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set fsso disable
        set application-list "default"
        set ssl-ssh-profile "certificate-inspection"
        set nat enable
    next
end
```

To configure an SD-WAN rule to use SIP and DIA using the CLI:

```
config system sdwan
    set status enable
    config members
        edit 1
            set interface "MPLS"
        next
        edit 2
            set interface "DIA"
        next
    end
    config service
        edit 1
            set name "SIP"
            set internet-service enable
            set internet-service-app-ctrl 34640 152305677 38938 26180 26179 30251
            set priority-members 2
        next
        edit 2
            set name "Internet"
            set dst "all"
            set priority-members 1
        next
    end
end
```

All SIP traffic uses MPLS. All other traffic goes to DIA. If DIA is broken, the traffic uses MPLS. If you use VPN instead of MPLS to run SIP traffic, you must configure a VPN interface, for example vpn1, and then replace member 1 from MPLS to vpn1 for SD-WAN member.

If no SD-WAN zone is specified, members are added to the default *virtual-wan-link* zone.

To use the diagnose command to check performance SLA status using the CLI:

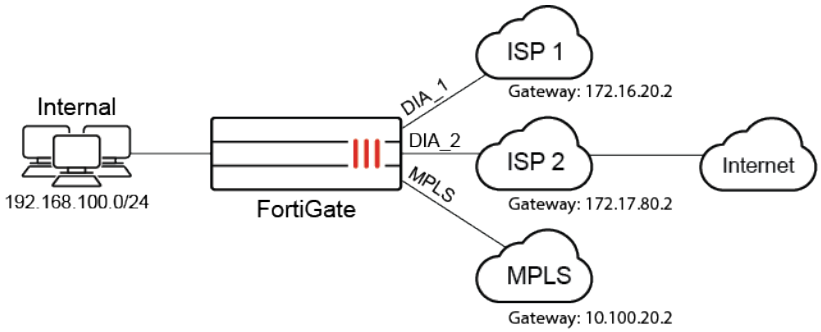
```
# diagnose sys sdwan service4 1

Service(1): Address Mode(IPV4) flags=0x0

TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
Members:<<BR>>

1: Seq_num(1), alive, selected

Internet Service: SIP(4294836224 34640) SIP.Method(4294836225 152305677) SIP.Via.NAT(4294836226
38938) SIP_Media.Type.Application(4294836227 26180) SIP_Message(4294836228 26179) SIP_Voice
(4294836229 30251)
```

```
# diagnose sys sdwan service4 2

Service(2): Address Mode(IPV4) flags=0x0

TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
Members:<<BR>>

1: Seq_num(2), alive, selected

Dst address: 0.0.0.0-255.255.255.255
```

```
# diagnose sys sdwan internet-service-app-ctrl-list
Ctrl application(SIP 34640):Internet Service ID(4294836224)
Ctrl application(SIP.Method 152305677):Internet Service ID(4294836225)
Ctrl application(SIP.Via.NAT 38938):Internet Service ID(4294836226)
Ctrl application(SIP_Media.Type.Application 26180):Internet Service ID(4294836227)
Ctrl application(SIP_Message 26179):Internet Service ID(4294836228)
Ctrl application(SIP_Voice 30251):Internet Service ID(4294836229)
```

Dynamic application steering with lowest cost and best quality strategies

In this example, the SD-WAN has three members: two ISPs (DIA_1 and DIA_2) that are used for access to internet applications, and an MPLS link that is used exclusively as a backup for business critical applications.



Business applications, such as Office365, Google, Dropbox, and SIP, use the *Lowest Cost (SLA)* strategy to provide application steering, and traffic falls back to MPLS only if both ISP1 and ISP2 are down. Non-business applications, such as Facebook and Youtube, use the *Best Quality* strategy to choose between the ISPs.

By default, individual applications and application groups cannot be selected in SD-WAN rules. To enable this functionality in the GUI, go to *System > Feature Visibility* and enable *Application Detection Based SD-WAN*. In the CLI, enter:

```
config system global
    set gui-app-detection-sdwan enable
end
```

To configure the SD-WAN members, static route, and firewall policy in the GUI:

1. Add port1 (DIA_1), port2 (DIA_2), and port3 (MPLS) as SD-WAN members. Set the cost of DIA_1 and DIA_2 to *0*, and MPLS to *20*. See Underlay on page 1042 for details.
2. Configure a static route. See Routing on page 1043 for details.
3. Create a firewall policy to allow traffic out on SD-WAN, with an *Application Control* profile configured. See Security on page 1044 for details.

To configure the SD-WAN rule and performance SLA checks for business critical application in the GUI:

1. Go to *Network > SD-WAN*, select the *SD-WAN Rules* tab, and click *Create New*.
2. Set the name to *BusinessCriticalApps*.
   This rule will steer your business critical traffic to the appropriate link based on the *Lowest Cost (SLA)*.
3. Set *Source address* to *all*.
4. Under *Destination*, set *Application* to your required applications. In this example: *Microsoft.Office.365*, *Microsoft.Office.Online*, *Google.Docs*, *Dropbox*, and *SIP*.
5. Under *Outgoing Interfaces*, select *Lowest Cost (SLA)*.
   The lowest cost is defined in the SD-WAN member interface settings (see Underlay on page 1042). The lowest possible cost is *0*, which represents the most preferred link. In this example, DIA_1 and DIA_2 both have a cost of *0*, while MPLS has a cost of *20* because it is used for backup.
6. In *Interface preference*, add the interfaces in order of preference when the cost of the links is tied. In this example, DIA_1, DIA_2, then MPLS.
   MPLS will always be chosen last, because it has the highest cost. DIA_1 and DIA_2 have the same cost, so an interface is selected based on their order in the *Interface preference* list.
7. Set *Required SLA target* to ensure that only links that pass your SLA target are chosen in this SD-WAN rule:

a. Click in the *Required SLA target* field.

b. In the *Select Entries* pane, click *Create*. The *New Performace SLA* pane opens.

c. Set *Name* to *BusinessCriticalApps_HC*.

This health check is used for business critical applications in your SD-WAN rule.

d. Leave *Protocol* set to *Ping*, and add up to two servers, such as *office.com* and *google.com*.

e. Set *Participants* to *Specify*, and add all three interfaces: DIA_1, DIA_2, and MPLS.

f. Enable *SLA Target*.

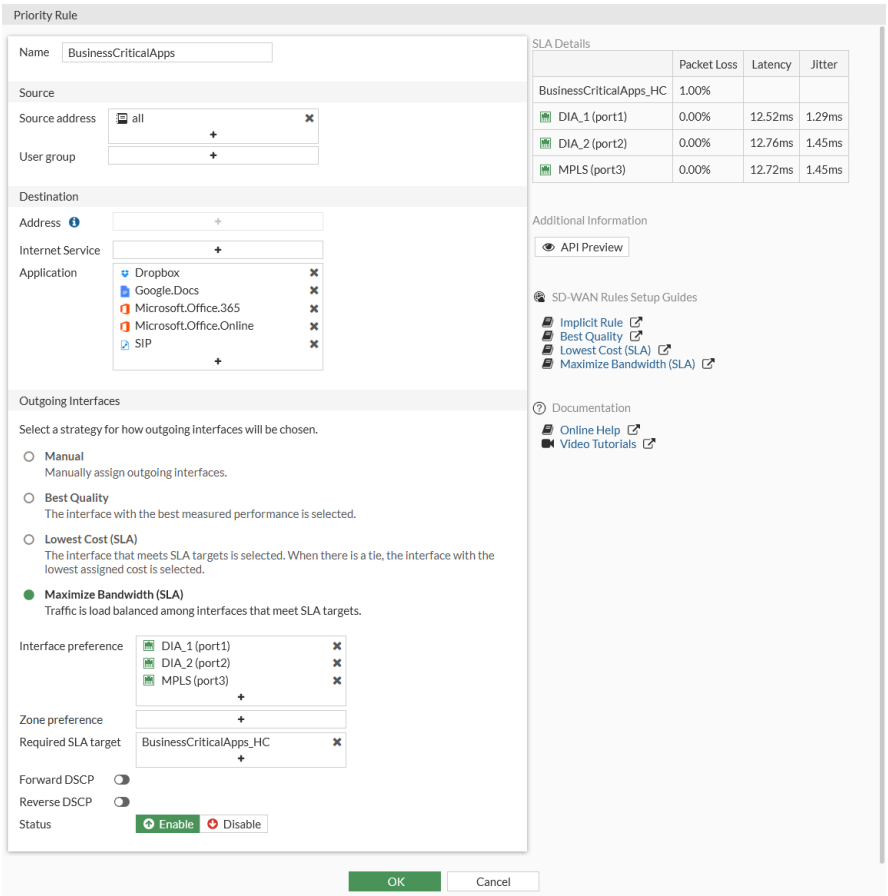
The attributes in your target determine the quality of your link. The SLA target of each link is compared when determining which link to use based on the lowest cost. Links that meet the SLA target are preferred over links that fail, and move to the next step of selection based on cost. If no links meet the SLA target, then they all move to the next step.

In this example, disable *Latency threshold* and *Jitter threshold*, and set *Packet loss threshold* to *1*.

g. Click *OK*.

h. Select the new performance SLA to set it as the *Required SLA target*.

When multiple SLA targets are added, you can choose which target to use in the SD-WAN rule.



8. Click *OK* to create the SD-WAN rule.

To configure the SD-WAN rule and performance SLA checks for non-business critical application in the GUI:

1. Go to *Network* > *SD-WAN*, select the *SD-WAN Rules* tab, and click *Create New*.
2. Set the name to *NonBusinessCriticalApps*.

   This rule will steer your non-business critical traffic to the appropriate link based on the *Best Quality*. No SLA target must be met, as the best link is selected based on the configured quality criteria and interface preference order.
3. Set *Source address* to *all*.
4. Under *Destination*, set *Application* to your required applications. In this example: *Facebook*, and *Youtube*.
5. Under *Outgoing Interfaces*, select *Best Quality*.
6. In *Interface preference*, add the interfaces in order of preference.

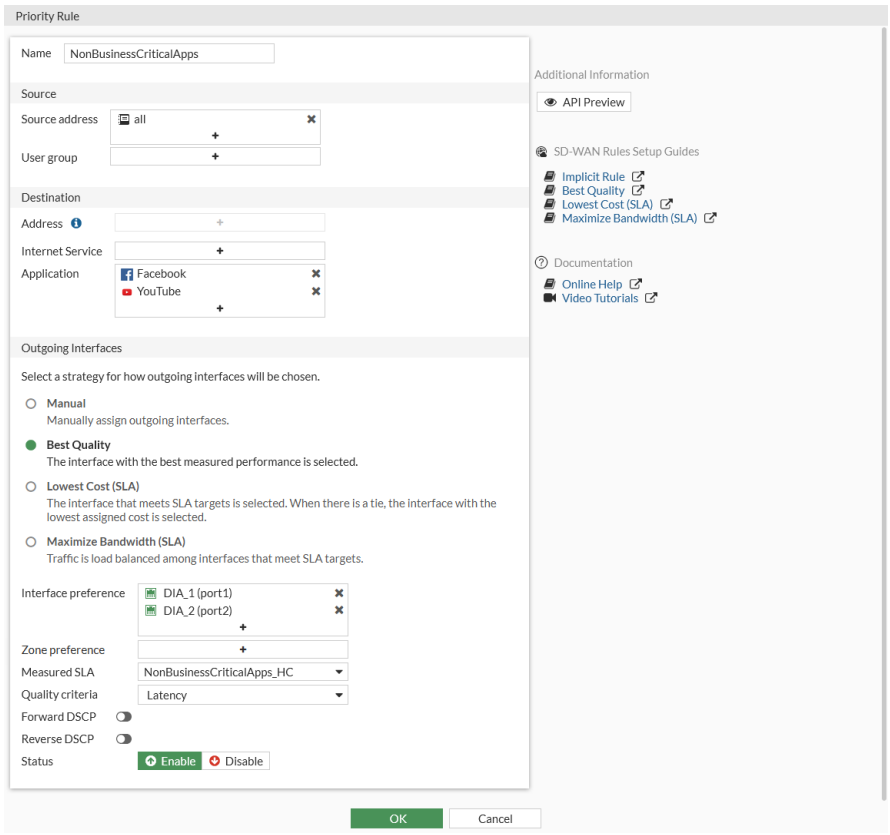
   By default, a more preferred link has an advantage of 10% over a less preferred link. For example, when latency is used, the preferred link's calculated latency = real latency / (1+10%).

   The preferred link advantage can be customized in the CLI when the mode is `priority` (*Best Quality*) or `auto`:

   ```
   config system sdwan
       config service
           edit <id>
               set link-cost-threshold <integer>
           next
       end
   end
   ```
7. Create and apply a new performance SLA profile:
   a. Click in the *Measured SLA* field.
   b. In the drop-down list, click *Create*. The *New Performace SLA* pane opens.
   c. Set *Name* to *NonBusinessCritical_HC*.

      This health check is used for non-business critical applications in your SD-WAN rule.
   d. Leave *Protocol* set to *Ping*, and add up to two servers, such as *youtube.com* and *facebook.com*.
   e. Set *Participants* to *Specify*, and add the DIA_1 and DIA_2 interfaces. In this example, MPLS is not used for non-business critical applications.
   f. Leave *SLA Target* disabled.
   g. Click *OK*.
   h. Select the new performance SLA from the list to set it as the *Measured SLA*.
8. Set *Quality criteria* as required. In this example, *Latency* is selected.

   For bandwidth related criteria, such as *Downstream*, *Upstream*, and *Bandwidth* (bi-directional), the selection is based on available bandwidth. An estimated bandwidth should be configured on the interface to provide a baseline, maximum available bandwidth.



9. Click *OK* to create the SD-WAN rule.

To configure the SD-WAN members, static route, and firewall policy in the CLI:

1. Configure the interfaces:

```
config system interface
    edit "port1"
        set ip <class_ip&net_netmask>
        set alias "DIA_1"
        set role wan
    next
    edit "port2"
        set ip <class_ip&net_netmask>
        set alias "DIA_2"
        set role wan
    next
    edit "port3"
        set ip <class_ip&net_netmask>
        set alias "MPLS"
        set role wan
    next
end
```

2. Configure the SD-WAN members:

```
config system sdwan
    set status enable
    config members
        edit 1
            set interface "port1"
            set gateway 172.16.20.2
        next
        edit 2
            set interface "port2"
            set gateway 172.17.80.2
        next
        edit 3
            set interface "port3"
            set gateway 10.100.20.2
            set cost 20
        next
    end
end
```

If no SD-WAN zone is specified, members are added to the default *virtual-wan-link* zone.

3. Configure a static route. See Routing on page 1043 for details.
4. Create a firewall policy to allow traffic out on SD-WAN, with an *Application Control* profile configured. See Security on page 1044 for details.

To configure the SD-WAN rule and performance SLA checks for business critical application in the CLI:

1. Configure the *BusinessCriticalApps_HC* health-check:

```
config system sdwan
    config health-check
        edit "BusinessCriticalApps_HC"
            set server "office.com" "google.com"
            set members 1 2 3
            config sla
                edit 1
                    set link-cost-factor packet-loss
                    set packetloss-threshold 1
                next
            end
        next
    end
end
```

2. Configure the *BusinessCriticalApps* service to use *Lowest Cost (SLA)*:

```
config system sdwan
    config service
        edit 1
            set name "BusinessCriticalApps"
```

```
            set mode sla
            set src "all"
            set internet-service enable
            set internet-service-app-ctrl 17459 16541 33182 16177 34640
            config sla
                edit "BusinessCriticalApps_HC"
                    set id 1
                next
            end
            set priority-members 1 2 3
        next
    end
end
```

To configure the SD-WAN rule and performance SLA checks for non-business critical application in the CLI:

1. Configure the *nonBusinessCriticalApps_HC* health-check:

```
config system sdwan
    config health-check
        edit "NonBusinessCriticalApps_HC"
            set server "youtube.com" "facebook.com"
            set members 1 2
        next
    end
end
```

2. Configure the *NonBusinessCriticalApps* service to use *Lowest Cost (SLA)*:

```
config system sdwan
    config service
        edit 4
            set name "NonBusinessCriticalApps"
            set mode priority
            set src "all"
            set internet-service enable
            set internet-service-app-ctrl 15832 31077
            set health-check "NonBusinessCriticalApps_HC"
            set priority-members 1 2
        next
    end
end
```

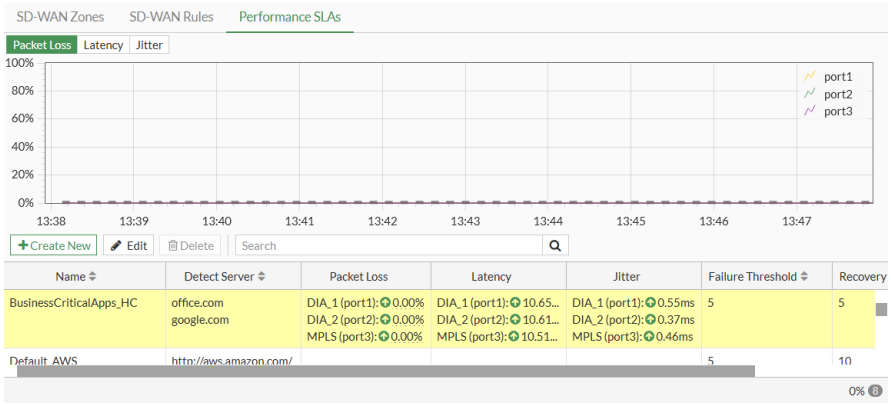
Verification

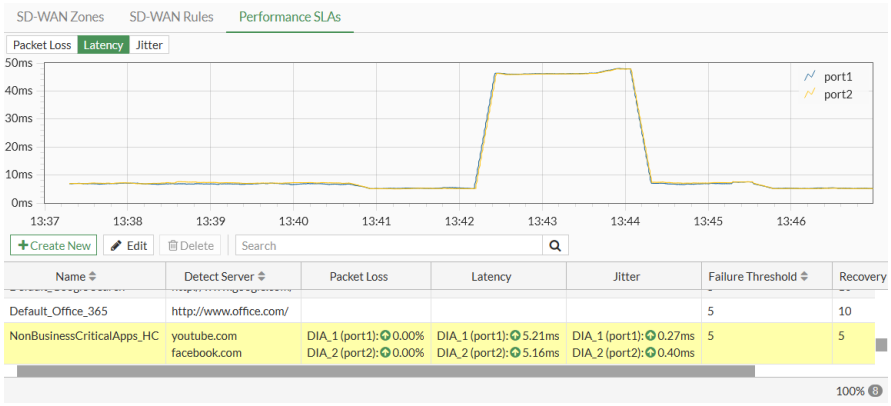
Check the following GUI pages, and run the following CLI commands to confirm that your traffic is being steered by the SD-WAN rules.

Health checks

To verify the status of each of the health checks in the GUI:

1. Go to *Network* > *SD-WAN*, select the *Performance SLAs* tab, and select each of the health checks from the list.





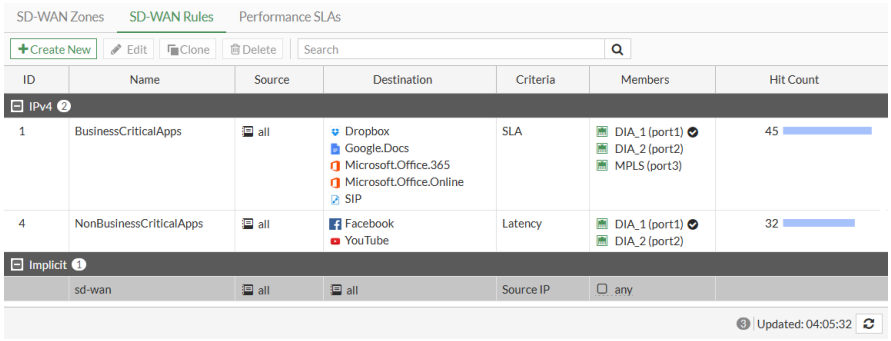
To verify the status of each of the health checks in the CLI:

```
# diagnose sys sdwan health-check
Health Check(BusinessCritical_HC):
Seq(1 port1): state(alive), packet-loss(0.000%) latency(12.884), jitter(0.919) sla_map=0x1
Seq(2 port2): state(alive), packet-loss(0.000%) latency(13.018), jitter(0.723) sla_map=0x1
Seq(3 port3): state(alive), packet-loss(0.000%) latency(13.018), jitter(0.923) sla_map=0x1
Health Check(NonBusinessCritical_HC):
Seq(1 port1): state(alive), packet-loss(0.000%) latency(6.888), jitter(0.953) sla_map=0x0
Seq(2 port2): state(alive), packet-loss(0.000%) latency(6.805), jitter(0.830) sla_map=0x0
```

Rule members and hit count

To verify the active members and hit count of the SD-WAN rule in the GUI:

1. Go to *Network* > *SD-WAN* and select the *SD-WAN Rules* tab.



   The interface that is currently selected by the rule has a checkmark next to its name in the *Members* column. Hover the cursor over the checkmark to open a tooltip that gives the reason why that member is selected. If multiple members are selected, only the highest ranked member is highlighted (unless the mode is *Maximize Bandwidth (SLA)*).

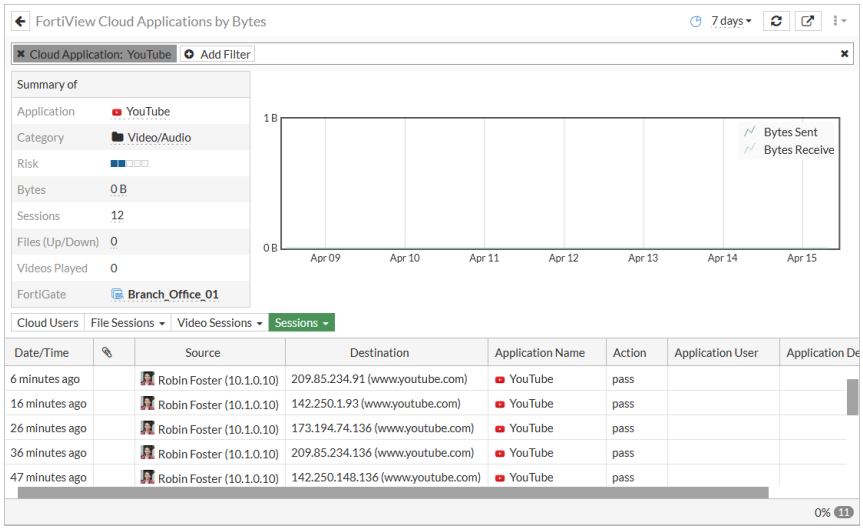
To verify the active members and hit count of the SD-WAN rule in the CLI:

```
# diagnose sys sdwan service4

Service(3): Address Mode(IPV4) flags=0x0
  Gen(13), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Members:
    1: Seq_num(1 port1), alive, sla(0x1), cfg_order(0), cost(0), selected
    2: Seq_num(2 port2), alive, sla(0x1), cfg_order(1), cost(0), selected
    3: Seq_num(3 port3), alive, sla(0x1), cfg_order(2), cost(20), selected
  Internet Service: Dropbox(4294836727,0,0,0 17459) Google.Docs(4294836992,0,0,0 16541)
Microsoft.Office.365(4294837472,0,0,0 33182) Microsoft.Office.Online(4294837475,0,0,0 16177) SIP
(4294837918,0,0,0 34640)
  Src address:
       0.0.0.0-255.255.255.255

Service(4): Address Mode(IPV4) flags=0x0
  Gen(211), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(priority), link-cost-factor(latency), link-
cost-threshold(10), health-check(NonBusinessCritical_HC)
  Members:
    1: Seq_num(1 port1), alive, latency: 5.712, selected
    2: Seq_num(2 port2), alive, latency: 5.511, selected
  Internet Service: Facebook(4294836806,0,0,0 15832) YouTube(4294838537,0,0,0 31077)
  Src address:
       0.0.0.0-255.255.255.255
```

Applications and sessions

To verify sessions in FortiView:

1. Go to a dashboard and add the *FortiView Cloud Applications* widget sorted by bytes. See Cloud application view on page 206 for details.
2. Drill down on an application, such as *YouTube*, then select the *Sessions* tab.



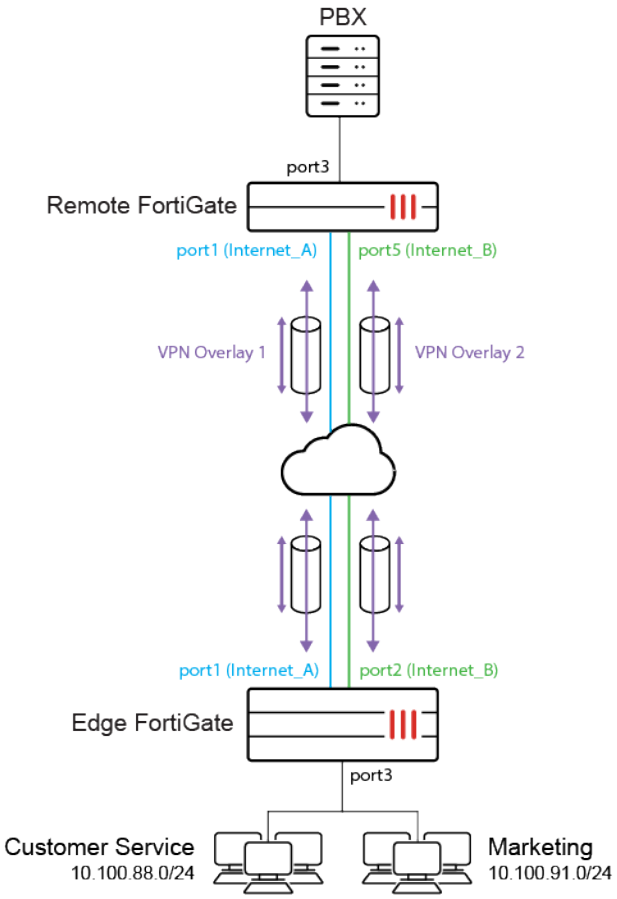
To verify applications identified by Application Control in SD-WAN:

```
# diagnose sys sdwan internet-service-app-ctrl-list

Steam(16518 4294838108): 23.6.148.10 6 443 Thu Apr 15 08:51:54 2021
Netflix(18155 4294837589): 54.160.93.182 6 443 Thu Apr 15 09:13:25 2021
Netflix(18155 4294837589): 54.237.226.164 6 443 Thu Apr 15 10:04:37 2021
Minecraft(27922 4294837491): 65.8.232.41 6 443 Thu Apr 15 09:12:19 2021
Minecraft(27922 4294837491): 65.8.232.46 6 443 Thu Apr 15 09:02:07 2021
Minecraft(27922 4294837491): 99.84.244.51 6 443 Thu Apr 15 10:23:57 2021
Minecraft(27922 4294837491): 99.84.244.63 6 443 Thu Apr 15 10:03:30 2021
YouTube(31077 4294838537): 74.125.69.93 6 443 Thu Apr 15 08:52:59 2021
YouTube(31077 4294838537): 108.177.112.136 6 443 Thu Apr 15 09:33:53 2021
YouTube(31077 4294838537): 142.250.1.93 6 443 Thu Apr 15 10:35:13 2021
...
```

DSCP tag-based traffic steering in SD-WAN

Differentiated Services Code Point (DSCP) tags can be used to categorize traffic for quality of service (QoS). SD-WAN traffic steering on an edge device can be provided based on the DSCP tags.

This section provides an example of using DSCP tag-based traffic steering using secure SD-WAN. Traffic from the customer service and marketing departments at a headquarters are marked with separate DSCP tags by the core switch and passed to the edge FortiGate. The edge FortiGate reads the tags, then steers traffic to the preferred interfaces based on the defined SD-WAN rules.



VoIP and social media traffic are steered. VoIP traffic from the customer service department is more important than social media traffic. The edge FortiGate identifies the tagged traffic based on SD-WAN rules then steers the traffic:

- VoIP traffic is marked with DSCP tag 011100 and steered to the VPN overlay with the lowest jitter, to provide the best quality voice communication with the remote PBX server.
- Social media traffic is marked with the DSCP tag 001100 and steered to the internet connection with the lowest cost.

The following is assumed to be already configured:

- Two IPsec tunnels (IPsec VPN on page 2760):
  - Branch-HQ-A on Internet_A (port 1)
  - Branch-HQ-B on Internet_B (port 5)
- Four SD-WAN members in two zones (Underlay on page 1042):
  - Overlay zone includes members Branch-HQ-A and Branch-HQ-B
  - virtual-wan-link zone includes members Internet_A and Internet_B

    Internet_A has a cost of 0 and Internet_B has a cost of 10. When using the lowest cost strategy, Internet_A will be preferred. Both members are participants in the Default_DNS performance SLA.
- A static route that points to the SD-WAN interface (Routing on page 1043).
- Two firewall policies:

Name	SD-WAN-OUT	Overlay-OUT

From	port3	port3
To	virtual-wan-link	Overlay
Source	all	all
Destination	all	all
Schedule	always	always
Service	all	all
Action	Accept	Accept
NAT	enabled	enabled

After the topology is configured, you can proceed with the configuration of the edge FortiGate:

- Configuring SD-WAN rules on page 1160
- Results on page 1162

Configuring SD-WAN rules

Configure SD-WAN rules to govern the steering of DSCP tag-based traffic to the appropriate interfaces. Traffic is steered based on the criteria that are configured in the SD-WAN rules.

In this example, three SD-WAN rules are configured to govern DSCP tagged traffic:

- *VoIP-Steer* for VoIP traffic.
- *Facebook-DSCP-steer* for Social media traffic.
- *All-traffic* for all of the Other web traffic.

After configuring the rules, go to *Network* > *SD-WAN* and select the *SD-WAN Rules* tab to check the rules.

VoIP traffic

VoIP traffic is steered to the *Overlay* zone.

DSCP values are usually 6-bit binary numbers that are padded with zeros at the end. VoIP traffic with DSCP tag 011100 will become 01110000. This 8-bit binary number is represented in its hexadecimal form, 0x70, as the type of service bit pattern (`tos`) value. The type of service evaluated bits (`tos-mask`) hexadecimal value of 0xf0 (11110000 in binary) is used to check the four most significant bits in the `tos` value. The four most significant bits of the `tos` (0111) are used to match the first four bits of the DSCP tag. Only the non-zero bit positions in the `tos-mask` are used for comparison; the zero bit positions are ignored.

The *Best quality* (`priority` mode) strategy is used to select the preferred interface, with the *Quality criteria* (`link-cost-members`) set to *Jitter*. The interface with the lowest amount of jitter is selected. For more information about configuring SD-WAN rules with the *Best Quality* strategy, see Best quality strategy on page 1123.

To configure the rule for DSCP tagged VoIP traffic using the CLI:

```
config sys sdwan
    config service
```

```
        edit 5
            set name "VoIP-Steer"
            set mode priority
            set tos 0x70
            set tos-mask 0xf0
            set dst "all"
            set health-check "Default_DNS"
            set link-cost-factor jitter
            set priority-members 4 3
        next
    end
end
```

Social media traffic

Social media traffic is steered to the *virtual-wan-link* zone.

DSCP values are usually 6-bit binary numbers that are padded with zeros at the end. Social media traffic traffic with DSCP tag 001100 will become 00110000. This 8-bit binary number is represented in its hexadecimal form, 0x30, as the `tos` value. The `tos-mask` hexadecimal value of 0xf0 (11110000 in binary) is used to check the four most significant bits in the tos value. The four most significant bits of the `tos` (0011) are used to match the first four bits of the DSCP tag. Only the non-zero bit positions in the `tos-mask` are used for comparison; the zero bit positions are ignored.

The *Manual* (`manual` mode) strategy is used to select the preferred interface. Internet_B (port5, priority member 2) is set as the preferred interface to steer all social media traffic to. For more information about configuring SD-WAN rules with the manual strategy, see Manual strategy on page 1119.

To configure SD-WAN rule for DSCP tagged social media traffic using the CLI:

```
config system sdwan
    config service
        edit 3
            set name "Facebook-DSCP-steer"
            set mode manual
            set tos 0x30
            set tos-mask 0xf0
            set dst "all"
            set priority-members 2 1
        next
    end
end
```

Other web traffic

Other web traffic is steered to the *virtual-wan-link* zone.

The *Lowest Cost (SLA)* strategy (`sla` mode) is used to select the preferred interface. The interface that meets the defined SLA targets (*Default_DNS* in this case) is selected. If there is a tie, the interface with the lowest cost is selected, Internet_A (port1) in this case.

For more information about configuring SD-WAN rules with the *Lowest Cost (SLA)* strategy, see Lowest cost (SLA) strategy on page 1126.

To configure SD-WAN rule for all other web traffic using the CLI:

```
config system sdwan
    config service
        edit 2
            set name "All-traffic"
            set mode sla
            set dst "all"
            config sla
                edit "Default_DNS"
                    set id 1
                next
            end
            set priority-members 1 2
        next
    end
end
```

Results

These sections show the function of SD-WAN with respect to DSCP tagged traffic steering, and can help confirm that it is running as expected:

- Verifying the DSCP tagged traffic on FortiGate on page 1162
- Verifying the service rules on page 1163
- Verifying traffic steering on the SD-WAN rules on page 1164
- Verifying that steered traffic is leaving from the expected interface on page 1164

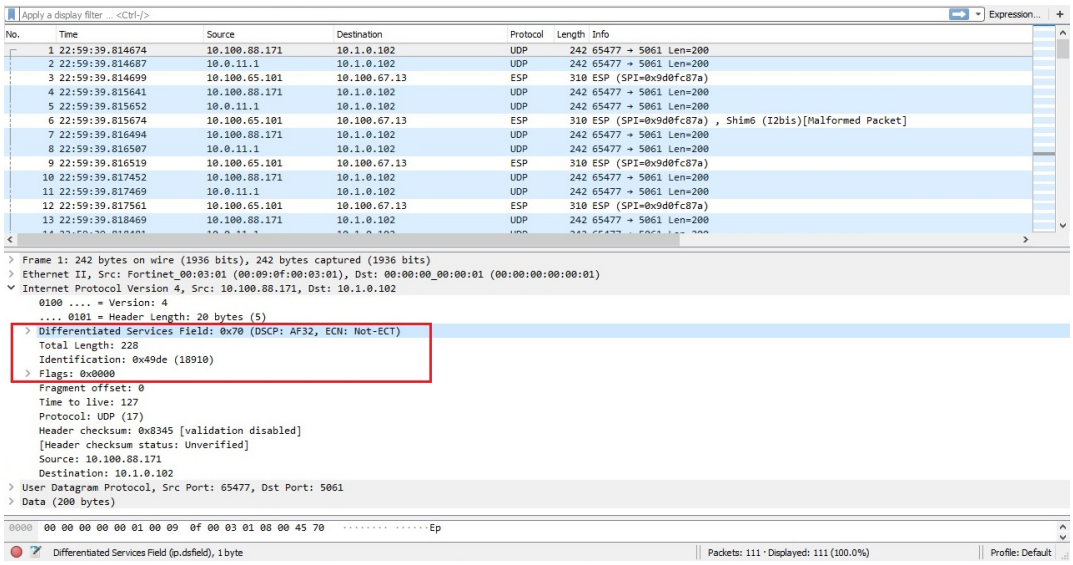
Verifying the DSCP tagged traffic on FortiGate

Packet sniffing is used to verify the incoming DSCP tagged traffic. See Using the FortiOS built-in packet sniffer for more information.

Wireshark is used to verify that VoIP traffic is tagged with the expected DSCP tag, 0x70 or 0x30.

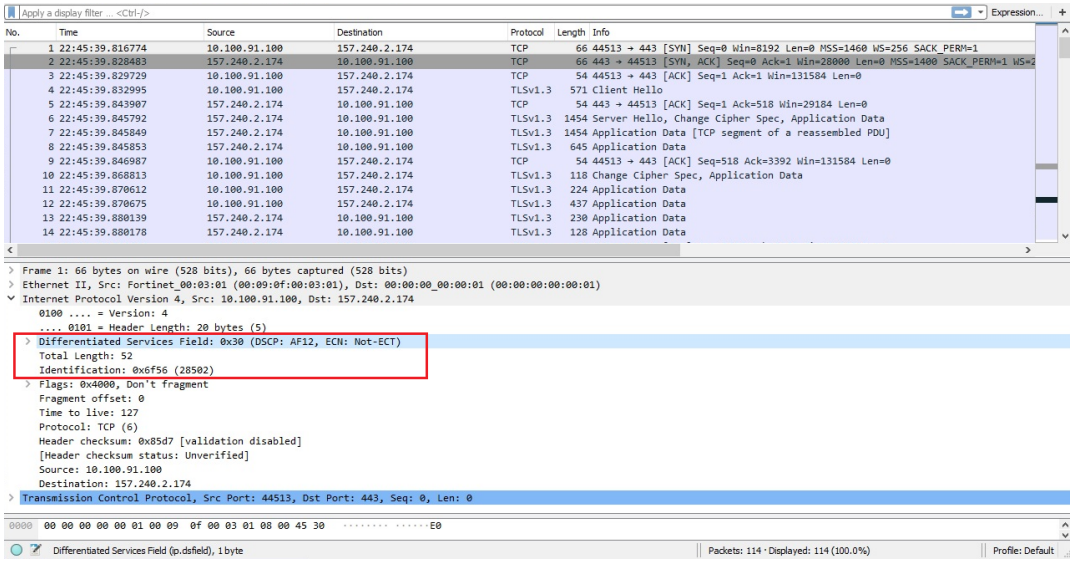
VoIP traffic marked with DSCP tag 0x70:

```
# diagnose sniffer packet any '(ip and ip[1] & 0xfc == 0x70)' 6 0 l
```



Web traffic marked with DSCP tag 0x30:

```
# diagnose sniffer packet any '(ip and ip[1] & 0xfc == 0x30)' 6 0 l
```



Verifying the service rules

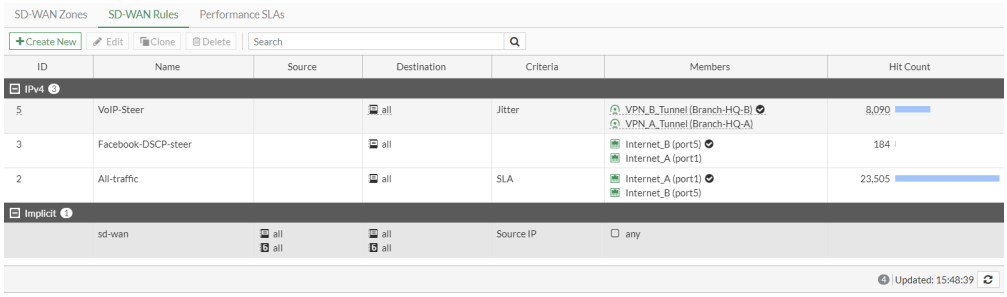
To check that the expected DSCP tags and corresponding interfaces are used by the SD-WAN rules to steer traffic:

```
# diagnose sys sdwan service4

Service(5): Address Mode(IPV4) flags=0x0
    Gen(1), TOS(<b>0x70</b>/0xf0), Protocol(0: 1-&gt;65535), Mode(manual)
    Members:
```

```
        1: Seq_num(4 <b>Branch-HQ-B</b>), alive, selected
    Dst address:
            0.0.0.0-255.255.255.255

Service(3): Address Mode(IPV4) flags=0x0
    Gen(1), TOS(<b>0x30</b>/0xf0), Protocol(0: 1-&gt;65535), Mode(manual)
    Members:
        1: Seq_num(2 <b>port5</b>), alive, selected
    Dst address:
            0.0.0.0-255.255.255.255

Service(2): Address Mode(IPV4) flags=0x0
    Gen(1), TOS(<b>0x0</b>/0x0), Protocol(0: 1-&gt;65535), Mode(sla), sla-compare-order
    Members:
        1: Seq_num(1 <b>port1</b>), alive, sla(0x1), cfg_order(0), cost(0), selected
        2: Seq_num(2 <b>port5</b>), alive, sla(0x1), cfg_order(1), cost(10), selected
    Dst address:
            0.0.0.0-255.255.255.255
```

Verifying traffic steering on the SD-WAN rules

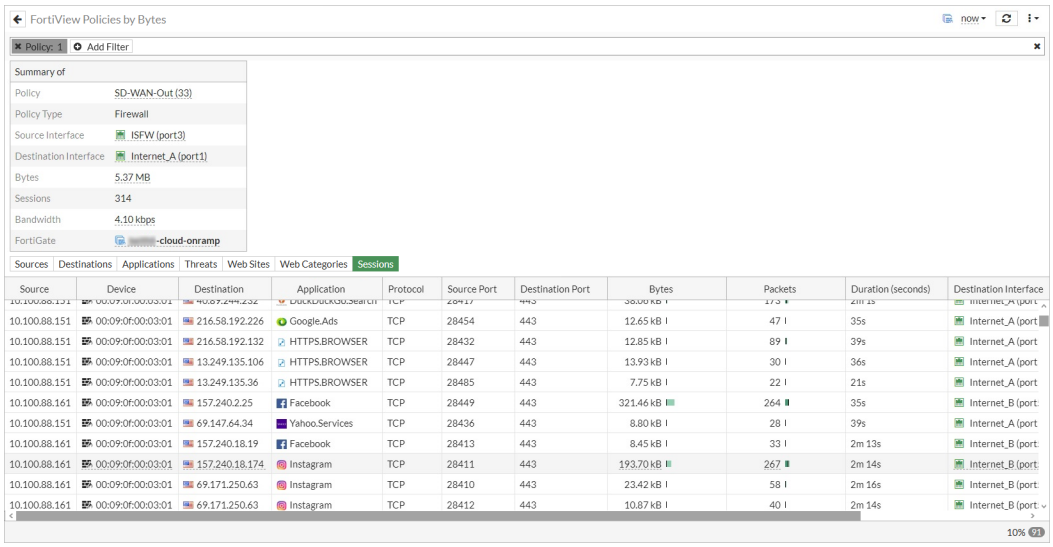
Go to *Network* > *SD-WAN* and select the *SD-WAN Rules* tab to check the *Hit Count* on the SD-WAN interfaces.



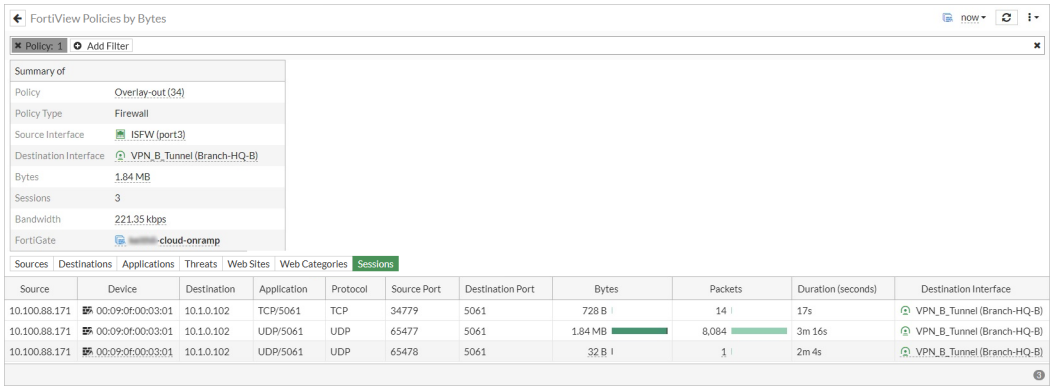
Verifying that steered traffic is leaving from the expected interface

To confirm that web traffic (port 443) flows through the correct underlay interface members, and VoIP traffic flows through the correct overlay interface members, go to *Dashboard* > *FortiView Policies* and double click on the policy name.

Web traffic is expected to leave on *Interface_A (port1)* or *Interface_B (port5)*:



VoIP traffic is expected to leave on the preferred *VPN_B_Tunnel (Branch-HQ-B)* interface:



ECMP support for the longest match in SD-WAN rule matching

The longest match SD-WAN rule can match ECMP best routes. The rule will select the egress ports on ECMP specific routes, and not the less specific routes, to transport traffic.

The service mode determines which egress port on the ECMP specific routes is selected to forward traffic:

- Manual (`manual`): The first configured alive port is selected.
- Best Quality (`priority`): The best quality port is selected.
- Lowest Cost (`sla`): The first configured or lower cost port in SLA is selected.

Example

By default, SD-WAN selects the outgoing interface from all of the links that have valid routes to the destination. In some cases, it is required that only the links that have the best (or longest match) routes (single or ECMP) to

the destination are considered.

In this example, four SD-WAN members in two zones are configured. The remote PC (PC_2 - 10.1.100.22) is accessible on port15 and port16, even though there are valid routes for all of the SD-WAN members. A single SD-WAN service rule is configured that allows traffic to balanced between all four of the members, but only chooses between port15 and port16 for the specific 10.1.100.22 address.

A performance SLA health check is configured to monitor 10.1.100.2. An SD-WAN service rule in Lowest Cost (SLA) mode is configured to select the best interface to steer the traffic. In the rule, the method of selecting a member if more than one meets the SLA (`tie-break`) is configured to select members that meet the SLA and match the longest prefix in the routing table (`fib-best-match`). If there are multiple ECMP routes with the same destination, the FortiGate will take the longest (or best) match in the routing table, and choose from those interface members.

To configure the SD-WAN:

```
config system sdwan
    config zone
        edit "virtual-wan-link"
        next
        edit "z1"
        next
    end
    config members
        edit 1
            set interface "port1"
            set gateway 172.16.200.2
        next
        edit 2
            set interface "dmz"
            set gateway 172.16.208.2
        next
        edit 3
            set interface "port15"
            set zone "z1"
            set gateway 172.16.209.2
        next
        edit 4
            set interface "port16"
            set zone "z1"
            set gateway 172.16.210.2
        next
    end
    config health-check
        edit "1"
            set server "10.1.100.2"
```

```
            set members 0
            config sla
                edit 1
                next
            end
        next
    end
    config service
        edit 1
            set name "1"
            set mode sla
            set dst "all"
            set src "172.16.205.0"
            config sla
                edit "1"
                    set id 1
                next
            end
            set priority-members 1 2 3 4
            set tie-break fib-best-match
        next
    end
end
```

To check the results:

1. The debug shows the SD-WAN service rule. All of the members meet SLA, and because no specific costs are attached to the members, the egress interface is selected based on the interface priority order that is configured in the rule:

```
FGT_A (root) # diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
  Gen(4), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Members(4):
    1: Seq_num(1 port1), alive, sla(0x1), gid(0), cfg_order(0), cost(0), selected
    2: Seq_num(2 dmz), alive, sla(0x1), gid(0), cfg_order(1), cost(0), selected
    3: Seq_num(3 port15), alive, sla(0x1), gid(0), cfg_order(2), cost(0), selected
    4: Seq_num(4 port16), alive, sla(0x1), gid(0), cfg_order(3), cost(0), selected
  Src address(1):
        172.16.205.0-172.16.205.255
  Dst address(1):
        0.0.0.0-255.255.255.255
```

2. The routing table shows that there are ECMP default routes on all of the members, and ECMP specific (or best) routes only on port15 and port16:

```
FGT_A (root) # get router info routing-table static
Routing table for VRF=0
S*      0.0.0.0/0 [1/0] via 172.16.200.2, port1
                        [1/0] via 172.16.208.2, dmz
                        [1/0] via 172.16.209.2, port15
```

```
                          [1/0] via 172.16.210.2, port16
S       10.1.100.22/32 [10/0] via 172.16.209.2, port15
                               [10/0] via 172.16.210.2, port16
```

Because `tie-break` is set to `fib-best-match`, the first configured member from port15 and port16 is selected to forward traffic to PC_2. For all other traffic, the first configured member from all four of the interfaces is selected to forward traffic.

3. On PC-1, generate traffic to PC-2:

```
ping 10.1.100.22
```

4. On FGT_A, sniff for traffic sent to PC_2:

```
# diagnose sniffer packet any 'host 10.1.100.22' 4
interfaces=[any]
filters=[host 10.1.100.22]
2.831299 port5 in 172.16.205.11 -> 10.1.100.22: icmp: echo request
2.831400 port15 out 172.16.205.11 -> 10.1.100.22: icmp: echo request
```

Traffic is leaving on port15, the first configured member from port15 and port16.

Override quality comparisons in SD-WAN longest match rule matching

In SD-WAN rules, the longest match routes will override the quality comparisons when all of the specific routes are out of SLA.

With this feature in an SD-WAN rule:

- Lowest Cost (`sla`): Even though all of the egress ports on specific routes (longest matched routes) are out of SLA, the SD-WAN rule still selects the first configured or lower-cost port from the egress ports to forward traffic.
- Best Quality (`priority`): Even though the egress ports on specific routes (longest matched routes) have worse quality that all other ports on less specific routes, the SD-WAN rule still selects the best quality port from the ports on specific routes to forward traffic.

This features avoids a situation where, if the members on specific routes (longest matched routes) are out of SLA or have worse quality, the traffic might be forwarded to the wrong members in SLA (higher quality) on the default or aggregate routes.

Example

In this example, four SD-WAN members in two zones are configured. The remote PC (PC_2 - 10.1.100.22) is accessible on port15 and port16, even though there are valid routes for all of the SD-WAN members. A single SD-WAN service rule is configured that allows traffic to balanced between all four of the members, but only chooses between port15 and port16 for the specific 10.1.100.22 address. If neither port15 nor port16 meet the SLAs, traffic will be forwarded on one of these interfaces, instead of on port1 or dmz.

A performance SLA health check is configured to monitor 10.1.100.2. An SD-WAN service rule in Lowest Cost (SLA) mode is configured to select the best interface to steer the traffic. In the rule, the method of selecting a member if more than one meets the SLA (`tie-break`) is configured to select members that meet the SLA and match the longest prefix in the routing table (`fib-best-match`). If there are multiple ECMP routes with the same destination, the FortiGate will take the longest (or best) match in the routing table, and choose from those interface members.

To configure the SD-WAN:

```
config system sdwan
    config zone
        edit "virtual-wan-link"
        next
        edit "z1"
        next
    end
    config members
        edit 1
            set interface "port1"
            set gateway 172.16.200.2
        next
        edit 2
            set interface "dmz"
            set gateway 172.16.208.2
        next
        edit 3
            set interface "port15"
            set zone "z1"
            set gateway 172.16.209.2
        next
        edit 4
            set interface "port16"
            set zone "z1"
            set gateway 172.16.210.2
        next
    end
    config health-check
        edit "1"
            set server "10.1.100.2"
            set members 0
            config sla
                edit 1
                next
            end
        next
    end
```

```
    config service
        edit 1
            set name "1"
            set mode sla
            set dst "all"
            set src "172.16.205.0"
            config sla
                edit "1"
                    set id 1
                next
            end
            set priority-members 1 2 3 4
            set tie-break fib-best-match
        next
    end
end
```

To check the results:

1. The debug shows the SD-WAN service rule. Both port15 and port16 are up, but out of SLA:

```
FGT_A (root) # diagnose sys sdwan service4
Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
  Gen(3), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Members(4):
    1: Seq_num(1 port1), alive, sla(0x1), gid(0), cfg_order(0), cost(0), selected
    2: Seq_num(2 dmz), alive, sla(0x1), gid(0), cfg_order(1), cost(0), selected
    3: Seq_num(3 port15), alive, sla(0x0), gid(0), cfg_order(2), cost(0), selected
    4: Seq_num(4 port16), alive, sla(0x0), gid(0), cfg_order(3), cost(0), selected
  Src address(1):
        172.16.205.0-172.16.205.255

  Dst address(1):
        0.0.0.0-255.255.255.255
```

2. The routing table shows that there are ECMP default routes on all of the members, and ECMP specific (or best) routes only on port15 and port16:

```
FGT_A (root) # get router info routing-table static
Routing table for VRF=0
S*      0.0.0.0/0 [1/0] via 172.16.200.2, port1
                        [1/0] via 172.16.208.2, dmz
                        [1/0] via 172.16.209.2, port15
                        [1/0] via 172.16.210.2, port16
S       10.1.100.22/32 [10/0] via 172.16.209.2, port15
                                  [10/0] via 172.16.210.2, port16
```

   Because `tie-break` is set to `fib-best-match`, even though both port15 and port16 are out of SLA, the first configured member of the two (port15) is selected to forward traffic to PC_2. For all other traffic, the first configured member from all of the interfaces that are in SLA is selected to forward traffic (port1).

3. On PC-1, generate traffic to PC-2:

```
ping 10.1.100.22
```

4. On FGT_A, sniff for traffic sent to PC_2:

```
# diagnose sniffer packet any 'host 10.1.100.22' 4
interfaces=[any]
filters=[host 10.1.100.22]
2.831299 port5 in 172.16.205.11 -> 10.1.100.22: icmp: echo request
2.831400 port15 out 172.16.205.11 -> 10.1.100.22: icmp: echo request
```

Traffic is leaving on port15, the first configured member from port15 and port16, even though both are out of SLA.

Internet service and application control steering

An application, application group, or application category can be selected as an SD-WAN service rule destination criterion for IPv4 and IPv6 address modes.

To configure from the CLI:

```
config system sdwan
    config service
        edit <id>
            set internet-service enable
            set internet-service-app-ctrl <app id> [app id]
            set internet-service-app-ctrl-group <app group> [app group]
            set internet-service-app-ctrl-category <category id> [category id]
        next
    end
end
```

To configure for IPv6 addressing mode from the CLI, enable `addr-mode ipv6`:

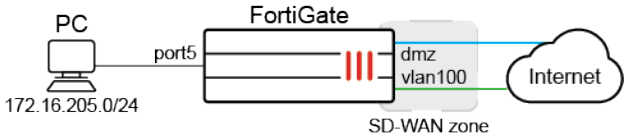
```
config system sdwan
    config service
        edit <id>
            set addr-mode ipv6
        next
    end
end
```

To view the detected application category details based on category ID, use `diagnose sys sdwan internet-service-app-ctrl-list cat-id <cat-id>`.

This topic includes a GUI and CLI Example for application category on page 1172 and a CLI Example for IPv6 on page 1176.

Example for application category

In this example, traffic steering is applied to traffic detected as video/audio (category ID 5) or email (category ID 21) and applies the lowest cost (SLA) strategy to this traffic. When costs are tied, the priority goes to member 1, dmz.



To configure application categories as an SD-WAN rule destination in the GUI:

1. Enable the feature visibility:
   a. Go to *System* > *Feature Visibility*.
   b. In the *Additional Features* section, enable *Application Detection Based SD-WAN*.
   c. Click *Apply*.

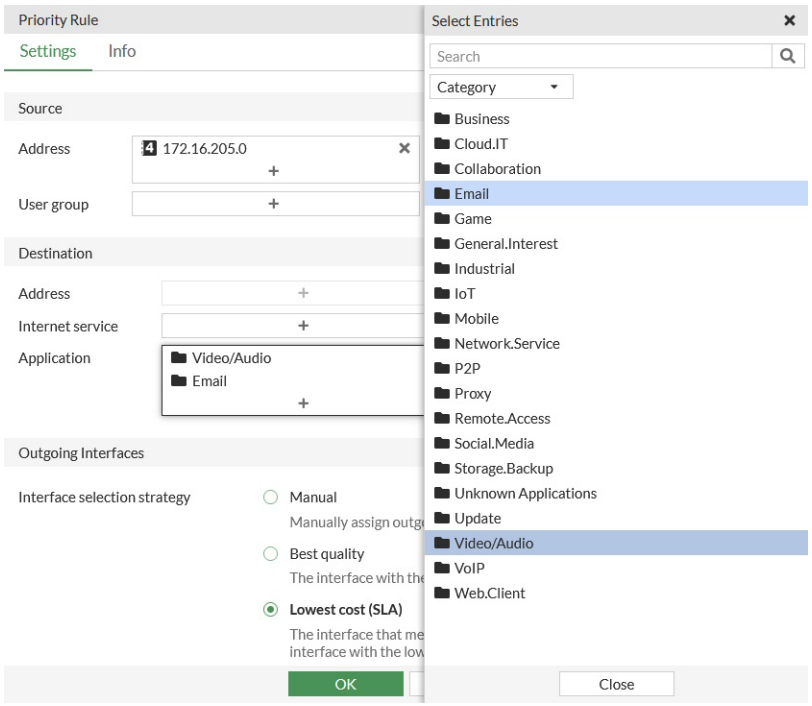
   To enable GUI visibility of application detection based SD-WAN in the CLI:

   ```
   config system global
       set gui-app-detection-sdwan enable
   end
   ```

2. Configure the SD-WAN members:
   a. Go to *Network* > *SD-WAN*, select the *SD-WAN Zones* tab, and click *Create New* > *SD-WAN Member*.
   b. Set the *Interface* to *dmz*, and set the *Gateway* to *172.16.208.2*.
   c. Click *OK*.
   d. Repeat these steps to create another member for the *vlan100* interface with gateway *172.16.206.2*.
3. Configure the performance SLA (health check):
   a. Go to *Network* > *SD-WAN*, and select the *Performance SLAs* tab, and click *Create New*.
   b. Configure the following settings:

Name	*1*
Protocol	*DNS*
Server	*8.8.8.8*
SLA Target	Enable

   c. Click *OK*.
4. Configure the SD-WAN rule to use the video/audio and email application categories:
   a. Go to *Network* > *SD-WAN*, select the *SD-WAN Rules* tab, and click *Create New*.
   b. In the *Destination* section, click the *+* in the *Application* field.
   c. Click *Category*, and select *Video/Audio* and *Email*.



d. Configure the other settings as needed.

e. Click *OK*.

5. Configure the firewall policy:

a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.

b. Configure the following settings:

Incoming Interface	*port5*
Outgoing Interface	*virtual-wan-link*
Source	*172.16.205.0*
Destination	*all*
Schedule	*always*
Service	*ALL*
Action	*ACCEPT*
Application Control	*g-default*
SSL Inspection	*certificate-inspection*

c. Click *OK*.

To configure application categories as an SD-WAN rule destination in the CLI:

1. Configure the SD-WAN settings:

```
config system sdwan
    set status enable
```

```
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "dmz"
            set gateway 172.16.208.2
        next
        edit 2
            set interface "vlan100"
            set gateway 172.16.206.2
        next
    end
    config health-check
        edit "1"
            set server "8.8.8.8"
            set protocol dns
            set members 0
            config sla
                edit 1
                next
            end
        next
    end
end
```

2. Configure the SD-WAN rule to use application categories 5 and 21:

```
config system sdwan
    config service
        edit 1
            set name "1"
            set mode sla
            set src "172.16.205.0"
            set internet-service enable
            set internet-service-app-ctrl-category 5 21
            config sla
                edit "1"
                    set id 1
                next
            end
            set priority-members 1 2
        next
    end
end
```

3. Configure the firewall policy:

```
config firewall policy
    edit 1
        set srcintf "port5"
```

```
        set dstintf "virtual-wan-link"
        set action accept
        set srcaddr 172.16.205.0
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "certificate-inspection"
        set application-list "g-default"
    next
end
```

To test the configuration:

1. Verify that the traffic is sent over dmz:

```
# diagnose firewall proute list
list route policy info(vf=root):
id=2133590017(0x7f2c0001) vwl_service=1(1) vwl_mbr_seq=1 2 dscp_tag=0xff 0xff flags=0x0
tos=0x00 tos_mask=0x00 protocol=0 sport=0-65535 iif=0 dport=1-65535 path(2) oif=5(dmz) oif=95
(vlan100)
source(1): 172.16.205.0-172.16.205.255
destination wildcard(1): 0.0.0.0/0.0.0.0
internet service(2): (null)(0,5,0,0,0) (null)(0,21,0,0,0)
hit_count=469 last_used=2021-12-15 15:06:05
```

2. View some videos and emails on the PC, then verify the detected application details for each category:

```
# diagnose sys sdwan internet-service-app-ctrl-list cat-id 5
List App Ctrl Database Entry(IPv4) in Kernel:

Max_App_Ctrl_Size=32768 Num_App_Ctrl_Entry=4

YouTube(31077 4294838537): IP=142.250.217.110 6 443
YouTube(31077 4294838537): IP= 173.194.152.89 6 443
YouTube(31077 4294838537): IP= 173.194.152.170 6 443
YouTube(31077 4294838537): IP= 209.52.146.205 6 443
```

```
# diagnose sys sdwan internet-service-app-ctrl-list cat-id 21
List App Ctrl Database Entry(IPv4) in Kernel:

Max_App_Ctrl_Size=32768 Num_App_Ctrl_Entry=1

Gmail(15817 4294836957): IP=172.217.14.197 6 443
```

3. Verify that the captured email traffic is sent over dmz:

```
# diagnose sniffer packet any 'host 172.217.14.197' 4
interfaces=[any]
filters=[host 172.217.14.197]
5.079814 dmz out 172.16.205.100.60592 -> 172.217.14.197.443: psh 2961561240 ack 2277134591
```

4. Edit the SD-WAN rule so that dmz has a higher cost and vlan100 is preferred.

5. Verify that the traffic is now sent over vlan100:

```
# diagnose firewall proute list
list route policy info(vf=root):
id=2134048769(0x7f330001) vwl_service=1(1) vwl_mbr_seq=2 1 dscp_tag=0xff 0xff flags=0x0
tos=0x00 tos_mask=0x00 protocol=0 sport=0-65535 iif=0 dport=1-65535 path(2) oif=95(vlan100)
oif=5(dmz)
source(1): 172.16.205.0-172.16.205.255
destination wildcard(1): 0.0.0.0/0.0.0.0
internet service(2): (null)(0,5,0,0,0) (null)(0,21,0,0,0)
hit_count=635 last_used=2021-12-15 15:55:43
```

```
# diagnose sniffer packet any 'host 172.217.14.197' 4
interfaces=[any]
filters=[host 172.217.14.197]
304.625168 vlan100 in 172.16.205.100.60592 -> 172.217.14.197.443: psh 2961572711 ack
2277139565
```

Example for IPv6

In this example, SD-WAN is configured to use an IPv6 service rule to steer traffic from FGT_A to FGT_B based on the following application control options:

- Application Telnet
- An application group for ping
- An application category that includes SSH

When the rule is matched, traffic is steered based on the lowest cost SLA strategy. In this example, vlan100 is the preferred interface, and traffic is routed to vlan100 on FGT_B.

To view the configuration:

1. View the SD-WAN configuration on FGT_A:

   SD-WAN has four members in the default virtual-wan-link zone, each with an IPv4 and IPv6 gateway. The SD-WAN service rule includes `internet-service-app-ctrl 16091` for the Telnet, `internet-service-app-ctrl-group "network-Ping"` for ping , and `internet-service-app-ctrl-category 15` for SSH applications.

```
(sdwan) # show
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "dmz"
            set gateway 172.16.208.2
            set gateway6 2000:172:16:208::2
```

```
        next
        edit 2
            set interface "IPSec-1"
        next
        edit 3
            set interface "agg1"
            set gateway 172.16.203.2
            set gateway6 2000:172:16:203::2
        next
        edit 4
            set interface "vlan100"
            set gateway 172.16.206.2
            set gateway6 2000:172:16:206::2
        next
    end
    config health-check
        edit "1"
            set addr-mode ipv6
            set server "2000::2:2:2:2"
            set members 0
            config sla
                edit 1
                next
            end
        next
    end
    config service
        edit 1
            set name "1"
            set addr-mode ipv6
            set mode sla
            set internet-service enable
            set internet-service-app-ctrl 16091
            set internet-service-app-ctrl-group "network-Ping"
            set internet-service-app-ctrl-category 15
            config sla
                edit "1"
                    set id 1
                next
            end
            set priority-members 4 1 2 3
        next
    end
end
```

2. View the default route for FGT_A:

```
config router static
    edit 5
        set distance 1
        set sdwan-zone "virtual-wan-link"
```

```
    next
end
```

3. View the firewall policy for FGT_A:

   The `utm-status` option is enabled to learn application 3T (3 tuple) information, and the default application profile of `g-default` is selected.

```
config firewall policy
    edit 1
        set uuid f09bddc4-def3-51ed-8517-0d8b6bc18f35
        set srcintf "any"
        set dstintf "any"
        set action accept
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "certificate-inspection"
        set application-list "g-default"
    next
end
```

To verify the configuration:

1. On FGT_A, check the routing table:

   The routing table has ECMP applied to default gateways for each SD-WAN member.

```
# get router info routing-table  static
Routing table for VRF=0
S*      0.0.0.0/0 [1/0] via 172.16.203.2, agg1, [1/0]
                        [1/0] via 172.16.206.2, vlan100, [1/0]
                        [1/0] via 172.16.208.2, dmz, [1/0]
                        [1/0] via IPSec-1 tunnel 172.16.209.2, [1/0]
```

2. Check the SD-WAN service:

   Based on the service rule, member 4 named vlan100 is preferred. Traffic must also match the highlighted internet services.

```
# diagnose system sdwan service

Service(1): Address Mode(IPV6) flags=0x4200 use-shortcut-sla use-shortcut
 Tie break: cfg
  Gen(2), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Members(4):
    1: Seq_num(4 vlan100), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), selected
2: Seq_num(1 dmz), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected
    3: Seq_num(2 IPSec-1), alive, sla(0x1), gid(0), cfg_order(2), local cost(0), selected
    4: Seq_num(3 agg1), alive, sla(0x1), gid(0), cfg_order(3), local cost(0), selected
  Internet Service(3): Telnet(4294837974,0,0,0,0 16091) IPv6.ICMP(4294837087,0,0,0,0 16321)
Network.Service(0,15,0,0,0)
```

3. Initiate traffic for ping, Telnet, and SSH to FGT_B, then FGT_A will learn 3T information for these applications, and use the SD-WAN rule to route traffic for the applications to the preferred interface of vlan100.
   - Following is the sniffer traffic for ping application. The ping traffic flows out of DMZ before 3T information is recognized, then out from vlan100 after T3 traffic is recognized:

```
# diagnose sniffer packet any 'host 2000::2:0:0:4' 4
interfaces=[any]
filters=[host 2000::2:0:0:4]
16.952138 port5 in 2000:172:16:205::100 -> 2000::2:0:0:4: icmp6: echo request seq 1
[flowlabel 0x5080d]
16.954571 dmz out 2000:172:16:205::100 -> 2000::2:0:0:4: icmp6: echo request seq 1
[flowlabel 0x5080d]
16.954920 dmz in 2000::2:0:0:4 -> 2000:172:16:205::100: icmp6: echo reply seq 1
16.955086 port5 out 2000::2:0:0:4 -> 2000:172:16:205::100: icmp6: echo reply seq 1
17.953277 port5 in 2000:172:16:205::100 -> 2000::2:0:0:4: icmp6: echo request seq 2
[flowlabel 0x5080d]
17.953455 dmz out 2000:172:16:205::100 -> 2000::2:0:0:4: icmp6: echo request seq 2
[flowlabel 0x5080d]
17.953622 dmz in 2000::2:0:0:4 -> 2000:172:16:205::100: icmp6: echo reply seq 2
17.953722 port5 out 2000::2:0:0:4 -> 2000:172:16:205::100: icmp6: echo reply seq 2
18.959823 port5 in 2000:172:16:205::100 -> 2000::2:0:0:4: icmp6: echo request seq 3
[flowlabel 0x5080d]
18.960005 vlan100 out 2000:172:16:205::100 -> 2000::2:0:0:4: icmp6: echo request seq 3
[flowlabel 0x5080d]
18.960015 agg1 out 2000:172:16:205::100 -> 2000::2:0:0:4: icmp6: echo request seq 3
[flowlabel 0x5080d]
18.960024 port4 out 2000:172:16:205::100 -> 2000::2:0:0:4: icmp6: echo request seq 3
[flowlabel 0x5080d]
18.960295 vlan100 in 2000::2:0:0:4 -> 2000:172:16:205::100: icmp6: echo reply seq 3
18.960449 port5 out 2000::2:0:0:4 -> 2000:172:16:205::100: icmp6: echo reply seq 3
19.983802 port5 in 2000:172:16:205::100 -> 2000::2:0:0:4: icmp6: echo request seq 4
[flowlabel 0x5080d]
```

   - Following is the sniffer traffic for Telnet application group. The Telnet traffic flows out of agg1 before 3T information is recognized, then out from vlan100 after T3 traffic is recognized:

```
# diagnose sniffer packet any 'host 2000::2:0:0:4 and dst port 23' 4  interfaces=[any]
filters=[host 2000::2:0:0:4 and dst port 23]
4.096393 port5 in 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: syn 2723132265
[flowlabel 0xd4e65]            4.096739 agg1 out 2000:172:16:205::100.43128 ->
2000::2:0:0:4.23: syn 2723132265  [flowlabel 0xd4e65]
4.096752 port4 out 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: syn 2723132265
[flowlabel 0xd4e65]
.........
5.503679 port5 in 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: psh 2723132345 ack
544895389  [flowlabel 0xd4e65]
5.503894 vlan100 out 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: psh 2723132345 ack
544895389  [flowlabel 0xd4e65]
5.503907 agg1 out 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: psh 2723132345 ack
544895389  [flowlabel 0xd4e65]
5.503918 port4 out 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: psh 2723132345 ack
```

```
544895389  [flowlabel 0xd4e65]
5.504641 port5 in 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: ack 544895390
[flowlabel 0xd4e65]
5.504713 vlan100 out 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: ack 544895390
[flowlabel 0xd4e65]
5.504721 agg1 out 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: ack 544895390
[flowlabel 0xd4e65]
5.504728 port4 out 2000:172:16:205::100.43128 -> 2000::2:0:0:4.23: ack 544895390
[flowlabel 0xd4e65]
```

- Following is the sniffer traffic for SSH application category. The SSH traffic flows out of dmz before 3T information is recognized, then out from vlan100 after T3 traffic is recognized:

```
# diagnose sniffer packet any 'host 2000::2:0:0:4 and dst port 22' 4
interfaces=[any]
filters=[host 2000::2:0:0:4 and dst port 22]
5.910752 port5 in 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: syn 980547187
[flowlabel 0xf1403]
5.911002 dmz out 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: syn 980547187  [flowlabel
0xf1403]
5.914550 port5 in 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: ack 583860244
[flowlabel 0xf1403]
5.914651 dmz out 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: ack 583860244  [flowlabel
0xf1403]
.....
8.116507 port5 in 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: psh 980549261 ack
583862554  [class 0x10] [flowlabel 0xf1403]
8.116663 vlan100 out 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: psh 980549261 ack
583862554  [class 0x10] [flowlabel 0xf1403]
8.116674 agg1 out 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: psh 980549261 ack
583862554  [class 0x10] [flowlabel 0xf1403]
8.116685 port4 out 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: psh 980549261 ack
583862554  [class 0x10] [flowlabel 0xf1403]
8.118135 port5 in 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: ack 583862598  [class
0x10] [flowlabel 0xf1403]
8.118171 vlan100 out 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: ack 583862598  [class
0x10] [flowlabel 0xf1403]
8.118179 agg1 out 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: ack 583862598  [class
0x10] [flowlabel 0xf1403]
8.118189 port4 out 2000:172:16:205::100.35146 -> 2000::2:0:0:4.22: ack 583862598  [class
0x10] [flowlabel 0xf1403]
```

4. View the IPv6 application control internet service ID list:

```
# diagnose system sdwan internet-service-app-ctrl6-list

Telnet(16091 4294837974): 2000::2:0:0:4 6 23 Thu Apr 20 17:43:00 2023
IPv6.ICMP(16321 4294837087): 2000::2:0:0:4 58 0 Thu Apr 20 17:43:00 2023
```

5. View the IPv6 application control internet service ID list by category:

```
# diagnose system sdwan internet-service-app-ctrl6-category-list

SSH(16060 4294837772): 2000::2:0:0:4 6 22 Thu Apr 20 17:43:00 2023
```

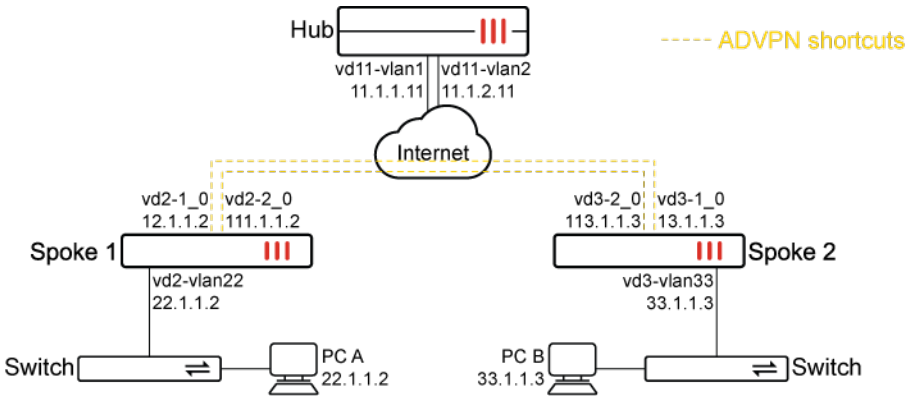
Use maximize bandwidth to load balance traffic between ADVPN shortcuts

When ADVPN is configured on a FortiGate spoke along with an SD-WAN rule set to *Maximize Bandwidth SLA* (GUI) or load balance mode (CLI) as well as `tie-break` set to `fib-best-match`, then spoke-to-spoke traffic is load balanced between multiple ADVPN shortcuts when the shortcuts are within the configured SLA conditions.

Following is an example configuration with `load-balance` enabled and `tie-break` set to `fib-best-match`:

```
config system sdwan
    config service
        edit 3
            set mode sla
            set load-balance enable
            set dst "all"
            config sla
                edit "ping"
                    set id 1
                next
            end
            set priority-members 1 2
            set tie-break fib-best-match
        next
    end
end
```

Example



In this example SD-WAN is configured between one hub and multiple spokes, and the SD-WAN configuration shows SD-WAN rule 3 with the following required settings to enable spoke-to-spoke traffic between multiple ADVPN shortcuts:

- set load-balance enable
- set tie-break fib-best-match

```
show system sdwan
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
        edit "zon2"
        next
    end
    config members
        edit 1
            set interface "vd2-1"
            set cost 10
        next
        edit 2
            set interface "vd2-2"
            set cost 20
        next
    end
    config health-check
        edit "ping"
            set server "11.11.11.11"
            set members 1 2
            config sla
                edit 1
                    set latency-threshold 200
                    set jitter-threshold 50
                next
                edit 2
                next
            end
        next
        edit "1"
        next
    end
    config service
        edit 1
            set dst "033"
            set priority-members 1
        next
        edit 2
            set dst "133"
            set priority-members 2
        next
        edit 3
            set mode sla
            set load-balance enable
            set dst "all"
            config sla
```

```
                edit "ping"
                    set id 1
                next
            end
            set priority-members 1 2
            set tie-break fib-best-match
        next
    end
end
```

To trigger spoke-to-spoke communication, run an ICMP ping on PC A with IP address 22.1.1.22 behind spoke 1 that is destined for PC B with IP address 33.1.1.33 behind spoke 2. The spoke-to-spoke traffic will be used to demonstrate load balancing between shortcuts in the CLI output of this topic.

To verify the configuration:

1. Confirm the ADVPN shortcuts are within the SLA conditions:

```
# diagnose system sdwan health-check
Health Check(ping):
Seq(1 vd2-1): state(alive), packet-loss(0.000%) latency(0.029), jitter(0.002), mos(4.404),
bandwidth-up(1999), bandwidth-dw(0), bandwidth-bi(1999) sla_map=0x3
Seq(1 vd2-1_0): state(alive), packet-loss(0.000%) latency(0.026), jitter(0.001), mos(4.404),
bandwidth-up(2000), bandwidth-dw(0), bandwidth-bi(2000) sla_map=0x3
Seq(2 vd2-2): state(alive), packet-loss(0.000%) latency(0.055), jitter(0.064), mos(4.404),
bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x3
Seq(2 vd2-2_0): state(alive), packet-loss(0.000%) latency(0.060), jitter(0.058), mos(4.404),
bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x3
```

2. Confirm the settings for SD-WAN rule 3:

```
# diagnose system sdwan service 3

Service(3): Address Mode(IPV4) flags=0x4200 use-shortcut-sla use-shortcut
 Tie break: fib
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(load-balance  hash-mode=round-robin)
  Member sub interface(4):
    1: seq_num(1), interface(vd2-1):
       1: vd2-1_0(125)
    3: seq_num(2), interface(vd2-2):
       1: vd2-2_0(127)
  Members(4):
    1: Seq_num(1 vd2-1), alive, sla(0x1), gid(2), num of pass(1), selected
    2: Seq_num(1 vd2-1_0), alive, sla(0x1), gid(2), num of pass(1), selected
    3: Seq_num(2 vd2-2), alive, sla(0x1), gid(2), num of pass(1), selected
    4: Seq_num(2 vd2-2_0), alive, sla(0x1), gid(2), num of pass(1), selected
  Dst address(1):
        0.0.0.0-255.255.255.255
```

3. Confirm firewall policing routing list:

```
# diagnose firewall proute list 2131230723
list route policy info(vf=vd2):

id=2131230723(0x7f080003) vwl_service=3 vwl_mbr_seq=1 1 2 2 dscp_tag=0xfc 0xfc flags=0x90
load-balance hash-mode=round-robin  fib-best-match tos=0x00 tos_mask=0x00 protocol=0 sport=0-
65535 iif=0(any) dport=1-65535 path(4) oif=116(vd2-1) num_pass=1 oif=125(vd2-1_0) num_pass=1
oif=117(vd2-2) num_pass=1 oif=127(vd2-2_0) num_pass=1
destination(1): 0.0.0.0-255.255.255.255
source wildcard(1): 0.0.0.0/0.0.0.0
hit_count=117 last_used=2023-04-21 15:49:59
```

4. Confirm the routing table:

```
# get router info routing-table bgp
Routing table for VRF=0
B*      0.0.0.0/0 [200/0] via 10.10.100.254 (recursive via vd2-1 tunnel 11.1.1.11), 01:26:14,
[1/0]
                  [200/0] via 10.10.200.254 (recursive via vd2-2 tunnel 11.1.2.11), 01:26:14,
[1/0]
B       1.1.1.1/32 [200/0] via 11.1.1.1 [2] (recursive via 12.1.1.1, vd2-vlan12), 01:26:14,
[1/0]
B       11.11.11.11/32 [200/0] via 10.10.100.254 (recursive via vd2-1 tunnel 11.1.1.11),
01:26:14, [1/0]
                       [200/0] via 10.10.200.254 (recursive via vd2-2 tunnel 11.1.2.11),
01:26:14, [1/0]
B       33.1.1.0/24 [200/0] via 10.10.100.3 [2] (recursive is directly connected, vd2-1_0),
01:19:41, [1/0]
                    [200/0] via 10.10.200.3 [2] (recursive is directly connected, vd2-2_0),
01:19:41, [1/0]
B       100.1.1.0/24 [200/0] via 10.10.100.254 (recursive via vd2-1 tunnel 11.1.1.11),
01:26:14, [1/0]
                     [200/0] via 10.10.200.254 (recursive via vd2-2 tunnel 11.1.2.11),
01:26:14, [1/0]
```

5. Check the packet sniffer output for the default setting.

   This step demonstrates routing for the default setting of `set tie-break zone`. The following packet sniffer output of ICMP pings demonstrates how spoke-to-spoke traffic (ping from 22.1.1.22 to 33.1.1.13) is load balanced between all parent tunnels and shortcuts, and is not limited to shortcuts within SLA.

```
# diagnose sniffer packet any "host 33.1.1.13" 4
interfaces=[any]
filters=[host 33.1.1.13]
14.665232 vd22-vlan22 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
14.665234 npu0_vlink1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
14.665240 vd2-vlan22 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
14.665262 vd2-1_0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
14.665274 vd3-1_0 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
14.665284 vd3-vlan33 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
14.665285 npu0_vlink0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
14.665289 vd33-vlan33 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
14.665299 vd33-vlan33 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
14.665300 npu0_vlink1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
```

```
14.665306 vd3-vlan33 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
14.665314 vd3-1_0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
14.665326 vd2-1_0 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
14.665331 vd2-vlan22 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
14.665332 npu0_vlink0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
14.665337 vd22-vlan22 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply

24.190955 vd22-vlan22 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.190957 npu0_vlink1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.190963 vd2-vlan22 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.190982 vd2-2 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.190993 p2 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.191002 p2 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.191020 vd3-2 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.191031 vd3-vlan33 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.191032 npu0_vlink0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.191036 vd33-vlan33 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.191046 vd33-vlan33 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.191047 npu0_vlink1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.191053 vd3-vlan33 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.191063 vd3-2 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.191074 p2 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.191079 p2 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.191090 vd2-2 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.191094 vd2-vlan22 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.191095 npu0_vlink0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.191100 vd22-vlan22 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply

51.064984 vd22-vlan22 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
51.064985 npu0_vlink1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
51.064991 vd2-vlan22 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
51.065011 vd2-2_0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
51.065022 vd3-2_0 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
51.065031 vd3-vlan33 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
51.065032 npu0_vlink0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
51.065036 vd33-vlan33 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
51.065046 vd33-vlan33 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
51.065047 npu0_vlink1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
51.065054 vd3-vlan33 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
51.065063 vd3-2_0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
51.065075 vd2-2_0 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
51.065082 vd2-vlan22 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
51.065082 npu0_vlink0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
51.065087 vd22-vlan22 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply

67.257123 vd22-vlan22 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
67.257125 npu0_vlink1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
67.257131 vd2-vlan22 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
67.257150 vd2-1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
67.257162 p1 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
67.257170 p1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
67.257189 vd3-1 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
```

```
67.257199 vd3-vlan33 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
67.257200 npu0_vlink0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
67.257205 vd33-vlan33 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
67.257216 vd33-vlan33 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
67.257217 npu0_vlink1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
67.257223 vd3-vlan33 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
67.257234 vd3-1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
67.257245 p1 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
67.257250 p1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
67.257261 vd2-1 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
67.257266 vd2-vlan22 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
67.257267 npu0_vlink0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
67.257272 vd22-vlan22 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply

^C
84 packets received by filter
0 packets dropped by kernel
```

6. Check the sniffer packet output after changing the setting to `set tie-break fib-best-match`.
   The following packet sniffer output of ICMP pings demonstrates how load balancing of spoke-to-spoke is limited and only occurs between shortcuts vd2-1_0 and vd2-2_0, which are within SLA.

```
# diagnose sniffer packet any "host 33.1.1.13" 4

interfaces=[any]
filters=[host 33.1.1.13]
2.592392 vd22-vlan22 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
2.592394 npu0_vlink1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
2.592400 vd2-vlan22 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
2.592420 vd2-1_0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
2.592432 vd3-1_0 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
2.592441 vd3-vlan33 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
2.592442 npu0_vlink0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
2.592447 vd33-vlan33 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
2.592484 vd33-vlan33 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
2.592485 npu0_vlink1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
2.592491 vd3-vlan33 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
2.592498 vd3-1_0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
2.592510 vd2-1_0 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
2.592515 vd2-vlan22 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
2.592516 npu0_vlink0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
2.592520 vd22-vlan22 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply

8.808792 vd22-vlan22 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.808793 npu0_vlink1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.808799 vd2-vlan22 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.808816 vd2-2_0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.808827 vd3-2_0 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.808838 vd3-vlan33 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.808838 npu0_vlink0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.808842 vd33-vlan33 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.808852 vd33-vlan33 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
```

```
8.808853 npu0_vlink1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.808858 vd3-vlan33 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.808866 vd3-2_0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.808877 vd2-2_0 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.808882 vd2-vlan22 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.808883 npu0_vlink0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.808887 vd22-vlan22 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply

18.024377 vd22-vlan22 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
18.024379 npu0_vlink1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
18.024385 vd2-vlan22 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
18.024400 vd2-1_0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
18.024411 vd3-1_0 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
18.024421 vd3-vlan33 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
18.024422 npu0_vlink0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
18.024427 vd33-vlan33 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
18.024436 vd33-vlan33 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
18.024437 npu0_vlink1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
18.024443 vd3-vlan33 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
18.024449 vd3-1_0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
18.024459 vd2-1_0 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
18.024463 vd2-vlan22 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
18.024464 npu0_vlink0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
18.024468 vd22-vlan22 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply

24.216469 vd22-vlan22 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.216470 npu0_vlink1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.216477 vd2-vlan22 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.216493 vd2-2_0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.216506 vd3-2_0 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.216518 vd3-vlan33 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.216519 npu0_vlink0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.216525 vd33-vlan33 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
24.216535 vd33-vlan33 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.216536 npu0_vlink1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.216542 vd3-vlan33 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.216548 vd3-2_0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.216559 vd2-2_0 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.216563 vd2-vlan22 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.216564 npu0_vlink0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
24.216568 vd22-vlan22 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
^C
70 packets received by filter
0 packets dropped by kernel
```

7. Check SD-WAN health.
   When an ADVPN shortcut is out of SLA, traffic does not run on it. Shortcut vd2-1_0 is out of SLA.

```
# diagnose system sdwan health-check
Health Check(ping):
Seq(1 vd2-1): state(alive), packet-loss(6.000%) latency(0.026), jitter(0.001), mos(4.401),
bandwidth-up(1999), bandwidth-dw(0), bandwidth-bi(1999) sla_map=0x0
Seq(1 vd2-1_0): state(alive), packet-loss(18.182%) latency(0.033), jitter(0.003), mos(4.395),
```

```
bandwidth-up(2000), bandwidth-dw(0), bandwidth-bi(2000) sla_map=0x0
Seq(2 vd2-2): state(alive), packet-loss(0.000%) latency(0.024), jitter(0.001), mos(4.404),
bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x3
Seq(2 vd2-2_0): state(alive), packet-loss(0.000%) latency(0.033), jitter(0.005), mos(4.404),
bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x3
```

8. Check the sniffer packet:
   No traffic runs on Shortcut vd2-1_0 because it is out of SLA.

```
# diagnose sniffer packet any "host 33.1.1.13" 4
interfaces=[any]
filters=[host 33.1.1.13]
8.723075 vd22-vlan22 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.723077 npu0_vlink1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.723084 vd2-vlan22 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.723103 vd2-2_0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.723115 vd3-2_0 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.723148 vd3-vlan33 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.723149 npu0_vlink0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.723154 vd33-vlan33 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
8.723166 vd33-vlan33 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.723166 npu0_vlink1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.723171 vd3-vlan33 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.723179 vd3-2_0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.723190 vd2-2_0 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.723195 vd2-vlan22 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.723195 npu0_vlink0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
8.723199 vd22-vlan22 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply

17.202681 vd22-vlan22 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
17.202683 npu0_vlink1 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
17.202688 vd2-vlan22 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
17.202704 vd2-2_0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
17.202716 vd3-2_0 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
17.202727 vd3-vlan33 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
17.202728 npu0_vlink0 out 22.1.1.22 -> 33.1.1.13: icmp: echo request
17.202733 vd33-vlan33 in 22.1.1.22 -> 33.1.1.13: icmp: echo request
17.202742 vd33-vlan33 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
17.202743 npu0_vlink1 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
17.202749 vd3-vlan33 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
17.202755 vd3-2_0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
17.202767 vd2-2_0 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
17.202771 vd2-vlan22 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
17.202772 npu0_vlink0 out 33.1.1.13 -> 22.1.1.22: icmp: echo reply
17.202777 vd22-vlan22 in 33.1.1.13 -> 22.1.1.22: icmp: echo reply
```

Administrators can use the `set use-shortcut-sla disable` command in an SD-WAN rule to ignore the shortcut's SLA and use the parent tunnel SLA instead. When combined with `set tie-break fib-best-match`, spoke-to-spoke traffic will be sent across the shortcut while the parent tunnel SLA is healthy, regardless of the shortcut tunnel's SLA.

Use SD-WAN rules to steer multicast traffic

SD-WAN rules can steer IPv4 and IPv6 multicast traffic. When an SD-WAN member is out of SLA, multicast traffic can fail over to another SD-WAN member, and switch back when SLA recovers.

The `pim-use-sdwan` option enables or disables the use of SD-WAN for PIM (Protocol Independent Multicast) when checking RP (Rendezvous Point) neighbors and sending packets.

```
config router multicast
    config pim-sm-global
        set pim-use-sdwan {enable | disable}
    end
end
```

or

```
config router multicast6
    config pim-sm-global
        set pim-use-sdwan {enable | disable}
    end
end
```

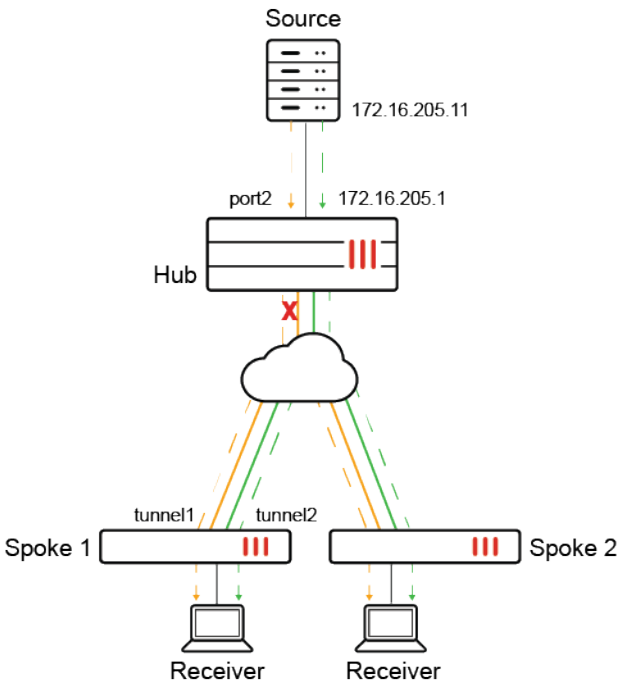
When SD-WAN steers multicast traffic, ADVPN is not supported. Use the `set shortcut` option to disable shortcuts for the service:

```
config system sdwan
    config service
        edit <id>
            set shortcut {enable | disable}
        next
    end
end
```

Example 1

In this hub and spoke example for IPv4 multicast traffic, the PIM source is behind the hub FortiGate, and the RP is set to internal port (port2) of the hub firewall. Each spoke connects to the two WAN interfaces on the hub by using an overlay tunnel. The overlay tunnels are members of SD-WAN.

Receivers behind the spoke FortiGates request a stream from the source to receive traffic on tunnel1 by default. When the overlay tunnel goes out of SLA, the multicast traffic fails over to tunnel2 and continues to flow.



Following is an overview of how to configure the topology:

1. Configure the hub FortiGate in front of the PIM source. The RP is configured on internal port (port2) of the hub FortiGate.
2. Configure the spoke FortiGates.
3. Verify traffic failover.

To configure the hub:

1. On the hub, enable multicast routing, configure the multicast RP, and enable PIM sparse mode on each interface:

```
config router multicast
    set multicast-routing enable
    config pim-sm-global
        config rp-address
            edit 1
                set ip-address 172.16.205.1
            next
        end
    end
    config interface
        edit "tport1"
            set pim-mode sparse-mode
        next
        edit "tagg1"
            set pim-mode sparse-mode
        next
        edit "port2"
            set pim-mode sparse-mode
```

```
        next
    end
end
```

To configure each spoke:

1. Enable SD-WAN with the following settings:
   - Configure the overlay tunnels as member of the SD-WAN zone.
   - Configure a performance SLA health-check using ping.
   - Configure a service rule for the PIM protocol with the following settings:
     - Use the lowest cost (SLA) strategy.
     - Monitor with the ping health-check.
   - Disable ADVPN shortcut.

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "tunnel1"
        next
        edit 2
            set interface "tunnel2"
        next
    end
    config health-check
        edit "ping"
            set server "172.16.205.1"
            set update-static-route disable
            set members 0
            config sla
                edit 1
                next
            end
        next
    end
    config service
        edit 1
            set mode sla
            set protocol 103
            set dst "all"
            config sla
                edit "ping"
                    set id 1
                next
            end
            set priority-members 1 2
```

```
            set use-shortcut-sla disable
            set shortcut disable
        next
        edit 2
            set mode sla
            set dst "all"
            config sla
                edit "ping"
                    set id 1
                next
            end
            set priority-members 1 2
        next
    end
end
```

2. Enable multicast routing and configure the multicast RP. Enable PIM sparse-mode on each interface:

```
config router multicast
    set multicast-routing enable
    config pim-sm-global
        set spt-threshold disable
        set pim-use-sdwan enable
        config rp-address
            edit 1
                set ip-address 172.16.205.1
            next
        end
    end
    config interface
        edit "tunnel1"
            set pim-mode sparse-mode
        next
        edit "tunnel2"
            set pim-mode sparse-mode
        next
        edit "port4"
            set pim-mode sparse-mode
        next
    end
end
```

To verify traffic failover:

With this configuration, multicast traffic starts on tunnel1. When tunnel1 becomes out of SLA, traffic switches to tunnel2. When tunnel1 is in SLA again, the traffic switches back to tunnel1.

The following health-check capture on the spokes shows tunnel1 in SLA with packet-loss (1.000%):

```
# diagnose sys sdwan health-check
Health Check(ping):
Seq(1 tunnel1): state(alive), packet-loss(0.000%) latency(0.056), jitter(0.002), mos(4.404),
```

```
bandwidth-up(999999), bandwidth-dw(1000000), bandwidth-bi(1999999) sla_map=0x1
Seq(2 tunnel2): state(alive), packet-loss(0.000%) latency(0.100), jitter(0.002), mos(4.404),
bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x1

# diagnose sys sdwan health-check
Health Check(ping):
Seq(1 tunnel1): state(alive), packet-loss(1.000%) latency(0.056), jitter(0.002), mos(4.404),
bandwidth-up(999999), bandwidth-dw(1000000), bandwidth-bi(1999999) sla_map=0x1
Seq(2 tunnel2): state(alive), packet-loss(0.000%) latency(0.100), jitter(0.002), mos(4.404),
bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x1
```

The following example shows tunnel1 out of SLA with packet-loss (3.000%):

```
# diagnose sys sdwan health-check
Health Check(ping):
Seq(1 tunnel1): state(alive), packet-loss(3.000%) latency(0.057), jitter(0.003), mos(4.403),
bandwidth-up(999999), bandwidth-dw(1000000), bandwidth-bi(1999999) sla_map=0x0
Seq(2 tunnel2): state(alive), packet-loss(0.000%) latency(0.101), jitter(0.002), mos(4.404),
bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x1
```

The following example shows tunnel1 back in SLA again:

```
# diagnose sys sdwan health-check
Health Check(ping):
Seq(1 tunnel1): state(alive), packet-loss(1.000%) latency(0.061), jitter(0.004), mos(4.404),
bandwidth-up(999999), bandwidth-dw(1000000), bandwidth-bi(1999999) sla_map=0x0
Seq(2 tunnel2): state(alive), packet-loss(0.000%) latency(0.102), jitter(0.002), mos(4.404),
bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x1

# diagnose sys sdwan health-check
Health Check(ping):
Seq(1 tunnel1): state(alive), packet-loss(0.000%) latency(0.061), jitter(0.004), mos(4.404),
bandwidth-up(999999), bandwidth-dw(1000000), bandwidth-bi(1999999) sla_map=0x0
Seq(2 tunnel2): state(alive), packet-loss(0.000%) latency(0.102), jitter(0.002), mos(4.404),
bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x1
```

The following example how traffic switches to tunnel2 while tunnel1 health-check is out of SLA. Source (172.16.205.11) sends traffic to the multicast group. Later the traffic switches back to tunnel1 once SLA returns to normal:

```
195.060797 tunnel1 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
195.060805 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
196.060744 tunnel1 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
196.060752 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
197.060728 tunnel1 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
197.060740 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
198.060720 tunnel2 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
198.060736 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
199.060647 tunnel2 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
199.060655 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
200.060598 tunnel2 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
200.060604 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
... ...
```

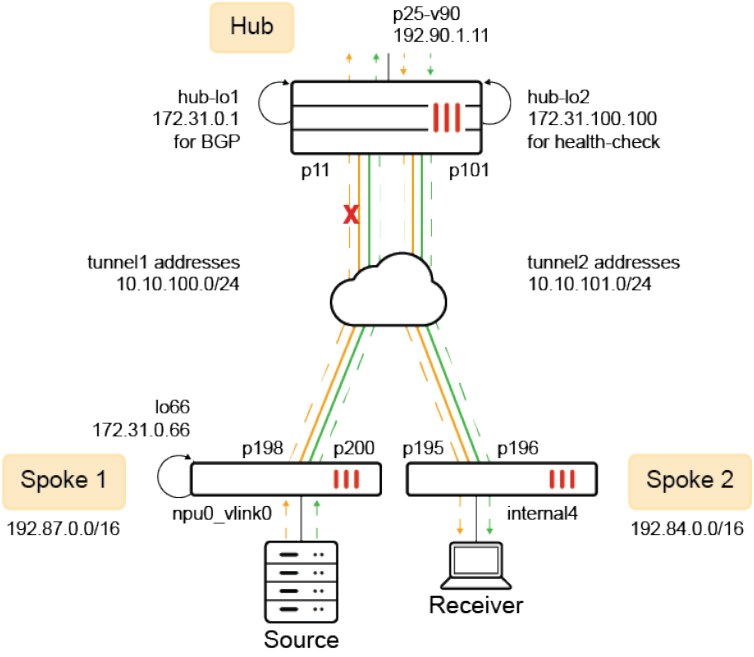
```
... ...
264.060974 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
265.060950 tunnel2 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
265.060958 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
266.060867 tunnel2 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
266.060877 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
267.060828 tunnel2 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
267.060835 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
268.060836 tunnel1 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
268.060854 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
269.060757 tunnel1 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
269.060767 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
270.060645 tunnel1 in 172.16.205.11 -> 225.1.1.1: icmp: echo request
270.060653 port4 out 172.16.205.11 -> 225.1.1.1: icmp: echo request
```

Example 2

In this hub and spoke example for IPv4 multicast traffic, the PIM source is behind spoke 1, and the RP is configured on the hub FortiGate. BGP is used for routing. The hub uses embedded SLA in ICMP probes to determine the health of each tunnel, allowing it to prioritize healthy IKE routes.

The receiver is on another spoke. Upon requesting a stream, source passes the traffic to the RP on the hub FortiGate, and routes the traffic to the receiver over tunnel1. If a tunnel falls out of SLA, the multicast traffic fails over to the other tunnel.

In this configuration, SD-WAN steers multicast traffic by using embedded SLA information in ICMP probes. See also Embedded SD-WAN SLA information in ICMP probes. With this feature, the hub FortiGate can use the SLA information of the spoke's health-check to control BGP and IKE routes over tunnels.



Following is an overview of how to configure the topology:

1. Configure the hub FortiGate. The RP is configured on the hub FortiGate.
2. Configure the spoke FortiGate in front of the traffic receiver.
3. Configure the spoke FortiGate in front of the PIM source.

To configure the hub:

1. Configure loopbacks hub-lo1 172.31.0.1 for BGP and hub-lo100 172.31.100.100 for health-check:

```
config system interface
    edit "hub-lo1"
        set vdom "hub"
        set ip 172.31.0.1 255.255.255.255
        set allowaccess ping
        set type loopback
        set snmp-index 82
    next
    edit "hub-lo100"
        set vdom "hub"
        set ip 172.31.100.100 255.255.255.255
        set allowaccess ping
        set type loopback
        set snmp-index 81
    next
end
```

2. Enable multicast routing with the following settings:
   - Configure internal interface p25-v90 as RP.
   - Enable interfaces for PIM sparse-mode.

```
config router multicast
    set multicast-routing enable
    config pim-sm-global
        config rp-address
            edit 1
                set ip-address 192.90.1.11
            next
        end
    end
    config interface
        edit "p11"
            set pim-mode sparse-mode
        next
        edit "p101"
            set pim-mode sparse-mode
        next
        edit "p25-v90"
            set pim-mode sparse-mode
        next
    end
end
```

3. Enable SD-WAN with the following settings:

- Add interfaces p11 and p101 as members.
- Configure embedded SLA health-checks to detect ICMP probes from each overlay tunnel. Prioritize based on the health of each tunnel.

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "p11"
        next
        edit 2
            set interface "p101"
        next
    end
    config health-check
        edit "1"
            set detect-mode remote
            set remote-probe-timeout 60000
            set recoverytime 1
            set sla-id-redistribute 1
            set members 1
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 100
                    set priority-in-sla 10
                    set priority-out-sla 20
                next
            end
        next
        edit "2"
            set detect-mode remote
            set remote-probe-timeout 60000
            set recoverytime 1
            set sla-id-redistribute 1
            set members 2
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 100
                    set priority-in-sla 15
                    set priority-out-sla 25
                next
            end
        next
    end
end
```

4. Configure BGP to peer with neighbors. Neighbor group is configured for tunnel interface IP addresses:

```
config router bgp
    set as 65505
    set router-id 172.31.0.1
    set ibgp-multipath enable
    set additional-path enable
    set recursive-inherit-priority enable
    config neighbor-group
        edit "gr1"
            set remote-as 65505
            set update-source "hub-lo1"
            set additional-path both
            set route-reflector-client enable
        next
    end
    config neighbor-range
        edit 1
            set prefix 10.10.0.0 255.255.0.0
            set neighbor-group "gr1"
        next
        edit 66
            set prefix 172.31.0.66 255.255.255.255
            set neighbor-group "gr1"
        next
    end
    config network
        ....
        edit 90
            set prefix 192.90.0.0 255.255.0.0
        next
    end
end
```

To configure the spoke (in front of the receiver):

1. Enable multicast routing to use SD-WAN. Configure the RP address. Enable interfaces for PIM sparse-mode.

```
config router multicast
    set multicast-routing enable
    config pim-sm-global
        set spt-threshold disable
        set pim-use-sdwan enable
        config rp-address
            edit 1
                set ip-address 192.90.1.11
            next
        end
    end
    config interface
        edit "p195"
            set pim-mode sparse-mode
        next
```

```
        edit "p196"
            set pim-mode sparse-mode
        next
        edit "internal4"
            set pim-mode sparse-mode
            set static-group "225-1-1-122"
        next
    end
end
```

2. Configure SD-WAN with the following settings:
   - Add overlay tunnel interfaces as members.
   - Configure a performance SLA health-check to send ping probes to the hub.
   - Configure a service rule for the PIM protocol. Use the lowest cost (SLA) strategy, and monitor with the ping health-check.
   - Disable ADVPN shortcuts.

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 6
            set interface "p196"
        next
        edit 5
            set interface "p195"
        next
    end
    config health-check
        edit "ping"
            set server "172.31.100.100"
            set update-static-route disable
            set members 0
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 100
                next
            end
        next
    end
    config service
        edit 1
            set mode sla
            set protocol 103
            set dst "all"
            config sla
                edit "ping"
```

```
                    set id 1
                next
            end
            set priority-members 5 6
            set use-shortcut-sla disable
            set shortcut disable
        next
        edit 2
            set mode sla
            set dst "all"
            config sla
                edit "ping"
                    set id 1
                next
            end
            set priority-members 5 6
        next
    end
end
```

3. Configure BGP and set neighbors to the overlay gateway IP address on the hub:

```
config router bgp
    set as 65505
    set router-id 122.1.1.122
    set ibgp-multipath enable
    set additional-path enable
    config neighbor
        edit "10.10.100.254"
            set soft-reconfiguration enable
            set remote-as 65505
            set connect-timer 10
            set additional-path both
        next
        edit "10.10.101.254"
            set soft-reconfiguration enable
            set remote-as 65505
            set connect-timer 10
            set additional-path both
        next
    end
    config network
        edit 3
            set prefix 192.84.0.0 255.255.0.0
        next
    end
end
```

4. Configure the default gateway to use the SD-WAN zone. Other routes are for the underlay to route traffic to the hub's WAN interfaces:

```
config router static
    edit 10
        set distance 1
        set sdwan-zone "virtual-wan-link"
    next
    ....
    next
end
```

To configure the spoke (in front of the source):

1. Enable multicast routing to use SD-WAN. Configure the RP address. Enable interfaces for PIM sparse-mode:

```
config router multicast
    set multicast-routing enable
    config pim-sm-global
        set pim-use-sdwan enable
        config rp-address
            edit 1
                set ip-address 192.90.1.11
            next
        end
    end
    config interface
        edit "p198"
            set pim-mode sparse-mode
        next
        edit "p200"
            set pim-mode sparse-mode
        next
        edit "npu0_vlink0"
            set pim-mode sparse-mode
        next
    end
end
```

2. Configure loopback interface lo66 for BGP and sourcing SD-WAN traffic:

```
config system interface
    edit "lo66"
        set vdom "root"
        set ip 172.31.0.66 255.255.255.255
        set allowaccess ping
        set type loopback
        set snmp-index 21
    next
end
```

3. Configure SD-WAN:
   - Add overlay tunnel interfaces as members.
   - Configure a performance SLA health-check to send ping probes to the hub.

- Configure a service rule for the PIM protocol. Use the lowest cost (SLA) strategy, and monitor with the ping health-check.
- Disable the use of an ADVPN shortcut.

In the following example, 11.11.11.11 is the underlay address for one of the WAN links on the hub, and 172.31.100.100 is the loopback address on the server.

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
        edit "overlay"
        next
    end
    config members
        edit 1
            set interface "p198"
            set zone "overlay"
            set source 172.31.0.66
        next
        edit 2
            set interface "p200"
            set zone "overlay"
            set source 172.31.0.66
        next
    end
    config health-check
        edit "ping"
            set server "11.11.11.11"
            set members 0
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 100
                next
            end
        next
        edit "HUB"
            set server "172.31.100.100"
            set embed-measured-health enable
            set members 0
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 100
                next
            end
        next
    end
     config service
        edit 1
            set mode sla
```

```
            set protocol 103
            set dst "all"
            config sla
                edit "ping"
                    set id 1
                next
            end
            set priority-members 1 2
            set use-shortcut-sla disable
            set shortcut disable
        next
        edit 2
            set mode sla
            set dst "all"
            config sla
                edit "ping"
                    set id 1
                next
            end
            set priority-members 1 2
        next
    end
end
```

4. Configure BGP:

```
config router bgp
    set as 65505
    set router-id 123.1.1.123
    set ibgp-multipath enable
    set additional-path enable
    config neighbor
        edit "172.31.0.1"
            set next-hop-self enable
            set soft-reconfiguration enable
            set remote-as 65505
            set update-source "lo66"
        next
    end
    config network
        edit 3
            set prefix 192.87.0.0 255.255.0.0
        next
    end
end
```

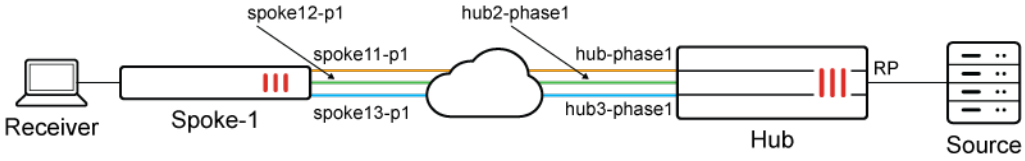
5. Configure the default gateway to use the SD-WAN zone. Other routes are for the underlay to route to the hub's WAN interfaces:

```
config router static
    edit 10
        set distance 1
```

```
        set sdwan-zone "virtual-wan-link" "overlay"
    next
    ...
    next
end
```

Example 3

In the following example for using SD-WAN rules to steer IPv6 multicast traffic, three PIM-SM enabled tunnels are configured between Spoke-1 and the Hub. The multicast source is located at Hub, and the multicast receiver is attached to Spoke-1.



This example focuses on configuration related allowing SD-WAN rules to steer IPv6 multicast traffic. Following is an overview of the configuration steps:

1. On the hub FortiGate, configure multicast routing for the source and the multicast RP.
2. On the spoke FortiGate, configuring multicast routing and enable SD-WAN for steering.
3. Verify traffic failover for the following scenarios:
   - When the cost of an SD-WAN member changes
   - When a link is in SLA
   - When a link is out of SLA

To configure the Hub:

1. On Hub, configure multicast routing for the source and the multicast RP:
   In this example, `port5` is used for the multicast source, and `20000:172:16:205::1` is the IPv6 address for the RP.

```
config router multicast6
    set multicast-routing enable
    config interface
        edit "hub-phase1"
        next
        edit "hub2-phase1"
        next
        edit "port5"
        next
        edit "hub3-phase1"
        next
    end
    config pim-sm-global
        config rp-address
            edit 1
                set ip6-address 2000:172:16:205::1
```

```
            next
        end
    end
end
```

2. Configure the firewall policy:

```
config firewall multicast-policy6
    edit 1
        set srcintf "port5"
        set dstintf "any"
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

3. Verify that all PIM-SM neighbors are established:

```
# get router info6 multicast pim sparse-mode neighbor
Neighbor                   Interface          Uptime/Expires    Ver   DR
Address                                                             Prio/Mode
fe80::1                    hub-phase1         06:49:35/00:01:39 v2    1 /
fe80::2                    hub2-phase1        06:49:34/00:01:42 v2    1 /
fe80::1                    hub3-phase1        02:41:17/00:01:31 v2    1 /
```

To configure Spoke-1:

1. On Spoke-1, configure multicast routing and enable SD-WAN for steering:
   In this example, `port5` is used for the multicast receiver, the use of SD-WAN for steering is enabled, and `20000:172:16:205::1` is the IPv6 address for the RP.

```
config router multicast6
    set multicast-routing enable
    config interface
        edit "spoke11-p1"
        next
        edit "spoke12-p1"
        next
        edit "port2"
        next
        edit "spoke13-p1"
        next
    end
    config pim-sm-global
        set pim-use-sdwan enable
        config rp-address
            edit 1
                set ip6-address 2000:172:16:205::1
            next
        end
    end
end
```

2. Configure the firewall policy:

```
config firewall multicast-policy6
    edit 1
        set srcintf "any"
        set dstintf "any"
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

3. Configure SD-WAN:
   In this example, the `protocol` is set to `103` to match PIM-SM join/register messages.

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "spoke11-p1"
        next
        edit 2
            set interface "spoke12-p1"
        next
        edit 3
            set interface "spoke13-p1"
        next
    end
    config health-check
        edit "1"
            set addr-mode ipv6
            set server "2000::9:0:0:1"
            set update-static-route disable
            set members 1
            config sla
                edit 1
                next
            end
        next
        edit "2"
            set addr-mode ipv6
            set server "2000::9:0:0:2"
            set update-static-route disable
            set members 2
            config sla
                edit 1
                next
            end
        next
```

```
        edit "3"
            set addr-mode ipv6
            set server "2000::9:0:0:3"
            set update-static-route disable
            set members 3
            config sla
                edit 1
                next
            end
        next
    end
    config service
        edit 1
            set name "1"
            set addr-mode ipv6
            set mode sla
            set protocol 103
            config sla
                edit "1"
                    set id 1
                next
                edit "2"
                    set id 1
                next
                edit "3"
                    set id 1
                next
            end
            set priority-members 1 2 3
            set sla-compare-method number
            set dst6 "all"
        next
    end
end
```

4. Verify that all PIM-SM neighbors are established:

```
# get router info6 multicast pim sparse-mode neighbor
Neighbor                   Interface          Uptime/Expires    Ver   DR
Address                                                             Prio/Mode
fe80:10:10:15::253         spoke11-p1         06:49:50/00:01:16 v2    1 / DR
fe80:10:10:16::253         spoke12-p1         06:49:50/00:01:26 v2    1 / DR
fe80:10:10:17::253         spoke13-p1         02:41:32/00:01:43 v2    1 / DR
```

To verify traffic failover:

1. On Spoke-1, diagnose the SD-WAN service. The preferred route is spoke11-p1 to hub-phase1:

```
# diagnose sys sdwan service6

Service(1): Address Mode(IPV6) flags=0x4200 use-shortcut-sla use-shortcut
 Tie break: cfg
```

```
 Shortcut priority: 2
  Gen(1), TOS(0x0/0x0), Protocol(103): src(1->65535):dst(1->65535), Mode(sla), sla-compare-
number
  Members(3):
    1: Seq_num(1 spoke11-p1 virtual-wan-link), alive, sla(0x1), gid(0), cfg_order(0), local
cost(0), selected           >>>>>>> spoke11-p1 which is connected to hub-phase1 is preferred
    2: Seq_num(2 spoke12-p1 virtual-wan-link), alive, sla(0x2), gid(0), cfg_order(1), local
cost(0), selected
    3: Seq_num(3 spoke13-p1 virtual-wan-link), alive, sla(0x4), gid(0), cfg_order(2), local
cost(0), selected
  Dst6 address(1): ::/0
```

2. When the receiver initiates IGMP to join group `ff15::10`, view `mroute` on Spoke-1 and Hub:
   - On Spoke-1:

     The `RPF idx` is connected to hub-phase1, indicating that PIM-SM join message follows SD-WAN service and is sent to `spoke11-p1`, and `port2` is connected to the receiver.

     ```
     FGT_B (root)# get router info6 multicast pim sparse-mode mroute ff15::10
     IP Multicast Routing Table

     ......

     (*, ff15::10)
     RP: 2000:172:16:205::1
     RPF nbr: fe80:10:10:15::253
     RPF idx: spoke11-p1
     Upstream State: JOINED
      Local:
          port2
      Joined:
      Asserted:
     FCR:
     Source: 2000:172:16:205::100
      Outgoing:
          port2
      KAT timer running, 196 seconds remaining
      Packet count 168
     ...
     ```

   - On the Hub:

     We see that `hub-phase1` is connected to spoke11-p1 on Spoke-1.

     ```
     FGT_A (root) (Interim)# get router info6 multicast pim sparse-mode mroute ff15::10
     IP Multicast Routing Table

     ......

     (*, ff15::10)
     RP: 2000:172:16:205::1
     RPF nbr: ::
     RPF idx: None
     ```

```
Upstream State: JOINED
 Local:
 Joined:
     hub-phase1
 Asserted:
FCR:

...
```

3. The server starts to send multicast traffic to group `ff15::10`, and Hub forwards the traffic to Spoke-1 through hub-phase1.

```
FGT_A (root) (Interim)# diagnose sniffer packet any 'host  ff15::10' 4
interfaces=[any]
filters=[host  ff15::10]
0.637174 port5 in 2000:172:16:205::100.38823 -> ff15::10.12345: udp 46 [flowlabel 0x8ea58]
0.637228 hub-phase1 out 2000:172:16:205::100.38823 -> ff15::10.12345: udp 46 [flowlabel
0x8ea58]
```

4. When the cost of member spoke11-p1 and spoke12-p1 is increased, SD-WAN prefers spoke13-p1.

   The PIM-SM join message from Spoke-1 to RP is sent to member spoke13-p1, and multicast traffic fails over to hub3-phase1 on the Hub accordingly.

   - On Spoke-1:

     In this example, `spoke13-p1`, which is connected to hub-phase3, is preferred.

```
FGT_B (root) (Interim)# diagnose sys sdwan service6

Service(1): Address Mode(IPV6) flags=0x4200 use-shortcut-sla use-shortcut
 Tie break: cfg
 Shortcut priority: 2
  Gen(1), TOS(0x0/0x0), Protocol(103): src(1->65535):dst(1->65535), Mode(sla), sla-
compare-number
  Members(3):
    1: Seq_num(3 spoke13-p1 virtual-wan-link), alive, sla(0x4), gid(0), cfg_order(2),
local cost(0), selected
    2: Seq_num(1 spoke11-p1 virtual-wan-link), alive, sla(0x1), gid(0), cfg_order(0),
local cost(20), selected
    3: Seq_num(2 spoke12-p1 virtual-wan-link), alive, sla(0x2), gid(0), cfg_order(1),
local cost(20), selected
  Dst6 address(1): ::/0
```

   - On the Hub:

     Once the cost of spoke11-p1 is increased, multicast traffic fails over to `hub2-phase1`. Once the cost of spoke12-p1 is increased, multicast traffic fails over to `hub3-phase1`.

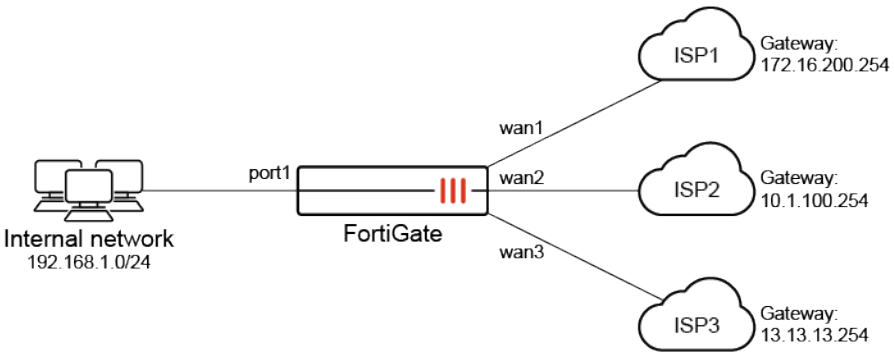
```
FGT_A (root) (Interim)# diagnose sniffer packet any 'host  ff15::10' 4
interfaces=[any]
filters=[host  ff15::10]

....
385.497887 port5 in 2000:172:16:205::100.41944 -> ff15::10.12345: udp 46 [flowlabel
```

```
0xa5e3d]
385.497927 hub-phase1 out 2000:172:16:205::100.41944 -> ff15::10.12345: udp 46 [flowlabel
0xa5e3d]
386.497967 port5 in 2000:172:16:205::100.41944 -> ff15::10.12345: udp 46 [flowlabel
0xa5e3d]
386.498258 hub2-phase1 out 2000:172:16:205::100.41944 -> ff15::10.12345: udp 46 [flowlabel
0xa5e3d]
387.498044 port5 in 2000:172:16:205::100.41944 -> ff15::10.12345: udp 46 [flowlabel
0xa5e3d]
...
400.499075 port5 in 2000:172:16:205::100.41944 -> ff15::10.12345: udp 46 [flowlabel
0xa5e3d]
400.499120 hub2-phase1 out 2000:172:16:205::100.41944 -> ff15::10.12345: udp 46 [flowlabel
0xa5e3d]
401.499180 port5 in 2000:172:16:205::100.41944 -> ff15::10.12345: udp 46 [flowlabel
0xa5e3d]
401.499515 hub3-phase1 out 2000:172:16:205::100.41944 -> ff15::10.12345: udp 46 [flowlabel
0xa5e3d]
402.499254 port5 in 2000:172:16:205::100.41944 -> ff15::10.12345: udp 46 [flowlabel
0xa5e3d]
402.499319 hub3-phase1 out 2000:172:16:205::100.41944 -> ff15::10.12345: udp 46 [flowlabel
0xa5e3d]
403.499330 port5 in 2000:172:16:205::100.41944 -> ff15::10.12345: udp 46 [flowlabel
0xa5e3d]
...
```

5. When spoke13-p1 becomes out of SLA, SD-WAN selects spoke11-p1 as the preferred member.
   This change redirects the PIM-SM join message from Spoke-1 to RP towards spoke11-p1, causing the multicast traffic to failover to hub-phase1 on the Hub.
6. Conversely, when spoke13-p1 is in SLA again, it is prioritized by SD-WAN.
   This adjustment redirects the PIM-SM join message from Spoke-1 to RP towards spoke13-p1, triggering a failover of the multicast traffic to hub3-phase1 on the Hub.

Use SD-WAN rules for WAN link selection with load balancing

This example covers a use case where a user has multiple WAN links and wants to optimize the WAN link selection and performance while limiting the use of more expensive and bandwidth intensive interfaces, such as 5G or LTE.



In this scenario, the user has three WAN links. The goal is to balance the load between wan1 and wan2; however, wan3, which is quite costly to operate, should only be used if both wan1 and wan2 are unavailable.

This configuration involves the following steps:

1. Configuring the SD-WAN members
2. Configuring the manual SD-WAN rule
3. Configuring a static route
4. Configuring a firewall policy for SD-WAN
5. Verifying the configuration

Configuring the SD-WAN members

SD-WAN must be enabled first, and member interfaces must be selected and added to a zone. See Underlay on page 1042 for more information.

To configure the SD-WAN members in the GUI:

1. Configure the wan1, wan2, and wan3 interfaces (see Interface settings on page 225 for more details).
   a. Set the wan1 interface *IP/Netmask* to *172.16.200.1 255.255.255.0*.
   b. Set the wan2 interface *IP/Netmask* to *10.1.100.1 255.255.255.0*.
   c. Set the wan3 interface *IP/Netmask* to *13.13.13.1 255.255.255.0*.
2. Go to *Network* > *SD-WAN*, select the *SD-WAN Zones* tab, and click *Create New* > *SD-WAN Member*.
3. Configure the wan1 SD-WAN member:
   a. Set the *Interface* to *wan1*.
   b. Leave the *SD-WAN Zone* as *virtual-wan-link*.
   c. Set the *Gateway* to *172.16.200.254*.
   d. Set the *Status* to *Enable*
   e. Click *OK*.
4. Repeat step 3 for wan2 and wan3.
   a. For wan2, set the *Gateway* to the ISP's gateway, *10.1.100.254*.
   b. For wan3, set the *Gateway* to the ISP's gateway, *13.13.13.254*.

To configure the SD-WAN members in the CLI:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "wan1"
            set gateway 172.16.200.254
        next
        edit 2
            set interface "wan2"
            set gateway 10.1.100.254
        next
        edit 3
            set interface "wan3"
            set gateway 13.13.13.254
        next
    end
end
```

Configuring the manual SD-WAN rule

SD-WAN rules define specific routing options to route traffic to an SD-WAN member. See SD-WAN rules on page 1105 and Manual strategy on page 1119 for more information.

To configure a manual SD-WAN rule in the GUI:

1. Go to *Network* > *SD-WAN*, select the *SD-WAN Rules* tab, and click *Create New*.
2. Configure the following settings:

Name	*test*
Source > Address	*all*
Destination > Address	*all*
Interface selection strategy	*Manual*
Interface preference	*wan1*, *wan2*
Load balancing	Enable this setting.

3. Configure the other settings as needed.
4. Click *OK*.

To configure a manual SD-WAN rule in the CLI:

```
config system sdwan
    config service
        edit 1
            set name "test"
            set load-balance enable
            set dst "all"
            set src "all"
            set priority-members 1 2
        next
    end
end
```

Configuring a static route

A default route for SD-WAN must be configured. See Routing on page 1043 for more information.

To configure a static route for SD-WAN in the GUI:

1. Go to *Network* > *Static Routes* and click *Create New*. The *New Static Route* page opens.
2. Set the *Destination* to *Subnet*, and leave the IP address and subnet mask as *0.0.0.0/0.0.0.0*.
3. Set the *Interface* to the SD-WAN zone, *virtual-wan-link*.
4. *Set the Status to Enabled*.
5. Click *OK*.

To configure a static route for SD-WAN in the CLI:

```
config router static
    edit 1
        set distance 1
        set sdwan-zone "virtual-wan-link"
    next
end
```

Configuring a firewall policy for SD-WAN

A firewall policy must be configured that allows traffic from the organization's internal network to the SD-WAN zone. See Security on page 1044 for more information.

To configure the firewall policy for SD-WAN in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
2. Configure the following settings:

Name	*sd-wan*

Incoming interface	*port1*
Outgoing interface	*virtual-wan-link*
Source	*all*
Destination	*all*
Schedule	*always*
Service	*ALL*
Action	*ACCEPT*
NAT	Enable and select *NAT*.
IP Pool Configuration	*Use Outgoing Interface Address*
Enable this policy	Enable this setting.

3. Configure the other settings as needed.
4. Click *OK*.

To configure the firewall policy for SD-WAN in the CLI:

```
config firewall policy
    edit 1
        set name "sd-wan"
        set srcintf "port1"
        set dstintf "virtual-wan-link"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

Verifying the configuration

To verify the SD-WAN member status:

```
# diagnose sys sdwan member
Member(1): interface: wan1, flags=0x0 , gateway: 172.16.200.254, priority: 1 1024, weight: 0
Member(2): interface: wan2, flags=0x0 , gateway: 10.1.100.254, priority: 1 1024, weight: 0
Member(3): interface: wan3, flags=0x0 , gateway: 13.13.13.254, priority: 1 1024, weight: 0
```

To verify the configuration when both wan1 and wan2 are up:

1. Verify the SD-WAN service rules status:

```
# diagnose sys sdwan service4
Service(1): Address Mode(IPV4) flags=0x24200 use-shortcut-sla use-shortcut
 Tie break: cfg
  Gen(1), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(manual  hash-
mode=round-robin)
  Members(2):
    1: Seq_num(2 wan2 virtual-wan-link), alive, gid(1), selected
    2: Seq_num(1 wan1 virtual-wan-link), alive, gid(1), selected
  Src address(1):
        0.0.0.0-255.255.255.255

  Dst address(1):
        0.0.0.0-255.255.255.255
```

   This output indicates that both wan1 and wan2 are operational.

2. Verify the policy route list:

```
# diagnose firewall proute list
list route policy info(vf=root):

id=2130706433(0x7f000001) vwl_service=1(test) vwl_mbr_seq=1 2 dscp_tag=0xfc 0xfc flags=0x10
load-balance hash-mode=round-robin  tos=0x00 tos_mask=0x00 protocol=0 port=src(0->0):dst(0->0)
iif=0(any)
path(2): oif=3(wan1) num_pass=0, oif=6(wan2) num_pass=0
source(1): 0.0.0.0-255.255.255.255
destination(1): 0.0.0.0-255.255.255.255
hit_count=154 last_used=2023-11-09 06:16:
```

   This output indicates that both wan1 and wan2 are used to steer traffic.

To verify the configuration when wan2 is down and wan1 is up:

1. Verify the SD-WAN service rules status:

```
# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x24200 use-shortcut-sla use-shortcut
 Tie break: cfg
  Gen(1), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(manual  hash-
mode=round-robin)
  Members(2):
    1: Seq_num(1 wan1 virtual-wan-link), alive, gid(1), selected
    2: Seq_num(2 wan2 virtual-wan-link), dead, gid(1)
  Src address(1):
        0.0.0.0-255.255.255.255

  Dst address(1):
        0.0.0.0-255.255.255.255
```

This output indicates that wan1 is operational, and wan2 is not.

2. Verify the policy route list:

```
# diagnose firewall proute list
list route policy info(vf=root):

id=2130706433(0x7f000001) vwl_service=1(test) vwl_mbr_seq=1 dscp_tag=0xfc 0xfc flags=0x10
load-balance hash-mode=round-robin  tos=0x00 tos_mask=0x00 protocol=0 port=src(0->0):dst(0->0)
iif=0(any)
path(1): oif=3(wan1) num_pass=0
source(1): 0.0.0.0-255.255.255.255
destination(1): 0.0.0.0-255.255.255.255
hit_count=482 last_used=2023-11-09 06:27:08
```

This output indicates that wan1 is used to steer traffic.

To verify the configuration when wan1 is down and wan2 is up:

1. Verify the SD-WAN service rules status:

```
# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x24200 use-shortcut-sla use-shortcut
 Tie break: cfg
  Gen(1), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(manual  hash-
mode=round-robin)
  Members(2):
    1: Seq_num(2 wan2 virtual-wan-link), alive, gid(1), selected
    2: Seq_num(1 wan1 virtual-wan-link), dead, gid(1)
  Src address(1):
        0.0.0.0-255.255.255.255

  Dst address(1):
        0.0.0.0-255.255.255.255
```

This output indicates that wan2 is operational, and wan1 is not.

2. Verify the policy route list:

```
# diagnose firewall  proute list
list route policy info(vf=root):

id=2130706433(0x7f000001) vwl_service=1(test) vwl_mbr_seq=2 dscp_tag=0xfc 0xfc flags=0x10
load-balance has
h-mode=round-robin  tos=0x00 tos_mask=0x00 protocol=0 port=src(0->0):dst(0->0) iif=0(any)
path(1): oif=6(wan2) num_pass=0
source(1): 0.0.0.0-255.255.255.255
destination(1): 0.0.0.0-255.255.255.255
hit_count=903 last_used=2023-11-09 06:41:55
```

This output indicates that wan2 is used to steer traffic.

To verify the configuration when both wan1 and wan2 down, and traffic is steered using wan3:

```
# diagnose sniffer packet wan3
Using Original Sniffing Mode
interfaces=[wan3]
filters=[none]
3.144417 13.13.13.1.52665 -> 204.79.197.239.443: 1610731732 ack 236747780
3.155250 204.79.197.239.443 -> 13.13.13.1.52665: ack 1610731733
5.047264 13.13.13.1.52613 -> 20.185.212.106.443: 1421254032 ack 3784884456
5.126008 20.185.212.106.443 -> 13.13.13.1.52613: ack 1421254033
```

This output indicates that wan3 is used to steer traffic.

To verify the configuration when either wan1 or wan2 is restored, and traffic ceases to be steered through wan3:

1. Verify the SD-WAN service rules status:

```
# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x24200 use-shortcut-sla use-shortcut
 Tie break: cfg
  Gen(1), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(manual  hash-
mode=round-robin)
  Members(2):
    1: Seq_num(1 wan1 virtual-wan-link), alive, gid(1), selected
    2: Seq_num(2 wan2 virtual-wan-link), dead, gid(1)
  Src address(1):
        0.0.0.0-255.255.255.255

  Dst address(1):
        0.0.0.0-255.255.255.255
```

   This output indicates that wan1 is operational.

2. Verify the policy route list:

```
# diagnose firewall  proute list
list route policy info(vf=root):

id=2130706433(0x7f000001) vwl_service=1(test) vwl_mbr_seq=1 dscp_tag=0xfc 0xfc flags=0x10
load-balance has
h-mode=round-robin  tos=0x00 tos_mask=0x00 protocol=0 port=src(0->0):dst(0->0) iif=0(any)
path(1): oif=3(wan1) num_pass=0
source(1): 0.0.0.0-255.255.255.255
destination(1): 0.0.0.0-255.255.255.255
hit_count=986 last_used=2023-11-09 06:45:13
```

   This output indicates that wan1 is used to steer traffic.

Ensuring SD-WAN prioritizes healthy SD-WAN members

When using the SD-WAN rule option `set tie-break fib-best-match`, traffic that matches this rule will be sent through member(s) with the best FIB match for the destination. This can result in traffic taking unhealthy links.

When combined with the setting `set minimum-sla-meet-members 1`, SD-WAN rules will prioritize members that are in-SLA, and use the FIB to select the best path if there are multiple members within SLA.

Example use cases:

1. In multi-hub environments where each hub provides access to a resource, one hub may be preferred over the other. When both hubs are in-SLA, the FIB can be used to prefer one hub over the other. If the preferred hub is not in SLA, the secondary hub (though less preferred) will be used as it is in SLA.
2. When using ADVPN, if the shortcut tunnels and the spoke-to-hub tunnels are both in SLA, spoke-to-spoke traffic will prefer ADVPN shortcut tunnels as they will be the best according to the FIB. If the shortcut tunnels are out of SLA, the spoke-to-hub tunnel (though less preferred) will be used as it is in SLA.

```
config system sdwan
    ...
    config members
        edit 1
            set interface "port3"
            set zone "WAN1"
        next
        edit 2
            set interface "port4"
            set zone "WAN2"
        next
        edit 3
            set interface "HUB1-VPN1"
            set zone "HUB1"
            set cost 10
            ...
        next
        edit 4
            set interface "HUB1-VPN1-2"
            set zone "HUB1"
            set cost 11
            ...
        next
        edit 5
            set interface "HUB2-VPN1"
            set zone "HUB2"
            set cost 20
            ...
        next
        edit 6
            set interface "HUB2-VPN1-2"
            set zone "HUB2"
            set cost 21
            ...
        next
```

```
    end
    ...
    config service
        edit 1
            set name "HUB"
            set mode sla
            set minimum-sla-meet-members 1
            set dst "Datacenter"
            set src "all"
            config sla
                edit "HUB"
                    set id 1
                next
            end
            set priority-members 3 4 5 6
            set tie-break fib-best-match
        next
    end
end
```

SD-WAN Underlay Bandwidth and Quality Monitoring service

The SD-WAN Underlay Bandwidth and Quality Monitoring Service bundles a set of tools and services designed to enhance the experience of deploying SD-WAN on the FortiGate. The tools can be sorted into two categories:

- Application performance and configuration on page 1218
- Speed tests on page 1236

Application performance and configuration

Fortinet's SD-WAN solution includes several features designed to improve network visibility and simplify setup. The FortiGuard performance SLA database streamlines SLA creation by offering hundreds of preconfigured thresholds for popular SaaS and cloud services, helping reduce setup time and errors while optimizing application performance. Application Performance Monitoring provides end-to-end visibility into user experience by passively analyzing TCP metrics, enabling quick identification of network bottlenecks and issues. To facilitate rapid deployment and utilization of the above features, a guided underlay setup wizard walks users through provisioning steps using Fortinet best practices.

Service	Overview	License
Application performance monitoring	Provides passive monitoring of common TCP metrics for each application and calculates application-level network performance metrics over multiple traffic sessions.	Requires a valid SD-WAN Underlay and Application Monitoring license.

Service	Overview	License
SD-WAN Setup wizard	Step through the configurations needed to provision a basic SD-WAN setup, including interface, networking, performance SLA, and SD-WAN rule settings	Requires a valid SD-WAN Underlay and Application Monitoring license.
FortiGuard SLA database for SD-WAN performance SLA	Includes popular SaaS and Internet destinations, as well as recommended settings that you can select as probe servers for SD-WAN Performance SLA configurations	Requires a valid SD-WAN Underlay and Application Monitoring license.

Application performance monitoring

FortiOS includes application performance monitoring (APM), which passively monitors common TCP metrics for each application and calculates application-level network performance metrics over multiple traffic sessions.

By leveraging these APM metrics, administrators can:

- Pinpoint latency and find issues quicker and proactively.
- Troubleshoot application performance through detailed network metrics.
- Simplify configuration and management of applications.
- Monitor trends over time of business-critical end user applications within your FortiGate LAN environment.

Starting in FortiOS 7.6.4, an SD-WAN passive health check does not need to be configured for APM to work.

Considerations

- When `app-monitor` is enabled in a firewall policy, NPU offloading for the firewall policy is automatically disabled.
- You can display TCP metrics using the `diagnose sys session list` command, or you can view traffic logs in the CLI or GUI.
- SD-WAN traffic steering remains independent from the measured TCP session metrics.

The new metrics include:

Metric	Description
Server Response Time	Latency between the time a packet is sent from the origin and the time a response is sent back from the destination (milliseconds).
Network Transfer Time	Latency between the time a packet is sent to the origin and the time to get a response from the origin (milliseconds). Also known as TCP connection time.
Latency	Overall latency between origin and destination (milliseconds).
RTT Sample	Number of sessions used to determine server response time, network transfer time, and latency metrics.

Metric	Description
Origin Jitter	Jitter at the origin (milliseconds).
Reply Jitter	Jitter at the destination (milliseconds).
Jitter	Jitter between origin and destination (milliseconds).
Origin Packet Loss	Packet loss at the origin (percentage).
Reply Packet Loss	Packet loss at the destination (percentage).
Packet Loss	Overall packet loss between origin and destination (percentage).
Retransmission Sample	Number of sessions used to determine packet loss and retransmission metrics.
Origin Retransmission	Total number of retransmits that occurred in the origin to the destination traffic flow.
Reply Retransmission	Total number of retransmits that occurred in the destination to the origin traffic flow.
SYN Retransmission	Total number of SYN retransmits that occurred from the origin.
SYN-ACK Retransmission	Total number of SYN-ACK retransmits that occurred from the origin.
Origin Reset	Total number of origin resets that occurred.
Reply Reset	Total number of destination resets that occurred.

The main network metrics used to quantify application-level performance are latency, jitter, and packet loss.

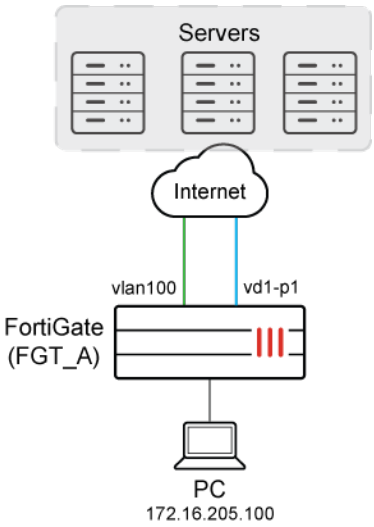
- Latency is the delay in seconds for packets to travel across the network, measured in milliseconds.
- Jitter is the variation in delay in seconds for packets to travel across the network, measured in milliseconds.
- Packet loss is the percentage of packets from the origin that fail to arrive at the destination when attempting to travel across the network.

To enable passive monitoring of applications:

```
config firewall policy
    edit <entry>
        set app-monitor enable
        set passive-wan-health-measurement enable
    next
end
```

Example

In this example, SD-WAN is configured with a zone named virtual-wan-link, and it contains two members (vlan100 and vd1-p1). A firewall policy is configured for the SD-WAN zonemetrics with application performance monitoring from the PC to a server.



To configure SD-WAN:

1. Configure SD-WAN:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "vd1-p1"
        next
        edit 2
            set interface "vlan100"
            set gateway 172.16.206.2
        next
    end
    config service
        edit 1
            set name "1"
            set dst "all"
            set src "172.16.205.0"
            set priority-members 1 2
        next
    end
end
```

2. Identify the preferred interface:
   In this example `vd1-p1` is the preferred SD-WAN member.

```
# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x4200 use-shortcut-sla use-shortcut
```

```
 Tie break: cfg
 Shortcut priority: 2
  Gen(1), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(manual)
  Members(2):
    1: Seq_num(1 vd1-p1 virtual-wan-link), alive, selected
    2: Seq_num(2 vlan100 virtual-wan-link), alive, selected
  Src address(1):
        172.16.205.0-172.16.205.255
  Dst address(1):
        0.0.0.0-255.255.255.255
```

3. Configure a firewall policy for the SD-WAN zone to monitor traffic from the PC:

   In this example, the `dstintf` option is set to the SD-WAN zone (`virtual-wan-link`), the `srcaddr` option identifies the PC (172.16.205.0), and application performance monitoring is enabled.

```
config firewall policy
    edit 1
        set name "APM"
        set srcintf "any"
        set dstintf "virtual-wan-link"
        set action accept
        set srcaddr "172.16.205.0"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set app-monitor enable
        set passive-wan-health-measurement enable
        set utm-status enable
        set ssl-ssh-profile "certificate-inspection"
        set application-list "g-default"
        set logtraffic all
        set auto-asic-offload disable
    next
end
```

4. As traffic passes from the PC through FortiGate to the server, TCP traffic is measured and logged, and you can view a session list:

```
# diagnose sys session list

session info: proto=6 proto_state=11 duration=172 expire=3577 timeout=3600 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ tun_id=172.16.209.2/0.0.0.0 vlan_cos=0/255
state=log may_dirty f00 f02 app_valid
statistic(bytes/packets/allow_err): org=59961/864/1 reply=2663311/2103/1 tuples=2
tx speed(Bps/kbps): 2/0 rx speed(Bps/kbps): 3/0
orgin->sink: org pre->post, reply pre->post dev=15->115/115->15
gwy=172.16.209.2/172.16.205.100
hook=pre dir=org act=noop 172.16.205.100:51128->172.16.202.2:22(0.0.0.0:0)
```

```
hook=post dir=reply act=noop 172.16.202.2:22->172.16.205.100:51128(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=1 pol_uuid_idx=843 auth_info=0 chk_client_info=0 vd=0
serial=00006eb8 tos=ff/ff app_list=6000 app=16060 url_cat=0
sdwan_mbr_seq=1 sdwan_service_id=1
rpdb_link_id=ff000001 ngfwid=n/a
tcp_srt=240 tcp_nrt=0 tcp_org_rtrs=17 tcp_rpl_rtrs=273 tcp_syn_rtrs=0 tcp_syn_ack_rtrs=0 tcp_
rst=00
npu_state=0x1041001 no_offload
no_ofld_reason:  disabled-by-policy non-npu-intf
total session: 1
```

5. View detailed application performance metrics in SD-WAN logs:

```
# execute log display

1: date=2025-03-06 time=09:40:33 eventtime=1741210833244790449 tz="+1200" logid="0113022941"
type="event" subtype="sdwan" level="information" vd="root" logdesc="SDWAN application
performance metrics via kernel" eventtype="Application Performance Metrics" appid=16091
interface="vd1-p1" serverresponsetime="162.0" networktransfertime="0.0" latency="162.0"
rttsample=5 originjitter="0" replyjitter="100" jitter="100.0" originpktloss="21.8"
replypktloss="2.6" packetloss="3.7" retransample=6 originretransmission=13
replyretransmission=17 synretransmission=0 synackretransmission=0 originreset=0 replyreset=0
msg="Application Performance Metrics via kernel"
```

To interpret this log: the application 16091 - Telnet is experiencing a latency of 162 ms on 5 session samples. Some packet losses were experienced in both origin and reply directions, leading to some retransmissions. Details are listed in the table.

Metric	Value
Server Response Time (ms)	162.0
Network Transfer Time (ms)	0.0
Latency (ms)	162.0
RTT Sample	5
Origin Jitter (ms)	0
Reply Jitter (ms)	100
Jitter (ms)	100.0
Origin Packet Loss (%)	21.8
Reply Packet Loss (%)	2.6
Packet Loss (%)	3.7
Retransmission Sample	6
Origin Retransmission	13
Reply Retransmission	17

Metric	Value
SYN Retransmission	0
SYN-ACK Retransmission	0
Origin Reset	0
Reply Reset	0

SD-WAN Setup wizard

The *SD-WAN Setup* wizard helps you configure the following settings for a simple SD-WAN setup:

- Interface
  The wizard supports a maximum of two interfaces.
- Networking
- Performance SLA
- SD-WAN Rule

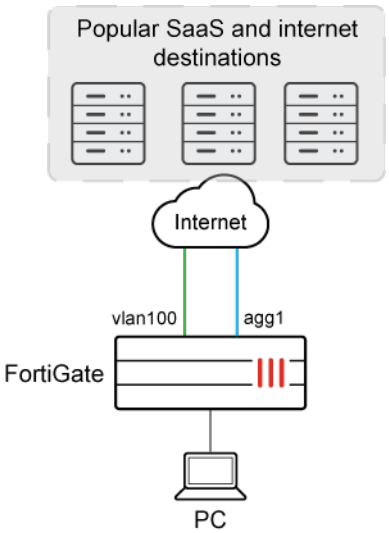
After completing the wizard, configure a default static route for the newly created SD-WAN interface.

FortiGate requires a valid SD-WAN Underlay and Application Monitoring license before the SD-WAN Setup wizard is visible.

See also FortiGuard SLA database for SD-WAN performance SLA on page 1231.

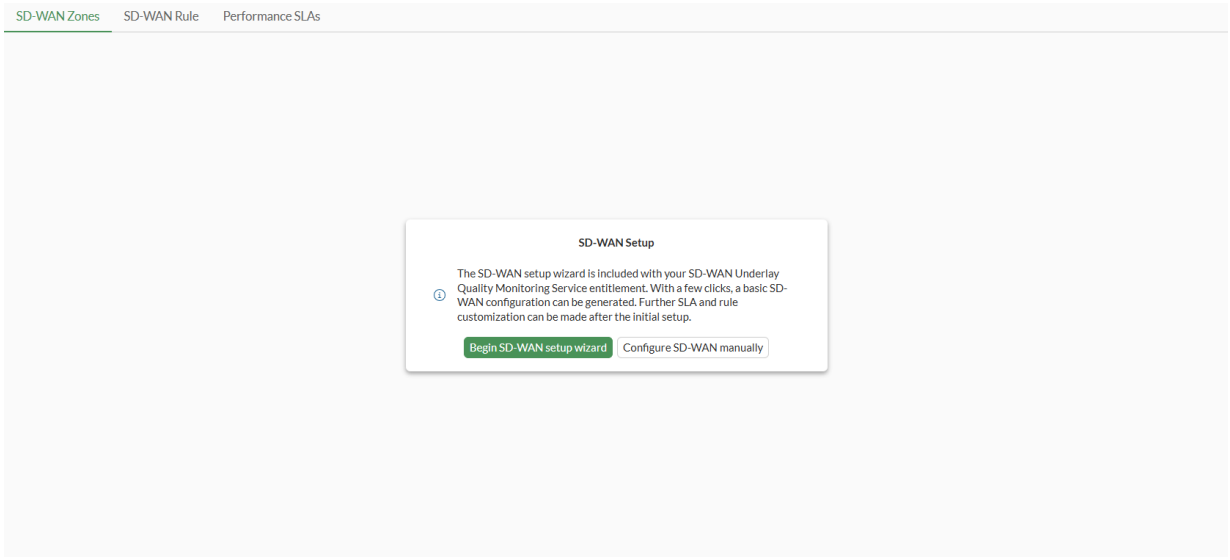
Example

This example describes how to use the SD-WAN Setup wizard to create an SD-WAN configuration with one SD-WAN zone (named *Test*) and two SD-WAN members (*agg1* and *vlan100*). The wizard also guides you to complete the networking, performance SLA, and SD-WAN rule. After the wizard completes, configure a default static route for the SD-WAN interface.

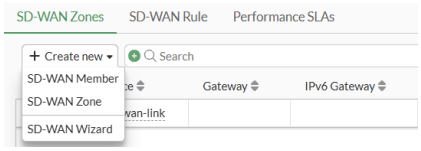


To use the SD-WAN Setup wizard to configure SD-WAN:

1. Go to the *Network* > *SD-WAN* > *SD-WAN Zones* page to access the wizard:
   - When no SD-WAN configuration exists, the following message is displayed. Click *Begin SD-WAN setup wizard* to access the wizard.



   - When an SD-WAN configuration exists, click *Create New* > *SD-WAN Wizard* to access the wizard.

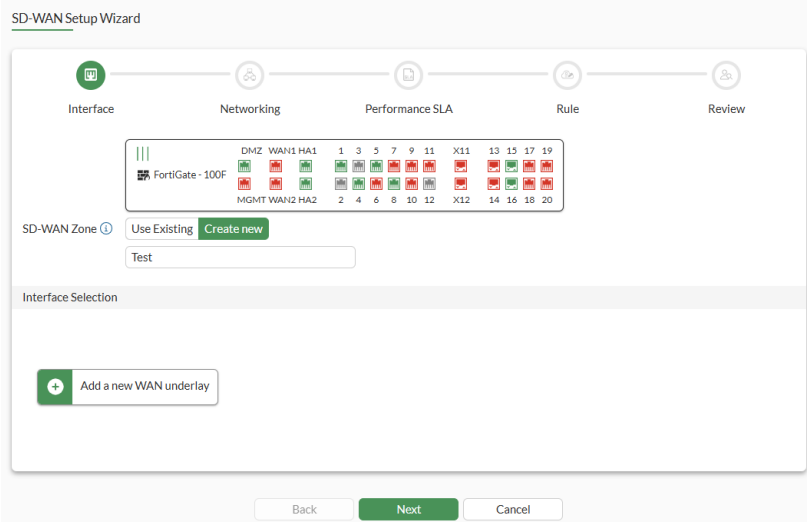


   The *SD-WAN Setup* wizard opens on the *Interface* step.
2. For the *Interface* step, identify a zone, and select one or two interfaces for the underlay:

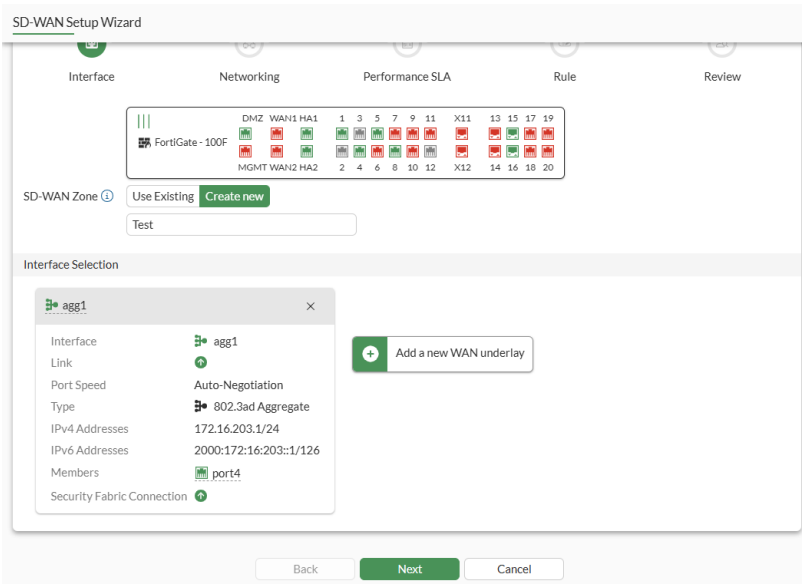
   The selected interfaces become members of the SD-WAN zone. This example creates a zone named *Test* and uses two interfaces (*agg1* and *vlan100*).

   a. Set *SD-WAN Zone* to:
      - *Use Existing* and select an existing SD-WAN zone.

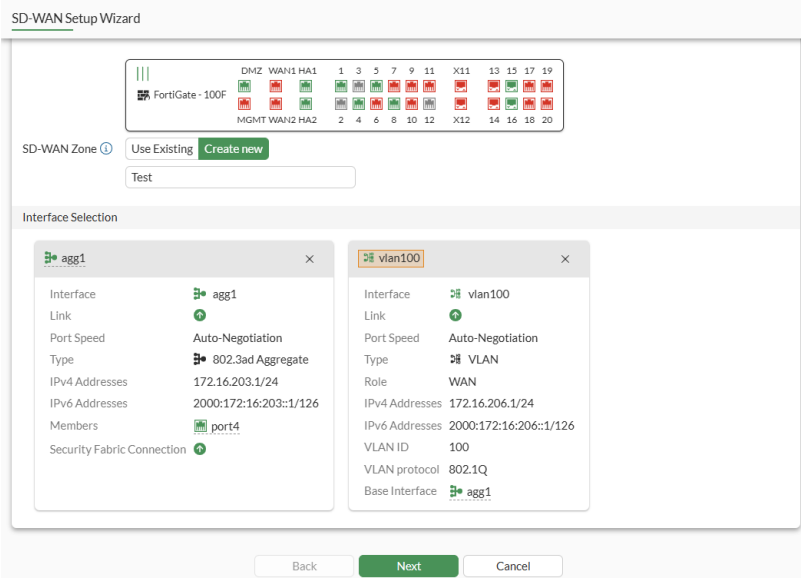
        Or:
      - *Create New* and type a name for the new SD-WAN zone. In this example, a new zone named *Test* is created.



b. Click *Add a new WAN underlay*, select an interface, and click *Apply*.



c. Click *Add a new WAN underlay* again, select an interface, and click *Apply*.

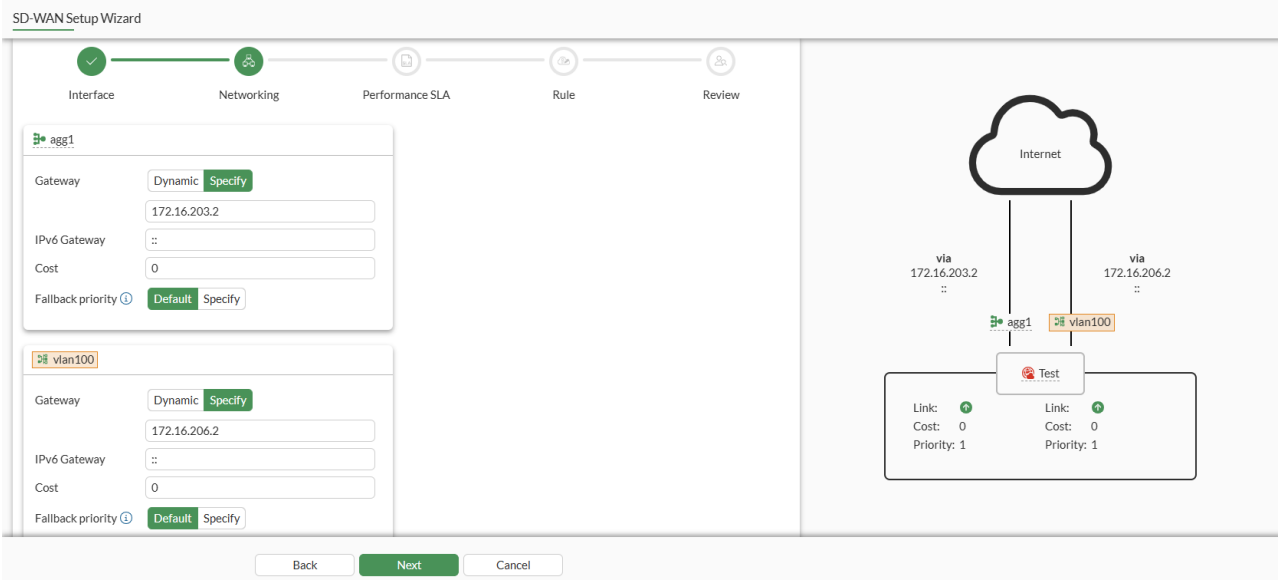


You have added two interfaces to the SD-WAN zone.

d. Click *Next* to proceed to the *Networking* step.

3. For the *Networking* step, set the gateway and priority for each interface, and click *Next*:

Gateway	Select one of the following methods to assign a gateway IP address to the interface: • *Dynamic*: Select to leave the gateway undefined and proceed to the next screen. Can be used with interfaces set to use DHCP as a client. • *Specify*: Select to specify an IPv4 or IPv6 address for the gateway.
Cost	Used by the lowest-cost SLA strategy. The link with the lowest cost is chosen to pass traffic. The lowest possible cost is 0
Fallback priority	Select one of the following methods to define the priority of SD-WAN members: • *Default*: Select to use the default, which is the same priority for all SD-WAN members. • *Specify*: Select to specify a priority for the SD-WAN member in the *Priority* box. The priority is used in the static route created for the SD-WAN member interface and in SD-WAN rules (including the implicit rule). When priority is used to determine the best route, the lower value takes precedence.
Priority	Available when *Fallback priority* is set to *Specify*. Specify a priority for the SD-WAN member (1 - 65535, default = 1).

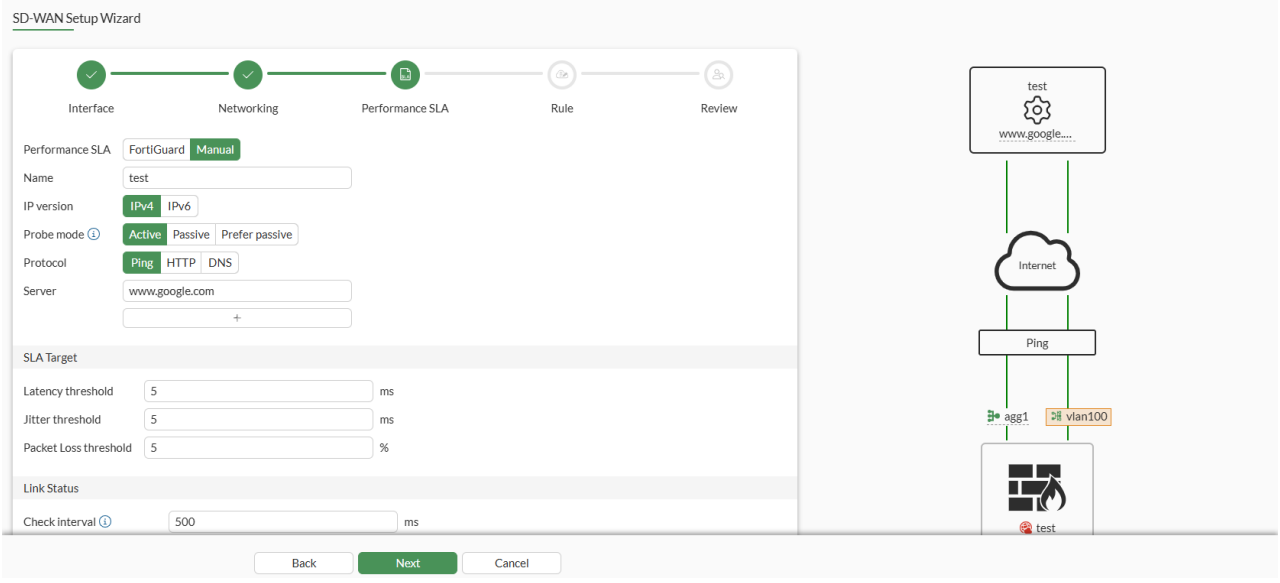


The wizard moves to the *Performance SLA* step.

4. For the *Performance SLA* step, configure health-check for SD-WAN members, and click *Next*:

Only the fields that you must set before you can proceed are described.

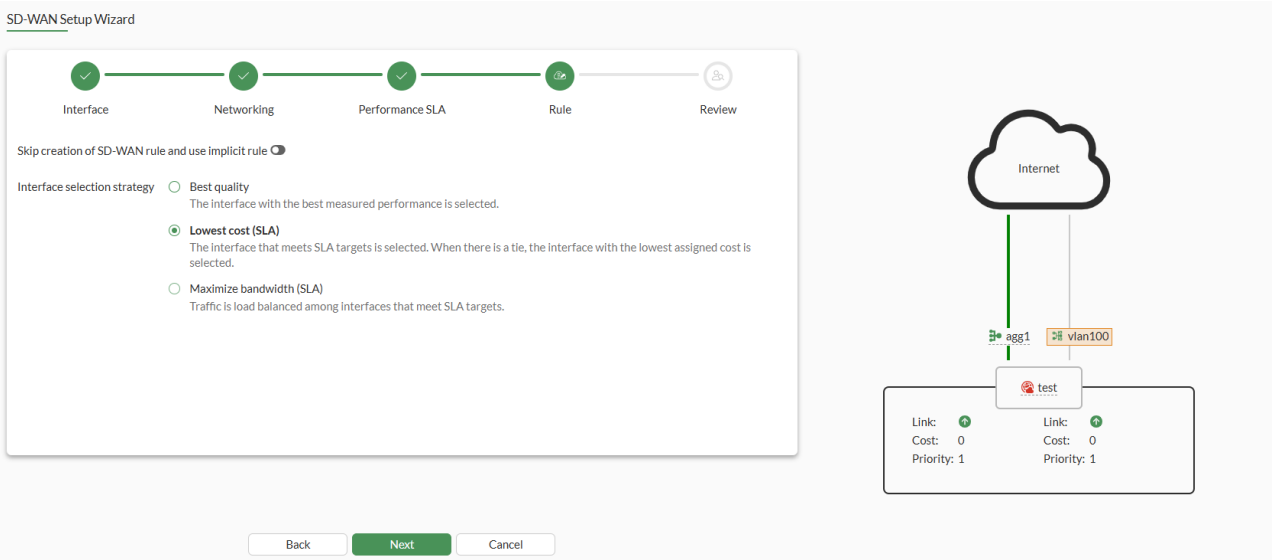
Performance SLA	Select one of the following to choose how to define performance SLA: • *FortiGuard*: Select to use the FortiGuard SLA database. Select a predefined server from the *Server* list, and specify the protocol to use. • *Manual*: Select to manually define a server for the SLA.



The wizard moves to the *Rule* step.

5. For the *Rule* step, create a service rule, and click *Next*:

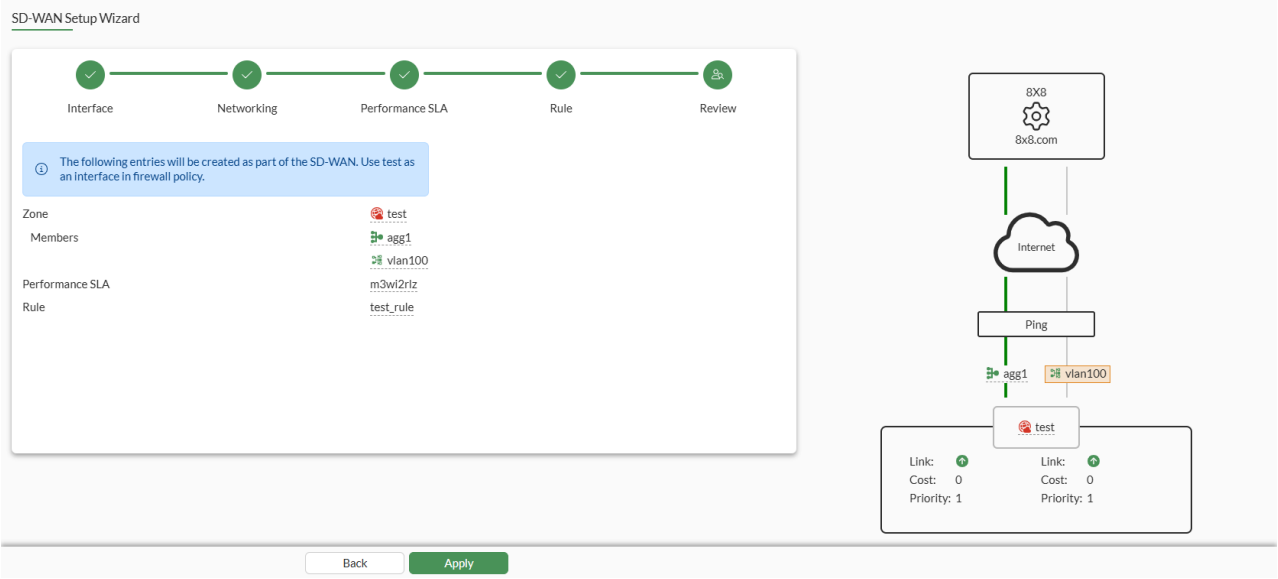
Skip creation of SD-WAN rule and use implicit rule	Enable to use the implicit rule instead of creating an SD-WAN rule.
Interface selection strategy	Available when *Skip creation of SD-WAN rule and use implicit rule* is disabled. Specify how SD-WAN should select an interface: • *Best quality*: Select to use the interface with the best measured performance. • *Lowest cost (SLA)*: Select to use the interface that meets the defined performance SLA targets. When a tie occurs, the interface with the lowest assigned cost is selected. • *Maximize bandwidth*: Traffic is load balanced among interfaces that meet SLA targets.



The wizard moves to the *Review* step.

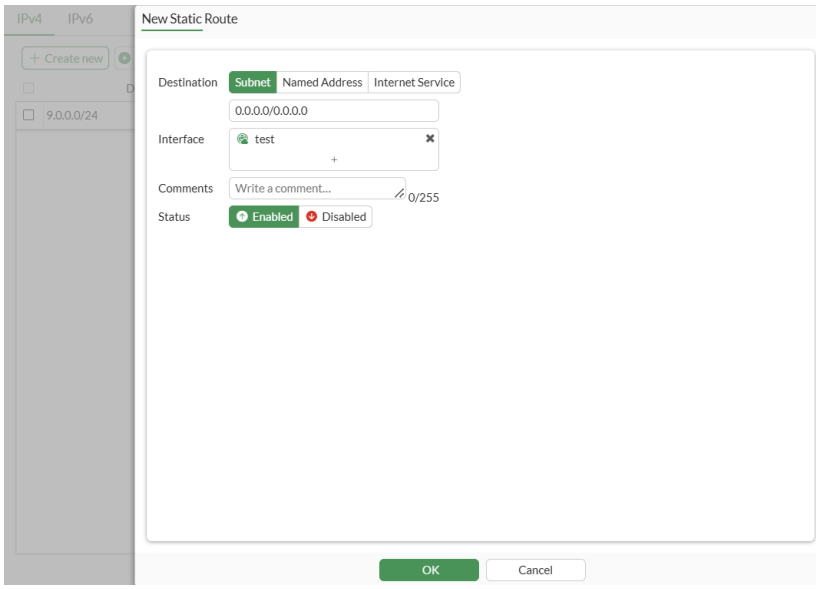
6. For the *Review* step, review the entries, and click *Apply* to create them:

Zone	Displays the name of the SD-WAN zone that will be created.
Members	Displays the name of the interface members that will be added to the SD-WAN zone.
Performance SLA	Displays the name of the performance SLA configuration that will be used for the SD-WAN configuration.
Rule	Displays the name of the SD-WAN rule that will be used for the SD-WAN configuration

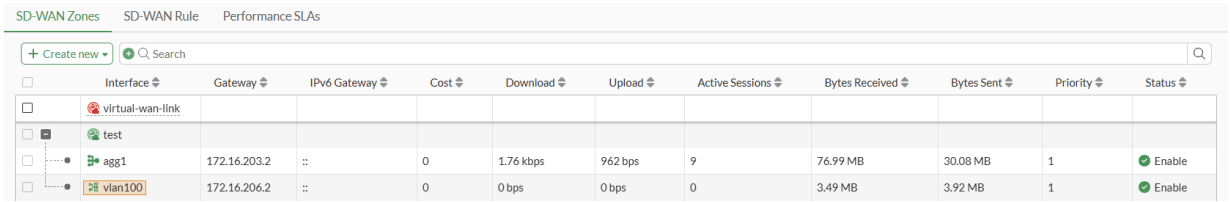


The entries are created, and the wizard completes.

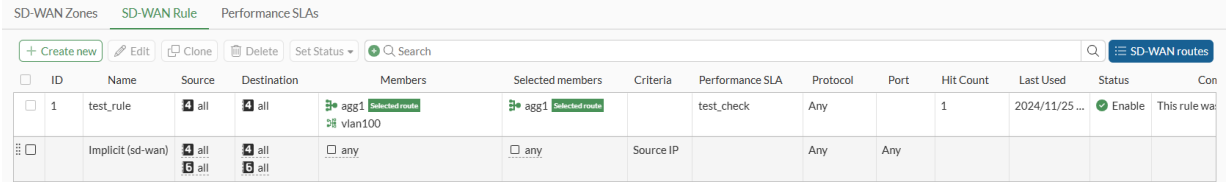
7. Create a static route for the SD-WAN interface (that is the SD-WAN zone):
   a. Go to *Network* > *Static Routes*, and click *Create new*.
   b. Complete the options, and click *OK*.



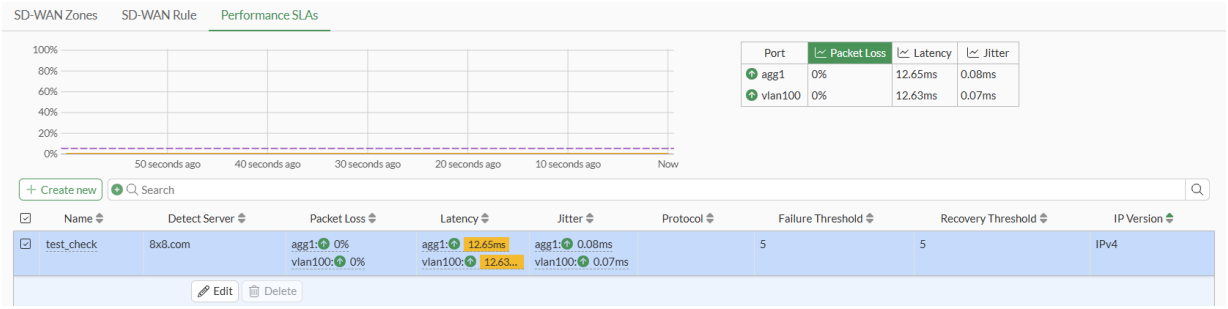
8. Review the SD-WAN configuration:
   a. Go to *Network* > *SD-WAN* > *SD-WAN Zones*, and view the zone and members that you created.



   b. On the *SD-WAN Rule* tab, view the rule that you created.



c. On the *Performance SLAs* tab, view the SLA configuration that you created.



FortiGuard SLA database for SD-WAN performance SLA

A FortiGuard SLA database includes popular SaaS and Internet destinations as well as recommended settings that you can select as probe servers for SD-WAN Performance SLA configurations in the GUI or CLI.

In the GUI, go to *Network* > *SD-WAN* > *Performance SLA*, and click *Create New* to access the new database.

In the CLI, use the following options:

```
config system sdwan
    config health-check
        edit <health-check name>
            set fortiguard {enable | disable}
            set fortiguard-name <string>
        next
    end
end
```

`fortiguard {enable | disable}`	Enable/disable use of FortiGuard SLA database.
`fortiguard-name <string>`	Name of the predefined health-check target from the FortiGuard SLA database.

The FortiGate requires a valid SD-WAN Underlay and Application Monitoring license before the FortiGuard SLA Database can be downloaded or updated.

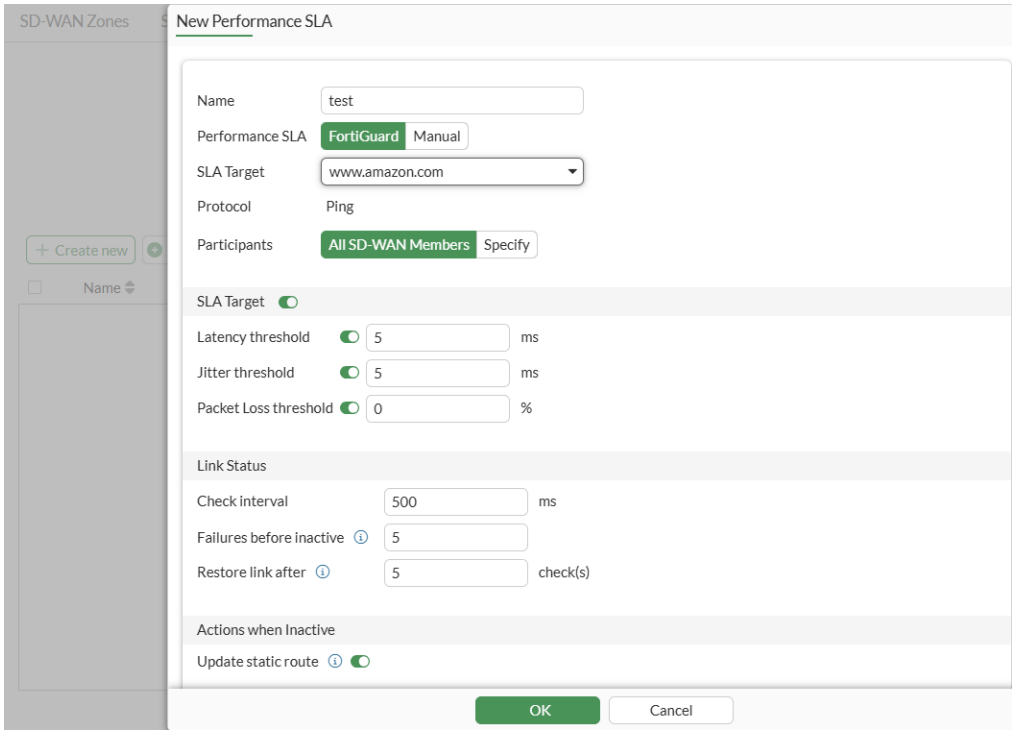
See also SD-WAN Setup wizard on page 1224.

Example

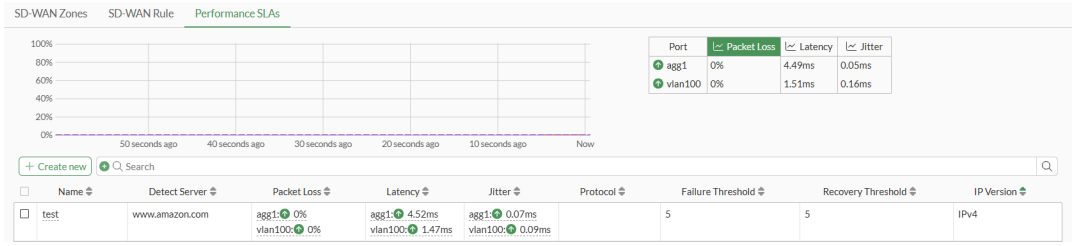
In this example, an SD-WAN performance SLA is configured to use the FortiGuard SLA database and its Amazon target.

To configure a performance SLA in the GUI:

1. Go to *Network* > *SD-WAN* > *Performance SLA*, and click *Create New*. The *New Performance SLA* pane is displayed.
2. Set *Performance SLA* to *FortiGuard* to select the database, and set *SLA Target* to the *www.amazon.com* target from the database.



3. Complete the remaining options, and click *OK*. The configuration is displayed on the *Performance SLAs* pane.
4. On the *Performance SLAs* pane, select the configuration to view the health-check status.



To configure performance SLA in the CLI:

1. Configure an SD-WAN health-check to use the SLA database and its Amazon target:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
```

```
    config members
        edit 1
            set interface "agg1"
            set gateway 172.16.203.2
        next
        edit 2
            set interface "vlan100"
            set gateway 172.16.206.2
        next
    end
    config health-check
        edit "test"
            set fortiguard enable
            set fortiguard-name "Amazon"
            set server "www.amazon.com"
            set members 0
            config sla
                edit 1
                next
            end
        next
    end
end
```

2. Check the health status:
   In this example, the SLA database is enabled and Amazon is configured.

```
# diagnose sys sdwan  health-check
Health Check(test):
Seq(1 agg1): state(alive), packet-loss(1.000%), latency(55.557), jitter(1.245), mos(4.373),
bandwidth-up(999993), bandwidth-dw(999982), bandwidth-bi(1999975), sla_map=0x0
Seq(2 vlan100): state(alive), packet-loss(4.000%), latency(55.534), jitter(1.211), mos(4.372),
bandwidth-up(697383), bandwidth-dw(437492), bandwidth-bi(1134875), sla_map=0x0
```

To view the performance SLA database in the CLI:

1. View the SLA database version:

```
# diagnose autoupdate version

...

SLA Database
---------
Version: 1.00003
Contract Expiry Date: Wed Apr 30 2025
Last Updated using scheduled update on Mon Nov 25 09:46:47 2024
Last Update Attempt: Wed Nov 27 14:36:01 2024
Result: No Updates

Timezone Database
---------
```

```
Version: 1.0006

...
```

2. List the targets predefined by FortiGuard in the SLA database:

```
# diagnose sladb target-list
target-name:8X8
deprecated:0
sz_domain:6

target-name:ADP
deprecated:0
sz_domain:5

target-name:AOL
deprecated:0
sz_domain:9

target-name:AWS dynamodb
deprecated:0
sz_domain:27

target-name:AWS ec2
deprecated:0
sz_domain:27

target-name:AWS ecs
deprecated:0
sz_domain:27

target-name:AWS es
deprecated:0
sz_domain:27

target-name:AWS lambda
deprecated:0
sz_domain:27
...
```

3. List the domains under a specific target predefined by FortiGuard in the SLA database:

```
# diagnose sladb  domain-list ADP
domain-name:www.adp.com
desc:ADP (www.adp.com)
deprecated:0
sz_protocol:2

domain-name:ipay.adp.com
desc:Online payroll management and payment platform.
deprecated:0
sz_protocol:2
```

```
domain-name:workforcenow.adp.com
desc:Human resource management platform.
deprecated:0
sz_protocol:2

domain-name:globalview.adp.com
desc:Global HR management platform.
deprecated:0
sz_protocol:2

domain-name:mobile.adp.com
desc:Mobile app for ADP services.
deprecated:0
sz_protocol:2
```

4. List the protocols under a specific target and domain predefined by FortiGuard in the SLA database:

```
# diagnose sladb protocol-list ADP www.adp.com
target-name:ADP
domain-name:www.adp.com

        protocol: ping
        protocol: https
```

5. View the communication method between FortiGate and servers predefined by FortiGuard for SD-WAN health-checks.

```
# show system health-check-fortiguard
config system health-check-fortiguard
    edit "8X8"
        set server "www.8x8.com"
        set protocol https
    next
    edit "ADP"
        set server "www.adp.com"
    next
    edit "AOL"
        set server "www.aol.com"
    next
    edit "AWS dynamodb"
        set server "dynamodb.me-central-1.amazonaws.com"
    next
    edit "AWS ec2"
        set server "ec2.us-east-1.amazonaws.com"
    next
    edit "AWS ecs"
        set server "ecs.me-central-1.amazonaws.com"
    next
    edit "AWS es"
        set server "es.us-east-1.amazonaws.com"
    next
    edit "AWS lambda"
```

```
        set server "lambda.us-east-1.amazonaws.com"
    next
...
```

Speed tests

Speed tests can be conducted either on-demand or according to a predetermined schedule, measuring upload and download speeds of up to 1 Gbps. The results of the tests can be used as reference for various applications, including the following:

- Configuring the estimated bandwidth of an interface, which can be employed in conjunction with various WAN intelligence strategies. See Using speed test results with SD-WAN on page 1267 for more information.
- Configuring the inbandwidth and outbandwidth of an interface for use in traffic shaping. See Using speed test results with traffic shaping on page 1269.
- Applying the speed test to dialup VPN tunnels in a hub and spoke deployment to conduct traffic shaping.

FortiOS offers a variety of methods for testing SD-WAN speed. The following table provides a brief overview of each method and guidance on when it might be most advantageous to use one method over the others.

Service	Overview	License
CLI speed test on page 1238	• Provides the most flexibility and options, which enables the speed test to operate with user-defined parameters. • Results can be used as reference to manually add to the interface's estimated bandwidth, or inbandwidth and outbandwidth. • Server is on the cloud, which is maintained by Fortinet. The nearest cloud servers are selected as the default server group to perform speed-test. • If the speed-test against a cloud server fails due to some issues, FortiGate will repeat the test in 5 minute intervals until it is successful or 30 minutes has elapsed.	Requires a valid SD-WAN Underlay and Application Monitoring license.
GUI speed test on page 1239	• Downloads the speed test server list automatically. • Results can be added to the interface's estimated bandwidth with one click. • Results are automatically updated in the interface `measured-upstream-bandwidth` and `measured-downstream-bandwidth` fields. • Results can be used as a reference to manually configure an interface's inbandwidth and outbandwidth. • Easier to use.	Requires a valid SD-WAN Underlay and Application Monitoring license.

Service	Overview	License
	• Server is on the cloud, which is maintained by Fortinet. The nearest cloud servers are selected as the default server group to perform speed-test. • If the speed-test against a cloud server fails due to some issues, FortiGate will repeat the test in 5 minute intervals until it is successful or 30 minutes has elapsed.	
Scheduled interface speed test on page 1240	• Speed tests can be scheduled to run automatically. • Results are automatically updated in the interface `measured-upstream-bandwidth` and `measured-downstream-bandwidth` fields. • Results can be automatically applied to an interface's inbandwidth and outbandwidth, enabling dynamic QoS application. Minimum and maximum values may be applied to overwrite these results. • Interface inbandwidth and outbandwidth shaping will always be bypassed during testing. If traffic shaping during the test is desired, `update-inbandwidth-maximum` and/or `update-outbandwidth-maximum` can be configured. • Server is on the cloud, which is maintained by Fortinet. The nearest cloud servers are selected as the default server group to perform speed-test. • If the speed-test against a cloud server fails due to some issues, FortiGate will repeat the test in 60 second intervals for 5 retries.	Requires a valid SD-WAN Underlay and Application Monitoring license.
Speed test from hub to spoke	• Server is the spoke. • Tests initiated from the hub. • Results are cached for future use. • Results can be dynamically applied to the dialup tunnel for egress traffic shaping. • Results can be used as a reference to manually configure an interface's inbandwidth and outbandwidth.	License not required.
Speed test from spokes to hub	• Server is the hub. • Tests initiated from spokes, even when a spoke is behind a NAT device. • Results are cached on the spoke for future use and sent to the hub.	License not required.

Service	Overview	License
	• Results can be dynamically applied to the dialup tunnel for egress traffic shaping. • Results can be used as a reference to manually configure an interface's inbandwidth and outbandwidth.	

CLI speed test

The speed test tool is compatible with iPerf3.6 with SSL support. It can test the upload bandwidth to the FortiGate Cloud speed test service. It can initiate the server connection and send download requests to the server. The tool can be run up to 10 times a day.

The FortiGate downloads the speed test server list. The list expires after 24 hours. One of the speed test servers is selected based on user input.

To configure the speed test settings:

```
config system speed-test-setting
    set latency-threshold <integer>
    set multiple-tcp-stream <integer>
end
```

Command	Description
`latency-threshold <integer>`	Set the speed test threshold for the auto mode, in milliseconds (0 - 2000, default = 60). If the latency exceeds this threshold, the speed test will use the UDP protocol; otherwise, it will use the TCP protocol.
`multiple-tcp-stream <integer>`	Set the number of parallel client streams for the TCP protocol to run during the speed test (1 - 64, default = 4).

To download the server list of speed tests:

1. Download the server list from FortiCloud:

```
# execute speed-test-server download
Download completed.
```

2. Verify the list:

```
# execute speed-test-server list
...
FTNT_CA_Toronto valid
        Host: 154.52.23.67 5200 fortinet
        ...
FTNT_CA_Vancouver valid
        Host: 154.52.20.6 5200 fortinet
        ...
```

```
FTNT_Global valid
        Host: 154.52.6.95 5203 fortinet
        ...
```

To run the speed test:

A speed test can be run with or without specifying a server. The system will automatically choose one server from the list and run the speed test. The test results are shown in the command terminal.

```
# execute speed-test <interface> <server> {Auto | TCP | UDP}
```

```
# diagnose netlink interface speed-test <interface> <server> {Auto | TCP | UDP}
```

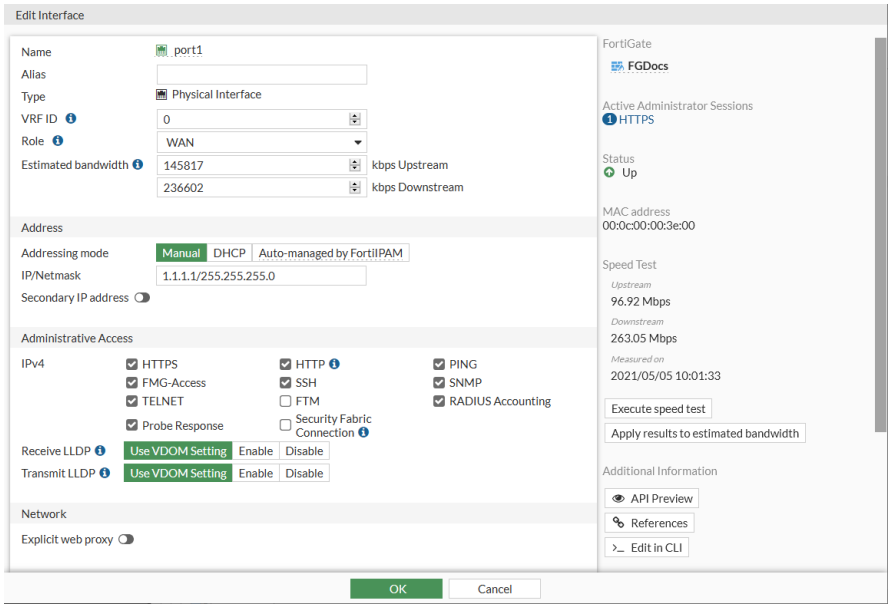
See Speed test examples on page 1269 for a sample configurations.

GUI speed test

An interface speed test can be manually performed on WAN interfaces in the GUI. The results of the test can be added to the interface's *Estimated bandwidth*. The estimated upstream and downstream bandwidths can be used in SD-WAN service rules to determine the best link to use when either load balancing or best quality strategies are selected.

To run an interface speed test on a WAN interface:

1. Go to *Network* > *Interfaces*.
2. Edit a WAN interface. The interfaces can be grouped by role using the grouping dropdown on the right side of the toolbar.
3. Click *Execute speed test* in the right pane.



   The CLI equivalent to running a speed test in the GUI is `execute speed-test <interface>`.
4. When the test completes, click *OK* in the *Confirm* pane to apply the results to the estimated bandwidth.

The results can also be applied later by clicking *Apply results to estimated bandwidth*.

The speed test results are used to populate the *Estimated bandwidth* fields.

5. Click *OK*.

The FortiGate must be connected to FortiGuard, and able to reach either the AWS or Google speed test servers.

Scheduled interface speed test

The SD-WAN Underlay and Quality Monitoring service supports running a speed test based on a schedule. The test results are automatically updated in the interface `measured-upstream-bandwidth` and `measured-downstream-bandwidth` fields. These fields do not impact the interface inbound bandwidth, outbound bandwidth, estimated upstream bandwidth, or estimated downstream bandwidth settings.

When the scheduled speed tests run, interface inbandwidth and outbandwidth shaping will be bypassed. If traffic shaping during the test is desired, `update-inbandwidth-maximum` and/or `update-outbandwidth-maximum` can be configured.

```
config system speed-test-schedule
    edit <interface>
        set status <enable* | disable>
        set diffserv <binary>
        set server-name <name>
        set legacy-server-mode <disable* | enable>
        set mode {Auto | TCP | UDP}
        set schedules <schedule> ...
        set update-bandwidth-limit-unit <value* | percentage>
        set update-inbandwidth-maximum <integer>
        set update-inbandwidth-minimum <integer>
        set update-outbandwidth-maximum <integer>
        set update-outbandwidth-minimum <integer>
        set inbandwidth-source <default* | measured>
        set outbandwidth-source <default* | measured>
    next
end
```

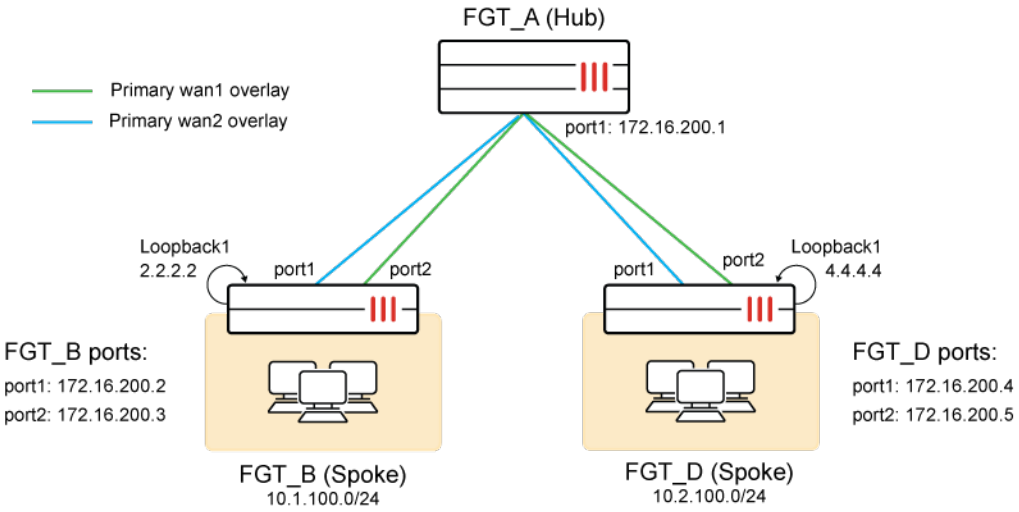
Command	Description
`status <enable* | disable>`	Enable/disable scheduled speed test.
`diffserv <binary>`	DSCP used for speed test (000000 - 111111).
`server-name <name>`	Select the speed-test server from a list. Use `show system speed-test-server` to display the list.
`legacy-server-mode <disable* | enable>`	Enable/disable legacy server selection mode.
`mode {Auto | TCP | UDP}`	Set the protocol used for the speed test: • `Auto` (default): dynamically select TCP or UDP based on the speed test setting.

Command	Description
	• UDP: use UDP. • TCP: use TCP.
schedules <schedule_name>	Select the name of a firewall recurring schedule.
update-bandwidth-limit-unit <value* | percentage>	Set the update bandwidth limits by values in kbps or percentages of the interface's bandwidth.
update-inbandwidth-maximum <integer>	Maximum downloading bandwidth (kbps) or percentage of the interface's inbandwidth to be used in shaping.
update-inbandwidth-minimum <integer>	Minimum downloading bandwidth (kbps) or percentage of the interface's inbandwidth to be used in shaping.
update-outbandwidth-maximum <integer>	Maximum uploading bandwidth (kbps) or percentage of the interface's outbandwidth to be used in shaping.
update-outbandwidth-minimum <integer>	Minimum uploading bandwidth (kbps) or percentage of the interface's outbandwidth to be used in shaping.
inbandwidth-source <default* | measured>	Determine which inbandwidth values to use for setting shaper: • default: Use manually configured bandwidth as inbandwidth source. • measured: Use speedtest measured bandwidth as inbandwidth source.
outbandwidth-source <default* | measured>	Determine which outbandwidth values to use for setting shaper: • default: Use manually configured bandwidth as outbandwidth source. • measured: Use speedtest measured bandwidth as outbandwidth source.

See Example 4 for a sample configuration.

Running speed tests from the hub to the spokes in dial-up IPsec tunnels

In this example, the hub is configured as a VPN dial-up server, and both of the spokes are connected to the hub. It is assumed that the VPN configuration is already done, with a dynamic gateway type and kernel device creation (net-device) disabled. Speed tests are configured on only one interface for the branch devices. Speed tests may be configured for all underlay interfaces, but care must be taken to avoid tests running at the same time when utilizing speedtest-bypass-routing to ensure there remains a viable path from Hub to Spokes.



The VPN interfaces and IP addresses are:

FortiGate	Interface	IP Address
FGT_A (Hub)	hub-phase1 hub-phase11	10.10.100.254 10.10.100.253
FGT_B (Spoke)	spoke11-p1 spoke12-p1	10.10.100.2 10.10.100.3
FGT_D (Spoke)	spoke21-p1 spoke22-p1	10.10.100.4 10.10.100.5

A recurring speed test is configured that runs on the hub over the dial-up interfaces. The speed tests are performed over the underlay interface from the hub to the spoke. Each spoke is configured to operate as a speed test server and to allow the speed test to run on its underlay interface. The spokes establish a BGP peering with the hub (can be BGP per overlay or BGP on loopback) and advertises its loopback network to the hub. The specific configuration is only shown for FGT_B.

To configure the spoke FortiGates (FGT_B) as a speed test server:

1. Enable a speed test server with custom speed-test listening ports:
   A speed test server is enabled on the hub. Port 7000 will run speed tests, and port 6000 will be the controller used to issue access tokens for speed test authentication.

```
config system global
     set speedtest-server enable
     set speedtestd-ctrl-port 6000
     set speedtestd-server-port 7000
end
```

2. Allow speed tests on one underlay:

```
config system interface
    edit "port1"
```

```
        append allowaccess speed-test
    next
end
```

3. Allow speed tests on one overlay:

```
config system interface
    edit "hub-spoke11-p1"
        set allowaccess ping speed-test
        ...
        set interface "port1"
    next
end
```

4. Configure SD-WAN with bypass routing enabled for speed tests on member *spoke11-p1*:

```
config system sdwan
    set speedtest-bypass-routing enable
    config members
        edit 1
            set interface "spoke11-p1"
        next
        edit 2
            set interface "spoke12-p1"
        next
    end
    config neighbor
        edit "10.10.100.254"
            set member 1
            set mode speedtest
        next
       edit "10.10.100.253"
            set member 2
        next

    end
end
```

5. Configure BGP routing:

```
config router route-map
    edit "No_Speed-Test"
        config rule
            edit 1
                set action permit
            next
        end
    next
    edit "Start_Speed-Test"
        config rule
            edit 1
                set action deny
```

```
            next
        end
    next
end
config router bgp
    set as 65412
    config neighbor
        edit "10.10.100.254"
            set advertisement-interval 1
            set remote-as 65412
            set route-map-out "Start_Speed-Test"
            set route-map-out-preferable "No_Speed-Test"
        next
    end
    config network
        edit 1
            set prefix 2.2.2.2 255.255.255.255
        next
        edit 2
            set prefix 10.1.100.0 255.255.255.0
        next
    end
end
```

Results

1. Before the speed test starts, FGT_A can receive the route from FGT_B by BGP from both overlays:

```
# get router info routing-table bgp
Routing table for VRF=0
B       2.2.2.2/32 [200/0] via 10.10.100.2 (recursive via 172.16.200.2, hub-phase1), 00:00:10
                           [200/0] via 10.10.100.3 (recursive via 172.16.200.3, hub-phase11), 00:00:10
B       10.1.100.0/24 [200/0] via 10.10.100.2 (recursive via 172.16.200.2, hub-phase1),
00:00:10
                      [200/0] via 10.10.100.3 (recursive via 172.16.200.3, hub-phase11),
00:00:10
```

2. At the scheduled time, the speed test starts for the hub-phase1 interface from hub to spoke:

```
# diagnose test application forticron 9
Speed test schedules:
    Interface     Server     Update    Up/Down-limit (kbps)               Days        H:M
TOS     Schedule
-------------------------------------------------------------------------------------------------
-----------------------------
    hub-phase1    dynamic                                                1111111    14:41
0x00    speedtest_recurring
Active schedules:
        64002f: hub-phase1(port1) 172.16.200.2     hub-phase1_1
        64002e: hub-phase1(port1) 172.16.200.4     hub-phase1_0
```

The `diagnose debug application speedtest -1` command can be used on both the hub and spokes to check the speed test execution.

3. While the speed test is running, FGT_A does not receive the route from FGT_B over that interface by BGP:

```
#  get router info routing-table bgp
Routing table for VRF=0
B       2.2.2.2/32 [200/0] via 10.10.100.3 (recursive via 172.16.200.3, hub-phase11), 00:00:20
B       10.1.100.0/24 [200/0] via 10.10.100.3 (recursive via 172.16.200.3, hub-phase11),
00:00:20
```

4. Speed tests results can be dynamically applied to the dial-up tunnel for egress traffic shaping:

```
# diagnose vpn tunnel list
------------------------------------------------------
name=hub-phase1_0 ver=2 serial=c 172.16.200.1:0->172.16.200.4:0 tun_id=172.16.200.4 dst_
mtu=1500 dpd-link=on remote_location=0.0.0.0 weight=1
...
egress traffic control:
        bandwidth=737210(kbps) lock_hit=0 default_class=2 n_active_class=3
        class-id=2      allocated-bandwidth=73720(kbps)         guaranteed-bandwidth=73720
(kbps)
                        max-bandwidth=73720(kbps)       current-bandwidth=0(kbps)
                        priority=low    forwarded_bytes=52
                        dropped_packets=0       dropped_bytes=0
        class-id=3      allocated-bandwidth=221163(kbps)        guaranteed-bandwidth=221162
(kbps)
                        max-bandwidth=294883(kbps)      current-bandwidth=0(kbps)
                        priority=medium         forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
        class-id=4      allocated-bandwidth=442325(kbps)        guaranteed-bandwidth=147441
(kbps)
                        max-bandwidth=442325(kbps)      current-bandwidth=0(kbps)
                        priority=high   forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
------------------------------------------------------
name=hub-phase1_1 ver=2 serial=d 172.16.200.1:0->172.16.200.2:0 tun_id=172.16.200.2 dst_
mtu=1500 dpd-link=on remote_location=0.0.0.0 weight=1
...
egress traffic control:
        bandwidth=726813(kbps) lock_hit=0 default_class=2 n_active_class=3
        class-id=2      allocated-bandwidth=72681(kbps)         guaranteed-bandwidth=72681
(kbps)
                        max-bandwidth=72681(kbps)       current-bandwidth=0(kbps)
                        priority=low    forwarded_bytes=123
                        dropped_packets=0       dropped_bytes=0
        class-id=3      allocated-bandwidth=218044(kbps)        guaranteed-bandwidth=218043
(kbps)
                        max-bandwidth=290725(kbps)      current-bandwidth=0(kbps)
                        priority=medium         forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
        class-id=4      allocated-bandwidth=436087(kbps)        guaranteed-bandwidth=145362
(kbps)
                        max-bandwidth=436087(kbps)      current-bandwidth=0(kbps)
                        priority=high   forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
```

5. Speed test results can be cached, indexed, and written to disk:

```
# diagnose test application forticron 10
Speed test results:
1: vdom=root, phase1intf=hub-phase1, peer-id='spoke11-p1', bandwidth=737210, last_
log=1624226603
2: vdom=root, phase1intf=hub-phase1, peer-id='spoke21-p1', bandwidth=726813, last_
log=1624226614
```

```
# diagnose test application forticron 11
Write 2 logs to disk.
```

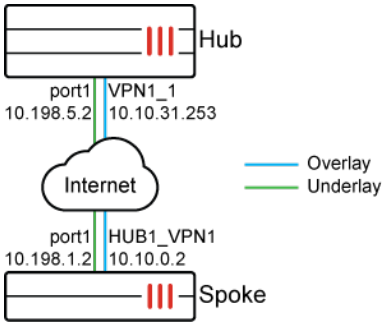
```
# diagnose test application forticron 12
load 2 results.
```

Disable then reenable the IPsec VPN tunnel and the cached speed test results can be applied to the tunnel again:

```
# diagnose vpn tunnel list
------------------------------------------------------
name=hub-phase1_0 ver=2 serial=c 172.16.200.1:0->172.16.200.4:0 tun_id=172.16.200.4 dst_
mtu=1500 dpd-link=on remote_location=0.0.0.0 weight=1
...
egress traffic control:
        bandwidth=737210(kbps) lock_hit=0 default_class=2 n_active_class=3
------------------------------------------------------
name=hub-phase1_1 ver=2 serial=d 172.16.200.1:0->172.16.200.2:0 tun_id=172.16.200.2 dst_
mtu=1500 dpd-link=on remote_location=0.0.0.0 weight=1
...
egress traffic control:
        bandwidth=726813(kbps) lock_hit=0 default_class=2 n_active_class=3
```

Running speed tests from spokes to the hub in dial-up IPsec tunnels

In this hub and spoke example, the hub is configured as an IPsec VPN dial-up server with an IPsec tunnel connected to the spoke. The hub is configured as a speed-test server to listen on custom ports (6000 and 7000), and the spoke is configured as a speed test client. This setup allows speed tests to successfully perform when spokes are behind NAT devices. The results of the speed test will be applied to the hub-phase1 overlay tunnel(s) as specified on the hub VPN interface.



The spokes are configured to initiate speed tests on a schedule on UDP. After the speed test completes, the results are sent to the hub, and the hub applies the results on its IPsec tunnel as egress traffic shaping. The results are also cached and can be used if an IPsec tunnel is disconnected and reconnected again.

To configure the Hub FortiGate as the speed test server:

1. Configure a shaping profile. In this example, the shaping profile is named profile_1.
   Three classes are used in the profile for low, medium, and high priority traffic. Each class is assigned a guaranteed and maximum bandwidth as a percentage of the measured bandwidth from the speed test.

```
config firewall shaping-profile
    edit "profile_1"
        set default-class-id 2
        config shaping-entries
            edit 1
                set class-id 2
                set priority low
                set guaranteed-bandwidth-percentage 10
                set maximum-bandwidth-percentage 10
            next
            edit 2
                set class-id 3
                set priority medium
                set guaranteed-bandwidth-percentage 30
                set maximum-bandwidth-percentage 40
            next
            edit 3
                set class-id 4
                set guaranteed-bandwidth-percentage 20
                set maximum-bandwidth-percentage 60
            next
        end
    end
end
```

2. Configure a shaping policy to assign certain traffic as a class ID. In this example, all traffic destined to the dialup tunnels are assigned class 3.

```
config firewall shaping-policy
    edit 0
        set service "ALL"
        set schedule "always"
        set dstintf "VPN1"
        set class-id 3
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

3. Enable a speed test server with custom speed-test listening ports.
   A speed test server is enabled on the hub. Port 7000 will run speed tests, and port 6000 will be the controller used to issue access tokens for speed test authentication.

```
config system global
     set speedtest-server enable
     set speedtestd-ctrl-port 6000
     set speedtestd-server-port 7000
end
```

4. Allow speed tests on the underlay:

```
config system interface
    edit "port1"
        set allowaccess ping speed-test
    next
end
```

5. Allow the speed test on the overlay and use the shaping profile in the interface. Ensure that the interface outbandwidth value uses the measured value from the speed test:

```
config system interface
    edit "hub-phase1"
        set allowaccess ping speed-test
        set egress-shaping-profile "profile_1"
        set interface "port1"
        set outbandwidth-source measured
    next
end
```

To configure the spoke FortiGate as a speed test client:

1. Configure a recurring schedule for the speed tests to run at 08:37 every day of the week:

```
config firewall schedule recurring
    edit "1"
        set start 08:37
        set day sunday monday tuesday wednesday thursday friday saturday
    next
end
```

2. Configure `system speed-test-schedule`:
   The protocol mode is set to UDP. The custom controller port used for authentication is set to 6000, and the custom port used to run the speed tests is set to 7000.

```
config system speed-test-schedule
    edit "spoke11-p1"
        set mode UDP
        set schedules "1"
        set dynamic-server enable
        set ctrl-port 6000
        set server-port 7000
    next
end
```

To view the speed test results:

On the spoke, authentication succeeds through port 6000, and the test runs on port 7000. UDP mode is used, and the test is successful.

Enable the following debugs prior to the start of the speedtest time:

```
Spoke # diagnose debug application speedtest -1
Spoke # diagnose debug enable
```

```
Hub # diagnose debug application speedtestd -1
Hub # diagnose debug enable
```

Once the speed test starts, the following output will be generated:

- Spoke:

```
zfcron_speedtest_ipsec_send_request()-580: root: HUB1-VPN1(HUB1-VPN1) try=0 token
request=0.0.0.0:0 -> 10.10.31.253:6000, test=10.198.1.2:0 -> 10.198.5.2:0
fcron_sptest_ipsec_on_start()-531: root: (00130007) HUB1-VPN1(HUB1-VPN1) server
10.198.5.2:7000 notify start token=7434e4fe-2ec5-51f1-78ae-4c3d21826993
__fork_run_ipsec_test()-991: [4598] Run test 00130007 for 'HUB1-VPN1'(port1) to server
10.198.5.2:7000 (tunnel:HUB1-VPN1)
[speedtest(4598)] blksize adjusted: 131072 -> 1460
[speedtest(4598)] start uploading test.

<Upload speed test results omitted>

[speedtest(4598)] speed test Done.
[speedtest(4598)] start downloading test.
[speedtest(4598)] Connecting to host 10.198.5.2, port 7000
[speedtest(4598)] Reverse mode, remote host 10.198.5.2 is sending
[speedtest(4598)] [ 27] local 10.198.1.2 port 10410 connected to 10.198.5.2 port 7000
[speedtest(4598)] [ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total
Datagrams
[speedtest(4598)] [ 27]   0.00-1.00   sec   100 MBytes   838 Mbits/sec  0.007 ms
179312/251121 (71%)
[speedtest(4598)] [ 27]   1.00-2.00   sec  93.4 MBytes   784 Mbits/sec  0.549 ms
176136/243220 (72%)
[speedtest(4598)] [ 27]   2.00-3.00   sec  87.1 MBytes   730 Mbits/sec  0.134 ms
196479/259043 (76%)
[speedtest(4598)] [ 27]   3.00-4.00   sec  90.7 MBytes   761 Mbits/sec  0.130 ms
183786/248934 (74%)
[speedtest(4598)] [ 27]   4.00-5.00   sec   105 MBytes   884 Mbits/sec  0.082 ms
197132/272787 (72%)
[speedtest(4598)] [ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total
Datagrams
[speedtest(4598)] [ 27]   0.00-5.00   sec  1.76 GBytes  3.03 Gbits/sec  0.000 ms  0/1297300
(0%)  sender
[speedtest(4598)] [SUM]  0.0- 5.0 sec  16 datagrams received out-of-order
[speedtest(4598)] [ 27]   0.00-5.00   sec   477 MBytes   799 Mbits/sec  0.082 ms
932845/1275105 (73%)  receiver
[speedtest(4598)] client(recver): bytes_recv=499699600, bytes_sent=1894058000, sender_
```

```
time=5.000, recver_time=5.000
[speedtest(4598)] client(recver): down_speed:  799 Mbits/sec
[speedtest(4598)]
[speedtest(4598)] speed test Done.
fcron_speedtest_notify_func()-1570: Speed test pid=4598 done

fcron_speedtest_on_test_finish()-1533: test 0x00130007 for 'HUB1-VPN1' succeed with
up=1509474, down=799469
fcron_speedtest_save_results()-1428: Write logs to disk: succ=1, fail=0
fcron_speedtest_sync_results()-1456: Sync cached results to secondary devices.
```

- Hub:

```
[sptestd::ctrl(0480):root] VPN1 ipv4(fd=11) recv request from=10.10.0.2, tun_id=10.10.0.2
[sptestd::ctrl(0522):root] VPN1(VPN1_1, 10.10.0.2) set client:addr=10.10.0.2,uuid=7434e4fe-
2ec5-51f1-78ae-4c3d21826993, wait=0
[speedtest(4002)::serv(0056)] auth result: 2, uuid=7434e4fe-2ec5-51f1-78ae-4c3d21826993,
peerv4=10.10.31.254
[speedtest(4002)] Accepted connection from 10.198.1.2, port 6868, dev port1(3), vf root(0)
[speedtest(4002)] [  9] local 10.198.5.2 port 7000 connected to 10.198.1.2 port 15316
[speedtest(4002)] notify SD-WAN test start, vfid=0, if=19 (VPN1), ip=10.10.31.254

<Download speed test results omitted>

[speedtest(4002)] server listening on 7000 (fd=5)
[speedtest(4002)::serv(0056)] auth result: 2, uuid=7434e4fe-2ec5-51f1-78ae-4c3d21826993,
peerv4=10.10.31.254
[speedtest(4002)] Accepted connection from 10.198.1.2, port 6872, dev port1(3), vf root(0)
[speedtest(4002)] [  9] local 10.198.5.2 port 7000 connected to 10.198.1.2 port 10410
[speedtest(4002)] notify SD-WAN test start, vfid=0, if=19 (VPN1), ip=10.10.31.254
[speedtest(4002)] [ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total
Datagrams
[speedtest(4002)] [  9]   0.00-5.06   sec  1.76 GBytes  2.99 Gbits/sec  0.000 ms  0/1297300
(0%)  sender
[speedtest(4002)] [  9]   0.00-5.00   sec   477 MBytes   800 Mbits/sec  0.000 ms
932845/1275105 (73%)  receiver
[speedtest(4002)] server(sender): bytes_recv=499699600, bytes_sent=1894058000, sender_
time=5.060, recver_time=5.000
[speedtest(4002)] server(sender): up_speed:  800 Mbits/sec
[speedtest(4002)] notify SD-WAN test stop, vfid=0, if=19, ip=10.10.31.254
[sptestd::ctrl(0303):root] VPN1(VPN1_1, 10.10.0.2) del client uuid=7434e4fe-2ec5-51f1-78ae-
4c3d21826993.
[sptestd::ctrl(0083):global] save logs to disk: succ=1  err=0
[speedtest(4002)::serv(0089)] report result uuid=7434e4fe-2ec5-51f1-78ae-4c3d21826993, done=1,
up=799519, down=0
[speedtest(4002)] server listening on 7000 (fd=5)
```

To verify that the result is cached on the spoke:

```
Hub # diagnose vpn tunnel list
......
-----------------------------------------------------
```

```
name=VPN1_1 ver=2 serial=5 10.198.5.2:0->10.198.1.2:0 nexthop=10.198.5.1 tun_id=10.10.0.2 tun_
id6=::10.0.0.5 status=up dst_mtu=1500 weight=1 country=ZZ
bound_if=3 real_if=3 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none options[0x22a8]=npu rgwy-
chg frag-rfc  run_state=0 role=primary accept_traffic=1 overlay_id=1

parent=VPN1 index=1
proxyid_num=1 child_num=0 refcnt=5 ilast=0 olast=0 ad=s/1
stat: rxp=23697 txp=23697 rxb=958226 txb=958637
dpd: mode=on-idle on=1 status=ok idle=60000ms retry=3 count=0 seqno=1
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
.......
egress traffic control:
        bandwidth=799519(kbps) lock_hit=0 default_class=2 n_active_class=3
        class-id=2      allocated-bandwidth=79951(kbps)         guaranteed-bandwidth=79951(kbps)
                        max-bandwidth=79951(kbps)       current-bandwidth=0(kbps)
                        priority=low    forwarded_bytes=1120
                        dropped_packets=0       dropped_bytes=0
        class-id=3      allocated-bandwidth=239856(kbps)        guaranteed-bandwidth=239855(kbps)
                        max-bandwidth=319807(kbps)      current-bandwidth=0(kbps)
                        priority=medium         forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
        class-id=4      allocated-bandwidth=479711(kbps)        guaranteed-bandwidth=159903(kbps)
                        max-bandwidth=479711(kbps)      current-bandwidth=0(kbps)
                        priority=high   forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
------------------------------------------------------
```

Hub learns egress shaping value from spoke over IKE

With peer egress shaping for IPsec tunnels enabled, IKE bandwidth negotiation is used for traffic shaping. This configuration is useful for speed tests of large-scale SD-WAN deployments with thousands of sites using diverse connections (4G/5G, satellite, broadband).

A static configuration of per-tunnel egress shaping values on spokes is automatically communicated to hubs during IKEv2 negotiation. The hub then immediately applies persistent policing based on these spoke-defined values, ensuring consistent QoS without performance overhead for variable-connectivity scenarios.

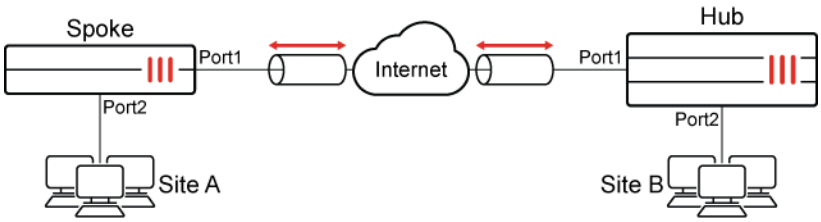
```
config vpn ipsec phase1-interface
    edit <tunnel_name>
        set ike-version 2
        set peer-egress-shaping {enable | disable}
        set peer-egress-shaping-value <integer>
    next
end
```

Command	Description
`peer-egress-shaping {enable | disable}`	Enable/disable peer egress shaping.

Command	Description
`peer-egress-shaping-value <integer>`	Configure outbound bandwidth to use for peer egress shaping, in kbps (0 - 80000000, default = 0).

NPU offloading for egress traffic shaping is not supported and must be disabled on the FortiGate applying the shaping.

In this hub and spoke example, a peer-egress-shaping-value is configured on the spoke, and the hub learns the value through IKE negotiations. Subsequently, the hub applies the shaping value to its shaping profile, which enforces egress traffic policing on the overlay tunnel to the spoke.



The IP addresses are:

FortiGate	Interface	IP Address
Spoke	Port2	10.1.100.1
	Port1	172.16.200.1
Hub	Port1	172.16.200.4
	Port2	192.168.5.4

The basic configuration steps are:

1. Configure the hub:
   a. Configure IPsec VPN
   b. Configure a static route
   c. Configure firewall policies
   d. Configure the shaping class
   e. Configure the shaping profile
   f. Apply the shaping profile to the hub's tunnel interface
2. Configure the spoke:
   a. Configure IPsec VPN
   b. Configure a static route
   c. Configure firewall policies

Configure the hub

To configure IPsec VPN on the hub:

```
config vpn ipsec phase1-interface
    edit "tospokes"
        set type dynamic
        set interface "port1"
        set ike-version 2
        set peertype any
        set net-device disable
        set peer-egress-shaping enable
        set mode-cfg enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set add-route disable
        set dpd on-idle
        set npu-offload disable
        set auto-discovery-sender enable
        set transport auto
        set ipv4-start-ip 100.100.100.10
        set ipv4-end-ip 100.100.100.20
        set ipv4-netmask 255.255.255.0
        set psksecret ******
        set dpd-retryinterval 60
    next
end
config vpn ipsec phase2-interface
    edit "tospokes"
        set phase1name "tospokes"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
    next
end
```

To configure a static route on the hub:

```
config router static
    edit 1
        set device "tospokes"
    next
end
```

To configure firewall policies on the hub:

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "tospokes"
        set action accept
```

```
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set srcintf "tospokes"
        set dstintf "port2"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
    next
end
```

To configure the shaping class on the hub:

```
config firewall traffic-class
    edit 2
        set class-name "class_id_2"
    next
end
```

To configure the shaping profile on the hub:

```
config firewall shaping-profile
    edit "hub_spoke"
        set type policing
        set default-class-id 2
        config shaping-entries
            edit 1
                set class-id 2
                set guaranteed-bandwidth-percentage 100
                set maximum-bandwidth-percentage 100
            next
        end
    next
end
```

To apply the shaping profile to the hub's tunnel interface:

```
config system interface
    edit "tospokes"
        set egress-shaping-profile "hub_spoke"
        set outbandwidth 0
    next
end
```

When configuring egress shaping on a tunnel interface, do not manually set the `outbandwidth` value. It is dynamically obtained from the `PEER_EGRESS_SHAPING_VALUE` for each tunnel and should be left at its default value of `0`.

Configure the spoke

To configure IPsec VPN on the spoke:

```
config vpn ipsec phase1-interface
    edit "tohub-d"
        set interface "port1"
        set ike-version 2
        set peertype any
        set net-device disable
        set peer-egress-shaping enable
        set peer-egress-shaping-value 500
        set mode-cfg enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set add-route disable
        set auto-discovery-receiver enable
        set auto-discovery-shortcuts dependent
        set transport auto
        set remote-gw 172.16.200.4
        set psksecret ******
    next
end
config vpn ipsec phase2-interface
    edit "tohub-d"
        set phase1name "tohub-d"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
        set auto-negotiate enable
    next
end
```

To configure a static route on the spoke:

```
config router static
    edit 1
        set device "tohub-d"
    next
end
```

To configure firewall policies on the spoke:

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "tohub-d"
```

```
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set srcintf "tohub-d"
        set dstintf "port2"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
    next
end
```

Verification

1. Verify that traffic can be sent from Site A to Site B:

```
# ping 192.168.5.55
Pinging 192.168.5.55 with 32 bytes of data:
Reply from 192.168.5.55: bytes=32 time=2ms TTL=62
Reply from 192.168.5.55: bytes=32 time=2ms TTL=62
Reply from 192.168.5.55: bytes=32 time=1ms TTL=62
Reply from 192.168.5.55: bytes=32 time<1ms TTL=62

Ping statistics for 192.168.5.55:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 1ms
```

2. Check the hub tunnel shaping:

```
# diagnose vpn tunnel list | grep -A 5 "egress traffic control"
egress traffic control:
        bandwidth=500(kbps) lock_hit=0 default_class=2 n_active_class=1
        class-id=2      allocated-bandwidth=500(kbps)   guaranteed-bandwidth=500(kbps)
                        max-bandwidth=500(kbps)         current-bandwidth=1(kbps)
                        priority=high   forwarded_bytes=1512
                        dropped_packets=0       dropped_bytes=0
```

3. Confirm IKE negotiation on the hub:

```
# diagnose debug application ike -1
ike V=root:0:ac59d6f4e4dc005e/0000000000000000:6: responder received SA_INIT msg
ike V=root:0:ac59d6f4e4dc005e/0000000000000000:6: received notify type NAT_DETECTION_SOURCE_IP
ike V=root:0:ac59d6f4e4dc005e/0000000000000000:6: received notify type NAT_DETECTION_
DESTINATION_IP
ike V=root:0:ac59d6f4e4dc005e/0000000000000000:6: received notify type FRAGMENTATION_SUPPORTED
ike V=root:0:ac59d6f4e4dc005e/0000000000000000:6: received notify type AUTO_DISCOVERY_RECEIVER
```

```
ike V=root:0:ac59d6f4e4dc005e/0000000000000000:6: received notify type PEER_EGRESS_SHAPING
ike V=root:0:ac59d6f4e4dc005e/0000000000000000:6: incoming proposal:
ike V=root:0:ac59d6f4e4dc005e/0000000000000000:6: proposal id = 1:
...
ike V=root:0:tospokes:6: processing notify type NAT_DETECTION_SOURCE_IP
ike V=root:0:tospokes:6: processing NAT-D payload
ike V=root:0:tospokes:6: NAT not detected
ike V=root:0:tospokes:6: process NAT-D ike V=root:0:tospokes:6: processing notify type NAT_
DETECTION_DESTINATION_IP
ike V=root:0:tospokes:6: processing NAT-D payload
ike V=root:0:tospokes:6: NAT not detected ike V=root:0:tospokes:6: process NAT-D
ike V=root:0:tospokes:6: processing notify type FRAGMENTATION_SUPPORTED
ike V=root:0:tospokes:6: processing notify type AUTO_DISCOVERY_RECEIVER
ike V=root:0:tospokes:6: processing notify type PEER_EGRESS_SHAPING
ike V=root:0:tospokes:6: received peer egress shaping
ike V=root:0:tospokes:6: responder preparing SA_INIT msg
...
ike V=root:0:tospokes:6: processing notify type INITIAL_CONTACT
ike V=root:0:tospokes:6: processing notify type MESSAGE_ID_SYNC_SUPPORTED
ike V=root:0:tospokes:6: processing notify type PEER_EGRESS_SHAPING_VALUE
ike V=root:0:tospokes:6: received peer egress shaping value=500
ike V=root:0:tospokes:6: peer identifier IPV4_ADDR 172.16.200.1
ike V=root:0:tospokes:6: re-validate gw ID
ike V=root:0:tospokes:6: gw validation OK
...
```

4. Confirm IKE negotiation on the spoke:

```
# diagnose debug application ike -1
ike V=root:0:tohub-d:3: processing notify type FRAGMENTATION_SUPPORTED
ike V=root:0:tohub-d:3: processing notify type AUTO_DISCOVERY_SENDER
ike V=root:0:tohub-d:3: processing notify type PEER_EGRESS_SHAPING
ike V=root:0:tohub-d:3: received peer egress shaping
ike V=root:0:tohub-d:3: incoming proposal: ike V=root:0:tohub-d:3: proposal id = 1:
...
ike V=root:0:tohub-d:3: authentication succeeded
ike V=root:0:tohub-d:3: processing notify type MESSAGE_ID_SYNC_SUPPORTED
ike V=root:0:tohub-d:3: processing notify type INTERFACE_ADDR4
ike V=root:0:tohub-d:3: INTERFACE-ADDR4 100.100.100.4
ike V=root:0:tohub-d:3: processing notify type PEER_EGRESS_SHAPING_VALUE
ike V=root:0:tohub-d:3: received peer egress shaping value=0
ike V=root:0:tohub-d:3: processing notify type ADVPN_EXT_IP4
ike V=root:0:tohub-d:3: ext IP = 172.16.200.1
```

Spoke learns egress shaping value from other spoke over ADVPN shortcut

With peer egress shaping for IPsec tunnels enabled, IKE bandwidth negotiation is used for traffic shaping. This configuration is useful for speed tests of large-scale SD-WAN deployments with thousands of sites using diverse connections (4G/5G, satellite, broadband).

ADVPN spoke-to-spoke shaping evaluates both the local and remote peers' egress shaping values and applies the lower of the two to optimize traffic flow. Additionally, this update enables the synchronization of the `peer-egress-shaping-value` during ADVPN shortcut reply messages sent to the shortcut initiator. Subsequently, during the regular IPsec negotiation, the initiator shares its shaping value with the responder, allowing shaping to be applied accordingly.

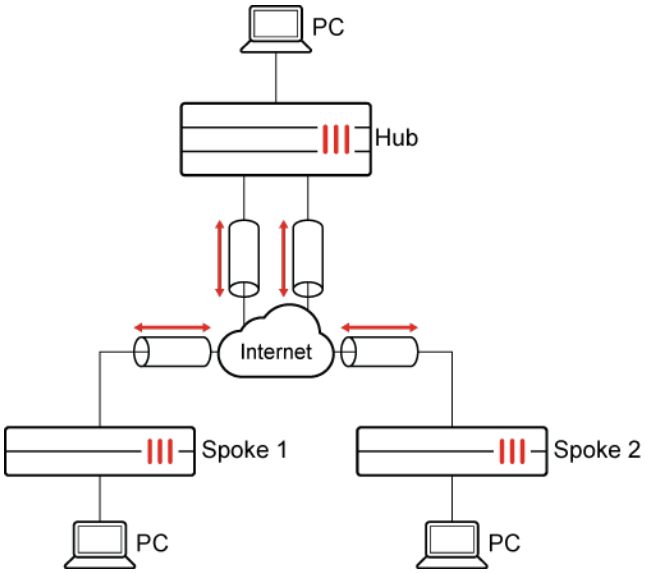
```
config vpn ipsec phase1-interface
    edit <tunnel_name>
        set ike-version 2
        set peer-egress-shaping {enable | disable}
        set peer-egress-shaping-value <integer>
    next
end
```

Command	Description
`peer-egress-shaping {enable | disable}`	Enable/disable peer egress shaping.
`peer-egress-shaping-value <integer>`	Configure outbound bandwidth to use for peer egress shaping, in kbps (0 - 80000000, default = 0).

NPU offloading for egress traffic shaping is not supported and must be disabled on the FortiGate applying the shaping.

Example

This example shows the configuration of a hub with two spokes.



A peer-egress-shaping-value is configured on both Spoke 1 and Spoke 2, and the peer spoke learns the value through ADVPN's shortcut-reply. Subsequently, each spoke applies the egress bandwidth limit to their overlay tunnel.

The IP addresses are as follows:

FortiGate	Interface	IP adress
Spoke1	Port1	172.16.200.1
Spoke1 PC	Port2	10.1.100.1
Spoke2	Port1	172.16.200.3
Spoke2 PC	Port2	192.168.4.3
Hub	Port1	172.16.200.4
Hub PC	Port2	192.168.5.4

The configuration steps are:

1. Configure the Hub:
   a. Configure the IPsec VPN
   b. Configure the static route
   c. Configure the firewall policies
2. Configure Spoke1:
   a. Configure the IPsec VPN
   b. Configure the static route
   c. Configure the firewall policies
   d. Configure the shaping class
   e. Configure the shaping profile
   f. Apply shaping profile to Spoke1's tunnel interface
3. Configure Spoke2:
   a. Configure the IPsec VPN
   b. Configure the static route
   c. Configure the firewall policies
   d. Configure the shaping class
   e. Configure the shaping profile
   f. Apply shaping profile to Spoke2's tunnel interface

To configure the hub:

1. Configure the IPsec VPN:

```
config vpn ipsec phase1-interface
    edit "tospokes"
        set type dynamic
        set interface "port1"
        set ike-version 2
        set peertype any
        set net-device disable
        set mode-cfg enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set add-route disable
```

```
        set dpd on-idle
        set npu-offload disable
        set auto-discovery-sender enable
        set transport auto
        set ipv4-start-ip 100.100.100.10
        set ipv4-end-ip 100.100.100.20
        set ipv4-netmask 255.255.255.0
        set psksecret ******
        set dpd-retryinterval 60
    next
end
```

```
config vpn ipsec phase2-interface
    edit "tospokes"
        set phase1name "tospokes"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
    next
end
```

2. Configure the static route:

```
config router static
    edit 1
        set device "tospokes"
    next
end
```

3. Configure the firewall policies:

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "tospokes"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set srcintf "tospokes"
        set dstintf "port2"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
    next
    edit 3
        set srcintf "tospokes"
        set dstintf "tospokes"
```

```
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
    next
end
```

To configure spoke1:

1. Configure the IPsec VPN:

```
config vpn ipsec phase1-interface
    edit "tohub-d"
        set interface "port1"
        set ike-version 2
        set peertype any
        set net-device disable
        set peer-egress-shaping enable
        set peer-egress-shaping-value 500
        set npu-offload disable
        set mode-cfg enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set add-route disable
        set auto-discovery-shortcuts enable
        set auto-discovery-receiver enable
        set transport auto
        set remote-gw 172.16.200.4
        set psksecret ******
    next
end
```

```
config vpn ipsec phase2-interface
    edit "tohub-d"
        set phase1name "tohub-d"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
        set auto-negotiate enable
    next
end
```

2. Configure the static route:

```
config router static
    edit 1
        set device "tohub-d"
    next
end
```

3. Configure the firewall policies:

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "tohub-d"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set srcintf "tohub-d"
        set dstintf "port2"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
    next
end
```

4. Configure the shaping class:

```
config firewall traffic-class
    edit 2
        set class-name "class_id_2"
    next
end
```

5. Configure the shaping profile:

```
config firewall shaping-profile
    edit "spoke_spoke"
        set type policing
        set default-class-id 2
        config shaping-entries
            edit 1
                set class-id 2
                set guaranteed-bandwidth-percentage 100
                set maximum-bandwidth-percentage 100
            next
        end
    next
end
```

6. Apply shaping profile to Spoke1's tunnel interface:

```
config system interface
    edit "tospokes"
        set egress-shaping-profile "hub_spoke"
    next
end
```

When configuring egress shaping on a tunnel interface, do not manually set the `outbandwidth` value. This value is dynamically obtained from the PEER_EGRESS_SHAPING_VALUE for each tunnel and should be left at its default value of 0.

To configure spoke2:

1. Configure the IPsec VPN:

```
config vpn ipsec phase1-interface
    edit "tohub-d"
        set interface "port1"
        set ike-version 2
        set peertype any
        set net-device disable
        set peer-egress-shaping enable
        set peer-egress-shaping-value 500
        set npu-offload disable
        set mode-cfg enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set add-route disable
        set auto-discovery-receiver enable
        set auto-discovery-shortcuts dependent
        set transport auto
        set remote-gw 172.16.200.4
        set psksecret ******
    next
end
```

```
config vpn ipsec phase2-interface
    edit "tohub-d"
        set phase1name "tohub-d"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
        set auto-negotiate enable
    next
end
```

2. Configure the static route:

```
config router static
    edit 1
        set device "tohub-d"
    next
end
```

3. Configure the firewall policies:

```
config firewall policy
    edit 1
        set srcintf "port2"
```

```
        set dstintf "tohub-d"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set srcintf "tohub-d"
        set dstintf "port2"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
    next
end
```

4. Configure the shaping class:

```
config firewall traffic-class
    edit 2
        set class-name "class_id_2"
    next
end
```

5. Configure the shaping profile:

```
config firewall shaping-profile
    edit "spoke_spoke"
        set type policing
        set default-class-id 2
        config shaping-entries
            edit 1
                set class-id 2
                set guaranteed-bandwidth-percentage 100
                set maximum-bandwidth-percentage 100
            next
        end
    next
end
```

6. Apply shaping profile to Spoke2's tunnel interface:

```
config system interface
    edit "tospokes"
        set egress-shaping-profile "hub_spoke"
    next
end
```

To verify the configuration:

1. Verify that traffic can be send from Spoke1 to Spoke2:

```
# ping 192.168.4.3
Pinging 192.168.4.3 with 32 bytes of data:
Reply from 192.168.4.3: bytes=32 time=2ms TTL=62
Reply from 192.168.4.3: bytes=32 time=2ms TTL=62
Reply from 192.168.4.3: bytes=32 time=1ms TTL=62
Reply from 192.168.4.3: bytes=32 time<1ms TTL=62

Ping statistics for 192.168.4.3:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 2ms, Average = 1ms
```

2. Check Spoke1 tunnel shaping:

```
# diagnose vpn tunnel list | grep -A 5 "egress traffic control"
egress traffic control:
        bandwidth=500(kbps) lock_hit=1 default_class=2 n_active_class=1
        class-id=2      allocated-bandwidth=500(kbps)   guaranteed-bandwidth=500(kbps)
                        max-bandwidth=500(kbps)         current-bandwidth=1(kbps)
                        priority=high   forwarded_bytes=84
                        dropped_packets=0       dropped_bytes=0
```

3. Check Spoke2 tunnel shaping:

```
# diagnose vpn tunnel list | grep -A 5 "egress traffic control"
egress traffic control:
        bandwidth=500(kbps) lock_hit=0 default_class=2 n_active_class=1
        class-id=2      allocated-bandwidth=500(kbps)   guaranteed-bandwidth=500(kbps)
                        max-bandwidth=500(kbps)         current-bandwidth=1(kbps)
                        priority=high   forwarded_bytes=824
                        dropped_packets=0       dropped_bytes=0
```

4. Confirm IKE negotiation on Spoke1:

```
# diagnose debug application ike -1
ike V=root:0:tohub-d:7: received informational request
ike V=root:0:tohub-d:7: processing notify type SHORTCUT_REPLY
ike V=root:0:tohub-d: recv shortcut-reply 3150081142027665143
6cb78f05ef0079bc/11b6b9160e729839 172.16.200.3 to 10.1.100.11 0 psk 64 ppk 0 ver 2 mode 0 ext-
mapping 172.16.200.3:0, network-id 0/0 shaping=500
ike V=root:0:tohub-d: iif 48 192.168.4.33->10.1.100.11 0 route lookup oif 8 port2 gwy 0.0.0.0
ike V=root:0:tohub-d: shortcut-reply received from 172.16.200.3:500, local-nat=no, peer-nat=no
ike V=root:0:tohub-d:tohub-d: created connection: 0x55c5c21d3860 7 172.16.200.1-
>172.16.200.3:500. ike V=root:0:tohub-d: adding new dynamic tunnel for 172.16.200.3:500
ike V=root:0:tohub-d_0: tunnel created tun_id 172.16.200.3/::172.16.200.3 remote_location
0.0.0.0
ike V=root:0:tohub-d_0: added new dynamic tunnel for 172.16.200.3:500
ike V=root:0:tohub-d_0: shortcut selector added new 1 a203
...
```

```
ike V=root:0:tohub-d_0:10: initiator received AUTH msg
ike V=root:0:tohub-d_0:10: peer identifier IPV4_ADDR 172.16.200.3
ike V=root:0:tohub-d_0:10: auth verify done
ike V=root:0:tohub-d_0:10: initiator AUTH continuation
ike V=root:0:tohub-d_0:10: authentication succeeded
ike V=root:0:tohub-d_0:10: processing notify type MESSAGE_ID_SYNC_SUPPORTED
ike V=root:0:tohub-d_0:10: processing notify type INTERFACE_ADDR4
ike V=root:0:tohub-d_0:10: INTERFACE-ADDR4 100.100.100.11
ike V=root:0:tohub-d_0:10: processing notify type PEER_EGRESS_SHAPING_VALUE
ike V=root:0:tohub-d_0:10: received peer egress shaping value=500
ike V=root:0:tohub-d_0:10: established IKE SA 6cb78f05ef0079bc/11b6b9160e729839
ike V=root:0:tohub-d_0:10: auto-discovery receiver
ike V=root:0:tohub-d_0:10: auto-discovery 2
ike V=root:0:tohub-d_0:10: check peer route: if_addr4_rcvd=1, if_addr6_rcvd=0, mode_cfg=0
ike V=root:0:tohub-d_0:10: update peer route 0.0.0.0 -> 100.100.100.11
ike V=root:0:tohub-d_0: add peer static route 100.100.100.11 via 172.16.200.3 dev tohub-d_0
ike V=root:0:tohub-d: schedule auto-negotiate
...
```

5. Confirm IKE negotiation on Spoke2:

```
# diagnose debug application ike -1
ike V=root:0:tohub-d:7: processing notify type SHORTCUT_QUERY
ike V=root:0:tohub-d: recv shortcut-query 3150081142027665143
6cb78f05ef0079bc/0000000000000000 172.16.200.1 10.1.100.11:2048->192.168.4.33:0 0 psk 64 ppk 0
ttl 31 nat 0 ver 2 mode 0 network-id 0 ike V=root:0:tohub-d: iif 46 10.1.100.11->192.168.4.33
0 route lookup oif 8 port2 gwy 0.0.0.0 ike V=root:0:tohub-d: shortcut-query received from
172.16.200.1:500, local-nat=no, peer-nat=no
ike 0:6cb78f05ef0079bc/11b6b9160e729839:8: out FF
ike V=root:0:ff00000000000000/2100000000000000:f32a0000:8: sent
IKE msg (keepalive): 172.16.200.3:500->172.16.200.1:500, len=1, vrf=0,
id=ff00000000000000/2100000000000000:f32a0000, oif=7
ike V=root:0:tohub-d: send vwl advpn-info request (0x763f04e2) for intf tohub-d ike
V=root:0:tohub-d: send vwl request error: No such file or directory
ike V=root:0:tohub-d: send shortcut-reply 3150081142027665143
6cb78f05ef0079bc/11b6b9160e729839 172.16.200.3 to 10.1.100.11 0 psk 64 ppk 0 ver 2 mode 0
network-id 0/0 shaping=500
...
ike V=root:0:tohub-d:8: responder received AUTH msg
ike V=root:0:tohub-d:8: processing notify type INITIAL_CONTACT
ike V=root:0:tohub-d:8: processing notify type INTERFACE_ADDR4
ike V=root:0:tohub-d:8: INTERFACE-ADDR4 100.100.100.10
ike V=root:0:tohub-d:8: processing notify type MESSAGE_ID_SYNC_SUPPORTED
ike V=root:0:tohub-d:8: processing notify type PEER_EGRESS_SHAPING_VALUE
ike V=root:0:tohub-d:8: received peer egress shaping value=500
ike V=root:0:tohub-d:8: peer identifier IPV4_ADDR 172.16.200.1
ike V=root:0:tohub-d:8: auth verify done
ike V=root:0:tohub-d:8: responder AUTH continuation
ike V=root:0:tohub-d:8: authentication succeeded
ike V=root:0:tohub-d:8: responder creating new child
...
```

Speed test usage

Using speed test results with SD-WAN

The interface speed test can be used to populate the bandwidth values based on the results.

To manually configure the upstream and downstream interface bandwidth values in the GUI:

1. Go to *Network* > *Interfaces*.
2. Edit a WAN interface.
3. Input the results from the speed test in the *Estimated bandwidth* section, specifically into the appropriate *kbps Upstream* or *kbps Downstream* fields.
4. Click *OK*.

Alternatively, use the GUI speed test on page 1239 to add the speed test results to the interface's estimated bandwidth with one click.

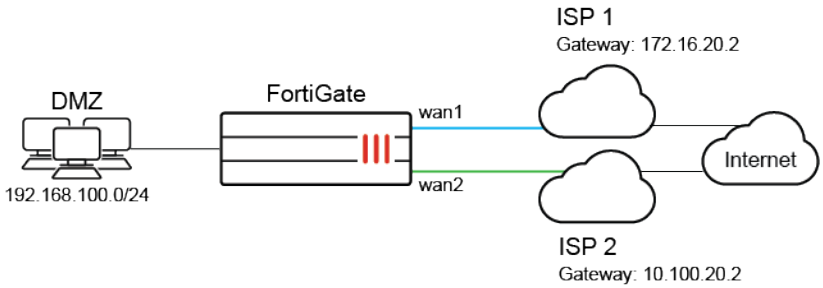
To manually configure the upstream and downstream interface bandwidth values in the CLI:

```
config system interface
    edit <interface>
        set estimated-upstream-bandwidth <speed in kbps>
        set estimated-downstream-bandwidth <speed in kbps>
    next
end
```

The populated bandwidth values can be employed in conjunction with various WAN intelligence strategies such as load balancing without SLA targets, load balancing with SLA targets, and best quality. These strategies are used to select an SD-WAN interface in SD-WAN service rules.

Example

In this example, SD-WAN interfaces, wan1 and wan2, are connected to two different internet service providers (ISPs), both providing access to the public internet. The preference is for Gmail services to use the link with the highest bandwidth.



To apply speed test results to a best quality SD-WAN rule:

1. Run theGUI speed test on page 1239 on wan1 and wan2:
   a. When the test is complete, click *OK* in the *Confirm* pane to apply the results to the estimated bandwidth.
2. Add wan1 and wan2 as SD-WAN members, then add a policy and static route. See SD-WAN quick start on page 1041 for more information.
3. Configure the health check:
   a. Go to *Network* > *SD-WAN*, select the *Performance SLAs* tab, and click *Create New*.
   b. Enter a name for the performance SLA, such as *google*, and set the *Server* to *google.com*. See Health checks for more information.
   c. Click *OK*.
4. Configure an SD-WAN rule to use best quality with bandwidth set as the quality criteria:
   a. Go to *Network* > *SD-WAN*, select the *SD-WAN Rules* tab, and click *Create New*.
   b. Enter a name for the rule, such as *gmail*.
   c. Configure the following settings:

Internet service	*Google-Gmail*
Interface selection strategy	*Best quality*
Interface preference	*wan1* and *wan2*
Measured SLA	*google*
Quality criteria	*Bandwidth*

   d. Click *OK*.
5. Verify that the *Estimated bandwidth* values are used to select an SD-WAN interface in SD-WAN service rules.
   a. Verify the health check status:

```
# diagnose sys sdwan health-check status
Health Check(google):
Seq(1 wan1): state(alive), packet-loss(0.000%) latency(4.996), jitter(0.824), mos(4.401),
bandwidth-up(414400), bandwidth-dw(405115), bandwidth-bi(819515) sla_map=0x0
Seq(2 wan2): state(alive), packet-loss(0.000%) latency(4.634), jitter(0.340), mos(4.402),
bandwidth-up(443975), bandwidth-dw(699264), bandwidth-bi(1143239) sla_map=0x0
```

   b. Verify the service rule 1 diagnostics:

```
# diagnose sys sdwan service4 1

Service(1): Address Mode(IPV4) flags=0x4200 use-shortcut-sla use-shortcut
 Tie break: cfg
  Gen(1), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(priority), link-
cost-factor(bibandwidth), link-cost-threshold(10), health-check(google)
  Members(2):
```

```
    1: Seq_num(2 wan2 virtual-wan-link), alive, bibandwidth: 1143239Kbps, selected
    2: Seq_num(1 wan1 virtual-wan-link), alive, bibandwidth: 819506Kbps, selected
  Internet Service(1): Google-Gmail(65646,0,0,0,0)
```

Since wan2 has superior bandwidth, SD-WAN will prioritize Seq_num(2) over Seq_num(1), and wan2 will be utilized to route Gmail traffic. The bandwidth is automatically determined by the *Estimated bandwidth* values, which were populated by running a speed test in the GUI.

Using speed test results with traffic shaping

The interface speed test can be used to populate the bandwidth values based on the results.

To configure the outbandwidth value in the GUI:

1. Go to *Network* > *Interfaces*.
2. Edit an interface.
3. In the *Traffic Shaping* section, enable *Outbound bandwidth* and enter the result obtained from the speed test's uploading mode.
4. Click *OK*.

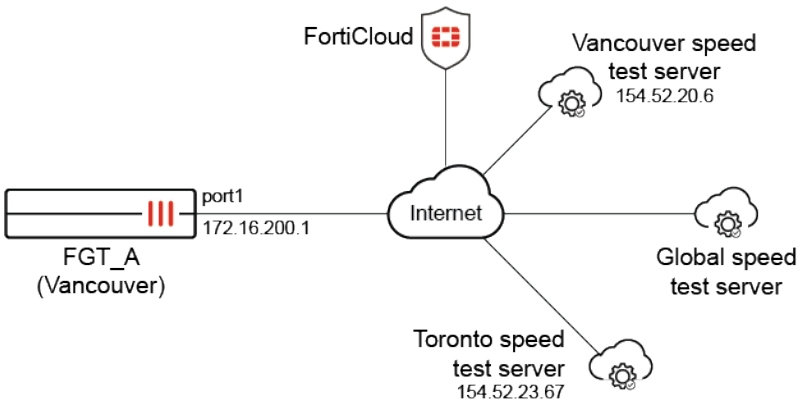
To configure the outbandwidth and inbandwidth values in the CLI:

```
config system interface
    edit <interface>
        set outbandwidth <speed_in_kbps>
        set inbandwidth <speed_in_kbps>
    next
end
```

The populated bandwidth values can be used in conjunction with either an egress or ingress traffic shaping profile.

Speed test examples

This topic includes examples that show various tests based on different modes (auto, TCP, UDP), latency thresholds, and test servers. Some test protocols and servers are manually configured, while others are chosen by the FortiGate.



These examples assume the FortiGate is connected to the internet, has a valid SD-WAN Underlay and Application Monitoring license, and has downloaded the server list of speed tests from FortiCloud. See CLI speed test on page 1238 for more information.

- Example 1: executing a speed test without specifying the interface, server, and mode
- Example 2: executing a speed test with a lower latency threshold setting
- Example 3: executing the speed test with diagnose netlink interface speed-test
- Example 4: executing the speed test according to the schedule
- Example 5: executing multiple speed tests with TCP and UDP connections
- Example 6: executing a manual speed test for debugging

Example 1: executing a speed test without specifying the interface, server, and mode

Geographically, the Vancouver server (154.52.20.6) has the smallest latency (around 7 ms) to FGT_A, so it will be automatically selected for the speed test because the latency 7 ms to 154.52.20.6 is less than the default `latency-threshold` of 60 ms. Meanwhile, four TCP connections will be initiated to perform the test since the default `multiple-tcp-stream` is 4.

To execute the speed test without specifying parameters:

1. Configure the speed test settings:

```
config system speed-test-setting
    set latency-threshold 60
    set multiple-tcp-stream 4
end
```

2. Execute a ping to the closest test server, 154.52.20.6, to learn the latency for the connection:

```
# execute ping 154.52.20.6
PING 154.52.20.6 (154.52.20.6): 56 data bytes
64 bytes from 154.52.20.6: icmp_seq=0 ttl=50 time=7.5 ms
64 bytes from 154.52.20.6: icmp_seq=1 ttl=50 time=7.2 ms
64 bytes from 154.52.20.6: icmp_seq=2 ttl=50 time=7.1 ms
64 bytes from 154.52.20.6: icmp_seq=3 ttl=50 time=7.1 ms
64 bytes from 154.52.20.6: icmp_seq=4 ttl=50 time=9.1 ms
```

```
--- 154.52.20.6 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 7.1/7.6/9.1 ms
```

3. Run the speed test with no parameters:

```
# execute speed-test
Initializing speed test.
current vdom=root
Run in uploading mode.
Connecting to host 154.52.20.6, port 5203
[  7] local 172.16.200.1 port 21219 connected to 154.52.20.6 port 5203
[  9] local 172.16.200.1 port 21220 connected to 154.52.20.6 port 5203
[ 11] local 172.16.200.1 port 21221 connected to 154.52.20.6 port 5203
[ 13] local 172.16.200.1 port 21222 connected to 154.52.20.6 port 5203
[ ID] Interval           Transfer     Bitrate         Retr  Cwnd
[  7]   0.00-1.00   sec  22.4 MBytes   188 Mbits/sec   17    140 KBytes
[  9]   0.00-1.00   sec  9.71 MBytes  81.4 Mbits/sec    6   73.5 KBytes
[ 11]   0.00-1.00   sec  18.5 MBytes   155 Mbits/sec   12    117 KBytes
...
[SUM]   0.00-5.02   sec   321 MBytes   536 Mbits/sec                  receiver

speed test Done.
Run in reverse downloading mode.
Connecting to host 154.52.20.6, port 5203
Reverse mode, remote host 154.52.20.6 is sending
[  7] local 172.16.200.1 port 21228 connected to 154.52.20.6 port 5203
[ 11] local 172.16.200.1 port 21229 connected to 154.52.20.6 port 5203
...
[SUM]   0.00-5.00   sec   331 MBytes   555 Mbits/sec                  receiver

speed test Done.
```

The tested upload/download speed for port1 is 536 Mbps/555 Mbps when connecting to the closest server with four TCP connections.

Example 2: executing a speed test with a lower latency threshold setting

The `latency-threshold` setting is changed to 5 ms, which is less than the latency 7 ms to 154.52.20.6. When executing the speed test, one UDP connection will be initiated as expected.

To execute the speed test with a lower latency threshold setting:

1. Edit the speed test settings:

```
config system speed-test-setting
    set latency-threshold 5
end
```

2. Run the speed test:

```
# execute speed-test
Speed test quota for 7/19 is 4
current vdom=root
Run in uploading mode.
Connecting to host 154.52.20.6, port 5202
[  7] local 172.16.200.1 port 5315 connected to 154.52.20.6 port 5202
[ ID] Interval           Transfer     Bitrate         Total Datagrams
[  7]   0.00-1.00   sec   111 MBytes   931 Mbits/sec  80337
[  7]   1.00-2.00   sec   111 MBytes   932 Mbits/sec  80476
[  7]   2.00-3.00   sec   111 MBytes   932 Mbits/sec  80451
[  7]   3.00-4.00   sec   111 MBytes   932 Mbits/sec  80460
[  7]   4.00-5.00   sec   111 MBytes   934 Mbits/sec  80640
- - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagrams
[  7]   0.00-5.00   sec   556 MBytes   932 Mbits/sec  0.000 ms  0/402364 (0%)  sender
[  7]   0.00-5.04   sec   550 MBytes   917 Mbits/sec  0.017 ms  3787/402339 (0.94%)  receiver

speed test Done.
Run in reverse downloading mode.
Connecting to host 154.52.20.6, port 5202
Reverse mode, remote host 154.52.20.6 is sending
[  7] local 172.16.200.1 port 19940 connected to 154.52.20.6 port 5202
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagrams
[  7]   0.00-1.00   sec  72.4 MBytes   607 Mbits/sec  0.013 ms  59813/112240 (53%)
[  7]   1.00-2.00   sec  70.9 MBytes   595 Mbits/sec  0.015 ms  58130/109486 (53%)
[  7]   2.00-3.00   sec  69.2 MBytes   581 Mbits/sec  0.012 ms  60192/110329 (55%)
[  7]   3.00-4.00   sec  71.3 MBytes   598 Mbits/sec  0.012 ms  58107/109710 (53%)
[  7]   4.00-5.00   sec  71.1 MBytes   596 Mbits/sec  0.014 ms  58786/110260 (53%)
- - - - - - - - - - - - - - - - - - - - - - - -
[ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total Datagrams
[  7]   0.00-5.04   sec   764 MBytes  1.27 Gbits/sec  0.000 ms  0/553023 (0%)  sender
[SUM]  0.0- 5.0 sec  2 datagrams received out-of-order
[  7]   0.00-5.00   sec   355 MBytes   595 Mbits/sec  0.014 ms  295028/552025 (53%)  receiver

speed test Done.
```

The tested upload/download speed for port1 is 917 Mbps/595 Mbps when connecting to the closest server with one UDP connection.

Example 3: executing the speed test with diagnose netlink interface speed-test

After running this diagnose command, the results are recorded in the interface settings for reference as `measured-upstream-bandwidth` and `measured-downstream-bandwidth`.

To execute the speed test:

```
# diagnose netlink interface speed-test port1 FTNT_CA_Vancouver TCP
speed-test test ID is b0066
...
```

To view the interface settings:

```
show system interface port1
config system interface
    edit "port1"
        ...
        set measured-upstream-bandwidth 735682
        set measured-downstream-bandwidth 746573
        set bandwidth-measure-time 1689811319
        ...
    next
end
```

Example 4: executing the speed test according to the schedule

In this example, a speed test is configured to occur within a two-hour window. Minimum in and out bandwidths are specified for the speed test results. The results of the speed test are then applied to the in and out bandwidths in memory so that QoS can use them.

To execute the speed test according to the schedule:

1. Configure the recurring schedule:

```
config firewall schedule recurring
    edit "1"
        set start 06:00
        set end 08:00
        set day sunday monday tuesday wednesday thursday friday saturday
    next
end
```

2. Configure the speed test schedule:

```
config system speed-test-schedule
    edit "port1"
        set mode TCP
        set schedules "1"
        set update-bandwidth-limit-unit value
        set update-inbandwidth-minimum 500000
        set update-outbandwidth-minimum 500000
    next
end
```

3. Make sure that the results of the speed test update the interface's in and out bandwidths:

```
config system interface
    edit "port1"
        set inbandwidth-source measured
        set outbandwidth-source measured
    next
end
```

4. Enable debugging prior to the speed-test window to view the process:

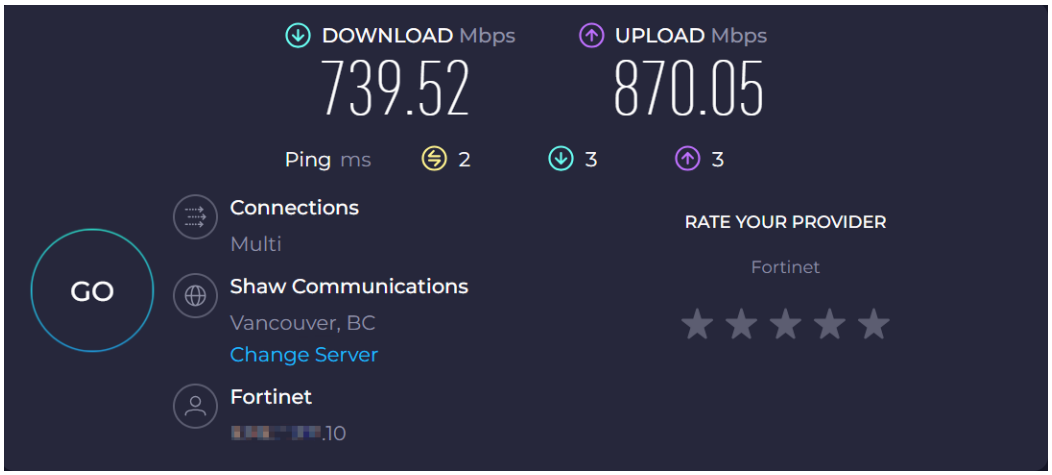
```
diagnose debug application speedtest -1
<speed test begins>
fcron_speedtest_arm_sched()-405: Speed test (0x5561aa61f0) for port1 will run in 33 s
...
[speedtest(15619)] speed test Done.
fcron_speedtest_notify_func()-1570: Speed test pid=15619 done
fcron_speedtest_on_test_finish()-1530: test 0x00630000 for 'port1' succeed with up=638101,
down=930393
fcron_sptest_cloud_report_result()-715: test(0x00630000): if=port1, server=FTNT_CA_Burnaby
(154.52.1.124:5213), uuid=6a82a804-45b1-477f-b94b-f4a21e17596 report result=0, up=638101,
dw=930393
fcron_sptest_cloud_result_reply()-663: test(0x00630000): if=port1, uuid=6a82a804-45b1-477f-
b94b-f4a21e17596 report result succeed (status=2).
```

5. Verify that the results have been applied to the interface:

```
# diagnose netlink interface list port1
if=port1 alias= family=00 type=1 index=99 mtu=1500 link=0 master=0
ref=28 state=start present fw_flags=10010400 flags=up broadcast run multicast
Qdisc=noqueue hw_addr=e0:23:ff:3e:7f:eb broadcast_addr=ff:ff:ff:ff:ff:ff
ingress traffic control:
        bandwidth=930393(kbps) lock_hit=0 default_class=2 n_active_class=3
...
egress traffic control:
        bandwidth=638101(kbps) lock_hit=0 default_class=2 n_active_class=3
```

Example 5: executing multiple speed tests with TCP and UDP connections

A speed test is executed to the closest server using 64 TCP connections and another speed test is executed using one UDP connection. The results can be checked with a third-party platform (such as Ookla), which returns comparable results.



To execute multiple speed tests with TCP and UDP connections:

1. Edit the speed test settings:

```
config system speed-test-setting
    set multiple-tcp-stream 64
end
```

2. Run the TCP speed test:

```
# execute speed-test port1 FTNT_CA_Vancouver TCP
...
Run in uploading mode.
...
[SUM]   0.00-5.00   sec   559 MBytes   938 Mbits/sec  2165             sender
[SUM]   0.00-5.01   sec   558 MBytes   933 Mbits/sec                  receiver

speed test Done.
Run in reverse downloading mode.
...
[SUM]   0.00-5.01   sec   505 MBytes   846 Mbits/sec  9329             sender
[SUM]   0.00-5.00   sec   491 MBytes   823 Mbits/sec                  receiver
```

3. Run the UDP speed test:

```
# execute speed-test port1 FTNT_CA_Vancouver UDP
...
Run in uploading mode.
...
[  7]   0.00-5.00   sec   556 MBytes   933 Mbits/sec  0.000 ms  0/402727 (0%)  sender
[  7]   0.00-5.04   sec   556 MBytes   925 Mbits/sec  0.020 ms  393/402717 (0.098%)  receiver
...
Run in reverse downloading mode.
...
[  7]   0.00-5.04   sec   869 MBytes  1.45 Gbits/sec  0.000 ms  0/629383 (0%)  sender
[SUM]  0.0- 5.0 sec  2 datagrams received out-of-order
[  7]   0.00-5.00   sec   489 MBytes   821 Mbits/sec  0.005 ms  274103/628393 (44%)  receiver

speed test Done.
```

Example 6: executing a manual speed test for debugging

This example provides the steps to enable debug output and manually initiate a speed-test. The debug process is reset to avoid debug output from other applications.

To execute a manual speed test with debugging:

1. Enable debug for speed test:

```
# diagnose debug reset
# diagnose debug application speedtest -1
# diagnose debug enable
```

2. Execute a manual speed test and review the debug output:

```
# diagnose netlink interface speed-test port1 FTNT_US_Miami UDP
speed-test test ID is 630000
[speedtest(8798)] [ 26]   1.00-2.00   sec  96.2 MBytes   807 Mbits/sec  69680
[speedtest(8798)] [ 26]   2.00-3.00   sec  96.5 MBytes   809 Mbits/sec  69848
[speedtest(8798)] [ 26]   3.00-4.00   sec  96.2 MBytes   807 Mbits/sec  69670
[speedtest(8798)] [ 26]   4.00-5.00   sec  96.4 MBytes   809 Mbits/sec  69830
[speedtest(8798)] [ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total
Datagrams
[speedtest(8798)] [ 26]   0.00-5.00   sec   480 MBytes   804 Mbits/sec  0.000 ms  0/347479
(0%)  sender
[speedtest(8798)] [ 26]   0.00-5.13   sec   260 MBytes   425 Mbits/sec  0.015 ms
159066/347240 (46%)  receiver
[speedtest(8798)] client(sender): bytes_recv=272475952, bytes_sent=503149592, sender_
time=5.004, recver_time=5.126
[speedtest(8798)] client(sender): up_speed:  425 Mbits/sec
[speedtest(8798)]
[speedtest(8798)] speed test Done.
[speedtest(8798)] Run in reverse downloading mode.
[speedtest(8798)] Connecting to host 208.97.222.31, port 5200
[speedtest(8798)] Reverse mode, remote host 208.97.222.31 is sending
[speedtest(8798)] [ 26] local 172.16.206.1 port 17904 connected to 208.97.222.31 port 5200
[speedtest(8798)] [ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total
Datagrams
[speedtest(8798)] [ 26]   0.00-1.00   sec  11.7 MBytes  97.4 Mbits/sec  0.071 ms  56797/65238
(87%)
[speedtest(8798)] [ 26]   1.00-2.00   sec  11.4 MBytes  95.8 Mbits/sec  0.057 ms
100853/109121 (92%)
[speedtest(8798)] [ 26]   2.00-3.00   sec  11.6 MBytes  97.7 Mbits/sec  0.077 ms
104724/113158 (93%)
[speedtest(8798)] [ 26]   3.00-4.00   sec  11.7 MBytes  97.8 Mbits/sec  0.064 ms
106371/114809 (93%)
[speedtest(8798)] [ 26]   4.00-5.00   sec  11.7 MBytes  98.0 Mbits/sec  0.013 ms
108068/116528 (93%)
[speedtest(8798)] [ ID] Interval           Transfer     Bitrate         Jitter    Lost/Total
Datagrams
[speedtest(8798)] [ 26]   0.00-5.12   sec   731 MBytes  1.20 Gbits/sec  0.000 ms  0/529586
(0%)  sender
[speedtest(8798)] [SUM]  0.0- 5.1 sec  1 datagrams received out-of-order
[speedtest(8798)] [ 26]   0.00-5.00   sec  58.1 MBytes  97.3 Mbits/sec  0.013 ms
476813/518854 (92%)  receiver
[speedtest(8798)] client(recver): bytes_recv=60875368, bytes_sent=766840528, sender_
time=5.121, recver_time=5.004
[speedtest(8798)] client(recver): down_speed: 97.3 Mbits/sec
[speedtest(8798)]
[speedtest(8798)] speed test Done.
fcron_speedtest_notify_func()-1592: Speed test pid=8798 done

fcron_speedtest_on_test_finish()-1519: Test 630000 for 'port1' succeed with up=425252,
down=97319
```

3. Disable and reset the debug once finished:

```
# diagnose debug reset
# diagnose debug disable
```

Overlay

The following topics provide instructions on SD-WAN VPN overlays:

- VPN on page 1277
- ADVPN on page 1283
- ADVPN 2.0 on page 1342
- Self-healing and application protection on page 1382

VPN

This section introduces some options available when creating your overlay VPN tunnels, such as the VPN tunnel wizard, and SD-WAN Overlay-as-a-Service. VPN enhancements, such as Forward Error Correction, are introduced as options to increase resiliency or provide QoS traffic shaping:

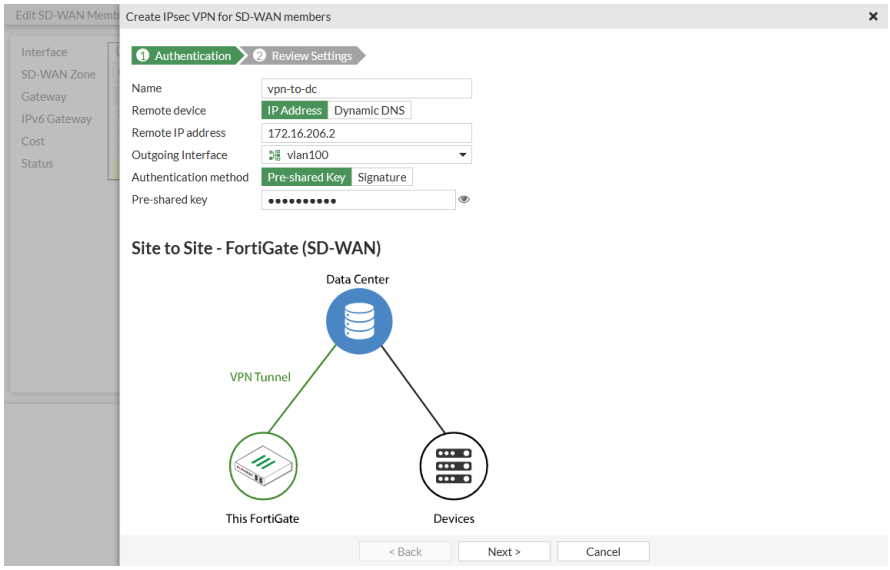
- Dual VPN tunnel wizard on page 1277
- SD-WAN Overlay-as-a-Service on page 1278
- Interface based QoS on individual child tunnels based on speed test results on page 1280

Dual VPN tunnel wizard

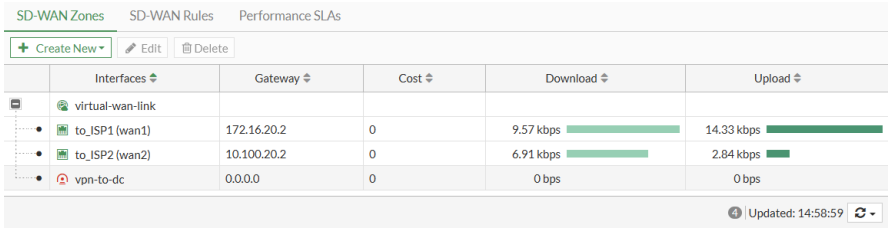
This wizard is used to automatically set up multiple VPN tunnels to the same destination over multiple outgoing interfaces. This includes automatically configuring IPsec, routing, and firewall settings, avoiding cumbersome and error-prone configuration steps.

To create a new SD-WAN VPN interface using the tunnel wizard:

1. Go to *Network* > *SD-WAN*, select the *SD-WAN Zones* tab, and click *Create New* > *SD-WAN Member*.
2. In the *Interface* drop-down, click *+VPN*. The *Create IPsec VPN for SD-WAN members* pane opens.



3. Enter the required information, then click *Next*.
4. Review the settings then click *Create*.
5. Click *Close* to return to the SD-WAN page.
   The newly created VPN interface will be highlighted in the *Interface* drop-down list.
6. Select the VPN interface to add it as an SD-WAN member, then click *OK*.



SD-WAN Overlay-as-a-Service

The FortiCloud Overlay-as-a-Service portal and the FortiGate Cloud Advanced license support SD-WAN overlay.

SD-WAN overlay is supported through an Overlay-as-a-Service (OaaS) license displayed as *SD-WAN Overlay as a Service* on the *System* > *FortiGuard* page. Each FortiGate used by the FortiCloud Overlay-as-a-Service or FortiGate Cloud portal for SD-WAN overlay must have this license applied to it.

See the Overlay-as-a-Service Administration Guide and SD-WAN Overlay in the FortiGate Cloud Administration Guide for more information on SD-WAN overlay.

To view the status of the OaaS license in the GUI:

1. Go to *System* > *FortiGuard*.
2. Expand *License Information*. The *SD-WAN Overlay as a Service* license status is listed as:
   - *Licensed*: OaaS is currently licensed and will expire on the provided date.
   - *Expires Soon*: OaaS is currently licensed but will expire soon on the provided date.
   - *Expired*: The OaaS license has already expired on the provided date.
   - *Not Licensed*: OaaS has not been licensed.

To view the status of the OaaS license in the CLI:

1. Verify that the entitlement can be updated:

   The SD-WAN Overlay-as-a-Service license is listed as `SWOS` in the CLI.

```
# diagnose test update info

System contracts:
    FMWR,Wed Dec 20 16:00:00 2023
    SPAM,Wed Dec 20 16:00:00 2023
    SBCL,Wed Dec 20 16:00:00 2023
    SWNO,Wed Dec 20 16:00:00 2023
    SWNM,Wed Sep 27 17:00:00 2023
    SWOS,Mon Aug 14 17:00:00 2023
    SPRT,Wed Dec 20 16:00:00 2023
    SDWN,Sun Dec 10 16:00:00 2023
    SBCL,Wed Dec 20 16:00:00 2023
    SBEN,Wed Dec 20 16:00:00 2023
```

2. Verify that the expiration date log can be generated:

```
# execute log display

1: date=2023-08-10 time=00:00:01 eventtime=1691650800645347120 tz="-0700" logid="0100020138"
type="event" subtype="system" level="warning" vd="root" logdesc="FortiGuard SD-WAN Overlay as
a Service license expiring" msg="FortiGuard SD-WAN Overlay Service license will expire in 4
day(s)"
```

To ensure FortiGate spoke traffic remains uninterrupted when configuration is orchestrated from SD-WAN Overlay-as-a-Service (OaaS) or FortiGate Cloud, support for an OaaS agent on the FortiGate is available. The OaaS agent communicates with the OaaS controller in FortiCloud, validates and compares the FortiOS configuration, and applies the FortiOS configuration to the FortiGate as a transaction when it has been orchestrated from the OaaS or FortiGate Cloud portal. Secure communication between the OaaS agent and the OaaS controller is achieved using the FGFM management tunnel.

If any configuration change fails to be applied, then the OaaS agent rolls back all configuration changes that were orchestrated. The OaaS status can be acquired using `get oaas status`.

To determine the status of OaaS:

```
# get oaas status
Account ID: 78992
Account: admin@domain.com
Site: site1
Configuration version: 4
Configuration sync status: SUCCESS
    Target version: 4
    Task ID: xxxxxxxxx
    Error:
```

Interface based QoS on individual child tunnels based on speed test results

In a hub and spoke SD-WAN topology that uses dial-up VPN overlays, QoS can be applied on individual tunnels based on the measured bandwidth between the hub and spokes. The FortiGate can use the built in speed test to dynamically populate the egress bandwidth to individual dial-up tunnels from the hub.

A bandwidth limit, derived from the speed test, and a traffic shaping profile can be applied on the dial-up IPsec tunnel interface on the hub. A class ID and percentage based QoS settings can be applied to individual child tunnels using a traffic shaping policy and profile.

CLI commands

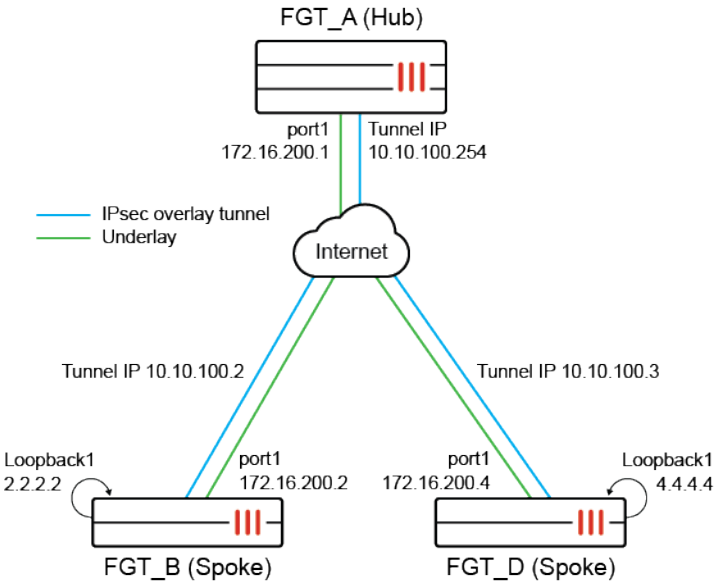
If the interface is an IPsec dial-up server, then egress shaping profile type can only be set to `policing`; it cannot be set to `queuing`:

```
config firewall shaping-profile
    edit <profile-name>
        set type policing
    next
end
```

The outbandwidth value is dynamically obtained from the speed test results for each individual child tunnel, and should not be set manually:

```
config system interface
    edit <dialup-server-phase1-name>
        set egress-shaping-profile <profile-name>
        set outbandwidth <bandwidth>
    next
end
```

Example



In this example, the hub is configured as a VPN dial-up server and both of the spokes are connected to the hub. It is assumed that the VPN configuration is already done, with a dynamic gateway type and kernel device creation (`net-device`) disabled. Only one SD-WAN interface is used, so there is only one VPN overlay member in the SD-WAN zone. Multiple WAN interfaces and VPN overlays could be used.

The VPN interfaces and IP addresses are:

FortiGate	Interface	IP Address
FGT_A (Hub)	hub-phase1	10.10.100.254
FGT_B (Spoke)	spoke11-p1	10.10.100.2
FGT_D (Spoke)	spoke21-p1	10.10.100.3

The hub VPN has two child tunnels, one to each spoke.

The speed test configuration is shown in Running speed tests from the hub to the spokes in dial-up IPsec tunnels on page 1241. This example shows applying a shaping profile to the hub's tunnel interface in order to apply interface based traffic shaping to the child tunnels.

A traffic shaping policy is used to match and assign traffic to the classes in the shaping profile.

To configure the hub FortiGate (FGT_A) and check the results:

1. Configure the hub FortiGate (FGT_A) as in Running speed tests from the hub to the spokes in dial-up IPsec tunnels on page 1241.
2. Configure the shaping profile:

```
config firewall shaping-profile
    edit "profile_1"
        config shaping-entries
            edit 1
```

```
                set class-id 2
                set priority low
                set guaranteed-bandwidth-percentage 10
                set maximum-bandwidth-percentage 10
            next
            edit 2
                set class-id 3
                set priority medium
                set guaranteed-bandwidth-percentage 30
                set maximum-bandwidth-percentage 40
            next
            edit 3
                set class-id 4
                set priority high
                set guaranteed-bandwidth-percentage 20
                set maximum-bandwidth-percentage 60
            next
        end
        set default-class-id 2
    next
end
```

3. Configure a traffic shaping policy:

```
config firewall shaping-policy
    edit 2
        set service "ALL"
        set schedule "always"
        set dstintf "hub-phase1"
        set class-id 3
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

In this example, all traffic through the hub-phase1 interface is put into class ID 3. Class IDs an be assigned based on your traffic requirements.

4. At the schedules time, the speed test will start for the hub-phase1 interface from the hub to the spokes. The speed test results can then be dynamically applied on individual child tunnels as egress traffic shaping, and the class ID percentage based QoS settings is applicable on them as templates.

```
# diagnose vpn tunnel list
------------------------------------------------------
name=hub-phase1_0 ver=2 serial=c 172.16.200.1:0->172.16.200.4:0 tun_id=172.16.200.4 dst_
mtu=1500 dpd-link=on remote_location=0.0.0.0 weight=1
...
egress traffic control:
        bandwidth=737210(kbps) lock_hit=0 default_class=2 n_active_class=3
        class-id=2      allocated-bandwidth=73720(kbps)         guaranteed-bandwidth=73720
(kbps)
                        max-bandwidth=73720(kbps)       current-bandwidth=0(kbps)
                        priority=low    forwarded_bytes=52
```

```
                        dropped_packets=0       dropped_bytes=0
        class-id=3      allocated-bandwidth=221163(kbps)        guaranteed-bandwidth=221162
(kbps)
                        max-bandwidth=294883(kbps)      current-bandwidth=0(kbps)
                        priority=medium         forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
        class-id=4      allocated-bandwidth=442325(kbps)        guaranteed-bandwidth=147441
(kbps)
                        max-bandwidth=442325(kbps)      current-bandwidth=0(kbps)
                        priority=high   forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
-----------------------------------------------------
name=hub-phase1_1 ver=2 serial=d 172.16.200.1:0->172.16.200.2:0 tun_id=172.16.200.2 dst_
mtu=1500 dpd-link=on remote_location=0.0.0.0 weight=1
...
egress traffic control:
        bandwidth=726813(kbps) lock_hit=0 default_class=2 n_active_class=3
        class-id=2      allocated-bandwidth=72681(kbps)         guaranteed-bandwidth=72681
(kbps)
                        max-bandwidth=72681(kbps)       current-bandwidth=0(kbps)
                        priority=low    forwarded_bytes=123
                        dropped_packets=0       dropped_bytes=0
        class-id=3      allocated-bandwidth=218044(kbps)        guaranteed-bandwidth=218043
(kbps)
                        max-bandwidth=290725(kbps)      current-bandwidth=0(kbps)
                        priority=medium         forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
        class-id=4      allocated-bandwidth=436087(kbps)        guaranteed-bandwidth=145362
(kbps)
                        max-bandwidth=436087(kbps)      current-bandwidth=0(kbps)
                        priority=high   forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
```

The guaranteed and maximum bandwidths equal 10% of the speed test result, as expected.

ADVPN

ADVPN (Auto Discovery VPN) is an IPsec technology that allows a traditional hub-and-spoke VPN's spokes to establish dynamic, on-demand, direct tunnels between each other to avoid routing through the topology's hub device. The primary advantage is that it provides full meshing capabilities to a standard hub-and-spoke topology. This greatly reduces the provisioning effort for full spoke-to-spoke low delay reachability and addresses the scalability issues associated with very large fully meshed VPN networks.

The following topics provide information on ADVPN for SD-WAN:

- ADVPN and shortcut paths on page 1284
- Active dynamic BGP neighbor triggered by ADVPN shortcut on page 1297
- SD-WAN monitor on ADVPN shortcuts on page 1308
- Hold down time to support SD-WAN service strategies on page 1309
- Keeping sessions in established ADVPN shortcuts while they remain in SLA on page 1311

- Using a single IKE selector in ADVPN to match all SD-WAN control plane traffic on page 1322
- SD-WAN in large scale deployments on page 1330

ADVPN and shortcut paths

This topic provides an example of how to use SD-WAN and ADVPN together.

ADVPN (Auto Discovery VPN) is an IPsec technology that allows a traditional hub-and-spoke VPN's spokes to establish dynamic, on-demand, direct tunnels between each other to avoid routing through the topology's hub device. The primary advantage is that it provides full meshing capabilities to a standard hub-and-spoke topology. This greatly reduces the provisioning effort for full spoke-to-spoke low delay reachability, and addresses the scalability issues associated with very large fully meshed VPN networks.

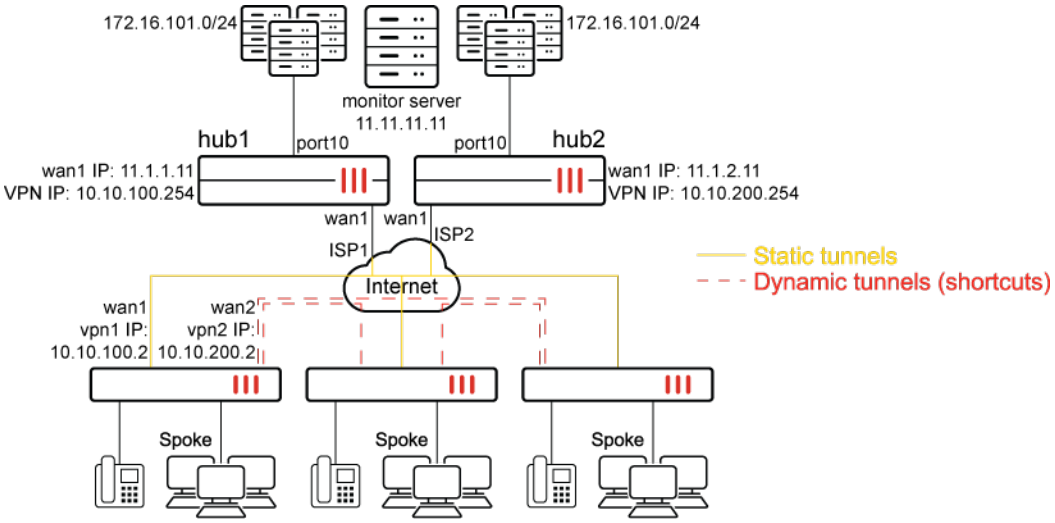
If a customer's head office and branch offices all have two or more internet connections, they can build a dual-hub ADVPN network. Combined with SD-WAN technology, the customer can load-balance traffic to other offices on multiple dynamic tunnels, control specific traffic using specific connections, or choose better performance connections dynamically.

SD-WAN load-balance mode rules (or services) do not support ADVPN members. Other modes' rules, such as SLA and priority, support ADVPN members.

This topic covers three parts:

1. Configure dual-hub ADVPN with multiple branches.
2. Configure BGP to exchange routing information among hubs and spokes.
3. Configure SD-WAN on spoke to do load-balancing and control traffic.

Configuration example



A typical ADVPN configuration with SD-WAN usually has two hubs, and each spoke connects to two ISPs and establishes VPN tunnels with both hubs.

This example shows a hub-and-spoke configuration using two hubs and one spoke:

- Hub1 and Hub2 both use wan1 to connect to the ISPs and port10 to connect to internal network.
- Spoke1 uses wan1 to connect to ISP1 and wan2 to connect to ISP2.
- wan1 sets up VPN to hub1.
- wan2 sets up VPN to hub2.

The SD-WAN is configured on the spoke. It uses the two VPN interfaces as members and two rules to control traffic to headquarters or other spokes using ADVPN VPN interfaces. You can create more rules if required.

For this example:

- Use SD-WAN member 1 (via ISP1) and its dynamic shortcuts for financial department traffic if member 1 meets SLA requirements. If it doesn't meet SLA requirements, it will use SD-WAN member 2 (via ISP2).
- Use SD-WAN member 2 (via ISP2) and its dynamic shortcuts for engineering department traffic.
- Load balance other traffic going to hubs and other spokes between these two members.
- Set up all other traffic to go with their original ISP connection. All other traffic does not go through SD-WAN.
- Set up basic network configuration to let all hubs and spokes connect to their ISPs and the Internet.

Hub internal network	172.16.101.0/24
Spoke1 internal network	10.1.100.0/24
ADVPN 1 network	10.10.100.0/24
ADVPN 2 network	10.10.200.0/24
Hub1 wan1 IP	11.1.1.11
Hub2 wan1 IP	11.1.2.11
Hub1 VPN IP	10.10.100.254
Hub2 VPN IP	10.10.200.254
Spoke1 to hub1 VPN IP	10.10.100.2
Spoke1 to hub2 VPN IP	10.10.200.2
Ping server in Headquarters	11.11.11.11
Internal subnet of spoke1	22.1.1.0/24
Internal subnet of spoke2	33.1.1.0/24
Firewall addresses	Configure hub_subnets and spoke_subnets before using in policies. These can be customized.

The GUI does not support some ADVPN related options, such as auto-discovery-sender, auto-discovery-receiver, auto-discovery-forwarder, and IBGP neighbor-group setting, so this example only provides CLI configuration commands.

Hub1 sample configuration

To configure the IPsec phase1 and phase2 interface:

```
config vpn ipsec phase1-interface
    edit "hub-phase1"
        set type dynamic
        set interface "wan1"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 3des-sha256 aes128-sha1 aes256-sha1 3des-sha1
        set add-route disable
        set dpd on-idle
        set auto-discovery-sender enable
        set psksecret sample
        set dpd-retryinterval 5
    next
end
config vpn ipsec phase2-interface
    edit "hub-phase2"
        set phase1name "hub-phase1"
        set proposal aes128-sha1 aes256-sha1 3des-sha1 aes128-sha256 aes256-sha256 3des-sha256
    next
end
```

When `net-device` is disabled, a tunnel ID is generated for each dynamic tunnel. This ID, in the form of an IP address, is used as the gateway in the route entry to that tunnel. The `tunnel-search` option is removed in FortiOS 7.0.0 and later.

To configure the VPN interface and BGP:

```
config system interface
    edit "hub-phase1"
        set ip 10.10.100.254 255.255.255.255
        set remote-ip 10.10.100.253 255.255.255.0
    next
end
config router bgp
    set as 65505
    config neighbor-group
        edit "advpn"
            set link-down-failover enable
            set remote-as 65505
            set route-reflector-client enable
        next
    end
    config neighbor-range
        edit 1
            set prefix 10.10.100.0 255.255.255.0
```

```
            set neighbor-group "advpn"
        next
    end
    config network
        edit 1
            set prefix 172.16.101.0 255.255.255.0
        next
        edit 2
            set prefix 11.11.11.0 255.255.255.0
         next
    end
end
```

To configure the firewall policy:

```
config firewall policy
    edit 1
        set name "spoke2hub"
        set srcintf "hub-phase1"
        set dstintf "port10"
        set srcaddr "spoke_subnets"
        set dstaddr "hub_subnets"
        set action accept
        set schedule "always"
        set service "ALL"
        set comments "allow traffic from spokes to headquarter"
    next
    edit 2
        set name "spoke2spoke"
        set srcintf "hub-phase1"
        set dstintf "hub-phase1"
        set srcaddr "spoke_subnets"
        set dstaddr "spoke_subnets"
        set action accept
        set schedule "always"
        set service "ALL"
        set comments "allow traffic from spokes to spokes"
    next
    edit 3
        set name "internal2spoke"
        set srcintf "port10"
        set dstintf "hub-phase1"
        set srcaddr "hub_subnets"
        set dstaddr "spoke_subnets"
        set action accept
        set schedule "always"
        set service "ALL"
        set comments "allow traffic from headquarter to spokes"
    next
end
```

Hub2 sample configuration

Hub2 configuration is the same as hub1 except the wan1 IP address, VPN interface IP address, and BGP neighbor-range prefix.

To configure the IPsec phase1 and phase2 interface:

```
config vpn ipsec phase1-interface
    edit "hub-phase1"
        set type dynamic
        set interface "wan1"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 3des-sha256 aes128-sha1 aes256-sha1 3des-sha1
        set add-route disable
        set dpd on-idle
        set auto-discovery-sender enable
        set psksecret sample
        set dpd-retryinterval 5
    next
end
config vpn ipsec phase2-interface
    edit "hub-phase2"
        set phase1name "hub-phase1"
        set proposal aes128-sha1 aes256-sha1 3des-sha1 aes128-sha256 aes256-sha256 3des-sha256
    next
end
```

To configure the VPN interface and BGP:

```
config system interface
    edit "hub-phase1"
        set ip 10.10.200.254 255.255.255.255
        set remote-ip 10.10.200.253 255.255.255.0
    next
end
config router bgp
    set as 65505
    config neighbor-group
        edit "advpn"
            set link-down-failover enable
            set remote-as 65505
            set route-reflector-client enable
        next
    end
    config neighbor-range
        edit 1
            set prefix 10.10.200.0 255.255.255.0
            set neighbor-group "advpn"
        next
```

```
    end
    config network
        edit 1
            set prefix 172.16.101.0 255.255.255.0
        next
        edit 2
            set prefix 11.11.11.0 255.255.255.0
         next
    end
end
```

To configure the firewall policy:

```
config firewall policy
    edit 1
        set name "spoke2hub"
        set srcintf "hub-phase1"
        set dstintf "port10"
        set srcaddr "spoke_subnets"
        set dstaddr "hub_subnets"
        set action accept
        set schedule "always"
        set service "ALL"
        set comments "allow traffic from spokes to headquarter"
    next
    edit 2
        set name "spoke2spoke"
        set srcintf "hub-phase1"
        set dstintf "hub-phase1"
        set srcaddr "spoke_subnets"
        set dstaddr "spoke_subnets"
        set action accept
        set schedule "always"
        set service "ALL"
        set comments "allow traffic from spokes to spokes"
    next
    edit 3
        set name "internal2spoke"
        set srcintf "port10"
        set dstintf "hub-phase1"
        set srcaddr "hub_subnets"
        set dstaddr "spoke_subnets"
        set action accept
        set schedule "always"
        set service "ALL"
        set comments "allow traffic from headquarter to spokes"
    next
end
```

Spoke1 sample configuration

To configure the IPsec phase1 and phase2 interface:

```
config vpn ipsec phase1-interface
    edit "spoke1-phase1"
        set interface "wan1"
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set add-route disable
        set dpd on-idle
        set auto-discovery-receiver enable
        set remote-gw 11.1.1.11
        set psksecret sample
        set dpd-retryinterval 5
    next
    edit "spoke1-2-phase1"
        set interface "wan2"
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set add-route disable
        set dpd on-idle
        set auto-discovery-receiver enable
        set remote-gw 11.1.2.11
        set psksecret sample
        set dpd-retryinterval 5
    next
end
config vpn ipsec phase2-interface
    edit "spoke1-phase2"
        set phase1name "spoke1-phase1"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
        set auto-negotiate enable
    next
    edit "spoke1-2-phase2"
        set phase1name "spoke1-2-phase1"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
        set auto-negotiate enable
    next
end
```

To configure the VPN interface and BGP:

```
config system interface
    edit "spoke1-phase1"
        set ip 10.10.100.2 255.255.255.255
        set remote-ip 10.10.100.254 255.255.255.0
```

```
    next
    edit "spoke1-2-phase1"
        set ip 10.10.200.2 255.255.255.255
        set remote-ip 10.10.200.254 255.255.255.0
    next
end
config router bgp
    set as 65505
    config neighbor
        edit "10.10.100.254"
            set advertisement-interval 1
            set link-down-failover enable
            set remote-as 65505
        next
        edit "10.10.200.254"
            set advertisement-interval 1
            set link-down-failover enable
            set remote-as 65505
        next
    end
    config network
        edit 1
            set prefix 10.1.100.0 255.255.255.0
        next
    end
end
```

To configure SD-WAN:

```
config system sdwan
    set status enable
    config members
        edit 1
            set interface "spoke1-phase1"
        next
        edit 2
            set interface "spoke1-2-phase1"
        next
    end
    config health-check
        edit "ping"
            set server "11.11.11.11"
            set members 1 2
            config sla
                edit 1
                    set latency-threshold 200
                    set jitter-threshold 50
                    set packetloss-threshold 5
                next
            end
        next
    end
```

```
    config service
        edit 1
            set mode sla
            set dst "financial-department"
            config sla
                edit "ping"
                    set id 1
                next
            end
            set priority-members 1 2
        next
        edit 2
            set priority-members 2
            set dst "engineering-department"
        next
    end
end
```

If no SD-WAN zone is specified, members are added to the default *virtual-wan-link* zone.

To configure the firewall policy:

```
config firewall policy
    edit 1
        set name "outbound_advpn"
        set srcintf "internal"
        set dstintf "virtual-wan-link"
        set srcaddr "spoke_subnets"
        set dstaddr "spoke_subnets" "hub_subnets"
        set action accept
        set schedule "always"
        set service "ALL"
        set comments "allow internal traffic going out to headquarter and other spokes"
    next
    edit 2
        set name "inbound_advpn"
        set srcintf "virtual-wan-link"
        set dstintf "internal"
        set srcaddr "spoke_subnets" "hub_subnets"
        set dstaddr "spoke_subnets"
        set action accept
        set schedule "always"
        set service "ALL"
        set comments "allow headquarter and other spokes traffic coming in"
    next
end
```

Troubleshooting ADVPN and shortcut paths

Before spoke vs spoke shortcut VPN is established

Use the following CLI commands to check status before spoke vs spoke shortcut VPN is established.

```
# get router info bgp summary
BGP router identifier 2.2.2.2, local AS number 65505
BGP table version is 13
3 BGP AS-PATH entries
0 BGP community entries

Neighbor        V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.10.100.254   4      65505    3286    3270       11    0    0 00:02:15        5
10.10.200.254   4      65505    3365    3319       12    0    0 00:02:14        5

Total number of neighbors 2
```

```
# get router info routing-table bgp

Routing table for VRF=0
B*      0.0.0.0/0 [200/0] via 10.10.200.254, spoke1-2-phase1, 00:00:58
                  [200/0] via 10.10.100.254, spoke1-phase1, 00:00:58
B       1.1.1.1/32 [200/0] via 11.1.1.1 (recursive via 12.1.1.1), 00:01:29
                   [200/0] via 11.1.1.1 (recursive via 12.1.1.1), 00:01:29
B       11.11.11.0/24 [200/0] via 10.10.200.254, spoke1-2-phase1, 00:01:29
                      [200/0] via 10.10.100.254, spoke1-phase1, 00:01:29
B       33.1.1.0/24 [200/0] via 10.10.200.3, spoke1-2-phase1, 00:00:58
                    [200/0] via 10.10.100.3, spoke1-phase1, 00:00:58
                    [200/0] via 10.10.200.3, spoke1-2-phase1, 00:00:58
                    [200/0] via 10.10.100.3, spoke1-phase1, 00:00:58
```

```
# diagnose vpn tunnel list
list all ipsec tunnel in vd 3
------------------------------------------------------
name=spoke1-phase1 ver=1 serial=5 12.1.1.2:0->11.1.1.11:0 tun_id=11.1.1.11 dst_mtu=15324
bound_if=48 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/536 options[0218]=npu create_dev frag-
rfc  accept_traffic=1

proxyid_num=1 child_num=0 refcnt=22 ilast=0 olast=0 ad=r/2
stat: rxp=1 txp=185 rxb=16428 txb=11111
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=4
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=spoke1 proto=0 sa=1 ref=4 serial=1 auto-negotiate adr
  src: 0:0.0.0.0/0.0.0.0:0
  dst: 0:0.0.0.0/0.0.0.0:0
  SA:  ref=6 options=1a227 type=00 soft=0 mtu=15262 expire=42820/0B replaywin=2048
       seqno=ba esn=0 replaywin_lastseq=00000002 itn=0 qat=0
  life: type=01 bytes=0/0 timeout=42903/43200
  dec: spi=03e01a2a esp=aes key=16 56e673f0df05186aa657f55cbb631c13
       ah=sha1 key=20 b0d50597d9bed763c42469461b03da8041f87e88
```

```
  enc: spi=2ead61bc esp=aes key=16 fe0ccd4a3ec19fe6d520c437eb6b8897
       ah=sha1 key=20 e3e669bd6df41b88eadaacba66463706f26fb53a
  dec:pkts/bytes=1/16368, enc:pkts/bytes=185/22360
  npu_flag=03 npu_rgwy=11.1.1.11 npu_lgwy=12.1.1.2 npu_selid=0 dec_npuid=1 enc_npuid=1
------------------------------------------------------
name=spoke1-2-phase1 ver=1 serial=6 112.1.1.2:0->11.1.2.11:0 tun_id=11.1.2.11 dst_mtu=15324
bound_if=90 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/536 options[0218]=npu create_dev frag-
rfc  accept_traffic=1

proxyid_num=1 child_num=0 refcnt=21 ilast=0 olast=0 ad=r/2
stat: rxp=1 txp=186 rxb=16498 txb=11163
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=74
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=spoke1-2 proto=0 sa=1 ref=4 serial=1 auto-negotiate adr
  src: 0:0.0.0.0/0.0.0.0:0
  dst: 0:0.0.0.0/0.0.0.0:0
  SA:  ref=6 options=1a227 type=00 soft=0 mtu=15262 expire=42818/0B replaywin=2048
       seqno=bb esn=0 replaywin_lastseq=00000002 itn=0 qat=0
  life: type=01 bytes=0/0 timeout=42901/43200
  dec: spi=03e01a2b esp=aes key=16 fe49f5042a5ad236250bf53312db1346
       ah=sha1 key=20 5dbb15c8cbc046c284bb1c6425dac2b3e15bec85
  enc: spi=2ead61bd esp=aes key=16 d6d97be52c3cccb9e88f28a9db64ac46
       ah=sha1 key=20 e20916ae6ea2295c2fbd5cbc8b8f5dd8b17f52f1
  dec:pkts/bytes=1/16438, enc:pkts/bytes=186/22480
  npu_flag=03 npu_rgwy=11.1.2.11 npu_lgwy=112.1.1.2 npu_selid=1 dec_npuid=1 enc_npuid=1
```

```
# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x0
  TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla)
  Member sub interface:
  Members:
    1: Seq_num(1), alive, sla(0x1), cfg_order(0), cost(0), selected
    2: Seq_num(2), alive, sla(0x1), cfg_order(1), cost(0), selected
  Dst address: 33.1.1.1-33.1.1.100

Service(2): Address Mode(IPV4) flags=0x0
  TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
  Member sub interface:
  Members:
    1: Seq_num(2), alive, selected
  Dst address: 33.1.1.101-33.1.1.200
```

```
# diagnose firewall proute list
list route policy info(vf=vd2):

id=2132869121 vwl_service=1 vwl_mbr_seq=1 dscp_tag=0xff 0xff flags=0x0 tos=0x00 tos_mask=0x00
protocol=0 sport=0:65535 iif=0 dport=1-65535 oif=70 oif=71
destination(1): 33.1.1.1-33.1.1.100
source wildcard(1): 0.0.0.0/0.0.0.0

id=2132869122 vwl_service=2 vwl_mbr_seq=2 dscp_tag=0xff 0xff flags=0x0 tos=0x00 tos_mask=0x00
```

```
protocol=0 sport=0:65535 iif=0 dport=1-65535 oif=71
destination(1): 33.1.1.101-33.1.1.200
source wildcard(1): 0.0.0.0/0.0.0.0
```

After spoke vs spoke shortcut VPN is established

Use the following CLI commands to check status after spoke vs spoke shortcut VPN is established.

```
# get router info routing-table bgp

Routing table for VRF=0
B*      0.0.0.0/0 [200/0] via 10.10.200.254, spoke1-2-phase1, 00:01:33
                  [200/0] via 10.10.100.254, spoke1-phase1, 00:01:33
B       1.1.1.1/32 [200/0] via 11.1.1.1 (recursive via 12.1.1.1), 00:02:04
                   [200/0] via 11.1.1.1 (recursive via 12.1.1.1), 00:02:04
B       11.11.11.0/24 [200/0] via 10.10.200.254, spoke1-2-phase1, 00:02:04
                      [200/0] via 10.10.100.254, spoke1-phase1, 00:02:04
B       33.1.1.0/24 [200/0] via 10.10.200.3, spoke1-2-phase1_0, 00:01:33
                    [200/0] via 10.10.100.3, spoke1-phase1_0, 00:01:33
                    [200/0] via 10.10.200.3, spoke1-2-phase1_0, 00:01:33
                    [200/0] via 10.10.100.3, spoke1-phase1_0, 00:01:33
```

```
# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x0
  TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla)
  Member sub interface:
    1: seq_num(1), interface(spoke1-phase1):
       1: spoke1-phase1_0(111)
    2: seq_num(2), interface(spoke1-2-phase1):
       1: spoke1-2-phase1_0(113)
  Members:
    1: Seq_num(1), alive, sla(0x1), cfg_order(0), cost(0), selected
    2: Seq_num(2), alive, sla(0x1), cfg_order(1), cost(0), selected
  Dst address: 33.1.1.1-33.1.1.100

Service(2): Address Mode(IPV4) flags=0x0
  TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
  Member sub interface:
    1: seq_num(2), interface(spoke1-2-phase1):
       1: spoke1-2-phase1_0(113)
  Members:
    1: Seq_num(2), alive, selected
  Dst address: 33.1.1.101-33.1.1.200
```

```
# diagnose vpn tunnel list
list all ipsec tunnel in vd 3
------------------------------------------------------
name=spoke1-phase1 ver=1 serial=5 12.1.1.2:0->11.1.1.11:0 tun_id=11.1.1.11 dst_mtu=15324
bound_if=48 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/536 options[0218]=npu create_dev frag-
rfc  accept_traffic=1
```

```
proxyid_num=1 child_num=1 refcnt=20 ilast=0 olast=0 ad=r/2
stat: rxp=1 txp=759 rxb=16428 txb=48627
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=4
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=vd2-1 proto=0 sa=1 ref=5 serial=1 auto-negotiate adr
  src: 0:0.0.0.0/0.0.0.0:0
  dst: 0:0.0.0.0/0.0.0.0:0
  SA:  ref=6 options=1a227 type=00 soft=0 mtu=15262 expire=42536/0B replaywin=2048
       seqno=2f8 esn=0 replaywin_lastseq=00000002 itn=0 qat=0
  life: type=01 bytes=0/0 timeout=42898/43200
  dec: spi=03e01a42 esp=aes key=16 1f131bda108d33909d49fc2778bd08bb
       ah=sha1 key=20 14131d3f0da9b741a2fd13d530b0553aa1f58983
  enc: spi=2ead61d8 esp=aes key=16 81ed24d5cd7bb59f4a80dceb5a560e1f
       ah=sha1 key=20 d2ccc2f3223ce16514e75f672cd88c4b4f48b681
  dec:pkts/bytes=1/16360, enc:pkts/bytes=759/94434
  npu_flag=03 npu_rgwy=11.1.1.11 npu_lgwy=12.1.1.2 npu_selid=0 dec_npuid=1 enc_npuid=1
-----------------------------------------------------
name=spoke1-2-phase1 ver=1 serial=6 112.1.1.2:0->11.1.2.11:0 tun_id=11.1.2.11 dst_mtu=15324
bound_if=90 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/536 options[0218]=npu create_dev frag-
rfc  accept_traffic=1

proxyid_num=1 child_num=1 refcnt=19 ilast=0 olast=0 ad=r/2
stat: rxp=1 txp=756 rxb=16450 txb=48460
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=74
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=vd2-2 proto=0 sa=1 ref=5 serial=1 auto-negotiate adr
  src: 0:0.0.0.0/0.0.0.0:0
  dst: 0:0.0.0.0/0.0.0.0:0
  SA:  ref=6 options=1a227 type=00 soft=0 mtu=15262 expire=42538/0B replaywin=2048
       seqno=2f5 esn=0 replaywin_lastseq=00000002 itn=0 qat=0
  life: type=01 bytes=0/0 timeout=42900/43200
  dec: spi=03e01a43 esp=aes key=16 7fc87561369f88b56d08bfda769eb45b
       ah=sha1 key=20 0ed554ef231c5ac16dc2e71d1907d7347dda33d6
  enc: spi=2ead61d9 esp=aes key=16 00286687aa1762e7d8216881d6720ef3
       ah=sha1 key=20 59d5eec6299ebcf038c190860774e2833074d7c3
  dec:pkts/bytes=1/16382, enc:pkts/bytes=756/94058
  npu_flag=03 npu_rgwy=11.1.2.11 npu_lgwy=112.1.1.2 npu_selid=1 dec_npuid=1 enc_npuid=1
-----------------------------------------------------
name=spoke1-phase1_0 ver=1 serial=55 12.1.1.2:0->13.1.1.3:0 tun_id=13.1.1.3 dst_mtu=15324
bound_if=48 lgwy=static/1 tun=intf/0 mode=dial_inst/3 encap=none/728 options[02d8]=npu create_dev
no-sysctl rgwy-chg frag-rfc  accept_traffic=1

 parent=vd2-1 index=0
proxyid_num=1 child_num=0 refcnt=18 ilast=8 olast=8 ad=r/2
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=vd2-1 proto=0 sa=1 ref=2 serial=1 auto-negotiate adr
  src: 0:0.0.0.0/0.0.0.0:0
  dst: 0:0.0.0.0/0.0.0.0:0
  SA:  ref=3 options=1a227 type=00 soft=0 mtu=15262 expire=42893/0B replaywin=2048
```

```
      seqno=1 esn=0 replaywin_lastseq=00000000 itn=0 qat=0
  life: type=01 bytes=0/0 timeout=42901/43200
  dec: spi=03e01a44 esp=aes key=16 c3b77a98e3002220e2373b73af14df6e
       ah=sha1 key=20 d18d107c248564933874f60999d6082fd7a78948
  enc: spi=864f6dba esp=aes key=16 eb6181806ccb9bac37931f9eadd4d5eb
       ah=sha1 key=20 ab788f7a372877a5603c4ede1be89a592fc21873
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=13.1.1.3 npu_lgwy=12.1.1.2 npu_selid=51 dec_npuid=0 enc_npuid=0
------------------------------------------------------
name=spoke1-2-phase1_0 ver=1 serial=57 112.1.1.2:0->113.1.1.3:0 tun_id=113.1.1.3 dst_mtu=15324
bound_if=90 lgwy=static/1 tun=intf/0 mode=dial_inst/3 encap=none/728 options[02d8]=npu create_dev
no-sysctl rgwy-chg frag-rfc  accept_traffic=1

 parent=vd2-2 index=0
proxyid_num=1 child_num=0 refcnt=17 ilast=5 olast=5 ad=r/2
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=vd2-2 proto=0 sa=1 ref=3 serial=1 auto-negotiate adr
  src: 0:0.0.0.0/0.0.0.0:0
  dst: 0:0.0.0.0/0.0.0.0:0
  SA:  ref=3 options=1a227 type=00 soft=0 mtu=15262 expire=42900/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 itn=0 qat=0
  life: type=01 bytes=0/0 timeout=42901/43200
  dec: spi=03e01a45 esp=aes key=16 0beb519ed9f800e8b4c0aa4e1df7da35
       ah=sha1 key=20 bc9f38db5296cce4208a69f1cc8a9f7ef4803c37
  enc: spi=864f6dbb esp=aes key=16 1d26e3556afcdb9f8e3e33b563b44228
       ah=sha1 key=20 564d05ef6f7437e1fd0a88d5fee7b6567f9d387e
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=113.1.1.3 npu_lgwy=112.1.1.2 npu_selid=53 dec_npuid=0 enc_npuid=0
```

```
# diagnose firewall proute list
list route policy info(vf=vd2):

id=2132869121 vwl_service=1 vwl_mbr_seq=1 dscp_tag=0xff 0xff flags=0x0 tos=0x00 tos_mask=0x00
protocol=0 sport=0:65535 iif=0 dport=1-65535 oif=111 oif=70 oif=113 oif=71
destination(1): 33.1.1.1-33.1.1.100
source wildcard(1): 0.0.0.0/0.0.0.0

id=2132869122 vwl_service=2 vwl_mbr_seq=2 dscp_tag=0xff 0xff flags=0x0 tos=0x00 tos_mask=0x00
protocol=0 sport=0:65535 iif=0 dport=1-65535 oif=113 oif=71
destination(1): 33.1.1.101-33.1.1.200
source wildcard(1): 0.0.0.0/0.0.0.0
```

Active dynamic BGP neighbor triggered by ADVPN shortcut

When a customer using SD-WAN with ADVPN has numerous IPv4 and IPv6 routes per spoke and there are many spokes in the topology, using ADVPN with a route reflector-based design poses the following challenges:

- The hub FortiGate will experience high CPU usage due to the amount of processing required to reflect the routes to the spoke FortiGates.

- Spoke FortiGates will learn many unnecessary routes.

For such cases, it is more suitable to deploy an IPv4- and IPv6-supported solution without a route-reflector that involves an active dynamic BGP neighbor triggered by an ADVPN shortcut. This solution allows a spoke FortiGate to form a BGP neighbor with another spoke FortiGate only after the shortcut tunnel between them has been established. As a result, the spoke only learns routes from its BGP neighbors.

How this solution differs from typical SD-WAN with ADVPN

In a topology where the Spoke 1 and Spoke 2 FortiGates are connected directly to the Hub FortiGate, route reflection will not be enabled. The Hub FortiGate is only configured with each spoke's summary route. An ADVPN shortcut tunnel is established between the Spoke 1 and Spoke 2 FortiGates. The valid routing between the Spoke 1 and Spoke 2 FortiGate is still through the Hub FortiGate at this point.

When a host behind Spoke 1 tries to connect to a host behind Spoke 2, Spoke 1 first reaches the Hub based on the valid routing table. The Hub determines that the destination is reachable, and the ADVPN shortcut tunnel between the spokes is established. Then, Spoke 1 and Spoke 2 will actively initiate a BGP connection to each other over the shortcut. Once established, they will exchange their routing information using BGP. On both spokes, BGP will resolve those routes on the shortcut and update the routing table accordingly.

For this solution, the following IPv4/IPv6 BGP configuration settings are required:

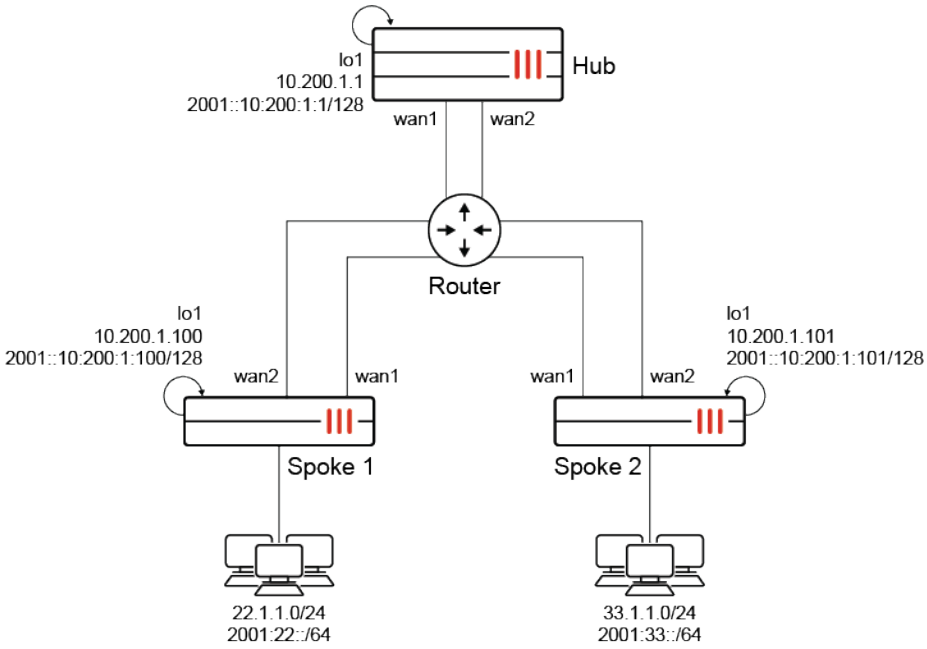
- The hub FortiGate should be configured with `neighbor-group` and `neighbor-range`/`neighbor-range6`.
- Each spoke FortiGate should be configured with `neighbor-group` and `neighbor-range`/`neighbor-range6` like the hub. More importantly, each spoke should be configured with `set passive disable` to ensure spokes are able to initiate dynamic BGP connections between each other.
- The hub FortiGate should have route reflection disabled (by default) where each `neighbor-group` setting should have `set route-reflector-client disable`.

In the configuration, each of the spokes will form a BGP neighbor relationship with the hub. This is unchanged from the typical SD-WAN with ADVPN configuration.

Example

This example configuration contains the following structure:

- Use SD-WAN member 1 (via ISP1) and its dynamic shortcuts for Financial Department traffic.
- Use SD-WAN member 2 (via ISP2) and its dynamic shortcuts for Engineering Department traffic.
- Internal subnets of Spoke 1:
  - IPv4: 22.1.1.0/24
  - IPv6: 2001:22::/64
- Internal subnets of Spoke 2:
  - IPv4: 33.1.1.0/24
    - Financial Department: 33.1.1.1 to 33.1.1.100
    - Engineering Department: 33.1.1.101 to 33.1.1.200
  - IPv6: 2001:33::/64
    - Financial Department: 2001:33::1 to 2001:33::100
    - Engineering Department: 2001:33::101 to 2001:33::200



To configure the Hub FortiGate:

1. Configure the BGP settings (neighbor group and ranges):

```
config router bgp
    set as 65100
    set router-id 10.200.1.1
    set ibgp-multipath enable
    config neighbor-group
        edit "EDGE"
            set activate6 disable
            set remote-as 65100
            set update-source "lo1"
            set route-reflector-client disable
        next
        edit "EDGEv6"
            set activate disable
            set remote-as 65100
            set update-source "lo1"
            set route-reflector-client disable
        next
    end
    config neighbor-range
        edit 2
            set prefix 10.200.1.0 255.255.255.0
            set neighbor-group "EDGE"
        next
    end
    config neighbor-range6
        edit 2
            set prefix6 2001::10:200:1:0/112
```

```
            set neighbor-group "EDGEv6"
        next
    end
    config network
        edit 2
            set prefix 10.200.1.0 255.255.255.0
        next
        edit 4
            set prefix 33.0.0.0 255.0.0.0
        next
        edit 5
            set prefix 22.0.0.0 255.0.0.0
        next
    end
    config network6
        edit 4
            set prefix6 2001:33::/32
        next
        edit 2
            set prefix6 2001:22::/32
        next
    end
end
```

2. Configure the static routes.
   a. For IPv4:

```
config router static
    edit 33
        set dst 33.0.0.0 255.0.0.0
        set blackhole enable
        set vrf 0
    next
    edit 22
        set dst 22.0.0.0 255.0.0.0
        set blackhole enable
        set vrf 0
    next
end
```

   b. For IPv6:

```
config router static6
    edit 33
        set dst 2001:33::/32
        set blackhole enable
        set vrf 0
    next
    edit 22
        set dst 2001:22::/32
        set blackhole enable
        set vrf 0
```

```
    next
end
```

The following IPv4 summary routes are advertised:

- 33.0.0.0/8
- 22.0.0.0/8

The following IPv6 summary routes are advertised:

- 2001:33::/32
- 2001:22::/32

Because route reflection has been disabled in this example, initially, Spoke 1 will not know the local subnet of Spoke 2, and Spoke 2 will not know the local subnet of Spoke 1. Therefore, for traffic routing, summary routes are configured on the hub as blackhole routes and then advertised to the spokes using BGP.

For example, for traffic from the local subnet of Spoke 2 destined for the local subnet of Spoke 1:

- For the IPv4 case, the summary route 22.0.0.0/8, which includes the local subnet of Spoke 1 (22.1.1.0/24), is advertised to Spoke 2. When Spoke 2 sends traffic destined for 22.1.1.0/24 to the Hub, the Hub forwards this traffic to Spoke 1 since they are BGP neighbors.
- For the IPv6 case, the summary route 2001:22::/32, which includes the local subnet of Spoke 1 (2001:22::/64), is advertised to Spoke 2. When Spoke 2 sends traffic destined for 2001:22::/64 to the Hub, the Hub forwards this traffic to Spoke 1 since they are BGP neighbors.

Although traffic from spoke-to-spoke goes through the hub first, it is expected that the spoke will eventually go through the shortcut tunnel.

To configure the Spoke 1 FortiGate:

1. Configure the SD-WAN settings:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "spoke1-1"
            set cost 10
        next
        edit 2
            set interface "spoke-2"
            set cost 20
        next
    end
    config health-check
        edit "ping"
            set server "11.11.11.11"
            set source 10.200.1.100
            set members 1 2
```

```
            config sla
                edit 1
                    set latency-threshold 200
                    set jitter-threshold 50
                next
            end
        next
    end
    config service
        edit 1
            set dst "financial-department"
            set priority-members 1
        next
        edit 2
            set dst "engineering-department"
            set priority-members 2
        next
        edit 61
            set addr-mode ipv6
            set priority-members 1
            set dst6 "financial-department-IPv6"
        next
        edit 62
            set addr-mode ipv6
            set priority-members 2
            set dst6 "engineering-department-IPv6"
        next
    end
end
```

2. Configure the BGP settings (neighbor group and ranges):

```
config router bgp
    set as 65100
    set router-id 10.200.1.100
    set ibgp-multipath enable
    config neighbor
        edit "10.200.1.1"
            set activate6 disable
            set remote-as 65100
            set connect-timer 10
            set update-source "lo1"
        next
        edit "2001::10:200:1:1"
            set advertisement-interval 1
            set activate disable
            set remote-as 65100
            set update-source "lo1"
        next
    end
    config neighbor-group
        edit "spokes"
```

```
            set activate6 disable
            set passive disable
            set remote-as 65100
            set update-source "lo1"
        next
        edit "spokesv6"
            set activate disable
            set passive disable
            set remote-as 65100
            set update-source "lo1"
        next
    end
    config neighbor-range
        edit 1
            set prefix 10.200.1.0 255.255.255.0
            set neighbor-group "spokes"
        next
    end
    config neighbor-range6
        edit 1
            set prefix6 2001::10:200:1:0/112
            set neighbor-group "spokesv6"
        next
    end
    config network
        edit 3
            set prefix 22.1.1.0 255.255.255.0
        next
    end
    config network6
        edit 1
            set prefix6 2001:22::/64
        next
    end
end
```

Verifying the configuration before a spoke-to-spoke shortcut VPN is established

IPv4 use case

To verify the status on Spoke 1:

1. Verify the BGP status:

```
# get router info bgp summary
VRF 0 BGP router identifier 10.200.1.100, local AS number 65100
BGP table version is 5
1 BGP AS-PATH entries
0 BGP community entries
Neighbor   V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
```

```
10.200.1.1 4      65100     222     225        3    0    0 00:15:14        3
Total number of neighbors 1
```

2. Verify the BGP routing table:

```
# get router info routing-table bgp
Routing table for VRF=0
B       11.11.11.11/32 [200/0] via 10.200.1.1 (recursive via spoke1-1 tunnel 11.1.1.11),
00:15:19
                                              (recursive via spoke1-2 tunnel 111.1.1.11),
00:15:19, [1/0]
B       22.0.0.0/8 [200/0] via 10.200.1.1 (recursive via spoke1-1 tunnel 11.1.1.11), 00:15:19
                                          (recursive via spoke1-2 tunnel 111.1.1.11),
00:15:19, [1/0]
B       33.0.0.0/8 [200/0] via 10.200.1.1 (recursive via spoke1-1 tunnel 11.1.1.11), 00:15:19
                                          (recursive via spoke1-2 tunnel 111.1.1.11),
00:15:19, [1/0]
```

IPv6 use case

To verify the status on Spoke 1:

1. Verify the BGP status:

```
# get router info6 bgp summary
VRF 0 BGP router identifier 10.200.1.100, local AS number 65100
BGP table version is 6
1 BGP AS-PATH entries
0 BGP community entries
Neighbor         V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2001::10:200:1:1 4      65100     223     224        4    0    0 00:15:21        3
Total number of neighbors 1
```

2. Verify the BGP routing table:

```
# get router info6 routing-table bgp
Routing table for VRF=0
B       2001::11:11:11:11/128 [200/0] via 2001::10:200:1:1 (recursive via spoke1-1 tunnel
::11.1.1.11), 00:15:29
                                                           (recursive via spoke1-2 tunnel
::111.1.1.11), 00:15:29, [1024/0]
B       2001:22::/32 [200/0] via 2001::10:200:1:1 (recursive via spoke1-1 tunnel ::11.1.1.11),
00:15:29
                                                  (recursive via spoke1-2 tunnel
::111.1.1.11), 00:15:29, [1024/0]
B       2001:33::/32 [200/0] via 2001::10:200:1:1 (recursive via spoke1-1 tunnel ::11.1.1.11),
00:15:29
                                                  (recursive via spoke1-2 tunnel
::111.1.1.11), 00:15:29, [1024/0]
```

Verifying the configuration after a single spoke-to-spoke shortcut VPN is established

IPv4 use case

To trigger a single spoke-to-spoke shortcut VPN, on host 22.1.1.22, ping the host 33.1.1.33 in the Financial Department. Because of the SD-WAN rule, use SD-WAN member 1 (via ISP1) and its dynamic shortcuts to reach hosts in the Financial Department.

To verify the status on Spoke 1:

1. Verify the BGP status:

```
# get router info bgp summary
VRF 0 BGP router identifier 10.200.1.100, local AS number 65100
BGP table version is 6
1 BGP AS-PATH entries
0 BGP community entries
Neighbor     V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.200.1.1   4      65100     252     254        3    0    0 00:17:22        3
10.200.1.101 4      65100       6       6        5    0    0 00:00:14        1
Total number of neighbors 2
```

   Spoke 1 has as its BGP neighbors:
   - Hub FortiGate at 10.200.1.1
   - Spoke 2 FortiGate at 10.200.1.101

2. Verify the BGP routing table:

```
# get router info routing-table bgp
Routing table for VRF=0
B       11.11.11.11/32 [200/0] via 10.200.1.1 (recursive via spoke1-1 tunnel 11.1.1.11),
00:17:26
                                              (recursive via spoke1-2 tunnel 111.1.1.11),
00:17:26, [1/0]
B       22.0.0.0/8 [200/0] via 10.200.1.1 (recursive via spoke1-1 tunnel 11.1.1.11), 00:17:26
                                          (recursive via spoke1-2 tunnel 111.1.1.11),
00:17:26, [1/0]
B       33.0.0.0/8 [200/0] via 10.200.1.1 (recursive via spoke1-1 tunnel 11.1.1.11), 00:17:26
                                          (recursive via spoke1-2 tunnel 111.1.1.11),
00:17:26, [1/0]
B       33.1.1.0/24 [200/0] via 10.200.1.101 (recursive via spoke1-1_0 tunnel 13.1.1.3),
00:00:18, [1/0]
```

   The remote route learned from Spoke 2 through the spoke1_1_0 tunnel and using BGP is 33.1.1.0/24.

IPv6 use case

To trigger a single spoke-to-spoke shortcut VPN over IPv6, on host 2001:22::22/64, ping the host 2001:33::33/64 in the Financial Department. Because of the SD-WAN rule, use SD-WAN member 1 (via ISP1) and its dynamic shortcuts to reach hosts in the Financial Department.

To verify the status on Spoke 1:

1. Verify the BGP status:

```
# get router info6 bgp summary
VRF 0 BGP router identifier 10.200.1.100, local AS number 65100
BGP table version is 7
1 BGP AS-PATH entries
0 BGP community entries
Neighbor           V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2001::10:200:1:1   4      65100     253     254        4    0    0 00:17:28        3
2001::10:200:1:101 4      65100       7       7        6    0    0 00:00:21        1
Total number of neighbors 2
```

   Spoke 1 has as its BGP neighbors:
   - Hub FortiGate at 2001::10:200:1:1
   - Spoke 2 FortiGate at 2001::10:200:1:101

2. Verify the BGP routing table:

```
# get router info6 routing-table bgp
Routing table for VRF=0
B       2001::11:11:11:11/128 [200/0] via 2001::10:200:1:1 (recursive via spoke1-1 tunnel
::11.1.1.11), 00:17:30
                                                           (recursive via spoke1-2 tunnel
::111.1.1.11), 00:17:30, [1024/0]
B       2001:22::/32 [200/0] via 2001::10:200:1:1 (recursive via spoke1-1 tunnel ::11.1.1.11),
00:17:30
                                                  (recursive via spoke1-2 tunnel
::111.1.1.11), 00:17:30, [1024/0]
B       2001:33::/32 [200/0] via 2001::10:200:1:1 (recursive via spoke1-1 tunnel ::11.1.1.11),
00:17:30
                                                  (recursive via spoke1-2 tunnel
::111.1.1.11), 00:17:30, [1024/0]
B       2001:33::/64 [200/0] via 2001::10:200:1:101 (recursive via spoke1-1_0 tunnel
::13.1.1.3), 00:00:24, [1024/0]
```

   The remote route learned from Spoke 2 through the spoke1-1_0 tunnel and using BGP is 2001:33::/64.

Verifying the configuration after a second spoke-to-spoke shortcut VPN is established

IPv4 use case

To trigger a second spoke-to-spoke shortcut VPN, on host 22.1.1.22, ping the host 33.1.1.133 in the Engineering Department. Because of the SD-WAN rule, use SD-WAN member 2 (via ISP2) and its dynamic shortcuts to reach hosts in the Engineering Department.

To verify the status on Spoke 1:

1. Verify the BGP status:

```
# get router info bgp summary
VRF 0 BGP router identifier 10.200.1.100, local AS number 65100
BGP table version is 6
1 BGP AS-PATH entries
0 BGP community entries
Neighbor     V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.200.1.1   4      65100     263     265        3    0    0 00:18:12        3
10.200.1.101 4      65100      17      17        5    0    0 00:01:04        1
Total number of neighbors
```

   Spoke 1 continues to have its BGP neighbors:
   - Hub FortiGate at 10.200.1.1
   - Spoke 2 FortiGate at 10.200.1.101

2. Verify the BGP routing table:

```
# get router info routing-table bgp
Routing table for VRF=0
B       11.11.11.11/32 [200/0] via 10.200.1.1 (recursive via spoke1-1 tunnel 11.1.1.11),
00:18:17
                                              (recursive via spoke1-2 tunnel 111.1.1.11),
00:18:17, [1/0]
B       22.0.0.0/8 [200/0] via 10.200.1.1 (recursive via spoke1-1 tunnel 11.1.1.11), 00:18:17
                                          (recursive via spoke1-2 tunnel 111.1.1.11),
00:18:17, [1/0]
B       33.0.0.0/8 [200/0] via 10.200.1.1 (recursive via spoke1-1 tunnel 11.1.1.11), 00:18:17
                                          (recursive via spoke1-2 tunnel 111.1.1.11),
00:18:17, [1/0]
B       33.1.1.0/24 [200/0] via 10.200.1.101 (recursive via spoke1-1_0 tunnel 13.1.1.3),
00:01:09
                                             (recursive via spoke1-2_0 tunnel 113.1.1.3),
00:01:09, [1/0]
```

   The remote route learned from Spoke 2 through the spoke1-2_0 tunnel and using BGP is 33.1.1.0/24.

IPv6 use case

To trigger a second spoke-to-spoke shortcut VPN over IPv6, on host 2001:22::22/64, ping the host 2001:33::133/64 in the Engineering Department. Because of the SD-WAN rule, use SD-WAN member 2 (via ISP2) and its dynamic shortcuts to reach hosts in the Engineering Department.

To verify the status on Spoke 1:

1. Verify the BGP status:

```
# get router info6 bgp summary
VRF 0 BGP router identifier 10.200.1.100, local AS number 65100
```

```
BGP table version is 7
1 BGP AS-PATH entries
0 BGP community entries
Neighbor           V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
2001::10:200:1:1   4      65100     264     265        4    0    0 00:18:18        3
2001::10:200:1:101 4      65100      19      19        6    0    0 00:01:11        1
Total number of neighbors 2
```

Spoke 1 continues to have its BGP neighbors:

- Hub FortiGate at 2001::10:200:1:1
- Spoke 2 FortiGate at 2001::10:200:1:101

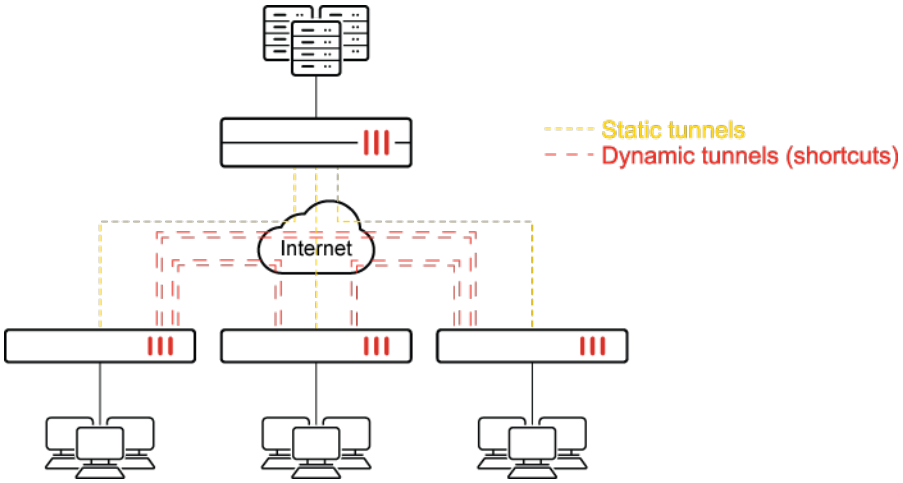
2. Verify the BGP routing table:

```
# get router info6 routing-table bgp
Routing table for VRF=0
B       2001::11:11:11:11/128 [200/0] via 2001::10:200:1:1 (recursive via spoke1-1 tunnel
::11.1.1.11), 00:18:20
                                                           (recursive via spoke1-2 tunnel
::111.1.1.11), 00:18:20, [1024/0]
B       2001:22::/32 [200/0] via 2001::10:200:1:1 (recursive via spoke1-1 tunnel ::11.1.1.11),
00:18:20
                                                  (recursive via spoke1-2 tunnel
::111.1.1.11), 00:18:20, [1024/0]
B       2001:33::/32 [200/0] via 2001::10:200:1:1 (recursive via spoke1-1 tunnel ::11.1.1.11),
00:18:20
                                                  (recursive via spoke1-2 tunnel
::111.1.1.11), 00:18:20, [1024/0]
B       2001:33::/64 [200/0] via 2001::10:200:1:101 (recursive via spoke1-1_0 tunnel
::13.1.1.3), 00:01:14
                                                    (recursive via spoke1-2_0 tunnel
::113.1.1.3), 00:01:14, [1024/0]
```

The remote route learned from Spoke 2 through the spoke1-2_0 tunnel and using BGP is 2001:33::/64.

SD-WAN monitor on ADVPN shortcuts

SD-WAN monitors ADVPN shortcut link quality by dynamically creating link monitors for each ADVPN link. The dynamic link monitor on the spoke will use ICMP probes and the IP address of the gateway as the monitored server. These ICMP probes will not be counted as actual user traffic that keeps the spoke-to-spoke tunnel alive.



- When no shortcut is established:

```
# diagnose sys sdwan health-check
Health Check(ping):
Seq(1 tunnel-1): state(alive), packet-loss(0.000%) latency(0.038), jitter(0.006) sla_map=0x3
Seq(2 tunnel-2): state(alive), packet-loss(0.000%) latency(0.035), jitter(0.004) sla_map=0x3
```

- When one shortcut is established:

```
# diagnose sys sdwan health-check
Health Check(ping):
Seq(1 tunnel-1): state(alive), packet-loss(0.000%) latency(0.039), jitter(0.003) sla_map=0x3
Seq(1 tunnel-1_0): state(alive), packet-loss(0.000%) latency(0.060), jitter(0.023) sla_map=0x3
Seq(2 tunnel-2): state(alive), packet-loss(0.000%) latency(0.035), jitter(0.002) sla_map=0x3
```

- When more than one shortcut is established:

```
# diagnose sys sdwan health-check
Health Check(ping):
Seq(1 tunnel-1): state(alive), packet-loss(0.000%) latency(0.036), jitter(0.004) sla_map=0x3
Seq(1 tunnel-1_0): state(alive), packet-loss(0.000%) latency(0.041), jitter(0.009) sla_map=0x3
Seq(2 tunnel-2): state(alive), packet-loss(0.000%) latency(0.030), jitter(0.005) sla_map=0x3
Seq(2 tunnel-2_0): state(alive), packet-loss(0.000%) latency(0.031), jitter(0.004) sla_map=0x3
```

Hold down time to support SD-WAN service strategies

In a hub and spoke SD-WAN topology with shortcuts created over ADVPN, a downed or recovered shortcut can affect which member is selected by an SD-WAN service strategy. When a downed shortcut tunnel recovers and the shortcut is added back into the service strategy, the shortcut is held at a low priority until the hold down time has elapsed.

By default, the hold down time is zero seconds. It can be set to 0 - 10000000 seconds.

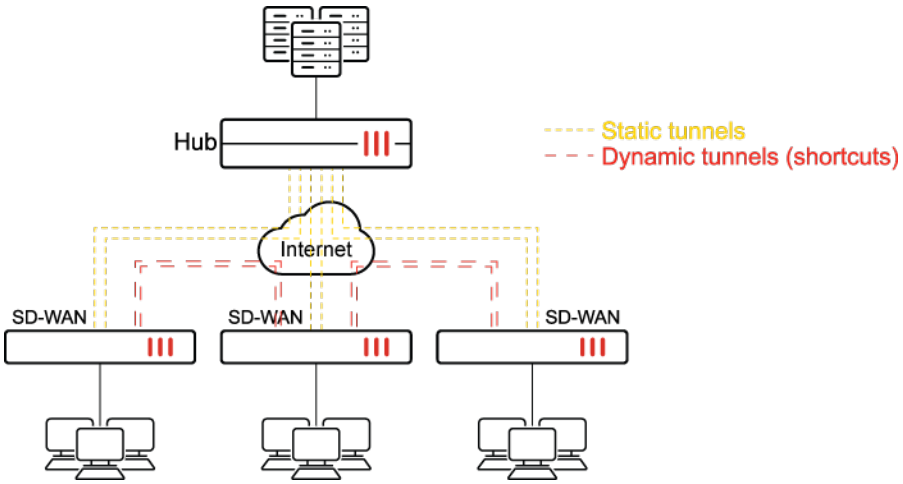
The hold down time also applies to the primary VPN tunnels between hub and spoke devices.

To configure the hold down time:

```
config system sdwan
    config service
        edit 1
            set hold-down-time <integer>
        next
    end
end
```

Example

In this example, the hold down time is set to 15 seconds, and then the SD-WAN service is looked at before and after the hold down elapses after a downed shortcut recovers.



To configure the hold down time:

```
config system sdwan
    config service
        edit 1
            set hold-down-time 15
        next
    end
end
```

To view which SD-WAN member is selected before and after the hold down time elapses:

Before the hold down time has elapsed:

```
# diagnose sys sdwan service4
Service(1): Address Mode(IPV4) flags=0x200
  Gen(34), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(priority), link-cost-factor(packet-loss),
link-cost-threshold(0), health-check(ping)
Hold down time(15) seconds, Hold start at 2003 second, now 2010
  Member sub interface(4):
```

```
    1: seq_num(1), interface(vd2-1):
       1: vd2-1_0(86)
    3: seq_num(2), interface(vd2-2):
       1: vd2-2_0(88)

  Members(4):
    1: Seq_num(1 vd2-1), alive, packet loss: 27.000%, selected
    2: Seq_num(2 vd2-2_0), alive, packet loss: 0.000%, selected
    3: Seq_num(2 vd2-2), alive, packet loss: 0.000%, selected
    4: Seq_num(1 vd2-1_0), alive, packet loss: 61.000%, selected
  Dst address(1):
        33.1.1.101-33.1.1.200
```

After the hold down time has elapsed:

```
# diagnose sys sdwan service4
Service(1): Address Mode(IPV4) flags=0x200
  Gen(35), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(priority), link-cost-factor(packet-loss),
link-cost-threshold(0), health-check(ping)
Hold down time(15) seconds, Hold start at 2018 second, now 2019
  Member sub interface(4):

    2: seq_num(2), interface(vd2-2):
       1: vd2-2_0(88)
    3: seq_num(1), interface(vd2-1):
       1: vd2-1_0(86)
  Members(4):
    1: Seq_num(2 vd2-2_0), alive, packet loss: 0.000%, selected
    2: Seq_num(2 vd2-2), alive, packet loss: 0.000%, selected
    3: Seq_num(1 vd2-1), alive, packet loss: 24.000%, selected
    4: Seq_num(1 vd2-1_0), alive, packet loss: 44.000%, selected
  Dst address(1):
        33.1.1.101-33.1.1.200\
```

Keeping sessions in established ADVPN shortcuts while they remain in SLA

In an SD-WAN hub and spoke configuration where ADVPN is used, when a primary shortcut goes out of SLA, traffic switches to the backup shortcut. During idle timeout, sessions will prefer using the primary parent tunnel and try to establish a new primary shortcut. However, because it is out of SLA, traffic switches back to the backup shortcut, which causes unnecessary traffic interruption.

The `sla-stickiness` option keeps existing sessions on the established ADVPN shortcuts while they remain in SLA instead of switching to a new link every idle timeout. New sessions will be routed through the primary shortcut if it is in SLA.

```
config system sdwan
    config service
        edit <id>
            set mode sla
            set sla-stickiness {enable | disable}
```

```
        next
    end
end
```

The `sla-stickiness` option can be applied in the following use cases.

Use case 1:

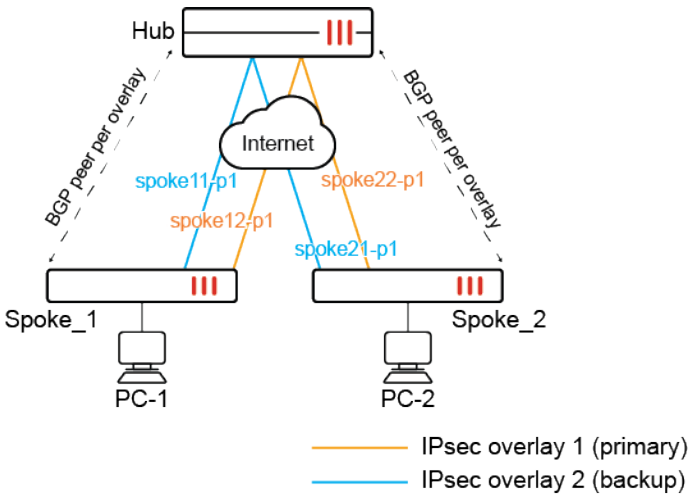
1. The sessions will switch over to the backup shortcut due to the primary shortcut being out of SLA.
2. After an idle timeout, the primary shortcut is torn down, and the routes will be reinstalled on the primary parent tunnel.
3. When `sla-stickiness` is enabled, even though the primary parent tunnel is preferred, established ADVPN sessions will remain on the backup shortcut (stickiness) instead of switching to the primary parent tunnel.
4. New sessions will be routed to the primary parent tunnel and trigger the primary shortcut, then traffic switches to the primary shortcut if it is in SLA.

Use case 2:

1. The sessions will switch over to the backup shortcut due to the primary shortcut being out of SLA.
2. After some time, the primary shortcut becomes in SLA.
3. When `sla-stickiness` is enabled, even though primary shortcut is preferred, established ADVPN sessions will remain on the backup shortcut (stickiness) instead of switching to the primary shortcut.
4. New sessions will be routed through the primary shortcut.

Example topology

The following examples demonstrates using the `sla-stickiness` option.



After an idle timeout occurs, existing sessions remain on the spoke12-p1_0 backup shortcut tunnel. New sessions will try to create a shortcut over spoke11-p1, but will fall back to spoke12-p1_0 when it detects spoke11-p1 is out of SLA.

Only use case 2 is applicable in ADVPN 2.0 with shortcut stickiness.

Use case 1 example

To configure shortcut stickiness for ADVPN shortcuts:

1. Configure SD-WAN on the Spoke_1 FortiGate:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "spoke11-p1"
        next
        edit 2
            set interface "spoke12-p1"
        next
    end
    config health-check
        edit "1"
            set server "9.0.0.1"
            set members 1 2
            config sla
                edit 1
                next
            end
        next
    end
    config service
        edit 1
            set name "1"
            set mode sla
            set sla-stickiness enable
            set dst "all"
            set src "10.1.100.0"
            config sla
                edit "1"
                    set id 1
                next
            end
            set priority-members 1 2
        next
    end
end
```

2. Verify the SD-WAN configuration.
   a. Verify the health check status:

```
# diagnose sys sdwan health-check
Health Check(1):
```

```
Seq(1 spoke11-p1): state(alive), packet-loss(0.000%) latency(0.368), jitter(0.051), mos
(4.404), bandwidth-up(999999), bandwidth-dw(1000000), bandwidth-bi(1999999) sla_map=0x1
Seq(2 spoke12-p1): state(alive), packet-loss(0.000%) latency(0.211), jitter(0.019), mos
(4.404), bandwidth-up(999999), bandwidth-dw(999979), bandwidth-bi(1999978) sla_map=0x1
```

b. Verify the service status:

```
# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x2200 use-shortcut-sla sla-stickiness
 Tie break: cfg
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Members(2):
    1: Seq_num(1 spoke11-p1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    2: Seq_num(2 spoke12-p1), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
  Src address(1):
        10.1.100.0-10.1.100.255

  Dst address(1):
        0.0.0.0-255.255.255.255
```

The SD-WAN service rule prefers the primary parent tunnel (spoke11-p1) over the backup parent tunnel (spoke12-p1) before shortcuts are established.

3. Send traffic from PC-1 to PC-2 to trigger the primary shortcut. Verify the diagnostics.

a. Run a sniffer trace:

```
# diagnose sniffer packet any 'host 192.168.5.44' 4
interfaces=[any]
filters=[host 192.168.5.44]
14.878761 port2 in 10.1.100.22 -> 192.168.5.44: icmp: echo request
14.878905 spoke11-p1 out 10.1.100.22 -> 192.168.5.44: icmp: echo request
14.879842 spoke11-p1 in 192.168.5.44 -> 10.1.100.22: icmp: echo reply
14.880082 port2 out 192.168.5.44 -> 10.1.100.22: icmp: echo reply
15.879761 port2 in 10.1.100.22 -> 192.168.5.44: icmp: echo request
15.879882 spoke11-p1_0 out 10.1.100.22 -> 192.168.5.44: icmp: echo request
15.880433 spoke11-p1_0 in 192.168.5.44 -> 10.1.100.22: icmp: echo reply
15.880496 port2 out 192.168.5.44 -> 10.1.100.22: icmp: echo reply
```

The SD-WAN service rule sends traffic to the parent tunnel (spoke11-p1) initially, and then switches to the primary shortcut tunnel (spoke11-p1_0) once it is established.

b. Verify the service status:

```
# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x2200 use-shortcut-sla sla-stickiness
 Tie break: cfg
  Gen(2), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Member sub interface(3):
    2: seq_num(1), interface(spoke11-p1):
```

```
      1: spoke11-p1_0(57)
  Members(3):
    1: Seq_num(1 spoke11-p1_0), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    2: Seq_num(1 spoke11-p1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    3: Seq_num(2 spoke12-p1), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
  Src address(1):
        10.1.100.0-10.1.100.255

  Dst address(1):
        0.0.0.0-255.255.255.255
```

The SD-WAN service rule prefers the primary shortcut tunnel (spoke11-p1_0) over other tunnels.

4. Make the primary shortcut be out of SLA. The traffic will switch to the backup parent tunnel and trigger the backup shortcut. Verify the diagnostics.
   a. Run a sniffer trace:

```
# diagnose sniffer packet any 'host 192.168.5.44' 4
interfaces=[any]
filters=[host 192.168.5.44]
20.588046 port2 in 10.1.100.22 -> 192.168.5.44: icmp: echo request
20.588157 spoke12-p1 out 10.1.100.22 -> 192.168.5.44: icmp: echo request
20.588791 spoke12-p1 in 192.168.5.44 -> 10.1.100.22: icmp: echo reply
20.588876 port2 out 192.168.5.44 -> 10.1.100.22: icmp: echo reply
21.589079 port2 in 10.1.100.22 -> 192.168.5.44: icmp: echo request
21.589190 spoke12-p1_0 out 10.1.100.22 -> 192.168.5.44: icmp: echo request
21.589661 spoke12-p1_0 in 192.168.5.44 -> 10.1.100.22: icmp: echo reply
21.589733 port2 out 192.168.5.44 -> 10.1.100.22: icmp: echo reply
```

When the primary shortcut tunnel goes out of SLA (spoke11-p1_0, alive, sla(0x0)), traffic reroutes to the backup parent tunnel (spoke12-p1) and then to the backup shortcut tunnel (spoke12-p1_0) once established.

   b. Verify the service status:

```
# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x2200 use-shortcut-sla sla-stickiness
 Tie break: cfg
  Gen(23), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Member sub interface(4):
    1: seq_num(1), interface(spoke11-p1):
       1: spoke11-p1_0(62)
    3: seq_num(2), interface(spoke12-p1):
       1: spoke12-p1_0(63)
  Members(4):
    1: Seq_num(1 spoke11-p1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    2: Seq_num(2 spoke12-p1_0), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
```

```
    3: Seq_num(2 spoke12-p1), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
    4: Seq_num(1 spoke11-p1_0), alive, sla(0x0), gid(0), cfg_order(0), local cost(0),
selected
  Src address(1):
        10.1.100.0-10.1.100.255

  Dst address(1):
        0.0.0.0-255.255.255.255
```

The backup shortcut tunnel (spoke12-p1_0) is now preferred.

5. After an idle timeout, the primary shortcut is torn down. The primary parent tunnel is now preferred, but traffic is still kept on the backup shortcut due to `sla-stickiness` being enabled. Verify the diagnostics.
   a. Verify the service status:

```
# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x2200 use-shortcut-sla sla-stickiness
 Tie break: cfg
  Gen(24), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Member sub interface(3):
    3: seq_num(2), interface(spoke12-p1):
       1: spoke12-p1_0(63)
  Members(3):
    1: Seq_num(1 spoke11-p1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    2: Seq_num(2 spoke12-p1_0), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
    3: Seq_num(2 spoke12-p1), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
  Src address(1):
        10.1.100.0-10.1.100.255

  Dst address(1):
        0.0.0.0-255.255.255.255
```

   b. Run a sniffer trace:

```
# diagnose sniffer packet any 'host 192.168.5.44' 4
interfaces=[any]
filters=[host 192.168.5.44]
1.065143 port2 in 10.1.100.22 -> 192.168.5.44: icmp: echo request
1.065218 spoke12-p1_0 out 10.1.100.22 -> 192.168.5.44: icmp: echo request
1.065471 spoke12-p1_0 in 192.168.5.44 -> 10.1.100.22: icmp: echo reply
1.065508 port2 out 192.168.5.44 -> 10.1.100.22: icmp: echo reply
2.066155 port2 in 10.1.100.22 -> 192.168.5.44: icmp: echo request
2.066198 spoke12-p1_0 out 10.1.100.22 -> 192.168.5.44: icmp: echo request
2.066442 spoke12-p1_0 in 192.168.5.44 -> 10.1.100.22: icmp: echo reply
2.066480 port2 out 192.168.5.44 -> 10.1.100.22: icmp: echo reply
3.067201 port2 in 10.1.100.22 -> 192.168.5.44: icmp: echo request
3.067255 spoke12-p1_0 out 10.1.100.22 -> 192.168.5.44: icmp: echo request
```

```
3.067507 spoke12-p1_0 in 192.168.5.44 -> 10.1.100.22: icmp: echo reply
3.067544 port2 out 192.168.5.44 -> 10.1.100.22: icmp: echo reply
```

6. Send new traffic from PC1 to PC2. The traffic is routed to the primary parent tunnel and triggers the primary shortcut, then traffic will switch to the primary shortcut if it is in SLA. Verify the connection.
   a. Run a sniffer trace:

```
# diagnose sniffer packet any 'host 192.168.5.4' 4
interfaces=[any]
filters=[host 192.168.5.4]
17.120310 port2 in 10.1.100.22 -> 192.168.5.4: icmp: echo request
17.120475 spoke11-p1 out 10.1.100.22 -> 192.168.5.4: icmp: echo request
17.121096 spoke11-p1 in 192.168.5.4 -> 10.1.100.22: icmp: echo reply
17.121151 port2 out 192.168.5.4 -> 10.1.100.22: icmp: echo reply
18.121331 port2 in 10.1.100.22 -> 192.168.5.4: icmp: echo request
18.121480 spoke11-p1_0 out 10.1.100.22 -> 192.168.5.4: icmp: echo request
18.121954 spoke11-p1_0 in 192.168.5.4 -> 10.1.100.22: icmp: echo reply
18.122007 port2 out 192.168.5.4 -> 10.1.100.22: icmp: echo reply
...
```

   At first, traffic tries to go to the primary parent tunnel so that it can trigger the primary shortcut to establish. The primary shortcut (spoke11-p1_0) is in SLA and new traffic flows through it.

```
...
14.194066 port2 in 10.1.100.22 -> 192.168.5.4: icmp: echo request
14.194247 spoke12-p1_0 out 10.1.100.22 -> 192.168.5.4: icmp: echo request
14.194499 spoke12-p1_0 in 192.168.5.4 -> 10.1.100.22: icmp: echo reply
14.194565 port2 out 192.168.5.4 -> 10.1.100.22: icmp: echo reply
15.195093 port2 in 10.1.100.22 -> 192.168.5.4: icmp: echo request
15.195174 spoke12-p1_0 out 10.1.100.22 -> 192.168.5.4: icmp: echo request
15.195326 spoke12-p1_0 in 192.168.5.4 -> 10.1.100.22: icmp: echo reply
15.195361 port2 out 192.168.5.4 -> 10.1.100.22: icmp: echo reply
```

   After the primary shortcut goes out of SLA, the traffic switches to the backup shortcut (spoke12-p1_0).
   b. Verify the service status:

```
# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x2200 use-shortcut-sla sla-stickiness
 Tie break: cfg
  Gen(36), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Member sub interface(4):
    1: seq_num(1), interface(spoke11-p1):
       1: spoke11-p1_0(67)
    3: seq_num(2), interface(spoke12-p1):
       1: spoke12-p1_0(66)
  Members(4):
    1: Seq_num(1 spoke11-p1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    2: Seq_num(2 spoke12-p1_0), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
```

```
    3: Seq_num(2 spoke12-p1), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected
    4: Seq_num(1 spoke11-p1_0), alive, sla(0x0), gid(0), cfg_order(0), local cost(0),
selected
  Src address(1):
        10.1.100.0-10.1.100.255

  Dst address(1):
        0.0.0.0-255.255.255.255
```

New traffic switches back to the backup shortcut while the primary shortcut is still out of SLA.

Use case 2 example

To configure shortcut stickiness for ADVPN shortcuts:

1. Configure SD-WAN on the Spoke_1 FortiGate:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
            set advpn-select enable
            set advpn-health-check "1"
        next
    end
    config members
        edit 1
            set interface "spoke11-p1"
            set transport-group 1
        next
        edit 2
            set interface "spoke12-p1"
            set transport-group 2
        next
    end
    config health-check
        edit "1"
            set server "9.0.0.1"
            set members 1 2
            config sla
                edit 1
                next
            end
        next
    end
    config service
        edit 1
            set name "1"
            set mode sla
            set dst "all"
            set src "10.1.100.0"
```

```
            set sla-stickiness enable
            config sla
                edit "1"
                    set id 1
                next
            end
            set priority-members 1 2
        next
    end
end
```

2. Verify the SD-WAN configuration.
   a. Verify the health check status:

```
# diagnose sys sdwan health-check
Health Check(1):
Seq(1 spoke11-p1): state(alive), packet-loss(0.000%), latency(0.186), jitter(0.040), mos
(4.404), custom_profile(0.000), bandwidth-up(999999), bandwidth-dw(999996), bandwidth-bi
(1999995), sla_map=0x1
Seq(2 spoke12-p1): state(alive), packet-loss(0.000%), latency(0.131), jitter(0.018), mos
(4.404), custom_profile(0.000), bandwidth-up(999999), bandwidth-dw(999996), bandwidth-bi
(1999995), sla_map=0x1
```

   b. Verify the service status:

```
# diagnose sys sdwan service4
Service(1): Address Mode(IPV4) flags=0xc200 use-shortcut-sla use-shortcut sla-stickiness
 Tie break: cfg
 Shortcut priority: 3
  Gen(1), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-compare-
order
  Members(2):
    1: Seq_num(1 spoke11-p1 virtual-wan-link), alive, sla(0x1), gid(0), cfg_order(0),
local cost(0), selected
    2: Seq_num(2 spoke12-p1 virtual-wan-link), alive, sla(0x1), gid(0), cfg_order(1),
local cost(0), selected
  Src address(1):
        10.1.100.0-10.1.100.255
  Dst address(1):
        0.0.0.0-255.255.255.255
```

   The SD-WAN service rule prefers the primary parent tunnel (spoke11-p1) over the backup parent tunnel (spoke12-p1) before shortcuts are established.

3. Send traffic from PC-1 to PC-2 to trigger the primary shortcut. Verify the diagnostics.
   a. Run a sniffer trace:

```
# diagnose sniffer packet any 'host 192.168.5.44' 4interfaces=[any]
filters=[host 192.168.5.44]
3.323508 port2 in 10.1.100.22 -> 192.168.5.44: icmp: echo request
3.323600 spoke11-p1 out 10.1.100.22 -> 192.168.5.44: icmp: echo request
3.324079 spoke11-p1 in 192.168.5.44 -> 10.1.100.22: icmp: echo reply
```

```
3.324246 port2 out 192.168.5.44 -> 10.1.100.22: icmp: echo reply
4.324515 port2 in 10.1.100.22 -> 192.168.5.44: icmp: echo request
4.324576 spoke11-p1_0 out 10.1.100.22 -> 192.168.5.44: icmp: echo request
4.325037 spoke11-p1_0 in 192.168.5.44 -> 10.1.100.22: icmp: echo reply
4.325066 port2 out 192.168.5.44 -> 10.1.100.22: icmp: echo reply
```

The SD-WAN service rule sends traffic to the parent tunnel (spoke11-p1) initially, and then switches to the primary shortcut tunnel (spoke11-p1_0) once it is established.

b. Verify the service status:

```
# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0xc200 use-shortcut-sla use-shortcut sla-stickiness
 Tie break: cfg
 Shortcut priority: 3
  Gen(27), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-compare-
order
  Member sub interface(3):
    2: seq_num(1), interface(spoke11-p1):
       1: spoke11-p1_0(82)
  Members(3):
    1: Seq_num(1 spoke11-p1_0 virtual-wan-link), alive, sla(0x1), gid(0), cfg_order(0),
local cost(0), selected
    2: Seq_num(1 spoke11-p1 virtual-wan-link), alive, sla(0x1), gid(0), cfg_order(0),
local cost(0), selected
    3: Seq_num(2 spoke12-p1 virtual-wan-link), alive, sla(0x1), gid(0), cfg_order(1),
local cost(0), selected
  Src address(1):
        10.1.100.0-10.1.100.255
  Dst address(1):
        0.0.0.0-255.255.255.255
```

The SD-WAN service rule prefers the primary shortcut tunnel (spoke11-p1_0) over other tunnels.

4. Make the primary shortcut be out of SLA. The traffic will switch to the backup parent tunnel and trigger the backup shortcut. Verify the diagnostics.

a. Run a sniffer trace:

```
# diagnose sniffer packet any 'host 192.168.5.44' 4
interfaces=[any]
filters=[host 192.168.5.44]
27.299268 port2 in 10.1.100.22 -> 192.168.5.44: icmp: echo request
27.299331 spoke12-p1 out 10.1.100.22 -> 192.168.5.44: icmp: echo request
27.299862 spoke12-p1 in 192.168.5.44 -> 10.1.100.22: icmp: echo reply
27.299889 port2 out 192.168.5.44 -> 10.1.100.22: icmp: echo reply
28.300301 port2 in 10.1.100.22 -> 192.168.5.44: icmp: echo request
28.300372 spoke12-p1_0 out 10.1.100.22 -> 192.168.5.44: icmp: echo request
28.300938 spoke12-p1_0 in 192.168.5.44 -> 10.1.100.22: icmp: echo reply
28.300970 port2 out 192.168.5.44 -> 10.1.100.22: icmp: echo reply
```

When the primary shortcut tunnel goes out of SLA (spoke11-p1_0, alive, sla(0x0)), traffic reroutes to the backup parent tunnel (spoke12-p1) and then to the backup shortcut tunnel (spoke12-p1_0) once established.

b. Verify the service status:

```
# diagnose sys sdwan service4
Service(1): Address Mode(IPV4) flags=0xc200 use-shortcut-sla use-shortcut sla-stickiness
 Tie break: cfg
 Shortcut priority: 3
  Gen(38), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-compare-
order
  Member sub interface(4):
    2: seq_num(1), interface(spoke11-p1):
       1: spoke11-p1_0(84)
    3: seq_num(2), interface(spoke12-p1):
       1: spoke12-p1_0(85)
  Members(4):
    1: Seq_num(2 spoke12-p1_0 virtual-wan-link), alive, sla(0x1), gid(0), cfg_order(1),
local cost(0), selected
    2: Seq_num(1 spoke11-p1 virtual-wan-link), alive, sla(0x1), gid(0), cfg_order(0),
local cost(0), selected
    3: Seq_num(2 spoke12-p1 virtual-wan-link), alive, sla(0x1), gid(0), cfg_order(1),
local cost(0), selected
    4: Seq_num(1 spoke11-p1_0 virtual-wan-link), alive, sla(0x0), gid(0), cfg_order(0),
local cost(0), selected
  Src address(1):
        10.1.100.0-10.1.100.255
  Dst address(1):
        0.0.0.0-255.255.255.255
```

The backup shortcut tunnel (spoke12-p1_0) is now preferred.

5. After an idle timeout, the primary shortcut is torn down. The primary parent tunnel is now preferred, but traffic is still kept on the backup shortcut due to `sla-stickiness` being enabled. Verify the diagnostics.

   a. Verify the service status:

```
# diagnose sys sdwan service4
Service(1): Address Mode(IPV4) flags=0xc200 use-shortcut-sla use-shortcut sla-stickiness
 Tie break: cfg
 Shortcut priority: 3
  Gen(42), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-compare-
order
  Member sub interface(4):
    3: seq_num(1), interface(spoke11-p1):
       1: spoke11-p1_0(84)
    4: seq_num(2), interface(spoke12-p1):
       1: spoke12-p1_0(85)
  Members(4):
    1: Seq_num(1 spoke11-p1_0 virtual-wan-link), alive, sla(0x1), gid(0), cfg_order(0),
local cost(0), selected
    2: Seq_num(2 spoke12-p1_0 virtual-wan-link), alive, sla(0x1), gid(0), cfg_order(1),
local cost(0), selected
    3: Seq_num(1 spoke11-p1 virtual-wan-link), alive, sla(0x1), gid(0), cfg_order(0),
local cost(0), selected
    4: Seq_num(2 spoke12-p1 virtual-wan-link), alive, sla(0x1), gid(0), cfg_order(1),
local cost(0), selected
```

```
  Src address(1):
        10.1.100.0-10.1.100.255
  Dst address(1):
        0.0.0.0-255.255.255.255
```

b. Run a sniffer trace:

```
# diagnose sniffer packet any 'host 192.168.5.44' 4interfaces=[any]
filters=[host 192.168.5.44]
0.636838 port2 in 10.1.100.22 -> 192.168.5.44: icmp: echo request
0.636866 spoke12-p1_0 out 10.1.100.22 -> 192.168.5.44: icmp: echo request
0.637067 spoke12-p1_0 in 192.168.5.44 -> 10.1.100.22: icmp: echo reply
0.637082 port2 out 192.168.5.44 -> 10.1.100.22: icmp: echo reply
1.637876 port2 in 10.1.100.22 -> 192.168.5.44: icmp: echo request
1.637900 spoke12-p1_0 out 10.1.100.22 -> 192.168.5.44: icmp: echo request
1.638090 spoke12-p1_0 in 192.168.5.44 -> 10.1.100.22: icmp: echo reply
1.638105 port2 out 192.168.5.44 -> 10.1.100.22: icmp: echo reply
```

6. Send new traffic from PC1 to PC2 and run a sniffer trace. The traffic is routed to the primary shortcut.

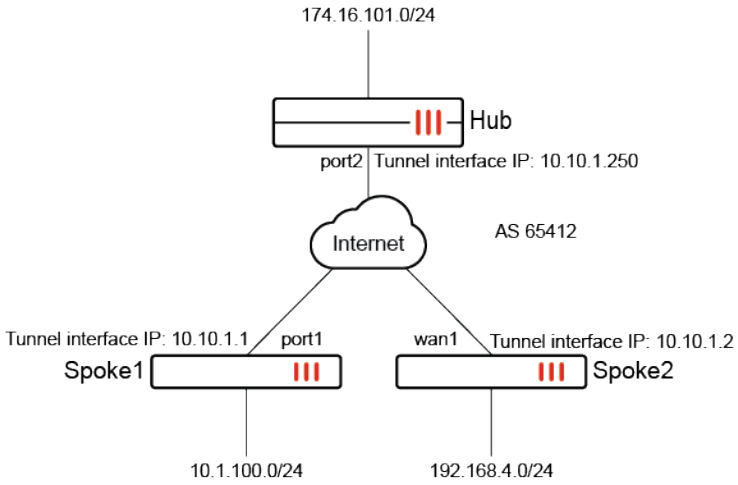
```
# diagnose sniffer packet any 'host 192.168.5.4' 4interfaces=[any]
filters=[host 192.168.5.44]
2.792200 port2 in 10.1.100.22 -> 192.168.5.44: icmp: echo request
2.792302 spoke11-p1_0 out 10.1.100.22 -> 192.168.5.44: icmp: echo request
2.792612 spoke11-p1_0 in 192.168.5.44 -> 10.1.100.22: icmp: echo reply
2.792651 port2 out 192.168.5.44 -> 10.1.100.22: icmp: echo reply
3.793217 port2 in 10.1.100.22 -> 192.168.5.44: icmp: echo request
3.793273 spoke11-p1_0 out 10.1.100.22 -> 192.168.5.44: icmp: echo request
3.793522 spoke11-p1_0 in 192.168.5.44 -> 10.1.100.22: icmp: echo reply
3.793560 port2 out 192.168.5.44 -> 10.1.100.22: icmp: echo reply
```

Using a single IKE selector in ADVPN to match all SD-WAN control plane traffic

In the SD-WAN with ADVPN use case, two spokes can communicate with each other on the control plane by an ADVPN shortcut. In order to separate the control traffic from data traffic, the IKE creates a dynamic selector for health check packets sent between the spokes. BGP traffic is also matched by this dynamic IKE selector. Therefore, when spokes establish BGP peering with other spokes, the BGP traffic does not count towards the data traffic and will not impact IPsec idle timeout and shortcut tunnel tear down.

Example

In this example, SD-WAN with ADVPN is configured. The IPsec ADVPN shortcut tunnel is required to tear down when it is idle. SD-WAN health checks are configured, and BGP neighbors established between the spokes is required.



To configure the Hub FortiGate:

1. Configure the phase 1 interface:

```
config vpn ipsec phase1-interface
    edit "Hub"
        set type dynamic
        set interface "port2"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set add-route disable
        set dpd on-idle
        set auto-discovery-sender enable
        set psksecret ************
        set dpd-retryinterval 60
    next
end
```

2. Configure the phase 2 interface:

```
config vpn ipsec phase2-interface
    edit "Hub"
        set phase1name "Hub"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
    next
end
```

3. Configure the VPN interface:

```
config system interface
    edit "Hub"
        set vdom "root"
```

```
        set ip 10.10.1.250 255.255.255.255
        set allowaccess ping
        set type tunnel
        set remote-ip 10.10.1.254 255.255.255.0
        set snmp-index 50
        set interface "port2"
    next
end
```

4. Configure the BGP settings:

```
config router bgp
    set as 65412
    config neighbor
        edit "10.10.1.1"
            set advertisement-interval 0
            set remote-as 65412
            set route-reflector-client enable
        next
        edit "10.10.1.2"
            set advertisement-interval 0
            set remote-as 65412
            set route-reflector-client enable
        next
    end
    config network
        edit 1
            set prefix 174.16.101.0 255.255.255.0
        next
    end
end
```

To configure the Spoke1 FortiGate:

1. Configure the phase 1 interface:

```
config vpn ipsec phase1-interface
    edit "Spoke1"
        set interface "port1"
        set ike-version 2
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set add-route disable
        set npu-offload disable
        set idle-timeout enable
        set idle-timeoutinterval 5
        set auto-discovery-receiver enable
        set remote-gw 172.16.200.4
        set psksecret ************
```

```
   next
end
```

2. Configure the phase 2 interface:

```
config vpn ipsec phase2-interface
    edit "Spoke1"
        set phase1name "Spoke1"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
    next
end
```

3. Configure the VPN interface:

```
config system interface
    edit "Spoke1"
        set vdom "root"
        set ip 10.10.1.1 255.255.255.255
        set allowaccess ping
        set type tunnel
        set remote-ip 10.10.1.254 255.255.255.0
        set snmp-index 28
        set interface "port1"
    next
end
```

4. Configure the BGP settings:

```
config router bgp
    set as 65412
    config neighbor
        edit "10.10.1.250"
            set advertisement-interval 0
            set remote-as 65412
        next
        edit "10.10.1.2"
            set remote-as 65412
        next
    end
    config network
        edit 1
            set prefix 10.1.100.0 255.255.255.0
        next
    end
end
```

5. Configure the SD-WAN settings:

```
config system sdwan
    set status enable
    config zone
```

```
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "Spoke1"
        next
    end
    config health-check
        edit "1"
            set server "174.16.101.44"
            set members 0
        next
    end
end
```

To configure the Spoke2 FortiGate:

1. Configure the phase 1 interface:

```
config vpn ipsec phase1-interface
    edit "Spoke2"
        set interface "wan1"
        set ike-version 2
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set add-route disable
        set npu-offload disable
        set idle-timeout enable
        set idle-timeoutinterval 5
        set auto-discovery-receiver enable
        set remote-gw 172.16.200.4
        set psksecret ************
    next
end
```

2. Configure the phase 2 interface:

```
config vpn ipsec phase2-interface
    edit "Spoke2"
        set phase1name "Spoke2"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
    next
end
```

3. Configure the VPN interface:

```
config system interface
    edit "Spoke2"
```

```
        set vdom "root"
        set ip 10.10.1.2 255.255.255.255
        set allowaccess ping
        set type tunnel
        set remote-ip 10.10.1.254 255.255.255.0
        set snmp-index 15
        set interface "wan1"
    next
end
```

4. Configure the BGP settings:

```
config router bgp
    set as 65412
    config neighbor
        edit "10.10.1.250"
            set advertisement-interval 0
            set remote-as 65412
        next
        edit "10.10.1.1"
            set remote-as 65412
        next
    end
    config network
        edit 1
            set prefix 192.168.4.0 255.255.255.0
        next
    end
end
```

5. Configure the SD-WAN settings:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "Spoke2"
        next
    end
    config health-check
        edit "1"
            set server "174.16.101.44"
            set members 0
        next
    end
end
```

To verify the configuration:

1. Send traffic between the spokes to establish the ADVPN shortcut.
2. Verify the IPsec tunnel state on the Spoke1 FortiGate:

```
Spoke1 # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=Spoke1_0 ver=2 serial=7 172.16.200.1:0->172.16.200.3:0 tun_id=10.10.1.2 tun_
id6=::10.0.0.3 dst_mtu=1500 dpd-link=on weight=1
bound_if=19 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/66224 options[102b0]=create_dev
rgwy-chg frag-rfc  role=primary accept_traffic=1 overlay_id=0

parent=Spoke1 index=0
proxyid_num=2 child_num=0 refcnt=6 ilast=0 olast=0 ad=r/2
stat: rxp=0 txp=1 rxb=0 txb=40
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=1
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=Spoke1 proto=0 sa=1 ref=5 serial=2 adr health-check
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.10.1.2-10.10.1.2:0
  SA:  ref=3 options=92626 type=00 soft=0 mtu=1438 expire=43055/0B replaywin=2048
       seqno=214 esn=0 replaywin_lastseq=00000213 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43189/43200
  dec: spi=17a473be esp=aes key=16 40dfada9532cefe5563de71ac5908aa1
       ah=sha1 key=20 36e967d9b6fce8807132c3923d0edfae6cb6c115
  enc: spi=75cde30a esp=aes key=16 9bf08196d6830455a75bc676e04c816f
       ah=sha1 key=20 638db13dc4db0a6e5f523047805d18413eea4d4d
  dec:pkts/bytes=1060/42958, enc:pkts/bytes=1062/77075
  npu_flag=00 npu_rgwy=172.16.200.3 npu_lgwy=172.16.200.1 npu_selid=c dec_npuid=0 enc_npuid=0
proxyid=Spoke1 proto=0 sa=1 ref=2 serial=1 adr
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=12226 type=00 soft=0 mtu=1438 expire=43055/0B replaywin=2048
       seqno=2 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43189/43200
  dec: spi=17a473bd esp=aes key=16 c78e5085857d0c5842e394fc44b38822
       ah=sha1 key=20 0bb885a85f77aa491a1209e4d36b7cddd7caf152
  enc: spi=75cde309 esp=aes key=16 6717935721e4a25428d6a7a633da75a9
       ah=sha1 key=20 eaf092280cf5b9f9db09ac95258786ffbfacead0
  dec:pkts/bytes=0/0, enc:pkts/bytes=2/144
  npu_flag=00 npu_rgwy=172.16.200.3 npu_lgwy=172.16.200.1 npu_selid=b dec_npuid=0 enc_npuid=0
------------------------------------------------------
name=Spoke1 ver=2 serial=1 172.16.200.1:0->172.16.200.4:0 tun_id=172.16.200.4 tun_
id6=::172.16.200.4 dst_mtu=1500 dpd-link=on weight=1
bound_if=19 lgwy=static/1 tun=intf mode=auto/1 encap=none/560 options[0230]=create_dev frag-
rfc  role=primary accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=1 refcnt=5 ilast=0 olast=0 ad=r/2
stat: rxp=542 txp=553 rxb=22117 txb=22748
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=0
```

```
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=Spoke1 proto=0 sa=1 ref=4 serial=1 adr
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=12226 type=00 soft=0 mtu=1438 expire=42636/0B replaywin=2048
       seqno=22a esn=0 replaywin_lastseq=0000021f qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42900/43200
  dec: spi=17a473bc esp=aes key=16 eff2dc03b48968bb55b9e3950ebde431
       ah=sha1 key=20 5db42a32aec15bc8a5fe392c256d1ae8ab3b4ef8
  enc: spi=bdc3bd80 esp=aes key=16 d0ec06b61ad572cc8813b599edde8c68
       ah=sha1 key=20 0306850f0184d957e9475da33d7971653a95c233
  dec:pkts/bytes=1084/44234, enc:pkts/bytes=1106/80932
  npu_flag=00 npu_rgwy=172.16.200.4 npu_lgwy=172.16.200.1 npu_selid=0 dec_npuid=0 enc_npuid=0
```

The dynamic selector is created (highlighted) for SD-WAN control traffic, SD-WAN health checks, and BGP between spokes traffic.

3. Verify the BGP neighbors and check the routing table:

```
Spoke1 # get router info bgp summary

VRF 0 BGP router identifier 172.16.200.1, local AS number 65412
BGP table version is 8
1 BGP AS-PATH entries
0 BGP community entries

Neighbor    V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.10.1.2   4      65412      52      76        7    0    0 00:06:27        1
10.10.1.250 4      65412      70      69        1    0    0 00:58:44        2

Total number of neighbors 2
```

4. Stop sending traffic between the spokes, and wait for a few minutes (idle timeout).
5. Verify the IPsec tunnel state on the Spoke1 FortiGate:

```
Spoke1 # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=Spoke1 ver=2 serial=1 172.16.200.1:0->172.16.200.4:0 tun_id=172.16.200.4 tun_
id6=::172.16.200.4 dst_mtu=1500 dpd-link=on weight=1
bound_if=19 lgwy=static/1 tun=intf mode=auto/1 encap=none/560 options[0230]=create_dev frag-
rfc  role=primary accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=4 ilast=0 olast=0 ad=r/2
stat: rxp=1467 txp=1469 rxb=60190 txb=60214
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=Spoke1 proto=0 sa=1 ref=3 serial=1 adr
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=12226 type=00 soft=0 mtu=1438 expire=42199/0B replaywin=2048
```

```
      seqno=5be esn=0 replaywin_lastseq=000005bc qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42903/43200
  dec: spi=76fdf7d1 esp=aes key=16 b26fd2dae76665f580d255b67f79df1e
       ah=sha1 key=20 14b0acc3c8c92a0af8ab43ff0437d2141b6d3f65
  enc: spi=bdc3bd85 esp=aes key=16 3eae3ad42aa32d7cdd972dfca286acd1
       ah=sha1 key=20 3655f67ee135f38e3f0790f1c7e3bd19c4a9285c
  dec:pkts/bytes=2934/120380, enc:pkts/bytes=2938/214606
  npu_flag=00 npu_rgwy=172.16.200.4 npu_lgwy=172.16.200.1 npu_selid=0 dec_npuid=0 enc_npuid=0
```

The shortcut tunnel between the spokes has been torn down. When data traffic is idle, the BGP traffic does not get sent on the data traffic selector, so the tunnel is not kept alive. This behavior is the expected, which consequently allows the shortcut tunnel to be torn down when idle.

6. Verify the IKE debugs messages to confirm the ADVPN shortcut was torn down:

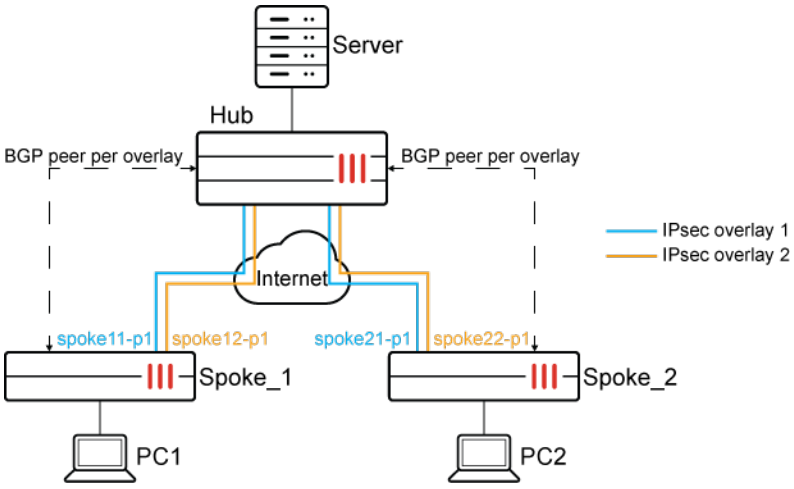
```
Spoke1 # diagnose debug enable
Spoke1 # diagnose debug application ike -1
...
ike 0:Spoke1_0: connection idle time-out
ike 0:Spoke1_0: deleting
ike 0:Spoke1_0: flushing
ike 0:Spoke1_0: deleting IPsec SA with SPI 75cde338
ike 0:Spoke1_0:Spoke1: deleted IPsec SA with SPI 75cde338, SA count: 0
ike 0:Spoke1_0: sending SNMP tunnel DOWN trap for Spoke1
ike 0:Spoke1_0: tunnel down event 0.0.0.0
ike 0:Spoke1_0:Spoke1: delete
ike 0:Spoke1_0: deleting IPsec SA with SPI 75cde337
ike 0:Spoke1_0:Spoke1: deleted IPsec SA with SPI 75cde337, SA count: 0
ike 0:Spoke1_0: sending SNMP tunnel DOWN trap for Spoke1
ike 0:Spoke1_0: tunnel down event 0.0.0.0
ike 0:Spoke1_0:Spoke1: delete
ike 0:Spoke1_0: flushed
ike 0:Spoke1_0:23:86: send informational
ike 0:Spoke1_0:23: sent IKE msg (INFORMATIONAL): 172.16.200.1:500->172.16.200.3:500, len=80,
vrf=0, id=0304e1284a432105/fa7d3fd75e7f481e:00000004
ike 0:Spoke1_0: delete connected route 10.10.1.1 -> 10.10.1.2
ike 0:Spoke1_0: delete dynamic
ike 0:Spoke1_0: deleted
ike 0:Spoke1: schedule auto-negotiate
ike 0: comes 172.16.200.3:500->172.16.200.1:500,ifindex=19,vrf=0....
ike 0: IKEv2 exchange=INFORMATIONAL_RESPONSE id=0304e1284a432105/fa7d3fd75e7f481e:00000004
len=80
```

SD-WAN in large scale deployments

Phase 2 selectors can be used to inject IKE routes on the ADVPN shortcut tunnel. When configuration method (`mode-cfg`) is enabled in IPsec phase 1 configuration, enabling `mode-cfg-allow-client-selector` allows custom phase 2 selectors to be configured. By also enabling the addition of a route to the peer destination selector (`add-route`) in the phase 1 configuration, IKE routes based on the phase 2 selectors can be injected. This means that routes do not need to be reflected on the hub to propagate them between spokes, avoiding possible BGP daemon process load issues and improving network scalability in a large-scale ADVPN network.

Route map rules can apply priorities to BGP routes. On the hub, priorities can be set in a route map's rules, and the route map can be applied on BGP routes. This allows the hub to mark the preferred path learned from the spokes with a priority value (lower priority is preferred), instead of using multiple SD-WAN policy routes on the hub. When a preferred outbound route map (`route-map-out-preferable`) is also configured in an SD-WAN neighbor on the spoke, deploying SD-WAN rules on the hub to steer traffic from the hub to a spoke is unnecessary.

SD-WAN members' local cost can be exchanged on the ADVPN shortcut tunnel so that spokes can use the remote cost as tiebreak to select a preferred shortcut. If multiple shortcuts originate from the same member to different members on the same remote spoke, then the remote cost on the shortcuts is used as the tiebreak to decide which shortcut is preferred.



In this example, SD-WAN is configured on an ADVPN network with a BGP neighbor per overlay.

Instead of reflecting BGP routes with the route-reflector on the hub, when the shortcuts are triggered, IKE routes on the shortcuts are directly injected based on the configured phase 2 selectors to allow routes to be exchanged between spokes.

Routes between the hub and the spokes are exchanged by BGP, and the spokes use the default route to send spoke-to-spoke traffic to the hub and trigger the shortcuts.

Instead of configuring SD-WAN rules on the hub, different priorities are configured on the BGP routes by matching different BGP communities to steer traffic from the hub to the spokes.

To configure Spoke 1:

1. Configure phase 1:

```
config vpn ipsec phase1-interface
    edit "spoke11-p1"
        ...
        set ike-version 2
        set net-device enable
        set add-route enable
        set mode-cfg enable
        set auto-discovery-receiver enable
        set mode-cfg-allow-client-selector enable
        set link-cost 11
```

```
        ...
    next
    edit "spoke12-p1"
       ...
        set ike-version 2
        set net-device enable
        set add-route enable
        set mode-cfg enable
        set auto-discovery-receiver enable
        set mode-cfg-allow-client-selector enable
        set link-cost 21
    next
end
```

2. Configure phase 2:

```
config vpn ipsec phase2-interface
    edit "spoke11-p2"
        ...
        set src-name "LAN_Net"
        set dst-name "all"
    next
    edit "spoke12-p2"
        ...
        set src-name "LAN_Net"
        set dst-name "all"
    next
end
```

3. Configure an address group:
   Spoke 1 uses LAN subnet 10.1-3.100.0/24.

```
config firewall addrgrp
    edit "LAN_Net"
        set member "10.1.100.0" "10.2.100.0" "10.3.100.0"
    next
end
```

4. Configure route maps:
   - If overlay 1 to the hub is in SLA, attach "65000:1" to the BGP routes advertised to the hub over overlay 1.
   - If overlay 2 to the hub is in SLA, attach "65000:2" to the BGP routes advertised to the hub over overlay 2.
   - If any overlay to the hub is out of SLA, attach "65000:9999" to the BGP routes advertised to the hub over any overlay.

```
config router route-map
    edit "HUB_CARRIER1"
        config rule
            edit 1
                set set-community "65000:1"
```

```
                ...
            next
        end
        ...
    next
    edit "HUB_CARRIER2"
        config rule
            edit 1
                set set-community "65000:2"
                ...
            next
        end
        ...
    next
    edit "HUB_BAD"
        config rule
            edit 1
                set set-community "65000:9999"
                ...
            next
        end
        ...
    next
end
```

5. Configure BGP and SD-WAN members and neighbors:

```
config router bgp
    set as 65412
    config neighbor
        edit "10.10.15.253"
            set remote-as 65412
            set route-map-out "HUB_BAD"
            set route-map-out-preferable "HUB_CARRIER1"
            ...
        next
        edit "10.10.16.253"
            set remote-as 65412
            set route-map-out "HUB_BAD"
            set route-map-out-preferable "HUB_CARRIER2"
            ...
        next
    end
end
```

```
config system sdwan
    config members
        edit 1
            set interface "spoke11-p1"
            set cost 10
        next
        edit 2
```

```
            set interface "spoke12-p1"
            set cost 20
        next
    end
    config neighbor
        edit "10.10.15.253"
            set member 1
            set health-check "1"
            set sla-id 1
        next
        edit "10.10.16.253"
            set member 2
            set health-check "11"
            set sla-id 1
        next
    end
end
```

To configure Spoke 2:

1. Configure phase 1:

```
config vpn ipsec phase1-interface
    edit "spoke21-p1"
        ...
        set ike-version 2
        set net-device enable
        set add-route enable
        set mode-cfg enable
        set auto-discovery-receiver enable
        set mode-cfg-allow-client-selector enable
        set link-cost 101
        ...
    next
    edit "spoke22-p1"
       ...
        set ike-version 2
        set net-device enable
        set add-route enable
        set mode-cfg enable
        set auto-discovery-receiver enable
        set mode-cfg-allow-client-selector enable
        set link-cost 201
    next
end
```

2. Configure phase 2:

```
config vpn ipsec phase2-interface
    edit "spoke21-p2"
        ...
        set src-name "LAN_Net"
```

```
        set dst-name "all"
    next
    edit "spoke22-p2"
        ...
        set src-name "LAN_Net"
        set dst-name "all"
    next
end
```

3. Configure an address group:
   Spoke 2 uses LAN subnet 192.168.5-7.0/24.

```
config firewall addrgrp
    edit "LAN_Net"
        set member "192.168.5.0" "192.168.6.0" "192.168.7.0"
    next
end
```

4. Configure route maps:
   - If overlay 1 to the hub is in SLA, attach "65000:1" to the BGP routes advertised to the hub over overlay 1.
   - If overlay 2 to the hub is in SLA, attach "65000:2" to the BGP routes advertised to the hub over overlay 2.
   - If any overlay to the hub is out of SLA, attach "65000:9999" to the BGP routes advertised to the hub over any overlay.

```
config router route-map
    edit "HUB_CARRIER1"
        config rule
            edit 1
                set set-community "65000:1"
                ...
            next
        end
        ...
    next
    edit "HUB_CARRIER2"
        config rule
            edit 1
                set set-community "65000:2"
                ...
            next
        end
        ...
    next
    edit "HUB_BAD"
        config rule
            edit 1
                set set-community "65000:9999"
                ...
            next
```

```
        end
        ...
    next
end
```

5. Configure BGP and SD-WAN members and neighbors:

```
config router bgp
    set as 65412
    config neighbor
        edit "10.10.15.253"
            set remote-as 65412
            set route-map-out "HUB_BAD"
            set route-map-out-preferable "HUB_CARRIER1"
            ...
        next
        edit "10.10.16.253"
            set remote-as 65412
            set route-map-out "HUB_BAD"
            set route-map-out-preferable "HUB_CARRIER2"
            ...
        next
    end
end
```

```
config system sdwan
    config members
        edit 1
            set interface "spoke21-p1"
            set cost 10
        next
        edit 2
            set interface "spoke22-p1"
            set cost 20
        next
    end
    config neighbor
        edit "10.10.15.253"
            set member 1
            set health-check "1"
            set sla-id 1
        next
        edit "10.10.16.253"
            set member 2
            set health-check "11"
            set sla-id 1
        next
    end
end
```

To configure the hub:

1. Configure the route maps:
   - Set the priority to 100 for routes with community 65000:1, indicating that they are in SLA for overlay 1.
   - Set the priority to 200 for routes with community 65000:2, indicating that they are in SLA for overlay 2.
   - Set the priority to 9999 for routes with community 65000:9999, indicating that they are out of SLA for any overlay.

```
config router route-map
     edit "Set_Pri"
        config rule
            edit 1
                set match-community "comm_65000:1"
                set set-priority 100
            next
            edit 2
                set match-community "comm_65000:2"
                set set-priority 200
            next
            edit 3
                set match-community "comm_65000:9999"
                set set-priority 9999
            next
        end
    next
end
```

2. Configure BGP:

```
config router bgp
    set as 65412
    config neighbor-group
        edit "advpn"
            set remote-as 65412
            set route-map-in "Set_Pri"
            ...
        next
        edit "advpn2"
            set remote-as 65412
            set route-map-in "Set_Pri"
            ...
        next
    end
    config neighbor-range
        edit 1
            set prefix 10.10.15.0 255.255.255.0
            set neighbor-group "advpn"
        next
        edit 2
            set prefix 10.10.16.0 255.255.255.0
            set neighbor-group "advpn2"
        next
```

```
    end
end
```

To test the configuration:

1. Check the routing tables on the spokes:
   Spoke 1:

```
spoke-1 (root) # get router info routing-table all
B*      0.0.0.0/0 [200/0] via 10.10.15.253 (recursive is directly connected, spoke11-p1),
00:01:17, [1/0]         // default route to hub
                         [200/0] via 10.10.16.253 (recursive is directly connected, spoke12-
p1), 00:01:17, [1/0]
B       9.0.0.0/24 [200/0] via 10.10.15.253 (recursive is directly connected, spoke11-p1),
00:01:17, [1/0]        // route to the server behind hub
                          [200/0] via 10.10.16.253 (recursive is directly connected, spoke12-
p1), 00:01:17, [1/0]
C       10.1.100.0/24 is directly connected, port2             // route to PC 1
C       10.10.15.0/24 is directly connected, spoke11-p1        // overlay 1
C       10.10.15.1/32 is directly connected, spoke11-p1
C       10.10.16.0/24 is directly connected, spoke12-p1        // overlay 2
C       10.10.16.1/32 is directly connected, spoke12-p1
```

   Spoke 2:

```
spoke-2 (root) # get router info routing-table all
B*      0.0.0.0/0 [200/0] via 10.10.15.253 (recursive is directly connected, spoke21-p1),
00:46:14, [1/0]         // default route to hub
                         [200/0] via 10.10.16.253 (recursive is directly connected, spoke22-
p1), 00:46:14, [1/0]
B       9.0.0.0/24 [200/0] via 10.10.15.253 (recursive is directly connected, spoke21-p1),
00:46:18, [1/0]        // route to the server behind hub
                          [200/0] via 10.10.16.253 (recursive is directly connected, spoke22-
p1), 00:46:18, [1/0]
C       10.10.15.0/24 is directly connected, spoke21-p1        // overlay 1
C       10.10.15.2/32 is directly connected, spoke21-p1
C       10.10.16.0/24 is directly connected, spoke22-p1        // overlay 2
C       10.10.16.2/32 is directly connected, spoke22-p1
C       192.168.5.0/24 is directly connected, port2            // route to PC 2
```

2. Send traffic from PC 1 to PC 2 and trigger the shortcut:
   The IKE routes on the shortcut are directly injected based on the phase 2 selectors, and spoke-to-spoke traffic then goes directly through the shortcut instead of going through the hub.
   Spoke 1:

```
spoke-1 (root) # get router info routing-table static
S       192.168.5.0/24 [15/0] via spoke11-p1_0 tunnel 172.16.200.4 vrf 0, [1/0]
S       192.168.6.0/24 [15/0] via spoke11-p1_0 tunnel 172.16.200.4 vrf 0, [1/0]
S       192.168.7.0/24 [15/0] via spoke11-p1_0 tunnel 172.16.200.4 vrf 0, [1/0]
```

```
spoke-1 (root) # diagnose sniffer packet any 'host 192.168.5.44' 4
interfaces=[any]
filters=[host 192.168.5.44]
1.446306 port2 in 10.1.100.22 -> 192.168.5.44: icmp: echo request
1.446327 spoke11-p1_0 out 10.1.100.22 -> 192.168.5.44: icmp: echo request
1.446521 spoke11-p1_0 in 192.168.5.44 -> 10.1.100.22: icmp: echo reply
1.446536 port2 out 192.168.5.44 -> 10.1.100.22: icmp: echo reply
```

Spoke 2:

```
spoke-2 (root) # get router info routing-table static
S       10.1.100.0/24 [15/0] via spoke21-p1_0 tunnel 10.10.15.1 vrf 0, [1/0]
S       10.2.100.0/24 [15/0] via spoke21-p1_0 tunnel 10.10.15.1 vrf 0, [1/0]
S       10.3.100.0/24 [15/0] via spoke21-p1_0 tunnel 10.10.15.1 vrf 0, [1/0]
```

3. Confirm that the overlays are in SLA on the spokes:
   Spoke 1:

```
spoke-1 (root) # diagnose sys sdwan  neighbor
Neighbor(10.10.15.253): member(1)role(standalone)
        Health-check(1:1)  sla-pass selected alive
Neighbor(10.10.16.253): member(2)role(standalone)
        Health-check(11:1)  sla-pass selected alive
```

   Spoke 2:

```
spoke-2 (root) # diagnose sys sdwan neighbor
Neighbor(10.10.15.253): member(1)role(standalone)
        Health-check(1:1)  sla-pass selected alive
Neighbor(10.10.16.253): member(2)role(standalone)
        Health-check(11:1)  sla-pass selected alive
```

4. On the hub, check that the routes received from the spokes have the expected priorities:

```
hub (root) # diagnose ip route list | grep proto=11
tab=254 vf=0 scope=0 type=1 proto=11 prio=100 0.0.0.0/0.0.0.0/0->10.1.100.0/24 pref=0.0.0.0
gwy=10.10.15.1 dev=101(hub-phase1)
tab=254 vf=0 scope=0 type=1 proto=11 prio=200 0.0.0.0/0.0.0.0/0->10.1.100.0/24 pref=0.0.0.0
gwy=10.10.16.1 dev=102(hub2-phase1)
tab=254 vf=0 scope=0 type=1 proto=11 prio=100 0.0.0.0/0.0.0.0/0->192.168.5.0/24 pref=0.0.0.0
gwy=10.10.15.2 dev=101(hub-phase1)
tab=254 vf=0 scope=0 type=1 proto=11 prio=200 0.0.0.0/0.0.0.0/0->192.168.5.0/24 pref=0.0.0.0
gwy=10.10.16.2 dev=102(hub2-phase1)
```

   The priority set by the hub's route map is based on the community string received from the spoke. The route with a lower priority value is selected, so traffic to Spoke 1 goes out on the hub-phase1 tunnel:

```
hub (root) # diagnose sniffer packet any 'host 9.0.0.2' 4
interfaces=[any]
filters=[host 9.0.0.2]
2.735456 R190 in 9.0.0.2 -> 10.1.100.22: icmp: echo request
2.735508 hub-phase1 out 9.0.0.2 -> 10.1.100.22: icmp: echo request
```

```
2.735813 hub-phase1 in 10.1.100.22 -> 9.0.0.2: icmp: echo reply
2.735854 R190 out 10.1.100.22 -> 9.0.0.2: icmp: echo reply
```

5. If overlay 1 goes out of SLA, the priorities of the routes on the hub are updated and traffic from the hub to Spoke 1 goes through overlay 2:

   Spoke 1:

```
spoke-1 (root) # diagnose sys sdwan  neighbor
Neighbor(10.10.15.253): member(1)role(standalone)
        Health-check(1:1)  sla-fail alive
Neighbor(10.10.16.253): member(2)role(standalone)
        Health-check(11:1)  sla-pass selected alive
```

   Spoke 2:

```
spoke-2 (root) # diagnose sys sdwan neighbor
Neighbor(10.10.15.253): member(1)role(standalone)
        Health-check(1:1)  sla-fail alive
Neighbor(10.10.16.253): member(2)role(standalone)
        Health-check(11:1)  sla-pass selected alive
```

   Hub:

```
hub (root) # diagnose ip route list | grep proto=11
tab=254 vf=0 scope=0 type=1 proto=11 prio=200 0.0.0.0/0.0.0.0/0->10.1.100.0/24 pref=0.0.0.0
gwy=10.10.16.1 dev=102(hub2-phase1)
tab=254 vf=0 scope=0 type=1 proto=11 prio=9999 0.0.0.0/0.0.0.0/0->10.1.100.0/24 pref=0.0.0.0
gwy=10.10.15.1 dev=101(hub-phase1)
tab=254 vf=0 scope=0 type=1 proto=11 prio=200 0.0.0.0/0.0.0.0/0->192.168.5.0/24 pref=0.0.0.0
gwy=10.10.16.2 dev=102(hub2-phase1)
tab=254 vf=0 scope=0 type=1 proto=11 prio=9999 0.0.0.0/0.0.0.0/0->192.168.5.0/24 pref=0.0.0.0
gwy=10.10.15.2 dev=101(hub-phase1)
```

```
hub (root) # diagnose sniffer packet any 'host 9.0.0.2' 4
interfaces=[any]
filters=[host 9.0.0.2]
3.550181 R190 in 9.0.0.2 -> 10.1.100.22: icmp: echo request
3.550234 hub2-phase1 out 9.0.0.2 -> 10.1.100.22: icmp: echo request
3.550713 hub2-phase1 in 10.1.100.22 -> 9.0.0.2: icmp: echo reply
3.550735 R190 out 10.1.100.22 -> 9.0.0.2: icmp: echo reply
```

6. Verify the service diagnostics on Spoke 1:

```
# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
 Tie break: cfg
  Gen(4), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Members(2):
    1: Seq_num(1 spoke11-p1), alive, sla(0x1), gid(0), cfg_order(0), local cost(10), selected
    2: Seq_num(2 spoke12-p1), alive, sla(0x1), gid(0), cfg_order(1), local cost(20), selected
  Src address(1):
```

```
        10.1.100.0-10.1.100.255

  Dst address(1):
        0.0.0.0-255.255.255.255
```

7. Verify the service diagnostics on Spoke 2:

```
# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
 Tie break: cfg
  Gen(2), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Members(2):
    1: Seq_num(1 spoke21-p1), alive, sla(0x1), gid(0), cfg_order(0), local cost(10), selected
    2: Seq_num(2 spoke22-p1), alive, sla(0x1), gid(0), cfg_order(1), local cost(20), selected
  Src address(1):
        192.168.5.0-192.168.5.255

  Dst address(1):
        0.0.0.0-255.255.255.255
```

8. Trigger shortcuts between Spoke 1 and Spoke 2:
   - Shortcuts spoke11-p1_1 and spoke11-p1_0 originate from spoke11-p1.
   - spoke11-p1_1 corresponds to spoke21-p1_0 on Spoke 2.
   - spoke11-p1_0 corresponds to spoke22-p1_0 on Spoke 2.

   Spoke 1:

```
# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
 Tie break: cfg
  Gen(11), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Member sub interface(4):
    3: seq_num(1), interface(spoke11-p1):
       1: spoke11-p1_0(80)
       2: spoke11-p1_1(81)
  Members(4):
    1: Seq_num(1 spoke11-p1_1), alive, sla(0x1), gid(0), remote cost(101), cfg_order(0), local
cost(10), selected
    2: Seq_num(1 spoke11-p1_0), alive, sla(0x1), gid(0), remote cost(201), cfg_order(0), local
cost(10), selected
    3: Seq_num(1 spoke11-p1), alive, sla(0x1), gid(0), cfg_order(0), local cost(10), selected
    4: Seq_num(2 spoke12-p1), alive, sla(0x1), gid(0), cfg_order(1), local cost(20), selected
  Src address(1):
        10.1.100.0-10.1.100.255

  Dst address(1):
        0.0.0.0-255.255.255.255
```

   Spoke 2:

```
# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
 Tie break: cfg
  Gen(15), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Member sub interface(4):
    2: seq_num(1), interface(spoke21-p1):
       1: spoke21-p1_0(75)
    4: seq_num(2), interface(spoke22-p1):
       1: spoke22-p1_0(74)
  Members(4):
    1: Seq_num(1 spoke21-p1_0), alive, sla(0x1), gid(0), remote cost(11), cfg_order(0), local
cost(10), selected
    2: Seq_num(1 spoke21-p1), alive, sla(0x1), gid(0), cfg_order(0), local cost(10), selected
    3: Seq_num(2 spoke22-p1_0), alive, sla(0x1), gid(0), remote cost(11), cfg_order(1), local
cost(20), selected
    4: Seq_num(2 spoke22-p1), alive, sla(0x1), gid(0), cfg_order(1), local cost(20), selected
  Src address(1):
        192.168.5.0-192.168.5.255

  Dst address(1):
        0.0.0.0-255.255.255.255
```

The spoke11-p1_1 shortcut on Spoke 1 is preferred over spoke11-p1_0 due to the lower remote link cost of 101 when they have the same local SD-WAN member cost of 10.

9. Verify the policy route list on Spoke 1:

```
# diagnose firewall proute list
list route policy info(vf=root):

id=2131755009(0x7f100001) vwl_service=1(1) vwl_mbr_seq=1 1 1 2 dscp_tag=0xfc 0xfc flags=0x0
tos=0x00 tos_mask=0x00 protocol=0 sport=0-65535 iif=0(any) dport=1-65535 path(4) oif=81
(spoke11-p1_1) oif=80(spoke11-p1_0) oif=54(spoke11-p1) oif=55(spoke12-p1)
source(1): 10.1.100.0-10.1.100.255
destination(1): 0.0.0.0-255.255.255.255
hit_count=176 last_used=2022-07-12 11:56:08
```

ADVPN 2.0

The SD-WAN with ADVPN solution has evolved to version 2.0 with major changes to ADVPN design and operation, including the introduction of edge discovery and path management for ADVPN spokes.

ADVPN 2.0 incorporates intelligence into the spokes to ensure shortcut tunnels (also known as shortcuts) are established using underlays available on both spokes and chosen based on matching certain link health criteria. In unique circumstances where spokes use different transports than the hub, ADVPN 2.0 can use overlay placeholders to trigger shortcuts between spokes over transports to which hubs are not connected.

ADVPN 2.0 provides a more flexible SD-WAN solution than the original ADVPN to achieve resiliency against underlay outages or degraded underlay performance because it no longer depends on specific BGP routing designs or mechanisms, including route reflection, BGP next hop recursive resolution, BGP per overlay, and BGP on loopback.

Currently, ADVPN 2.0 only supports IPv4.

The following topics describe further details about ADVPN 2.0:

- SD-WAN with ADVPN 2.0 versus previous ADVPN on page 1343
- SD-WAN CLI configuration on page 1343
- Example SD-WAN configurations using ADVPN 2.0 on page 1345

SD-WAN with ADVPN 2.0 versus previous ADVPN

With the previous version of ADVPN and SD-WAN, shortcut path selection relied entirely on the overlays between the spokes. The hub and overlays were used to exchange IKE shortcut messages, and policy routes were configured on the hub to ensure shortcuts were established on the same overlay. In addition, user traffic was needed to trigger the process of establishing shortcuts.

With the latest version of ADVPN and SD-WAN, shortcut path selection is achieved through edge discovery and path management functionality on the ADVPN spokes.

1. Edge discovery:
   - Expand IKE Shortcut-Reply message to allow the local spoke (spoke where user traffic is initiated) to obtain the remote spoke (destination spoke for user traffic) WAN link information, which includes IP address, transport group, link quality, link cost, and member configuration order.
   - After shortcut establishment, WAN link information can be exchanged on the shortcut regularly every 5 seconds through UDP traffic. The path management function on the local spoke is regularly updated to pick up changes to remote or local overlays and select the best shortcut path accordingly.
2. Path management:

   The local spoke handles the remote spoke WAN link information, calculates the best shortcut path per SD-WAN service or rule, and then advises IKE to establish a shortcut using the selected path.
   - The local spoke directly sends a shortcut-query to a remote spoke to trigger a shortcut after path management chooses a path.
   - Path management can trigger multiple shortcuts for load balancing SD-WAN rules. All created shortcuts use unique underlay paths. Traffic can be load balanced over these multiple shortcuts to use as much of the available WAN bandwidth as possible without wasting idle links if they are healthy. The algorithm to calculate multiple shortcuts for the load balancing service will consider transport group and in-SLA status for local overlays.
   - Spokes can automatically deactivate all shortcuts connecting to the same spoke when user traffic is not observed for a specified time interval. This is achieved by enabling a shared idle timeout setting in the IPsec VPN Phase 1 interface settings for associated overlays.
   - When overlay placeholders are enabled, spokes can use overlay placeholders to trigger shortcuts between spokes over transports to which hubs are not connected.

Currently, ADVPN 2.0 only supports IPv4.

SD-WAN CLI configuration

The `config system sdwan` command is used to configure ADVPN 2.0 on the spokes:

```
config system sdwan
    config zone
        edit <zone-name>
            set advpn-select {enable | disable}
            set advpn-health-check <health-check name>
        next
    end
    config members
        edit <integer>
            set transport-group <integer>
        next
    end
    config service
        edit <integer>
            set shortcut-priority {enable | disable | auto}
        next
    end
end
```

Command	Description
`advpn-select {enable | disable}`	Enable or disable SDWAN/ADVPN-2.0 (default=disabled).
`advpn-health-check <health-check name>`	Specify the health check for the spoke whose info will be sent to the peer spoke.
`transport-group <integer>`	Specify different group ID between (1 -255) to differentiate link-type, such as Internet, MPLS, LTE, Satellite.
`shortcut-priority {enable | disable | auto}`	Enable or disable making ADVPN shortcut a high priority over overlay parent interfaces, if SLA mode or link cost factor mode conditions are met: • `enable`: enable a high priority of ADVPN shortcut for this service. • `disable`: disable a high priority of ADVPN shortcut for this service. • `auto`: automatically enable a high priority of ADVPN shortcut for this service if ADVPN2.0 is enabled.
`diagnose sys sdwan advpn-session`	Diagnostic command run on local spoke to view remote spoke WAN link information and path manager shortcut path selection. The displayed name is the hostname or HA group name, if in HA.

As with the previous version of ADVPN, on the hub, you must enable ADVPN and configure firewall policies between spokes.

The `config vpn ipsec phase1-interface` command is used to configure the shared idle timeout for overlays used by ADVPN.

```
config vpn ipsec phase1-interface
   edit <phase1-interface name>
      set idle-timeout {enable | disable}
      set shared-idle-timeout {enable | disable}
      set idle-timeoutinterval <integer>
```

```
  next
end
```

Command	Description
`idle-timeout {enable | disable}`	Enable/disable IPsec tunnel idle timeout (default = `disable`). Must be set to `enable` when `shared-idle-timeout` is enabled.
`shared-idle-timeout {enable | disable}`	Enable/disable `shared-idle-timeout` on involved overlays (default = `disable`).
`idle-timeoutinterval <integer>`	IPsec tunnel idle timeout, in minutes (5 - 43200, default = 5).

The `config vpn ipsec phase1-interface` command is used to configure overlay placeholders, and shortcut tunnels require each spoke to be configured with these settings:

```
config vpn ipsec phase1-interface
    edit <placeholder_phase1_interface_name>
        set type dynamic
        ...
        set net-device enable
        ...
        set auto-discovery-dialup-placeholder {enable | disable}
    next
end
```

Command	Description
`auto-discovery-dialup-placeholder {enable | disable}`	Enable/disable overlay placeholder tunnels (default = disable).

Currently, ADVPN 2.0 only supports IPv4.

Example SD-WAN configurations using ADVPN 2.0

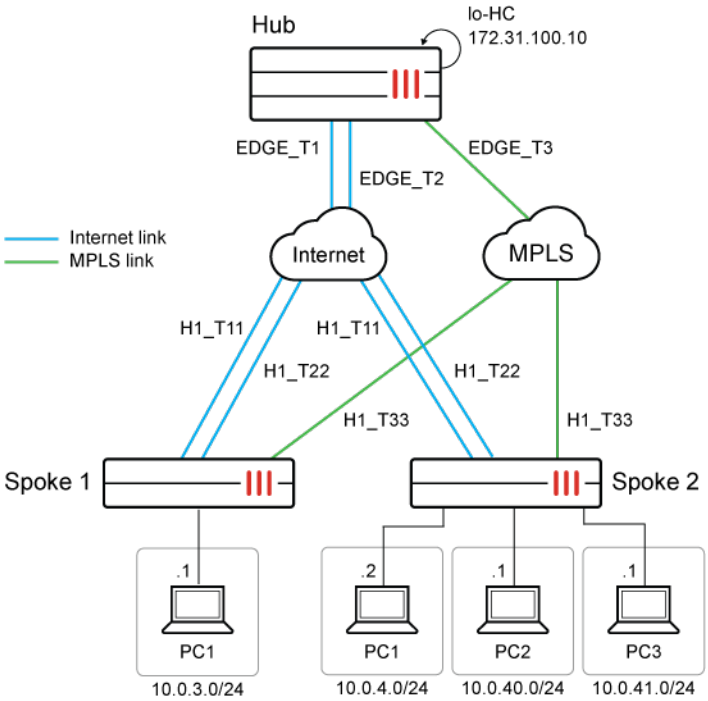
The configuration example illustrates the edge discovery and path management processes for a typical hub and spoke topology. This example focuses on SD-WAN configuration for steering traffic and establishing shortcuts in the direction from Spoke 1 to Spoke 2.

- Network Topology on page 1346
- SD-WAN configuration and health check status on page 1347
- Scenario 1: Traffic matching SD-WAN rule 1 on page 1350
- Scenario 2: Traffic matching SD-WAN rule 2 on page 1351
- Scenario 3: Traffic matching SD-WAN rule 3 on page 1353
- Scenario 4: Spoke 2 H1_T22 overlay link out-of-SLA on page 1355
- Scenario 5: Traffic matching SD-WAN rule configured for load balancing on page 1357

Currently, ADVPN 2.0 only supports IPv4.

Network Topology

In this example, BGP per overlay was used for dynamic routing to distribute the LAN routes behind each spoke to the other spoke. However, this was a design choice. You can also use BGP on loopback for this example.



Spokes 1 and 2 have the following VPN overlays between themselves and the hub:

VPN Overlays	IP address on Spoke 1	IP address on Spoke 2
H1_T11	172.31.80.1/32	172.31.80.2/32
H1_T22	172.31.81.1/32	172.31.81.2/32
H1_T33	172.31.82.1/32	172.31.82.2/32

SD-WAN Rules/Services defined on Spoke 1:

	SD-WAN Rule/Service 1	SD-WAN Rule/Service 2	SD-WAN Rule/Service 3
	H1_T11	H1_T22	H1_T33
	H1_T22	H1_T11	H1_T11
	H1_T33	H1_T33	H1_T22
Strategy for choosing outgoing interfaces	Lowest cost (SLA)	Lowest cost (SLA)	Best quality, link cost factor: packet loss

Throughout this example, transport group 1 is used for VPN overlays over Internet links while transport group 2 is used for the VPN overlay over an MPLS link.

In this example, user traffic is initiated behind Spoke 1 and destined to Spoke 2. Because of this, Spoke 1 is considered the local spoke, and Spoke 2 is considered the remote spoke.

SD-WAN configuration and health check status

This section includes:

- SD-WAN configuration and health check status on Spoke 1: on page 1347
- SD-WAN configuration and health check status on Spoke 2: on page 1349

SD-WAN configuration and health check status on Spoke 1:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
        edit "overlay"
            set advpn-select enable
            set advpn-health-check "HUB"
        next
    end
    config members
        edit 1
            set interface "H1_T11"
            set zone "overlay"
            set transport-group 1
        next
        edit 2
            set interface "H1_T22"
            set zone "overlay"
            set transport-group 1
        next
        edit 3
            set interface "H1_T33"
            set zone "overlay"
            set transport-group 2
        next
    end
    config health-check
        edit "HUB"
            set server "172.31.100.100"
            set members 1 2 3
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 100
                next
            end
```

```
        next
    end
    config service
        edit 1
            set name "1"
            set mode sla
            set shortcut-priority enable
            set dst "spoke-2_LAN-1" "Tunnel_IPs"
            set src "spoke-1_LAN-1" "Tunnel_IPs"
            config sla
                edit "HUB"
                    set id 1
                next
            end
            set priority-members 1 2 3
        next
        edit 2
            set name "2"
            set mode sla
            set shortcut-priority enable
            set dst "spoke-2_LAN-2" "Tunnel_IPs"
            set src "spoke-1_LAN-1" "Tunnel_IPs"
            config sla
                edit "HUB"
                    set id 1
                next
            end
            set priority-members 2 1 3
        next
        edit 3
            set name "3"
            set mode priority
            set dst "spoke-2_LAN-3" "Tunnel_IPs"
            set src "spoke-1_LAN-1" "Tunnel_IPs"
            set health-check "HUB"
            set link-cost-factor packet-loss
            set priority-members 3 1 2
        next
    end
end
```

```
# diagnose sys sdwan health-check
Health Check(HUB):
Seq(1 H1_T11): state(alive), packet-loss(0.000%) latency(0.231), jitter(0.029), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x1
Seq(2 H1_T22): state(alive), packet-loss(0.000%) latency(0.193), jitter(0.010), mos(4.404),
bandwidth-up(999994), bandwidth-dw(999997), bandwidth-bi(1999991) sla_map=0x1
Seq(3 H1_T33): state(alive), packet-loss(0.000%) latency(0.144), jitter(0.007), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x1
```

SD-WAN configuration and health check status on Spoke 2:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
        edit "overlay"
            set advpn-select enable
            set advpn-health-check "HUB"
        next
    end
    config members
        edit 1
            set interface "H1_T11"
            set zone "overlay"
            set cost 100
            set transport-group 1
        next
        edit 2
            set interface "H1_T22"
            set zone "overlay"
            set transport-group 1
        next
        edit 3
            set interface "H1_T33"
            set zone "overlay"
            set transport-group 2
        next
    end
    config health-check
        edit "HUB"
            set server "172.31.100.100"
            set members 3 1 2
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 100
                next
            end
        next
    end
end
```

```
# diagnose sys sdwan health-check
Health Check(HUB):
Seq(3 H1_T33): state(alive), packet-loss(0.000%) latency(0.124), jitter(0.009), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x1
Seq(1 H1_T11): state(alive), packet-loss(0.000%) latency(0.216), jitter(0.043), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x1
Seq(2 H1_T22): state(alive), packet-loss(0.000%) latency(0.184), jitter(0.012), mos(4.404),
bandwidth-up(999994), bandwidth-dw(999998), bandwidth-bi(1999992) sla_map=0x1
```

Scenario 1: Traffic matching SD-WAN rule 1

In this scenario, PC1 connected to Spoke 1 initiates an ICMP ping destined for PC1 connected to Spoke 2. Therefore, this user traffic matches SD-WAN rule 1 and triggers shortcut path selection and establishment.

The Path Manager of Spoke 1 calculates the best shortcut path by comparing transport group, link quality (for SLA mode), link cost, and member configuration order between Spoke 1 and Spoke 2.

For an SLA mode service, the following algorithm is used to consider endpoints of the best shortcut path:

1. Overlays with the same transport group
2. In-SLA overlays
3. Lowest link-cost overlays
4. Member configuration order as a final tiebreaker

Based on this algorithm, the Path Manager on Spoke 1 selects Spoke 1 H1_T11 because it is first in the priority-members order for SD-WAN rule 1, it has the lowest link cost, and it is within SLA. Likewise, the Path Manager on Spoke 1 selects Spoke 2 H1_T22 since it has the lowest link cost compared to Spoke 2 H1_T11 (which has a cost of 100), it is within SLA, and has the same transport group as Spoke 1 H1_T11. Therefore, the Path Manager of Spoke 1 calculates the best shortcut path as Spoke 1 H1_T11 to Spoke 2 H1_T22.

The Path Manager will advise IKE to establish the best shortcut and add it to SD-WAN rule 1 as follows:

```
Branch1_FGT# diagnose sys sdwan service4
 Service(1): Address Mode(IPV4) flags=0x4200 use-shortcut-sla use-shortcut
  Tie break: cfg
  Shortcut priority: 1
   Gen(11), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-compare-order
   Member sub interface(4):
     2: seq_num(1), interface(H1_T11):
        1: H1_T11_0(71)
   Members(4):
     1: Seq_num(1 H1_T11_0 overlay), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected     2: Seq_num(1 H1_T11 overlay), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
     3: Seq_num(2 H1_T22 overlay), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected
     4: Seq_num(3 H1_T33 overlay), alive, sla(0x1), gid(0), cfg_order(2), local cost(0), selected
   Src address(2):
         172.31.0.0-172.31.255.255
         10.0.3.0-10.0.3.255
   Dst address(2):
         172.31.0.0-172.31.255.255
         10.0.4.0-10.0.4.255 …
```

Since shortcut-priority is enabled, we observe that the shortcut is formed over the selected overlay path and prioritized over the parent overlay.

From the diagnostic command on Spoke 1, we observe the selected shortcut path in **bold**. (Note that the remote IP matches Spoke 2 H1_T22 in the corresponding table above.)

```
Branch1_FGT# diagnose sys sdwan advpn-session
 Session head(Branch2_FGT-0-overlay:1)
 (1) Service ID(1), last access(7809088), remote health check info(3) Selected path: local(H1_T11,
port1) gw: 172.31.3.1 remote IP: 172.31.3.105(172.31.81.2)
```

```
 Remote information:
 1: latency: 0.176267 jitter: 0.005733 pktloss: 0.000000 mos: 4.404302 sla: 0x1 cost: 0 transport_
group: 1 bandwidth up: 999994 down: 999997 bidirection: 1999991
 ipv4: 172.31.3.105(172.31.81.2) ipv6 2000:172:31:3::105(::)
 2: latency: 0.119133 jitter: 0.004800 pktloss: 0.000000 mos: 4.404331 sla: 0x1 cost: 0 transport_
group: 2 bandwidth up: 999999 down: 999997 bidirection: 1999996
 ipv4: 172.31.4.101(172.31.82.2) ipv6 1410:4b02::f088:93ee:7f00:0(c010:4b02::788a:93ee:7f00:0)
 3: latency: 0.182400 jitter: 0.008800 pktloss: 0.000000 mos: 4.404295 sla: 0x1 cost: 100
transport_group: 1 bandwidth up: 999999 down: 999997 bidirection: 1999996
 ipv4: 172.31.3.101(172.31.80.2) ipv6 2000:172:31:3::101(d88a:93ee:7f00:0:d88a:93ee:7f00:0)
```

From the diagnostic command on Spoke 2, we observe the selected shortcut in **bold**:

```
Branch2_FGT# diagnose sys sdwan health-check
 Health Check(HUB):
 Seq(3 H1_T33): state(alive), packet-loss(0.000%) latency(0.122), jitter(0.004), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x1
 Seq(1 H1_T11): state(alive), packet-loss(0.000%) latency(0.186), jitter(0.011), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x1
 Seq(2 H1_T22): state(alive), packet-loss(0.000%) latency(0.180), jitter(0.005), mos(4.404),
bandwidth-up(999994), bandwidth-dw(999997), bandwidth-bi(1999991) sla_map=0x1
 Seq(2 H1_T22_0): state(alive), packet-loss(0.000%) latency(0.265), jitter(0.011), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
```

Scenario 2: Traffic matching SD-WAN rule 2

In this scenario, PC1 connected to Spoke 1 initiates an ICMP ping destined for PC2 connected to Spoke 2. Therefore, this user traffic matches SD-WAN rule 2, and traffic will go through shortcut H1_T11_0 of Spoke 1 previously established in Scenario 1 above.

The local spoke sends a shortcut-query to the remote spoke to trigger a shortcut after ADVPN 2.0 path management makes a path decision with updated remote spoke WAN link information, which is received periodically (5 seconds) on the established shortcuts.

For an SLA mode service, the following algorithm is followed for considering endpoints of the best shortcut path:

1. Overlays with the same transport group
2. In-SLA overlays
3. Lowest link-cost overlays
4. Member configuration order as a final tiebreaker

Based on this algorithm, the Path Manager on Spoke 1 selects Spoke 1 H1_T22 because it is the first in the priority-members order for SD-WAN rule 2, it has the lowest link cost, and it is within SLA. Likewise, the Path Manager on Spoke 1 selects Spoke 2 H1_T22 since it has the lowest link cost compared to Spoke 2 H1_T11 (which has a cost of 100), it is within SLA, and has the same transport group as Spoke 1 H1_T11. Therefore, the Path Manager of Spoke 1 calculates the best shortcut path as Spoke 1 H1_T22 to Spoke 2 H1_T22.

The Path Manager will advise IKE to establish the best shortcut and add it to SD-WAN rule 2 as follows:

```
Branch1_FGT# diagnose sys sdwan service4
…
Service(2): Address Mode(IPV4) flags=0x4200 use-shortcut-sla use-shortcut
```

```
 Tie break: cfg
 Shortcut priority: 1
  Gen(12), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-compare-order
  Member sub interface(5):
    3: seq_num(2), interface(H1_T22):
       1: H1_T22_0(72)
    4: seq_num(1), interface(H1_T11):
       1: H1_T11_0(71)
  Members(5):
    1: Seq_num(2 H1_T22_0 overlay), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), selected
    2: Seq_num(1 H1_T11_0 overlay), alive, sla(0x1), gid(0), cfg_order(1), local cost(0),
selected, last_used=2023-12-05 14:34:07
    3: Seq_num(2 H1_T22 overlay), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), selected
    4: Seq_num(1 H1_T11 overlay), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected
    5: Seq_num(3 H1_T33 overlay), alive, sla(0x1), gid(0), cfg_order(2), local cost(0), selected
  Src address(2):
        172.31.0.0-172.31.255.255
        10.0.3.0-10.0.3.255
  Dst address(2):
        172.31.0.0-172.31.255.255
        10.0.40.0-10.0.40.255
…
```

The newly selected shortcut is prioritized over the previously selected shortcut as seen in the **bolded** output above.

From the diagnostic command on Spoke 1, we observe the selected shortcut path in **bold**. (Note that the remote IP matches Spoke 2 H1_T22 in the corresponding table above.)

```
Branch1_FGT# diagnose sys sdwan advpn-session
Session head(Branch2_FGT-0-overlay:2)
(1) Service ID(1), last access(8024725), remote health check info(3)
Selected path: local(H1_T11, port1) gw: 172.31.3.1  remote IP: 172.31.3.105(172.31.81.2)
Remote information:
1: latency: 0.118267 jitter: 0.004633 pktloss: 0.000000 mos: 4.404331 sla: 0x1 cost: 0 transport_
group: 2 bandwidth up: 999999 down: 999997 bidirection: 1999996
ipv4: 172.31.4.101(172.31.82.2) ipv6 180:adfb::d88a:93ee:7f00:0(d88a:93ee:7f00:0:d88a:93ee:7f00:0)
2: latency: 0.176067 jitter: 0.006567 pktloss: 0.000000 mos: 4.404301 sla: 0x1 cost: 0 transport_
group: 1 bandwidth up: 999994 down: 999997 bidirection: 1999991
ipv4: 172.31.3.105(172.31.81.2) ipv6 2000:172:31:3::105(::)
3: latency: 0.170333 jitter: 0.008133 pktloss: 0.000000 mos: 4.404302 sla: 0x1 cost: 100
transport_group: 1 bandwidth up: 999999 down: 999997 bidirection: 1999996
ipv4: 172.31.3.101(172.31.80.2) ipv6 2000:172:31:3::101(c010:4b02::788a:93ee:7f00:0)
(1) Service ID(2), last access(8024725), remote health check info(3)
Selected path: local(H1_T22, port2) gw: 172.31.3.5  remote IP: 172.31.3.105(172.31.81.2)
Remote information:
1: latency: 0.118267 jitter: 0.004633 pktloss: 0.000000 mos: 4.404331 sla: 0x1 cost: 0 transport_
group: 2 bandwidth up: 999999 down: 999997 bidirection: 1999996
ipv4: 172.31.4.101(172.31.82.2) ipv6 180:adfb::d88a:93ee:7f00:0(d88a:93ee:7f00:0:d88a:93ee:7f00:0)
2: latency: 0.176067 jitter: 0.006567 pktloss: 0.000000 mos: 4.404301 sla: 0x1 cost: 0 transport_
group: 1 bandwidth up: 999994 down: 999997 bidirection: 1999991
ipv4: 172.31.3.105(172.31.81.2) ipv6 2000:172:31:3::105(::)
3: latency: 0.170333 jitter: 0.008133 pktloss: 0.000000 mos: 4.404302 sla: 0x1 cost: 100
```

```
transport_group: 1 bandwidth up: 999999 down: 999997 bidirection: 1999996
ipv4: 172.31.3.101(172.31.80.2) ipv6 2000:172:31:3::101(c010:4b02::788a:93ee:7f00:0)
…
```

From the diagnostic command on Spoke 2, we observe the selected shortcut in **bold**:

```
Branch2_FGT# diagnose sys sdwan health-check
Health Check(HUB):
Seq(3 H1_T33): state(alive), packet-loss(0.000%) latency(0.118), jitter(0.005), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x1
Seq(1 H1_T11): state(alive), packet-loss(0.000%) latency(0.171), jitter(0.005), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x1
Seq(2 H1_T22): state(alive), packet-loss(0.000%) latency(0.175), jitter(0.006), mos(4.404),
bandwidth-up(999994), bandwidth-dw(999998), bandwidth-bi(1999992) sla_map=0x1
Seq(2 H1_T22_0): state(alive), packet-loss(0.000%) latency(0.240), jitter(0.009), mos(4.404),
bandwidth-up(1000000), bandwidth-dw(1000000), bandwidth-bi(2000000) sla_map=0x1
Seq(2 H1_T22_1): state(alive), packet-loss(0.000%) latency(0.259), jitter(0.019), mos(4.404),
bandwidth-up(1000000), bandwidth-dw(1000000), bandwidth-bi(2000000) sla_map=0x1
```

Scenario 3: Traffic matching SD-WAN rule 3

In this scenario, PC1 connected to Spoke 1 initiates an ICMP ping destined for PC3 connected to Spoke 2. Therefore, this user traffic matches SD-WAN rule 3, and traffic will go through shortcut H1_T11_0 of Spoke 1 previously established in Scenario 1 above.

The local spoke sends a shortcut-query to the remote spoke to trigger a shortcut after ADVPN 2.0 path management makes a path decision with updated remote spoke WAN link information, which is received periodically (5 seconds) on the established shortcuts.

For a best quality mode service, the following algorithm is followed for considering endpoints of the best shortcut path:

1. Overlays with the same transport group
2. Best quality overlays (link cost factor of packet loss, in this scenario)
3. Member configuration order as a final tiebreaker

Based on this algorithm, the Path Manager on Spoke 1 selects Spoke 1 H1_T33 because it is the first in the priority-members order for SD-WAN rule 3, and it has the best quality link. Likewise, the Path Manager on Spoke 1 selects Spoke 2 H1_T33 since it has the same transport group as Spoke 1 H1_T33. Therefore, the Path Manager of Spoke 1 calculates the best shortcut path as Spoke 1 H1_T33 to Spoke 2 H1_T33.

The Path Manager will advise IKE to establish the best shortcut and add it to SD-WAN rule 3 as follows:

```
Branch1_FGT# diagnose sys sdwan service4
…
Service(3): Address Mode(IPV4) flags=0x4200 use-shortcut-sla use-shortcut
 Tie break: cfg
 Shortcut priority: 3
  Gen(13), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(priority), link-cost-
factor(packet-loss), link-cost-threshold(10), health-check(HUB)
  Member sub interface(6):
    4: seq_num(3), interface(H1_T33):
       1: H1_T33_0(73)
```

```
    5: seq_num(1), interface(H1_T11):
       1: H1_T11_0(71)
    6: seq_num(2), interface(H1_T22):
       1: H1_T22_0(72)
  Members(6):
    1: Seq_num(3 H1_T33_0 overlay), alive, packet loss: 0.000%, selected
    2: Seq_num(1 H1_T11_0 overlay), alive, packet loss: 0.000%, selected, last_used=2023-12-05
14:38:02
    3: Seq_num(2 H1_T22_0 overlay), alive, packet loss: 0.000%, selected
    4: Seq_num(3 H1_T33 overlay), alive, packet loss: 0.000%, selected
    5: Seq_num(1 H1_T11 overlay), alive, packet loss: 0.000%, selected
    6: Seq_num(2 H1_T22 overlay), alive, packet loss: 0.000%, selected
  Src address(2):
        172.31.0.0-172.31.255.255
        10.0.3.0-10.0.3.255
  Dst address(2):
        172.31.0.0-172.31.255.255
        10.0.41.0-10.0.41.255
```

From the diagnostic command on Spoke 1, we observe the selected shortcut path in **bold**. (Note that the remote IP matches Spoke 2 H1_T33 in the corresponding table above.)

```
Branch1_FGT# diagnose sys sdwan advpn-session
Session head(Branch2_FGT-0-overlay:3)
(1) Service ID(3), last access(8047297), remote health check info(3)
Selected path: local(H1_T33, port3) gw: 172.31.4.1  remote IP: 172.31.4.101(172.31.82.2)
Remote information:
1: latency: 0.116600 jitter: 0.004600 pktloss: 0.000000 mos: 4.404332 sla: 0x1 cost: 0 transport_
group: 2 bandwidth up: 999999 down: 999998 bidirection: 1999997
ipv4: 172.31.4.101(172.31.82.2) ipv6 180:adfb::d88a:93ee:7f00:0(d88a:93ee:7f00:0:d88a:93ee:7f00:0)
2: latency: 0.174767 jitter: 0.005533 pktloss: 0.000000 mos: 4.404303 sla: 0x1 cost: 0 transport_
group: 1 bandwidth up: 999994 down: 999998 bidirection: 1999992
ipv4: 172.31.3.105(172.31.81.2) ipv6 2000:172:31:3::105(c010:4b02::788a:93ee:7f00:0)
3: latency: 0.172900 jitter: 0.005167 pktloss: 0.000000 mos: 4.404304 sla: 0x1 cost: 100
transport_group: 1 bandwidth up: 999999 down: 999998 bidirection: 1999997
ipv4: 172.31.3.101(172.31.80.2) ipv6 2000:172:31:3::101(::)
```

From the diagnostic command on Spoke 2, we observe the selected shortcut in **bold**:

```
Branch2_FGT# diagnose sys sdwan  health-check
Health Check(HUB):
Seq(3 H1_T33): state(alive), packet-loss(0.000%) latency(0.116), jitter(0.005), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x1
Seq(3 H1_T33_0): state(alive), packet-loss(0.000%) latency(0.113), jitter(0.005), mos(4.404),
bandwidth-up(1000000), bandwidth-dw(1000000), bandwidth-bi(2000000) sla_map=0x1
Seq(1 H1_T11): state(alive), packet-loss(0.000%) latency(0.171), jitter(0.004), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x1
Seq(2 H1_T22): state(alive), packet-loss(0.000%) latency(0.174), jitter(0.008), mos(4.404),
bandwidth-up(999994), bandwidth-dw(999998), bandwidth-bi(1999992) sla_map=0x1
Seq(2 H1_T22_0): state(alive), packet-loss(0.000%) latency(0.239), jitter(0.007), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
Seq(2 H1_T22_1): state(alive), packet-loss(0.000%) latency(0.260), jitter(0.014), mos(4.404),
bandwidth-up(1000000), bandwidth-dw(1000000), bandwidth-bi(2000000) sla_map=0x1
```

Scenario 4: Spoke 2 H1_T22 overlay link out-of-SLA

In this scenario, we place remote Spoke 2 H1_T22 out-of-SLA and observe that this link quality change is sensed by the local spoke through regular WAN link information updates on shortcuts. Because service 1 and 2 are the only rules that have new best shortcut paths when Spoke 2 H1_T22 is out-of-SLA, the local Spoke 1 will directly send shortcut-queries to the remote Spoke 2 to trigger shortcuts for service 1 and 2 after ADVPN 2.0 path management makes path decisions with updated remote spoke WAN link information.

For an SLA mode service, the following algorithm is followed for considering endpoints of the best shortcut path:

1. Overlays with the same transport group
2. In-SLA overlays
3. Lowest link-cost overlays
4. Member configuration order as a final tiebreaker

Based on this algorithm, the Path Manager on Spoke 1 still selects these Spoke 1 interfaces:

- SD-WAN Rule 1: H1_T11
- SD-WAN Rule 2: H1_T22

These are the first in the priority-members order for SD-WAN rules 1 and 2, respectively.

Based on the updated WAN link information, the Path Manager on Spoke 1 selects these Spoke 2 interfaces because they are the only remaining in-SLA VPN overlays over Internet links (transport group 1):

- SD-WAN Rule 1: H1_T11
- SD-WAN Rule 2: H1_T11

Therefore, the Path Manager of Spoke 1 calculates the best shortcut paths as follows:

- SD-WAN Rule 1: Spoke 1 H1_T11 to Spoke 2 H1_T11
- SD-WAN Rule 2: Spoke 1 H1_T22 to Spoke 2 H1_T11

The Path Manager will advise IKE to establish the best shortcuts and add them to SD-WAN rules 1 and 2 as follows:

- For SD-WAN Rule 1, H1_T11_1 is the new best shortcut.
- For SD-WAN Rule 2, H1_T22_1 is the new best shortcut.

```
# diagnose sys sdwan service4
Service(1): Address Mode(IPV4) flags=0x4200 use-shortcut-sla use-shortcut
 Tie break: cfg
 Shortcut priority: 1
  Gen(17), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-compare-order
  Member sub interface(8):
    6: seq_num(1), interface(H1_T11):
       1: H1_T11_0(74)
       2: H1_T11_1(75)
    7: seq_num(2), interface(H1_T22):
       1: H1_T22_0(72)
       2: H1_T22_1(76)
    8: seq_num(3), interface(H1_T33):
       1: H1_T33_0(73)
  Members(8):
```

```
    1: Seq_num(1 H1_T11_0 overlay), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), selected
    2: Seq_num(1 H1_T11_1 overlay), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), selected

    3: Seq_num(2 H1_T22_0 overlay), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected
    4: Seq_num(2 H1_T22_1 overlay), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected

    5: Seq_num(3 H1_T33_0 overlay), alive, sla(0x1), gid(0), cfg_order(2), local cost(0), selected
    6: Seq_num(1 H1_T11 overlay), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), selected
    7: Seq_num(2 H1_T22 overlay), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected
    8: Seq_num(3 H1_T33 overlay), alive, sla(0x1), gid(0), cfg_order(2), local cost(0), selected
  Src address(2):
        172.31.0.0-172.31.255.255
        10.0.3.0-10.0.3.255
  Dst address(2):
        172.31.0.0-172.31.255.255
        10.0.4.0-10.0.4.255

Service(2): Address Mode(IPV4) flags=0x4200 use-shortcut-sla use-shortcut
 Tie break: cfg
 Shortcut priority: 1
  Gen(17), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-compare-order
  Member sub interface(8):
    6: seq_num(2), interface(H1_T22):
       1: H1_T22_0(72)
       2: H1_T22_1(76)
    7: seq_num(1), interface(H1_T11):
       1: H1_T11_0(74)
       2: H1_T11_1(75)
    8: seq_num(3), interface(H1_T33):
       1: H1_T33_0(73)
  Members(8):
    1: Seq_num(2 H1_T22_0 overlay), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), selected
    2: Seq_num(2 H1_T22_1 overlay), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), selected
    3: Seq_num(1 H1_T11_1 overlay), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected
    4: Seq_num(1 H1_T11_0 overlay), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected
    5: Seq_num(3 H1_T33_0 overlay), alive, sla(0x1), gid(0), cfg_order(2), local cost(0), selected
    6: Seq_num(2 H1_T22 overlay), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), selected
    7: Seq_num(1 H1_T11 overlay), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected
    8: Seq_num(3 H1_T33 overlay), alive, sla(0x1), gid(0), cfg_order(2), local cost(0), selected
  Src address(2):
        172.31.0.0-172.31.255.255
        10.0.3.0-10.0.3.255
  Dst address(2):
        172.31.0.0-172.31.255.255
        10.0.40.0-10.0.40.255
…
```

From the diagnostic command on Spoke 1, we observe the newly selected shortcut paths in **bold**. (Note that the remote IP 172.31.80.2 matches Spoke 2 H1_T11, which is the VPN overlay over the Internet link with cost 100 in the corresponding table above.)

```
# diagnose sys sdwan advpn-session
Session head(Branch2_FGT-0-overlay:3)
(1) Service ID(1), last access(8293060), remote health check info(3)
Selected path: local(H1_T11, port1) gw: 172.31.3.1  remote IP: 172.31.3.101(172.31.80.2)
Remote information:
1: latency: 0.119500 jitter: 0.006067 pktloss: 0.000000 mos: 4.404329 sla: 0x1 cost: 0 transport_
group: 2 bandwidth up: 999999 down: 999997 bidirection: 1999996
ipv4: 172.31.4.101(172.31.82.2) ipv6 180:adfb::d88a:93ee:7f00:0(d88a:93ee:7f00:0:d88a:93ee:7f00:0)
2: latency: 250.170761 jitter: 0.011500 pktloss: 0.000000 mos: 3.992655 sla: 0x0 cost: 0
transport_group: 1 bandwidth up: 999994 down: 999997 bidirection: 1999991
ipv4: 172.31.3.105(172.31.81.2) ipv6 2000:172:31:3::105(c010:4b02::788a:93ee:7f00:0)
3: latency: 0.182200 jitter: 0.012000 pktloss: 0.000000 mos: 4.404292 sla: 0x1 cost: 100
transport_group: 1 bandwidth up: 999999 down: 999997 bidirection: 1999996
ipv4: 172.31.3.101(172.31.80.2) ipv6 2000:172:31:3::101(::)
(1) Service ID(2), last access(8293060), remote health check info(3)
Selected path: local(H1_T22, port2) gw: 172.31.3.5  remote IP: 172.31.3.101(172.31.80.2)
Remote information:
1: latency: 0.119500 jitter: 0.006067 pktloss: 0.000000 mos: 4.404329 sla: 0x1 cost: 0 transport_
group: 2 bandwidth up: 999999 down: 999997 bidirection: 1999996
ipv4: 172.31.4.101(172.31.82.2) ipv6 180:adfb::d88a:93ee:7f00:0(d88a:93ee:7f00:0:d88a:93ee:7f00:0)
2: latency: 250.170761 jitter: 0.011500 pktloss: 0.000000 mos: 3.992655 sla: 0x0 cost: 0
transport_group: 1 bandwidth up: 999994 down: 999997 bidirection: 1999991
ipv4: 172.31.3.105(172.31.81.2) ipv6 2000:172:31:3::105(c010:4b02::788a:93ee:7f00:0)
3: latency: 0.182200 jitter: 0.012000 pktloss: 0.000000 mos: 4.404292 sla: 0x1 cost: 100
transport_group: 1 bandwidth up: 999999 down: 999997 bidirection: 1999996
ipv4: 172.31.3.101(172.31.80.2) ipv6 2000:172:31:3::101(::)
```

From the diagnostic command on Spoke 2, we observe the selected shortcuts in **bold**:

```
Branch2_FGT# diagnose sys sdwan  health-check
Health Check(HUB):
Seq(3 H1_T33): state(alive), packet-loss(0.000%) latency(0.120), jitter(0.007), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x1
Seq(3 H1_T33_0): state(alive), packet-loss(0.000%) latency(0.128), jitter(0.003), mos(4.404),
bandwidth-up(1000000), bandwidth-dw(1000000), bandwidth-bi(2000000) sla_map=0x1
Seq(1 H1_T11): state(alive), packet-loss(0.000%) latency(0.180), jitter(0.008), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x1
Seq(1 H1_T11_0): state(alive), packet-loss(0.000%) latency(0.259), jitter(0.023), mos(4.404),
bandwidth-up(1000000), bandwidth-dw(1000000), bandwidth-bi(2000000) sla_map=0x1
Seq(1 H1_T11_1): state(alive), packet-loss(0.000%) latency(0.257), jitter(0.014), mos(4.404),
bandwidth-up(1000000), bandwidth-dw(1000000), bandwidth-bi(2000000) sla_map=0x1
Seq(2 H1_T22): state(alive), packet-loss(0.000%) latency(250.169), jitter(0.009), mos(3.993),
bandwidth-up(999994), bandwidth-dw(999997), bandwidth-bi(1999991) sla_map=0x0
Seq(2 H1_T22_1): state(alive), packet-loss(0.000%) latency(0.245), jitter(0.013), mos(4.404),
bandwidth-up(1000000), bandwidth-dw(1000000), bandwidth-bi(2000000) sla_map=0x1
Seq(2 H1_T22_0): state(alive), packet-loss(0.000%) latency(0.223), jitter(0.005), mos(4.404),
bandwidth-up(1000000), bandwidth-dw(1000000), bandwidth-bi(2000000) sla_map=0x1
```

Scenario 5: Traffic matching SD-WAN rule configured for load balancing

This example relies on the same Network Topology on page 1346 used above, except Spoke 1 has a single SD-WAN rule/service that uses the load balancing strategy with SLA targets. For details, see Load balancing

strategy with SLA targets on page 1129.

This section shows the SD-WAN configuration, selected IPsec configuration, and health check status on Spoke 1: on page 1358 and Spoke 2: on page 1360:

Spoke 1:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
        edit "overlay"
            set advpn-select enable
            set advpn-health-check "HUB"
        next
    end
    config members
        edit 1
            set interface "H1_T11"
            set zone "overlay"
            set transport-group 1
        next
        edit 2
            set interface "H1_T22"
            set zone "overlay"
            set transport-group 1
        next
        edit 3
            set interface "H1_T33"
            set zone "overlay"
            set transport-group 2
        next
    end
    config health-check
        edit "HUB"
            set server "172.31.100.100"
            set members 1 2 3
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 100
                next
            end
        next
    end
    config service
        edit 1
            set name "1"
            set load-balance enable
            set mode sla
            set dst "CORP_LAN"
```

```
            set src "CORP_LAN"
            config sla
                edit "HUB"
                    set id 1
                next
            end
            set priority-members 1 2 3
        next
    end
end
config vpn ipsec phase1-interface
    edit "H1_T11"
        ...
        set idle-timeout enable
        set shared-idle-timeout enable
        set idle-timeoutinterval 5
        ...
    next
end

config vpn ipsec phase1-interface
    edit "H1_T22"
        ...
        set idle-timeout enable
        set shared-idle-timeout enable
        set idle-timeoutinterval 5
        ...
    next
end

config vpn ipsec phase1-interface
    edit "H1_T33"
        ...
        set idle-timeout enable
        set shared-idle-timeout enable
        set idle-timeoutinterval 5
        ...
    next
end
```

```
# diagnose sys sdwan health-check
Health Check(HUB):
Seq(1 H1_T11): state(alive), packet-loss(0.000%) latency(0.223), jitter(0.018), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x1
Seq(2 H1_T22): state(alive), packet-loss(0.000%) latency(0.191), jitter(0.009), mos(4.404),
bandwidth-up(999993), bandwidth-dw(999998), bandwidth-bi(1999991) sla_map=0x1
Seq(3 H1_T33): state(alive), packet-loss(0.000%) latency(0.139), jitter(0.007), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x1
```

Spoke 2:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
        edit "overlay"
            set advpn-select enable
            set advpn-health-check "HUB"
        next
    end
    config members
        edit 1
            set interface "H1_T11"
            set zone "overlay"
            set transport-group 1
        next
        edit 2
            set interface "H1_T22"
            set zone "overlay"
            set transport-group 1
        next
        edit 3
            set interface "H1_T33"
            set zone "overlay"
            set transport-group 2
        next
    end
    config health-check
        edit "HUB"
            set server "172.31.100.100"
            set members 3 1 2
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 100
                next
            end
        next
    end
end
config vpn ipsec phase1-interface
    edit "H1_T11"
        ...
        set idle-timeout enable
        set shared-idle-timeout enable
        set idle-timeoutinterval 5
        ...
    next
end
```

```
config vpn ipsec phase1-interface
    edit "H1_T22"
        ...
        set idle-timeout enable
        set shared-idle-timeout enable
        set idle-timeoutinterval 5
        ...
    next
end

config vpn ipsec phase1-interface
    edit "H1_T33"
        ...
        set idle-timeout enable
        set shared-idle-timeout enable
        set idle-timeoutinterval 5
        ...
    next
end
```

```
# diagnose sys sdwan health-check
Health Check(HUB):
Seq(3 H1_T33): state(alive), packet-loss(0.000%) latency(0.148), jitter(0.021), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x1
Seq(1 H1_T11): state(alive), packet-loss(0.000%) latency(0.183), jitter(0.010), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x1
Seq(2 H1_T22): state(alive), packet-loss(0.000%) latency(0.163), jitter(0.005), mos(4.404),
bandwidth-up(999994), bandwidth-dw(999998), bandwidth-bi(1999992) sla_map=0x1
```

In this scenario, PC1 connected to Spoke 1 initiates an ICMP ping destined for PC1 connected to Spoke 2. Therefore, this user traffic matches SD-WAN rule 1 and triggers shortcut path selection and establishment.

On Spoke 1, in the IKE debug (diagnose debug application ike -1), debug messages indicate that multiple direct shortcut-query packets are being sent to Spoke 2:

```
ike :VWL_ADVPN_MSG_T_TRIGGER
ike V=root:0 looking up shortcut by addr 172.31.80.2, resp-name:H1_T11, name H1_T22, peer-addr
172.31.3.101:0
ike V=root:0:H1_T22: send shortcut-query
...
ike :VWL_ADVPN_MSG_T_TRIGGER
ike V=root:0 looking up shortcut by addr 172.31.81.2, resp-name:H1_T22, name H1_T22, peer-addr
172.31.3.105:0
ike V=root:0:H1_T22: send shortcut-query
...
ike :VWL_ADVPN_MSG_T_TRIGGER
ike V=root:0 looking up shortcut by addr 172.31.82.2, resp-name:H1_T33, name H1_T33, peer-addr
172.31.4.101:0
ike V=root:0:H1_T33: send shortcut-query
...
```

From the diagnostic command on Spoke 1, observe that multiple shortcuts are triggered in **bold** based on the ADVPN 2.0 path management calculation where in-SLA overlays within the same transport group were selected.

```
Branch1_FGT# diagnose system sdwan service4
Service(1): Address Mode(IPV4) flags=0x24200 use-shortcut-sla use-shortcut
 Tie break: cfg
 Shortcut priority: 3
  Gen(69), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla  hash-mode=round-
robin)
  Member sub interface(8):
    1: seq_num(1), interface(H1_T11):
       1: H1_T11_0(103)
       2: H1_T11_1(104)
    2: seq_num(2), interface(H1_T22):
       1: H1_T22_0(105)
       2: H1_T22_1(106)
    3: seq_num(3), interface(H1_T33):
       1: H1_T33_0(100)
  Members(8):
    1: Seq_num(1 H1_T11 overlay), alive, sla(0x1), gid(2), num of pass(1), selected
    2: Seq_num(2 H1_T22 overlay), alive, sla(0x1), gid(2), num of pass(1), selected
    3: Seq_num(3 H1_T33 overlay), alive, sla(0x1), gid(2), num of pass(1), selected
    4: Seq_num(3 H1_T33_0 overlay), alive, sla(0x1), gid(2), num of pass(1), selected
    5: Seq_num(1 H1_T11_0 overlay), alive, sla(0x1), gid(2), num of pass(1), selected
    6: Seq_num(1 H1_T11_1 overlay), alive, sla(0x1), gid(2), num of pass(1), selected
    7: Seq_num(2 H1_T22_0 overlay), alive, sla(0x1), gid(2), num of pass(1), selected
    8: Seq_num(2 H1_T22_1 overlay), alive, sla(0x1), gid(2), num of pass(1), selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        10.0.0.0-10.255.255.255
```

From the diagnostic command on Spoke 2, observe the shortcuts in **bold**:

```
Branch2_FGT# diagnose sys sdwan  health-check
 Health Check(HUB):
 Seq(3 H1_T33): state(alive), packet-loss(0.000%) latency(0.120), jitter(0.007), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x1
 Seq(3 H1_T33_0): state(alive), packet-loss(0.000%) latency(0.128), jitter(0.003), mos(4.404),
bandwidth-up(1000000), bandwidth-dw(1000000), bandwidth-bi(2000000) sla_map=0x1
 Seq(1 H1_T11): state(alive), packet-loss(0.000%) latency(0.180), jitter(0.008), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x1
 Seq(1 H1_T11_0): state(alive), packet-loss(0.000%) latency(0.259), jitter(0.023), mos(4.404),
bandwidth-up(1000000), bandwidth-dw(1000000), bandwidth-bi(2000000) sla_map=0x1
 Seq(1 H1_T11_1): state(alive), packet-loss(0.000%) latency(0.257), jitter(0.014), mos(4.404),
bandwidth-up(1000000), bandwidth-dw(1000000), bandwidth-bi(2000000) sla_map=0x1
 Seq(2 H1_T22): state(alive), packet-loss(0.000%) latency(250.169), jitter(0.009), mos(3.993),
bandwidth-up(999994), bandwidth-dw(999997), bandwidth-bi(1999991) sla_map=0x0
 Seq(2 H1_T22_1): state(alive), packet-loss(0.000%) latency(0.245), jitter(0.013), mos(4.404),
bandwidth-up(1000000), bandwidth-dw(1000000), bandwidth-bi(2000000) sla_map=0x1
 Seq(2 H1_T22_0): state(alive), packet-loss(0.000%) latency(0.223), jitter(0.005), mos(4.404),
bandwidth-up(1000000), bandwidth-dw(1000000), bandwidth-bi(2000000) sla_map=0x1
```

At this point, PC1 connected to Spoke 1 initiated multiple ICMP pings destined for PC1 connected to Spoke 2. The packet capture diagnostic command on Spoke 1 demonstrates that these ICMP pings have been load balanced over all shortcuts:

```
Branch1_FGT# diagnose sniffer packet any 'host 10.0.4.2' 4
interfaces=[any]
filters=[host 10.0.4.2]
3.481994 port4 in 10.0.3.2 -> 10.0.4.2: icmp: echo request
3.482103 H1_T11_1 out 10.0.3.2 -> 10.0.4.2: icmp: echo request
3.482799 H1_T11_1 in 10.0.4.2 -> 10.0.3.2: icmp: echo reply
3.482928 port4 out 10.0.4.2 -> 10.0.3.2: icmp: echo reply
4.614480 port4 in 10.0.3.2 -> 10.0.4.2: icmp: echo request
4.614580 H1_T33_0 out 10.0.3.2 -> 10.0.4.2: icmp: echo request
4.615122 H1_T33_0 in 10.0.4.2 -> 10.0.3.2: icmp: echo reply
4.615152 port4 out 10.0.4.2 -> 10.0.3.2: icmp: echo reply
5.286394 port4 in 10.0.3.2 -> 10.0.4.2: icmp: echo request
5.286497 H1_T22_0 out 10.0.3.2 -> 10.0.4.2: icmp: echo request
5.287129 H1_T22_0 in 10.0.4.2 -> 10.0.3.2: icmp: echo reply
5.287155 port4 out 10.0.4.2 -> 10.0.3.2: icmp: echo reply
6.079759 port4 in 10.0.3.2 -> 10.0.4.2: icmp: echo request
6.079883 H1_T22_1 out 10.0.3.2 -> 10.0.4.2: icmp: echo request
6.080496 H1_T22_1 in 10.0.4.2 -> 10.0.3.2: icmp: echo reply
6.080537 port4 out 10.0.4.2 -> 10.0.3.2: icmp: echo reply
7.983357 port4 in 10.0.3.2 -> 10.0.4.2: icmp: echo request
7.983447 H1_T11_0 out 10.0.3.2 -> 10.0.4.2: icmp: echo request
7.984078 H1_T11_0 in 10.0.4.2 -> 10.0.3.2: icmp: echo reply
7.984120 port4 out 10.0.4.2 -> 10.0.3.2: icmp: echo reply
```

Without user traffic traversing the shortcut during the idle interval time, from the diagnostic command on Spoke 1, observe that all shortcuts have been removed:

Shared idle timeout keeps all tunnels up if any tunnel is passing traffic. Without shared idle timeout, tunnels are torn down individually when no traffic is observed on the specific tunnel and rebuilt at the start of the next cycle if a single tunnel remains active.

```
Branch1_FGT# diagnose system sdwan service4
Service(1): Address Mode(IPV4) flags=0x24200 use-shortcut-sla use-shortcut
 Tie break: cfg
 Shortcut priority: 3
  Gen(16), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla  hash-mode=round-
robin)
  Members(3):
    1: Seq_num(1 H1_T11 overlay), alive, sla(0x1), gid(2), num of pass(1), selected
    2: Seq_num(2 H1_T22 overlay), alive, sla(0x1), gid(2), num of pass(1), selected
    3: Seq_num(3 H1_T33 overlay), alive, sla(0x1), gid(2), num of pass(1), selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        10.0.0.0-10.255.255.255
```

Example SD-WAN overlay placeholders using ADVPN 2.0

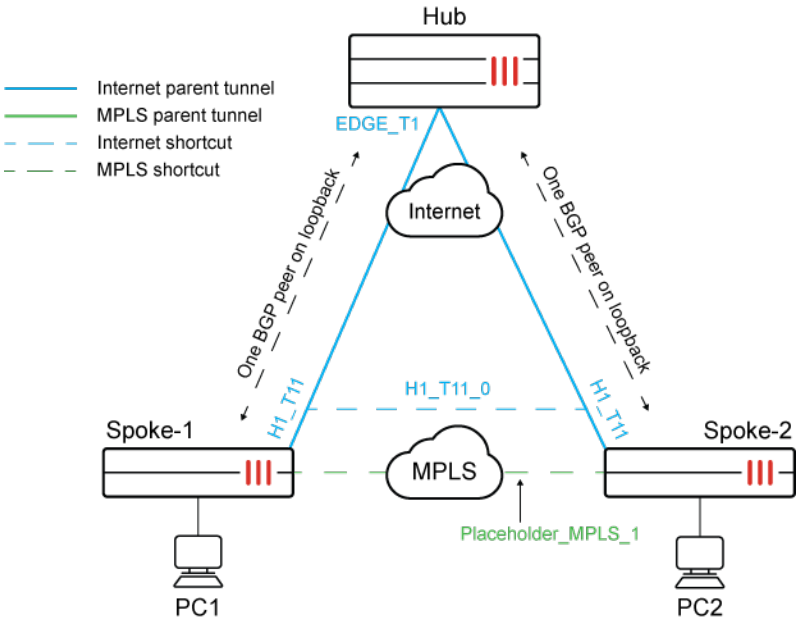
In a typical hub and spoke SD-WAN configuration, the hub uses the same underlay transports as the spokes. However, in unique circumstances, spokes may use different transports than the hub. For example, spokes may have direct MPLS connections that are not connected to the hub. Hubs are not necessarily connected to all the same underlay transports as spokes. In these instances, ADVPN 2.0 can use overlay placeholders to trigger shortcuts between spokes over transports to which hubs are not connected. Once the regular shortcut and placeholder shortcuts are established, SD-WAN will steer traffic based on the service rule, such as which link has the best quality.

In this SD-WAN example, the Hub, Spoke-1, and Spoke-2 all have broadband WAN connections. However, Spoke-1 and Spoke-2 have an additional MPLS connection between them. With ADVPN 2.0 enabled, Spoke-1 and Spoke-2 have regular parent tunnels (H1_T11) to the Hub. Spoke-1 and Spoke-2 also have placeholder parent tunnels configured (Placeholder_MPLS_1), where `auto-discovery-dialup-placeholder` is enabled and `remote-gateway` isn't statically specified. As a placeholder tunnel, no connection is established between the spoke and the hub. Instead, the tunnel interface is utilized in the SD-WAN configurations as if it were a regular overlay because shortcuts can be established between spokes in a placeholder overlay.

Traffic is sent from PC-1 to PC-2. This traffic matches an SD-WAN rule that include two interfaces: a VPN tunnel to the Hub, and a placeholder tunnel. As only the first tunnel is alive, the first bit of traffic goes through this tunnel to the Hub and triggers `SHORTCUT_QUERY/SHORTCUT_REPLY` exchange with the Hub. When Spoke-1 receives `SHORTCUT_REPLY` message, which includes WAN and SD-WAN information about Spoke-2's eligible shortcut interfaces, Spoke-1 calculates and creates the shortcut between regular parent tunnels and triggers a shortcut between the placeholder parent tunnels too.

As the `SHORTCUT_REPLY` from the Hub contains additional information for a second WAN interface on Spoke-2, a `SHORTCUT_QUERY` is triggered on Spoke-1 to see if another shortcut can be established. This `SHORTCUT_QUERY` utilizes the H1_T11 tunnel to facilitate the `SHORTCUT_QUERY/SHORTCUT_REPLY` exchange with the Hub.

Health-checks, which are automatically running on the regular shortcut and the placeholder shortcut, help decide which interface to use for forwarding the remaining traffic.



Settings on the spoke relevant to the example:

```
config vpn ipsec phase1-interface
    edit <placeholder_phase1_interface_name>
        set type dynamic
        ...
        set net-device enable
        ...
        set auto-discovery-dialup-placeholder enable
    next
end
```

To enable overlay placeholder tunnels on Spoke-1 and Spoke-2:

1. Configure key components on Spoke-1:
   a. Configure a parent tunnel and a placeholder tunnel:
      Tunnel H1_T11 is the parent tunnel, and tunnel Placeholder_MPLS_1 is the placeholder tunnel:

```
config vpn ipsec phase1-interface
    edit "H1_T11"
        set interface "port1"
        set ike-version 2
        set keylife 28800
        set peertype any
        set net-device enable
        set exchange-ip-addr4 172.31.0.65
        set proposal aes256gcm-prfsha384
        set add-route disable
        set dpd on-idle
        set idle-timeout enable
        set idle-timeoutinterval 5
        set auto-discovery-receiver enable
        set encapsulation vpn-id-ipip
        set network-overlay enable
        set network-id 11
        set transport udp
        set remote-gw 172.31.1.1
        set psksecret ENC
fRkYaHGx3MoAASxMsHoJSObbOfphOmBntEpKhZAY8/9OjjqJxbmTehe4Z7LOGTvjFRXTULiPBkfPupnr0JtBo0KJb0
IoUfJ8zWU3mJttCyfHNaDBXG/t5d0D93iu3ZlRoSs1EBRD7KX5fO1QnSHm0maXTJZzrKLWVSaRztdOF246z/RbEXFz
lm4+64ccLHG/3X/Ha1lmMjY3dkVA
        set dpd-retryinterval 5
    next
    edit "Placeholder_MPLS_1"
        set type dynamic
        set interface "port3"
        set ike-version 2
        set keylife 28800
        set peertype any
        set net-device enable
        set exchange-ip-addr4 172.31.0.65
```

```
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set add-route disable
        set dpd on-idle
        set idle-timeout enable
        set idle-timeoutinterval 5
        set auto-discovery-receiver enable
        set auto-discovery-dialup-placeholder enable
        set encapsulation vpn-id-ipip
        set network-overlay enable
        set network-id 250
        set transport udp
        set psksecret ENC
e3XlBFTNA+fk4Kn7NMLfzJLgU0Pk6ssX9oJ7VSA2Q371x+oqIRSomFhRyZiErb1j07FmGT/lPUFHAB2BJG7vhNIh0S
SjEqThai8rIKWqHsIpjHwUAM0/maBzH1iTXZAyjEX0vFiu65QSLAoQzfgXkfX16P82Q1gjIim/GX4phzA+cIYJnfeg
bqXylFg4fxkqCrYFGFlmMjY3dkVA
        set dpd-retryinterval 60
    next
end
```

b. Configure SD-WAN:

Enable ADVPN 2.0 and health-checks for the overlay. Configure the SD-WAN members and their transport groups.

```
config system sdwan
    set status enable
    config zone
        edit "overlay"
            set advpn-select enable
            set advpn-health-check "HUB"
        next
    end
    config members
        edit 4
            set interface "H1_T11"
            set zone "overlay"
            set source 172.31.0.65
            set priority 10
            set transport-group 1
        next
        edit 13
            set interface "Placeholder_MPLS_1"
            set zone "overlay"
            set source 172.31.0.65
            set priority 10
            set transport-group 2
        next
    end
    config health-check
        edit "HUB"
            set server "172.31.100.100"
            set members 4 13
```

```
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 100
                next
            end
        next
    end
    config service
        edit 1
            set name "1"
            set mode sla
            set dst "spoke-2_LAN-1"
            set src "spoke-1_LAN-1"
            config sla
                edit "HUB"
                    set id 1
                next
            end
            set priority-members 4 13
        next
    end
end
```

2. Configure key components on Spoke-2:
   a. Configure a parent tunnel and a placeholder tunnel:
      Tunnel H1_T11 is the parent tunnel, and tunnel Placeholder_MPLS_1 is the placeholder tunnel:

```
   config vpn ipsec phase1-interface
   edit "H1_T11"
        set interface "port1"
        set ike-version 2
        set keylife 28800
        set peertype any
        set net-device enable
        set exchange-ip-addr4 172.31.0.66
        set proposal aes256gcm-prfsha384
        set add-route disable
        set dpd on-idle
        set idle-timeout enable
        set idle-timeoutinterval 5
        set auto-discovery-receiver enable
        set encapsulation vpn-id-ipip
        set network-overlay enable
        set network-id 11
        set transport udp
        set remote-gw 172.31.1.1
        set psksecret ENC
7tP6FktH3z3nIpiGoANS/PbxfMOeLf0KCp65MbG2/yBmeAZz0XXcF9XMbnfBtaWhZhoTnGx086ST1aRmLDxCfVu/BM
6S+j1XOUSxSZb6xgwRoaQsFTFaXq/8PYjUzJK7SdPZiJDsCrGaslMR/mTLzFTasA4y9YGteJM+tELW5K0C6Ntrwlq8
UnAtrdFMGN/3BrLNyllmMjY3dkVA
```

```
        set dpd-retryinterval 5
    next
    edit "Placeholder_MPLS_1"
        set type dynamic
        set interface "port3"
        set ike-version 2
        set keylife 28800
        set peertype any
        set net-device enable
        set exchange-ip-addr4 172.31.0.66
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set add-route disable
        set dpd on-idle
        set idle-timeout enable
        set idle-timeoutinterval 5
        set auto-discovery-receiver enable
        set auto-discovery-dialup-placeholder enable
        set encapsulation vpn-id-ipip
        set network-overlay enable
        set network-id 250
        set transport udp
        set psksecret ENC
Un++iggqXo1wU8fAArjS9JY5rsAcd8+Xw3WR7MtRVLU0BpXT56bAwrP6wQ4gpMt32ABTQ/mAddlq1Mq5LxCV0X+lYa
xj2vVIulTb1n71Wn1WeTbDXjiTzDbrMnp4fM9+1uBKwy+RJ+u782IpI9WRhCWE/H4LNgZhYzUmyluTCouNoBpJyzdJ
rekAXOtSvxLJfju/7FlmMjY3dkVA
        set dpd-retryinterval 60
    next
end
```

b. Configure SD-WAN:

Enable ADVPN 2.0 and health-checks for the overlay. Configure the SD-WAN members and their transport groups. The internet overlay (H1_T11) is added to transport group 1, and the MPLS overlay (Placeholder_MPLS_1) is added to transport group 2.

```
config system sdwan
    set status enable
    config zone
        edit "overlay"
            set advpn-select enable
            set advpn-health-check "HUB"
        next
    end
    config members
        edit 4
            set interface "H1_T11"
            set zone "overlay"
            set source 172.31.0.66
            set priority 10
            set transport-group 1
        next
        edit 13
```

```
            set interface "Placeholder_MPLS_1"
            set zone "overlay"
            set source 172.31.0.66
            set priority 10
            set transport-group 2
        next
    end
    config health-check
        edit "HUB"
            set server "172.31.100.100"
            set members 4 13
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 100
                next
            end
        next
    end
end
```

To check the health status and traffic:

1. Check the health status on Spoke-1 and Spoke-2, and check the SD-WAN status of Spoke-1:
   a. Check the health of Spoke-1:
   The placeholder tunnel (Placeholder_MPLS_1) is dead.

```
# diagnose sys sdwan  health-check
Health Check(HUB):
Seq(4 H1_T11): state(alive), packet-loss(0.000%), latency(0.235), jitter(0.011), mos
(4.404), bandwidth-up(999998), bandwidth-dw(999998), bandwidth-bi(1999996), sla_map=0x1
Seq(13 Placeholder_MPLS_1): state(dead), packet-loss(100.000%), sla_map=0x0
```

   b. Check the health of Spoke-2:
   The placeholder tunnel (Placeholder_MPLS_1) is dead.

```
# diagnose sys sdwan  health-check
Health Check(HUB):
Seq(4 H1_T11): state(alive), packet-loss(0.000%), latency(0.250), jitter(0.041), mos
(4.404), bandwidth-up(999998), bandwidth-dw(999998), bandwidth-bi(1999996), sla_map=0x1
Seq(13 Placeholder_MPLS_1): state(dead), packet-loss(100.000%), sla_map=0x0
```

   c. Check the SD-WAN status of Spoke-1:

```
# diagnose sys sdwan service4
Service(1): Address Mode(IPV4) flags=0x4200 use-shortcut-sla use-shortcut
 Tie break: cfg
 Shortcut priority: 3
  Gen(21), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-compare-
order
  Members(2):
```

```
    1: Seq_num(4 H1_T11 overlay), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    2: Seq_num(13 Placeholder_MPLS_1 overlay), dead, sla(0x0), gid(0), cfg_order(1), local
cost(0)
  Src address(1):
        10.0.3.0-10.0.3.255
  Dst address(1):
        10.0.4.0-10.0.4.255
```

2. Send traffic from PC-1 to PC-2:

   When Spoke-1 receives SHORTCUT_REPLY message, which includes SD-WAN information about Spoke-2, Spoke-1 calculates and creates the shortcut between regular parent tunnels, and it triggers a shortcut between the placeholder tunnels too.

   a. Check the health of Spoke-1:

      The health-check is automatically running on the regular shortcut (H1_T11_0) and on the placeholder shortcut (Placeholder_MPLS_1_0).

```
# diagnose sys sdwan  health-check
Health Check(HUB):
Seq(4 H1_T11): state(alive), packet-loss(0.000%), latency(0.239), jitter(0.017), mos
(4.404), bandwidth-up(999998), bandwidth-dw(999998), bandwidth-bi(1999996), sla_map=0x1
Seq(4 H1_T11_0): state(alive), packet-loss(0.000%), latency(0.260), jitter(0.010), mos
(4.404), bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998), sla_map=0x1

Seq(13 Placeholder_MPLS_1): state(dead), packet-loss(100.000%), sla_map=0x0
Seq(13 Placeholder_MPLS_1_0): state(alive), packet-loss(0.000%), latency(0.139), jitter
(0.006), mos(4.404), bandwidth-up(1000000), bandwidth-dw(1000000), bandwidth-bi(2000000),
sla_map=0x1
```

   b. Check the health of Spoke-2:

      The health-check is automatically running on the regular shortcut (H1_T11_0) and on the placeholder shortcut (Placeholder_MPLS_1_0).

```
# diagnose sys sdwan  health-check
Health Check(HUB):
Seq(4 H1_T11): state(alive), packet-loss(0.000%), latency(0.214), jitter(0.012), mos
(4.404), bandwidth-up(999998), bandwidth-dw(999997), bandwidth-bi(1999995), sla_map=0x1
Seq(4 H1_T11_0): state(alive), packet-loss(0.000%), latency(0.296), jitter(0.030), mos
(4.404), bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997), sla_map=0x1

Seq(13 Placeholder_MPLS_1): state(dead), packet-loss(100.000%), sla_map=0x0
Seq(13 Placeholder_MPLS_1_0): state(alive), packet-loss(0.000%), latency(0.159), jitter
(0.023), mos(4.404), bandwidth-up(1000000), bandwidth-dw(1000000), bandwidth-bi(2000000),
sla_map=0x1
```

   c. Check the SD-WAN status of Spoke-1:

      The regular shortcut (H1_T11_0 overlay) is preferred, because it is the better performing link, as determined by SD-WAN intelligence.

```
# diagnose sys sdwan service4
```

```
Service(1): Address Mode(IPV4) flags=0x4200 use-shortcut-sla use-shortcut
 Tie break: cfg
 Shortcut priority: 3
  Gen(33), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-compare-
order
  Member sub interface(4):
    3: seq_num(4), interface(H1_T11):
       1: H1_T11_0(1152)
    4: seq_num(13), interface(dummy_MPLS_1):
       1: dummy_MPLS_1_0(1153)
  Members(4):
    1: Seq_num(4 H1_T11_0 overlay), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    2: Seq_num(13 Placeholder_MPLS_1_0 overlay), alive, sla(0x1), gid(0), cfg_order(1),
local cost(0), selected
    3: Seq_num(4 H1_T11 overlay), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    4: Seq_num(13 Placeholder_MPLS_1 overlay), dead, sla(0x0), gid(0), cfg_order(1), local
cost(0)
  Src address(1):
        10.0.3.0-10.0.3.255
  Dst address(1):
        10.0.4.0-10.0.4.255
```

3. Diagnose the sniffer packet:

   The first few packets travel over the regular parent tunnel (H1_T11), and then the packets switch to the regular shortcut tunnel (H1_T11_0).

```
# diagnose sniffer packet any 'host 10.0.4.2' 4
interfaces=[any]
filters=[host 10.0.4.2]
4.967575 port4 in 10.0.3.2 -> 10.0.4.2: icmp: echo request
4.967976 H1_T11 out 10.0.3.2 -> 10.0.4.2: icmp: echo request
4.969587 H1_T11 in 10.0.4.2 -> 10.0.3.2: icmp: echo reply
4.969629 port4 out 10.0.4.2 -> 10.0.3.2: icmp: echo reply
5.968744 port4 in 10.0.3.2 -> 10.0.4.2: icmp: echo request
5.968838 H1_T11 out 10.0.3.2 -> 10.0.4.2: icmp: echo request
5.969785 H1_T11 in 10.0.4.2 -> 10.0.3.2: icmp: echo reply
5.969842 port4 out 10.0.4.2 -> 10.0.3.2: icmp: echo reply
6.969970 port4 in 10.0.3.2 -> 10.0.4.2: icmp: echo request
6.970187 H1_T11 out 10.0.3.2 -> 10.0.4.2: icmp: echo request
9.969251 port4 in 10.0.3.2 -> 10.0.4.2: icmp: echo request
9.969366 H1_T11_0 out 10.0.3.2 -> 10.0.4.2: icmp: echo request
9.970229 H1_T11_0 in 10.0.4.2 -> 10.0.3.2: icmp: echo reply
9.970278 port4 out 10.0.4.2 -> 10.0.3.2: icmp: echo reply
10.969982 port4 in 10.0.3.2 -> 10.0.4.2: icmp: echo request
10.970199 H1_T11_0 out 10.0.3.2 -> 10.0.4.2: icmp: echo request
```

4. When the regular shortcut tunnel (H1_T11_0) is out of SLA, traffic switches to the placeholder shortcut tunnel (Placeholder_MPLS_1).

a. Diagnose the SD-WAN service:

The placeholder shortcut tunnel (Placeholder_MPLS_1) is preferred, and the regular shortcut tunnel (H1_T11_0 overlay) is out of SLA.

```
# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x4200 use-shortcut-sla use-shortcut
 Tie break: cfg
 Shortcut priority: 3
  Gen(87), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-compare-
order
  Member sub interface(4):
    2: seq_num(4), interface(H1_T11):
       1: H1_T11_0(1167)
    4: seq_num(13), interface(Placeholder_MPLS_1):
       1: Placeholder_MPLS_1_0(1168)
  Members(4):
    1: Seq_num(13 Placeholder_MPLS_1_0 overlay), alive, sla(0x1), gid(0), cfg_order(1),
local cost(0), selected
    2: Seq_num(4 H1_T11 overlay), alive, sla(0x1), gid(0), cfg_order(0), local cost(0),
selected
    3: Seq_num(4 H1_T11_0 overlay), alive, sla(0x0), gid(0), cfg_order(0), local cost(0),
selected
    4: Seq_num(13 Placeholder_MPLS_1 overlay), dead, sla(0x0), gid(0), cfg_order(1), local
cost(0)
  Src address(1):
        10.0.3.0-10.0.3.255
  Dst address(1):
        10.0.4.0-10.0.4.255
```

b. Sniff the packet to see the traffic switch to the placeholder shortcut tunnel (Placeholder_MPLS_1_0):

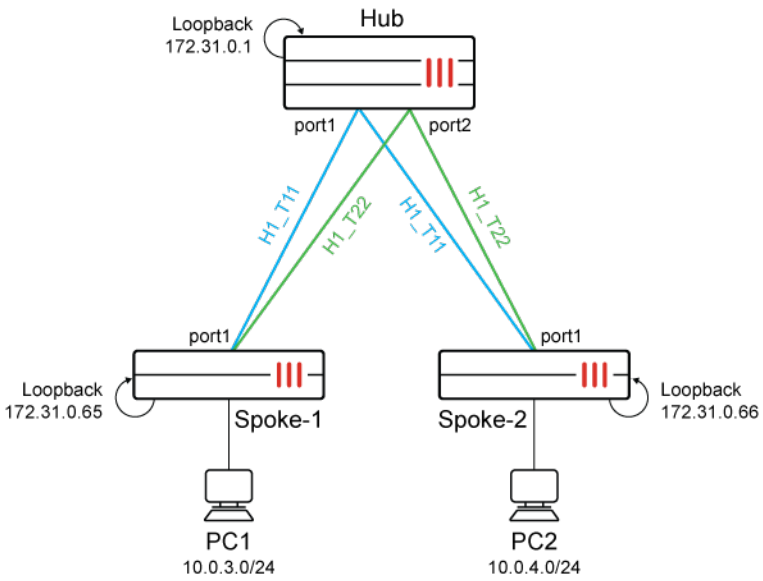
```
# diagnose sniffer packet any 'host 10.0.4.2' 4
interfaces=[any]
filters=[host 10.0.4.2]

17.356165 port4 in 10.0.3.2 -> 10.0.4.2: icmp: echo request
17.356261 Placeholder_MPLS_1_0 out 10.0.3.2 -> 10.0.4.2: icmp: echo request
17.356857 Placeholder_MPLS_1_0 in 10.0.4.2 -> 10.0.3.2: icmp: echo reply
17.356913 port4 out 10.0.4.2 -> 10.0.3.2: icmp: echo reply
18.361038 port4 in 10.0.3.2 -> 10.0.4.2: icmp: echo request
18.361247 Placeholder_MPLS_1_0 out 10.0.3.2 -> 10.0.4.2: icmp: echo request
```

Example shortcut over distinct underlay path using ADVPN 2.0

When ADVPN 2.0 was introduced, path management between spokes was enhanced to enable each spoke to learn the links from other spokes and use the information to more efficiently build shortcuts. When applied to a service rule that utilizes SLA mode with load-balancing, SD-WAN attempts to build all possible shortcuts using priority members in the service rule and overlays learned from the peer spoke. All created shortcuts use unique underlay paths. Once shortcuts are created on all distinct underlay paths, user traffic is load-balanced between in-SLA shortcuts, rather than between in-SLA shortcuts and parents.

Example

In this example with one hub and two spokes, each spoke has one underlay link, and the hub has two underlay links. For redundancy, each spoke builds two overlays to hub:



In a service rule named Corporate-H1 on Spoke-1, both H1_T11 and H1_T22 are named as priority members for this load-balancing rule. Likewise, Spoke-2 has two overlay paths to the hub. Instead of creating 2x2 shortcut paths based on the overlays, this enhancement identifies the single unique underlay path over port1 and creates a shortcut on this path.

This section reviews the following settings to demonstrate how the ADVPN 2.0 enhancement works:

- Settings on spokes and hub relevant to the example on page 1373
- Routing and SD-WAN status on spokes on page 1378

Settings on spokes and hub relevant to the example

Settings on spoke-1 relevant to the example:

```
config vpn ipsec phase1-interface
    edit "H1_T11"
        set interface "port1"
        set ike-version 2
        set keylife 28800
        set peertype any
        set net-device enable
        set exchange-ip-addr4 172.31.0.65
        set proposal aes256gcm-prfsha384
        set add-route disable
        set dpd on-idle
        set idle-timeout enable
        set idle-timeoutinterval 5
        set auto-discovery-receiver enable
        set encapsulation vpn-id-ipip
```

```
        set network-overlay enable
        set network-id 11
        set transport udp
        set remote-gw 172.31.1.1
        set psksecret ENC
YLI/3/nBszUUeTcFXvYlc18L1y2gjuWXSyBIKXOpQxnRulw+XW8/4P8DD1mMLf5K5jxhk0z05F2mdgrHtiEdH6rzZxRd62Sq6n
xPyWT3zf6+KR4yHrej9nn9HjVPNuDPjb2Q0fqInBVyPf8SbcN2adxvaslKYEEIt3wKqz0ZJ4oM3qd5/EuNyCaeZ5mXMEBY9T91
cFlmMjY3dkVA
        set dpd-retryinterval 5
    next
    edit "H1_T22"
        set interface "port1"
        set ike-version 2
        set keylife 28800
        set peertype any
        set net-device enable
        set exchange-ip-addr4 172.31.0.65
        set proposal aes256gcm-prfsha384
        set add-route disable
        set dpd on-idle
        set idle-timeout enable
        set idle-timeoutinterval 5
        set auto-discovery-receiver enable
        set encapsulation vpn-id-ipip
        set network-overlay enable
        set network-id 12
        set transport udp
        set remote-gw 172.31.1.5
        set psksecret ENC
x86NW9dT0mEpkJZIKAA819lxkqzcnDngeWHS3hivsfAiQcFpaUm5Bvwo4zsmXeX0n5UWH5CHhn5yxcdw2vtAeuuIwXQH7lOTek
IBklCD6aHt8zResI1B3bVSq7+eabMQL3RNnw8PI7IBImXpO2xJs5dt4oyxomfkfA8tCOU7w5kiegUk7lgUU5BwKJmbKtmE1plD
IFlmMjY3dkVA
        set dpd-retryinterval 5
    next

config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
        edit "overlay"
            set advpn-select enable
            set advpn-health-check "HUB"
        next
        edit "underlay"
        next
    end
    config members
      ...
        edit 4
            set interface "H1_T11"
            set zone "overlay"
```

```
            set source 172.31.0.65
            set priority 10
            set transport-group 1
        next
        edit 5
            set interface "H1_T22"
            set zone "overlay"
            set source 172.31.0.65
            set priority 10
            set transport-group 1
        next
      ...
    end
    config health-check
        edit "HUB"
            set server "172.31.100.100"
            set embed-measured-health enable
            set sla-fail-log-period 10
            set sla-pass-log-period 10
            set members 4 5
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 100
                next
            end
        next
    end
    config service
        edit 1
            set name "Corporate-H1"
            set load-balance enable
            set mode sla
            set dst "CORP_LAN"
            set src "CORP_LAN"
            config sla
                edit "HUB"
                    set id 1
                next
            end
            set priority-members 4 5
        next
    end
end
```

Settings on spoke-2 relevant to the example:

```
config vpn ipsec phase1-interface
    edit "H1_T11"
        set interface "port1"
        set ike-version 2
        set keylife 28800
```

```
        set peertype any
        set net-device enable
        set exchange-ip-addr4 172.31.0.66
        set proposal aes256gcm-prfsha384
        set add-route disable
        set dpd on-idle
        set idle-timeout enable
        set idle-timeoutinterval 5
        set auto-discovery-receiver enable
        set encapsulation vpn-id-ipip
        set network-overlay enable
        set network-id 11
        set transport udp
        set remote-gw 172.31.1.1
        set psksecret ENC
nVKierKcpKKdMZjidGD7OsHfdasVAQyBkMKzDtXq2Go76J2ASJckRj/NOt8BwepXUjJEVu7FoYsxeilBCeoYrR/6UU3KxgF/0m
ZmqMOYL/nQVznStRCkICt6dCgw4I1+ks6AK8eROgpR/12xAPCXKImi/G6Y8vCiVCKCYNmbLFYjq26E3g3H9ZLm0WuXyBWyLahh
PllmMjY3dkVA
        set dpd-retryinterval 5
    next
    edit "H1_T22"
        set interface "port1"
        set ike-version 2
        set keylife 28800
        set peertype any
        set net-device enable
        set exchange-ip-addr4 172.31.0.66
        set proposal aes256gcm-prfsha384
        set add-route disable
        set dpd on-idle
        set idle-timeout enable
        set idle-timeoutinterval 5
        set auto-discovery-receiver enable
        set encapsulation vpn-id-ipip
        set network-overlay enable
        set network-id 12
        set transport udp
        set remote-gw 172.31.1.5
        set psksecret ENC
nurNrgSmyldnHa9ngJb66s+cXQlWq43We2qVnJ8rT1Dkpga8ITA6bDC4qnOi/8guo3RGEvG0jfasRBHvuQtXLWf2Fzid3QsNP9
UPel+PO2/vJAHhvjPEPgEeJH33vAiFZ9bzr3FqKM0UhbDICtuNrAcAve3v9mnjr19XePrN85yHZt4uggeh3xXNv2hVjEFG7v6n
2VlmMjY3dkVA
        set dpd-retryinterval 5
    next

config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
        edit "overlay"
            set advpn-select enable
```

```
            set advpn-health-check "HUB"
        next
        edit "underlay"
        next
    end
    config members
    ...
        edit 4
            set interface "H1_T11"
            set zone "overlay"
            set source 172.31.0.66
            set priority 10
            set transport-group 1
        next
        edit 5
            set interface "H1_T22"
            set zone "overlay"
            set source 172.31.0.66
            set priority 10
            set transport-group 1
        next
     ...
    end
    config health-check
        edit "HUB"
            set server "172.31.100.100"
            set embed-measured-health enable
            set sla-fail-log-period 10
            set sla-pass-log-period 10
            set members 4 5
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 100
                next
            end
        next
    end
end
```

Settings on the hub relevant to the example:

```
config vpn ipsec phase1-interface
    edit "EDGE_T1"
        set type dynamic
        set interface "port1"
        set ike-version 2
        set peertype any
        set net-device disable
        set exchange-ip-addr4 172.31.0.1
        set proposal aes256gcm-prfsha384
        set add-route disable
```

```
        set dpd on-idle
        set auto-discovery-sender enable
        set encapsulation vpn-id-ipip
        set network-overlay enable
        set network-id 11
        set transport udp
        set psksecret ENC
nveTNKiRH+Vw2v0kFCz6VOdGKhrVgPs67H+k1HA322+ICHSW/mPKqMWSduvctoU1Ag1UvTTNUpIYQJ3V8U3U2+O1YNYrSa76Ut
8EnYEJfXTgRtq8jXbUMJzRoMl29Z4fLjBddLqT1OJCVGMM+5YkbWzICCz9o4Y1VNOAy+XtNPTsRcGaJzezotNlh87fOQEsjdXI
/FlmMjY3dkVA
        set dpd-retryinterval 5
    next
    edit "EDGE_T2"
        set type dynamic
        set interface "port2"
        set ike-version 2
        set peertype any
        set net-device disable
        set exchange-ip-addr4 172.31.0.1
        set proposal aes256gcm-prfsha384
        set add-route disable
        set dpd on-idle
        set auto-discovery-sender enable
        set encapsulation vpn-id-ipip
        set network-overlay enable
        set network-id 12
        set transport udp
        set psksecret ENC
UHZf3N8MSmzKQRsjmawCen60WiB3It4OByiA2cbTYtLv1785gzmlXRgLvn1mzCM96sW8kC7MCKpCNo/LA2z8MnvV5BDzrqrKyY
ZTC++NKg80kW9xn1Rf+P2gVs1yoghHF9dTK8lSaU4XTNN0/jxKilFz8vqLFi3nEAXNRIk1XMPTdMQcdUUVHXoL6pU1QIzR1Rf/
21lmMjY3dkVA
        set dpd-retryinterval 5
    next
```

Routing and SD-WAN status on spokes

To view the routing and SD-WAN status on spokes:

1. Check the routing and SD-WAN status on the spokes before the shortcut is initiated.
   a. On spoke-1, the service rule has two equal, viable paths over the parent overlay tunnels to reach the 10.0.4.0/24 network on Spoke-2:

```
# diagnose sys sdwan  health-check
Health Check(HUB):
Seq(4 H1_T11): state(alive), packet-loss(0.000%), latency(0.288), jitter(0.034), mos
(4.404), custom_profile(0.000), bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi
(1999997), sla_map=0x1
Seq(5 H1_T22): state(alive), packet-loss(0.000%), latency(0.215), jitter(0.017), mos
(4.404), custom_profile(0.000), bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi
(1999997), sla_map=0x1
```

```
Branch1_A_FGT (root) (Interim)# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x24200 use-shortcut-sla use-shortcut
 Tie break: cfg
 Shortcut priority: 3
  Gen(5), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla hash-
mode=round-robin)
  Members(2):
    1: Seq_num(4 H1_T11 overlay), alive, sla(0x1), gid(2), num of pass(1), selected
    2: Seq_num(5 H1_T22 overlay), alive, sla(0x1), gid(2), num of pass(1), selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        10.0.0.0-10.255.255.255

Branch1_A_FGT (root) (Interim)# get router info routing-table details 10.0.4.0/24

Routing table for VRF=0
Routing entry for 10.0.4.0/24
  Known via "bgp", distance 200, metric 0, best
  Last update 00:18:24 ago
  * vrf 0 172.31.0.66, tag 1 priority 1 (recursive via H1_T11 tunnel 172.31.1.1), tag-
match
                                                          (recursive via H1_T22 tunnel
172.31.1.5), tag-match
```

b. On spoke-2:

```
# diagnose sys sdwan  health-check
Health Check(HUB):
Seq(4 H1_T11): state(alive), packet-loss(0.000%), latency(0.256), jitter(0.019), mos
(4.404), custom_profile(0.000), bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi
(1999996), sla_map=0x1
Seq(5 H1_T22): state(alive), packet-loss(0.000%), latency(0.208), jitter(0.007), mos
(4.404), custom_profile(0.000), bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi
(1999996), sla_map=0x1

Branch2_FGT (root) (Interim)# get router info routing-table details 10.0.3.0/24

Routing table for VRF=0
Routing entry for 10.0.3.0/24
  Known via "bgp", distance 200, metric 0, best
  Last update 00:20:22 ago
  * vrf 0 172.31.0.65, tag 1 priority 1 (recursive via H1_T11 tunnel 172.31.1.1), tag-
match
                                                          (recursive via H1_T22 tunnel
172.31.1.5), tag-match
```

2. Initiate a ping from PC1 to PC2 to match SLA load-balance service 1 on Spoke-1. This initiates the shortcut path negotiations.

Instead of triggering all four possible shortcuts on all four overlay paths:

- local H1_T11 -- remote H1_T11
- local H1_T11 -- remote H1_T22
- local H1_T22 -- remote H1_T11
- local H1_T22 -- remote H1_T22

Only one shortcut is triggered on the distinct underlay path between two spokes:

- local port1 -- remote port1

On spoke-1:

A bigger gid value (33554434) is assigned to the shortcut, resulting in this path being the preferred for service rule 1.

```
Branch1_A_FGT (root) (Interim)# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x24200 use-shortcut-sla use-shortcut
 Tie break: cfg
 Shortcut priority: 3
  Gen(7), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla hash-mode=round-
robin)
  Member sub interface(3):
    3: seq_num(4), interface(H1_T11):
       1: H1_T11_0(56)
  Members(3):
    1: Seq_num(4 H1_T11_0 overlay), alive, sla(0x1), gid(33554434), num of pass(1), selected
    2: Seq_num(5 H1_T22 overlay), alive, sla(0x1), gid(2), num of pass(1), selected
    3: Seq_num(4 H1_T11 overlay), alive, sla(0x1), gid(2), num of pass(1), selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        10.0.0.0-10.255.255.255
```

This diagnostic command shows that only one path is calculated to avoid multiple duplicate paths.

```
Branch1_A_FGT (root) (Interim)# diagnose sys sdwan advpn-session
Session head(Branch2_FGT-0-overlay:1)
(1) Service ID(1), last access(233927), remote health check info(2)
Selected path: local(H1_T11, port1) gw: 172.31.3.1  remote IP: 172.31.3.101(172.31.0.66)
remote gw-name: H1_T11, status: up
Remote information:
1: latency: 0.261667 jitter: 0.023933 pktloss: 0.000000 mos: 4.404241 sla: 0x1 cost: 0 remote
gw: H1_T11 transport_group: 1 bandwidth up: 999999 down: 999997 bidirection: 1999996
ipv4: 172.31.3.101(172.31.0.66) ipv6 ::(42:cfc:ec43:9cb8:90a3:82e0:7f00:0)
2: latency: 0.210367 jitter: 0.012100 pktloss: 0.000000 mos: 4.404278 sla: 0x1 cost: 0 remote
gw: H1_T22 transport_group: 1 bandwidth up: 999999 down: 999997 bidirection: 1999996
ipv4: 172.31.3.101(172.31.0.66) ipv6 ::(7f00:0:78a5:82e0:7f00:0:78a5:82e0)
```

Finally, health check on the new shortcut indicates the health is good.

```
Branch1_A_FGT (root) (Interim)# diagnose sys sdwan  health-check
Health Check(HUB):
Seq(4 H1_T11): state(alive), packet-loss(0.000%), latency(0.236), jitter(0.012), mos(4.404),
custom_profile(0.000), bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996), sla_
map=0x1
Seq(4 H1_T11_0): state(alive), packet-loss(0.000%), latency(0.268), jitter(0.010), mos(4.404),
```

```
custom_profile(0.000), bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997), sla_
map=0x1
Seq(5 H1_T22): state(alive), packet-loss(0.000%), latency(0.206), jitter(0.007), mos(4.404),
custom_profile(0.000), bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996), sla_
map=0x1
```

The following sniffer trace shows the switch from parent to shortcut.

```
Branch1_A_FGT (root) (Interim)# diagnose sniffer packet any 'host 10.0.4.2' 4
interfaces=[any]
filters=[host 10.0.4.2]
2.436550 port4 in 10.0.3.2 -> 10.0.4.2: icmp: echo request
2.436993 H1_T11 out 10.0.3.2 -> 10.0.4.2: icmp: echo request
2.438112 H1_T11 in 10.0.4.2 -> 10.0.3.2: icmp: echo reply
2.438276 port4 out 10.0.4.2 -> 10.0.3.2: icmp: echo reply
3.438446 port4 in 10.0.3.2 -> 10.0.4.2: icmp: echo request
3.438529 H1_T11_0 out 10.0.3.2 -> 10.0.4.2: icmp: echo request
3.439346 H1_T11_0 in 10.0.4.2 -> 10.0.3.2: icmp: echo reply
3.439393 port4 out 10.0.4.2 -> 10.0.3.2: icmp: echo reply
4.439533 port4 in 10.0.3.2 -> 10.0.4.2: icmp: echo request
4.439745 H1_T11_0 out 10.0.3.2 -> 10.0.4.2: icmp: echo request
```

The following debug-flow trace shows the change from matching the H1_T11 tunnel initially to matching the H1_T11_0 shortcut tunnel once the shortcut was established.

```
Branch1_A_FGT (root) (Interim)#

id=65308 trace_id=1 func=print_pkt_detail line=6143 msg="vd-root:0 received a packet(proto=1,
10.0.3.2:7609->10.0.4.2:2048) tun_id=0.0.0.0 from port4. type=8, code=0, id=7609, seq=1."
id=65308 trace_id=1 func=init_ip_session_common line=6354 msg="allocate a new session-
00000336"
id=65308 trace_id=1 func=rpdb_srv_match_input line=1182 msg="Match policy routing
id=2130706433: to 10.0.4.2 via ifindex-44"
id=65308 trace_id=1 func=vf_ip_route_input_common line=2615 msg="find a route: flag=04000000
gw-172.31.1.1 via H1_T11"
id=65308 trace_id=1 func=__iprope_tree_check line=528 msg="gnum-100004, use int hash, slot=28,
len=4"
id=65308 trace_id=1 func=fw_forward_handler line=1003 msg="Allowed by Policy-1:"
id=65308 trace_id=1 func=ip_session_confirm_final line=3209 msg="npu_state=0x1041000, hook=4"
id=65308 trace_id=1 func=ids_receive line=464 msg="send to ips"
id=65308 trace_id=1 func=ipsecdev_hard_start_xmit line=669 msg="enter IPSec interface H1_T11,
tun_id=0.0.0.0"
id=65308 trace_id=1 func=_do_ipsecdev_hard_start_xmit line=229 msg="output to IPSec tunnel H1_
T11, tun_id=172.31.1.1, vrf 0"
id=65308 trace_id=1 func=esp_output4 line=870 msg="IPsec encrypt/auth"
id=65308 trace_id=1 func=ipsec_output_finish line=667 msg="send to 172.31.3.2 via intf-port1"
id=65308 trace_id=2 func=print_pkt_detail line=6143 msg="vd-root:0 received a packet(proto=1,
10.0.3.2:7609->10.0.4.2:2048) tun_id=0.0.0.0 from port4. type=8, code=0, id=7609, seq=2."
id=65308 trace_id=2 func=resolve_ip_tuple_fast line=6251 msg="Find an existing session, id-
00000336, original direction"
id=65308 trace_id=2 func=rpdb_srv_match_input line=1182 msg="Match policy routing
id=2130706433: to 10.0.4.2 via ifindex-61"
```

```
id=65308 trace_id=2 func=vf_ip_route_input_common line=2615 msg="find a route: flag=04000000
gw-172.31.3.101 via H1_T11_0"
id=65308 trace_id=2 func=__iprope_tree_check line=528 msg="gnum-100004, use int hash, slot=28,
len=4"
id=65308 trace_id=2 func=ip_session_core_in line=6853 msg="dir-0, tun_id=172.31.1.1"
id=65308 trace_id=2 func=ipsecdev_hard_start_xmit line=669 msg="enter IPSec interface H1_T11_
0, tun_id=172.31.1.1"
id=65308 trace_id=2 func=_do_ipsecdev_hard_start_xmit line=229 msg="output to IPSec tunnel H1_
T11_0, tun_id=172.31.3.101, vrf 0"
id=65308 trace_id=2 func=esp_output4 line=870 msg="IPsec encrypt/auth"
id=65308 trace_id=2 func=ipsec_output_finish line=667 msg="send to 172.31.3.2 via intf-port1"
```

Self-healing and application protection

In order to remediate network connectivity problems, SD-WAN proactively measures network conditions such as latency, jitter, and packet loss on WAN links and use these metrics to adjust traffic according to the application service-level agreements (SLAs). This is often referred to as SLA-based path steering and is discussed in SD-WAN rules on page 1105.

In the same way that applications are evaluated against their specific SLAs, SD-WAN overlay members can be evaluated to determine whether they are healthy or not using SLAs of their own. This enables the SD-WAN to leverage BGP to signal to other nodes in the SD-WAN network that one or more of their members is unhealthy, or out-of-SLA. The nodes can then use this SLA status to prioritize IKE and BGP routes, favoring those that meet performance thresholds (within SLA). This real-time health signaling across nodes improves link visibility, which is especially critical in hub-and-spoke topologies with many connections. By offloading the SLA reporting task to spokes, the processing load on hubs is reduced, allowing for increased efficiency and scaling.

In some situations, there may be no healthy (or in-SLA) links for SD-WAN to direct traffic over. In this case, FortiGate has advanced techniques, such as forward error correction (FEC) and packet duplication, to boost resilience when multiple paths fail. This preserves the user experience and provides business continuity when there would be an outage otherwise.

This section contains the following topics:

- Embedded SD-WAN SLA priorities in ICMP probes on page 1382
- Map SD-WAN member priorities to BGP MED attribute when spoke advertises routes using iBGP to hub on page 1394
- Adaptive Forward Error Correction on page 1402
- Duplicate packets based on SD-WAN rules on page 1421
- Duplicate packets on other zone members on page 1423

Embedded SD-WAN SLA priorities in ICMP probes

In SD-WAN hub-and-spoke topologies, each spoke can embed an SLA priority (`priority-in-sla` and `priority-out-sla`) in ICMP probes and send them to the hub. The hub can use the received SLA priorities from each spoke to manage route priority for hub-to-spoke traffic. By offloading the evaluation of the spoke and hub routes to each spoke, the processing is spread amongst many FortiGates, instead of all on the hub.

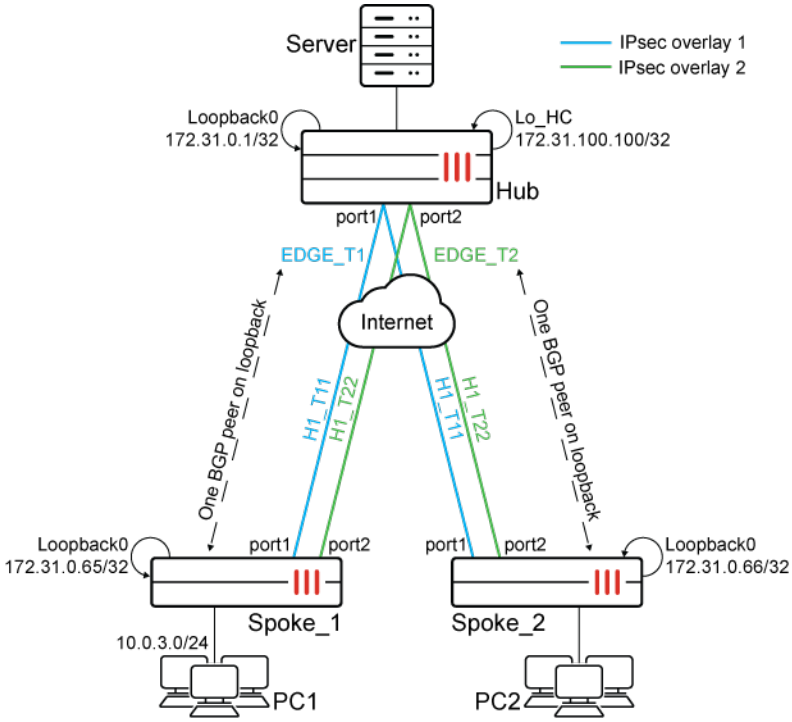
The `priority-in-sla` and `priority-out-sla` settings can be customized for each member of a spoke, allowing different members to have different values when in or out of SLA. This is particularly useful when a spoke has underlays of different types and uses. For example, a spoke that has a broadband connection and an LTE connection will typically prefer to send traffic across the broadband underlay when both are in SLA.

Spoke-initiated speed tests can also use the embedded information. When configured, a spoke will continually embed an out-of SLA priority into ICMP probes on the overlay during a speed test. The hub will use the received out-of SLA priority information to manage route priorities and detour hub-to-spoke user traffic to other tunnels. This is to ensure that the speed test is not affected by any traffic for the duration of the test.

Examples

The SD-WAN topology with ADVPN and BGP neighbor on loopback is used for the following two examples:

- Example 1: Embedded priority configuration on page 1383
- Example 2: Speed test rerouting based on embedded SLA priorities on page 1390



Example 1: Embedded priority configuration

In this example, a health check named `hub` is configured to embed its measured health status. When the health check is in SLA, the spoke sends an in-SLA priority of 10. When it is out of SLA, the spoke sends an out-of-SLA priority of 100. As there are two link-cost-factor thresholds (latency: 100, packet-loss: 50), the spoke sends in-SLA only if both factors are satisfied.

To configure the example:

1. Configure SD-WAN and BGP on Spoke_1:
   a. Configure SD-WAN:

```
config system sdwan
    set status enable
    config zone
        edit "overlay"
        next
    end
    config members
        edit 4
            set interface "H1_T11"
            set zone "overlay"
            set source 172.31.0.65
            set priority 10
            set priority-in-sla 10
            set priority-out-sla 20
        next
        edit 5
            set interface "H1_T22"
            set zone "overlay"
            set source 172.31.0.65
            set priority 10
            set priority-in-sla 15
            set priority-out-sla 25
        next
    end
    config health-check
        edit "HUB"
            set embed-measured-health enable
            set sla-id-redistribute 1
            set members 4 5
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 50
                next
            end
        next
    end
end
```

   b. Configure BGP:

```
config router bgp
    set as 65001
    set router-id 172.31.0.65
    ...
    config neighbor
        edit "172.31.0.1"
```

```
            ...
            set remote-as 65001
            set update-source "Loopback0"
        next
    end
    config network
        edit 1
            set prefix 10.0.3.0 255.255.255.0
        next
    end
    ...
end
```

c. View the health-check settings:

```
# diagnose sys sdwan health-check
Health Check(HUB):
Seq(4 H1_T11): state(alive), packet-loss(0.000%) latency(0.230), jitter(0.040), mos
(4.404), bandwidth-up(999998), bandwidth-dw(999996), bandwidth-bi(1999994) sla_map=0x1
Seq(5 H1_T22): state(alive), packet-loss(0.000%) latency(0.188), jitter(0.007), mos
(4.404), bandwidth-up(999998), bandwidth-dw(999996), bandwidth-bi(1999994) sla_map=0x1
```

2. Configure SD-WAN and BGP on Spoke_2:

a. Configure SD-WAN:

```
config system sdwan
    set status enable
    config zone
        edit "overlay"
        next
    end
    config members
        edit 4
            set interface "H1_T11"
            set zone "overlay"
            set source 172.31.0.65
            set priority 10
            set priority-in-sla 30
            set priority-out-sla 40
        next
        edit 5
            set interface "H1_T22"
            set zone "overlay"
            set source 172.31.0.65
            set priority 10
            set priority-in-sla 35
            set priority-out-sla 45
        next
    end
    config health-check
        edit "HUB"
            set embed-measured-health enable
```

```
            set sla-id-redistribute 1
            set members 4 5
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 70
                next
            end
        next
    end
end
```

b. Configure BGP:

```
config router bgp
    set as 65001
    set router-id 172.31.0.65
    ...
    config neighbor
        edit "172.31.0.1"
            ...
            set remote-as 65001
            set update-source "Loopback0"
        next
    end
    config network
        edit 1
            set prefix 10.0.4.0 255.255.255.0
        next
    end
    ...
end
```

c. View the health-check settings:

```
# diagnose sys sdwan health-check
Health Check(HUB):
Seq(4 H1_T11): state(alive), packet-loss(0.000%) latency(0.230), jitter(0.040), mos
(4.404), bandwidth-up(999998), bandwidth-dw(999996), bandwidth-bi(1999994) sla_map=0x1
Seq(5 H1_T22): state(alive), packet-loss(0.000%) latency(0.188), jitter(0.007), mos
(4.404), bandwidth-up(999998), bandwidth-dw(999996), bandwidth-bi(1999994) sla_map=0x1
```

3. Configure SD-WAN and BGP on the Hub:

a. Configure SD-WAN:

```
config system sdwan
    set status enable
    config zone
        edit "overlay"
        next
    end
    config members
```

```
        edit 1
            set interface "EDGE_T1"
            set zone "overlay"
        next
        edit 2
            set interface "EDGE_T2"
            set zone "overlay"
        next
    end
    config health-check
        edit "Remote_HC"
            set detect-mode remote
            set sla-id-redistribute 1
            set members 1 2
            config sla
                edit 1
                    set link-cost-factor remote
                next
            end
        next
    end
end
```

In mixed deployments where some spokes may not send the SLA priority and only send SLA status (such as SLA-in or SLA-out), the hub can be configured to set default in-SLA and out-SLA values for routes. The following bold lines could added to the hub configuration:

```
config sla
    edit 1
        set link-cost-factor remote
        set link-cost-factor-default latency
        set priority-in-sla 11
        set priority-out-sla 101
    next
end
```

b. Configure BGP:

```
config router bgp
    set as 65001
    set router-id 172.31.0.1
    set recursive-inherit-priority enable
    ...
    config neighbor-group
        edit "EDGE"
            set passive enable
            set remote-as 65001
            set update-source "Loopback0"
            set route-reflector-client enable
        next
    end
```

```
    config neighbor-range
        edit 1
            set prefix 172.31.0.64 255.255.255.192
            set neighbor-group "EDGE"
        next
    end
    ...
end
```

When BGP passive mode is disabled on an ADVPN hub FortiGate with embedded SLA, SD-WAN health-check events can trigger repeated BGP session resets resulting in session flapping.

c. View the health-check settings:

The following example shows:

- `rmt_ver=2` indicates that SLA information, SLA status, and overlay priority have been received.
- `rmt_sla=out` indicates that received SLA status is out of SLA.
- `rmt_sla=in` indicates that received SLA status is in SLA.
- `rmt_prio` indicates the received overlay priority value.
- `EDGE_T1_0` is to `H1_T11` on Spoke_1, and `EDGE_T1_1` is to `H1_T11` on Spoke_2.
- `EDGE_T2_1` is to `H1_T22` on Spoke_1, and `EDGE_T1_0` is to `H1_T22` on Spoke_2.

```
# diagnose sys sdwan health-check remote

Remote Health Check: Remote_HC(1)
  Passive remote statistics of EDGE_T1(46):
    EDGE_T1_0(10.0.0.20): timestamp=06-12 14:04:29, src=172.31.0.65, latency=0.247,
jitter=0.028, pktloss=0.000%, mos=4.404, SLA id=1(pass), rmt_ver=2, rmt_sla=in, rmt_
prio=10
    EDGE_T1_1(172.31.0.66): timestamp=06-12 14:04:30, src=172.31.0.66, latency=0.246,
jitter=0.031, pktloss=0.000%, mos=4.404, SLA id=1(pass), rmt_ver=2, rmt_sla=in, rmt_
prio=30

Remote Health Check: Remote_HC(2)
  Passive remote statistics of EDGE_T2(47):
    EDGE_T2_0(10.0.0.15): timestamp=06-12 14:04:30, src=172.31.0.66, latency=0.191,
jitter=0.008, pktloss=0.000%, mos=4.404, SLA id=1(pass), rmt_ver=2, rmt_sla=in, rmt_
prio=35
    EDGE_T2_1(172.31.0.65): timestamp=06-12 14:04:29, src=172.31.0.65, latency=0.201,
jitter=0.008, pktloss=0.000%, mos=4.404, SLA id=1(pass), rmt_ver=2, rmt_sla=in, rmt_
prio=15
```

4. After the spokes' overlay priorities are embedded in ICMP probes and transported to the hub, view the routing tables on the hub.

The hub sets overlay priorities on IKE routes over EDGE_T1 and EDGE_T2. Meanwhile, recursively resolved BGP routes inherit the priorities from those IKE routes.

a. On the hub, get the static routing table:

```
# get router info routing-table static
Routing table for VRF=0
S       172.31.0.65/32 [15/0] via EDGE_T1 tunnel 10.0.0.20, [10/0]
                       [15/0] via EDGE_T2 tunnel 172.31.0.65, [15/0]
S       172.31.0.66/32 [15/0] via EDGE_T1 tunnel 172.31.0.66, [30/0]
                       [15/0] via EDGE_T2 tunnel 10.0.0.15, [35/0]
```

b. On the hub, get the BGP routing table:

```
# get router info routing-table bgp
Routing table for VRF=0
B       10.0.3.0/24 [200/0] via 172.31.0.65 (recursive via EDGE_T1 tunnel 10.0.0.20 [10]),
01:15:46
                                            (recursive via EDGE_T2 tunnel 172.31.0.65
[15]), 01:15:46, [1/0]
B       10.0.4.0/24 [200/0] via 172.31.0.66 (recursive via EDGE_T1 tunnel 172.31.0.66
[30]), 01:13:46
                                            (recursive via EDGE_T2 tunnel 10.0.0.15 [35]),
01:13:46, [1/0]
```

5. Change the latency on Spoke_1 and Spoke_2, and view the results.
   a. On Spoke_1, increase H1_T11's latency to 60 to make it out of SLA.
   b. On Spoke_2, increase H1_T11's latency to 80 to make it out of SLA.
   c. On Spoke_1, run a health check:

```
# diagnose sys sdwan  health-check
Health Check(HUB):
Seq(4 H1_T11): state(alive), packet-loss(0.000%) latency(60.247), jitter(0.036), mos
(4.373), bandwidth-up(999998), bandwidth-dw(999997), bandwidth-bi(1999995) sla_map=0x0
Seq(5 H1_T22): state(alive), packet-loss(0.000%) latency(0.218), jitter(0.016), mos
(4.404), bandwidth-up(999998), bandwidth-dw(999998), bandwidth-bi(1999996) sla_map=0x
```

   d. On Spoke_2, run a health check:

```
# diagnose sys sdwan  health-check
Health Check(HUB):
Seq(4 H1_T11): state(alive), packet-loss(0.000%) latency(80.217), jitter(0.022), mos
(4.361), bandwidth-up(999998), bandwidth-dw(999998), bandwidth-bi(1999996) sla_map=0x0
Seq(5 H1_T22): state(alive), packet-loss(0.000%) latency(0.202), jitter(0.016), mos
(4.404), bandwidth-up(999998), bandwidth-dw(999997), bandwidth-bi(1999995) sla_map=0x1
```

6. After the hub receives the updated overlay priorities, run a health check on the hub and view the routing tables.

   The hub has updated the route priorities.

   a. Run a health check:

```
# diagnose sys sdwan health-check remote

Remote Health Check: Remote_HC(1)
  Passive remote statistics of EDGE_T1(46):
```

```
  EDGE_T1_0(10.0.0.20): timestamp=06-12 14:19:26, src=172.31.0.65, latency=60.249,
jitter=0.031, pktloss=0.000%, mos=4.373, SLA id=1(remote), rmt_ver=2, rmt_sla=out, rmt_
prio=20
EDGE_T1_1(172.31.0.66): timestamp=06-12 14:19:26, src=172.31.0.66, latency=80.222,
jitter=0.021, pktloss=0.000%, mos=4.361, SLA id=1(remote), rmt_ver=2, rmt_sla=out, rmt_
prio=40

Remote Health Check: Remote_HC(2)
  Passive remote statistics of EDGE_T2(47):
  EDGE_T2_0(10.0.0.15): timestamp=06-12 14:19:26, src=172.31.0.66, latency=0.205,
jitter=0.011, pktloss=0.000%, mos=4.404, SLA id=1(pass), rmt_ver=2, rmt_sla=in, rmt_
prio=35
  EDGE_T2_1(172.31.0.65): timestamp=06-12 14:19:26, src=172.31.0.65, latency=0.215,
jitter=0.009, pktloss=0.000%, mos=4.404, SLA id=1(pass), rmt_ver=2, rmt_sla=in, rmt_
prio=15
```

b. View the static routing table:

For EDGE_T1, the priority changed from 10 to 20 and from 30 to 40 because it is out of SLA.

```
# get router info routing-table static
Routing table for VRF=0
S       172.31.0.65/32 [15/0] via EDGE_T2 tunnel 172.31.0.65, [15/0]
                              [15/0] via EDGE_T1 tunnel 10.0.0.20, [20/0]
S       172.31.0.66/32 [15/0] via EDGE_T2 tunnel 10.0.0.15, [35/0]
                              [15/0] via EDGE_T1 tunnel 172.31.0.66, [40/0]
```

c. View the BGP routing table:

- For 10.0.3.0/24, EDGE_T2 is preferred. Priority for EDGE_T1 changed from 10 to 20.
- For 10.0.4.0/24, EDGE_T2 is preferred. Priority for EDGE_T1 changed from 30 to 40.

```
# get router info routing-table  bgp
Routing table for VRF=0
B       10.0.3.0/24 [200/0] via 172.31.0.65 (recursive via EDGE_T2 tunnel 172.31.0.65
[15]), 00:07:28
                                          (recursive via EDGE_T1 tunnel 10.0.0.20 [20]),
00:07:28, [1/0]
B       10.0.4.0/24 [200/0] via 172.31.0.66 (recursive via EDGE_T2 tunnel 10.0.0.15 [35]),
00:07:28                                                  (recursive via EDGE_T1
tunnel 172.31.0.66 [40]), 00:07:28, [1/0]
```

Example 2: Speed test rerouting based on embedded SLA priorities

When spoke-initiated speed tests are enabled for this configuration, the out-of SLA priority is used by the hub to choose other routes during the speed test.

To configure speed tests:

1. On the hub, enable speed tests and allow them on the underlays and overlays.
   a. Enable speed tests:

```
config system global
    set speedtest-server enable
    set speedtestd-ctrl-port 6000
    set speedtestd-server-port 7000
end
```

   b. Allow speed tests on the underlays:

```
config system interface
    edit "port1"
       ...
        set allowaccess ping speed-test
      ...
    next
    edit "port2"
        ...
        set allowaccess ping speed-test
       ...
    next
end
```

   c. Allow speed tests on the overlay, and specify a shaping profile:
   Update the interface to use measured values reported by the speed test for in and out bandwidth.

```
config system interface
    edit "EDGE_T1"
        ...
        set allowaccess ping speed-test
        set type tunnel
        set egress-shaping-profile "profile_1"
        set outbandwidth-source measured
        set inbandwidth-source measured
        ...
        set interface "port1"
    next
    edit "EDGE_T2"
        ...
        set allowaccess ping speed-test
        set type tunnel
        set egress-shaping-profile "profile_1"
        set outbandwidth-source measured
        set inbandwidth-source measured
        ...
        set interface "port2"
    next
end
```

   d. View the shaping profile:

```
config firewall shaping-profile
    edit "profile_1"
        set default-class-id 2
        config shaping-entries
            edit 1
                set class-id 2
                set priority low
                set guaranteed-bandwidth-percentage 10
                set maximum-bandwidth-percentage 10
            next
            edit 2
                set class-id 3
                set priority medium
                set guaranteed-bandwidth-percentage 30
                set maximum-bandwidth-percentage 40
            next
            edit 3
                set class-id 4
                set guaranteed-bandwidth-percentage 20
                set maximum-bandwidth-percentage 50
            next
        end
    next
end
```

2. On Spoke_1, schedule speed tests:

```
config system speed-test-schedule
    edit "H1_T11"
        set mode TCP
        set schedules "speed-test"
        set dynamic-server enable
        set ctrl-port 6000
        set server-port 7000
    next
    edit "H1_T22"
        set mode UDP
        set schedules "speed-test"
        set dynamic-server enable
        set ctrl-port 6000
        set server-port 7000
    next
end
```

Before starting the speed test on Spoke_1, route priorities are based on the received overlay priorities on both H1_T11 and H1_T22

```
HUB (root) (Interim)# get router info routing-table bgp
Routing table for VRF=0
B       10.0.3.0/24 [200/0] via 172.31.0.65 (recursive via EDGE_T1 tunnel 10.0.0.20 [10]),
00:11:56
                                           (recursive via EDGE_T2 tunnel 172.31.0.65 [15]),
00:11:56, [1/0]
```

While the speed test is running on H1_T11 of Spoke_1, Spoke_1 will constantly embed out-of SLA overlay priority into probes on H1_T11 and send them to the hub. Then the Hub updates route priorities accordingly and detours hub-to-spoke traffic to H1_T22 to avoid the impact on speed test of H1_T11. EDGE_T2 is preferred, and the EDGE_T1 priority changed from 10 to 20.

```
HUB (root) (Interim)# get router info routing-table  bgp
Routing table for VRF=0Routing table for VRF=0
B       10.0.3.0/24 [200/0] via 172.31.0.65 (recursive via EDGE_T2 tunnel 172.31.0.65 [15]),
00:03:49
                                           (recursive via EDGE_T1 tunnel 10.0.0.20 [20]),
00:03:49, [1/0]
```

During the speed test on H1_T22 of Spoke_1, Spoke_1 constantly embeds out-of SLA overlay priority into probes on H1_T22 and sends them to the hub. Then the Hub updates route priorities accordingly and detours hub-to-spoke traffic to H1_T11 to avoid the impact on speed test of H1_T22. EDGE_T1 is preferred, and the EDGE_T2 priority changed from 15 to 25.

```
HUB (root) (Interim)# get router info routing-table  bgp
Routing table for VRF=0
B       10.0.3.0/24 [200/0] via 172.31.0.65 (recursive via EDGE_T1 tunnel 10.0.0.20 [10]),
00:04:06
                                                                (recursive via EDGE_T2 tunnel
172.31.0.65 [25]), 00:04:06, [1/0]
```

Once speed test completes, the results are applied on child tunnels as egress-shaping-profile on the hub.

```
HUB (root) (Interim)# diagnose vpn tunnel  list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=EDGE_T1_0 ver=2 serial=22 172.31.1.1:0->172.31.3.1:0 nexthop=172.31.1.2 tun_id=10.0.0.20 tun_
id6=::10.0.0.31 status=up dst_mtu=1500 weight=1
.....
  dec: spi=9932cb6a esp=aes-gcm key=36
0819373cc74d0eb2dae8ac559519b756010fb42d2453b1c046429f8aad4d7dcbb507b990
       ah=null key=0
  enc: spi=dea67361 esp=aes-gcm key=36
c2ac45b9c1144d6728b5a7a542d1a35c86d018af35f6f661888dc39f771dc4820e2a40ff
       ah=null key=0
  dec:pkts/bytes=0/0, enc:pkts/bytes=3789/517724
  npu_flag=03 npu_rgwy=172.31.3.1 npu_lgwy=172.31.1.1 npu_selid=26 dec_npuid=1 enc_npuid=1 npu_
isaidx=719 npu_osaidx=39
egress traffic control:
        bandwidth=711495(kbps) lock_hit=0 default_class=2 n_active_class=3
        class-id=2      allocated-bandwidth=71149(kbps)         guaranteed-bandwidth=71149(kbps)
                        max-bandwidth=71149(kbps)       current-bandwidth=1(kbps)
                        priority=low    forwarded_bytes=82K
                        dropped_packets=0       dropped_bytes=0
        class-id=3      allocated-bandwidth=284597(kbps)        guaranteed-bandwidth=213448(kbps)
                        max-bandwidth=284597(kbps)      current-bandwidth=0(kbps)
                        priority=medium         forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
        class-id=4      allocated-bandwidth=355747(kbps)        guaranteed-bandwidth=142298(kbps)
                        max-bandwidth=355747(kbps)      current-bandwidth=0(kbps)
```

```
                        priority=high   forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
-----------------------------------------------------
name=EDGE_T2_1 ver=2 serial=1c 172.31.1.5:0->172.31.3.5:0 nexthop=172.31.1.6 tun_id=172.31.0.65
tun_id6=::10.0.0.25 status=up dst_mtu=1500 weight=1
......
  dec: spi=9932cb69 esp=aes-gcm key=36
01842dbbe1fe98fc6503b491c0768a844d815074950589b48db8a9a00c7505e73a96dc84
       ah=null key=0
  enc: spi=dea67360 esp=aes-gcm key=36
c2e4cb1065325de7970d6f8edb3cac8829a33420ff79acedd6dcf4012e4614dd92d7b16b
       ah=null key=0
  dec:pkts/bytes=0/0, enc:pkts/bytes=3518/499252
  npu_flag=03 npu_rgwy=172.31.3.5 npu_lgwy=172.31.1.5 npu_selid=27 dec_npuid=1 enc_npuid=1 npu_
isaidx=718 npu_osaidx=40
egress traffic control:
        bandwidth=374876(kbps) lock_hit=0 default_class=2 n_active_class=3
        class-id=2      allocated-bandwidth=37487(kbps)         guaranteed-bandwidth=37487(kbps)
                        max-bandwidth=37487(kbps)       current-bandwidth=2(kbps)
                        priority=low    forwarded_bytes=78K
                        dropped_packets=0       dropped_bytes=0
        class-id=3      allocated-bandwidth=149950(kbps)        guaranteed-bandwidth=112462(kbps)
                        max-bandwidth=149950(kbps)      current-bandwidth=0(kbps)
                        priority=medium         forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
        class-id=4      allocated-bandwidth=187438(kbps)        guaranteed-bandwidth=74975(kbps)
                        max-bandwidth=187438(kbps)      current-bandwidth=0(kbps)
                        priority=high   forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
```

Map SD-WAN member priorities to BGP MED attribute when spoke advertises routes using iBGP to hub

When a spoke advertises routes using iBGP to a hub, SD-WAN member priorities are mapped into the BGP multiple exit discriminator (MED) attribute using the following CLI commands:

```
config system sdwan
    config neighbor
        edit <bgp-peer-IP>
            set member <num_1> ... <num_n>
            set route-metric {preferable | priority}
            set health-check <health-check-name>
        next
    end
end
```

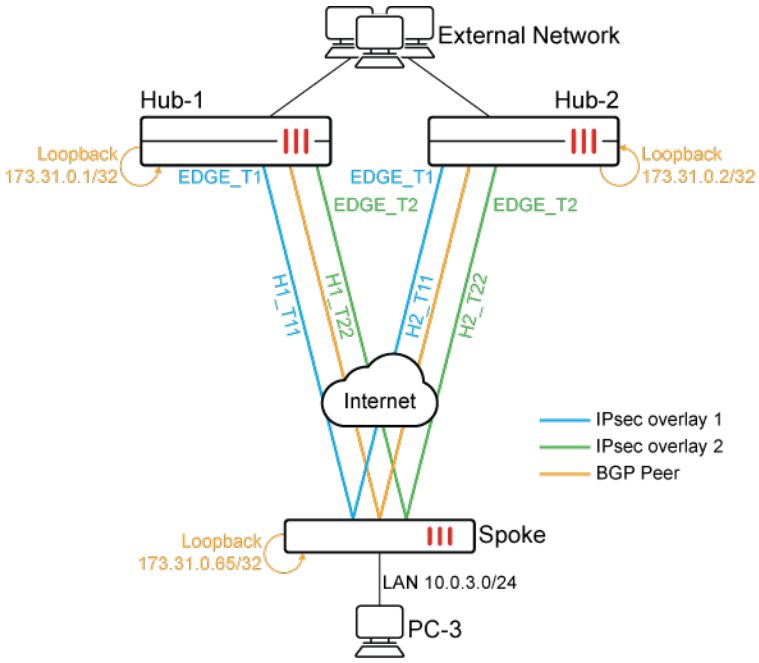
Option	Description
`preferable`	Select neighbor based on its HC to match BGP preferable/unpreferable route_map.
`priority`	Select neighbor based on its members' priority-in-sla/priority-out-sla value.

Routes to prefixes behind spokes are advertised by the SD-WAN hub to eBGP peers on an external network. The relative values of the BGP MED attribute for each hub are used to indicate to eBGP peers the more preferred paths, that is, the preferred hub used to route to spoke prefixes.

Example

This example includes one spoke and two hubs:

- The spoke has two overlays to Hub-1 and Hub-2 respectively.
- BGP neighbors between the spoke and hubs are over loopback IP addresses
- SD-WAN is configured on the spoke.



To configure and test the example:

1. Configure Hub-1:

   The Hub-2 configuration is identical except for the `router-id`, which reflects the different loopback IP address.

   a. Configure SD-WAN:

```
config system sdwan
    set status enable
    config zone
        edit "overlay"
```

```
        next
    end
    config members
        edit 1
            set interface "EDGE_T1"
            set zone "overlay"
        next
        edit 2
            set interface "EDGE_T2"
            set zone "overlay"
        next
    end
    config health-check
        edit "Remote_HC"
            set detect-mode remote
            set sla-id-redistribute 1
            set members 1 2
            config sla
                edit 1
                    set link-cost-factor remote
                    set priority-in-sla 100
                    set priority-out-sla 200
                next
            end
        next
    end
end
```

The `priority-[in|out]-sla` settings are used if a spoke does not send a priority. This is used in legacy configurations where spokes only send in or out for the SLA status.

b. Configure BGP:

```
config router bgp
    set as 65001
    set router-id 172.31.0.1        //172.31.0.2 for Hub-2
    set recursive-inherit-priority enable
    ...
    config neighbor-group
        edit "EDGE"
            set remote-as 65001
            set update-source "Loopback0"
            set route-reflector-client enable
        next
    end
    config neighbor-range
        edit 1
            set prefix 172.31.0.64 255.255.255.192
            set neighbor-group "EDGE"
        next
    end
```

```
    ...
end
```

2. Configure the spoke:
   a. Configure SD-WAN:

```
config system sdwan
    set status enable
    config zone
        edit "overlay"
        next
    end
    config members
        edit 4
            set interface "H1_T11"
            set zone "overlay"
            set source 172.31.0.65
            set priority-in-sla 50
            set priority-out-sla 100
        next
        edit 5
            set interface "H1_T22"
            set zone "overlay"
            set source 172.31.0.65
            set priority-in-sla 70
            set priority-out-sla 120
        next
        edit 7
            set interface "H2_T11"
            set zone "overlay"
            set source 172.31.0.65
            set priority-in-sla 60
            set priority-out-sla 110
        next
        edit 8
            set interface "H2_T22"
            set zone "overlay"
            set source 172.31.0.65
            set priority-in-sla 80
            set priority-out-sla 130
        next
    end
    config health-check
        edit "HUB"
            set server "172.31.100.100"
            set embed-measured-health enable
            set sla-id-redistribute 1
            set sla-fail-log-period 10
            set sla-pass-log-period 10
            set members 4 5 7 8
            config sla
                edit 1
```

```
                    set link-cost-factor latency
                    set latency-threshold 100
                next
            end
        next
    end
    config neighbor
        edit "172.31.0.1"
            set member 4 5
            set route-metric priority
            set health-check "HUB"
        next
        edit "172.31.0.2"
            set member 7 8
            set route-metric priority
            set health-check "HUB"
        next
    end
end
```

b. Configure BGP:

```
config router bgp
    set as 65001
    set router-id 172.31.0.65
    ...
    config neighbor
        edit "172.31.0.1"
            ......
            set remote-as 65001
            set update-source "Loopback0"
        next
        edit "172.31.0.2"
            ...
            set remote-as 65001
            set update-source "Loopback0"
        next

    end
    config network
        edit 1
            set prefix 10.0.3.0 255.255.255.0
        next
    end
    ...
end
```

The routes with MEDs are advertised to a router on the external network that establishes a BGP neighbor relationship with Hub-1 and Hub-2. When sending traffic destined for 10.0.3.0/24, the router on the external network will prefer to send traffic to the hub with the lower MED.

3. All overlays are in SLA:

```
Spoke # diagnose sys sdwan health-check
Health Check(HUB):
Seq(4 H1_T11): state(alive), packet-loss(0.000%), latency(0.225), jitter(0.035), mos(4.404),
bandwidth-up(999998), bandwidth-dw(999997), bandwidth-bi(1999995), sla_map=0x1
Seq(5 H1_T22): state(alive), packet-loss(0.000%), latency(0.203), jitter(0.016), mos(4.404),
bandwidth-up(999998), bandwidth-dw(999998), bandwidth-bi(1999996), sla_map=0x1
Seq(7 H2_T11): state(alive), packet-loss(0.000%), latency(0.260), jitter(0.035), mos(4.404),
bandwidth-up(999997), bandwidth-dw(999997), bandwidth-bi(1999994), sla_map=0x1
Seq(8 H2_T22): state(alive), packet-loss(0.000%), latency(0.201), jitter(0.018), mos(4.404),
bandwidth-up(999998), bandwidth-dw(999998), bandwidth-bi(1999996), sla_map=0x1
```

When sending traffic destined for 10.0.3.0/24, the router on the external network will prefer to send traffic to Hub-1 with lower MED 50 over Hub-2 with higher MED 60.

a. Check Hub-1 metric from the external network router:

```
External_Network_Router # get router info bgp neighbors 172.31.0.1 advertised-routes
VRF 0 BGP table version is 31, local router ID is 172.31.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network             Next Hop         Metric     LocPrf Weight RouteTag Path
*>i10.0.3.0/24      172.31.0.65     50            100  32768        0 i <-/->

Total number of prefixes 1
```

b. Check Hub-2 metric from the external network router:

```
External_Network_Router # get router info bgp neighbors 172.31.0.2 advertised-routes
VRF 0 BGP table version is 31, local router ID is 172.31.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network             Next Hop        Metric     LocPrf Weight RouteTag Path
*>i10.0.3.0/24      172.31.0.65     60            100  32768        0 i <-/->

Total number of prefixes 1
```

4. Verify that the metric updates when a link becomes out of SLA.

When sending traffic destined for 10.0.3.0/24, the router on the external network will prefer to send traffic to Hub-2 with lower MED 60 over Hub-1 with higher MED 70.

```
# diagnose sys sdwan health-check
Health Check(HUB):
Seq(4 H1_T11): state(alive), packet-loss(0.000%), latency(120.225), jitter(0.037), mos(4.338),
bandwidth-up(999997), bandwidth-dw(999996), bandwidth-bi(1999993), sla_map=0x0
Seq(5 H1_T22): state(alive), packet-loss(0.000%), latency(0.203), jitter(0.015), mos(4.404),
bandwidth-up(999998), bandwidth-dw(999997), bandwidth-bi(1999995), sla_map=0x1
Seq(7 H2_T11): state(alive), packet-loss(0.000%), latency(0.249), jitter(0.026), mos(4.404),
bandwidth-up(999998), bandwidth-dw(999996), bandwidth-bi(1999994), sla_map=0x1
Seq(8 H2_T22): state(alive), packet-loss(0.000%), latency(0.205), jitter(0.018), mos(4.404),
bandwidth-up(999998), bandwidth-dw(999997), bandwidth-bi(1999995), sla_map=0x1
```

a. Check Hub-1 metric from the external network router:

```
External_Network_Router # get router info bgp neighbors 172.31.0.1 advertised-routes
VRF 0 BGP table version is 31, local router ID is 172.31.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network             Next Hop        Metric     LocPrf Weight RouteTag Path
*>i10.0.3.0/24      172.31.0.65     70            100  32768        0 i <-/->

Total number of prefixes 1
```

b. Check Hub-2 metric from the external network router:

```
External_Network_Router # get router info bgp neighbors 172.31.0.2 advertised-routes
VRF 0 BGP table version is 31, local router ID is 172.31.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network             Next Hop        Metric     LocPrf Weight RouteTag Path
*>i10.0.3.0/24      172.31.0.65     60            100  32768        0 i <-/->

Total number of prefixes 1
```

5. H1_T11 and H2_T11 are out of SLA.

When sending traffic destined for 10.0.3.0/24, the router on the external network will prefer to send traffic to Hub-1 with lower MED 70 over Hub-2 with higher MED 80.

```
Spoke # diagnose sys sdwan  health-check
Health Check(HUB):
Seq(4 H1_T11): state(alive), packet-loss(0.000%), latency(120.194), jitter(0.011), mos(4.338),
bandwidth-up(999997), bandwidth-dw(999997), bandwidth-bi(1999994), sla_map=0x0
Seq(5 H1_T22): state(alive), packet-loss(0.000%), latency(0.191), jitter(0.008), mos(4.404),
bandwidth-up(999998), bandwidth-dw(999998), bandwidth-bi(1999996), sla_map=0x1
Seq(7 H2_T11): state(alive), packet-loss(0.000%), latency(120.235), jitter(0.029), mos(4.338),
bandwidth-up(999997), bandwidth-dw(999997), bandwidth-bi(1999994), sla_map=0x0
Seq(8 H2_T22): state(alive), packet-loss(0.000%), latency(0.215), jitter(0.016), mos(4.404),
bandwidth-up(999998), bandwidth-dw(999998), bandwidth-bi(1999996), sla_map=0x1
```

a. Check Hub-1 metric from the external network router:

```
External_Network_Router # get router info bgp neighbors 172.31.0.1 advertised-routes
VRF 0 BGP table version is 31, local router ID is 172.31.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network             Next Hop        Metric     LocPrf Weight RouteTag Path
*>i10.0.3.0/24      172.31.0.65     70            100  32768        0 i <-/->

Total number of prefixes 1
```

b. Check Hub-2 metric from the external network router:

```
External_Network_Router # get router info bgp neighbors 172.31.0.2 advertised-routes
VRF 0 BGP table version is 31, local router ID is 172.31.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network             Next Hop        Metric     LocPrf Weight RouteTag Path
*>i10.0.3.0/24      172.31.0.65     80            100  32768        0 i <-/->

Total number of prefixes 1
```

6. H1_T11, H2_T11, and H1_T22 are out of SLA.

   When sending traffic destined for 10.0.3.0/24, the router on the external network will prefer to send traffic to Hub-2 with lower MED 80 over Hub-1 with higher MED 100.

```
Spoke # diagnose sys sdwan  health-check
Health Check(HUB):
Seq(4 H1_T11): state(alive), packet-loss(0.000%), latency(120.199), jitter(0.016), mos(4.338),
bandwidth-up(999997), bandwidth-dw(999996), bandwidth-bi(1999993), sla_map=0x0
Seq(5 H1_T22): state(alive), packet-loss(0.000%), latency(120.190), jitter(0.012), mos(4.338),
bandwidth-up(999998), bandwidth-dw(999997), bandwidth-bi(1999995), sla_map=0x0
Seq(7 H2_T11): state(alive), packet-loss(0.000%), latency(120.251), jitter(0.022), mos(4.338),
bandwidth-up(999997), bandwidth-dw(999996), bandwidth-bi(1999993), sla_map=0x0
Seq(8 H2_T22): state(alive), packet-loss(0.000%), latency(0.211), jitter(0.014), mos(4.404),
bandwidth-up(999998), bandwidth-dw(999997), bandwidth-bi(1999995), sla_map=0x1
```

   a. Check Hub-1 metric from the external network router:

```
External_Network_Router # get router info bgp neighbors 172.31.0.1 advertised-routes
VRF 0 BGP table version is 31, local router ID is 172.31.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network             Next Hop        Metric     LocPrf Weight RouteTag Path
*>i10.0.3.0/24      172.31.0.65     100           100  32768        0 i <-/->

Total number of prefixes 1
```

   b. Check Hub-2 metric from the external network router:

```
External_Network_Router # get router info bgp neighbors 172.31.0.2 advertised-routes
VRF 0 BGP table version is 31, local router ID is 172.31.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network             Next Hop        Metric     LocPrf Weight RouteTag Path
*>i10.0.3.0/24      172.31.0.65     80            100  32768        0 i <-/->

Total number of prefixes 1
```

7. All of the overlays are out of SLA.

   When sending traffic destined for 10.0.3.0/24, the router on the external network will prefer to send traffic to Hub-1 with lower MED 100 over Hub-2 with higher MED 110.

```
Spoke # diagnose sys sdwan  health-check
Health Check(HUB):
Seq(4 H1_T11): state(alive), packet-loss(0.000%), latency(120.201), jitter(0.013), mos(4.338),
bandwidth-up(999997), bandwidth-dw(999997), bandwidth-bi(1999994), sla_map=0x0
Seq(5 H1_T22): state(alive), packet-loss(0.000%), latency(120.187), jitter(0.008), mos(4.338),
bandwidth-up(999998), bandwidth-dw(999998), bandwidth-bi(1999996), sla_map=0x0
Seq(7 H2_T11): state(alive), packet-loss(0.000%), latency(120.251), jitter(0.023), mos(4.338),
bandwidth-up(999997), bandwidth-dw(999997), bandwidth-bi(1999994), sla_map=0x0
Seq(8 H2_T22): state(alive), packet-loss(0.000%), latency(120.194), jitter(0.010), mos(4.338),
bandwidth-up(999998), bandwidth-dw(999998), bandwidth-bi(1999996), sla_map=0x0
```

a. Check Hub-1 metric from the external network router:

```
External_Network_Router # get router info bgp neighbors 172.31.0.1 advertised-routes
VRF 0 BGP table version is 31, local router ID is 172.31.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network             Next Hop        Metric     LocPrf Weight RouteTag Path
*>i10.0.3.0/24      172.31.0.65     100           100  32768        0 i <-/->

Total number of prefixes 1
```

b. Check Hub-2 metric from the external network router:

```
External_Network_Router # get router info bgp neighbors 172.31.0.2 advertised-routes
VRF 0 BGP table version is 31, local router ID is 172.31.0.65
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network             Next Hop        Metric     LocPrf Weight RouteTag Path
*>i10.0.3.0/24      172.31.0.65     110           100  32768        0 i <-/->

Total number of prefixes 1
```

Adaptive Forward Error Correction

Forward Error Correction (FEC) is used to control and correct errors in data transmission by sending redundant data across the VPN in anticipation of dropped packets occurring during transit. The mechanism sends out *x* number of redundant packets for every *y* number of base packets.

Adaptive FEC considers link conditions and dynamically adjusts the FEC packet ratio.

- The FEC base and redundant packet relationship is dynamically adjusted based on changes to the network SLA metrics defined in the SD-WAN SLA health checks. For example, when there is no or low packet loss in the network, FEC can work on a low redundant level sending only one redundant packet for every 10 base packets. As packet loss increases, the number of redundant packets sent can rise accordingly.
- The ToS/DSCP value is taken into consideration when mapping the FEC profile. In the first round of matching, traffic will be matched against packet-loss, latency, and bandwidth. In the second round of matching, the ToS value and mask will be used. See ToS matching and negate options on adaptive FEC profiles NEW on page 1407.

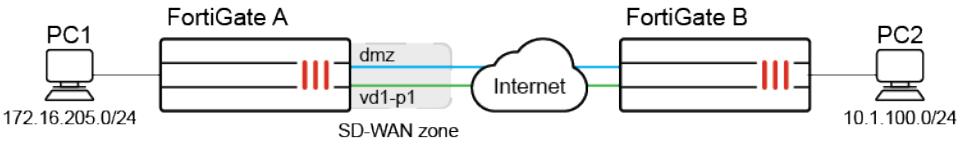
- Negate options allow users to match against the negative of a threshold. For example, negate packet-loss of 5% means any packet-loss less than 5% will be matched. See ToS matching and negate options on adaptive FEC profiles NEW on page 1407.
- FEC can be applied only to streams that are sensitive to packet loss. For Example, policies that allow the UDP based VoIP protocol can enable FEC, while TCP based traffic policies do not. This reduces unnecessary bandwidth consumption by FEC.
- FEC parity packets can be sent in a secondary overlay tunnel that is associated with the same destination as the overlay tunnel. See Use secondary tunnel for FEC redundant parity packets NEW on page 1415.
- In 8.0.0 and later, FEC can be offloaded to the NPU. This requires both the sending and receiving FortiGates to be on 8.0.0 or later.

NPU offload and anti-replay should not be used at the same time when FEC is enabled.

Basic example

In this example, an IPsec tunnel is configured between two FortiGates that have FEC enabled and supporting configuration to protect traffic that egresses FortiGate A and ingresses FortiGate B. The tunnel is an SD-WAN zone, and an SLA health-check is used to monitor the quality of the VPN overlay. The intention is to apply FEC to UDP traffic that is passing through the VPN overlay, while allowing all other traffic to pass through without FEC. An FEC profile is configured to adaptively increase redundant levels if the link quality exceeds a 10% packet loss threshold, or the bandwidth exceeds 950 Mbps.

The DMZ interface and IPsec tunnel vd1-p1 are SD-WAN members. FEC is enabled on vd1-p1, and health-check works on vd1-p1.



To configure the FortiGates:

1. On FortiGate A, enable FEC for egress traffic and NPU offloading on the IPsec tunnel vd1-p1:

```
config vpn ipsec phase1-interface
    edit "vd1-p1"
        set npu-offload enable
        set fec-egress enable
    next
end
```

2. On FortiGate B, enable FEC for ingress traffic and NPU offloading on the IPsec tunnel vd1-p1:

```
config vpn ipsec phase1-interface
    edit "vd1-p1"
        set npu-offload enable
        set fec-ingress enable
    next
end
```

3. On FortiGate A, configure SD-WAN:

The VPN overlay member (vd1-p1) must be included in the health-check and configured as the higher priority member in the SD-WAN rule.

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "dmz"
            set gateway 172.16.208.2
        next
        edit 2
            set interface "vd1-p1"
        next
    end
    config health-check
        edit "1"
            set server "2.2.2.2"
            set members 2
            config sla
                edit 1
                next
            end
        next
    end
    config service
        edit 1
            set name "1"
            set dst "all"
            set src "172.16.205.0"
            set priority-members 2 1
        next
    end
end
```

4. On FortiGate A, create a policy to specify performing FEC on UDP traffic, and a policy for other traffic:

```
config firewall policy
    edit 1
        set srcintf "port5"
        set dstintf "virtual-wan-link"
        set action accept
        set srcaddr "172.16.205.0"
        set dstaddr "all"
        set schedule "always"
        set service "ALL_UDP"
        set fec enable
    next
    edit 2
        set srcintf "any"
```

```
        set dstintf "any"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set fec enable
    next
end
```

5. On FortiGate A, configure FEC mapping to bind network SLA metrics and FEC base and redundant packets:

```
config vpn ipsec fec
    edit "m1"
        config mappings
            edit 1
                set base 8
                set redundant 2
                set packet-loss-threshold 10
            next
            edit 2
                set base 9
                set redundant 3
                set bandwidth-up-threshold 950000
            next
        end
    next
end
```

The mappings are matched from top to bottom: packet loss greater than 10% with eight base and two redundant packets, and then uploading bandwidth greater than 950 Mbps with nine base and three redundant packets.

6. On FortiGate A, apply the FEC mappings on vd1-p1:

```
config vpn ipsec phase1-interface
    edit "vd1-p1"
        set fec-health-check "1"
        set fec-mapping-profile "m1"
        set fec-base 10
        set fec-redundant 1
    next
end
```

The FEC base and redundant values are used when the link quality has not exceeded the limits specified in the FEC profile mapping. If `fec-codec` is set to `xor` the base and redundant packet values will not be updated.

To verify the results:

1. Send TCP and UDP traffic from PC1 to PC2, then check the sessions on FortiGate A:

```
# diagnose sys session list

session info: proto=6 proto_state=01 duration=12 expire=3587 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty npu
statistic(bytes/packets/allow_err): org=112/2/1 reply=112/2/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=15->102/102->15 gwy=172.16.209.2/172.16.205.11
hook=pre dir=org act=noop 172.16.205.11:39176->10.1.100.22:5001(0.0.0.0:0)
hook=post dir=reply act=noop 10.1.100.22:5001->172.16.205.11:39176(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=2 pol_uuid_idx=719 auth_info=0 chk_client_info=0 vd=0
serial=00020f7a tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=2 sdwan_service_id=1
rpdb_link_id=ff000001 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x5000c00
npu info: flag=0x82/0x81, offload=8/8, ips_offload=0/0, epid=249/74, ipid=74/86,
vlan=0x0000/0x0000
vlifid=74/249, vtag_in=0x0000/0x0001 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=5/5

session info: proto=17 proto_state=01 duration=396 expire=3591 timeout=3600 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ tun_id=0.0.0.0/10.0.0.1 vlan_cos=0/255
state=may_dirty npu
statistic(bytes/packets/allow_err): org=100366/67/1 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=15->102/102->15 gwy=172.16.209.2/0.0.0.0
hook=pre dir=org act=noop 172.16.205.11:49052->10.1.100.22:5001(0.0.0.0:0)
hook=post dir=reply act=noop 10.1.100.22:5001->172.16.205.11:49052(0.0.0.0:0)
misc=0 policy_id=2 pol_uuid_idx=593 auth_info=0 chk_client_info=0 vd=0
serial=000210fa tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=2 sdwan_service_id=1
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
npu_state=0x000c00 ofld-O ofld-R
npu info: flag=0x81/0x82, offload=9/9, ips_offload=0/0, epid=129/3910, ipid=3942/129,
vlan=0x0001/0x0000
vlifid=3942/129, vtag_in=0x0001/0x0000 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=10/10, ha_
divert=0/0
hrx info: valid=1/1, qid=11/11, npuid=1/1, sublink=1/1
```

   Non-FEC protected TCP traffic is offloaded, while FEC protected UDP traffic is not offloaded

2. On FortiGate A, check the health-check result and the corresponding FEC base and redundant packets:

```
# diagnose sys sdwan health-check
Health Check(1):
Seq(2 vd1-p1): state(alive), packet-loss(0.000%) latency(0.168), jitter(0.021), bandwidth-up
(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x1
```

Because bandwidth-up is more than 950000kbps, base and redundant are set to 9 and 3:

```
# diagnose vpn tunnel fec vd1-p1
egress:
    enabled=1 base=9 redundant=3 codec=0 timeout=10(ms)
    encode=6621 encode_timeout=6621 encode_fail=0
    tx_data=6880 tx_parity=18601
ingress:
    enabled=0 timeout=0(ms)
    fasm_cnt=0 fasm_full=0
    ipsec_fec_chk_fail=0 complete=0
    rx_data=0 rx_parity=0
    recover=0 recover_timeout=0 recover_fail=0
    rx=0 rx_fail=0
```

3. Make packet loss more than 10%, then check the health-check result and the corresponding FEC base and redundant packets again on FortiGate A:

```
# diagnose sys sdwan  health-check
Health Check(1):
Seq(2 vd1-p1): state(alive), packet-loss(15.000%) latency(0.168), jitter(0.017), bandwidth-up
(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x0
```

Because packet loss is more than 10%, entry one in FEC mapping is first matched, and base and redundant are set to 8 and 2:

```
# diagnose vpn tunnel fec  vd1-p1
egress:
    enabled=1 base=8 redundant=2 codec=0 timeout=10(ms)
    encode=6670 encode_timeout=6670 encode_fail=0
    tx_data=6976 tx_parity=18748
ingress:
    enabled=0 timeout=0(ms)
    fasm_cnt=0 fasm_full=0
    ipsec_fec_chk_fail=0 complete=0
    rx_data=0 rx_parity=0
    recover=0 recover_timeout=0 recover_fail=0
    rx=0 rx_fail=0
```

ToS matching and negate options on adaptive FEC profiles NEW

Adaptive FEC can take the ToS/DSCP value into consideration when mapping the FEC profile. In the first round of matching, traffic will be matched against packet-loss, latency, and bandwidth. In the second round of matching, the TOS value and mask will be used.

Negate options allow users to match against the negative of a threshold. For example, negate packet-loss of 5% means any packet-loss less than 5% will be matched.

For more information about how to configure ToS value and netmask and how it maps to DSCP values, please see Configuring SD-WAN rules.

To configure ToS matching:

```
config vpn ipsec fec
    edit <name>
        config mappings
            edit <id>
                config tos
                    edit <id>
                        set tos <value>
                        set tos-mask <value>
                        set base <value>
                        set redundant <value>
                    next
                end
            next
        end
    next
end
```

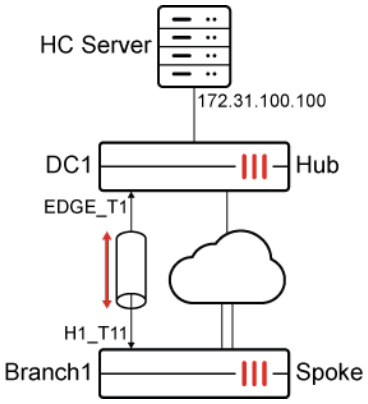
Command	Description
tos <value>	Type of service bit pattern (0x00 - 0xFF).
tos-mask <value>	Type of service evaluated bits (0x01 - 0xFF).
base <value>	Number of base FEC packets (1 - 40).
redundant <value>	Number of redundant FEC packets (0 - 20).

To configure negate options:

```
config vpn ipsec fec
    edit <name>
        config mappings
            edit <id>
                set packet-loss-threshold-negate {enable | disable}
                set latency-threshold-negate {enable | disable}
                set bandwidth-up-threshold-negate {enable | disable}
                set bandwidth-down-threshold-negate {enable | disable}
                set bandwidth-bi-threshold-negate {enable | disable}
            next
        end
    next
end
```

Command	Description
packet-loss-threshold-negate {enable | disable}	Negate packet loss threshold (default = disable).
latency-threshold-negate {enable | disable}	Negate latency threshold (default = disable).
bandwidth-up-threshold-negate {enable | disable}	Negate up bandwidth threshold (default = disable).
bandwidth-down-threshold-negate {enable | disable}	Negate down bandwidth threshold (default = disable).
bandwidth-bi-threshold-negate {enable | disable}	Negate bi-bandwidth threshold (default = disable).

Example



In this single-hub SD-WAN deployment, the spoke has defined a performance health-check targeting the HC Server behind the Hub. Adaptive FEC is configured based on the health-checks using the following criteria:

	Health Check	Matching tos 0x01 mask 0xff	Matching tos 0x02 mask 0xff	Not matching either traffic class
Rule 1	Packet-loss is above the threshold of 5%	Apply base 5 redundant 1	Apply base 5 redundant 2	Apply default base 11 redundant 1
Rule 2	Latency is less than the threshold of 20ms	Apply base 6 redundant 1	Apply base 6 redundant 2	Apply default base 12 redundant 2
Rule 3	Available upstream bandwidth is more than 50Mbps	Apply base 7 redundant 1	Apply base 7 redundant 2	Apply default base 13 redundant 3
Rule 4	Packet-loss is below the threshold of 5%	Do not send any redundant packets		

The rules are examined in the order that they are configured.

To configure the spoke:

1. Configure SD-WAN:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
        edit "overlay"
        next
        edit "underlay"
        next
    end
    config members
        ...
        edit 3
            set interface "H1_T11"
            set zone "overlay"
            set source 172.31.0.65
        next
        ...
    end
    config health-check
        edit "HUB"
            set server "172.31.100.100"
            set update-static-route disable
            set members 3
            config sla
                edit 1
                next
            end
        next
    end
    config service
        edit 1
            set dst "CORP_LAN"
            set priority-members 3
        next
    end
end
```

2. Configure VPN:

```
config vpn ipsec fec
    edit "fec_mapping"
        config mappings
            edit 1
                set base 11
                set redundant 1
                set packet-loss-threshold 5
                config tos
                    edit 1
```

```
                    set tos 0x01
                    set tos-mask 0xff
                    set base 5
                    set redundant 1
                next
                edit 2
                    set tos 0x02
                    set tos-mask 0xff
                    set base 5
                    set redundant 2
                next
            end
        next
        edit 2
            set base 12
            set redundant 2
            set latency-threshold 20
            config tos
                edit 1
                    set tos 0x01
                    set tos-mask 0xff
                    set base 6
                    set redundant 1
                next
                edit 2
                    set tos 0x02
                    set tos-mask 0xff
                    set base 6
                    set redundant 2
                next
            end
        next
        edit 3
            set base 13
            set redundant 3
            set bandwidth-up-threshold 50000
            config tos
                edit 1
                    set tos 0x01
                    set tos-mask 0xff
                    set base 7
                    set redundant 1
                next
                edit 2
                    set tos 0x02
                    set tos-mask 0xff
                    set base 7
                    set redundant 2
                next
            end
        next
```

```
            edit 4
                set base 40
                set redundant 0
                set packet-loss-threshold 5
                set packet-loss-threshold-negate enable
            next
        end
    next
end
```

```
config vpn ipsec phase1-interface
    edit "H1_T11"
        ...
        set fec-egress enable
        set fec-codec rs
        set fec-health-check "HUB"
        set fec-mapping-profile "fec_mapping"
        ...
    next
end
```

3. Configure a firewall policy:

```
config firewall policy
    edit 1
       ...
        set fec enable
    next
end
```

To configure the hub:

```
config vpn ipsec phase1-interface
    edit "EDGE_T1"
        ...
        set fec-ingress enable
        ...
    next
end
```

To test the configuration:

1. Check SD-WAN metrics on H1_T11 and the applied FEC settings:
   In this first scenario, the first 3 rules were not matched. Therefore, the 4th rule implies that no redundant packets are sent.

```
Branch1_A_FGT (root) (Interim)# diagnose sys sdwan health-check
Health Check(HUB):
Seq(3 H1_T11): state(alive), packet-loss(0.000%), latency(30.247), jitter(0.016), mos(4.389),
custom_profile(0.000), bandwidth-up(49758), bandwidth-dw(89990), bandwidth-bi(139748), sla_
map=0x0
```

```
Branch1_A_FGT# diagnose vpn  tunnel  fec  H1_T11
egress:  enabled=1
        base=40 redundant=0 codec=0 timeout=10(ms)
        encode=0 encode_timeout=0 encode_fail=0
        tx_data=0 tx_parity=0
        total_tx_data=0 total_tx_parity=0
ingress: enabled=0
        timeout=0(ms) fasm_cnt=0 fasm_full=0
        ipsec_fec_chk_fail=0 complete=0
        recover=0 recover_timeout=0 recover_fail=0
        rx_data=0 rx_parity=0
        rx=0 rx_fail=0
```

2. Make the available upstream bandwidth more than 50M by decreasing traffic on H1_T11, then check the applied FEC settings. The third rule is matched:

```
# diagnose sys sdwan health-check
Health Check(HUB):
Seq(3 H1_T11): state(alive), packet-loss(0.000%), latency(30.234), jitter(0.022), mos(4.389),
custom_profile(0.000), bandwidth-up(63244), bandwidth-dw(89993), bandwidth-bi(153237), sla_
map=0x0

Branch1_A_FGT (root) (Interim)# diagnose vpn  tunnel  fec  H1_T11
egress:  enabled=1
        base=13 redundant=3 codec=0 timeout=10(ms)
        encode=0 encode_timeout=0 encode_fail=0
        tx_data=0 tx_parity=0
        tos/mask=0x01/0xFF:
                base=7 redundant=1
                encode=0 encode_timeout=0 encode_fail=0
                tx_data=0 tx_parity=0
        tos/mask=0x02/0xFF:
                base=7 redundant=2
                encode=0 encode_timeout=0 encode_fail=0
                tx_data=0 tx_parity=0
        total_tx_data=0 total_tx_parity=0
ingress: enabled=0
        timeout=0(ms) fasm_cnt=0 fasm_full=0
        ipsec_fec_chk_fail=0 complete=0
        recover=0 recover_timeout=0 recover_fail=0
        rx_data=0 rx_parity=0
        rx=0 rx_fail=0
```

3. Make latency on H1_T11 less than 20ms, then check the applied FEC settings again. Both the latency and upstream bandwidth are above the threshold, so because rule 2 comes first, rule 2 is applied:

```
# diagnose sys sdwan  health-check
Health Check(HUB):
Seq(3 H1_T11): state(alive), packet-loss(0.000%), latency(0.247), jitter(0.027), mos(4.404),
custom_profile(0.000), bandwidth-up(89999), bandwidth-dw(89997), bandwidth-bi(179996), sla_
map=0x1
```

```
Branch1_A_FGT (root) (Interim)# diagnose vpn  tunnel  fec  H1_T11
egress:  enabled=1
        base=12 redundant=2 codec=0 timeout=10(ms)
        encode=0 encode_timeout=0 encode_fail=0
        tx_data=0 tx_parity=0
        tos/mask=0x01/0xFF:
                base=6 redundant=1
                encode=0 encode_timeout=0 encode_fail=0
                tx_data=0 tx_parity=0
        tos/mask=0x02/0xFF:
                base=6 redundant=2
                encode=0 encode_timeout=0 encode_fail=0
                tx_data=0 tx_parity=0
        total_tx_data=0 total_tx_parity=0
ingress: enabled=0
        timeout=0(ms) fasm_cnt=0 fasm_full=0
        ipsec_fec_chk_fail=0 complete=0
        recover=0 recover_timeout=0 recover_fail=0
        rx_data=0 rx_parity=0
        rx=0 rx_fail=0
```

4. Make packet-loss on H1_T11 more than 5%, then check the applied FEC settings again. All 3 rules match since health checks are above the threshold, so because rule 1 comes first, rule 1 is applied.

```
# diagnose sys sdwan  health-check
Health Check(HUB):
Seq(3 H1_T11): state(alive), packet-loss(6.000%), latency(0.254), jitter(0.029), mos(4.401),
custom_profile(0.000), bandwidth-up(89999), bandwidth-dw(89997), bandwidth-bi(179996), sla_
map=0x0

Branch1_A_FGT (root) (Interim)# diagnose vpn  tunnel  fec  H1_T11
egress:  enabled=1
        base=11 redundant=1 codec=0 timeout=10(ms)
        encode=0 encode_timeout=0 encode_fail=0
        tx_data=0 tx_parity=0
        tos/mask=0x01/0xFF:
                base=5 redundant=1
                encode=0 encode_timeout=0 encode_fail=0
                tx_data=0 tx_parity=0
        tos/mask=0x02/0xFF:
                base=5 redundant=2
                encode=0 encode_timeout=0 encode_fail=0
                tx_data=0 tx_parity=0
        total_tx_data=0 total_tx_parity=0
ingress: enabled=0
        timeout=0(ms) fasm_cnt=0 fasm_full=0
        ipsec_fec_chk_fail=0 complete=0
        recover=0 recover_timeout=0 recover_fail=0
        rx_data=0 rx_parity=0
        rx=0 rx_fail=0
```

Use secondary tunnel for FEC redundant parity packets NEW

In the original implementation of adaptive FEC, FEC parity packets are sent inside the same overlay tunnel. However, the parity packets are susceptible to being dropped at the same rate as other VPN packets.

This enhancement to adaptive FEC introduces the capability to send FEC parity packets in a secondary overlay tunnel that is associated with the same destination. The following setting can be configured on the primary tunnel that requires parity packets to be sent on a redundant tunnel:

```
config vpn ipsec phase1-interface
    edit <tunnel>
        set fec-separate-redundant-tunnel enable
    next
end
```

The redundant tunnel must have the same destination `location-id` as the original tunnel.

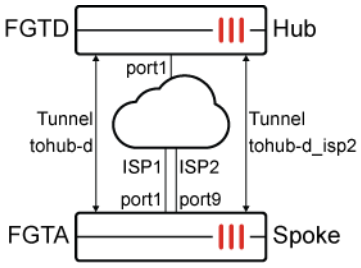
Example

In the following example, a VPN Hub (FGTD) is configured as a dialup VPN gateway. A spoke has two underlays going to different ISPs, where the primary (port1) has 10G bandwidth and the secondary (port9) has 1G bandwidth. Two tunnels are established from the Spoke to the Hub over the two underlays. The intention is to send data over the primary tunnel while using the secondary tunnel for redundancy and parity packets for the primary tunnel.

The FEC mappings are configured to send two parity packets per 8 base packets upon packet-loss threshold above 2%. Additionally, it will send three parity packets per nine base packets when bandwidth exceeds 950Mbps.

This example utilizes one Hub/destination, so the default location-id will be shared between the tunnels without any conflicts. When multiple Hubs are in use, and you need to apply the parity packet duplication on the same Hub and destination, use a different location-id for each hub.

```
config system settings
    set location-id <unique id for each hub>
end
```



To configure the hub, FGTD:

```
config vpn ipsec phase1-interface
    edit "tospokes"
        set type dynamic
        set interface "port1"
```

```
        set ike-version 2
        set peertype any
        set net-device disable
        set mode-cfg enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set add-route disable
        set dpd on-idle
        set dhgrp 20 21
        set auto-discovery-sender enable
        set fec-ingress enable
        set transport auto
        set ipv4-start-ip 100.100.100.10
        set ipv4-end-ip 100.100.100.20
        set ipv4-netmask 255.255.255.0
        set psksecret ****************
        set dpd-retryinterval 60
    next
end
```

```
config vpn ipsec phase2-interface
    edit "tospokes"
        set phase1name "tospokes"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
    next
end
```

To configure the spoke, FGTA:

1. Configure FEC mapping to bind network SLA metrics and FEC base and redundant packets:

```
config vpn ipsec fec
    edit "m1"
        config mappings
            edit 1
                set base 8
                set redundant 2
                set packet-loss-threshold 2
            next
            edit 2
                set base 9
                set redundant 3
                set bandwidth-up-threshold 950000
            next
        end
    next
end
```

2. Configure the IPsec tunnels:

   On the primary tunnel, enable FEC enabled and apply FEC mapping. The redundant tunnel does not need FEC enabled, or the fec-separate-redundant-tunnel setting.

```
config vpn ipsec phase1-interface
    edit "tohub-d"
        set interface "port1"
        set ike-version 2
        set peertype any
        set net-device disable
        set mode-cfg enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set add-route disable
        set dhgrp 20 21
        set auto-discovery-receiver enable
        set auto-discovery-shortcuts dependent
        set fec-egress enable
        set fec-separate-redundant-tunnel enable
        set fec-codec rs
        set fec-health-check "1"
        set fec-mapping-profile "m1"
        set transport auto
        set remote-gw 172.16.200.4
        set psksecret ****************
    next
    edit "tohub-d_isp2"
        set interface "port9"
        set ike-version 2
        set peertype any
        set net-device disable
        set mode-cfg enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set add-route disable
        set dhgrp 20 21
        set auto-discovery-receiver enable
        set auto-discovery-shortcuts dependent
        set transport auto
        set remote-gw 172.16.200.4
        set psksecret ****************
    next
end
```

```
config vpn ipsec phase2-interface
    edit "tohub-d"
        set phase1name "tohub-d"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
        set auto-negotiate enable
    next
    edit "tohub-d_isp2"
        set phase1name "tohub-d_isp2"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
        set auto-negotiate enable
```

```
    next
end
```

3. Configure SD-WAN:
   Assign the overlay interface as an SD-WAN member and attach it to a zone. Configure health-check on the overlay member in order to detect packet-loss and bandwidth utilization.

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "port1"
            set gateway 172.16.200.3
        next
        edit 4
            set interface "tohub-d"
        next
        edit 5
            set interface "port9"
            set gateway 11.101.1.2
        next
    end
    config health-check
            edit "1"
            set server "192.168.5.44"
            set members 4
            config sla
                edit 1
                next
            end
        next
    end
    config service
        edit 2
            set name "1_clone"
            set dst "all"
            set src "10.1.100.0"
            set priority-members 4 1
        next
    end
end
```

To verify the results:

1. Confirm that both tunnels are established on their respective interfaces on FGTA:

```
# diagnose vpn ike gateway list
```

```
vd: root/0
name: tohub-d_isp2
version: 2
interface: port9 15
addr: 11.101.1.1:5929 -> 172.16.200.4:4500
ext addr: 11.101.1.1 5929
tun_id: 10.0.0.2/::10.0.0.2
remote_location: 0.0.0.0
network-id: 0
transport: TCP
virtual-interface-addr: remote: 100.100.100.4
created: 407s ago
peer-id: 172.16.200.4
peer-id-auth: no
assigned IPv4 address: 100.100.100.11/255.255.255.0
nat: me
auto-discovery: 2 receiver
pending-queue: 0
PPK: no
IKE SA: created 1/1  established 1/1  time 36000/36000/36000 ms
IPsec SA: created 1/1  established 1/1  time 36000/36000/36000 ms

  id/spi: 10217 77f9b317d1c985cd/cd40650eb175ec62
  direction: initiator
  status: established 407-371s ago = 36000ms
  proposal: aes128-sha256
  child: no
  SK_ei: 3fc9a5151af50e58-fec5cfed3c436fb1
  SK_er: de07bdc42f053f82-cdc945fc5a7ad5f4
  SK_ai: 0262f60ee41bafa2-2f7b9fc40f38e2b6-415b31885b81b778-b28c62a9915d2e5d
  SK_ar: 693d5e98d6695490-bc9c26b8a15e5322-0285e96df10e48cc-3535038e2c9a442d
  PPK: no
  message-id sent/recv: 2/0
  QKD: no
  PQC-KEM (IKE): no
  PQC-KEM (all IPsec): no
  lifetime/rekey: 86400/85728
  DPD sent/recv: 00000000/00000000
  peer-id: 172.16.200.4

vd: root/0
name: tohub-d
version: 2
interface: port1 7
addr: 172.16.200.1:500 -> 172.16.200.4:500
ext addr: 172.16.200.1 500
tun_id: 172.16.200.4/::172.16.200.4
remote_location: 0.0.0.0
network-id: 0
transport: UDP
virtual-interface-addr: remote: 100.100.100.4
created: 377s ago
```

```
peer-id: 172.16.200.4
peer-id-auth: no
assigned IPv4 address: 100.100.100.10/255.255.255.0
auto-discovery: 2 receiver
pending-queue: 0
PPK: no
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 10218 43bc6decd5bc56af/d5a405f1ff957a2f
  direction: initiator
  status: established 377-377s ago = 0ms
  proposal: aes128-sha256
  child: no
  SK_ei: 677aadd417ae719f-6cc99b2215be8711
  SK_er: f2b69ab1aecc08d6-04236a6849b56cae
  SK_ai: e9294b9b9087f973-9aebb458b27f5b13-39aab03bfee864bf-e616994e29cae238
  SK_ar: 7ef28f030c54ef5a-84c9765946e408da-d026e7294a209973-3d607b182eb86ec4
  PPK: no
  message-id sent/recv: 2/0
  QKD: no
  PQC-KEM (IKE): no
  PQC-KEM (all IPsec): no
  lifetime/rekey: 86400/85722
  DPD sent/recv: 00000000/00000000
  peer-id: 172.16.200.4
```

2. Confirm that FEC is enabled in the egress direction for the tohub-d tunnel:

```
# diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=tohub-d_isp2 ver=2 serial=5 11.101.1.1:5929->172.16.200.4:4500 nexthop=11.101.1.2 tun_
id=10.0.0.2 tun_id6=::10.0.0.2 status=up dst_mtu=1500 weight=1 country=ZZ
bound_if=15 real_if=15 lgwy=static/1 tun=intf mode=auto/1 encap=none options[0x228]=npu frag-
rfc  run_state=0 role=primary accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=5 ilast=31 olast=31 ad=r/2
stat: rxp=347 txp=22 rxb=458878 txb=1410
dpd: mode=on-demand on=1 status=ok idle=20000ms retry=3 count=0 seqno=1802
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=tohub-d_isp2 proto=0 sa=1 ref=3 serial=1 auto-negotiate adr
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=1a227 type=00 soft=0 mtu=1438 expire=42519/0B replaywin=2048
       seqno=17 esn=0 replaywin_lastseq=0000015c qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42902/43200
  dec: spi=18e0b19e esp=aes key=16 c005a5c587fcd4ad85686868323f7a21
       ah=sha256 key=32 3da0b40b525f363355acb3dd445f0131c924b14bb3f704dfd8d7b205412e01d9
  enc: spi=c0e46a65 esp=aes key=16 a161fb71279cd02bed75614c16ce48d0
       ah=sha256 key=32 34977a64e77716c20d4fdfb4619f0745f8d850e448e0330a64baed64dbc24f91
  dec:pkts/bytes=347/458878, enc:pkts/bytes=22/2952
```

```
  npu_flag=00 npu_rgwy=0.0.0.0:0 npu_lgwy=0.0.0.0:0npu_selid=6
  dec_npuid=0 enc_npuid=0 dec_engid=-1 enc_engid=-1 dec_saidx=-1 enc_saidx=-1
-----------------------------------------------------
name=tohub-d ver=2 serial=4 172.16.200.1:0->172.16.200.4:0 nexthop=172.16.200.4 tun_
id=172.16.200.4 tun_id6=::172.16.200.4 status=up dst_mtu=1500 weight=1 country=ZZ
bound_if=7 real_if=7 lgwy=static/1 tun=intf mode=auto/1 encap=none options[0x228]=npu frag-rfc
 run_state=0 role=primary accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=4 ilast=1 olast=1 ad=r/2
stat: rxp=375 txp=937 rxb=40092 txb=61856
dpd: mode=on-demand on=1 status=ok idle=20000ms retry=3 count=0 seqno=16
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=1 ingress=0
proxyid=tohub-d proto=0 sa=1 ref=31 serial=1 auto-negotiate adr
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=6 options=1a227 type=00 soft=0 mtu=1342 expire=42509/0B replaywin=2048
       seqno=17aa esn=0 replaywin_lastseq=0000004d qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42898/43200
  dec: spi=18e0b19f esp=aes key=16 3e578ce917adbf940e9ad602baa71a92
       ah=sha256 key=32 4a23b8861918b8ad336cae572dd17f808e64ee22367a219d00249d06405736cf
  enc: spi=c0e46a64 esp=aes key=16 b5e7464c94b4847c30c804f7effa8a0d
       ah=sha256 key=32 bbde5831078147c8b3fd88d0d5112b6110cce75bdb5b23392cc3c2f90c9e4336
  dec:pkts/bytes=6/240, enc:pkts/bytes=937/125532
  npu_flag=03 npu_rgwy=172.16.200.4:0 npu_lgwy=172.16.200.1:0npu_selid=5
  dec_npuid=1 enc_npuid=1 dec_engid=-1 enc_engid=-1 dec_saidx=7399 enc_saidx=4
```

3. Confirm that the profile selected is sending three parity packets per nine base packets:

```
# diagnose vpn tunnel fec tohub-d
egress:  enabled=1
        base=9 redundant=3 codec=0 timeout=10(ms)
        encode=132 encode_timeout=129 encode_fail=0
        tx_data=166 tx_parity=396
        total_tx_data=166 total_tx_parity=396
ingress: enabled=0
        timeout=0(ms) fasm_cnt=0 fasm_full=0
        ipsec_fec_chk_fail=0 complete=0
        recover=0 recover_timeout=0 recover_fail=0
        rx_data=0 rx_parity=0
        rx=0 rx_fail=0
```

Duplicate packets based on SD-WAN rules

SD-WAN duplication rules can specify SD-WAN service rules to trigger packet duplication. This allows the duplication to occur based on an SD-WAN rule instead of the source, destination, and service parameters in the duplication rule.

1. Packets can be forced to duplicate to all members of the same SD-WAN zone. See Duplicate packets on other zone members on page 1423 for details.

For example, in Spoke 1 set `packet-duplication` to `force` so that when a client sends a packet to the server, it is duplicated to all members of the same zone as long as its health check is alive. If a members health check is dead, then the member is removed from the SD-WAN duplication zone.

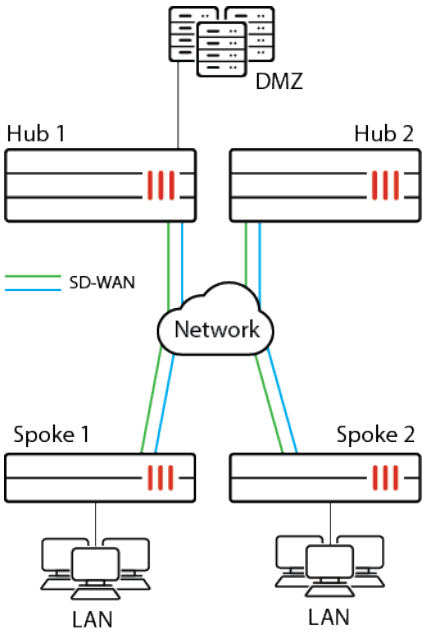
2. Packets can be duplicated to other members of the SD-WAN zone on-demand only when the condition of the link is not good enough.

   Set `packet-duplication` to `on-demand`. If `sla-match-service` is disabled, when all the SLAs of the member exceed threshold (sla_map=0), the packet is duplicated. But when the SLAs are within threshold (sla_map!=0), the packet is not duplicated.

   If `sla-match-service` is enabled, then only the SLA health checks and targets used in the service rule need to exceed threshold in order to trigger packet duplication.

3. Packets can be duplicated to all members of the same SD-WAN zone when the traffic matches one or more regular SD-WAN service rules.

The following example shows the third type of packet duplication.



In this example, SD-WAN is configured with three members: vpn1, vpn2, and vpn3. Service rule 1 controls all traffic from 10.100.20.0/24 to 172.16.100.0/24 using member 1.

To send a duplicate of the traffic that matches service rule 1 using member 2, members 1 and 2 are added to the same SD-WAN zone, and a duplication rule is configured with service-id set to 1.

To send a duplicate of the traffic that matches service rule 1 using member 2:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
        edit "zone2"
        next
    end
```

```
    config members
        edit 1
            set interface "vpn1"
        next
        edit 2
            set interface "vpn2"
        next
        edit 3
            set interface "vpn3"
            set zone "zone2"
        next
    end
    config service
        edit 1
            set dst "172.16.100.0"
            set src "10.100.20.0"
            set priority-members 1
        next
    end
    config duplication
        edit 1
            set service-id 1
            set packet-duplication force
        next
    end
end
```

Duplicate packets on other zone members

When duplication rules are used, packets are duplicated on other good links within the SD-WAN zone and de-duplicated on the destination FortiGate. Use `force` mode to force duplication on other links within the SD-WAN zone, or use `on-demand` mode to trigger duplication only when SLA fails on the selected member.

The duplication rule is configured in the CLI by using the `config duplication` command. The following options can be configured:

Parameter	Description
srcaddr	Source address or address group names.
dstaddr	Destination address or address group names.
srcaddr6	Source IPv6 address or IPv6 address group names.
dstaddr6	Destination IPv6 address or IPv6 address group names.
srcintf	Incoming (ingress) interfaces or zones.
dstintf	Outgoing (egress) interfaces or zones.
service	Service and service group names.
packet-duplication	Configure packet duplication method.

Parameter	Description
	• disable: Disable packet duplication (default). • force: Duplicate packets across all interface members of the SD-WAN zone. • on-demand: Duplicate packets across all interface members of the SD-WAN zone based on the link quality.
packet-de-duplication	Enable/disable discarding of packets that have been duplicated (default = disable).

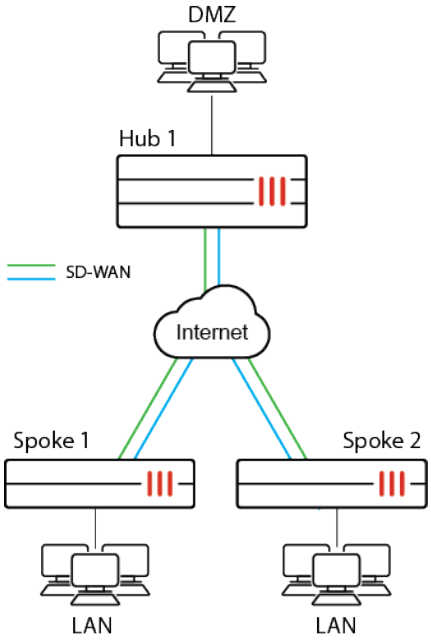
Additional settings under `config system sdwan`

The `duplication-max-num <integer>` option under `config system sdwan` is the maximum number of interface members that a packet is duplicated on in the SD-WAN zone (2 - 4, default = 2). If this value is set to 3, the original packet plus two more copies are created. If there are three member interfaces in the SD-WAN zone and the `duplication-max-num` is set to 2, the packet duplication follows the configuration order, so the packets are duplicated on the second member.

The `duplication-max-discrepancy <latency>` option under `config system sdwan` is configured to be 250ms by default and accepts a value from 250-1000ms. This latency setting defines the interval in which a duplicate packet received will be dropped when `packet-de-duplication` is enabled. It is recommended to keep this value at 250ms to minimize the packets cached in memory. However, if the discrepancy in latency between members is high, such as when one transport is through a satellite link, then this value can be adjusted upwards.

Example

The packet duplication feature works best in a spoke-spoke or hub-and-spoke topology. In this example, a hub-and-spoke ADVPN topology is used. Before shortcuts are established, Hub 1 forwards the duplicate packets from Spoke 1 to Spoke 2. Once shortcuts are established, Hub 1 is transparent, and duplicate packets are exchanged directly between the spokes.



To configure packet duplication between Spoke 1 and Spoke 2:

1. Configure Spoke 1:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
        edit "sdwanzone_v4"
        next
    end
    config members
        edit 1
            set interface "t1"
            set zone "sdwanzone_v4"
        next
        edit 4
            set interface "t21"
            set zone "sdwanzone_v4"
        next
        edit 2
            set interface "t2"
            set zone "sdwanzone_v4"
        next
    end
    config health-check
        edit "h1"
            set server "10.34.1.1"
            set interval 1000
            set failtime 10
            set members 1 2
            config sla
                edit 1
                    set packetloss-threshold 40
                next
            end
        next
    end
    config duplication
        edit 1
            set srcaddr "all"
            set dstaddr "all"
            set srcintf "port1"
            set dstintf "sdwanzone_v4"
            set service "ALL"
            set packet-duplication force
            set packet-de-duplication enable
        next
    end
end
```

2. Configure Spoke 2 with similar settings.

Routing

The following topics provide instructions on SD-WAN advanced routing:

- Routing overview on page 1426
  - Dynamic BGP on loopback dual hub example on page 1427
  - Add ADVPN to dynamic BGP on loopback example on page 1433
- Advanced BGP settings on page 1434
  - Using BGP tags with SD-WAN rules on page 1434
  - BGP multiple path support on page 1438
  - Controlling traffic with BGP route mapping and service rules on page 1440
  - Applying BGP route-map to multiple BGP neighbors on page 1448
  - Using multiple members per SD-WAN neighbor configuration on page 1454

Routing overview

In a conventional (non-SD-WAN) design, routing oversees the steering of traffic. It is, therefore, the responsibility of routing to select the best path out of all available options. While different routing techniques to steer traffic are compatible with Fortinet's SD-WAN implementation, it is recommended to use the SD-WAN intelligence to select the best path to steer traffic. This is the purpose of SD-WAN, as it accounts for various types of link and application qualities, like the health of each path and the type of applications, before steering traffic towards an SD-WAN member.

In the case of direct internet access on a single site, it is recommended to implement ECMP (Equal cost multi-path) routes over all available paths, and allow SD-WAN rules to then specify the traffic steering. For more information about ECMP, see Equal cost multi-path on page 595.

In multi-site Hub and Spoke SD-WAN deployments, routes over your SD-WAN overlay network can be learned several ways. Static routing is supported; however, most routing implementations will leverage dynamic routing to ensure their network scales and can respond to changes in the environment. BGP is recommended to exchange routes between all sites over the overlays. When utilizing BGP for dynamic routing, several options for BGP peering are available.

BGP per overlay

This is the traditional way of sharing routes between all SD-WAN nodes. BGP per overlay builds an iBGP session using each end of the tunnel as a peer. This reduces the complexity of the deployment but results in a large number of iBGP sessions and routing table updates on the hub device(s). This method of routing is still supported, however it is not recommended.

BGP on loopback

In the BGP on loopback design, every Spoke establishes a single iBGP session towards the hub(s). This single iBGP session is terminated on the loopback interface, which uniquely identifies each SD-WAN node (Hub and

Spokes). The Spoke advertises its LAN prefix(es) over this single iBGP session per Hub. This method of routing reduces the load on the Hub(s) and is required for ADVPN 2.0.

Dynamic BGP versus route reflection

Starting from FOS 7.4, a new method to exchange routes directly between a pair of Spokes by dynamically building a BGP session between them over an ADVPN shortcut was added. This new mechanism is called "Dynamic BGP". While the Hubs learn all the Spoke routes, they no longer have to re-advertise them to other Spokes, meaning BGP Route Reflection (RR) on the Hubs is no longer required.

Routing method	BGP per overlay	BGP on loopback
BGP add-path	Required	No
Route exchange method	Route Reflector or Dynamic BGP	Route Reflector or Dynamic BGP
Main advantage	Simplicity	Reduced routing table ADVPN 2.0 support
Main limitations	Limited scalability Does not work with ADVPN 2.0	None

Dynamic BGP on loopback dual hub example

This example shows how to configure SD-WAN for a dual hub topology using dynamic BGP on loopback. The steps assume the spoke and hub FortiGates have configured LAN & WAN interfaces and default routing in place.

The FortiGates will require further SD-WAN configuration relevant to your business, such as SD-WAN rules for various applications, and firewall policies to permit and deny traffic as required. Similarly, security profiles are not discussed but should be implemented in accordance to the business's security requirements.

Note that both hubs share the same IP address for HUB-Lo. This is one of the benefits of BGP on loopback.

Spoke

To configure the spoke:

1. Configure the Loopback interface to be used for BGP peering.

```
config system interface
    edit "Branch-Lo"
        set vdom "root"
        set ip 172.16.0.1 255.255.255.255
        set allowaccess ping
        set type loopback
    next
end
```

2. Configure a unique location ID.

```
config system settings
    set location-id 172.16.0.1
end
```

3. Configure IKE to inject a static route to reach the loopback on all phase1-interfaces toward the hubs. Note, additional VPN configuration required. See Site-to-site VPN.

```
config vpn ipsec phase1-interface
    edit HUB1_VPN1
        set exchange-ip-addr4 172.16.0.1
    next
    edit HUB1_VPN2
        set exchange-ip-addr4 172.16.0.1
    next
    edit HUB2_VPN1
        set exchange-ip-addr4 172.16.0.1
    next
    edit HUB2_VPN2
        set exchange-ip-addr4 172.16.0.1
    next
end
```

4. Configure BGP:

```
config router bgp
    set as 65000
    set router-id 172.16.0.1
    set ibgp-multipath enable
    set recursive-next-hop enable
    set graceful-restart enable
    config neighbor
        edit 172.16.255.250
            set soft-reconfiguration enable
            set capability-graceful-restart enable
            set advertisement-interval 1
            set interface " Branch-Lo"
            set update-source " Branch-Lo"
            set connect-timer 1
            set remote-as 65000
        next
        edit 172.16.255.252
            set soft-reconfiguration enable
            set capability-graceful-restart enable
            set advertisement-interval 1
            set interface " Branch-Lo"
            set update-source " Branch-Lo"
            set connect-timer 1
            set remote-as 65000
        next
    end
    config network
        edit 1
```

```
            set prefix 10.1.1.0/24
        next
    end
end
```

5. Configure SD-WAN members to utilize the loopback as the source for health checks. Extra configuration provided to define SD-WAN interfaces referenced in later steps. This is required as the VPN tunnel interfaces do not have IP addresses assigned.

```
config system sdwan
    config zone
        edit "virtual-wan-link"
        next
        edit "WAN1"
        next
        edit "WAN2"
        next
        edit "HUB1"
        next
        edit "HUB2"
        next
    end
    config members
        edit 1
            set interface "port1"
            set zone "WAN1"
        next
        edit 2
            set interface "port2"
            set zone "WAN2"
        next
        edit 3
            set interface "HUB1-VPN1"
            set zone "HUB1"
            set source 172.16.0.1
        next
        edit 4
            set interface "HUB1-VPN2"
            set zone "HUB1"
            set source 172.16.0.1
        next
        edit 5
            set interface "HUB2-VPN1"
            set zone "HUB2"
            set source 172.16.0.1
        next
        edit 6
            set interface "HUB2-VPN2"
            set zone "HUB2"
            set source 172.16.0.1
        next
```

```
    end
end
```

6. Configure a firewall policy to permit health check probes.

```
config firewall policy
    edit 4
        set name "Health Check Access"
        set srcintf "Branch-Lo"
        set dstintf "HUB1" "HUB2"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "PING BGP"
    next
end
```

Hub

To configure the hub:

1. Configure the Loopback interfaces to be used for BGP peering and health checks.

```
config system interface
    edit "HUB-Lo"
        set vdom "root"
        set ip 172.16.255.251 255.255.255.255
        set allowaccess ping
        set type loopback
    next
    edit "BGP-Lo"
        set vdom "root"
        set ip 172.16.255.252 255.255.255.255
        set allowaccess ping
        set type loopback
    next
end
```

> ⚠ You should never share one loopback interface for both BGP peering and HUB health checks (done by spokes). See HUB and BGP loopback interfaces on page 1432.

2. Configure a unique location ID.

```
config system settings
    set location-id 172.16.255.252
end
```

3. Configure IKE to inject a static route to reach the loopback on all phase1-interfaces toward the spokes.

```
config vpn ipsec phase1-interface
    edit VPN1
        set exchange-ip-addr4 172.16.255.252
    next
    edit VPN2
        set exchange-ip-addr4 172.16.255.252
    next
end
```

4. Configure BGP:

```
config router bgp
    set as 65000
    set router-id 172.16.0.1
    set ibgp-multipath enable
    set recursive-next-hop enable
    set graceful-restart enable
    config neighbor group
        edit “EDGE”
            set soft-reconfiguration enable
            set capability-graceful-restart enable
            set advertisement-interval 1
            set next-hop-self enable
            set interface " BGP-Lo"
            set update-source " BGP-Lo"
            set remote-as 65000
        next
    end
       config neighbor-range
        edit 1
            set prefix 172.16.0.0 255.255.224.0
            set neighbor-group "DYN_BRANCH"
    next
    end
    config network
        edit 1
            set prefix 172.16.0.0 255.255.0.0
            set route-map "LOCAL_REGION"
        next
        edit 10
            set prefix 10.1.0.0 255.255.0.0
        next
    end
end
```

5. Configure VPN tunnels to be SD-WAN members:

```
config system sdwan
    set status enable
    config zone
        edit "OVERLAYS"
        next
```

```
    end
    config members
        edit 1
            set interface "VPN1"
            set zone "OVERLAYS"
        next
        edit 2
            set interface "VPN2"
            set zone "OVERLAYS"
        next
    end
end
```

6. Configure a firewall policy to permit health check probes and BGP peering.

```
config firewall policy
    edit 4
        set name "Health Check Access"
        set srcintf "VPN1" "VPN2"
        set dstintf "HUB-Lo"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
    next
    edit 5
        set name "Peering"
        set srcintf "VPN1" "VPN2"
        set dstintf "BGP-Lo"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "PING" "BGP"
    next
end
```

HUB and BGP loopback interfaces

The reason to configure separate interfaces on the hub for BGP peering and hub health checks (done by spokes, and/or other hubs) is that, when a health check is configured, a kernel route is installed for that IP address. These kernel routes exist to support the health check and will not be removed even when the health check fails.

If the BGP neighbor configuration incorrectly references the same IP address as the health check, the spoke will continue to send BGP traffic over the tunnel that the peering was initially established over, even if the health check for this tunnel is failing.

Example spoke health-check configuration:

```
config health-check
    edit "Hub_HC"
        set server "172.16.255.251"
        set members 1 2
    next
end
```

Spoke kernel route lookup:

```
get router info kernel | grep 172.16.255.251

tab=254 vf=0 scope=0 type=1 proto=18 prio=0 172.16.0.1/255.255.255.255/0->172.16.255.251/32
pref=0.0.0.0 gwy=20.0.0.1 dev=20(advpn)
tab=254 vf=0 scope=0 type=1 proto=18 prio=0 172.16.0.1/255.255.255.255/0->172.16.255.251/32
pref=0.0.0.0 gwy=30.0.0.1 dev=21(advpn_b)
```

Add ADVPN to dynamic BGP on loopback example

Spoke

To configure the spoke:

1. 1. Define a health check for spoke to hub. Example settings provided for SLA metrics.

```
config system sdwan
    config health-check
        edit "HUB_HC"
            set server “172.16.255.251”
            set update-cascade-interface disable
            set update-static-route disable
            set sla-fail-log-period 10
            set sla-pass-log-period 10
            set members 3 4 5 6
            config sla
                edit 1
                    set latency-threshold 255
                    set jitter-threshold 55
                    set packetloss-threshold 1
                next
            end
        next
    next
end
```

2. Enable ADVPN 2.0 on the HUB SD-WAN Zones and assign the HUB_HC health-check.

```
config system sdwan
    config zone
        edit "HUB1"
            set advpn-select enable
```

```
            set advpn-health-check "HUB_HC"
        next
        edit "HUB2"
            set advpn-select enable
            set advpn-health-check "HUB_HC"
        next
    end
end
```

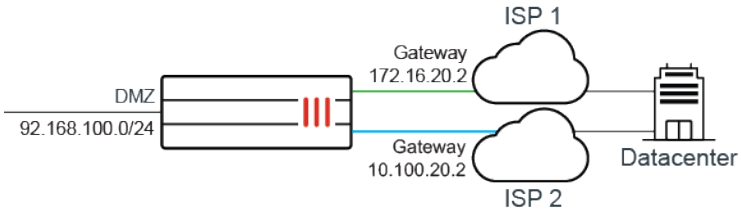
Advanced BGP settings

This section includes the following topics about advanced BGP settings:

- Using BGP tags with SD-WAN rules on page 1434
- BGP multiple path support on page 1438
- Controlling traffic with BGP route mapping and service rules on page 1440
- Applying BGP route-map to multiple BGP neighbors on page 1448
- Using multiple members per SD-WAN neighbor configuration on page 1454
- Allow SD-WAN hubs to suppress BGP routes when all links to a spoke are down on page 1460

Using BGP tags with SD-WAN rules

SD-WAN rules can use Border Gateway Protocol (BGP) learned routes as dynamic destinations.



In this example, a customer has two ISP connections, wan1 and wan2. wan1 is used primarily for direct access to internet applications, and wan2 is used primarily for traffic to the customer's data center.

The customer could create an SD-WAN rule using the data center's IP address range as the destination to force that traffic to use wan2, but the data center's IP range is not static. Instead, a BGP tag can be used.

For this example, wan2's BGP neighbor advertises the data center's network range with a community number of 30:5.

This example assumes that SD-WAN is enabled on the FortiGate, wan1 and wan2 are added as SD-WAN members in the *virtual-wan-link* SD-WAN zone, and a policy and static route have been created. See SD-WAN quick start on page 1041 for details.

FortiOS supports IPv4 and IPv6 route tags.

To configure BGP tags with SD-WAN rules:

1. Configure the community list:

```
config router community-list
    edit "30:5"
        config rule
            edit 1
                set action permit
                set match "30:5"
            next
        end
    next
end
```

2. Configure the route map:

```
config router route-map
    edit "comm1"
        config rule
            edit 1
                set match-community "30:5"
                set set-route-tag 15
            next
        end
    next
end
```

3. Configure BGP:

```
config router bgp
    set as xxxxx
    set router-id xxxx
    config neighbor
        edit "10.100.20.2"
            set soft-reconfiguration enable
            set remote-as xxxxx
            set route-map-in "comm1"
        next
    end
end
```

4. Configure the route tag address object:

```
config firewall address
    edit "DataCenter_route_tag_15"
        set type route-tag
        set route-tag 15
    next
end
```

5. Configure a firewall policy:

```
config firewall policy
    edit 1
        set name "1"
        set srcintf "dmz"
        set dstintf "virtual-wan-link"
        set srcaddr "all"
        set dstaddr "DataCenter_route_tag_15"
        set action accept
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

6. Edit the SD-WAN configuration:

```
config system sdwan
    set status enable
    config members
        edit 1
            set interface "wan1"
            set gateway 172.16.20.2
        next
        edit 2
            set interface "wan2"
        next
    end
    config service
        edit 1
            set name "DataCenter"
            set mode manual
            set priority-members 2
            set dst "DataCenter_route_tag_15"
        next
    end
end
```

Troubleshooting BGP tags with SD-WAN rules

Check the network community

Use the `get router info bgp network` command to check the network community:

```
# get router info bgp network
BGP table version is 5, local router ID is 1.1.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal,
              S Stale
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network Next Hop Metric LocPrf Weight RouteTag Path
```

```
*> 0.0.0.0/0 10.100.1.5 32768 0 ?
*> 1.1.1.1/32 0.0.0.0 32768 0 ?
*> 10.1.100.0/24 172.16.203.2 32768 0 ?
*> 10.100.1.0/30 0.0.0.0 32768 0 ?
*> 10.100.1.4/30 0.0.0.0 32768 0 ?
*> 10.100.1.248/29 0.0.0.0 32768 0 ?
*> 10.100.10.0/24 10.100.1.5 202 10000 15 20 e
*> 172.16.200.0/24 0.0.0.0 32768 0 ?
*> 172.16.200.200/32
                    0.0.0.0 32768 0 ?
*> 172.16.201.0/24 172.16.200.4 32768 0 ?
*> 172.16.203.0/24 0.0.0.0 32768 0 ?
*> 172.16.204.0/24 172.16.200.4 32768 0 ?
*> 172.16.205.0/24 0.0.0.0 32768 0 ?
*> 172.16.206.0/24 0.0.0.0 32768 0 ?
*> 172.16.207.1/32 0.0.0.0 32768 0 ?
*> 172.16.207.2/32 0.0.0.0 32768 0 ?
*> 172.16.212.1/32 0.0.0.0 32768 0 ?
*> 172.16.212.2/32 0.0.0.0 32768 0 ?
*> 172.17.200.200/32
                    0.0.0.0 32768 0 ?
*> 172.27.1.0/24 0.0.0.0 32768 0 ?
*> 172.27.2.0/24 0.0.0.0 32768 0 ?
*> 172.27.5.0/24 0.0.0.0 32768 0 ?
*> 172.27.6.0/24 0.0.0.0 32768 0 ?
*> 172.27.7.0/24 0.0.0.0 32768 0 ?
*> 172.27.8.0/24 0.0.0.0 32768 0 ?
*> 172.29.1.0/24 0.0.0.0 32768 0 ?
*> 172.29.2.0/24 0.0.0.0 32768 0 ?
*> 192.168.1.0 0.0.0.0 32768 0 ?

Total number of prefixes 28

# get router info bgp network 10.100.11.0
BGP routing table entry for 10.100.10.0/24
Paths: (2 available, best 1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
   172.10.22.2
  20
    10.100.20.2 from 10.100.20.2 (6.6.6.6)
      Origin EGP metric 200, localpref 100, weight 10000, valid, external, best
      Community: 30:5 <<<<==========================
      Last update: Wen Mar 20 18:45:17 2019
```

Check dynamic BGP addresses used in policy routes

Use the `diagnose firewall proute list` command to check dynamic BGP addresses used in policy routes:

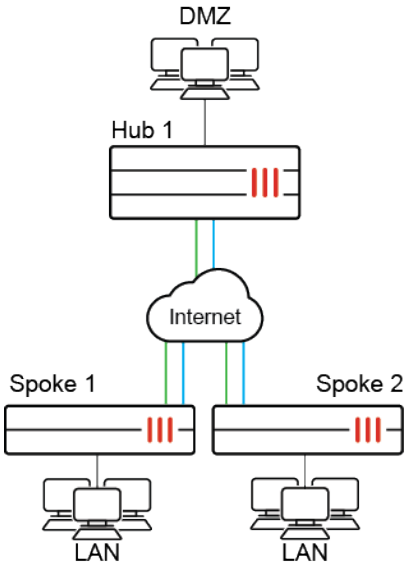
```
# diagnose firewall proute list
list route policy info(vf=root):

id=4278779905 vwl_service=1(DataCenter) flags=0x0 tos=0x00 tos_mask=0x00 protocol=0 sport=0:65535
```

```
iif=0 dport=1-65535 oif=16
source wildcard(1): 0.0.0.0/0.0.0.0
destination wildcard(1): 10.100.11.0/255.255.255.0
```

BGP multiple path support

BGP supports multiple paths, allowing an ADVPN to advertise multiple paths. This allows BGP to extend and keep additional network paths according to RFC 7911.

In this example, Spoke1 and Spoke2 each have four VPN tunnels that are connected to the Hub with ADVPN. The Spoke-Hub has established four BGP neighbors on all four tunnels.



Spoke 1 and Spoke 2 can learn four different routes from each other.

To configure the hub:

```
config router bgp
    set as 65505
    set router-id 11.11.11.11
    set ibgp-multipath enable
    set additional-path enable
    set additional-path-select 4
    config neighbor-group
        edit "gr1"
            set capability-default-originate enable
            set remote-as 65505
            set additional-path both
            set adv-additional-path 4
            set route-reflector-client enable
        next
    end
    config neighbor-range
```

```
        edit 1
            set prefix 10.10.0.0 255.255.0.0
            set neighbor-group "gr1"
        next
    end
    config network
        edit 12
            set prefix 11.11.11.11 255.255.255.255
        next
    end
end
```

To configure a spoke:

```
config router bgp
    set as 65505
    set router-id 2.2.2.2
    set ibgp-multipath enable
    set additional-path enable
    set additional-path-select 4
    config neighbor
        edit "10.10.100.254"
            set soft-reconfiguration enable
            set remote-as 65505
            set additional-path both
            set adv-additional-path 4
        next
        edit "10.10.200.254"
            set soft-reconfiguration enable
            set remote-as 65505
            set additional-path both
            set adv-additional-path 4
        next
        edit "10.10.203.254"
            set soft-reconfiguration enable
            set remote-as 65505
            set additional-path both
            set adv-additional-path 4
        next
        edit "10.10.204.254"
            set soft-reconfiguration enable
            set remote-as 65505
            set additional-path both
            set adv-additional-path 4
        next
    end
    config network
        edit 3
            set prefix 22.1.1.0 255.255.255.0
        next
    end
end
```

To view the BGP routing table on a spoke:

```
Spoke1 # get router info routing-table bgp
    Routing table for VRF=0
    B*      0.0.0.0/0 [200/0] via 10.10.200.254, vd2-2, 03:57:26
        [200/0] via 10.10.203.254, vd2-3, 03:57:26
        [200/0] via 10.10.204.254, vd2-4, 03:57:26
        [200/0] via 10.10.100.254, vd2-1, 03:57:26
    B       1.1.1.1/32 [200/0] via 11.1.1.1 (recursive via 12.1.1.1), 03:57:51
        [200/0] via 11.1.1.1 (recursive via 12.1.1.1), 03:57:51
        [200/0] via 11.1.1.1 (recursive via 12.1.1.1), 03:57:51
        [200/0] via 11.1.1.1 (recursive via 12.1.1.1), 03:57:51
    B       11.11.11.11/32 [200/0] via 10.10.200.254, vd2-2, 03:57:51
        [200/0] via 10.10.203.254, vd2-3, 03:57:51
        [200/0] via 10.10.204.254, vd2-4, 03:57:51
        [200/0] via 10.10.100.254, vd2-1, 03:57:51
    B       33.1.1.0/24 [200/0] via 10.10.204.3, vd2-4, 03:57:26
        [200/0] via 10.10.203.3, vd2-3, 03:57:26
        [200/0] via 10.10.200.3, vd2-2, 03:57:26
        [200/0] via 10.10.100.3, vd2-1, 03:57:26
        [200/0] via 10.10.204.3, vd2-4, 03:57:26
        [200/0] via 10.10.203.3, vd2-3, 03:57:26
        [200/0] via 10.10.200.3, vd2-2, 03:57:26
        [200/0] via 10.10.100.3, vd2-1, 03:57:26
        [200/0] via 10.10.204.3, vd2-4, 03:57:26
        [200/0] via 10.10.203.3, vd2-3, 03:57:26
        [200/0] via 10.10.200.3, vd2-2, 03:57:26
        [200/0] via 10.10.100.3, vd2-1, 03:57:26
        [200/0] via 10.10.204.3, vd2-4, 03:57:26
        [200/0] via 10.10.203.3, vd2-3, 03:57:26
        [200/0] via 10.10.200.3, vd2-2, 03:57:26
        [200/0] via 10.10.100.3, vd2-1, 03:57:26
```

Controlling traffic with BGP route mapping and service rules

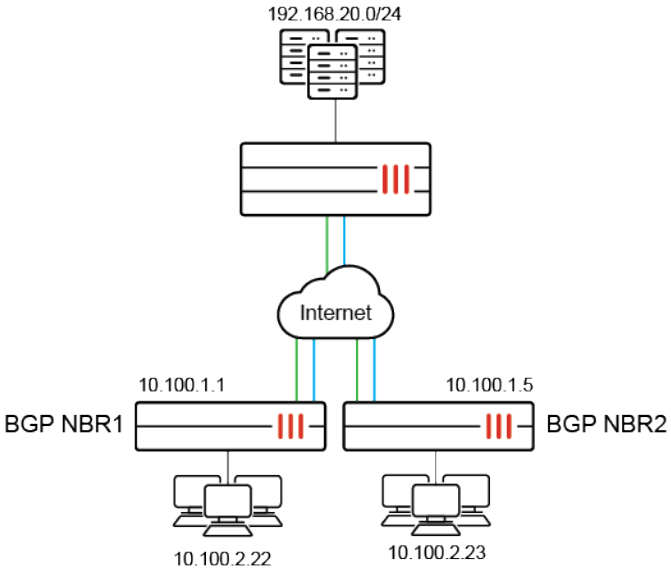
SD-WAN allows you to select different outbound WAN links based on performance SLAs. It is important that BGP neighbors are aware of these settings, and changes to them.

BGP can adapt to changes in SD-WAN link SLAs in the following ways:

- Applying different route-maps based on the SD-WAN's health checks. For example, different BGP community strings can be advertised to BGP neighbors when SLAs are not met.
- Traffic can be selectively forwarded based on the active BGP neighbor. If the SD-WAN service's role matches the active SD-WAN neighbor, the service is enabled. If there is no match, then the service is disabled.

Example

In this topology, a branch FortiGate has two SD-WAN gateways serving as the primary and secondary gateways. The gateways reside in different datacenters, but have a full mesh network between them.



This example shows how route-maps and service rules are selected based on performance SLAs and the member that is currently active. Traffic flows through the primary gateway unless the neighbor's health check is outside of its SLA. If that happens, traffic routes to the secondary gateway.

BGP NBR1 is the primary neighbor and BGP NBR2 is the secondary neighbor.

The branch FortiGate's wan1 and wan2 interfaces are members of the SD-WAN. When the SD-WAN neighbor status is primary, it will advertise community *20:1* to BGP NBR1 and *20:5* to BGP NBR2. When the SD-WAN neighbor status is secondary, it will advertise *20:5* to BGP NBR1 and *20:2* to BGP NBR2.

Only one of the primary or secondary neighbors can be active at one time. The SD-WAN neighbor status is used to decide which neighbor is selected:

- **Primary**: The primary neighbor takes precedence if its SLAs are met.
- **Secondary**: If the primary neighbor's SLAs are not met, the secondary neighbor becomes active if its SLAs are met.
- **Standalone**: If neither the primary or secondary neighbor's SLAs are met, the SD-WAN neighbor status becomes standalone.

Route map

SD-WAN is configured to let BGP advertise different communities when the SLA status changes. When the SLA is missed, it triggers BGP to advertise a different community to its BGP neighbor based on its route-map. The BGP neighbors can use the received community string to select the best path to reach the branch.

To configure BGP route-maps and neighbors:

1. Configure an access for the routes to be matched:

```
config router access-list
    edit "net192"
        config rule
            edit 1
                set prefix 192.168.20.0 255.255.255.0
            next
        end
    next
end
```

2. Configure the primary neighbor's preferred route-map:

```
config router route-map
    edit "comm1"
        config rule
            edit 1
                set match-ip-address "net192"
                set set-community "20:1"
            next
        end
    next
end
```

3. Configure the secondary neighbor's preferred route-map:

```
config router route-map
    edit "comm2"
        config rule
            edit 1
                set match-ip-address "net192"
                set set-community "20:2"
            next
        end
    next
end
```

4. Configure the failed route-map:

```
config router route-map
    edit "comm5"
        config rule
            edit 1
                set match-ip-address "net192"
                set set-community "20:5"
            next
        end
    next
end
```

5. Configure BGP neighbors:

```
config router bgp
    set as 65412
    set router-id 1.1.1.1
    set ibgp-multipath enable
    config neighbor
        edit "10.100.1.1"
            set soft-reconfiguration enable
            set remote-as 20
            set route-map-out "comm5"
            set route-map-out-preferable "comm1"
        next
        edit "10.100.1.5"
            set soft-reconfiguration enable
            set remote-as 20
            set route-map-out "comm5"
            set route-map-out-preferable "comm2"
        next
    end
end
```

When SLAs are met, `route-map-out-preferable` is used. When SLAs are missed, `route-map-out` is used.

To configure SD-WAN:

1. Configure the SD-WAN members:

```
config system sdwan
    set status enable
    config members
        edit 1
            set interface "port1"
        next
        edit 2
            set interface "port2"
        next
    end
end
```

2. Configure health checks for each member:

```
config system sdwan
    config health-check
        edit "ping"
            set server "10.100.2.22"
            set members 1
            config sla
                edit 1
                    set link-cost-factor packet-loss
                    set packetloss-threshold 1
                next
            end
        next
```

```
        edit "ping2"
            set server "10.100.2.23"
            set members 2
            config sla
                edit 1
                    set link-cost-factor packet-loss
                    set packetloss-threshold 1
                next
            end
        next
    end
end
```

3. Configure the SD-WAN neighbors and assign them a role and the health checks used to determine if the neighbor meets the SLA:

   SD-WAN neighbors can only be configured in the CLI.

```
config system sdwan
    config neighbor
        edit "10.100.1.1"
            set member 1
            set role primary
            set health-check "ping"
            set sla-id 1
        next
        edit "10.100.1.5"
            set member 2
            set role secondary
            set health-check "ping2"
            set sla-id 1
        next
    end
end
```

Service rules

Create SD-WAN service rules to direct traffic to the primary neighbor when its SLAs are met, and to the secondary neighbor when the primary neighbor's SLAs are missed.

To configure the SD-WAN service rules:

```
config system sdwan
    config service
        edit 1
            set name "Primary-Out"
            set role primary
            set dst "all"
            set src "all"
            set priority-members 1
        next
        edit 2
```

```
            set name "Secondary-Out"
            set role secondary
            set dst "all"
            set src "all"
            set priority-members 2
        next
    end
end
```

If neither the primary nor secondary neighbors are active, the SD-WAN neighbor status becomes standalone. Only service rules with `standalone-action` enabled will continue to pass traffic. This option is disabled by default.

Verification

To verify when the primary neighbor is passing traffic:

1. Verify the health check status:

```
FortiGate-Branch # diagnose sys sdwan health-check
Health Check(ping):
Seq(1 port1): state(alive), packet-loss(0.000%) latency(0.569), jitter(0.061) sla_map=0x1
Health Check(ping2):
Seq(2 port2): state(alive), packet-loss(0.000%) latency(3.916), jitter(2.373) sla_map=0x1
```

2. Verify SD-WAN neighbor status:

```
FortiGate-Branch # diagnose sys sdwan neighbor
SD-WAN neighbor status: hold-down(disable), hold-down-time(0), hold_boot_time(0)
        Selected role(primary) last_secondary_select_time/current_time in seconds 0/572
Neighbor(10.100.1.1): member(1) role(primary)
        Health-check(ping:1)  sla-pass selected alive
Neighbor(10.100.1.5): member(2) role(secondary)
        Health-check(ping2:1)  sla-pass alive
```

3. Verify service rules status:

```
FortiGate-Branch # diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x0
  Gen(3), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
  Service role: primary
  Members:
    1: Seq_num(1 port1), alive, selected
  Src address:
        0.0.0.0-255.255.255.255

  Dst address:
        0.0.0.0-255.255.255.255

Service(2): Address Mode(IPV4) flags=0x0
```

```
Gen(6), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
Service role: secondary, disabled by unselected.
Members:
  1: Seq_num(2 port2), alive, selected
Src address:
      0.0.0.0-255.255.255.255

Dst address:
      0.0.0.0-255.255.255.255
```

4. Verify neighbor routers:
   a. Primary neighbor router:

```
FGT-NBR1 # get router info bgp network 192.168.20.0
BGP routing table entry for 192.168.20.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  64512
    10.100.1.2 from 10.100.1.2 (192.168.122.98)
      Origin IGP metric 0, localpref 100, valid, external, best
      Community: 20:1
      Last update: Thu Apr 30 13:41:40 2020
```

   b. Secondary neighbor router:

```
FGT-NBR2 # get router info bgp network 192.168.20.0
VRF 0 BGP routing table entry for 192.168.20.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  64512
    10.100.1.6 from 10.100.1.6 (192.168.122.98)
      Origin IGP metric 0, localpref 100, valid, external, best
      Community: 20:5
      Last update: Thu Apr 30 13:41:39 2020
```

To verify when the secondary neighbor is passing traffic:

1. Verify the health check status:

```
FortiGate-Branch # diagnose sys sdwan health-check
Health Check(ping):
Seq(1 port1): state(dead), packet-loss(54.000%) sla_map=0x0
Health Check(ping2):
Seq(2 port2): state(alive), packet-loss(0.000%) latency(4.339), jitter(3.701) sla_map=0x1
```

2. Verify SD-WAN neighbor status:

```
FortiGate-Branch # diagnose sys sdwan neighbor
SD-WAN neighbor status: hold-down(disable), hold-down-time(0), hold_boot_time(0)
        Selected role(secondary) last_secondary_select_time/current_time in seconds 936/936
Neighbor(10.100.1.1): member(1) role(primary)
```

```
        Health-check(ping:1)  sla-fail dead
Neighbor(10.100.1.5): member(2) role(secondary)
        Health-check(ping2:1)  sla-pass selected alive
```

3. Verify service rules status:

```
FortiGate-Branch # diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x0
  Gen(4), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
  Service role: primary, disabled by unselected.
  Members:
    1: Seq_num(1 port1), alive, selected
  Src address:
        0.0.0.0-255.255.255.255

  Dst address:
        0.0.0.0-255.255.255.255

Service(2): Address Mode(IPV4) flags=0x0
  Gen(7), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
  Service role: secondary
  Members:
    1: Seq_num(2 port2), alive, selected
  Src address:
        0.0.0.0-255.255.255.255

  Dst address:
        0.0.0.0-255.255.255.255
```

4. Verify neighbor routers:
   a. Primary neighbor router:

```
FGT-NBR1 # get router info bgp network 192.168.20.0
BGP routing table entry for 192.168.20.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  64512
    10.100.1.2 from 10.100.1.2 (192.168.122.98)
      Origin IGP metric 0, localpref 100, valid, external, best
      Community: 20:5
      Last update: Thu Apr 30 15:41:58 2020
```

   b. Secondary neighbor router:

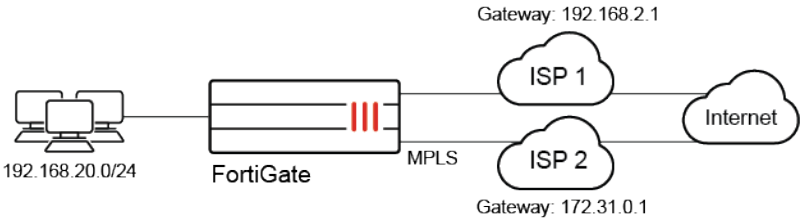
```
FGT-NBR2 # get router info bgp network 192.168.20.0
VRF 0 BGP routing table entry for 192.168.20.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  64512
    10.100.1.6 from 10.100.1.6 (192.168.122.98)
```

```
      Origin IGP metric 0, localpref 100, valid, external, best
      Community: 20:2
      Last update: Thu Apr 30 15:42:07 2020
```

Applying BGP route-map to multiple BGP neighbors

Controlling traffic with BGP route mapping and service rules explained how BGP can apply different route-maps to the primary and secondary SD-WAN neighbors based on SLA health checks.

In this example, SD-WAN neighbors that are not bound to primary and secondary roles are configured.



The FortiGate has multiple SD-WAN links and has formed BGP neighbors with both ISPs.

ISP1 is used primarily for outbound traffic, and has an SD-WAN service rule using the lowest cost algorithm applied to it. When SLAs for ISP1 are not met, it will fail over to the MPLS line.

Inbound traffic is allowed by both WAN links, with each WAN advertising a community string when SLAs are met. When SLAs are not met, the WAN links advertise a different community string.

This example uses two SD-WAN links. The topology can be expanded to include more links as needed.

To configure BGP route-maps and neighbors:

1. Configure an access list for routes to be matched:

```
config router access-list
    edit "net192"
        config rule
            edit 1
                set prefix 192.168.20.0 255.255.255.0
            next
        end
    next
end
```

2. Configure route-maps for neighbor ISP1:

```
config router route-map
    edit "comm1"
        config rule
            edit 1
                set match-ip-address "net192"
                set set-community "64511:1"
            next
        end
```

```
    next
    edit "comm-fail1"
        config rule
            edit 1
                set match-ip-address "net192"
                set set-community "64511:5"
            next
        end
    next
end
```

3. Configure route-maps for neighbor ISP2:

```
config router route-map
    edit "comm2"
        config rule
            edit 1
                set match-ip-address "net192"
                set set-community "64522:1"
            next
        end
    next
    edit "comm-fail2"
        config rule
            edit 1
                set match-ip-address "net192"
                set set-community "64522:5"
            next
        end
    next
end
```

4. Configure the BGP neighbors:

```
config router bgp
    set as 64512
    set keepalive-timer 1
    set holdtime-timer 3
    config neighbor
        edit "192.168.2.1"
            set soft-reconfiguration enable
            set remote-as 64511
            set route-map-out "comm-fail1"
            set route-map-out-preferable "comm1"
        next
        edit "172.31.0.1"
            set soft-reconfiguration enable
            set remote-as 64522
            set route-map-out "comm-fail2"
            set route-map-out-preferable "comm2"
        next
    end
```

```
    config network
        edit 1
            set prefix 192.168.20.0 255.255.255.0
        next
    end
end
```

When SLAs are met, `route-map-out-preferable` is used. When SLAs are missed, `route-map-out` is used.

To configure SD-WAN:

1. Configure the SD-WAN members:

```
config system sdwan
    set status enable
    config members
        edit 1
            set interface "port1"
            set gateway 192.168.2.1
        next
        edit 2
            set interface "MPLS"
            set cost 20
        next
    end
end
```

2. Configure the health checks that must be met:

```
config system sdwan
    config health-check
        edit "pingserver"
            set server "8.8.8.8"
            set members 2 1
            config sla
                edit 1
                    set link-cost-factor packet-loss
                    set packetloss-threshold 2
                next
            end
        next
    end
end
```

3. Configure the SD-WAN neighbors and assign them a role and the health checks used to determine if the neighbor meets the SLA:

   When no role is defined, the default role, `standalone`, is used.

```
config system sdwan
    config neighbor
        edit "192.168.2.1"
            set member 1
```

```
            set health-check "pingserver"
            set sla-id 1
        next
        edit "172.31.0.1"
            set member 2
            set health-check "pingserver"
            set sla-id 1
        next
    end
end
```

Service rules

Create SD-WAN service rules to direct traffic to the SD-WAN links based on the lowest cost algorithm The same SLA health check and criteria that are used for the SD-WAN neighbor are used for this SD-WAN service rule.

When no roles are defined in the service rule, the default role, `standalone`, is used.

To configure the SD-WAN service rule:

```
config system sdwan
    config service
        edit 1
            set name "OutboundAll"
            set mode sla
            set dst "all"
            set src "all"
            config sla
                edit "pingserver"
                    set id 1
                next
            end
            set priority-members 1 2
        next
    end
end
```

Verification

To verify that when both SLAs are met, port1 is selected due to its lower cost:

1. Verify the health check status:

```
FortiGate-Branch # diagnose sys sdwan health-check
Health Check(pingserver):
Seq(2 MPLS): state(alive), packet-loss(0.000%) latency(24.709), jitter(14.996) sla_map=0x1
Seq(1 port1): state(alive), packet-loss(0.000%) latency(28.771), jitter(14.840) sla_map=0x1
```

2. Verify SD-WAN neighbor status:

```
FortiGate-Branch # diagnose sys sdwan neighbor
Neighbor(192.168.2.1): member(1) role(standalone)
        Health-check(pingserver:1)  sla-pass selected alive
Neighbor(172.31.0.1): member(2) role(standalone)
        Health-check(pingserver:1)  sla-pass selected alive
```

3. Verify service rules status:

   Because the service role is `standalone`, it matches both neighbors. The mode (SLA) determines that port1 is lower cost.

```
FortiGate-Branch # diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x0
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Service role: standalone
  Members:
    1: Seq_num(1 port1), alive, sla(0x1), cfg_order(0), cost(0), selected
    2: Seq_num(2 MPLS), alive, sla(0x1), cfg_order(1), cost(20), selected
  Src address:
        0.0.0.0-255.255.255.255

  Dst address:
        0.0.0.0-255.255.255.255
```

4. Verify neighbor routers:

   a. Primary neighbor router:

```
FGT-NBR1 # get router info bgp network 192.168.20.0
BGP routing table entry for 192.168.20.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  64512
    192.168.2.5 from 192.168.2.5 (192.168.122.98)
      Origin IGP metric 0, localpref 100, valid, external, best
      Community: 64511:1
      Last update: Thu Apr 30 23:59:05 2020
```

   b. Secondary neighbor router:

```
FGT-NBR2 # get router info bgp network 192.168.20.0
VRF 0 BGP routing table entry for 192.168.20.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  64512
    172.31.0.2 from 172.31.0.2 (192.168.122.98)
      Origin IGP metric 0, localpref 100, valid, external, best
      Community: 64522:1
      Last update: Fri May  1 00:11:28 2020
```

To verify that when neighbor ISP1 misses SLAs, MPLS is selected and BGP advertises a different community string for ISP1:

1. Verify the health check status:

```
FortiGate-Branch # diagnose sys sdwan health-check
Health Check(pingserver):
Seq(2 MPLS): state(alive), packet-loss(0.000%) latency(25.637), jitter(17.820) sla_map=0x1
Seq(1 port1): state(dead), packet-loss(16.000%) sla_map=0x0
```

2. Verify SD-WAN neighbor status:

```
FortiGate-Branch # diagnose sys sdwan neighbor
Neighbor(192.168.2.1): member(1) role(standalone)
        Health-check(pingserver:1)  sla-fail dead
Neighbor(172.31.0.1): member(2) role(standalone)
        Health-check(pingserver:1)  sla-pass selected alive
```

3. Verify service rules status:
   As SLA failed for neighbor ISP1, MPLS is preferred.

```
FortiGate-Branch # diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x0
  Gen(3), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Service role: standalone
  Members:
    1: Seq_num(2 MPLS), alive, sla(0x1), cfg_order(1), cost(20), selected
    2: Seq_num(1 port1), dead, sla(0x0), cfg_order(0), cost(0)
  Src address:
        0.0.0.0-255.255.255.255

  Dst address:
        0.0.0.0-255.255.255.255
```

4. Verify neighbor routers:
   The community received on ISP1 is updated.
   a. Primary neighbor router:

```
FGT-NBR1 # get router info bgp network 192.168.20.0
BGP routing table entry for 192.168.20.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  64512
    192.168.2.5 from 192.168.2.5 (192.168.122.98)
      Origin IGP metric 0, localpref 100, valid, external, best
      Community: 64511:5
      Last update: Fri May  1 00:33:26 2020
```

   b. Secondary neighbor router:

```
FGT-NBR2 # get router info bgp network 192.168.20.0
VRF 0 BGP routing table entry for 192.168.20.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  64512
    172.31.0.2 from 172.31.0.2 (192.168.122.98)
      Origin IGP metric 0, localpref 100, valid, external, best
      Community: 64522:1
      Last update: Fri May  1 00:22:42 2020
```

Using multiple members per SD-WAN neighbor configuration

SD-WAN BGP neighbor configurations are used to define the SLA health check in which an SD-WAN member must meet to qualify as being up. When the SD-WAN member meets the SLA threshold, the FortiGate will apply the route map defined in the BGP neighbor's `route-map-out-preferable` option. If the SD-WAN member fails to meet the SLA, the FortiGate will apply the route map defined in the BGP neighbor's `route-map-out` option instead. This allows the FortiGate to advertise the health of the SD-WAN member to its BGP neighbor by advertising different community strings based on its SLA status.

For more information, refer to the following BGP examples: Controlling traffic with BGP route mapping and service rules on page 1440 and Applying BGP route-map to multiple BGP neighbors on page 1448.

Selecting multiple SD-WAN members allows the SD-WAN neighbor feature to support topologies where there are multiple SD-WAN overlays and/or underlays to a neighbor. The `minimum-sla-meet-members` option is used to configure the minimum number of members that must be in an SLA per neighbor for the preferable route map to be used.

```
config system sdwan
    config neighbor
        edit <ip>
            set member {<seq-num_1>} [<seq-num_2>] ... [<seq-num_n>]
            set minimum-sla-meet-members <integer>
        next
    end
end
```

Option	Description
`member {<seq-num_1>} [<seq-num_2>] ... [<seq-num_n>]`	Enter the member sequence number list. Multiple members can be defined.
`minimum-sla-meet-members <integer>`	Set the minimum number of members that meet SLA when the neighbor is preferred (1 - 255, default = 1). • If the number of in SLA members is less than the `minimum-sla-meet-members` value, the default route map will be used. • If the number of in SLA members is equal or larger than the `minimum-sla-meet-members` value, the preferable route map will be used.

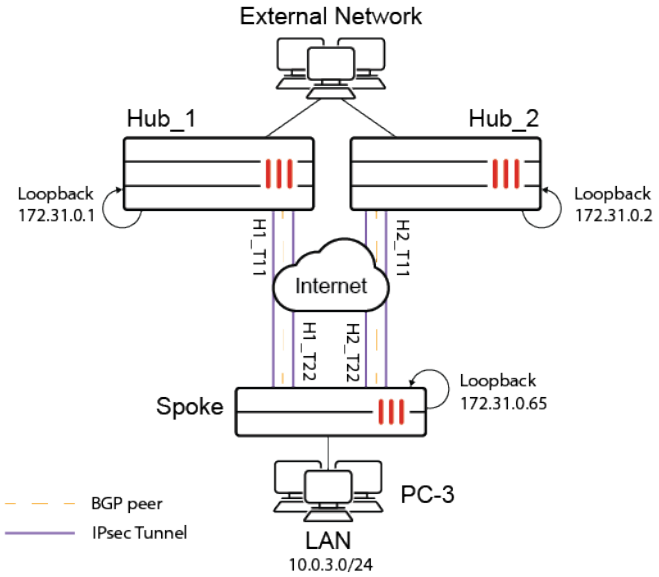
Example

In the following example, the spoke FortiGate has four tunnels: two tunnels to Hub_1 and two tunnels to Hub_2. The spoke has two BGP neighbors: one to Hub_1 and one to Hub-2. BGP neighbors are established on loopback IPs.

The SD-WAN neighbor plus `route-map-out-preferable`configuration is deployed on the spoke to achieve the following:

- If any tunnel to Hub_1 or Hub_2 is in SLA, the preferable route map will be applied on the BGP neighbor to Hub_1 or Hub_2.
- If both tunnels to Hub_1 or Hub_2 are out of SLA, the default route map will be applied on the BGP neighbor to Hub_1 or Hub_2.

The preferable route map and default route map are used to set different custom BGP communities as the spoke advertises its LAN routes to the hub. Each hub can translate communities into different BGP MED or AS prepends and signal them to the external peers to manipulate inbound traffic, thereby routing traffic to the spoke only when the SLAs are met on at least one of two VPN overlays. In this example, community string 10:1 signals to the neighbor that SLAs are met, and 10:2 signals that SLAs are not met.



To configure the BGP route maps and neighbors:

1. Configure an access list of prefixes to be matched:

```
config router access-list
    edit "net10"
        config rule
            edit 1
                set prefix 10.0.3.0 255.255.255.0
            next
        end
    next
end
```

2. Configure route maps for neighbors in SLA (preferable) and out of SLA (default):

```
config router route-map
    edit "in_sla"
        config rule
            edit 1
                set match-ip-address "net10"
                set set-community "10:1"
            next
        end
    next
    edit "out_sla"
        config rule
            edit 1
                set match-ip-address "net10"
                set set-community "10:2"
            next
        end
    next
end
```

3. Configure the BGP neighbors:

```
config router bgp
    set router-id 172.31.0.65
    config neighbor
        edit "172.31.0.1"
            set route-map-out "out_sla"
            set route-map-out-preferable "in_sla"
            set update-source "Loopback0"
        next
        edit "172.31.0.2"
            set route-map-out "out_sla"
            set route-map-out-preferable "in_sla"
            set update-source "Loopback0"
        next
    end
    config network
        edit 1
            set prefix 10.0.3.0 255.255.255.0
        next
    end
end
```

To configure SD-WAN:

1. Configure the SD-WAN members:

```
config system sdwan
    set status enable
    config members
        edit 1
            set interface "H1_T11"
            set source 172.31.0.65
```

```
        next
        edit 4
            set interface "H1_T22"
            set source 172.31.0.65
        next
        edit 6
            set interface "H2_T11"
            set source 172.31.0.65
        next
        edit 9
            set interface "H2_T22"
            set source 172.31.0.65
        next
    end
end
```

2. Configure the health check that must be met:

```
config system sdwan
    config health-check
        edit "HUB"
            set server "172.31.100.100"
            set members 0
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 100
                next
            end
        next
    end
end
```

3. Configure the SD-WAN neighbors:

```
config system sdwan
    config neighbor
        edit "172.31.0.1"
            set member 1 4
            set health-check "HUB"
            set sla-id 1
            set minimum-sla-meet-members 1
        next
        edit "172.31.0.2"
            set member 6 9
            set health-check "HUB"
            set sla-id 1
            set minimum-sla-meet-members 1
        next
    end
end
```

To verify that when two members to Hub_1/Hub_2 are in SLA, the preferable route map is be applied on BGP neighbors to Hub_1/Hub_2:

```
Branch1_A_FGT (root) # diagnose sys sdwan health-check
Health Check(HUB):
Seq(1 H1_T11): state(alive), packet-loss(0.000%) latency(0.209), jitter(0.017), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
Seq(4 H1_T22): state(alive), packet-loss(0.000%) latency(0.171), jitter(0.004), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x1
Seq(6 H2_T11): state(alive), packet-loss(0.000%) latency(0.175), jitter(0.014), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
Seq(9 H2_T22): state(alive), packet-loss(0.000%) latency(0.176), jitter(0.019), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x1
```

```
# diagnose sys sdwan neighbor
Neighbor(172.31.0.1): member(1 4 )role(standalone)
        Health-check(HUB:1)  sla-pass selected alive
Neighbor(172.31.0.2): member(6 9 )role(standalone)
        Health-check(HUB:1)  sla-pass selected alive
```

On Hub_1 and Hub_2, the expected communities have been attached into the spoke's LAN route:

```
Hub_1_FGT (root) # get router info bgp network 10.0.3.0/24
VRF 0 BGP routing table entry for 10.0.3.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local, (Received from a RR-client)
    172.31.0.65 from 172.31.0.65 (172.31.0.65)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Community: 10:1
      Last update: Wed Dec 29 22:38:29 2021
```

```
Hub_2_FGT (root) # get router info bgp network 10.0.3.0/24
VRF 0 BGP routing table entry for 10.0.3.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local, (Received from a RR-client)
    172.31.0.65 from 172.31.0.65 (172.31.0.65)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Community: 10:1
      Last update: Wed Dec 29 22:43:10 2021
```

If one member for each neighbor becomes out of SLA, the preferable route map is still applied:

```
Branch1_A_FGT (root) # diagnose sys sdwan health-check
Health Check(HUB):
Seq(1 H1_T11): state(alive), packet-loss(0.000%) latency(120.207), jitter(0.018), mos(4.338),
bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x0
Seq(4 H1_T22): state(alive), packet-loss(0.000%) latency(0.182), jitter(0.008), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x1
```

```
Seq(6 H2_T11): state(alive), packet-loss(0.000%) latency(120.102), jitter(0.009), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x0
Seq(9 H2_T22): state(alive), packet-loss(0.000%) latency(0.176), jitter(0.009), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x1
```

```
# diagnose sys sdwan neighbor
Neighbor(172.31.0.1): member(1 4 )role(standalone)
        Health-check(HUB:1)  sla-pass selected alive
Neighbor(172.31.0.2): member(6 9 )role(standalone)
        Health-check(HUB:1)  sla-pass selected alive
```

```
Hub_1_FGT (root) # get router info bgp network 10.0.3.0/24
VRF 0 BGP routing table entry for 10.0.3.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local, (Received from a RR-client)
    172.31.0.65 from 172.31.0.65 (172.31.0.65)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Community: 10:1
      Last update: Thu Dec 30 10:44:47 2021
```

```
Hub_2_FGT (root) # get router info bgp network 10.0.3.0/24
VRF 0 BGP routing table entry for 10.0.3.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local, (Received from a RR-client)
    172.31.0.65 from 172.31.0.65 (172.31.0.65)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Community: 10:1
      Last update: Wed Dec 29 22:43:10 2021
```

If both members for Hub_1 become out of SLA, the default route map is applied:

```
Branch1_A_FGT (root) # diagnose sys sdwan health-check
Health Check(HUB):
Seq(1 H1_T11): state(alive), packet-loss(0.000%) latency(120.194), jitter(0.018), mos(4.338),
bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x0
Seq(4 H1_T22): state(alive), packet-loss(0.000%) latency(120.167), jitter(0.006), mos(4.338),
bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x0
Seq(6 H2_T11): state(alive), packet-loss(0.000%) latency(120.180), jitter(0.012), mos(4.338),
bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x0
Seq(9 H2_T22): state(alive), packet-loss(0.000%) latency(0.170), jitter(0.005), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999997), bandwidth-bi(1999996) sla_map=0x1
```

```
# diagnose sys sdwan  neighbor
Neighbor(172.31.0.1): member(1 4 )role(standalone)
        Health-check(HUB:1)  sla-fail alive
Neighbor(172.31.0.2): member(6 9 )role(standalone)
        Health-check(HUB:1)  sla-pass selected alive
```

```
Hub_1_FGT (root) # get router info bgp network 10.0.3.0/24
VRF 0 BGP routing table entry for 10.0.3.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local, (Received from a RR-client)
    172.31.0.65 from 172.31.0.65 (172.31.0.65)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Community: 10:2
      Last update: Thu Dec 30 10:57:33 2021
```

```
Hub_2_FGT (root) # get router info bgp network 10.0.3.0/24
VRF 0 BGP routing table entry for 10.0.3.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local, (Received from a RR-client)
    172.31.0.65 from 172.31.0.65 (172.31.0.65)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Community: 10:1
      Last update: Wed Dec 29 22:43:10 2021
```

Allow SD-WAN hubs to suppress BGP routes when all links to a spoke are down

The feature enables hubs to detect when a spoke is dead (no SLA probes over a configurable duration) and suppress routes to that spoke. A BGP route-map-out is used to match this suppression status, and adjusts the multiple exit discriminator (MED) to inform BGP peers of the hub to direct traffic to the spoke through another hub.

```
config system sdwan
    config health-check
        edit <name>
            set update-bgp-route {enable | disable}
        next
    end
end
```

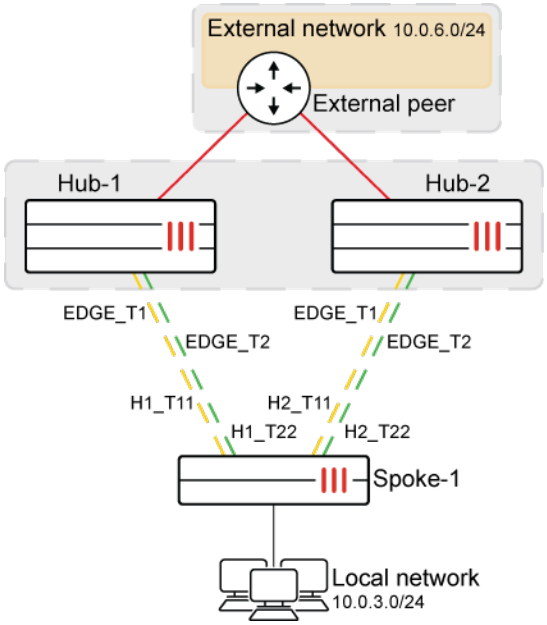
```
config router route-map
    edit <name>
        config rule
            edit 1
                set match-suppress {enable | disable}
            next
        end
    next
end
```

```
config router bgp
    config neighbor
        edit <BGP peer IP>
            set attribute-unchanged med
            set route-map-out "suppress_dead_spoke"
        next
    end
end
```

Example

In this example:

- Spoke-1 has two tunnels, H1_T11 and H1_T22, to Hub-1 and traffic from the external network to spoke-1 normally goes through Hub-1.
- Spoke-1 has two other tunnels, H2_T11 and H2_T22, to Hub-2 and, if all links and tunnels between Spoke-1 and Hub-1 fail, it is expected that traffic from the external network to Spoke-1 will switchover to Hub-2 by advertising Spoke-1's routes with a higher MED from Hub-1 to the external network.



To configure Spoke-1:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
        edit "overlay"
        next
        ...
    end
    config members
```

```
        ...
        edit 4
            set interface "H1_T11"
            set zone "overlay"
            set source 172.31.0.65
            ...
        next
        edit 5
            set interface "H1_T22"
            set zone "overlay"
            set source 172.31.0.65
            ...
        next
        ...
        edit 7
            set interface "H2_T11"
            set zone "overlay"
            set source 172.31.0.65
            ...
        next
        edit 8
            set interface "H2_T22"
            set zone "overlay"
            set source 172.31.0.65
            ...
        next
        ...
    end
    config health-check
        edit "HUB"
            set server "172.31.100.100"
            set embed-measured-health enable
            set sla-id-redistribute 1
            ...
            set members 4 5 7 8
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 100
                next
            end
        next
    end
end
```

To configure Hub-1:

1. Configure SD-WAN:

   As long as one probe packet on a member is not received, the member will be considered as *dead*. If all members to Spoke-1 are *dead*, then Spoke-1's routes will be flagged as *suppressed*. `failtime` can be used to control convergence time.

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
        ...
        edit "overlay"
        next
    end
    config members
        edit 1
            set interface "EDGE_T1"
            set zone "overlay"
        next
        edit 2
            set interface "EDGE_T2"
            set zone "overlay"
        next
        ...
    end
    config health-check
        edit "rmt"
            set detect-mode remote
            set remote-probe-timeout 1000
            set failtime 1
            set recoverytime 1
            set update-bgp-route enable
            set sla-id-redistribute 1
            set members 1 2
            config sla
                edit 1
                    set link-cost-factor remote
                    ...
                next
            end
        next
    end
end
```

2. Configure route maps:
   Suppressed routes are set with an MED of 999, and normal routes with an MED of 10.

```
config router route-map
    edit "ACTION_SUPPRESS"
        config rule
            edit 1
                set match-suppress enable
                set set-metric 999
            next
            edit 2
                set set-metric 10
            next
```

```
        end
    next
end
```

3. Configure BGP settings:

```
config router bgp
   ...
    config neighbor
        edit "172.31.0.129"
            ...
            set attribute-unchanged med
            ...
            set route-map-out "ACTION_SUPPRESS"
        next
    end
end
```

To configure Hub-2:

1. Configure SD-WAN:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
        ...
        edit "overlay"
        next
    end
    config members
        edit 1
            set interface "EDGE_T1"
            set zone "overlay"
        next
        edit 2
            set interface "EDGE_T2"
            set zone "overlay"
        next
        ...
    end
    config health-check
        edit "rmt"
            set detect-mode remote
            set remote-probe-timeout 1000
            set failtime 1
            set recoverytime 1
            set update-bgp-route enable
            set sla-id-redistribute 1
            set members 1 2
            config sla
```

```
                edit 1
                    set link-cost-factor remote
                    ...
                next
            end
        next
    end
end
```

2. Configure route maps:
   Suppressed routes are set with an MED of 999, and normal routes with an MED of 20.

```
config router route-map
    edit "ACTION_SUPPRESS"
        config rule
            edit 1
                set match-suppress enable
                set set-metric 999
            next
            edit 2
                set set-metric 20
            next
        end
    next
end
```

3. Configure BGP settings:

```
config router bgp
   ...
    config neighbor
        edit "172.31.0.129"
            ...
            set attribute-unchanged med
            ...
            set route-map-out "ACTION_SUPPRESS"
        next
    end
end
```

To test the configuration:

1. The SLA status detected by the remote health-check on Hub-1 indicates that the two connections between Hub-1 and Spoke-1 are alive:

```
# diagnose sys sdwan  health-check remote
Remote Health Check: rmt(2)
  Passive remote statistics of EDGE_T2(46):
EDGE_T2_0(10.0.0.14): timestamp=10-15 15:59:46.694, src=172.31.0.65, latency=0.233,
jitter=0.010, pktloss=0.000%, mos=4.404, SLA id=1(pass), rmt_ver=2, rmt_sla=in, rmt_prio=150,
last_sla_change=10-15 15:06:36.610
Remote Health Check: rmt(1)
```

```
  Passive remote statistics of EDGE_T1(45):
EDGE_T1_0(172.31.0.65): timestamp=10-15 15:59:46.694, src=172.31.0.65, latency=0.276,
jitter=0.029, pktloss=0.000%, mos=4.404, SLA id=1(pass), rmt_ver=2, rmt_sla=in, rmt_prio=100,
last_sla_change=10-15 15:06:36.609

DC1_A_FGT (root) (Interim)# diagnose sys link-monitor dynamic-bgp-peers
172.31.0.65(0): alive-count=2, total=2, route-suppressed=0, last_route_change=10-15
15:06:33.999
```

2. The external peer receives Spoke-1's routes from Hub-1 and Hub-2.

   The route received from Hub-1 is set as MED 10 and preferred. The route received from Hub-2 is set as MED 20.

```
# get router info bgp network 10.0.3.0/24
VRF 0 BGP routing table entry for 10.0.3.0/24
Paths: (2 available, best #1, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
   172.31.0.2
  Advertised to peer-groups:
  "EDGE"
  Original VRF 0
  65001
    172.31.0.1 from 172.31.0.1 (172.31.0.1)

      Origin IGP distance 20 metric 10, localpref 100, valid, external, best
      Community: 10:2
      Last update: Wed Oct 15 15:06:52 2025

  Original VRF 0
  65001
    172.31.0.2 from 172.31.0.2 (172.31.0.2)
      Origin IGP distance 20 metric 20, localpref 100, valid, external
      Community: 10:2
      Last update: Wed Oct 15 15:00:54 2025
```

3. Send traffic from external network to Spoke-1's network. The traffic will go through Hub-1 and reach Spoke-1 on H1_T11:

```
# diagnose sniffer packet any 'host 10.0.6.1' 4
interfaces=[any]
filters=[host 10.0.6.1]
9.053176 H1_T11 in 10.0.6.1 -> 10.0.3.2: icmp: echo request
9.053584 port4 out 10.0.6.1 -> 10.0.3.2: icmp: echo request
9.053689 port4 in 10.0.3.2 -> 10.0.6.1: icmp: echo reply
9.053760 H1_T11 out 10.0.3.2 -> 10.0.6.1: icmp: echo reply
10.077224 H1_T11 in 10.0.6.1 -> 10.0.3.2: icmp: echo request
10.077354 port4 out 10.0.6.1 -> 10.0.3.2: icmp: echo request
10.077463 port4 in 10.0.3.2 -> 10.0.6.1: icmp: echo reply
10.077526 H1_T11 out 10.0.3.2 -> 10.0.6.1: icmp: echo reply
...
```

4. Disable H1_T11 on Spoke-1. Hub-1 will detect it and consider it as dead, as a result, traffic will switch to H1_T22.

   Once probe packet cannot be received on the member from Spoke-1, no statistics will be shown, and the `alive-count` is changed from `2` to `1`.

```
# diagnose sys sdwan  health-check remote
Remote Health Check: rmt(2)
  Passive remote statistics of EDGE_T2(46):
EDGE_T2_0(10.0.0.14): timestamp=10-15 16:14:56.136, src=172.31.0.65, latency=0.298,
jitter=0.037, pktloss=0.000%, mos=4.404, SLA id=1(pass), rmt_ver=2, rmt_sla=in, rmt_prio=150,
last_sla_change=10-15 15:06:36.610
Remote Health Check: rmt(1)
  Passive remote statistics of EDGE_T1(45):

DC1_A_FGT (root) (Interim)# diagnose sys link-monitor dynamic-bgp-peers
172.31.0.65(0): alive-count=1,total=1, route-suppressed=0, last_route_change=10-15
15:06:33.999
```

```
# diagnose sniffer packet any 'host 10.0.6.1' 4
interfaces=[any]
filters=[host 10.0.6.1]

......

541.094223 port4 in 10.0.3.2 -> 10.0.6.1: icmp: echo reply
541.094279 H1_T22 out 10.0.3.2 -> 10.0.6.1: icmp: echo reply
542.103485 H1_T22 in 10.0.6.1 -> 10.0.3.2: icmp: echo request
542.103614 port4 out 10.0.6.1 -> 10.0.3.2: icmp: echo request
542.103737 port4 in 10.0.3.2 -> 10.0.6.1: icmp: echo reply
542.103794 H1_T22 out 10.0.3.2 -> 10.0.6.1: icmp: echo reply
543.113468 H1_T22 in 10.0.6.1 -> 10.0.3.2: icmp: echo request
543.113602 port4 out 10.0.6.1 -> 10.0.3.2: icmp: echo request
```

5. Disable H1_T22 on Spoke-1 as well. All connections between Hub-1 and Spoke-1 are dead. At this point, Spoke-1's routes will be flagged as `suppressed` and set as `med 999`, then advertised to the external peer. As a result, traffic will switch over to Hub-2 and reach Spoke-1 on H2_T11.

   The `alive-count` is also changed from `1` to `0` and the route received from Hub-2 is set as med 20 and preferred.

```
# diagnose sys sdwan  health-check remote
Remote Health Check: rmt(2)
  Passive remote statistics of EDGE_T2(46):
Remote Health Check: rmt(1)
  Passive remote statistics of EDGE_T1(45):

DC1_A_FGT (root) (Interim)# diagnose sys link-monitor dynamic-bgp-peers
172.31.0.65(0): alive-count=0,total=1, route-suppressed=1, last_route_change=10-15
16:29:20.629

DC3_FGT (root) (Interim)# get router info bgp network 10.0.3.0/24
VRF 0 BGP routing table entry for 10.0.3.0/24
Paths: (2 available, best #2, table Default-IP-Routing-Table)
```

```
  Advertised to non peer-group peers:
   172.31.0.1
  Advertised to peer-groups:
  "EDGE"
  Original VRF 0
  65001
    172.31.0.1 from 172.31.0.1 (172.31.0.1)
      Origin IGP distance 20 metric 999, localpref 100, valid, external
      Community: 10:2
      Last update: Wed Oct 15 16:29:21 2025

  Original VRF 0
  65001
    172.31.0.2 from 172.31.0.2 (172.31.0.2)
      Origin IGP distance 20 metric 20, localpref 100, valid, external, best
      Community: 10:2
      Last update: Wed Oct 15 15:00:54 2025
```

```
# diagnose sniffer packet any 'host 10.0.6.1' 4
interfaces=[any]
filters=[host 10.0.6.1]

......

141.569312 H2_T11 in 10.0.6.1 -> 10.0.3.2: icmp: echo request
141.569454 port4 out 10.0.6.1 -> 10.0.3.2: icmp: echo request
141.569571 port4 in 10.0.3.2 -> 10.0.6.1: icmp: echo reply
141.569630 H2_T11 out 10.0.3.2 -> 10.0.6.1: icmp: echo reply
142.579272 H2_T11 in 10.0.6.1 -> 10.0.3.2: icmp: echo request
142.579410 port4 out 10.0.6.1 -> 10.0.3.2: icmp: echo request
142.579535 port4 in 10.0.3.2 -> 10.0.6.1: icmp: echo reply
142.579599 H2_T11 out 10.0.3.2 -> 10.0.6.1: icmp: echo reply
```

Local out traffic

Local out, or self-originating, traffic is traffic that originates from the FortiGate going to external servers and services. The traffic can be from Syslog, FortiAnalyzer logging, FortiGuard services, remote authentication, and others.

By default, local out traffic relies on routing table lookups to determine the egress interface that is used to initiate the connection. However, many types of local out traffic support selecting the egress interface based on SD-WAN or manually specified interfaces. When manually specifying the egress interface, ensure it is the actual physical interface through which traffic exits the FortiGate. In such cases, the source IP address may also be manually configured, even if it is not assigned to that interface.

Go to *Network* > *Local Out Routing* to configure the available types of local out traffic. Some types of traffic can only be configured in the CLI.

By default *Local Out Routing* is not visible in the GUI. Go to *System > Feature Visibility* to enable it. See Feature visibility on page 3952 for more information.

When VDOMs are enabled, the following entries are available on the local out routing page:

Global view	VDOM view
External Resources	**LDAP Servers**
AWS_IP_Blacklist	ldap
AWS_Malware_Hash	**Log**
Log	Log FortiAnalyzer Override Settings
Log FortiAnalyzer Setting	Log Syslogd Override Settings
Log FortiAnalyzer Cloud Setting	**RADIUS Servers**
FortiGate Cloud Log Settings	fac_radius_server
Log Syslogd Setting	**TACACS+**
System	TACACS
System DNS	
System FortiGuard	
System FortiSandbox	

If a service is disabled, it is grayed out. To enable it, select the service and click *Enable Service*. If a service is enabled, there is a *Local Out Setting* button in the gutter of that service's edit page to directly configure the local-out settings.

Examples

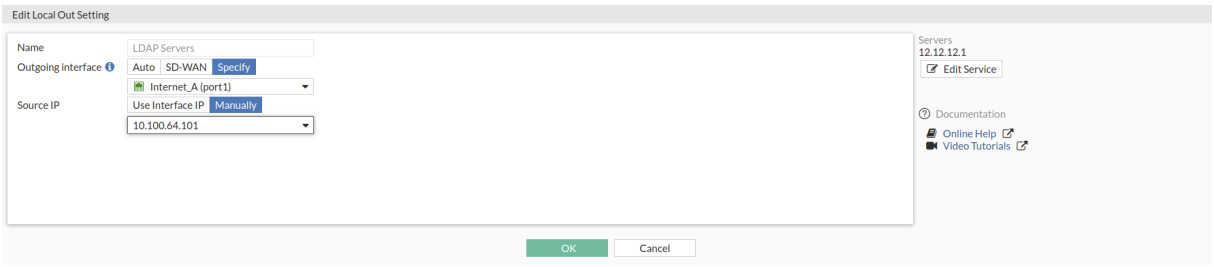
To configure DNS local-out routing:

1. Go to *Network > Local Out Routing* and double-click *System DNS*.
2. For *Outgoing interface*, select one of the following:

Auto	Select the outgoing interface automatically based on the routing table.
SD-WAN	Select the outgoing interface using the configured SD-WAN interfaces and rules.
Specify	Select the outgoing interface from the dropdown.

3. If *Specify* is selected, select a setting for *Source IP*:

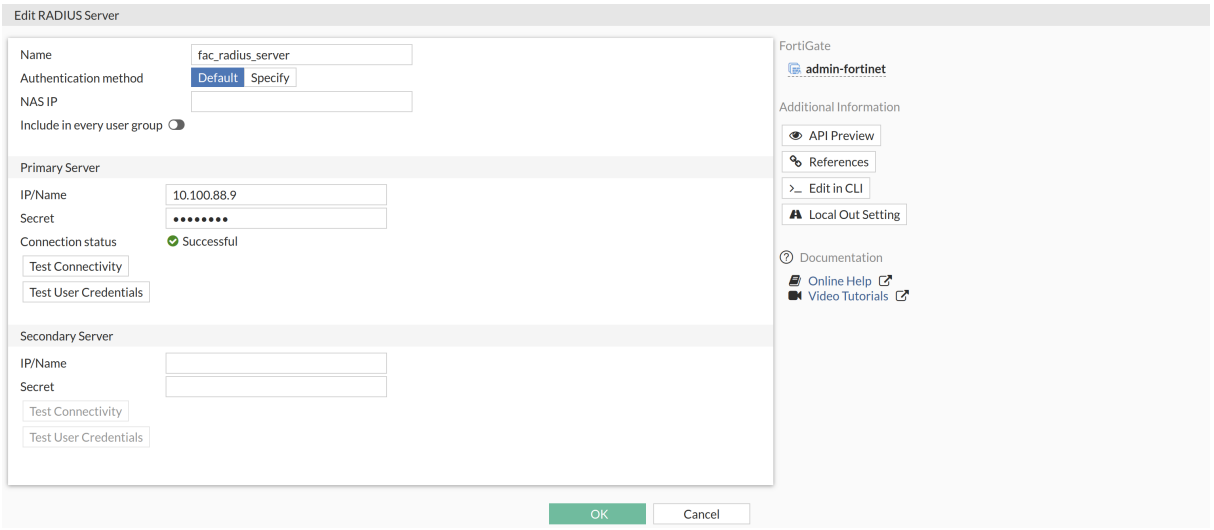
Use Interface IP	Use the primary IP, which cannot be configured by the user.
Manually	Selected an IP from the list, if the selected interface has multiple IPs configured.



4. Click *OK*.

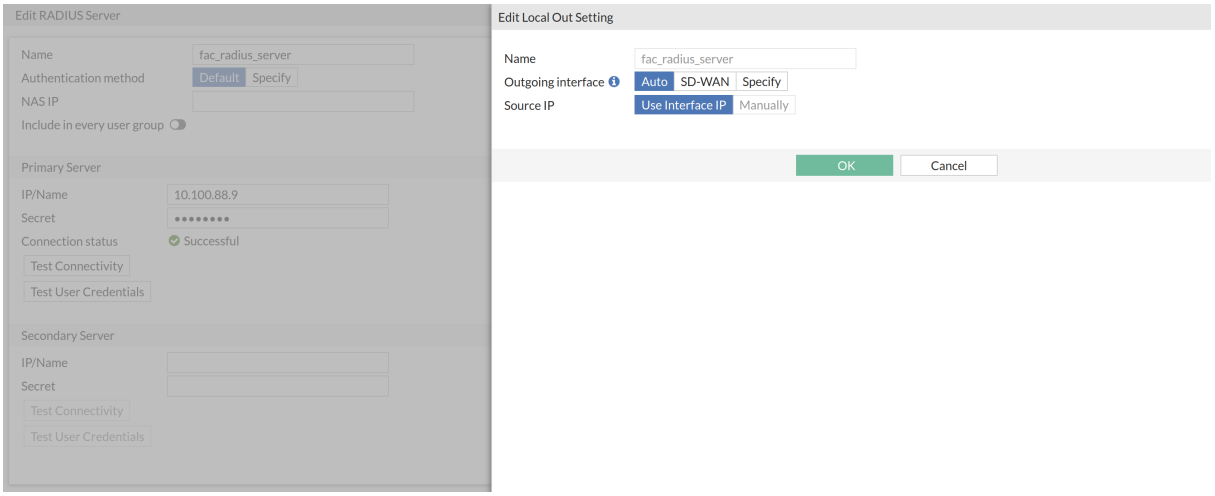
To edit local-out settings from a RADIUS server entry:

1. Go to *User & Authentication* > *RADIUS Servers* and double-click an entry to edit it.
2. Click *Local Out Setting*.



The *Edit Local Out Setting* pane opens.

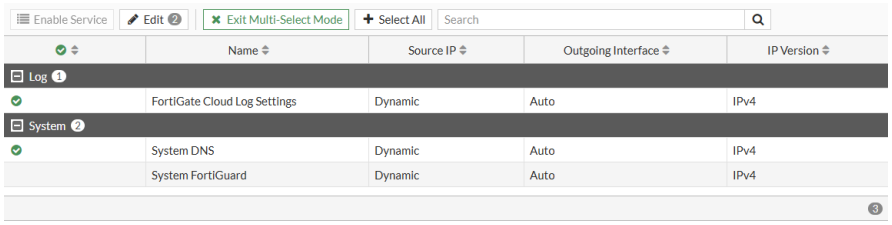
3. Configure the settings for *Outgoing interface* and *Source IP*.



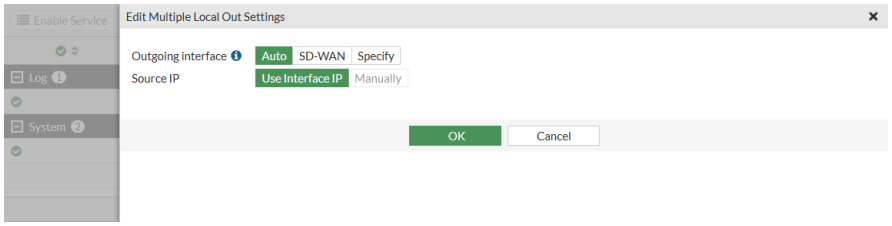
4. Click *OK*.

To edit multiple entries concurrently:

1. Go to *Network* > *Local Out Routing*.
2. If applicable, select *IPv4* or *IPv6*. *IPv4+IPv6* does not support multi-select.
3. Click *Multi-Select Mode*. All of the local out settings that can be edited concurrently are shown.
4. Select the specific entries, or click *Select All* to select all of the entries.



5. Click *Edit* and configure the local out settings as required.



6. Click *OK*.
7. Click *Exit Multi-Select Mode* to return to the normal view.

Configuring local out routing in the CLI

Some local out routing settings can only be configured using the CLI.

PING

IPv4 and IPv6 pings can be configured to use SD-WAN rules:

```
execute ping-options use-sdwan {yes | no}
execute ping6-options use-sd-wan {yes | no}
```

Traceroute

IPv4 traceroute can be configured to use SD-WAN rules:

```
execute traceroute-options use-sdwan {yes | no}
```

Central management

Central management traffic can use SD-WAN rules or a specific interface:

```
config system central-management
    set interface-select-method {auto | sdwan | specify}
```

```
    set interface <interface>
end
```

NTP server

NTP server traffic can use SD-WAN rules or a specific interface:

```
config system ntp
    config ntpserver
        edit <id>
            set interface-select-method {auto | sdwan | specify}
            set interface <interface>
        next
    end
end
```

DHCP proxy

DHCP proxy traffic can use SD-WAN rules or a specific interface:

```
config system settings
    set dhcp-proxy-interface-select-method {auto | sdwan | specify}
    set dhcp-proxy-interface <interface>
end
```

dhcp-proxy-interface-select-method {auto | sdwan | specify}	Select the interface selection method: • `auto`: Set the outgoing interface automatically (default). • `sdwan`: Set the interface by SD-WAN or policy routing rules. • `specify`: Set the interface manually.
dhcp-proxy-interface <interface>	Specify the outgoing interface. This option is only available and must be configured when `interface-select-method` is `specify`.

DHCP relay

DHCP relay traffic can use SD-WAN rules or a specific interface:

```
config system interface
    edit <interface>
        set dhcp-relay-interface-select-method {auto | sdwan | specify}
        set dhcp-relay-interface <interface>
    next
end
```

dhcp-relay-interface-select-method {auto | sdwan | specify}	Select the interface selection method: • `auto`: Set the outgoing interface automatically (default). • `sdwan`: Set the interface by SD-WAN or policy routing rules. • `specify`: Set the interface manually.
dhcp-relay-interface <interface>	Specify the outgoing interface. This option is only available and must be configured when `interface-select-method` is `specify`.

CA and local certificate renewal with SCEP

Certificate renewal with SCEP traffic can use SD-WAN rules or a specific interface:

```
config vpn certificate setting
    set interface-select-method {auto | sdwan | specify}
    set interface <interface>
end
```

IPS TLS protocol active probing

TLS active probing can use SD-WAN rules or a specific interface:

```
config ips global
    config tls-active-probe
        set interface-selection-method {auto | sdwan | specify}
        set interface <interface>
        set vdom <VDOM>
        set source-ip <IPv4 address>
        set source-ip6 <IPv6 address>
    end
end
```

interface-select-method {auto | sdwan | specify}	Select the interface selection method: • `auto`: Set the outgoing interface automatically (default). • `sdwan`: Set the interface by SD-WAN or policy routing rules. • `specify`: Set the interface manually.
interface <interface>	Specify the outgoing interface. This option is only available and must be configured when `interface-select-method` is `specify`.
vdom <VDOM>	Specify the VDOM. This option is only available and must be configured when `interface-select-method` is `sdwan` or `specify`.
source-ip <IPv4 address>	Specify the source IPv4 address. This option is only available and must be configured when `interface-select-method` is `sdwan` or `specify`.
source-ip6 <IPv6 address>	Specify the source IPv6 address. This option is only available and must be configured when `interface-select-method` is `sdwan` or `specify`.

NetFlow and sFlow

NetFlow and sFlow can use SD-WAN rules or a specific interface.

The `netflow` command is global, and utilized by the management VDOM. The `vdom-netflow` command is only available for non-management VDOMs.

```
config system {netflow | vdom-netflow}
    set interface-select-method {auto | sdwan | specify}
    set interface <interface>
end
```

```
config system {sflow | vdom-sflow}
    config collectors
        edit <id>
            set interface-select-method {auto | sdwan | specify}
            set interface <interface>
        next
    end
end
```

interface-select-method {auto | sdwan | specify}	Select the interface selection method: • `auto`: Set the outgoing interface automatically (default). • `sdwan`: Set the interface by SD-WAN or policy routing rules. • `specify`: Set the interface manually.
interface <interface>	Specify the outgoing interface. This option is only available and must be configured when `interface-select-method` is `specify`.

FortiClient EMS

FortiClient EMS endpoint control traffic can use SD-WAN rules or a specific interface:

```
config endpoint-control fctems
    edit 1
        set status enable
        set name fctems1
        set interface-select-method {auto | sdwan | specify}
        set interface <interface>
    next
end
```

TACACS+

System log entries can be sent to external TACACS+ accounting servers. TACACS+ traffic can use SD-WAN rules or a specific IP address:

```
config log tacacs+accounting setting
    set interface-select-method {auto | sdwan | specify}
    set source-ip <IP address>
end
```

SNMP

SNMP Trap traffic can use SD-WAN rules or a specific interface:

```
config system snmp community
    edit <id>
        config hosts
            edit <id>
                set interface-select-method {auto | sdwan | specify}
            next
        end
```

```
config system snmp user
    edit <user>
        set interface-select-method {auto | sdwan | specify}
    next
end
```

Security

Effective security within Fortinet's SD-WAN architecture begins with grouping interfaces into logical zones, such as “overlay” and “underlay.” These zones allow administrators to apply tailored firewall rules and security profiles to sets of interfaces serving similar functions. Not only does this simplify configuration, but it also ensures consistency and clarity in enforcing security policies across the network. An additional benefit is a reduction in the needed firewall policies to secure the traffic. As SD-WAN intelligence will steer application traffic between multiple interfaces as link performance fluctuates, the member interfaces grouped into zones are then used as the destination in firewall policies.

A critical principle of SD-WAN security is applying security at the originating site, particularly in hub-and-spoke topologies. This “shift-left” approach helps organizations address vulnerabilities at the earliest stage possible, reducing both cost and risk. By catching threats before they propagate downstream, you minimize potential damage to connected systems and users. This is particularly clear when implementing ADVPN, where spoke to spoke traffic may bypass the hub FortiGate completely. Often hubs typically maintain permissive policies for transit traffic, but will still implement security for policies that protect and provide access to hosted workloads.

Different traffic paths (e.g., direct internet access vs. corporate datacenter access) call for distinct security treatments. Rather than deploying a one-size-fits-all profiles, using purpose-built security profiles ensures that inspections are efficient and relevant. Mismatching security measures and traffic flows wastes resources and can create blind spots or increase false positives. Similarly, tuning each profile to reflect its target traffic type is essential. For example, a profile designed to protect a high-volume application server will have different thresholds, protocols, and inspection requirements compared to one safeguarding user web browsing.

Advanced configuration

The following topics provide instructions on SD-WAN advanced configuration:

- HA on page 1476
  - SD-WAN with FGCP HA on page 1476
  - Configuring SD-WAN in an HA cluster using virtual VLAN switch on page 1483
  - Configuring SD-WAN in an HA cluster using internal hardware switches on page 1486
- SD-WAN configuration portability on page 1490
- SD-WAN segmentation over a single overlay on page 1495
- SD-WAN segmentation over a single overlay using IPv6 on page 1512
- Copying the DSCP value from the session original direction to its reply direction on page 1520

- Matching BGP extended community route targets in route maps on page 1524
- SD-WAN multi-PoP multi-hub large scale design and failover on page 1530
- SD-WAN traffic shaping and QoS on page 1549

See also Packet distribution for aggregate static IPsec tunnels in SD-WAN on page 3222.

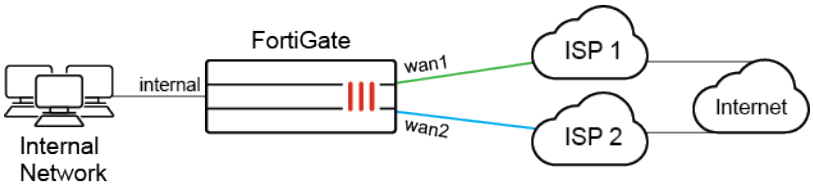
HA

SD-WAN with FGCP HA

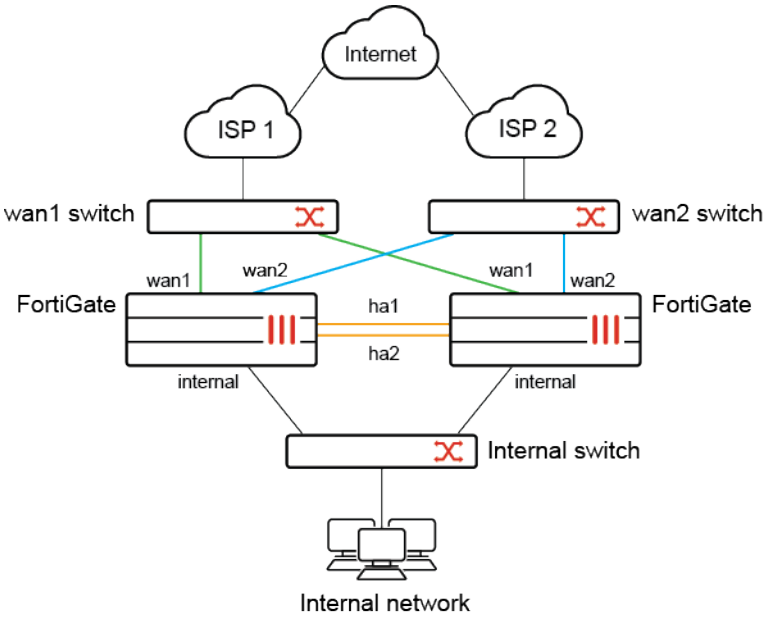
This example shows how to convert a standalone FortiGate SD-WAN solution to a FGCP HA cluster with full-mesh WAN set up. This configuration allows you to load balance your internet traffic between multiple ISP links. It also provides redundancy for your internet connection if your primary ISP in unavailable, or if one of the FortiGates in the HA cluster fails.

This example assumes that a standalone FortiGate has already been configured for SD-WAN by following the SD-WAN quick start on page 1041.

Standalone FortiGate:



FGCP HA cluster:



The following devices are required to convert the topology to HA:

- A second FortiGate that is the same model running the same firmware version.
- Two switches for connecting each FortiGate's WAN interface to the corresponding ISP modem.

Before you begin:

- Ensure that the licenses and subscriptions on both HA members match.
- Ensure that there are one or more ports reserved for HA heartbeat.
- Ensure you have physical access to both HA members.

> ⚠ Enabling HA and re-cabling the WAN interfaces will cause network interruptions.
> This procedure should be performed during a maintenance window.

Configuring the standalone FortiGate for HA

After running the following commands, the FortiGate negotiates to establish an HA cluster. You might temporarily lose connectivity with the FortiGate as FGCP negotiations take place and the MAC addresses of the FortiGate interfaces are changed to HA virtual MAC addresses.

This configurations sets the HA mode to active-passive.

The ha1 and ha2 interfaces are configured as the heartbeat interfaces, with priorities set to 200 and 100 respectively. Setting different priorities for the heartbeat interfaces is a best practice, but is not required.

If you have more than one cluster on the same network, each cluster should have a different group ID. Changing the group ID changes the cluster interface's virtual MAC addresses. If the group IP causes a MAC address conflict on your network, select a different group ID.

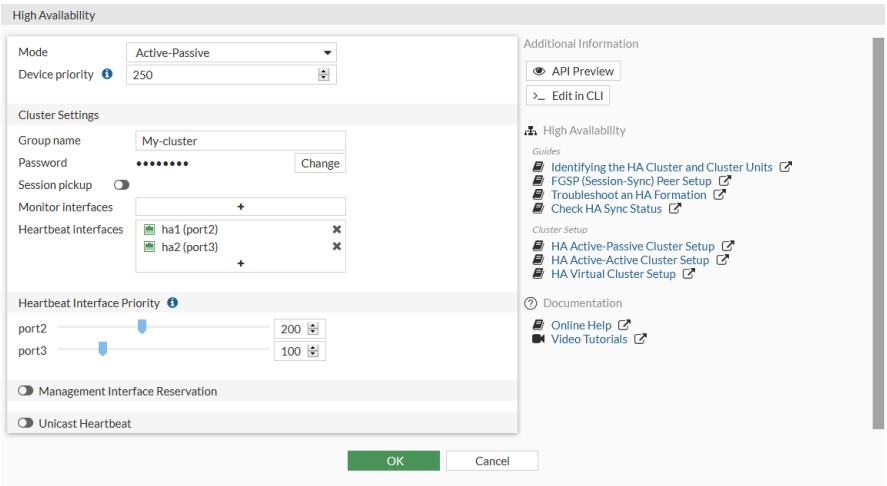
Enabling override and increasing the device priority means that this FortiGate always becomes the primary unit.

To configure the standalone FortiGate for HA in the GUI:

1. Go to *System* > *Settings* and change the *Host name* so that the FortiGate can be easily identified as the primary unit.
2. Go to *System* > *HA* and configure the following options:

Mode	Active-Passive
Device priority	250
Group name	My-cluster
Password	<password>
Heartbeat interfaces	ha1 and ha2
Heartbeat Interface Priority	**port2** (ha1): 200 **port3** (ha2): 100

Override and the group ID can only be configured from the CLI.



3. Click *OK*.
   Connectivity with the FortiGate will temporarily be lost.

To configure the standalone FortiGate for HA in the CLI:

1. Change the host name so that the FortiGate can be easily identified:

```
config system global
    set hostname primary_FG
end
```

2. Configure HA:

```
config system ha
    set mode a-p
    set group-id 100
```

```
    set group-name My-cluster
    set password <password>
    set priority 250
    set override enable
    set hbdev ha1 200 ha2 100
end
```

If HA mode does not start after running the above steps, ensure that none of the FortiGate's interfaces use DHCP or PPPoE addressing.

Configuring the secondary FortiGate for HA

The secondary FortiGate must be the same model and running the same firmware version as the primary FortiGate. The HA settings are the same as the for the primary unit, except the secondary device has a lower priority and override is not enabled.

It is best practice to reset the FortiGate to factory default settings prior to configuring HA. This reduces the chance of synchronization problems.

```
# execute factoryreset
This operation will reset the system to factory default!
Do you want to continue? (y/n) y
```

This is unnecessary if the device is new from the factory.

To configure the secondary FortiGate for HA in the GUI:

1. Go to *System* > *Settings* and change the *Host name* so that the FortiGate can be easily identified as the backup unit.
2. Go to *System* > *HA* and configure the options the same as for the primary FortiGate, except with a lower priority:

Mode	Active-Passive
Device priority	128
Group name	My-cluster
Password	<password>
Heartbeat interfaces	ha1 and ha2
Heartbeat Interface Priority	**port2** (ha1): 200 **port3** (ha2): 100

3. Click *OK*.

To configure the secondary FortiGate for HA in the CLI:

1. Change the host name so that the secondary FortiGate can be easily identified:

```
config system global
    set hostname secondary_FG
end
```

2. Configure HA:

```
config system ha
    set mode a-p
    set group-id 100
    set group-name My-cluster
    set password <password>
    set priority 128
    set hbdev ha1 200 ha2 100
end
```

Connecting the heartbeat interfaces between the FortiGates

To connect and check the heartbeat interfaces:

1. Connect the heartbeat interfaces ha1 and ha2 between the primary and secondary FortiGate.
   a. An HA primary device is selected. Because the primary FortiGate has a higher priority and override enabled, it assumes the role of HA primary.
   b. The secondary FortiGate synchronizes its configuration from the primary device.
2. Verify that the checksums match between the primary and secondary FortiGates:

```
# diagnose sys ha checksum cluster

================== FG5H0XXXXXXXXX0 ==================

is_manage_primary()=1, is_root_primary()=1
debugzone
global: 2b e9 81 38 c2 9d 4f db b7 0e 1f 49 42 c6 1e fb
root: af a6 48 c5 c2 9a 8b 81 a5 53 fb 27 e9 ae 01 6a
all: 89 1f 63 77 48 8a 30 ee 57 06 ca eb 71 e6 8e ad

checksum
global: 2b e9 81 38 c2 9d 4f db b7 0e 1f 49 42 c6 1e fb
root: af a6 48 c5 c2 9a 8b 81 a5 53 fb 27 e9 ae 01 6a
all: 89 1f 63 77 48 8a 30 ee 57 06 ca eb 71 e6 8e ad

================== FG5H0XXXXXXXXX1 ==================

is_manage_primary()=0, is_root_primary()=0
debugzone
global: 2b e9 81 38 c2 9d 4f db b7 0e 1f 49 42 c6 1e fb
root: af a6 48 c5 c2 9a 8b 81 a5 53 fb 27 e9 ae 01 6a
all: 89 1f 63 77 48 8a 30 ee 57 06 ca eb 71 e6 8e ad
```

```
checksum
global: 2b e9 81 38 c2 9d 4f db b7 0e 1f 49 42 c6 1e fb
root: af a6 48 c5 c2 9a 8b 81 a5 53 fb 27 e9 ae 01 6a
all: 89 1f 63 77 48 8a 30 ee 57 06 ca eb 71 e6 8e ad
```

If all of the cluster members have identical checksums, then their configurations are synchronized. If the checksums are not the same, wait for a few minutes, then repeat the command. Some parts of the configuration might take a significant amount of time to synchronize (tens of minutes).

Connecting other traffic interfaces

After the device configurations are synchronized, you can connect the rest of the traffic interfaces. Making these connections will disrupt traffic as cables are disconnected and reconnected.

Switches must be used between the cluster and the ISPs, and between the cluster and the internal network, as shown in the topology diagram.

Checking cluster operations

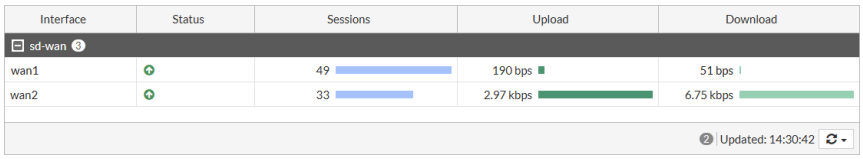
The *HA Status* dashboard widget shows the synchronization status. Hover over the host names of each FortiGate in the widget to verify that they are synchronized and have the same checksum.

To view more information about the cluster status, including the number of sessions passing through the cluster members, go to *System* > *HA*.

See Check HA synchronization status on page 4201 for more information.

Results

1. Browse the internet on a computer in the internal network.
2. Go to *Network* > *SD-WAN* > *SD-WAN Zones* to see the bandwidth, volume, and sessions for traffic on the SD-WAN interfaces. See Results on page 1047 for details.
3. Go to *Dashboard* > *Network Monitor* > *SD-WAN* to see information about each interface, such as the number of sessions and the bit rate.



Testing HA failover

All traffic should currently be flowing through the primary FortiGate. If it becomes unavailable, traffic fails over to the secondary FortiGate. When the primary FortiGate rejoins the cluster, the secondary FortiGate continues to operate as the primary FortiGate.

To test this, ping a reliable IP address from a computer in the internal network, and then power off the primary FortiGate.

There will be a momentary pause in the ping results until traffic diverts to the backup FortiGate, allowing the ping traffic to continue:

```
64 bytes from 184.25.76.114: icmp_seq=69 ttl=52 time=8.719 ms\
64 bytes from 184.25.76.114: icmp_seq=70 ttl=52 time=8.822 ms\
64 bytes from 184.25.76.114: icmp_seq=74 ttl=52 time=8.901 ms\
Request timeout for icmp_seq 75\
64 bytes from 184.25.76.114: icmp_seq=76 ttl=52 time=8.860 ms\
64 bytes from 184.25.76.114: icmp_seq=77 ttl=52 time=9.174 ms\
64 bytes from 184.25.76.114: icmp_seq=83 ttl=52 time=8.639 ms}
```

If you are using port monitoring, you can also unplug the primary FortiGate's internet facing interface to test failover.

After the secondary FortiGate becomes the primary, you can log into the cluster using the same IP address as before the fail over. If the primary FortiGate is powered off, you will be logged into the backup FortiGate. Check the host name to verify what device you have logged into. The FortiGate continues to operate in HA mode, and if you restart the primary FortiGate, it will rejoin the cluster and act as the backup FortiGate. Traffic is not disrupted when the restarted FortiGate rejoins the cluster.

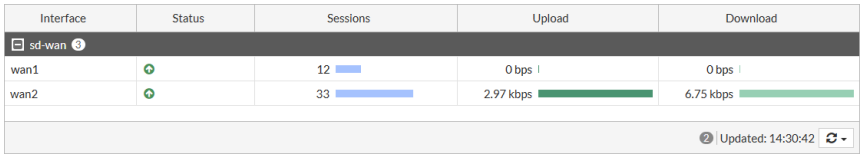
You can also use the CLI to force an HA failover. See Force HA failover for testing and demonstrations on page 4199 for information.

Testing ISP failover

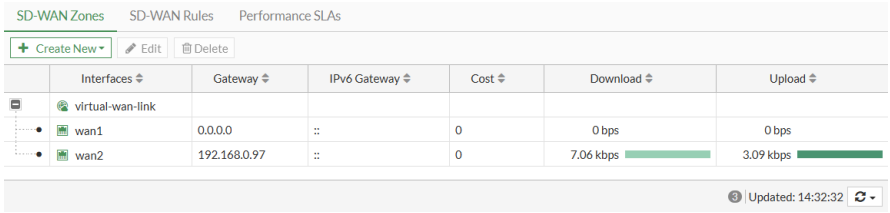
To test a failover of the redundant internet configuration, you need to simulate a failed internet connection to one of the ports. You can do this by disconnecting power from the wan1 switch, or by disconnecting the wan1 interfaces of both FortiGates from ISP1.

After disconnecting, verify that users still have internet access

- Go to *Dashboard* > *Network Monitor* > *SD-WAN*. The *Upload* and *Download* columns for wan1 show that traffic is not going through that interface.



- Go to *Network* > *SD-WAN* > *SD-WAN Zones*. The *Bandwidth*, *Volume*, and *Sessions* tabs show that traffic is entirely diverted to wan2.



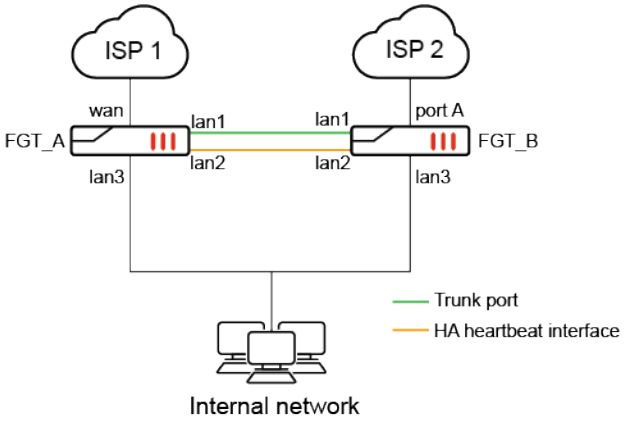
Users on the network should not notice the wan1 failure. If you are using the wan1 gateway IP address to connect to the administrator dashboard, it will appear as though you are still connecting through wan1.

After verifying a successful failover, reestablish the connection to ISP1.

Configuring SD-WAN in an HA cluster using virtual VLAN switch

In this SD-WAN configuration, two FortiGates in an active-passive (A-P) HA pair are used to provide hardware redundancy. Instead of using external switches to provide a mesh network connection to the ISP routers, each FortiGate connects to an ISP and uses a virtual VLAN switch to connect to each other. Virtual VLAN switch mode allows 802.1Q VLANs to be assigned to it, and the configuration of one interface as a trunk port.

Only FortiGate models that support the virtual VLAN switch feature can be used for this solution. See Virtual VLAN switch on page 258 for details.



In this topology:

- An HA cluster of two FortiGate 40Fs is used as an example
- The lan interface is broken up into lan1, lan2 and lan3
- lan1 is connected between FGT_A and FGT_B and designated as a virtual VLAN switch trunk port
- lan2 is used for HA heartbeat, connecting the two FortiGates in HA
- lan3 is connected to the internal network
- wan is connected to ISP1 on FGT_A
- port A is connected to ISP on FGT_B

Two VLANs are created (VLAN 10 and VLAN 20), and assigned to two VLAN switches. Each VLAN switch has the WAN interfaces (wan and port A) as its member.

When FGT_A is the primary device, it reaches ISP1 directly via the local wan interface and reaches ISP2 via the trunk port towards FGT_B. Packets are tagged/untagged as VLAN 10 as it passes through the trunk before egressing on port A on FGT_B.

When FGT_B is the primary device, it reaches ISP2 directly via the local port A interface and reaches ISP1 via the trunk port towards FGT_A. Packets are tagged/untagged as VLAN 20 as it passes through the trunk before egressing on the wan interface on FGT_A.

Using virtual VLAN switches for an SD-WAN HA configuration, as described in this example, requires fewer interfaces than the hardware switch configuration described in Configuring SD-WAN in an HA cluster using internal hardware switches on page 1486

A virtual VLAN switch configuration with 2 WAN connections requires 5 interfaces. With 3 WAN connections, a total of 6 interfaces are required.

For the internal hardware switch solution, each WAN connection requires a corresponding hardware switch interface. With 2 WAN connections, a total of 6 interfaces are needed. With 3 WAN connections, a total of 8 interfaces are needed.

HA failover

This is not a standard HA configuration with external switches. In the case of a device failure, one of the ISPs will no longer be available because the switch that is connected to it will be down.

For example, if FGT_A loses power, HA failover will occur and FGT_B will become the primary unit. Its connection to VLAN 10 over the trunk port will be down, so it will be unable to connect to ISP 1. Its SD-WAN SLAs will be broken, and traffic will only be routed through ISP 2.

SD-WAN SLA health checks should be used to monitor the health of each ISP. HA link monitoring will not be possible, since at least one of the WAN interfaces will be down on each FortiGate at all times.

Configuration

In the configuration example, it is assumed HA A-P mode is already configured and FGT_A and FGT_B are in sync.

To configure the virtual VLAN switch, VLANs and interface settings in the GUI:

1. Enable Virtual VLAN switch from the GUI under the *System* > *Settings* > *General* page. In the *System operation settings* section, enable *VLAN switch mode*. Click *Apply*.
2. Create a new VLAN switch and assign the wan interface in the GUI.
   a. Go to *Network* > *Interfaces* and click *Create New* > *Interface*.
   b. Enter the name *ISP1*
   c. Set the *Type* to *VLAN Switch*.
   d. Enter the *VLAN ID 10*.
   e. Click the + and add the Interface Members. Select the interface *wan*.
   f. Configure the *Address* (192.168.10.99/24) and *Administrative Access* settings as needed.
   g. Click *OK*.
3. Repeat the same steps for a new VLAN switch and assign the port A interface in the GUI.
   a. Go to *Network* > *Interfaces* and click *Create New* > *Interface*.
   b. Enter the name ISP2
   c. Set the Type to VLAN Switch.
   d. Enter the VLAN ID 20.
   e. Click the + and add the Interface Members. Select the interface ‘a’.
   f. Configure the Address (192.168.20.99/24) and Administrative Access settings as needed.

g. Click OK

4. Designate an interface as the trunk port from the CLI:

```
config system interface
    edit lan1
        set trunk enable
    next
end
```

To configure the virtual VLAN switch, VLANs and interface settings in the CLI:

```
config system global
    set virtual-switch-vlan enable
end
config system virtual-switch
    edit "ISP1"
        set physical-switch "sw0"
        set vlan 10
        config port
            edit "wan"
            next
        end
    next
    edit "ISP2"
        set physical-switch "sw0"
        set vlan 20
        config port
            edit "a"
            next
        end
    next
end
config system interface
    edit "ISP1"
        set vdom "root"
        set ip 192.168.10.99 255.255.255.0
        set allowaccess ping
        set type hard-switch
    next
    edit "ISP2"
        set vdom "root"
        set ip 192.168.20.99 255.255.255.0
        set allowaccess ping
        set type hard-switch
    next
end
config system interface
    edit lan1
        set trunk enable
    next
end
```

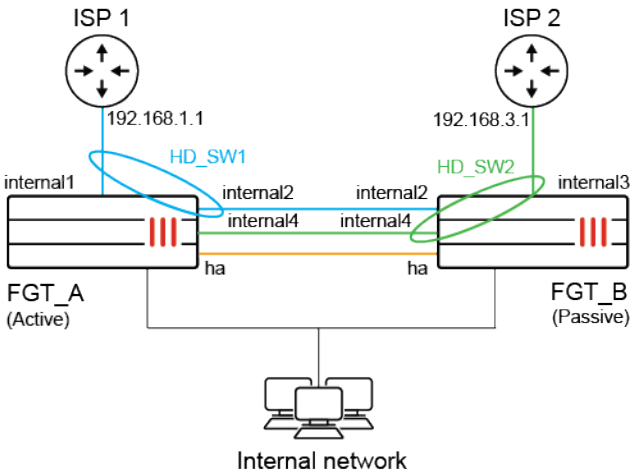
To configure SD-WAN in the GUI:

1. On the FortiGate, go to *Network* > *SD-WAN*, select the *SD-WAN Zones* tab, and click *Create New* > *SD-WAN Member*.
2. In the *Interface* dropdown, select *ISP1*.
3. Leave *SD-WAN Zone* set to *virtual-wan-link*.
4. Enter the *Gateway* address *192.168.10.1*.
5. Click *OK*.
6. Repeat these steps to add the second interface (*ISP2*) with the gateway *192.168.20.1*.
7. Click *Apply*.
8. Create a health check:
   a. Go to *Network* > *SD-WAN*, select the *Performance SLA* tab, and click *Create New*.
   b. Set *Name* to *GW_HC*.
   c. Set *Protocol* to *Ping* and *Servers* to *8.8.8.8*.
   d. Set *Participants* to *All SD-WAN Members*.
   e. Enable *SLA Target* and leave the default values.
   f. Click *OK*.
9. Create SD-WAN rules as needed. The SLA health check can be used to determine when the ISP connections are in or out of SLA, and to failover accordingly.
10. Configure your firewall policy as needed for the virtual-wan-link SD-WAN zone.

Configuring SD-WAN in an HA cluster using internal hardware switches

In this SD-WAN configuration, two FortiGates in an active-passive (A-P) HA pair are used to provide hardware redundancy. Instead of using external switches to provide a mesh network connection to the ISP routers, the FortiGates use their built-in hardware switches to connect to the ISP routers.

> ⚠ Only FortiGate models that have hardware switches can be used for this solution. Ports in a software switch are not in a forwarding state when a FortiGate is acting as a secondary device in a A-P cluster.



In this topology:

- Two hardware switches are created, HD_SW1 and HD_SW2.
- HD_SW1 is used to connect to ISP 1 Router and includes the internal1 and internal2 ports.
- HD_SW2 is used to connect to ISP 2 Router and includes the internal3 and internal4 ports.
- Another interface on each device is used as the HA heartbeat interface, connecting the two FortiGates in HA.

The FortiGates create two hardware switches to connect to ISP 1 and ISP2. When FGT_A is the primary device, it reaches ISP 1 on internal1 in HD_SW1 and ISP 2 on internal4 in HD_SW2. When FGT_B is the primary device, it reaches ISP 1 on internal2 in HD_SW1 and ISP 2 on internal3 on HD_SW2.

HA failover

This is not a standard HA configuration with external switches. In the case of a device failure, one of the ISPs will no longer be available because the switch that is connected to it will be down.

For example, If FGT_A loses power, HA failover will occur and FGT_B will become the primary unit. Its connection to internal2 on HD_SW1 will also be down, so it will be unable to connect to ISP 1. Its SD-WAN SLAs will be broken, and traffic will only be routed through ISP 2.

> ⚠ A link on a hardware switch cannot be monitored in HA monitor, so it is impossible to perform link failure when a port in either of the hardware switches fails. Performing a link failure is unnecessary in this configuration though, because any link failure on the hardware switch will be experienced by both cluster members. SD-WAN SLA health checks should be used to monitor the health of each ISP.

Failure on a hardware switch or ISP router

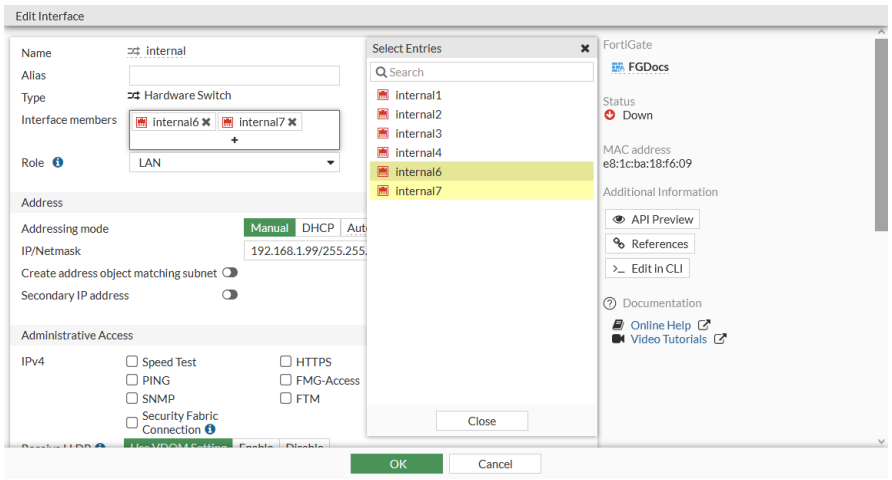
If a hardware switch or switch interface is down, or the ISP router is down, the SD-WAN can detect the broken SLA and continue routing to the other ISP.

For example, if FGT_A is the primary unit, and ISP 2 Router becomes unreachable, the SLA health checks on SD-WAN will detect the broken SLA and cause traffic to stop routing to ISP 2.

Configuration

To configure the HA A-P cluster with internal hardware switches:

1. Configure two FortiGates with internal switches in an A-P HA cluster (follow the steps in HA active-passive cluster setup on page 4079), starting by connecting the heartbeat interface.
2. When the HA cluster is up, connect to the primary FortiGate's GUI.
3. Remove the existing interface members from the default hardware switch:
   a. Go to *Network* > *Interfaces*.
   b. In the *LAN* section, double-click the *internal* interface to edit it.
   c. In *Interface Members*, remove all of the interfaces



d. Click *OK*.

4. Configure the hardware switch interfaces for the two ISPs:

a. Go to *Network* > *Interfaces* and click *Create New* > *Interface*.

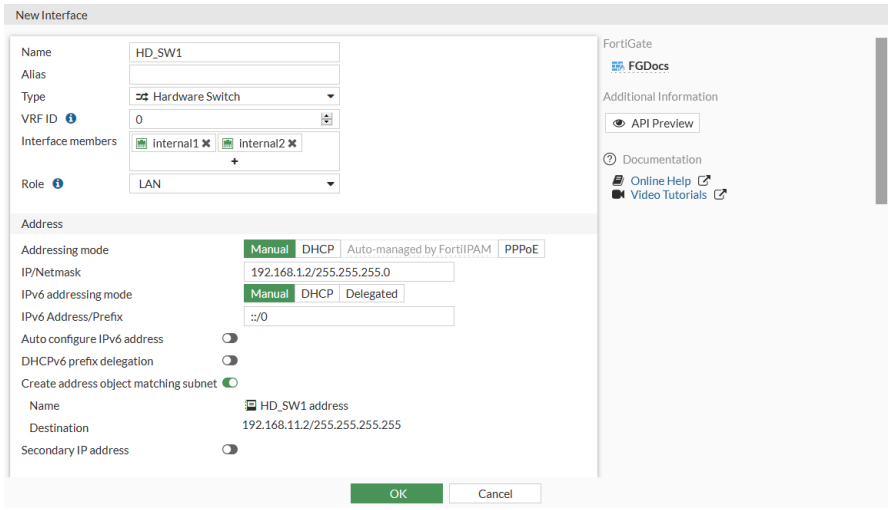
b. Enter a name (*HD_SW1*).

c. Set *Type* to *Hardware Switch*.

d. In *Interface Members*, add two interfaces (*internal1* and *internal2*).

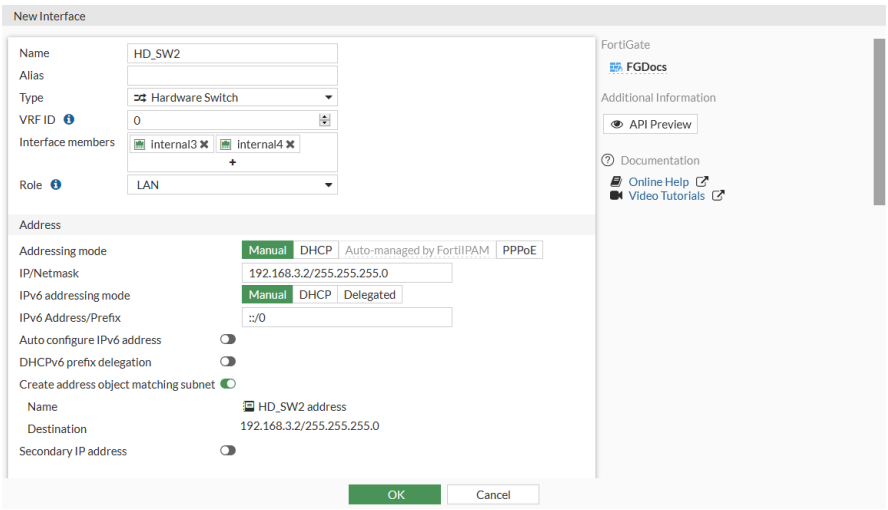
e. Set *IP/Netmask* to 192.168.1.2/24.

f. Configure the remaining settings as needed.



g. Click *OK*.

h. Repeat these steps to create a second hardware switch interface (*HD_SW2*) with two interface members (*internal3* and *internal4*) and *IP/Netmask* set to *192.168.3.2/24*.



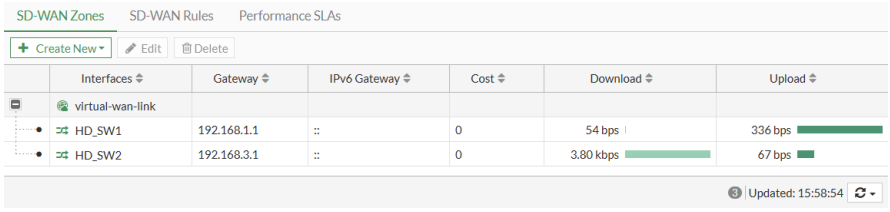
To connect the devices as shown in the topology:

1. Connect the incoming interface to the internal switch on both FortiGates.
2. On FGT_A, connect internal1 of HD_SW1 to ISP 1 Router.
3. On FGT_B, connect internal3 of HD_SW2 to ISP 2 Router.
4. For HD_SW1, connect FGT_A internal2 directly to FGT_B internal2.
5. For HD_SW2, connect FGT_A internal4 directly to FGT_B internal4.

To configure SD-WAN:

The primary FortiGate makes all the SD-WAN decisions.

1. On the primary FortiGate, go to *Network* > *SD-WAN*, select the *SD-WAN Zones* tab, and click *Create New* > *SD-WAN Member*.
2. In the *Interface* dropdown, select *HD_SW1*.
3. Leave *SD-WAN Zone* set to *virtual-wan-link*.
4. Enter the *Gateway* address *192.168.1.1*.
5. Click *OK*.
6. Repeat these steps to add the second interface (*HD_SW2*) with the gateway *192.168.3.1*.
7. Click *Apply*.



8. Create a health check:
   a. Go to *Network* > *SD-WAN*, select the *Performance SLA* tab, and click *Create New*.
   b. Set *Name* to *GW_HC*.
   c. Set *Protocol* to *Ping* and *Servers* to *8.8.8.8*.

d. Set *Participants* to *All SD-WAN Members*.
e. Enable *SLA Target* and leave the default values.
f. Click *OK*.

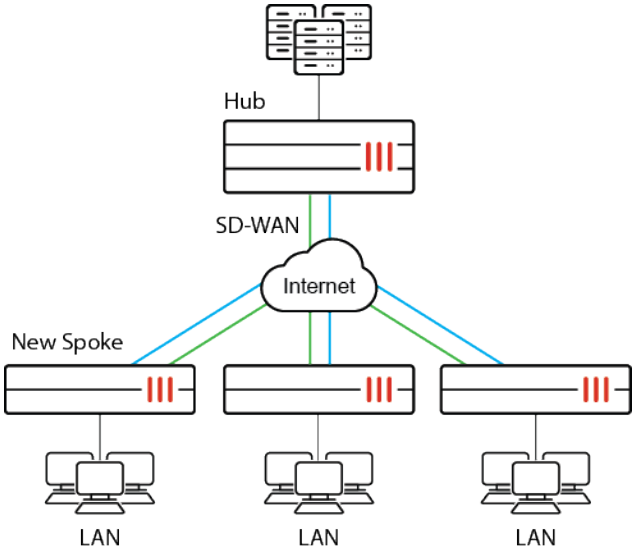
9. Create SD-WAN rules as needed. The SLA health check can be used to determine when the ISP connections are in or out of SLA, and to failover accordingly.

SD-WAN configuration portability

When configuring SD-WAN, adding interfaces to members is optional.

This allows the SD-WAN to be configured without associating any interfaces to SD-WAN members. It also allows a configuration to be copied directly from one device to another, without requiring the devices to have interfaces with the same names.

After the configuration is created, add interfaces to the members make it functional.

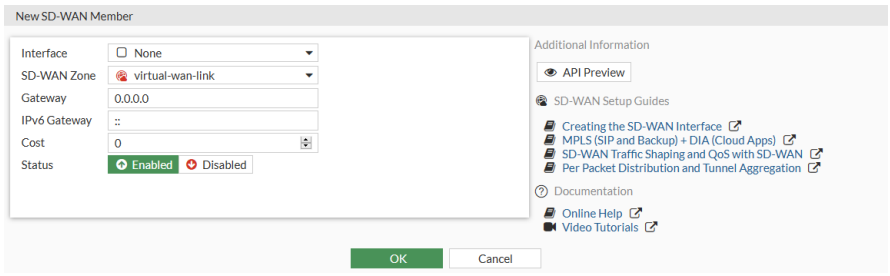


Example 1

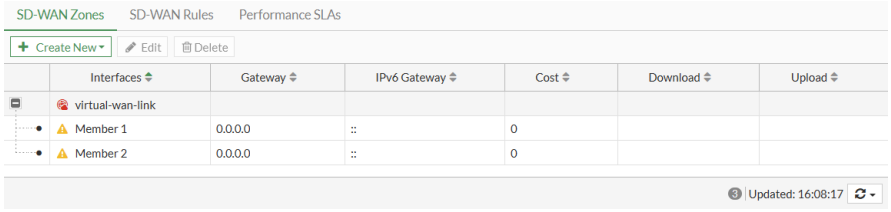
In this example, we create a template with two SD-WAN members configured without assigned interfaces that are used in a performance SLA and SD-WAN rule. The template can be used to configure new devices, as in Example 2 on page 1494. Interfaces are then assigned to the members, and the configuration becomes active.

To create the SD-WAN members in the GUI:

1. Go to *Network* > *SD-WAN*, select the *SD-WAN Zones* tab, and click *Create New* > *SD-WAN Member*.
2. Leave all the settings set to their default values and click *OK*.



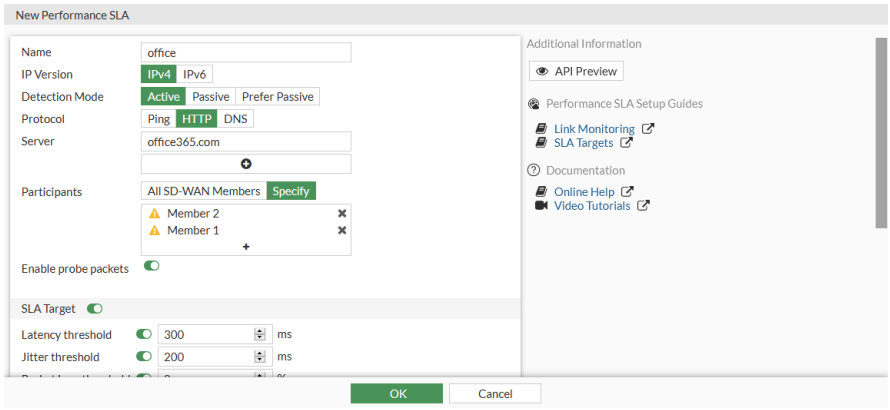
3. Repeat the above steps to create a second member.
   The empty members are listed on the *SD-WAN Zones* tab.



   The members are disabled until interfaces are configured, but can still be used in performance SLAs and SD-WAN rules.

To create a performance SLA in the GUI:

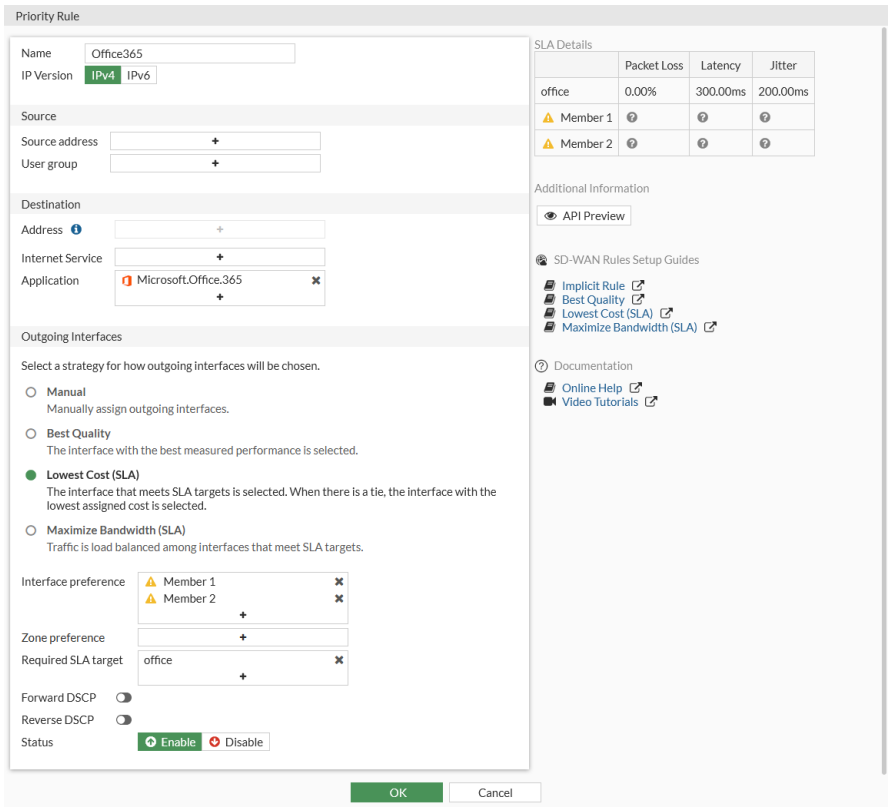
1. Go to *Network* > *SD-WAN* and select the *Performance SLAs* tab.
2. Click *Create New*.
3. Configure the performance SLA, specifying the empty members as participants.



4. Click *OK*.

To create an SD-WAN rule in the GUI:

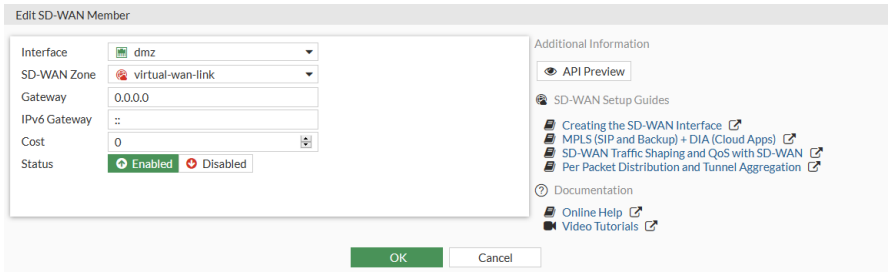
1. Go to *Network* > *SD-WAN*, select the *SD-WAN Rules* tab, and click *Create New*.
2. Configure the rule, adding both members to the *Interface preference* field:



3. Click *OK*.

To assign interfaces to the SD-WAN members in the GUI:

1. Go to *Network* > *SD-WAN* and select the *SD-WAN Zones* tab.
2. Edit the first member
3. Set *Interface* to an actual interface.



4. Click *OK*.
5. Repeat the above steps to assign an interface to the second member.

To configure the SD-WAN in the CLI:

1. Create SD-WAN members:

```
config system sdwan
    set status enable
    config members
```

```
        edit 1
        next
        edit 2
        next
    end
end
```

2. Create a health check (performance SLA):

```
config system sdwan
    config health-check
        edit "office"
            set server "office365.com"
            set protocol http
            set sla-fail-log-period 300
            set sla-pass-log-period 300
            set members 2 1
            config sla
                edit 1
                    set latency-threshold 300
                    set jitter-threshold 200
                next
                edit 2
                    set link-cost-factor latency
                    set latency-threshold 20
                next
            end
        next
    end
end
```

3. Create a service (rule):

```
config system sdwan
    config service
        edit 3
            set name "Office365"
            set mode sla
            set internet-service enable
            set internet-service-app-ctrl 33182
            config sla
                edit "office"
                    set id 2
                next
            end
            set priority-members 1 2
        next
    end
end
```

The SD-WAN configuration can now be used in as a template for new spokes, as in Example 2 on page 1494.

Example 2

In this example, the configuration from Example 1 is copied onto a new FortiGate.

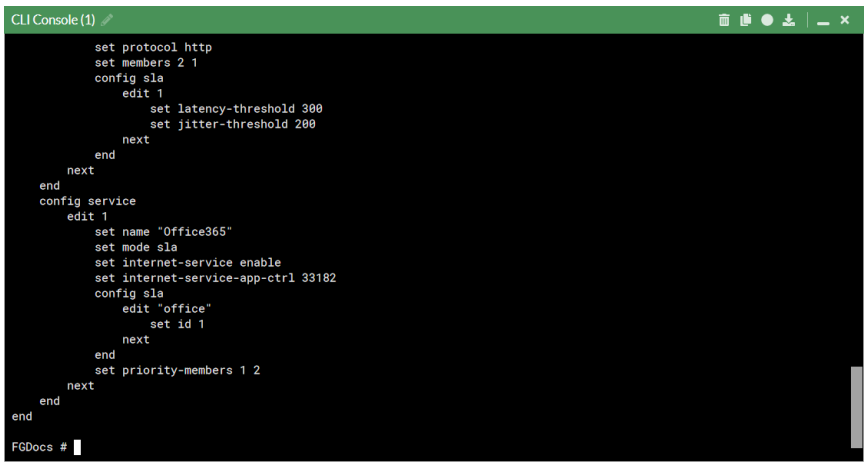
Using the CLI console and the GUI

To copy the SD-WAN configuration from the original FortiGate:

1. Optionally, change the console screen paging setting. See Screen paging on page 81 for details.
2. Open the CLI console.
3. If necessary, click *Clear console* to empty the console.
4. Enter the following command:

```
show system sdwan
```

5. Either click *Download* and open the file in a text editor, or click *Copy to clipboard* and paste the content into a text editor.



6. Edit the CLI configuration as necessary. For example, the first line that shows the `show` command should be deleted, and the default health checks can be removed.
7. If required, save the CLI configuration as a text file.

To paste the SD-WAN configuration onto a new FortiGate:

1. Copy the SD-WAN configuration from the text editor.
2. On the new FortiGate, open the CLI console.
3. Press *Ctrl + v* to paste the CLI commands.
4. In necessary, press *Enter* to apply the last `end` command.

   The SD-WAN configuration is copied to the new FortiGate.

   If the interfaces do not exist, the SD-WAN members are created without interfaces, and are disabled until interfaces are configured.

To assign interfaces to the SD-WAN members:

1. Go to *Network* > *SD-WAN* and select the *SD-WAN Zones* tab.
2. Edit the first member
3. Set *Interface* to an actual interface.
4. Click *OK*.
5. Repeat the above steps to assign an interface to the second member.

Using a terminal emulator

The following instructions use PuTTy. The steps may vary in other terminal emulators.

To copy the SD-WAN configuration from the original FortiGate:

1. Connect to the FortiGate. See Connecting to the CLI on page 73 for details.
2. Enter the following command:

```
show system sdwan
```

3. Select the output, press *Ctrl + c* to copy it, and then paste it into a text editor.
4. Edit the CLI configuration as necessary. For example, the default health checks can be removed.
5. If required, save the CLI configuration as a text file.

To paste the SD-WAN configuration onto a new FortiGate:

1. Connect to the new FortiGate. See Connecting to the CLI on page 73 for details.
2. Copy the SD-WAN configuration from the text editor.
3. Right-click to paste the SD-WAN configuration.
4. In necessary, press *Enter* to apply the last `end` command.

   The SD-WAN configuration is copied to the new FortiGate.

   If the interfaces do not exist, the SD-WAN members are created without interfaces, and are disabled until interfaces are configured.

SD-WAN segmentation over a single overlay

SD-WAN, VPN, and BGP configurations support L3 VPN segmentation over a single overlay. In these configurations, a hub and spoke SD-WAN deployment requires that branch sites, or spokes, are able to accommodate multiple companies or departments, and each company's subnet is separated by a different VRF. A subnet on one VRF cannot communicate with a subnet on another VRF between different branches, but can communicate with the same VRF.

SD-WAN options

VRF-aware SD-WAN health checks

SD-WAN on the originating spoke can tag the health check probes with the correct VRF when transmitting to a multi-VRF tunnel. The hub can then forward the probes to the correct health check server in the same VRF as the hub. The IP version (`addr-mode`) can be either IPv4 or IPv6.

```
config system sdwan
   config health-check
       edit <name>
           set vrf <vrf id>
           set source <address>
       next
   end
end
```

Command	Description
`vrf <vrf id>`	Virtual Routing Forwarding ID.
`source <address>`	Source IP address used in the health-check packet to the server.

Overlay stickiness

When a hub has multiple overlays, traffic received on one overlay should egress on the same overlay when possible. The `service-sla-tie-break` option ensures overlay stickiness. In SD-WAN service rules, options are available to ensure that traffic received in a zone stays in that zone.

```
config system sdwan
    config zone
        edit <name>
            set service-sla-tie-break input-device
        next
    end
    config service
        edit <id>
            set input-zone <zone>
            set tie-break input-device
        next
    end
end
```

Command	Description
`service-sla-tie-break input-device`	Members that meet the SLA are selected by matching the input device.
`input-zone <zone>`	Source input-zone name.
`tie-break input-device`	Members that meet the SLA are selected by matching the input device.

IPsec options

Configurable rate limit for shortcut offers sent by the hub

By default, the hub sends a shortcut offer to a spoke every five seconds. If the hub continues to send offers that keep failing, and there are a large number of spokes, this can cause a high load on the hub. This setting makes the interval between shortcut offers configurable.

```
config vpn ipsec phase1-interface
    edit <name>
        set auto-discovery-offer-interval <interval>
    next
end
```

Command	Description
`auto-discovery-offer-interval <interval>`	Interval between shortcut offer messages, in seconds (1 - 300, default = 5).

Segmentation over a single overlay

Segmentation requires that VRF info is encapsulated within the IPsec VPN tunnel. This setting enables multi-VRF IPSEC tunnels. It is only applicable for forwarding traffic, and cannot be used for local-out traffic.

```
config vpn ipsec phase1-interface
    edit <name>
        set encapsulation vpn-id-ipip
    next
end
```

Command	Description
`encapsulation vpn-id-ipip`	VPN ID with IPIP encapsulation.

VPN configuration for BGP

The role of a VRF can be specified, along with other VRF details. Up to 252 VRFs can be configured per VDOM for any device.

```
config router bgp
    config vrf
        edit <vrf>
            set role {standalone | ce | pe}
            set rd <string>
            set export-rt <route_target>
            set import-rt <route_target>
            set import-route-map <route_map>
            config leak-target
                edit <vrf>
```

```
                    set route-map <route-map>
                    set interface <interface>
                next
            end
        next
    end
end
```

Command	Description
`role {standalone | ce | pe}`	VRF role: standalone, customer edge (CE), or provider edge (PE).
`rd <string>`	Route Distinguisher: AA|AA:NN. This option is only available when the role is CE.
`export-rt <route_target>`	List of export route target. This option is only available when the role is CE.
`import-rt <route_target>`	List of import route target. This option is only available when the role is CE.
`import-route-map <route_map>`	Import route map. This option is only available when the role is CE.
`route-map <route-map>`	Route map of VRF leaking.
`interface <interface>`	Interface that is used to leak routes to the target VRF.

In FortiOS 7.0, `config vrf` was `config vrf-leak`, and `config leak-target` was `config target`.

Display BGP routes by VRF and neighbor

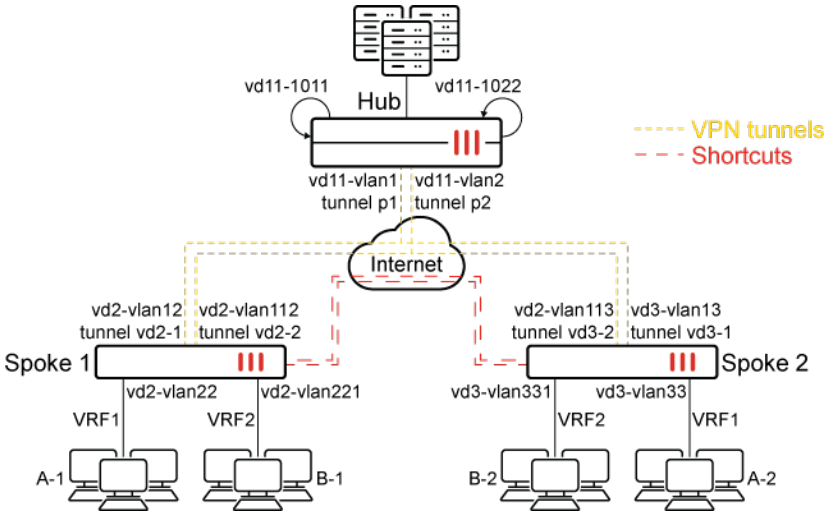
```
# diagnose ip router bgp set-filter vrf <vrf>
# diagnose ip router bgp set-filter neighbor <neighbor address>
# diagnose ip router bgp set-filter reset
# execute router clear bgp vpnv4 unicast soft {in | out}
# get router info filter show
# get router info filter vrf {vrf | all}
```

Examples

In example 1, multiple companies (or departments of a company) share the ADVPN. Company A and company B each have two branches in two different locations. Company A's branches (A-1 and A-2) can talk to each other using the VPN shortcut, but not to company B's branches (B-1 and B-2). Likewise, company B's branches can talk to each other using the VPN shortcut, but not to company A's branches. Traffic can share the tunnels and shortcuts, but cannot be mixed up.

Example 2 shows that performance SLA health checks can be sent from a spoke's VRF to the loopback on the hub that is in the same VRF.

Example 3 shows that when traffic is ingress on the hub on one overlay, it will preferably egress on the same overlay.



Device	Interface	IP Address
Hub	vd11-1011	11.11.11.11
	vd11-1022	11.11.22.11
	vd11-vlan1 tunnel p1	11.1.1.1.11/24 10.10.100.254
	vd11-vlan2 tunnel p2	11.1.2.1.11/24 10.10.200.254
	vd11-vlan10	100.1.1.1/24
Spoke 1	vd2-vlan12 tunnel vd2-1	12.1.1.2 10.10.100.2
	vd2-vlan112 tunnel vd2-2	112.1.1.2 10.10.200.2
	vd2-vlan22	22.1.1.2/24
	vd2-vlan221	22.1.1.2/24
Spoke 2	vd3-vlan13 tunnel vd3-1	13.1.1.3 10.10.100.3
	vd2-vlan113 tunnel vd3-2	113.1.1.3 10.10.200.3
	vd3-vlan33	33.1.1.3/24
	vd3-vlan331	33.1.1.3/24

Example 1

In this example, two spokes each have two tunnels to the hub.

- Each spoke has two VRFs behind it that can use the same IP address or subnets.
- The computers in VRF1 behind spoke 1 can talk to the computers in VRF1 behind spoke 2, but not to any of the computers in the VRF2s behind either spoke.
- The computers in VRF2 behind spoke 1 can talk to the computers in VRF2 behind spoke 2, but not to any of the computers in the VRF1s behind either spoke.

To configure the hub:

```
config router bgp
    set as 65505
    set router-id 11.11.11.11
    set ibgp-multipath enable
    set additional-path enable
    set additional-path-vpnv4 enable
    set cluster-id 11.12.13.14
    set additional-path-select 3
    config neighbor-group
        edit "gr1"
            set capability-graceful-restart enable
            set capability-default-originate enable
            set next-hop-self-rr enable
            set soft-reconfiguration-vpnv4 enable
            set remote-as 65505
            set additional-path both
            set additional-path-vpnv4 both
            set adv-additional-path 3
            set route-reflector-client enable
            set route-reflector-client-vpnv4 enable
        next
        edit "gr2"
            set capability-graceful-restart enable
            set capability-default-originate enable
            set next-hop-self-rr enable
            set soft-reconfiguration-vpnv4 enable
            set remote-as 65505
            set additional-path both
            set additional-path-vpnv4 both
            set adv-additional-path 3
            set route-reflector-client enable
            set route-reflector-client-vpnv4 enable
        next
    end
    config neighbor-range
        edit 1
            set prefix 10.10.100.0 255.255.255.0
            set neighbor-group "gr1"
        next
        edit 2
            set prefix 10.10.200.0 255.255.255.0
            set neighbor-group "gr2"
        next
```

```
    end
    config network
        edit 12
            set prefix 11.11.11.11 255.255.255.255
        next
        edit 22
            set prefix 11.11.22.11 255.255.255.255
        next
        edit 10
            set prefix 100.1.1.0 255.255.255.0
        next
        edit 33
            set prefix 11.1.1.0 255.255.255.0
        next
    end
    config vrf
        edit "0"
            set role pe
        next
        edit "1"
            set role ce
            set rd "1:1"
            set export-rt "1:1"
            set import-rt "1:1"
        next
        edit "2"
            set role ce
            set rd "2:1"
            set export-rt "2:1"
            set import-rt "2:1"
        next
    end
end
```

```
config vpn ipsec phase1-interface
    edit "p1"
        set type dynamic
        set interface "vd11-vlan1"
        set peertype any
        set net-device disable
        set proposal aes128-sha1
        set add-route disable
        set dpd on-idle
        set dhgrp 5
        set auto-discovery-sender enable
        set auto-discovery-offer-interval 10
        set encapsulation vpn-id-ipip
        set psksecret **********
        set dpd-retryinterval 60
    next
    edit "p2"
        set type dynamic
```

```
        set interface "vd11-vlan2"
        set peertype any
        set net-device disable
        set proposal aes128-sha1
        set add-route disable
        set dpd on-idle
        set dhgrp 5
        set auto-discovery-sender enable
        set auto-discovery-offer-interval 10
        set encapsulation vpn-id-ipip
        set psksecret **********
        set dpd-retryinterval 60
    next
end
```

```
config vpn ipsec phase2-interface
    edit "p1"
        set phase1name "p1"
        set proposal aes128-sha1
        set dhgrp 5
    next
    edit "p2"
        set phase1name "p2"
        set proposal aes128-sha1
        set dhgrp 5
    next
end
```

To configure a spoke:

```
config router bgp
    set as 65505
    set router-id 2.2.2.2
    set ebgp-multipath enable
    set ibgp-multipath enable
    set network-import-check disable
    set additional-path enable
    set additional-path6 enable
    set additional-path-vpnv4 enable
    set recursive-next-hop enable
    set graceful-restart enable
    set additional-path-select 4
    config neighbor
        edit "10.10.100.254"
            set capability-dynamic enable
            set capability-graceful-restart-vpnv4 enable
            set soft-reconfiguration enable
            set soft-reconfiguration-vpnv4 enable
            set remote-as 65505
            set additional-path both
            set additional-path-vpnv4 both
```

```
            set adv-additional-path 3
        next
        edit "10.10.200.254"
            set capability-dynamic enable
            set capability-graceful-restart-vpnv4 enable
            set soft-reconfiguration enable
            set soft-reconfiguration-vpnv4 enable
            set remote-as 65505
            set additional-path both
            set additional-path-vpnv4 both
            set adv-additional-path 3
        next
    end
    config network
        edit 3
            set prefix 22.1.1.0 255.255.255.0
        next
        edit 4
            set prefix 12.12.12.0 255.255.255.0
        next
    end
    config vrf
        edit "0"
            set role pe
        next
        edit "1"
            set role ce
            set rd "1:1"
            set export-rt "1:1"
            set import-rt "1:1"
        next
        edit "2"
            set role ce
            set rd "2:1"
            set export-rt "2:1"
            set import-rt "2:1"
        next
    end
end
```

```
config vpn ipsec phase1-interface
    edit "vd2-1"
        set interface "vd2-vlan12"
        set peertype any
        set net-device enable
        set proposal aes128-sha1
        set add-route disable
        set dhgrp 5
        set idle-timeout enable
        set idle-timeoutinterval 5
        set auto-discovery-receiver enable
        set encapsulation vpn-id-ipip
```

```
        set remote-gw 11.1.1.11
        set psksecret **********
    next
    edit "vd2-2"
        set interface "vd2-vlan112"
        set peertype any
        set net-device enable
        set proposal aes128-sha1
        set add-route disable
        set dhgrp 5
        set auto-discovery-receiver enable
        set encapsulation vpn-id-ipip
        set remote-gw 11.1.2.11
        set psksecret **********
    next
end
```

```
config vpn ipsec phase2-interface
    edit "vd2-1"
        set phase1name "vd2-1"
        set proposal aes128-sha1
        set dhgrp 5
        set auto-negotiate enable
    next
    edit "vd2-2"
        set phase1name "vd2-2"
        set proposal aes128-sha1
        set dhgrp 5
        set auto-negotiate enable
    next
end
```

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
        edit "SASE"
        next
        edit "zon2"
        next
    end
    config members
        edit 1
            set interface "vd2-1"
            set cost 10
        next
        edit 2
            set interface "vd2-2"
            set cost 20
        next
    end
```

```
    config health-check
        edit "ping"
            set server "11.11.11.11"
            set members 1 2
            config sla
                edit 1
                    set latency-threshold 200
                    set jitter-threshold 50
                next
            end
        next
        edit "1"
            set server "22.1.1.2"
            set vrf 1
            set members 1 2
        next
    end
    config service
        edit 2
            set mode sla
            set dst "100-200"
            config sla
                edit "ping"
                    set id 1
                next
            end
            set priority-members 2
            set use-shortcut-sla disable
        next
        edit 1
            set name "test-tag"
            set mode sla
            set dst "001-100"
            config sla
                edit "ping"
                    set id 1
                next
            end
            set priority-members 1 2
        next
    end
end
```

To check the spoke 1 routes:

```
# get router info routing-table bgp
Routing table for VRF=0
B*      0.0.0.0/0 [200/0] via 10.10.100.254 (recursive via vd2-1 tunnel 11.1.1.11 vrf 0),
04:42:57, [1/0]
                  [200/0] via 10.10.200.254 (recursive via vd2-2 tunnel 11.1.2.11 vrf 0),
04:42:57, [1/0]
B       1.1.1.1/32 [200/0] via 11.1.1.1 [2] (recursive via 12.1.1.1, vd2-vlan12), 04:42:57, [1/0]
```

```
B       1.222.222.222/32 [200/0] via 11.1.1.1 [2] (recursive via 12.1.1.1, vd2-vlan12), 04:42:57,
[1/0]
B       11.11.11.11/32 [200/0] via 10.10.100.254 (recursive via vd2-1 tunnel 11.1.1.11 vrf 0),
04:42:57, [1/0]
                       [200/0] via 10.10.200.254 (recursive via vd2-2 tunnel 11.1.2.11 vrf 0),
04:42:57, [1/0]
B       33.1.1.0/24 [200/0] via 10.10.100.254 [2] (recursive via vd2-1 tunnel 11.1.1.11 vrf 0),
04:42:57, [1/0]
                    [200/0] via 10.10.200.254 [2] (recursive via vd2-2 tunnel 11.1.2.11 vrf 0),
04:42:57, [1/0]
B       100.1.1.0/24 [200/0] via 10.10.100.254 (recursive via vd2-1 tunnel 11.1.1.11 vrf 0),
04:42:57, [1/0]
                     [200/0] via 10.10.200.254 (recursive via vd2-2 tunnel 11.1.2.11 vrf 0),
04:42:57, [1/0]

Routing table for VRF=1
B V     33.1.1.0/24 [200/0] via 10.10.100.3 [2] (recursive via vd2-1 tunnel 11.1.1.11 vrf 0),
04:42:57, [1/0]
                    [200/0] via 10.10.200.3 [2] (recursive is directly connected, vd2-2_0),
04:42:57, [1/0]

Routing table for VRF=2
B V     33.1.1.0/24 [200/0] via 10.10.100.3 [2] (recursive via vd2-1 tunnel 11.1.1.11 vrf 0),
04:42:56, [1/0]
                    [200/0] via 10.10.200.3 [2] (recursive is directly connected, vd2-2_0),
04:42:56, [1/0]
```

VRF1 routes:

```
# get router info filter vrf 1
# get router info routing-table bgp
Routing table for VRF=1
B V     33.1.1.0/24 [200/0] via 10.10.100.3 [2] (recursive via vd2-1 tunnel 11.1.1.11 vrf 0),
04:44:11, [1/0]
                    [200/0] via 10.10.200.3 [2] (recursive is directly connected, vd2-2_0),
04:44:11, [1/0]
```

To test the configuration on shortcut 1:

1. From VRF1 of spoke 1 ping VRF1 of spoke 2 and from VRF2 of spoke 1 ping VRF2 spoke 2. Both VRF1 and VRF2 source and destination IP addresses are the same, so you can see how the traffic is isolated
2. Check sessions on spoke 1:

   The output `vd=<vdom ID>:<VRF ID>` indicates that sessions are created in and stay in the corresponding VRFs.

   - User at 22.1.1.22 in VRF1 on spoke 1 pings 33.1.1.33 in VRF1 on spoke2.

     ```
     # diagnose sys session list
     session info: proto=1 proto_state=00 duration=21 expire=42 timeout=0 flags=00000000
     socktype=0 sockport=0 av_idx=0 use=3
     origin-shaper=
     reply-shaper=
     ```

```
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty
statistic(bytes/packets/allow_err): org=420/5/1 reply=420/5/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=89->131/131->89 gwy=10.10.200.3/22.1.1.22
hook=pre dir=org act=noop 22.1.1.22:48417->33.1.1.33:8(0.0.0.0:0)
hook=post dir=reply act=noop 33.1.1.33:48417->22.1.1.22:0(0.0.0.0:0)
src_mac=02:4c:a5:fc:6a:7f
misc=0 policy_id=1 pol_uuid_idx=566 auth_info=0 chk_client_info=0 vd=1:1
serial=00092eee tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=1
rpdb_link_id=ff000001 ngfwid=n/a
npu_state=0x5040001 no_offload
no_ofld_reason:  disabled-by-policy non-npu-intf
```

- User at 22.1.1.22 in VRF2 on spoke 1 pings 33.1.1.33 in VRF2 on spoke2:

```
# diagnose sys session list
session info: proto=1 proto_state=00 duration=4 expire=56 timeout=0 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty
statistic(bytes/packets/allow_err): org=168/2/1 reply=168/2/1 tuples=2
tx speed(Bps/kbps): 39/0 rx speed(Bps/kbps): 39/0
orgin->sink: org pre->post, reply pre->post dev=113->131/131->113
gwy=10.10.200.3/22.1.1.22
hook=pre dir=org act=noop 22.1.1.22:55841->33.1.1.33:8(0.0.0.0:0)
hook=post dir=reply act=noop 33.1.1.33:55841->22.1.1.22:0(0.0.0.0:0)
src_mac=02:4c:a5:fc:6a:7f
misc=0 policy_id=1 pol_uuid_idx=566 auth_info=0 chk_client_info=0 vd=1:2
serial=00092f77 tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=1
rpdb_link_id=ff000001 ngfwid=n/a
npu_state=0x5040001 no_offload
no_ofld_reason:  disabled-by-policy non-npu-intf
```

3. Check sessions on spoke 2:

   The output `vd=<vdom ID>:<VRF ID>` indicates that sessions are created in and stay in the corresponding VRFs.

   - User at 22.1.1.22 in VRF1 on spoke 1 pings 33.1.1.33 in VRF1 on spoke 2:

```
# diagnose sys session list
session info: proto=1 proto_state=00 duration=11 expire=49 timeout=0 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
```

```
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty npu
statistic(bytes/packets/allow_err): org=168/2/1 reply=168/2/1 tuples=2
tx speed(Bps/kbps): 14/0 rx speed(Bps/kbps): 14/0
orgin->sink: org pre->post, reply pre->post dev=132->92/92->132 gwy=33.1.1.33/10.10.200.2
hook=pre dir=org act=noop 22.1.1.22:27733->33.1.1.33:8(0.0.0.0:0)
hook=post dir=reply act=noop 33.1.1.33:27733->22.1.1.22:0(0.0.0.0:0)
misc=0 policy_id=1 pol_uuid_idx=630 auth_info=0 chk_client_info=0 vd=6:1
serial=000a29fd tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000001 no_offload
npu info: flag=0x00/0x82, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0,
vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
no_ofld_reason:  disabled-by-policy
```

- User at 22.1.1.22 in VRF2 on spoke 1 pings 33.1.1.33 in VRF2 on spoke 2:

```
# diagnose sys session list
session info: proto=1 proto_state=00 duration=17 expire=43 timeout=0 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty npu
statistic(bytes/packets/allow_err): org=168/2/1 reply=168/2/1 tuples=2
tx speed(Bps/kbps): 9/0 rx speed(Bps/kbps): 9/0
orgin->sink: org pre->post, reply pre->post dev=132->115/115->132
gwy=33.1.1.33/10.10.200.2
hook=pre dir=org act=noop 22.1.1.22:24917->33.1.1.33:8(0.0.0.0:0)
hook=post dir=reply act=noop 33.1.1.33:24917->22.1.1.22:0(0.0.0.0:0)
dst_mac=02:4c:a5:fc:6a:7f
misc=0 policy_id=1 pol_uuid_idx=630 auth_info=0 chk_client_info=0 vd=6:2
serial=000a29ca tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000001 no_offload
npu info: flag=0x00/0x82, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0,
vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
no_ofld_reason:  disabled-by-policy
```

To test the configuration on shortcut 2:

1. From VRF1 of spoke 1 ping VRF1 of spoke 2 and from VRF2 of spoke 1 ping VRF2 spoke 2. Both VRF1 and VRF2 source and destination IP addresses are the same, so you can see how the traffic is isolated
2. Check sessions on spoke 1:

   The output `vd=<vdom ID>:<VRF ID>` indicates that sessions are created in and stay in the corresponding VRFs.

- User at 22.1.1.22 in VRF1 on spoke 1 pings 33.1.1.133 in VRF1 on spoke 2:

```
# diagnose sys session listsession info: proto=1 proto_state=00 duration=17 expire=45
timeout=0 flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty
statistic(bytes/packets/allow_err): org=336/4/1 reply=336/4/1 tuples=2
tx speed(Bps/kbps): 19/0 rx speed(Bps/kbps): 19/0
orgin->sink: org pre->post, reply pre->post dev=89->137/137->89 gwy=10.10.200.3/22.1.1.22
hook=pre dir=org act=noop 22.1.1.22:25968->33.1.1.133:8(0.0.0.0:0)
hook=post dir=reply act=noop 33.1.1.133:25968->22.1.1.22:0(0.0.0.0:0)
src_mac=02:4c:a5:fc:6a:7f
misc=0 policy_id=1 pol_uuid_idx=566 auth_info=0 chk_client_info=0 vd=1:1
serial=000aa475 tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=2
rpdb_link_id=ff000002 ngfwid=n/a
npu_state=0x5040001 no_offload
no_ofld_reason:  disabled-by-policy non-npu-intf
```

- User at 22.1.1.22 in VRF2 on spoke 1 pings 33.1.1.133 in VRF2 on spoke 2:

```
# diagnose sys session listsession info: proto=1 proto_state=00 duration=8 expire=53
timeout=0 flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty
statistic(bytes/packets/allow_err): org=252/3/1 reply=252/3/1 tuples=2
tx speed(Bps/kbps): 30/0 rx speed(Bps/kbps): 30/0
orgin->sink: org pre->post, reply pre->post dev=113->137/137->113
gwy=10.10.200.3/22.1.1.22
hook=pre dir=org act=noop 22.1.1.22:28528->33.1.1.133:8(0.0.0.0:0)
hook=post dir=reply act=noop 33.1.1.133:28528->22.1.1.22:0(0.0.0.0:0)
src_mac=02:4c:a5:fc:6a:7f
misc=0 policy_id=1 pol_uuid_idx=566 auth_info=0 chk_client_info=0 vd=1:2
serial=000aa49f tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=2
rpdb_link_id=ff000002 ngfwid=n/a
npu_state=0x5040001 no_offload
no_ofld_reason:  disabled-by-policy non-npu-intf
```

3. Check sessions on spoke 2:

   The output `vd=<vdom ID>:<VRF ID>` indicates that sessions are created in and stay in the corresponding VRFs.

   - User at 22.1.1.22 in VRF1 on spoke 1 pings 33.1.1.133 in VRF1 on spoke 2:

```
# diagnose sys session list
session info: proto=1 proto_state=00 duration=24 expire=38 timeout=0 flags=00000000
```

```
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty npu
statistic(bytes/packets/allow_err): org=336/4/1 reply=336/4/1 tuples=2
tx speed(Bps/kbps): 13/0 rx speed(Bps/kbps): 13/0
orgin->sink: org pre->post, reply pre->post dev=138->92/92->138 gwy=33.1.1.133/10.10.200.2
hook=pre dir=org act=noop 22.1.1.22:25968->33.1.1.133:8(0.0.0.0:0)
hook=post dir=reply act=noop 33.1.1.133:25968->22.1.1.22:0(0.0.0.0:0)
misc=0 policy_id=1 pol_uuid_idx=630 auth_info=0 chk_client_info=0 vd=6:1
serial=000aa476 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000001 no_offload
npu info: flag=0x00/0x82, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0,
vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
no_ofld_reason:  disabled-by-policy
```

- User at 22.1.1.22 in VRF2 on spoke 1 pings 33.1.1.133 in VRF2 on spoke2:

```
# diagnose sys session list
session info: proto=1 proto_state=00 duration=15 expire=46 timeout=0 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty npu
statistic(bytes/packets/allow_err): org=252/3/1 reply=252/3/1 tuples=2
tx speed(Bps/kbps): 16/0 rx speed(Bps/kbps): 16/0
orgin->sink: org pre->post, reply pre->post dev=138->115/115->138
gwy=33.1.1.133/10.10.200.2
hook=pre dir=org act=noop 22.1.1.22:28528->33.1.1.133:8(0.0.0.0:0)
hook=post dir=reply act=noop 33.1.1.133:28528->22.1.1.22:0(0.0.0.0:0)
misc=0 policy_id=1 pol_uuid_idx=630 auth_info=0 chk_client_info=0 vd=6:2
serial=000aa4a0 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000001 no_offload
npu info: flag=0x00/0x82, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0,
vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
no_ofld_reason:  disabled-by-policy
```

Example 2

In this example, SLA health checks are sent from a spoke's VRF to the loopback on the hub that is in the same VRF.

To configure the health check:

```
config system sdwan
    config health-check
        edit "1"
            set server "11.11.22.11"
            set vrf 1
            set source 22.1.1.2
            set members 1 2
            config sla
                edit 1
                    set latency-threshold 200
                    set jitter-threshold 50
                next
            end
        next
    end
end
```

To check the health check status:

```
# diagnose sys sdwan health-check status  1
Health Check(1):
Seq(1 vd2-1): state(alive), packet-loss(0.000%) latency(0.023), jitter(0.002), mos(4.404),
bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x1
Seq(2 vd2-2): state(alive), packet-loss(0.000%) latency(0.022), jitter(0.002), mos(4.404),
bandwidth-up(0), bandwidth-dw(0), bandwidth-bi(0) sla_map=0x1
```

Example 3

In this example, when traffic from spoke 1 arrives at the hub on tunnel 1, it will egress the hub on tunnel 1 to go to other spokes. If traffic arrives on tunnel 2, it will egress on tunnel 2, and not tunnel 1.

To configure SD-WAN on the hub:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
            set service-sla-tie-break input-device
        next
    end
    config members
        edit 1
            set interface "p1"
        next
        edit 2
            set interface "p2"
        next
    end
```

```
    config health-check
        edit "1"
            set server "22.1.1.2"
            set members 1 2
            config sla
                edit 1
                next
            end
        next
    end
    config service
        edit 1
            set mode sla
            set dst "all"
            config sla
                edit "1"
                    set id 1
                next
            end
            set priority-members 1 2
            set tie-break input-device
        next
    end
end
```

To verify that traffic stays in the same overlay on ingress and egress on the hub:

1. Confirm that the SD-WAN service rule has `Tie break` set to `input-device` so that, when SLAs are met on all of the members, traffic prefers to egress on the same member as the input device:

```
# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x200 use-shortcut-sla
 Tie break: input-device
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Members(2):
    1: Seq_num(1 p1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), selected
    2: Seq_num(2 p2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected
  Dst address(1):
        0.0.0.0-255.255.255.255
```

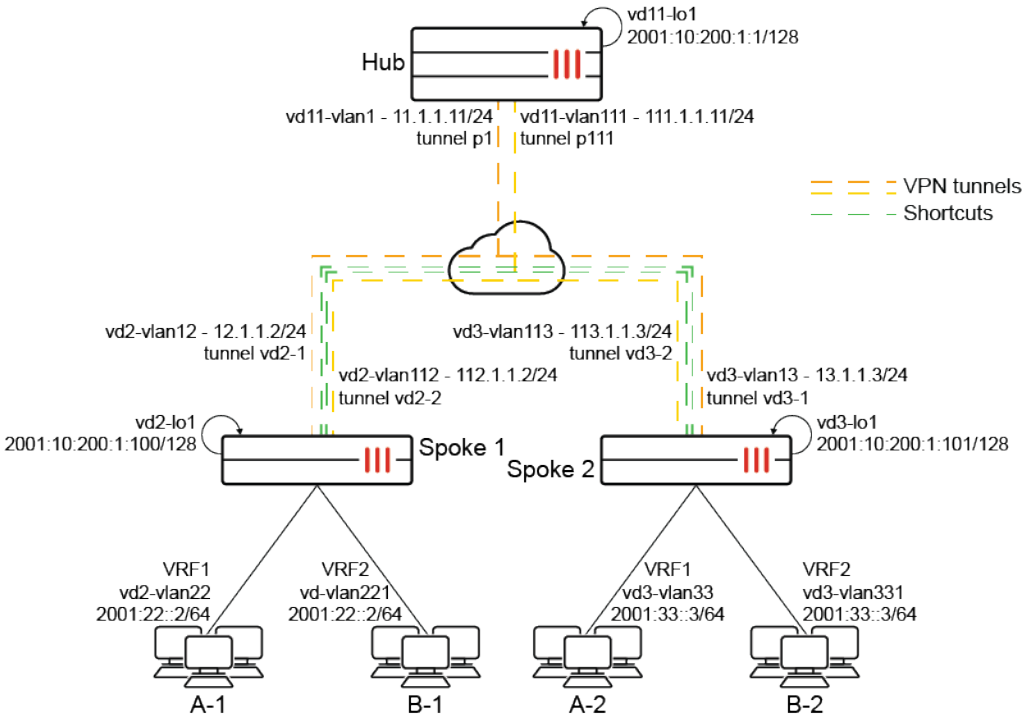
2. Use `diagnose sniffer packet` commands to verify that traffic ingress and egress are on the same overlay.

SD-WAN segmentation over a single overlay using IPv6

IPv6 is supported for SD-WAN segmentation over a single overlay. This allows seamless communication between IPv6 devices within virtual routing and forwarding (VRF) overlay networks, benefiting organizations transitioning to IPv6 or operating in a dual-stack environment.

Example

In this example, multiple companies (or departments of a company) share the ADVPN. Company A and company B each have two branches in two different locations. Company A's branches (A-1 and A-2) can talk to each other using the VPN shortcut, but not to company B's branches (B-1 and B-2). Likewise, company B's branches can talk to each other using the VPN shortcut, but not to company A's branches. Traffic can share the tunnels and shortcuts, but cannot be mixed up.



In this example, two spokes each have two tunnels to the hub.

- Each spoke has two VRFs behind it that can use the same IP address or subnets.
- The computers in VRF1 behind spoke 1 can talk to the computers in VRF1 behind spoke 2, but not to any of the computers in the VRF2s behind either spoke.
- The computers in VRF2 behind spoke 1 can talk to the computers in VRF2 behind spoke 2, but not to any of the computers in the VRF1s behind either spoke.
- Loopback addresses are used for communication between the spokes and the hub instead of tunnel IP address.

> The `exchange-ip-addr6` option allows a loopback IPv6 address to be exchanged between the spokes and the hub in a network. This means that instead of using the tunnel IP address, which is typically used for communication, the loopback IPv6 address is used.

See config router bgp and config router route-map in the CLI Reference for a comprehensive list of commands.

To configure the hub:

1. Configure the BGP settings:

```
config router bgp
    set as 65100
    set router-id 10.200.1.1
    set keepalive-timer 5
    set holdtime-timer 15
    set ibgp-multipath enable
    set network-import-check disable
    set additional-path6 enable
    set additional-path-vpnv6 enable
    set additional-path-select6 4
    config neighbor-group
        edit "EDGEv6"
            set advertisement-interval 1
            set activate disable
            set activate-vpnv4 disable
            set capability-graceful-restart enable
            set next-hop-self-rr6 enable
            set soft-reconfiguration6 enable
            set remote-as 65100
            set update-source "vd11-lo1"
            set additional-path6 both
            set adv-additional-path6 4
            set route-reflector-client6 enable
            set route-reflector-client-vpnv6 enable
        next
    end
    config neighbor-range6
        edit 2
            set prefix6 2001::10:200:1:0/112
            set neighbor-group "EDGEv6"
        next
    end
    config network6
        edit 1
            set prefix6 2001::10:200:1:0/112
        next
    end
    config vrf6
        edit "0"
            set role pe
        next
        edit "1"
            set role ce
            set rd "1:1"
            set export-rt "1:1"
            set import-rt "1:1"
        next
        edit "2"
```

```
            set role ce
            set rd "2:1"
            set export-rt "2:1"
            set import-rt "2:1"
        next
    end
end
```

2. Configure the IPsec phase 1 interface settings:

```
config vpn ipsec phase1-interface
    edit "p1"
        set type dynamic
        set interface "vd11-vlan1"
        set ike-version 2
        set peertype any
        set net-device disable
        set exchange-ip-addr6 2001::10:200:1:1
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set add-route disable
        set dpd on-idle
        set npu-offload disable
        set dhgrp 5
        set auto-discovery-sender enable
        set encapsulation vpn-id-ipip
        set psksecret **********
        set dpd-retryinterval 60
    next
    edit "p111"
        set type dynamic
        set interface "vd11-vlan111"
        set ike-version 2
        set peertype any
        set net-device disable
        set exchange-ip-addr6 2001::10:200:1:1
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set add-route disable
        set dpd on-idle
        set npu-offload disable
        set dhgrp 5
        set auto-discovery-sender enable
        set encapsulation vpn-id-ipip
        set psksecret **********
        set dpd-retryinterval 60
    next
end
```

3. Configure the IPsec phase 2 interface settings:

```
config vpn ipsec phase2-interface
    edit "p1-v6"
        set phase1name "p1"
        set proposal aes128-sha1
        set replay disable
        set src-addr-type subnet6
        set dst-addr-type subnet6
    next
    edit "p111-v6"
        set phase1name "p111"
        set proposal aes128-sha1
        set replay disable
        set src-addr-type subnet6
        set dst-addr-type subnet6
    next
end
```

To configure a spoke:

1. Configure the BGP settings:

```
config router bgp
    set as 65100
    set router-id 10.200.1.100
    set keepalive-timer 5
    set holdtime-timer 15
    set ibgp-multipath enable
    set additional-path6 enable
    set additional-path-vpnv6 enable
    set recursive-next-hop enable
    set tag-resolve-mode merge
    set graceful-restart enable
    set additional-path-select6 4
    config neighbor
        edit "2001::10:200:1:1"
            set advertisement-interval 1
            set activate disable
            set activate-vpnv4 disable
            set capability-dynamic enable
            set capability-graceful-restart6 enable
            set capability-graceful-restart-vpnv6 enable
            set soft-reconfiguration6 enable
            set remote-as 65100
            set route-map-in6 "tag"
            set route-map-in-vpnv6 "tag"
            set connect-timer 10
            set update-source "vd2-lo1"
            set additional-path6 both
            set additional-path-vpnv6 both
        next
    end
    config network6
```

```
        edit 1
            set prefix6 2001:22::/64
        next
        edit 2
            set prefix6 2001::10:200:1:100/128
        next
    end
    config vrf6
        edit "0"
            set role pe
        next
        edit "1"
            set role ce
            set rd "1:1"
            set export-rt "1:1"
            set import-rt "1:1"
        next
        edit "2"
            set role ce
            set rd "2:1"
            set export-rt "2:1"
            set import-rt "2:1"
        next
    end
end
```

2. Configure the IPsec phase 1 interface settings:

```
config vpn ipsec phase1-interface
    edit "vd2-1"
        set interface "vd2-vlan12"
        set ike-version 2
        set peertype any
        set net-device enable
        set exchange-ip-addr6 2001::10:200:1:100
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set add-route disable
        set npu-offload disable
        set dhgrp 5
        set auto-discovery-receiver enable
        set encapsulation vpn-id-ipip
        set remote-gw 11.1.1.11
        set psksecret **********
    next
    edit "vd2-2"
        set interface "vd2-vlan112"
        set ike-version 2
        set peertype any
        set net-device enable
        set exchange-ip-addr6 2001::10:200:1:100
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
```

```
chacha20poly1305-prfsha256
        set add-route disable
        set npu-offload disable
        set dhgrp 5
        set auto-discovery-receiver enable
        set encapsulation vpn-id-ipip
        set remote-gw 111.1.1.11
        set psksecret **********
    next
end
```

3. Configure the IPsec phase 2 interface settings:

```
config vpn ipsec phase2-interface
    edit "vd2-1-6"
        set phase1name "vd2-1"
        set proposal aes128-sha1
        set dhgrp 5
        set replay disable
        set auto-negotiate enable
        set src-addr-type subnet6
        set dst-addr-type subnet6
    next
    edit "vd2-2-6"
        set phase1name "vd2-2"
        set proposal aes128-sha1
        set dhgrp 5
        set replay disable
        set auto-negotiate enable
        set src-addr-type subnet6
        set dst-addr-type subnet6
    next
end
```

4. Configure the SD-WAN settings:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "vd2-1"
            set cost 10
        next
        edit 2
            set interface "vd2-2"
            set cost 20
        next
    end
```

```
    config health-check
        edit "ping6"
            set addr-mode ipv6
            set server "2001::10:200:1:1"
            set source6 2001::10:200:1:100
            set members 1 2
            config sla
                edit 1
                next
            end
        next
    end
    config service
        edit 61
            set addr-mode ipv6
            set priority-members 1
            set dst6 "6001-100"
        next
        edit 62
            set addr-mode ipv6
            set priority-members 2
            set dst6 "6100-200"
        next
    end
end
```

To check the spoke 1 routes:

```
# get router info6 routing-table bgp
Routing table for VRF=0
B       2001::10:200:1:0/112 [200/0] via 2001::10:200:1:1 tag 100 (recursive via vd2-1 tunnel
::11.1.1.11), 1d15h41m
(recursive via vd2-2 tunnel ::111.1.1.11), 1d15h41m, [1024/0]
B       2001::10:200:1:101/128 [200/0] via 2001::10:200:1:1 tag 100 (recursive via vd2-1 tunnel
::11.1.1.11), 1d15h41m
(recursive via vd2-2 tunnel ::111.1.1.11), 1d15h41m, [1024/0]

Routing table for VRF=1
B V     2001:33::/64 [200/0] via 2001::10:200:1:101 tag 100 (recursive via vd2-1 tunnel
::11.1.1.11), 1d15h41m
(recursive via vd2-2 tunnel ::111.1.1.11), 1d15h41m, [1024/0]

Routing table for VRF=2
B V     2001:33::/64 [200/0] via 2001::10:200:1:101 tag 100 (recursive via vd2-1 tunnel
::11.1.1.11), 1d15h41m
(recursive via vd2-2 tunnel ::111.1.1.11), 1d15h41m, [1024/0]
```

To test the configuration on shortcut 1:

1. From VRF1 of spoke 1, ping VRF1 of spoke 2.
2. From VRF2 of spoke 1, ping VRF2 spoke 2. Both VRF1 and VRF2 source and destination IP addresses are the same, so you can see how the traffic is isolated.
3. Verify the session list:

```
# diagnose sys session6 list
session6 info: proto=58 proto_state=00 duration=3 expire=59 timeout=0 refresh_dir=both
flags=00000000 sockport=0 socktype=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=may_dirty
statistic(bytes/packets/allow_err): org=416/4/0 reply=416/4/0 tuples=2
tx speed(Bps/kbps): 136/1 rx speed(Bps/kbps): 136/1
orgin->sink: org pre->post, reply pre->post dev=100->223/223->100
hook=pre dir=org act=noop 2001:22::55:398->2001:33::44:128(:::0)
hook=post dir=reply act=noop 2001:33::44:398->2001:22::55:129(:::0)
src_mac=02:4c:a5:fc:77:6f
misc=0 policy_id=1 pol_uuid_idx=1070 auth_info=0 chk_client_info=0 vd=3:2
serial=0001104d tos=ff/ff ips_view=0 app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=61
rpdb_link_id=ff00003d ngfwid=n/a
npu_state=0x1040001 no_offload
no_ofld_reason:  disabled-by-policy non-npu-intf
total session6: 1
```

In the output, `vd=<vdom_ID>:<VRF_ID>` indicates that sessions are created in and stay in the corresponding VRFs.

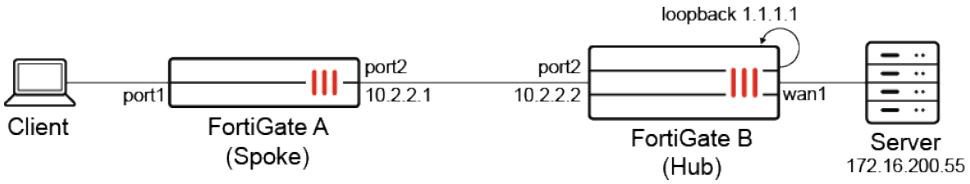
Copying the DSCP value from the session original direction to its reply direction

In an SD-WAN scenario when DSCP tags are used to mark traffic from the spoke to the hub, it is sometimes desirable for the hub to mark the reply traffic with the same DSSP tags. The `diffserv-copy` setting in firewall policy configurations allows the DSCP tag to be copied to the reply direction.

```
config firewall policy
    edit <id>
        set diffserv-copy {enable | disable}
    next
end
```

Example

The use cases in this example are for a hub and spoke SD-WAN deployment. Traffic from the spoke (either real traffic or SLA health check probes) can be marked with a certain DSCP tag when leaving the spoke. QoS may be applied by an upstream device based on the DSCP tag. When the traffic arrives on the hub, the hub may also want to mark the reply traffic to the spoke with the same DSCP tag. This would allow QoS to be applied to the traffic in the reply direction as well, which is traffic in the hub to spoke direction associated with the same session in the spoke to hub direction.



While this topology simplifies the SD-WAN deployment into a single hub and spoke, this feature applies to the following configurations:

- Multiple spokes (branch sites)
- One or more hubs (data center sites)
- Multiple overlays connecting spokes to hubs
- SD-WAN configured on spokes to pick the best overlay

Use case 1: typical forwarding traffic

Traffic originates from the spoke and is destined for a server behind the hub. The spoke marks the traffic with a DSCP tag of 101010. This is done by enabling `diffserv-foward` on the spoke firewall policy. It can also be accomplished by enabling `dscp-forward` in an SD-WAN rule.

The hub allows the traffic in through a firewall policy. By enabling `diffserv-copy` on the firewall policy, it will mark the reply traffic on the corresponding sessions with the same DSCP tag in which it came.

To configure the FortiGates:

1. Configure the firewall policy on the spoke (FortiGate A):

```
config firewall policy
    edit 1
        set srcintf "port1"
        set dstintf "virtual-wan-link"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all6"
        set dstaddr6 "all6"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set nat enable
        set diffserv-forward enable
        set diffservcode-forward 101010
```

```
    next
end
```

2. Configure the firewall policy on the hub (FortiGate B):

```
config firewall policy
    edit 3
        set srcintf "virtual-wan-link"
        set dstintf "wan1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set auto-asic-offload disable
        set nat enable
        set diffserv-copy enable
    next
end
```

To test the configuration:

1. Generate some forwarding traffic.
2. Verify that the session's tos value from the original direction is applied to the reply direction:

```
# diagnose sys session filter policy 3
# diagnose sys session filter dst 172.16.200.55
# diagnose sys session list

session info: proto=1 proto_state=00 duration=35 expire=59 timeout=0 flags=00000000 socktype=0
sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty f00
statistic(bytes/packets/allow_err): org=3024/36/1 reply=3024/36/1 tuples=2
tx speed(Bps/kbps): 82/0 rx speed(Bps/kbps): 82/0
orgin->sink: org pre->post, reply pre->post dev=20->17/17->20 gwy=172.16.200.55/10.2.2.1
hook=post dir=org act=snat 10.2.2.1:25290->172.16.200.55:8(172.16.200.2:25290)
hook=pre dir=reply act=dnat 172.16.200.55:25290->172.16.200.2:0(10.2.2.1:25290)
misc=0 policy_id=3 pol_uuid_idx=1097 auth_info=0 chk_client_info=0 vd=3
serial=0000a018 tos=6a/6a app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000001 no_offload
no_ofld_reason:  disabled-by-policy
total session 1
```

Use case 2: local-in traffic destined to a loopback interface

SLA health checks from the spoke are destined for a loopback interface on the hub. The health check is marked with a DSCP tag of 000001 by the spoke. When the hub receives the probes to its loopback, it will mark the replies with the same DSCP tags in which it came.

To configure the FortiGates:

1. Configure the health check on the spoke (FortiGate A):

```
config system sdwan
    config health-check
        edit "ping"
            set server "1.1.1.1"
            set diffservcode 000001
        set members 0
    next
end
```

2. Configure the loopback interface on the hub (FortiGate B):

```
config system interface
    edit "loopback"
        set vdom "vdom1"
        set ip 1.1.1.1 255.255.255.255
        set allowaccess ping https ssh  http telnet
        set type loopback
        set role lan
        set snmp-index 35
    next
end
```

3. Configure the firewall policy on the hub:

```
config firewall policy
    edit 1
        set srcintf "virtual-wan-link"
        set dstintf "loopback"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set auto-asic-offload disable
        set nat enable
        set diffserv-copy enable
    next
end
```

To test the configuration:

1. Generate some local-in traffic.
2. Verify that the session's tos value from the original direction is applied to the reply direction:

```
# diagnose sys session list

session info: proto=1 proto_state=00 duration=1 expire=59 timeout=0 flags=00000000 socktype=0
sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=log local may_dirty
statistic(bytes/packets/allow_err): org=80/2/1 reply=80/2/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->in, reply out->post dev=20->42/42->20 gwy=1.1.1.1/0.0.0.0
hook=pre dir=org act=noop 10.2.2.1:15->1.1.1.1:8(0.0.0.0:0)
hook=post dir=reply act=noop 1.1.1.1:15->10.2.2.1:0(0.0.0.0:0)
misc=0 policy_id=1 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=3
serial=0001a846 tos=41/41 app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  local disabled-by-policy
total session 1
```

Capture packets can also be used verify that the DSCP value from the original direction is applied to the reply direction.

Matching BGP extended community route targets in route maps

BGP extended community route targets can be matched in route maps. This can be applied in a scenario where the BGP route reflector receives routes from many VRFs, and instead of reflecting all routes from all VRFs, users only want to reflect routes based on a specific extended community route target.

To configure the extended community list:

```
config router extcommunity-list
    edit <name>
        set type {standard | expanded}
        config rule
            edit <id>
                set action {deny | permit}
                set type {rt | soo}
                set match <extended_community_specifications>
                set regexp <ordered_list_of_attributes>
```

```
            next
        end
    next
end
```

`type {standard | expanded}`	Set the extended community list type (standard or expanded).
`action {deny | permit}`	Deny or permit route-based operations based on the route's extended community attribute.
`type {rt | soo}`	Set the extended community type: • rt: route target • soo: site of origin
`match <extended_community_ specifications>`	Set the extended community specifications for matching a reserved extended community (community number in AA:NN format; use quotation marks complex expressions, `"123:234 345:456"`).
`regexp <ordered_list_of_ attributes>`	Set the ordered list of extended community attributes as a regular expression.

To configure the BGP extended community list in the route map:

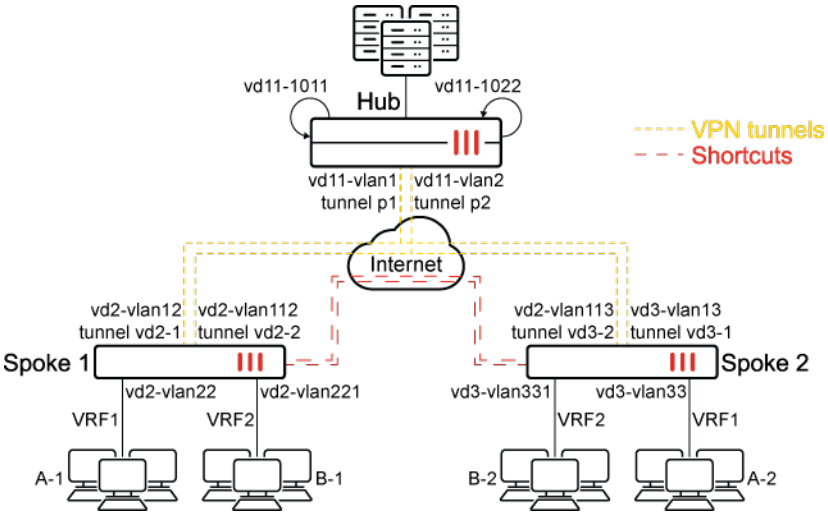
```
config router route-map
    edit <name>
        config rule
            edit <id>
                set match-extcommunity <list>
                set match-extcommunity-exact {enable | disable}
            next
        end
    next
end
```

`match-extcommunity <list>`	Set the BGP extended community list to match to.
`match-extcommunity-exact {enable | disable}`	Enable/disable exact matching of extended communities.

Example

In this example, multiple companies (or departments of a company) share the same hub and spoke VPN infrastructure. Company A and company B each have two branches in two different locations. The goal is for company A's branches (A-1 and A-2) to be able to communicate only with each other over VPN but not with company B's branches. Likewise, company B's branches (B-1 and B-2) can only communicate with each other over VPN but not with company A's. This is accomplished by placing each branch VLAN into their respective VRFs (VRF1 and VRF2), and encapsulating the VRF information within the VPN tunnel. The hub forms BGP peering with its neighbors, spoke 1 and spoke 2, over each IPsec overlay. The hub's BGP route reflector reflects the routes to the corresponding VRFs, allowing each spoke to form ADVPN shortcuts with the other spoke for each VRF.

However, in this scenario, we want A-1 and A-2 to use an ADVPN shortcut, but we do not want B-1 and B-2 to use ADVPN. A route map is configured on the hub to match the desired extended community route target number where only this route target is permitted, and others are denied. This allows the hub's BGP route reflector to only reflect routes associated with VRF1, allowing the spokes to form an ADVPN shortcut for VRF1. Routes associated with VRF2 are not reflected, and each spoke must route traffic through the hub to reach VRF2 on the other spoke.



Device	Interface	IP Address
Hub	vd11-1011	11.11.11.11
	vd11-1022	11.11.22.11
	vd11-vlan1 tunnel p1	11.1.1.1.11/24 10.10.100.254
	vd11-vlan2 tunnel p2	11.1.2.1.11/24 10.10.200.254
	vd11-vlan10	100.1.1.1/24
Spoke 1	vd2-vlan12 tunnel vd2-1	12.1.1.2 10.10.100.2
	vd2-vlan112 tunnel vd2-2	112.1.1.2 10.10.200.2
	vd2-vlan22	22.1.1.2/24
	vd2-vlan221	22.1.1.2/24
Spoke 2	vd3-vlan13 tunnel vd3-1	13.1.1.3 10.10.100.3
	vd2-vlan113 tunnel vd3-2	113.1.1.3 10.10.200.3

Device	Interface	IP Address
	vd3-vlan33	33.1.1.3/24
	vd3-vlan331	33.1.1.3/24

Configure the topology by following the instructions of Example 1 in SD-WAN segmentation over a single overlay on page 1495. Note that when checking the spoke 1 routes in example 1, there is a VRF2 route:

```
Spoke 1 # get router info routing-table bgp
…
Routing table for VRF=2
B V     33.1.1.0/24 [200/0] via 10.10.100.3 [2] (recursive via vd2-1 tunnel 11.1.1.11), 00:00:20,
[1/0]
                    [200/0] via 10.10.200.3 [2] (recursive via vd2-2 tunnel 11.1.2.11), 00:00:20,
[1/0]
```

The following procedure applies a route map on the hub's BGP configurations to limit route reflection to only routes matching the external community target of 1:1. This external community target corresponds to BGP paths for VRF1 learned from spoke 1 and spoke 2. The external community target of 2:1 corresponds to BGP paths for VRF2. By not explicitly permitting this target (2:1) in the community list and denying everything other than the permitted target (1:1) in the route map, the VRF2 BGP paths are essentially omitted from being reflected to the spokes.

To configure BGP filtering for an extended community route target on the hub:

1. Identify the external community target of VRF1 to be permitted:

```
# get router info bgp network 33.1.1.0/24
VRF 0 BGP routing table entry for 33.1.1.0/24
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
   11.1.1.1
  Advertised to peer-groups:
  gr1 gr2
…
VRF 1 BGP routing table entry for 33.1.1.0/24
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0 external duplicated
  Local, (Received from a RR-client)
    0.0.0.0 from 10.10.100.3 (3.3.3.3)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Extended Community: RT:1:1
      Receive Path ID: 1
      Advertised Path ID: 1
       Last update: Wed Aug 17 10:31:02 2022
  Original VRF 0 external duplicated
  Local, (Received from a RR-client)
    0.0.0.0 from 10.10.200.3 (3.3.3.3)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Extended Community: RT:1:1
```

```
      Receive Path ID: 1
      Advertised Path ID: 2
       Last update: Wed Aug 17 10:31:02 2022
VRF 2 BGP routing table entry for 33.1.1.0/24
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0 external duplicated
  Local, (Received from a RR-client)
    0.0.0.0 from 10.10.100.3 (3.3.3.3)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Extended Community: RT:2:1
      Receive Path ID: 1
      Advertised Path ID: 1
       Last update: Wed Aug 17 10:31:02 2022
  Original VRF 0 external duplicated
  Local, (Received from a RR-client)
    0.0.0.0 from 10.10.200.3 (3.3.3.3)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Extended Community: RT:2:1
      Receive Path ID: 1
      Advertised Path ID: 2
       Last update: Wed Aug 17 10:31:02 2022
```

2. Configure the extended community list:

```
config router extcommunity-list
    edit "extcomm1"
        config rule
            edit 1
                set action permit
                set match "1:1"
                set type rt
            next
        end
    next
end
```

3. Apply the extended community list to the route map:

```
config router route-map
    edit "rmp11"
        config rule
            edit 1
                set match-extcommunity "extcomm1"
            next
            edit 2
                set action deny
            next
        end
    next
end
```

4. Update the related BGP neighbor group settings:

```
config router bgp
    config neighbor-group
        edit "gr1"
            set route-map-out-vpnv4 "rmp11"
        next
        edit "gr2"
            set route-map-out-vpnv4 "rmp11"
        next
    end
end
```

5. Refresh the routes:

```
# execute router clear bgp all vpnv4 unicast out
```

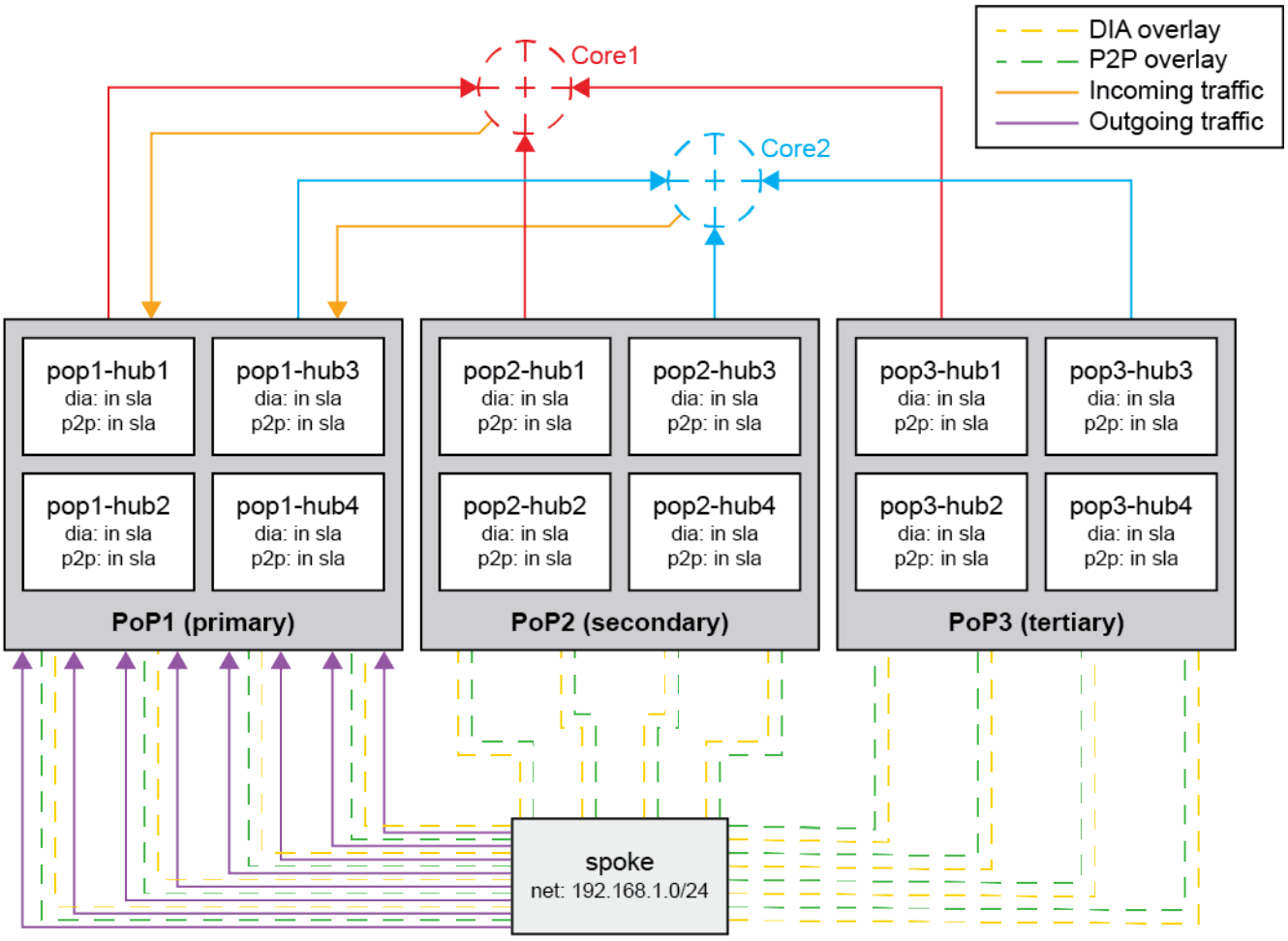
6. Check the spoke 1 routes. Since the extended community route target is applied, the VFR2 route does not appear in the BGP routing table:

```
# get router info routing-table bgp
Routing table for VRF=0
B*      0.0.0.0/0 [200/0] via 10.10.100.254 (recursive via vd2-1 tunnel 11.1.1.11), 03:47:50,
[1/0]
                  [200/0] via 10.10.200.254 (recursive via vd2-2 tunnel 11.1.2.11), 03:47:50,
[1/0]
B       1.1.1.1/32 [200/0] via 11.1.1.1 [2] (recursive via 12.1.1.1, vd2-vlan12), 03:47:50,
[1/0]
B       1.222.222.222/32 [200/0] via 11.1.1.1 [2] (recursive via 12.1.1.1, vd2-vlan12),
03:47:50, [1/0]
B       11.11.11.11/32 [200/0] via 10.10.100.254 (recursive via vd2-1 tunnel 11.1.1.11),
03:47:50, [1/0]
                       [200/0] via 10.10.200.254 (recursive via vd2-2 tunnel 11.1.2.11),
03:47:50, [1/0]
B       33.1.1.0/24 [200/0] via 10.10.100.254 [2] (recursive via vd2-1 tunnel 11.1.1.11),
03:47:21, [1/0]
                    [200/0] via 10.10.200.254 [2] (recursive via vd2-2 tunnel 11.1.2.11),
03:47:21, [1/0]

Routing table for VRF=1
B V     11.11.22.11/32 [200/0] via 10.10.100.254 (recursive via vd2-1 tunnel 11.1.1.11),
03:47:50, [1/0]
                       [200/0] via 10.10.200.254 (recursive via vd2-2 tunnel 11.1.2.11),
03:47:50, [1/0]
B V     33.1.1.0/24 [200/0] via 10.10.100.3 [2] (recursive via vd2-1 tunnel 11.1.1.11),
03:47:21, [1/0]
                    [200/0] via 10.10.200.3 [2] (recursive via vd2-2 tunnel 11.1.2.11),
03:47:21, [1/0]
B V     100.1.1.0/24 [200/0] via 10.10.100.254 (recursive via vd2-1 tunnel 11.1.1.11),
03:47:50, [1/0]
                     [200/0] via 10.10.200.254 (recursive via vd2-2 tunnel 11.1.2.11),
03:47:50, [1/0]
```

SD-WAN multi-PoP multi-hub large scale design and failover

FortiOS 7.2.0 introduced a feature to define the minimum number of SD-WAN interface members that must meet SLA in order for the spoke to select a hub to process its SD-WAN traffic. This design is suitable for a single-PoP multi-hub architecture in order to achieve hub-to-hub failover. See Using multiple members per SD-WAN neighbor configuration on page 1454.

In FortiOS 7.4.1 and later, the design is enhanced to support a multi-PoP multi-hub architecture in which incoming and outgoing traffic failover between PoPs is supported.



Based on the preceding diagram, incoming and outgoing traffic to the spoke is preferred over PoP1. If a single hub within PoP1 goes out of SLA, traffic will continue to flow through the PoP. If the minimum number of members to meet SLA in the PoP cannot be met, then traffic will fail over to PoP2.

The following enhancements have been made to support the multi-PoP failover scenario.

- Add `minimum-sla-meet-members` setting in the SD-WAN zone configurations and `zone-mode` setting in the SD-WAN service configurations:

```
config system sdwan
    config zone
        edit <name>
            set minimum-sla-meet-members <integer>
        next
    end
    config service
        edit <id>
            set mode sla
            set zone-mode {enable | disable}
        next
    end
end
```

When `zone-mode` is enabled on an SD-WAN service rule, the traffic is steered based on the status of the zone.

The state of the health check referenced in the SD-WAN service can be defined as follows:

- If the number of in SLA members in a zone is less than the `minimum-sla-meet-members`, then the zone's state is out of SLA; otherwise, it is in SLA.
- If a zone's state is out of SLA, then all members in the zone are out of SLA.
- If a zone's state is in SLA, then the health check's state of individual members in the zone is determined by its own state.

- Add `service-id` setting in the SD-WAN neighbor configurations:

```
config system sdwan
    config neighbor
        edit <bgp_neighbor_ip>
            set member <member_id>
            set service-id <id>
        next
    end
end
```

The SD-WAN neighbor's behavior can be determined by SD-WAN service and naturally synchronizes with SD-WAN service.

- The SD-WAN service defines priority zones, whose SLA state determines the advertised community preferable string.
- The SD-WAN service defines the `hold-down-time`, which determines how long an advertised community preferable string can be kept when it is expected to be changed.

- Add `sla-stickness` setting in the SD-WAN service configurations:

```
config system sdwan
    config service
        edit <id>
            set mode sla
            set sla-stickiness {enable | disable}
        next
    end
end
```

The switch-over of an existing session is determined as follows:

  - If the outgoing interface of the session is in SLA, then the session can keep its outgoing interface.
  - Otherwise, the session switches to a preferable path if one exists.
- Allow the neighbor group to be configured in the SD-WAN neighbor configurations:

```
config system sdwan
    config neighbor
        edit <bgp_neighbor_group>
            set member <member_id>
            set health-check <name>
            set sla-id <id>
        next
    end
end
```

Outgoing path control

The outgoing path from spoke to hub operates as follows:

1. Overlays to the primary and secondary PoP are assigned separately into an SD-WAN primary and secondary zone on the spoke.
2. One SD-WAN service rule is defined to include these zones as SD-WAN members.
3. When the primary zone is in SLA (`minimum-sla-meet-members` is met), the SD-WAN service rule steers traffic to the in SLA overlay members.
4. When the primary zone is out of SLA (`minimum-sla-meet-members` is not met), the SD-WAN service rule steers traffic to the in SLA overlay members in the secondary zone.
5. When the primary zone SLA is recovered:
   a. If `sla-stickness` is disabled on the SD-WAN service rule, then traffic will wait the duration of the `hold-down-time` before switching back to in SLA overlays in the primary zone.
   b. If `sla-stickness` is enabled on the SD-WAN service rule, then existing traffic will be kept on the in SLA overlays on the secondary zone, but new traffic will be steered to in SLA overlays in the primary zone.

Incoming path control

The incoming traffic from the core/external peers, to PoP, to spoke operates as follows:

1. When the primary zone is in SLA, the spoke uses the preferable route map to advertise local routes with the in SLA community to hubs in the primary and secondary PoPs.
   a. Hubs in the primary PoP translate the in SLA community into a short AS path and advertise it to external peers to attract incoming traffic.
   b. Hubs in the secondary PoP translate the in SLA community into a longer AS path and advertise it to external peers to deflect incoming traffic.
2. If the number of in SLA overlays in the primary zone is less than the `minimum-sla-meet-members`, then the spoke will use the default route map to advertise routes instead of with an out of SLA community to hubs in the primary PoP.
   a. Hubs in the primary PoP translate the out of SLA community into a longest AS path, and advertise it to external peers to deflect incoming traffic.
   b. As a result, inbound traffic is routed to hubs in the secondary PoP.

3. When the primary zone SLA is recovered:
   a. The spoke will wait the duration of the predefined `hold-down-time` in the SD-WAN service rule to use the preferable route map again to advertise routes with the in SLA community to hubs in the primary PoP.
   b. As a result, inbound traffic will be routed back to hubs in the primary PoP.

Neighbor group configuration

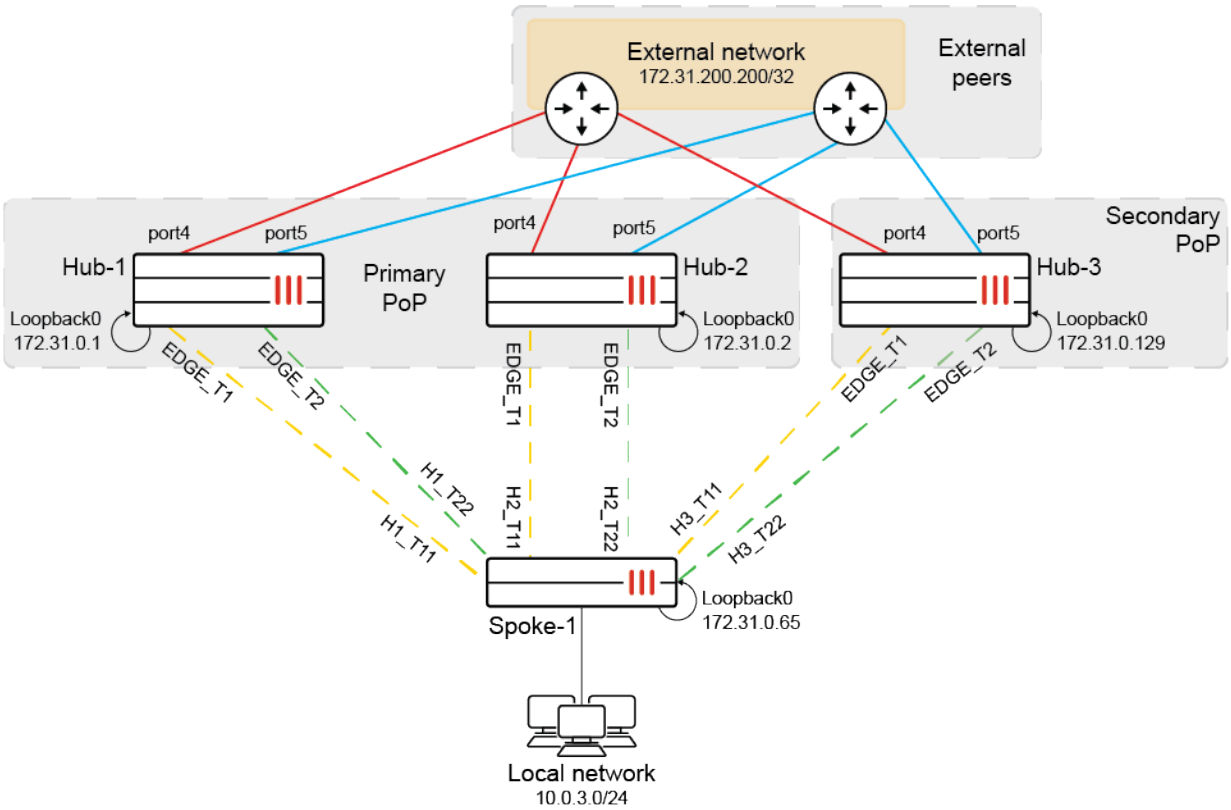
By configuring the neighbor group for spokes under the hub's SD-WAN neighbor configuration, if all paths from the hub to external peers are detected as out of SLA, then the hub will use the default route map to deny external routes to spokes that belong to this neighbor group defined on the hub. As a result, spokes will skip that specific hub and connect to external peers from other hubs.

This allows spokes to only measure overlay quality to each hub, and hubs to manage health checks to services by external peers. This significantly decreases the number of health check probes directly from the spoke to services and decreases the overall complexity. The complexity is further simplified by using multiple VRFs or segmentation where each spoke needs to send health check probes.

Example

This example configuration contains the following components:

- Two PoPs:
  - The primary PoP has two hubs (Hub-1 and Hub-2).
  - The secondary PoP has one hub (Hub-3).
- Spoke-1 has six overlays, with two overlay connections to each hub.
- Spoke-1 has three BGP neighbors, with one BGP neighbor for each hub.
  - All BGP neighbors are established on loopback IPs.
- Each hub has two paths to external peers.



Normally, outbound and inbound traffic go through hubs in the primary PoP. If the number of in SLA overlays to the primary PoP is less than the `minimum-sla-meet-members` (set to 2 in this example), bi-directional traffic needs to be switched to hubs in the secondary PoP. But when the primary PoP recovers and the `minimum-sla-meet-members` is met again, bi-directional traffic is forced back to hubs in the primary PoP after the predefined `hold-down-time` duration.

The hubs do not require SD-WAN configurations to the spokes. However, they use SD-WAN for connections to external peer routers.

Configuring the FortiGates

The following configurations highlight important routing and SD-WAN settings that must be configured on the spoke and the hubs. It is assumed that other configurations such as underlays, IPsec VPN overlays, loopbacks, static routes, and so on are already configured.

To configure Spoke-1:

1. Create the primary (PoP1) and secondary (PoP2) zones, and set the `minimum-sla-meet-members` to 2 on PoP1:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
        edit "PoP1"
```

```
            set minimum-sla-meet-members 2
        next
        edit "PoP2"
        next
    end
end
```

2. Add the overlay members to each zone. Four overlays are defined for PoP1, and two overlays are defined for PoP2:

```
config system sdwan
    config members
        edit 1
            set interface "H1_T11"
            set zone "PoP1"
        next
        edit 2
            set interface "H1_T22"
            set zone "PoP1"
        next
        edit 3
            set interface "H2_T11"
            set zone "PoP1"
        next
        edit 4
            set interface "H2_T22"
            set zone "PoP1"
        next
        edit 5
            set interface "H3_T11"
            set zone "PoP2"
        next
        edit 6
            set interface "H3_T22"
            set zone "PoP2"
        next
    end
end
```

3. Configure a performance SLA health check to a probe server behind the three hubs:

```
config system sdwan
    config health-check
        edit "Hubs"
            set server "172.31.100.100"
            set source 172.31.0.65
            set members 0
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 200
                next
```

```
            end
        next
    end
end
```

4. Configure the service rule with the following settings: use SLA mode, enable zone mode to steer traffic based on the zone statuses, enable `sla-stickiness`, and use a 30-second hold down so that upon a recovery, existing sessions will remain on the secondary PoP while new sessions will switch back to the primary PoP once the 30-second duration ends:

```
config system sdwan
    config service
        edit 1
            set mode sla
            set zone-mode enable
            set dst "all"
            set src "CORP_LAN"
            set hold-down-time 30
            set sla-stickiness enable
            config sla
                edit "Hubs"
                    set id 1
                next
            end
            set priority-zone "PoP1" "PoP2"
        next
    end
end
```

Since the PoP1 zone is specified before PoP2, PoP1 is regarded as the primary and preferred over the PoP2 zone.

5. Configure the in_sla and out_sla route maps that define the communities that are advertised to the hub when the zones are in and out of SLA.
   a. Configure the access list:

```
config router access-list
    edit "net10"
        config rule
            edit 1
                set prefix 10.0.3.0 255.255.255.0
            next
        end
    next
end
```

   b. Configure the route maps:

```
config router route-map
    edit "in_sla"
        config rule
            edit 1
```

```
                set match-ip-address "net10"
                set set-community "10:1"
            next
        end
    next
    edit "out_sla"
        config rule
            edit 1
                set match-ip-address "net10"
                set set-community "10:2"
            next
        end
    next
end
```

6. Configure the default route map for out of SLA scenarios, preferable route map for in SLA scenarios, and the local network to be advertised:

```
config router bgp
    config neighbor
        edit "172.31.0.1"
            ...
            set route-map-out "out_sla"
            set route-map-out-preferable "in_sla"
            ...
        next
        edit "172.31.0.2"
            ...
            set route-map-out "out_sla"
            set route-map-out-preferable "in_sla"
            ...
        next
        edit "172.31.0.129"
            ...
            set route-map-out "out_sla"
            set route-map-out-preferable "in_sla"
            ...
        next
    end
    config network
        edit 1
            set prefix 10.0.3.0 255.255.255.0
        next
    end
    ...
end
```

7. Define SD-WAN neighbors for each hub. The `minimum-sla-meet-members` is configured for the Hub-1 neighbor so that bi-directional traffic goes through Hub-1 as long as the in SLA overlays to Hub-1 are no less than 1. Associate the previously defined service rule to each SD-WAN neighbor:

```
config system sdwan
    config neighbor
        edit "172.31.0.1"
            set member 1 2
            set minimum-sla-meet-members 1
            set service-id 1
        next
        edit "172.31.0.2"
            set member 3 4
            set service-id 1
        next
        edit "172.31.0.129"
            set member 5 6
            set service-id 1
        next
    end
end
```

To configure the hubs:

1. Configure the SD-WAN zone, members, and health check for the external connections to peer routers. Performance SLA health checks are sent to external servers in order to measure the health of the external connections:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "port4"
        next
        edit 2
            set interface "port5"
        next
    end
    config health-check
        edit "external_peers"
            set server "10.0.1.2"
            set members 1 2
            config sla
                edit 1
                    set link-cost-factor latency
                    set latency-threshold 200
                next
            end
        next
    end
end
```

2. Configure the route maps for in and out of SLA scenarios. When out of SLA (one of the external connections is down), external routes are denied to be advertised to the spokes that are part of the neighbor group.
   a. Configure the access list:

```
config router access-list
    edit "net_Lo"
        config rule
            edit 1
                set prefix 172.31.200.200 255.255.255.255
            next
        end
    next
end
```

   b. Configure the route maps:

```
config router route-map
    edit "in_sla"
        config rule
            edit 1
                set match-ip-address "net_Lo"
            next
        end
    next
    edit "out_sla"
        config rule
            edit 1
                set action deny routes
                set match-ip-address "net_Lo"
            next
        end
    next
end
```

3. In the BGP settings, configure the external network prefix to advertise. Then configure the neighbor group and neighbor range for the spokes. Configure the preferable and default route maps to define the behavior when the external connections are in and out of SLA:

```
config router bgp
    ...
    config network
        edit 1
            set prefix 172.31.200.200 255.255.255.255
        next
    end
    config neighbor-group
        edit "EDGE"
            ...
            set route-map-out "out_sla"
            set route-map-out-preferable "in_sla"
            ...
        next
```

```
    end
    config neighbor-range
        edit 1
            set prefix 172.31.0.64 255.255.255.192
            set neighbor-group "EDGE"
        next
    end
    ...
end
```

4. Configure the SD-WAN neighbor to match the neighbor group that includes spokes as members. Specify that at least one of the external peer connections needs to be up to be considered in SLA:

```
config system sdwan
    config neighbor
        edit "EDGE"
            set member 1 2
            set minimum-sla-meet-members 1
            set health-check "external_peers"
            set sla-id 1
        next
    end
end
```

Testing and verification

The following tests use diagnostic commands on various FortiGates to verify the connections in the SD-WAN configuration.

Test case 1: the primary PoP and Hub-1 are in SLA

To verify the configuration:

1. Verify the SD-WAN service rules status on Spoke-1. When all six overlays are in SLA on Spoke-1, the primary PoP and primary zone PoP1 are preferred. In particular, the overlay H1_T11 over PoP1 is preferred:

```
Spoke-1 (root) # diagnose sys sdwan service4
Service(1): Address Mode(IPV4) flags=0x1c200 use-shortcut-sla use-shortcut sla-stickiness
 Tie break: cfg
  Gen(1), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-compare-order
Hold down time(30) seconds, Hold start at 362646 second, now 362646
  Service role: standalone
  Members(6):
    1: Seq_num(1 H1_T11 PoP1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), selected
    2: Seq_num(2 H1_T22 PoP1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), selected
    3: Seq_num(3 H2_T11 PoP1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), selected
    4: Seq_num(4 H2_T22 PoP1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), selected
    5: Seq_num(5 H3_T11 PoP2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected
    6: Seq_num(6 H3_T22 PoP2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected
  Src address(1):
        10.0.0.0-10.255.255.255
```

```
  Dst address(1):
        0.0.0.0-255.255.255.255
```

2. Verify the BGP learned routes on Hub-1. The local route with in SLA community 10:1 is advertised to all hubs. Though, the AS paths on Hub-1 and Hub-2 are shorter than Hub-3:

```
PoP1-Hub1 (root) # get router info bgp network 10.0.3.0/24
VRF 0 BGP routing table entry for 10.0.3.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local, (Received from a RR-client)
    172.31.0.65 from 172.31.0.65 (172.31.0.65)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Community: 10:1
      Last update: Mon Jul 17 15:16:57 2023
```

3. Send traffic from a host behind Spoke-1 to 172.31.200.200.
4. Run a sniffer trace on Spoke-1. Traffic leaves and returns on the H1_T11 overlay :

```
Spoke-1 (root) # diagnose sniffer packet any 'host 172.31.200.200' 4
interfaces=[any]
filters=[host 172.31.200.200]
5.098248 port4 in 10.0.3.2 -> 172.31.200.200: icmp: echo request
5.098339 H1_T11 out 10.0.3.2 -> 172.31.200.200: icmp: echo request
5.098618 H1_T11 in 172.31.200.200 -> 10.0.3.2: icmp: echo reply
5.098750 port4 out 172.31.200.200 -> 10.0.3.2: icmp: echo reply
```

Test case 2: a single SD-WAN member on Hub-1 is out of SLA

Hub-1 and PoP1 are still preferred in this scenario.

To verify the configuration:

1. Verify the health check status on Spoke-1. The H1_T11 overlay on Hub-1/PoP1 is out of SLA:

```
Spoke-1 (root) # diagnose sys sdwan health-check
Health Check(Hubs):
Seq(1 H1_T11): state(alive), packet-loss(0.000%) latency(220.214), jitter(0.015), mos(4.104),
bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x0
Seq(2 H1_T22): state(alive), packet-loss(0.000%) latency(0.196), jitter(0.014), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
Seq(3 H2_T11): state(alive), packet-loss(0.000%) latency(0.173), jitter(0.008), mos(4.404),
bandwidth-up(999998), bandwidth-dw(999997), bandwidth-bi(1999995) sla_map=0x1
…
```

2. Verify the SD-WAN neighbor status. The SD-WAN neighbor still displays Hub-1's zone status as pass/alive:

```
Spoke-1 (root) # diagnose sys sdwan neighbor
SD-WAN neighbor status: hold-down(disable), hold-down-time(0), hold_boot_time(0)
        Selected role(standalone) last_secondary_select_time/current_time in seconds 0/436439
Neighbor(172.31.0.1): member(1 2)role(standalone)
```

```
        Health-check(:0)  sla-pass selected alive
Neighbor(172.31.0.2): member(3 4)role(standalone)
        Health-check(:0)  sla-pass selected alive
Neighbor(172.31.0.129): member(5 6)role(standalone)
        Health-check(:0)  sla-pass selected alive
```

3. Verify the SD-WAN service rules status. Spoke-1 steers traffic to the H1_T22 overlay through Hub-1:

```
Spoke-1 (root) # diagnose sys sdwan service4
Service(1): Address Mode(IPV4) flags=0x1c200 use-shortcut-sla use-shortcut sla-stickiness
 Tie break: cfg
  Gen(2), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-compare-order
Hold down time(30) seconds, Hold start at 364162 second, now 364162
  Service role: standalone
  Members(6):
    1: Seq_num(2 H1_T22 PoP1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), selected
    2: Seq_num(3 H2_T11 PoP1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), selected
    3: Seq_num(4 H2_T22 PoP1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), selected
    4: Seq_num(5 H3_T11 PoP2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected
    5: Seq_num(6 H3_T22 PoP2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected
    6: Seq_num(1 H1_T11 PoP1), alive, sla(0x0), gid(0), cfg_order(0), local cost(0), selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255
```

4. Verify the BGP learned routes on Hub-1. The hubs continue to receive community 10:1 from the spoke and continue to route incoming traffic through Hub-1:

```
PoP1-Hub1 (root) # get router info bgp network 10.0.3.0/24
VRF 0 BGP routing table entry for 10.0.3.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local, (Received from a RR-client)
    172.31.0.65 from 172.31.0.65 (172.31.0.65)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Community: 10:1
      Last update: Mon Jul 17 15:16:57 2023
```

5. Send traffic from a host behind Spoke-1 to 172.31.200.200.
6. Run a sniffer trace on Spoke-1. Traffic leaves and returns on the H1_T22 overlay:

```
Spoke-1 (root) # diagnose sniffer packet any 'host 172.31.200.200' 4
interfaces=[any]
filters=[host 172.31.200.200]
25.299006 port4 in 10.0.3.2 -> 172.31.200.200: icmp: echo request
25.299080 H1_T22 out 10.0.3.2 -> 172.31.200.200: icmp: echo request
25.299323 H1_T22 in 172.31.200.200 -> 10.0.3.2: icmp: echo reply
25.299349 port4 out 172.31.200.200 -> 10.0.3.2: icmp: echo reply
```

Test case 3: both SD-WAN members on Hub-1 are out of SLA

Other in SLA overlays in zone PoP1 though Hub-2 are still preferred over PoP2 in this scenario.

To verify the configuration:

1. Verify the health check status on Spoke-1. Both H1_T11 and H1_T22 overlays on Hub-1/PoP1 are out of SLA:

```
Spoke-1 (root) # diagnose sys sdwan health-check
Health Check(Hubs):
Seq(1 H1_T11): state(alive), packet-loss(0.000%) latency(220.220), jitter(0.018), mos(4.103),
bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x0
Seq(2 H1_T22): state(alive), packet-loss(0.000%) latency(220.174), jitter(0.007), mos(4.104),
bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x0
Seq(3 H2_T11): state(alive), packet-loss(0.000%) latency(0.184), jitter(0.015), mos(4.404),
bandwidth-up(999998), bandwidth-dw(999997), bandwidth-bi(1999995) sla_map=0x1
Seq(4 H2_T22): state(alive), packet-loss(0.000%) latency(0.171), jitter(0.008), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
Seq(5 H3_T11): state(alive), packet-loss(0.000%) latency(0.173), jitter(0.011), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
Seq(6 H3_T22): state(alive), packet-loss(0.000%) latency(0.179), jitter(0.011), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x1
```

2. Verify the SD-WAN neighbor status. The SD-WAN neighbor displays Hub-1's zone status as failed. However, SD-WAN Hub-2 is pass/alive:

```
Spoke-1 (root) # diagnose sys sdwan neighbor
SD-WAN neighbor status: hold-down(disable), hold-down-time(0), hold_boot_time(0)
        Selected role(standalone) last_secondary_select_time/current_time in seconds 0/436535
Neighbor(172.31.0.1): member(1 2)role(standalone)
        Health-check(:0)  sla-fail alive
Neighbor(172.31.0.2): member(3 4)role(standalone)
        Health-check(:0)  sla-pass selected alive
Neighbor(172.31.0.129): member(5 6)role(standalone)
        Health-check(:0)  sla-pass selected alive
```

3. Verify the SD-WAN service rules status. Spoke-1 steers traffic to the H2_T11 overlay through Hub-2:

```
Spoke-1 (root) # diagnose sys sdwan service4
Service(1): Address Mode(IPV4) flags=0x1c200 use-shortcut-sla use-shortcut sla-stickiness
 Tie break: cfg
  Gen(3), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-compare-order
Hold down time(30) seconds, Hold start at 364489 second, now 364490
  Service role: standalone
  Members(6):
    1: Seq_num(3 H2_T11 PoP1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), selected
    2: Seq_num(4 H2_T22 PoP1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), selected
    3: Seq_num(5 H3_T11 PoP2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected
    4: Seq_num(6 H3_T22 PoP2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected
    5: Seq_num(1 H1_T11 PoP1), alive, sla(0x0), gid(0), cfg_order(0), local cost(0), selected
    6: Seq_num(2 H1_T22 PoP1), alive, sla(0x0), gid(0), cfg_order(0), local cost(0), selected
  Src address(1):
        10.0.0.0-10.255.255.255
```

```
  Dst address(1):
        0.0.0.0-255.255.255.255
```

4. Verify the BGP learned routes on Hub-1 and Hub-2. Hub-2 and Hub-3 continue to receive community 10:1 from Spoke-1, but Hub-1 receives the out of SLA community of 10:2.
   a. On Hub-1:

```
PoP1-Hub1 (root) # get router info bgp network 10.0.3.0/24
VRF 0 BGP routing table entry for 10.0.3.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local, (Received from a RR-client)
    172.31.0.65 from 172.31.0.65 (172.31.0.65)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Community: 10:2
      Last update: Mon Jul 17 18:08:58 2023
```

   b. On Hub-2:

```
PoP1-Hub2 (root) # get router info bgp network 10.0.3.0/24
VRF 0 BGP routing table entry for 10.0.3.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local, (Received from a RR-client)
    172.31.0.65 from 172.31.0.65 (172.31.0.65)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Community: 10:1
      Last update: Mon Jul 17 15:31:43 2023
```

5. Send traffic from a host behind Spoke-1 to 172.31.200.200.
6. Run a sniffer trace on Spoke-1. Traffic leaves and returns on the H2_T11 overlay:

```
Spoke-1 (root) # diagnose sniffer packet any 'host 172.31.200.200' 4
interfaces=[any]
filters=[host 172.31.200.200]
13.726009 port4 in 10.0.3.2 -> 172.31.200.200: icmp: echo request
13.726075 H2_T11 out 10.0.3.2 -> 172.31.200.200: icmp: echo request

13.726354 H2_T11 in 172.31.200.200 -> 10.0.3.2: icmp: echo reply
13.726382 port4 out 172.31.200.200 -> 10.0.3.2: icmp: echo reply
```

Test case 4: three SD-WAN members on PoP1 are out of SLA

The number of in SLA overlays in zone PoP1 is less than the `minimum-sla-meet-members` in zone PoP1. The SD-WAN service rule for Hub-2 is forcibly marked as `sla(0x0)` or out of SLA.

To verify the configuration:

1. Verify the health check status on Spoke-1. All three H1_T11, H1_T22, and H2_T11 overlays on PoP1 are out of SLA:

```
Spoke-1 (root) # diagnose sys sdwan health-check
Health Check(Hubs):
Seq(1 H1_T11): state(alive), packet-loss(0.000%) latency(220.219), jitter(0.019), mos(4.103),
bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x0
Seq(2 H1_T22): state(alive), packet-loss(0.000%) latency(220.184), jitter(0.008), mos(4.104),
bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x0
Seq(3 H2_T11): state(alive), packet-loss(0.000%) latency(220.171), jitter(0.009), mos(4.104),
bandwidth-up(999998), bandwidth-dw(999997), bandwidth-bi(1999995) sla_map=0x0
Seq(4 H2_T22): state(alive), packet-loss(0.000%) latency(0.180), jitter(0.013), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
Seq(5 H3_T11): state(alive), packet-loss(0.000%) latency(0.174), jitter(0.014), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
Seq(6 H3_T22): state(alive), packet-loss(0.000%) latency(0.179), jitter(0.015), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999998), bandwidth-bi(1999997) sla_map=0x1
```

2. Verify the SD-WAN neighbor status. The SD-WAN neighbor displays Hub-1 and Hub-2's zone status as failed:

```
Spoke-1 (root) # diagnose sys sdwan neighbor
SD-WAN neighbor status: hold-down(disable), hold-down-time(0), hold_boot_time(0)
        Selected role(standalone) last_secondary_select_time/current_time in seconds 0/436605
Neighbor(172.31.0.1): member(1 2)role(standalone)
        Health-check(:0)  sla-fail alive
Neighbor(172.31.0.2): member(3 4)role(standalone)
        Health-check(:0)  sla-fail alive
Neighbor(172.31.0.129): member(5 6)role(standalone)
        Health-check(:0)  sla-pass selected alive
```

3. Verify the SD-WAN service rules status. Since the minimum SLA members is not met for the primary zone (PoP1), the remaining overlay in PoP1 associated with the SD-WAN service rule is forcibly set to out of SLA. Spoke-1 steers traffic to the H3_T11 overlay through Hub-3:

```
Spoke-1 (root) # diagnose sys sdwan service4
Service(1): Address Mode(IPV4) flags=0x1c200 use-shortcut-sla use-shortcut sla-stickiness
 Tie break: cfg
  Gen(6), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-compare-order
Hold down time(30) seconds, Hold start at 365341 second, now 365341
  Service role: standalone
  Members(6):
    1: Seq_num(5 H3_T11 PoP2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected
    2: Seq_num(6 H3_T22 PoP2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected
    3: Seq_num(1 H1_T11 PoP1), alive, sla(0x0), gid(0), cfg_order(0), local cost(0), selected
    4: Seq_num(2 H1_T22 PoP1), alive, sla(0x0), gid(0), cfg_order(0), local cost(0), selected

    5: Seq_num(3 H2_T11 PoP1), alive, sla(0x0), gid(0), cfg_order(0), local cost(0), selected
    6: Seq_num(4 H2_T22 PoP1), alive, sla(0x0), gid(0), cfg_order(0), local cost(0), selected
  Src address(1):
        10.0.0.0-10.255.255.255
  Dst address(1):
        0.0.0.0-255.255.255.255
```

4. Verify the BGP learned routes on each hub. Hub-3 continues to receive community 10:1 from Spoke-1, but Hub-1 and Hub-2 receive the out of SLA community of 10:2.
   a. On Hub-1:

```
PoP1-Hub1 (root) # get router info bgp network 10.0.3.0/24
VRF 0 BGP routing table entry for 10.0.3.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local, (Received from a RR-client)
    172.31.0.65 from 172.31.0.65 (172.31.0.65)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Community: 10:2
      Last update: Mon Jul 17 18:22:14 2023
```

   b. On Hub-2:

```
PoP1-Hub2 (root) # get router info bgp network 10.0.3.0/24
VRF 0 BGP routing table entry for 10.0.3.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local, (Received from a RR-client)
    172.31.0.65 from 172.31.0.65 (172.31.0.65)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Community: 10:2
      Last update: Mon Jul 17 18:37:53 2023
```

   c. On Hub-3:

```
PoP2-Hub3 (root) # get router info bgp network 10.0.3.0/24
VRF 0 BGP routing table entry for 10.0.3.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
  Original VRF 0
  Local, (Received from a RR-client)
    172.31.0.65 from 172.31.0.65 (172.31.0.65)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Community: 10:1
      Last update: Mon Jul 17 14:39:04 2023
```

5. Send traffic from a host behind Spoke-1 to 172.31.200.200.
6. Run a sniffer trace on Spoke-1. Traffic leaves and returns on the H3_T11 overlay:

```
Spoke-1 (root) # diagnose sniffer packet any 'host 172.31.200.200' 4
interfaces=[any]
filters=[host 172.31.200.200]
38.501449 port4 in 10.0.3.2 -> 172.31.200.200: icmp: echo request
38.501519 H3_T11 out 10.0.3.2 -> 172.31.200.200: icmp: echo request
38.501818 H3_T11 in 172.31.200.200 -> 10.0.3.2: icmp: echo reply
38.501845 port4 out 172.31.200.200 -> 10.0.3.2: icmp: echo reply
```

Test case 5: an SD-WAN member on PoP1 recovers

SD-WAN member H2_T11 recovers and brings the number of overlays in SLA back to being above the `minimum-sla-meet-members` threshold in PoP1. After the hold down time duration (30 seconds), in SLA overlays in zone PoP1 are preferred over PoP2 again. With `sla-stickiness` enabled, existing traffic is kept on H3_T11, but new traffic is steered to H2_T11.

To verify the configuration:

1. Verify the SD-WAN service rules status on Spoke-1. The hold down timer has not yet passed, so H2_T11 is not yet preferred—even though the SLA status is pass/alive:

```
Spoke-1 (root) # diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x1c200 use-shortcut-sla use-shortcut sla-stickiness
 Tie break: cfg
  Gen(16), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-compare-
order
Hold down time(30) seconds, Hold start at 431972 second, now 432000
  Service role: standalone
  Members(6):
    1: Seq_num(5 H3_T11 PoP2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected
    2: Seq_num(6 H3_T22 PoP2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected
    3: Seq_num(1 H1_T11 PoP1), alive, sla(0x0), gid(0), cfg_order(0), local cost(0), selected
    4: Seq_num(2 H1_T22 PoP1), alive, sla(0x0), gid(0), cfg_order(0), local cost(0), selected
    5: Seq_num(3 H2_T11 PoP1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), selected
    6: Seq_num(4 H2_T22 PoP1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), selected
```

2. Verify the SD-WAN service rules status again after the hold down timer passes. H2_T11 and H2_T22 from PoP1 are now preferred:

```
Spoke-1 (root) # diagnose sys sdwan service4
Service(1): Address Mode(IPV4) flags=0x1c200 use-shortcut-sla use-shortcut sla-stickiness
 Tie break: cfg
  Gen(17), TOS(0x0/0x0), Protocol(0): src(1->65535):dst(1->65535), Mode(sla), sla-compare-
order
Hold down time(30) seconds, Hold start at 432003 second, now 432003
  Service role: standalone
  Members(6):
    1: Seq_num(3 H2_T11 PoP1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), selected
    2: Seq_num(4 H2_T22 PoP1), alive, sla(0x1), gid(0), cfg_order(0), local cost(0), selected
    3: Seq_num(5 H3_T11 PoP2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected
    4: Seq_num(6 H3_T22 PoP2), alive, sla(0x1), gid(0), cfg_order(1), local cost(0), selected
    5: Seq_num(1 H1_T11 PoP1), alive, sla(0x0), gid(0), cfg_order(0), local cost(0), selected
    6: Seq_num(2 H1_T22 PoP1), alive, sla(0x0), gid(0), cfg_order(0), local cost(0), selected
```

3. Verify the BGP learned routes on Hub-2, which now receives community 10:1 from Spoke-1:

```
PoP1-Hub2 (root) #  get router info bgp network 10.0.3.0/24
VRF 0 BGP routing table entry for 10.0.3.0/24
Paths: (1 available, best #1, table Default-IP-Routing-Table)
  Not advertised to any peer
```

```
  Original VRF 0
  Local, (Received from a RR-client)
    172.31.0.65 from 172.31.0.65 (172.31.0.65)
      Origin IGP metric 0, localpref 100, valid, internal, best
      Community: 10:1
      Last update: Tue Jul 18 14:41:32 2023
```

4. Send traffic from a host behind Spoke-1 to 172.31.200.200.
5. Run a sniffer trace on Spoke-1. Because of `sla-stickiness`, the existing traffic is kept on H3_T11:

```
Spoke-1 (root) # diagnose sniffer packet any 'host 172.31.200.200' 4
interfaces=[any]
filters=[host 172.31.200.200]

0.202708 port4 in 10.0.3.2 -> 172.31.200.200: icmp: echo request
0.202724 H3_T11 out 10.0.3.2 -> 172.31.200.200: icmp: echo request
0.202911 H3_T11 in 172.31.200.200 -> 10.0.3.2: icmp: echo reply
0.202934 port4 out 172.31.200.200 -> 10.0.3.2: icmp: echo reply
```

Test case 6: Hub-1 has an in SLA path to external peers

Since Hub-1 has an in SLA path to external peers, it will advertise the external route with destination 172.31.200.200/32 to Spoke-1.

To verify the configuration:

1. Verify the health check status on Hub-1. Note that port4 meets SLA, but port5 does not:

```
PoP1-Hub1 (root) # diagnose sys sdwan health-check
Health Check(external_peers):
Seq(1 port4): state(alive), packet-loss(0.000%) latency(0.161), jitter(0.009), mos(4.404),
bandwidth-up(999999), bandwidth-dw(999999), bandwidth-bi(1999998) sla_map=0x1
Seq(2 port5): state(dead), packet-loss(100.000%) sla_map=0x0
```

2. Verify the SD-WAN neighbor status. The `minimum-sla-meet-members` threshold of 1 is still met:

```
PoP1-Hub1 (root) # diagnose sys sdwan neighbor
Neighbor(EDGE): member(1 2)role(standalone)
        Health-check(external_peers:1)  sla-pass selected alive
```

3. Verify the BGP learned routes. Hub-1 still advertises the external route to the Spoke-1 BGP neighbor:

```
PoP1-Hub1 (root) # get router info bgp neighbors 172.31.0.65 advertised-routes
VRF 0 BGP table version is 13, local router ID is 172.31.0.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete
   Network                Next Hop      Metric     LocPrf  Weight  RouteTag Path
*>i172.31.200.200/32     172.31.0.1     100        32768        0          i <-/->
Total number of prefixes 1
```

Test case 7: all external peers on Hub-1 are out of SLA

In this case, Hub-1 will now advertise the default route map, which denies the advertisement of the external route. Spoke-1 will now route traffic to the next hub.

To verify the configuration:

1. Verify the health check status on Hub-1. Note that port4 and port5 do not meet SLA:

```
PoP1-Hub1 (root) # diagnose sys sdwan health-check
Health Check(external_peers):
Seq(1 port4): state(dead), packet-loss(100.000%) sla_map=0x0
Seq(2 port5): state(dead), packet-loss(100.000%) sla_map=0x0
```

2. Verify the SD-WAN neighbor status. The `minimum-sla-meet-members` threshold of 1 is not met:

```
PoP1-Hub1 (root) # diagnose sys sdwan neighbor
Neighbor(EDGE): member(1 2)role(standalone)
        Health-check(external_peers:1)  sla-fail dead
```

3. Verify the BGP learned routes. Hub-1 does not advertise any external routes to the Spoke-1 BGP neighbor:

```
PoP1-Hub1 (root) # get router info bgp neighbors 172.31.0.65 advertised-routes
% No prefix for neighbor 172.31.0.65
```

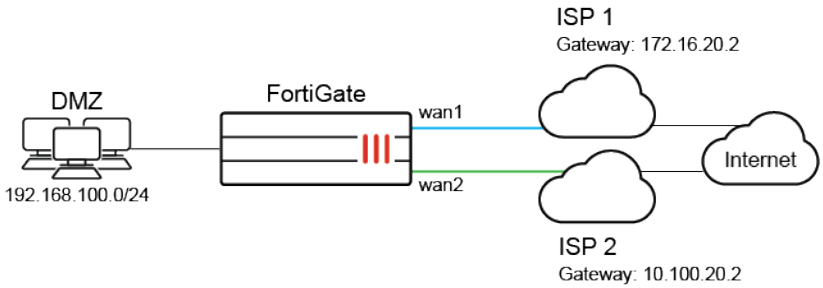
SD-WAN traffic shaping and QoS

Use a traffic shaper in a firewall shaping policy to control traffic flow. You can use it to control maximum and guaranteed bandwidth, or put certain traffic to one of the three different traffic priorities: high, medium, or low.

An advanced shaping policy can classify traffic into 30 groups. Use a shaping profile to define the percentage of the interface bandwidth that is allocated to each group. Each group of traffic is shaped to the assigned speed limit based on the outgoing bandwidth limit configured on the interface.

For more information, see Traffic shaping on page 2108.

Sample topology



Sample configuration

This example shows a typical customer usage where the customer's SD-WAN uses the default zone, and has two member: wan1 and wan2, each set to 10Mb/s.

An overview of the procedures to configure SD-WAN traffic shaping and QoS with SD-WAN includes:

1. Give HTTP/HTTPS traffic high priority and give FTP low priority so that if there are conflicts, FortiGate will forward HTTP/HTTPS traffic first.
2. Even though FTP has low priority, configure FortiGate to give it a 1Mb/s guaranteed bandwidth on each SD-WAN member so that if there is no FTP traffic, other traffic can use all the bandwidth. If there is heavy FTP traffic, it can still be guaranteed a 1Mb/s bandwidth.
3. Traffic going to specific destinations such as a VOIP server uses wan1 to forward, and SD-WAN forwards with an Expedited Forwarding (EF) DSCP tag 101110.

To configure SD-WAN traffic shaping and QoS with SD-WAN in the GUI:

1. On the FortiGate, add wan1 and wan2 as SD-WAN members, then add a policy and static route.
   See SD-WAN quick start on page 1041.
2. Add a firewall policy with *Application Control* enabled. See Security on page 1044.
3. Go to *Policy & Objects* > *Traffic Shaping*, select the *Traffic Shaper* tab, and edit *low-priority*.
   a. Enable *Guaranteed Bandwidth* and set it to *1000* kbps.
4. Go to *Policy & Objects* > *Traffic Shaping*, select the *Traffic Shaping Policy* tab, and click *Create New*.
   a. Name the traffic shaping policy, for example, *HTTP-HTTPS*.
   b. Set the following:

Source	*all*
Destination	*all*
Service	*HTTP* and *HTTPS*
Outgoing interface	*virtual-wan-link*
Shared Shaper	Enable and set to *high-priority*
Reverse Shaper	Enable and set to *high-priority*

   c. Click *OK*.
5. Go to *Policy & Objects* > *Traffic Shaping*, select the *Traffic Shaping Policy* tab, and click *Create New*.
   a. Name the traffic shaping policy, for example, *FTP*.
   b. Set the following:

Source	*all*
Destination	*all*
Service	*FTP*, *FTP_GET*, and *FTP_PUT*
Outgoing interface	*virtual-wan-link*
Shared Shaper	Enable and set to *low-priority*
Reverse Shaper	Enable and set to *low-priority*

   c. Click *OK*
6. Go to *Network > SD-WAN*, select the *SD-WAN Rules* tab, and click *Create New*.
   a. Enter a name for the rule, such as *Internet*.
   b. In the *Destination* section, click *Address* and select the VoIP server that you created in the firewall address.
   c. Under *Outgoing Interfaces* select *Manual*.
   d. For *Interface preference* select *wan1*.
   e. Click *OK*.
7. Use CLI commands to modify DSCP settings. See the DSCP CLI commands below.

To configure the firewall policy using the CLI:

```
config firewall policy
    edit 1
        set name "1"
        set srcintf "dmz"
        set dstintf "virtual-wan-link"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "certificate-inspection"
        set application-list "default"
        set nat enable
    next
end
```

To configure the firewall traffic shaper priority using the CLI:

```
config firewall shaper traffic-shaper
    edit "high-priority"
        set maximum-bandwidth 1048576
        set per-policy enable
    next
    edit "low-priority"
        set guaranteed-bandwidth 1000
        set maximum-bandwidth 1048576
        set priority low
        set per-policy enable
    next
end
```

To configure the firewall traffic shaping policy using the CLI:

```
config firewall shaping-policy
    edit 1
        set name "http-https"
```

```
        set service "HTTP" "HTTPS"
        set dstintf "virtual-wan-link"
        set traffic-shaper "high-priority"
        set traffic-shaper-reverse "high-priority"
        set srcaddr "all"
        set dstaddr "all"
    next
    edit 2
        set name "FTP"
        set service "FTP" "FTP_GET" "FTP_PUT"
        set dstintf "virtual-wan-link"
        set traffic-shaper "low-priority"
        set traffic-shaper-reverse "low-priority"
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

To configure SD-WAN traffic shaping and QoS with SD-WAN in the CLI:

```
config system sdwan
    set status enable
    config members
        edit 1
            set interface "wan1"
            set gateway 172.16.20.2
        next
        edit 2
            set interface "wan2"
            set gateway 10.100.20.2
        next
    end
    config service
        edit 1
            set name "SIP"
            set priority-members 1
            set dst "voip-server"
            set dscp-forward enable
            set dscp-forward-tag 101110
        next
    end
end
```

If no SD-WAN zone is specified, members are added to the default *virtual-wan-link* zone.

To use the diagnose command to check if specific traffic is attached to the correct traffic shaper:

```
# diagnose firewall iprope list 100015

policy index=1 uuid_idx=0 action=accept
```

```
flag (0):
shapers: orig=high-priority(2/0/134217728) reply=high-priority(2/0/134217728)
cos_fwd=0  cos_rev=0
group=00100015 av=00000000 au=00000000 split=00000000
host=0 chk_client_info=0x0 app_list=0 ips_view=0
misc=0 dd_type=0 dd_mode=0
zone(1): 0 -> zone(2): 36 38
source(1): 0.0.0.0-255.255.255.255, uuid_idx=6,
dest(1): 0.0.0.0-255.255.255.255, uuid_idx=6,
service(2):
        [6:0x0:0/(1,65535)->(80,80)] helper:auto
        [6:0x0:0/(1,65535)->(443,443)] helper:auto

policy index=2 uuid_idx=0 action=accept
flag (0):
shapers: orig=low-priority(4/128000/134217728) reply=low-priority(4/128000/134217728)
cos_fwd=0  cos_rev=0
group=00100015 av=00000000 au=00000000 split=00000000
host=0 chk_client_info=0x0 app_list=0 ips_view=0
misc=0 dd_type=0 dd_mode=0
zone(1): 0 -> zone(2): 36 38
source(1): 0.0.0.0-255.255.255.255, uuid_idx=6,
dest(1): 0.0.0.0-255.255.255.255, uuid_idx=6,
service(3):
        [6:0x0:0/(1,65535)->(21,21)] helper:auto
        [6:0x0:0/(1,65535)->(21,21)] helper:auto
        [6:0x0:0/(1,65535)->(21,21)] helper:auto
```

To use the diagnose command to check if the correct traffic shaper is applied to the session:

```
# diagnose sys session list
session info: proto=6 proto_state=01 duration=11 expire=3599 timeout=3600 flags=00000000
sockflag=00000000 sockport=0 av_idx=0 use=5
origin-shaper=low-priority prio=4 guarantee 128000Bps max 1280000Bps traffic 1050Bps drops 0B
reply-shaper=
per_ip_shaper=
class_id=0 shaping_policy_id=2 ha_id=0 policy_dir=0 tunnel=/ helper=ftp vlan_cos=0/255
state=may_dirty npu npd os mif route_preserve
statistic(bytes/packets/allow_err): org=868/15/1 reply=752/10/1 tuples=2
tx speed(Bps/kbps): 76/0 rx speed(Bps/kbps): 66/0
orgin->sink: org pre->post, reply pre->post dev=39->38/38->39 gwy=172.16.200.55/0.0.0.0
hook=post dir=org act=snat 10.1.100.11:58241->172.16.200.55:21(172.16.200.1:58241)
hook=pre dir=reply act=dnat 172.16.200.55:21->172.16.200.1:58241(10.1.100.11:58241)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=1 auth_info=0 chk_client_info=0 vd=4
serial=0003255f tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id = 00000000
dd_type=0 dd_mode=0
npu_state=0x100000
npu info: flag=0x00/0x00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0, vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
```

```
no_ofld_reason:  offload-denied helper
total session 1
```

To use the diagnose command to check the status of a shared traffic shaper:

```
# diagnose firewall shaper traffic-shaper list

name high-priority
maximum-bandwidth 131072 KB/sec
guaranteed-bandwidth 0 KB/sec
current-bandwidth 0 B/sec
priority 2
tos ff
packets dropped 0
bytes dropped 0

name low-priority
maximum-bandwidth 131072 KB/sec
guaranteed-bandwidth 125 KB/sec
current-bandwidth 0 B/sec
priority 4
tos ff
packets dropped 0
bytes dropped 0

name high-priority
maximum-bandwidth 131072 KB/sec
guaranteed-bandwidth 0 KB/sec
current-bandwidth 0 B/sec
priority 2
policy 1
tos ff
packets dropped 0
bytes dropped 0

name low-priority
maximum-bandwidth 131072 KB/sec
guaranteed-bandwidth 125 KB/sec
current-bandwidth 0 B/sec
priority 4
policy 2
tos ff
packets dropped 0
bytes dropped 0
```

Troubleshooting SD-WAN

The following topics provide instructions on SD-WAN troubleshooting:

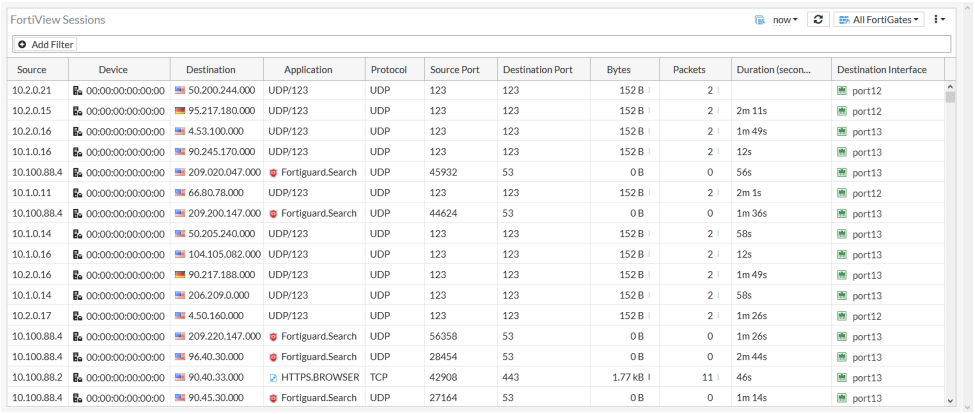
- Tracking SD-WAN sessions on page 1555
- Understanding SD-WAN related logs on page 1555
- SD-WAN related diagnose commands on page 1558
- Using SNMP to monitor health check on page 1563

Tracking SD-WAN sessions

You can check the destination interface in *Dashboard > FortiView Sessions* in order to see which port the traffic is being forwarded to.

The example below demonstrates a source-based load-balance between two SD-WAN members:

- If the source IP address is an *even* number, it will go to *port13*.
- If the source IP address is an *odd* number, it will go to *port12*.



Source	Device	Destination	Application	Protocol	Source Port	Destination Port	Bytes	Packets	Duration (secon...	Destination Interface
10.2.0.21	00:00:00:00:00:00	50.200.244.000	UDP/123	UDP	123	123	152 B	2		port12
10.2.0.15	00:00:00:00:00:00	95.217.180.000	UDP/123	UDP	123	123	152 B	2	2m 11s	port12
10.2.0.16	00:00:00:00:00:00	4.53.100.000	UDP/123	UDP	123	123	152 B	2	1m 49s	port13
10.1.0.16	00:00:00:00:00:00	90.245.170.000	UDP/123	UDP	123	123	152 B	2	12s	port13
10.100.88.4	00:00:00:00:00:00	209.020.047.000	Fortiguard.Search	UDP	45932	53	0 B	0	56s	port13
10.1.0.11	00:00:00:00:00:00	66.80.78.000	UDP/123	UDP	123	123	152 B	2	2m 1s	port12
10.100.88.4	00:00:00:00:00:00	209.200.147.000	Fortiguard.Search	UDP	44624	53	0 B	0	1m 36s	port13
10.1.0.14	00:00:00:00:00:00	50.205.240.000	UDP/123	UDP	123	123	152 B	2	58s	port13
10.1.0.16	00:00:00:00:00:00	104.105.082.000	UDP/123	UDP	123	123	152 B	2	12s	port13
10.2.0.16	00:00:00:00:00:00	90.217.188.000	UDP/123	UDP	123	123	152 B	2	1m 49s	port13
10.1.0.14	00:00:00:00:00:00	206.209.0.000	UDP/123	UDP	123	123	152 B	2	58s	port13
10.2.0.17	00:00:00:00:00:00	4.50.160.000	UDP/123	UDP	123	123	152 B	2	1m 26s	port12
10.100.88.4	00:00:00:00:00:00	209.220.147.000	Fortiguard.Search	UDP	56358	53	0 B	0	1m 26s	port13
10.100.88.4	00:00:00:00:00:00	96.40.30.000	Fortiguard.Search	UDP	28454	53	0 B	0	2m 44s	port13
10.100.88.2	00:00:00:00:00:00	90.40.33.000	HTTPS.BROWSER	TCP	42908	443	1.77 kB	11	46s	port13
10.100.88.4	00:00:00:00:00:00	90.45.30.000	Fortiguard.Search	UDP	27164	53	0 B	0	1m 14s	port13

Understanding SD-WAN related logs

This topic lists the SD-WAN related logs and explains when the logs will be triggered.

Health-check detects a failure:

- When health-check detects a failure, it will record a log:

```
1: date=2021-04-20 time=17:06:31 eventtime=1618963591590008160 tz="-0700" logid="0100022921"
type="event" subtype="system" level="critical" vd="root" logdesc="Routing information changed"
name="test" interface="R150" status="down" msg="Static route on interface R150 may be removed
by health-check test. Route:  (10.100.1.2->10.100.2.22 ping-down)"
```

- When health-check detects a recovery, it will record a log:

```
2: date=2021-04-20 time=17:11:46 eventtime=1618963906950174240 tz="-0700" logid="0100022921"
type="event" subtype="system" level="critical" vd="root" logdesc="Routing information changed"
name="test" interface="R150" status="up" msg="Static route on interface R150 may be added by
health-check test. Route:  (10.100.1.2->10.100.2.22 ping-up)"
```

Health-check has an SLA target and detects SLA qualification changes:

- When health-check has an SLA target and detects SLA changes, and changes to fail:

```
1: date=2021-04-20 time=21:32:33 eventtime=1618979553388763760 tz="-0700" logid="0113022923"
type="event" subtype="sdwan" level="notice" vd="root" logdesc="SDWAN status" eventtype="Health
Check" healthcheck="test" slatargetid=1 oldvalue="2" newvalue="1" msg="Number of pass member
changed."
```

```
2: date=2021-04-20 time=21:32:33 eventtime=1618979553388751880 tz="-0700" logid="0113022923"
type="event" subtype="sdwan" level="notice" vd="root" logdesc="SDWAN status" eventtype="Health
Check" healthcheck="test" slatargetid=1 member="1" msg="Member status changed. Member out-of-
sla."
```

- When health-check has an SLA target and detects SLA changes, and changes to pass:

```
1: date=2021-04-20 time=21:38:49 eventtime=1618979929908765200 tz="-0700" logid="0113022923"
type="event" subtype="sdwan" level="notice" vd="root" logdesc="SDWAN status" eventtype="Health
Check" healthcheck="test" slatargetid=1 oldvalue="1" newvalue="2" msg="Number of pass member
changed."
```

```
2: date=2021-04-20 time=21:38:49 eventtime=1618979929908754060 tz="-0700" logid="0113022923"
type="event" subtype="sdwan" level="information" vd="root" logdesc="SDWAN status"
eventtype="Health Check" healthcheck="test" slatargetid=1 member="1" msg="Member status
changed. Member in sla."
```

SD-WAN calculates a link's session/bandwidth over/under its ratio and stops/resumes traffic:

- When SD-WAN calculates a link's session/bandwidth over its configured ratio and stops forwarding traffic:

```
1: date=2021-04-20 time=21:55:14 eventtime=1618980914728863220 tz="-0700" logid="0113022924"
type="event" subtype="sdwan" level="notice" vd="root" logdesc="SDWAN volume status"
eventtype="Volume" interface="R160" member="2" msg="Member enters into conservative status
with limited ablity to receive new sessions for too much traffic."
```

- When SD-WAN calculates a link's session/bandwidth according to its ratio and resumes forwarding traffic:

```
2: date=2021-04-20 time=22:12:52 eventtime=1618981972698753360 tz="-0700" logid="0113022924"
type="event" subtype="sdwan" level="notice" vd="root" logdesc="SDWAN volume status"
eventtype="Volume" interface="R160" member="2" msg="Member resume normal status to receive new
sessions for internal adjustment"
```

The SLA mode service rule's SLA qualified member changes:

- When the SLA mode service rule's SLA qualified member changes. In this example R150 fails the SLA check, but is still alive:

```
1: date=2021-04-20 time=22:40:46 eventtime=1618983646428803040 tz="-0700" logid="0113022923"
type="event" subtype="sdwan" level="notice" vd="root" logdesc="SDWAN status"
eventtype="Service" serviceid=1 service="test" seq="2,1" msg="Service prioritized by SLA will
be redirected in sequence order."
```

- When the SLA mode service rule's SLA qualified member changes. In this example R150 changes from fail to pass:

```
2: date=2021-04-20 time=22:41:51 eventtime=1618983711678827920 tz="-0700" logid="0113022923"
type="event" subtype="sdwan" level="notice" vd="root" logdesc="SDWAN status"
eventtype="Service" serviceid=1 service="test" seq="1,2" msg="Service prioritized by SLA will
be redirected in sequence order."
```

The priority mode service rule member's link status changes:

- When priority mode service rule member's link status changes. In this example R150 changes to better than R160, and both are still alive:

```
1: date=2021-04-20 time=22:56:55 eventtime=1618984615708804760 tz="-0700" logid="0113022923"
type="event" subtype="sdwan" level="notice" vd="root" logdesc="SDWAN status"
eventtype="Service" serviceid=1 service="test" metric="packet-loss" seq="2,1" msg="Service
prioritized by performance metric will be redirected in sequence order."
```

- When priority mode service rule member's link status changes. In this example R160 changes to better than R150, and both are still alive:

```
2: date=2021-04-20 time=22:56:58 eventtime=1618984618278852140 tz="-0700" logid="0113022923"
type="event" subtype="sdwan" level="notice" vd="root" logdesc="SDWAN status"
eventtype="Service" serviceid=1 service="test" metric="packet-loss" seq="1,2" msg="Service
prioritized by performance metric will be redirected in sequence order."
```

SD-WAN member is used in service and it fails the health-check:

- When SD-WAN member fails the health-check, it will stop forwarding traffic:

```
1: date=2021-04-20 time=23:04:32 eventtime=1618985072898756700 tz="-0700" logid="0113022923"
type="event" subtype="sdwan" level="notice" vd="root" logdesc="SDWAN status"
eventtype="Service" interface="R150" member="1" serviceid=1 service="test" gateway=10.100.1.1
msg="Member link is unreachable or miss threshold. Stop forwarding traffic. "
```

- When SD-WAN member passes the health-check again, it will resume forwarding logs:

```
2: date=2021-04-20 time=23:06:08 eventtime=1618985168018789600 tz="-0700" logid="0113022923"
type="event" subtype="sdwan" level="notice" vd="root" logdesc="SDWAN status"
eventtype="Service" interface="R150" member="1" serviceid=1 service="test" gateway=10.100.1.1
msg="Member link is available. Start forwarding traffic. "
```

Load-balance mode service rule's SLA qualified member changes:

- When load-balance mode service rule's SLA qualified member changes. In this example R150 changes to not meet SLA:

```
1: date=2021-04-20 time=23:10:24 eventtime=1618985425048820800 tz="-0700" logid="0113022923"
type="event" subtype="sdwan" level="notice" vd="root" logdesc="SDWAN status"
eventtype="Service" serviceid=1 service="test" member="2(R160)" msg="Service will be load
balanced among members with available routing."
```

- When load-balance mode service rule's SLA qualified member changes. In this example R150 changes to meet SLA:

```
2: date=2021-04-20 time=23:11:34 eventtime=1618985494478807100 tz="-0700" logid="0113022923"
type="event" subtype="sdwan" level="notice" vd="root" logdesc="SDWAN status"
eventtype="Service" serviceid=1 service="test" member="2(R160),1(R150)" msg="Service will be
load balanced among members with available routing."
```

SLA link status logs, generated with interval sla-fail-log-period or sla-pass-log-period:

- When SLA fails, SLA link status logs will be generated with interval sla-fail-log-period:

```
1: date=2021-04-20 time=23:18:10 eventtime=1618985890469018260 tz="-0700" logid="0113022925"
type="event" subtype="sdwan" level="notice" vd="root" logdesc="SDWAN SLA information"
eventtype="SLA" healthcheck="test" slatargetid=1 interface="R150" status="up" latency="0.061"
jitter="0.004" packetloss="2.000%" inbandwidthavailable="0kbps"
outbandwidthavailable="200.00Mbps" bibandwidthavailable="200.00Mbps" inbandwidthused="1kbps"
outbandwidthused="1kbps" bibandwidthused="2kbps" slamap="0x0" metric="packetloss" msg="Health
Check SLA status. SLA failed due to being over the performance metric threshold."
```

- When SLA passes, SLA link status logs will be generated with interval sla-pass-log-period:

```
2: date=2021-04-20 time=23:18:12 eventtime=1618985892509027220 tz="-0700" logid="0113022925"
type="event" subtype="sdwan" level="information" vd="root" logdesc="SDWAN SLA information"
eventtype="SLA" healthcheck="test" slatargetid=1 interface="R150" status="up" latency="0.060"
jitter="0.003" packetloss="0.000%" inbandwidthavailable="0kbps"
outbandwidthavailable="200.00Mbps" bibandwidthavailable="200.00Mbps" inbandwidthused="1kbps"
outbandwidthused="1kbps" bibandwidthused="2kbps" slamap="0x1" msg="Health Check SLA status."
```

SD-WAN related diagnose commands

This topic lists the SD-WAN related diagnose commands and related output.

To check SD-WAN health-check status:

```
FGT # diagnose sys sdwan health-check
Health Check(server):
Seq(1 R150): state(alive), packet-loss(0.000%) latency(0.110), jitter(0.024) sla_map=0x0
Seq(2 R160): state(alive), packet-loss(0.000%) latency(0.068), jitter(0.009) sla_map=0x0
```

```
FGT # diagnose sys sdwan health-check
Health Check(ping):
Seq(1 R150): state(alive), packet-loss(0.000%) latency(0.100), jitter(0.017) sla_map=0x0
Seq(2 R160): state(dead), packet-loss(100.000%) sla_map=0x0
```

```
FGT # diagnose sys sdwan health-check google
Health Check(google):
Seq(1 R150): state(alive), packet-loss(0.000%) latency(0.081), jitter(0.019) sla_map=0x0
Seq(2 R160): state(alive), packet-loss(0.000%) latency(0.060), jitter(0.004) sla_map=0x0
```

To check SD-WAN member status:

- When SD-WAN load-balance mode is *source-ip-based*/*source-dest-ip-based*.

```
FGT # diagnose sys sdwan member
Member(1): interface: R150, gateway: 10.100.1.1 2000:10:100:1::1, priority: 0 1024, weight: 0
Member(2): interface: R160, gateway: 10.100.1.5 2000:10:100:1::5, priority: 0 1024, weight: 0
```

- When SD-WAN load-balance mode is *weight-based*.

```
FGT # diagnose sys sdwan member
Member(1): interface: R150, gateway: 10.100.1.1 2000:10:100:1::1, priority: 0 1024, weight: 33
  Session count: 15
Member(2): interface: R160, gateway: 10.100.1.5 2000:10:100:1::5, priority: 0 1024, weight: 66
  Session count: 1
```

- When SD-WAN load-balance mode is *measured-volume-based*.
  - Both members are under volume and still have room:

```
FGT # diagnose sys sdwan member
Member(1): interface: R150, gateway: 10.100.1.1 2000:10:100:1::1, priority: 0 1024,
weight: 33
  Config volume ratio: 33, last reading: 218067B, volume room 33MB
Member(2): interface: R160, gateway: 10.100.1.5 2000:10:100:1::5, priority: 0 1024,
weight: 66
  Config volume ratio: 66, last reading: 202317B, volume room 66MB
```

  - Some members are overloaded and some still have room:

```
FGT # diagnose sys sdwan member
Member(1): interface: R150, gateway: 10.100.1.1 2000:10:100:1::1, priority: 0 1024,
weight: 0
  Config volume ratio: 33, last reading: 1287767633B, overload volume 517MB
Member(2): interface: R160, gateway: 10.100.1.5 2000:10:100:1::5, priority: 0 1024,
weight: 63
  Config volume ratio: 66, last reading: 1686997898B, volume room 63MB
```

- When SD-WAN load balance mode is *usage-based*/*spillover*.
  - When no spillover occurs:

```
FGT # diagnose sys sdwan member
Member(1): interface: R150, gateway: 10.100.1.1 2000:10:100:1::1, priority: 0 1024,
weight: 255
  Egress-spillover-threshold: 400kbit/s, ingress-spillover-threshold: 300kbit/s
  Egress-overbps=0, ingress-overbps=0
Member(2): interface: R160, gateway: 10.100.1.5 2000:10:100:1::5, priority: 0 1024,
weight: 254
  Egress-spillover-threshold: 0kbit/s, ingress-spillover-threshold: 0kbit/s
  Egress-overbps=0, ingress-overbps=0
```

  - When member has reached limit and spillover occurs:

```
FGT # diagnose sys sdwan member
Member(1): interface: R150, gateway: 10.100.1.1 2000:10:100:1::1, priority: 0 1024,
weight: 255
  Egress-spillover-threshold: 400kbit/s, ingress-spillover-threshold: 300kbit/s
  Egress-overbps=1, ingress-overbps=0
Member(2): interface: R160, gateway: 10.100.1.5 2000:10:100:1::5, priority: 0 1024,
weight: 254
  Egress-spillover-threshold: 0kbit/s, ingress-spillover-threshold: 0kbit/s
  Egress-overbps=0, ingress-overbps=0
```

- You can also use the `diagnose netlink dstmac list` command to check if you are over the limit.

```
FGT # diagnose netlink dstmac list R150
dev=R150 mac=00:00:00:00:00:00 vwl rx_tcp_mss=0 tx_tcp_mss=0 egress_overspill_
threshold=50000 egress_bytes=100982 egress_over_bps=1 ingress_overspill_threshold=37500
ingress_bytes=40 ingress_over_bps=0 sampler_rate=0 vwl_zone_id=1 intf_qua=0
```

To check SD-WAN service rules status:

- *Manual mode* service rules.

```
FGT # diagnose sys sdwan service4
Service(1): Address Mode(IPV4) flags=0x200
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
  Members(2):
    1: Seq_num(1 R150), alive, selected
    2: Seq_num(2 R160), alive, selected
  Dst address(1):
        10.100.21.0-10.100.21.255
```

- *Auto mode* service rules.

```
FGT # diagnose sys sdwan service4
Service(1): Address Mode(IPV4) flags=0x200
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(auto), link-cost-factor(latency), link-
cost-threshold(10), health-check(ping)
  Members(2):
    1: Seq_num(2 R160), alive, latency: 0.066, selected
    2: Seq_num(1 R150), alive, latency: 0.093
  Dst address(1):
        10.100.21.0-10.100.21.255
```

- *Priority mode* service rules.

```
FGT # diagnose sys sdwan service4
Service(1): Address Mode(IPV4) flags=0x200
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(priority), link-cost-factor(latency),
link-cost-threshold(10), health-check(ping)
  Members(2):
    1: Seq_num(2 R160), alive, latency: 0.059, selected
    2: Seq_num(1 R150), alive, latency: 0.077, selected
```

```
 Dst address(1):
       10.100.21.0-10.100.21.255
```

- *Load-balance* mode service rules.

```
FGT # diagnose sys sdwan service4
Service(1): Address Mode(IPV4) flags=0x200
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(load-balance hash-mode=round-robin)
  Members(2):
    1: Seq_num(1 R150), alive, sla(0x1), gid(2), num of pass(1), selected
    2: Seq_num(2 R160), alive, sla(0x1), gid(2), num of pass(1), selected
  Dst address(1):
        10.100.21.0-10.100.21.255
```

- *SLA mode* service rules.

```
FGT # diagnose sys sdwan service4
Service(1): Address Mode(IPV4) flags=0x200
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(sla), sla-compare-order
  Members(2):
    1: Seq_num(1 R150), alive, sla(0x1), gid(0), cfg_order(0), cost(0), selected
    2: Seq_num(2 R160), alive, sla(0x1), gid(0), cfg_order(1), cost(0), selected
  Dst address(1):
        10.100.21.0-10.100.21.255
```

To check interface logs from the past 15 minutes:

```
FGT (root) # diagnose sys sdwan intf-sla-log R150
Timestamp: Wed Apr 21 16:58:27 2021, used inbandwidth: 655bps, used outbandwidth: 81655306bps,
used bibandwidth: 81655961bps, tx bys: 3413479982bytes, rx bytes: 207769bytes.
Timestamp: Wed Apr 21 16:58:37 2021, used inbandwidth: 649bps, used outbandwidth: 81655540bps,
used bibandwidth: 81656189bps, tx bys: 3515590414bytes, rx bytes: 208529bytes.
Timestamp: Wed Apr 21 16:58:47 2021, used inbandwidth: 655bps, used outbandwidth: 81655546bps,
used bibandwidth: 81656201bps, tx bys: 3617700886bytes, rx bytes: 209329bytes.
Timestamp: Wed Apr 21 16:58:57 2021, used inbandwidth: 620bps, used outbandwidth: 81671580bps,
used bibandwidth: 81672200bps, tx bys: 3719811318bytes, rx bytes: 210089bytes.
Timestamp: Wed Apr 21 16:59:07 2021, used inbandwidth: 620bps, used outbandwidth: 81671580bps,
used bibandwidth: 81672200bps, tx bys: 3821921790bytes, rx bytes: 210889bytes.
Timestamp: Wed Apr 21 16:59:17 2021, used inbandwidth: 665bps, used outbandwidth: 81688152bps,
used bibandwidth: 81688817bps, tx bys: 3924030936bytes, rx bytes: 211926bytes.
Timestamp: Wed Apr 21 16:59:27 2021, used inbandwidth: 671bps, used outbandwidth: 81688159bps,
used bibandwidth: 81688830bps, tx bys: 4026141408bytes, rx bytes: 212726bytes.
```

To check SLA logs in the past 10 minutes:

```
FGT (root) # diagnose sys sdwan sla-log ping 1
Timestamp: Wed Apr 21 17:10:11 2021, vdom root, health-check ping, interface: R150, status: up,
latency: 0.079, jitter: 0.023, packet loss: 0.000%.
Timestamp: Wed Apr 21 17:10:12 2021, vdom root, health-check ping, interface: R150, status: up,
latency: 0.079, jitter: 0.023, packet loss: 0.000%.
Timestamp: Wed Apr 21 17:10:12 2021, vdom root, health-check ping, interface: R150, status: up,
```

```
latency: 0.081, jitter: 0.024, packet loss: 0.000%.
Timestamp: Wed Apr 21 17:10:13 2021, vdom root, health-check ping, interface: R150, status: up,
latency: 0.081, jitter: 0.025, packet loss: 0.000%.
Timestamp: Wed Apr 21 17:10:13 2021, vdom root, health-check ping, interface: R150, status: up,
latency: 0.082, jitter: 0.026, packet loss: 0.000%.
Timestamp: Wed Apr 21 17:10:14 2021, vdom root, health-check ping, interface: R150, status: up,
latency: 0.083, jitter: 0.026, packet loss: 0.000%.
Timestamp: Wed Apr 21 17:10:14 2021, vdom root, health-check ping, interface: R150, status: up,
latency: 0.084, jitter: 0.026, packet loss: 0.000%.
```

To check Application Control used in SD-WAN and the matching IP addresses:

```
FGT # diagnose sys sdwan internet-service-app-ctrl-list
Gmail(15817 4294836957): 64.233.191.19 6 443 Thu Apr 22 10:10:34 2021
Gmail(15817 4294836957): 142.250.128.83 6 443 Thu Apr 22 10:06:47 2021
Facebook(15832 4294836806): 69.171.250.35 6 443 Thu Apr 22 10:12:00 2021
Amazon(16492 4294836342): 3.226.60.231 6 443 Thu Apr 22 10:10:57 2021
Amazon(16492 4294836342): 52.46.135.211 6 443 Thu Apr 22 10:10:58 2021
Amazon(16492 4294836342): 52.46.141.85 6 443 Thu Apr 22 10:10:58 2021
Amazon(16492 4294836342): 52.46.155.13 6 443 Thu Apr 22 10:10:58 2021
Amazon(16492 4294836342): 54.82.242.32 6 443 Thu Apr 22 10:10:59 2021
YouTube(31077 4294838537): 74.125.202.138 6 443 Thu Apr 22 10:06:51 2021
YouTube(31077 4294838537): 108.177.121.119 6 443 Thu Apr 22 10:08:24 2021
YouTube(31077 4294838537): 142.250.136.119 6 443 Thu Apr 22 10:02:02 2021
YouTube(31077 4294838537): 142.250.136.132 6 443 Thu Apr 22 10:08:16 2021
YouTube(31077 4294838537): 142.250.148.100 6 443 Thu Apr 22 10:07:28 2021
YouTube(31077 4294838537): 142.250.148.132 6 443 Thu Apr 22 10:10:32 2021
YouTube(31077 4294838537): 172.253.119.91 6 443 Thu Apr 22 10:02:01 2021
YouTube(31077 4294838537): 184.150.64.211 6 443 Thu Apr 22 10:04:36 2021
YouTube(31077 4294838537): 184.150.168.175 6 443 Thu Apr 22 10:02:26 2021
YouTube(31077 4294838537): 184.150.168.211 6 443 Thu Apr 22 10:02:26 2021
YouTube(31077 4294838537): 184.150.186.141 6 443 Thu Apr 22 10:02:26 2021
YouTube(31077 4294838537): 209.85.145.190 6 443 Thu Apr 22 10:10:36 2021
YouTube(31077 4294838537): 209.85.200.132 6 443 Thu Apr 22 10:02:03 2021
```

To check the dynamic tunnel status:

```
# diagnose sys link-monitor interface <name> <name>_0
```

For example:

```
# diagnose sys link-monitor interface vd2-2
Interface(vd2-2): state(up, since Tue Jun 15 12:31:28 2021), bandwidth(up:1299bps, down:0bps),
session count(IPv4:2, IPv6:0), tx(2409919 bytes), rx(5292290 bytes), latency(0.03), jitter(0.00),
packet-loss(0.00).

# diagnose sys link-monitor interface vd2-2 vd2-2_0
Interface(vd2-2_0): state(up, since Tue Jun 15 15:21:52 2021), bandwidth(up:640bps, down:0bps),
session count(IPv4:0, IPv6:0), tx(102242 bytes), rx(16388 bytes), latency(0.03), jitter(0.00),
packet-loss(0.00).
```

To check BGP learned routes and determine if they are used in SD-WAN service:

```
FGT # get router info bgp network 10.100.11.0/24
VRF 0 BGP routing table entry for 10.100.11.0/24
Paths: (2 available, best #2, table Default-IP-Routing-Table)
  Advertised to non peer-group peers:
   10.100.1.1
  Original VRF 0
  20 10
    10.100.1.1 from 10.100.1.1 (5.5.5.5)
      Origin incomplete metric 0, route tag 15, localpref 100, valid, external, best
      Community: 30:5
      Advertised Path ID: 2
       Last update: Thu Apr 22 10:27:27 2021

  Original VRF 0
  20 10
    10.100.1.5 from 10.100.1.5 (6.6.6.6)
      Origin incomplete metric 0, route tag 15, localpref 100, valid, external, best
      Community: 30:5
      Advertised Path ID: 1
       Last update: Thu Apr 22 10:25:50 2021
```

```
FGT # diagnose sys sdwan route-tag-list
Route-tag: 15, address: v4(1), v6(0)Last write/now: 6543391 6566007
        service(1), last read route-tag 15 at 6543420
Prefix(24): Address list(1):
        10.100.11.0-10.100.11.255 oif: 50 48
```

```
FGT # diagnose firewall proute list
list route policy info(vf=root):
id=2133196801(0x7f260001) vwl_service=1(DataCenter) vwl_mbr_seq=1 2 dscp_tag=0xff 0xff flags=0x40
order-addr tos=0x00 tos_mask=0x00 protocol=0 sport=0-65535 iif=0 dport=1-65535 oif=48(R150) oif=50
(R160)
destination(1): 10.100.11.0-10.100.11.255
source wildcard(1): 0.0.0.0/0.0.0.0
hit_count=0 last_used=2021-04-22 10:25:10
```

Using SNMP to monitor health check

You can monitor SD-WAN health check related statistics using SNMP. The MIB file can be downloaded by going to *System > SNMP* and clicking *Download FortiGate MIB File*.

The following OIDs can be monitored:

Name	OID	Description
fgVWLHealthCheckLinkNumber	.1.3.6.1.4.1.12356.101.4.9.1	The number of health check links in fgVWLHealthCheckLinkTable
fgVWLHealthCheckLinkTable	.1.3.6.1.4.1.12356.101.4.9.2	SD-WAN health check statistics table. This table has a dependent expansion relationship with fgVdTable.Only health checks with a configured member link are present in this table.
fgVWLHealthCheckLinkTableEntry	.1.3.6.1.4.1.12356.101.4.9.2.1	SD-WAN health check statistics on a virtual domain.
fgVWLHealthCheckLinkID	.1.3.6.1.4.1.12356.101.4.9.2.1.1	SD-WAN health check link ID. Only health checks with configured member link are present in this table. Virtual-wan-link health check link IDs are only unique within a virtual domain.
fgVWLHealthCheckLinkName	.1.3.6.1.4.1.12356.101.4.9.2.1.2	Health check name.
fgVWLHealthCheckLinkSeq	.1.3.6.1.4.1.12356.101.4.9.2.1.3	SD-WAN member link sequence.
fgVWLHealthCheckLinkState	.1.3.6.1.4.1.12356.101.4.9.2.1.4	Health check state on a specific member link.
fgVWLHealthCheckLinkLatency	.1.3.6.1.4.1.12356.101.4.9.2.1.5	The average latency of a health check on a specific member link within last 30 probes, in float number.
fgVWLHealthCheckLinkJitter	.1.3.6.1.4.1.12356.101.4.9.2.1.6	The average jitter of a health check on a specific member link within last 30 probes, in float number.
fgVWLHealthCheckLinkPacketSend	.1.3.6.1.4.1.12356.101.4.9.2.1.7	The total number of packets sent by a health check on a specific member link.
fgVWLHealthCheckLinkPacketRecv	.1.3.6.1.4.1.12356.101.4.9.2.1.8	The total number of packets received by a health check on a specific member link.

Name	OID	Description
fgVWLHealthCheckLinkPacketLoss	.1.3.6.1.4.1.12356.101.4.9.2.1.9	The packet loss percentage of a health check on a specific member link within last 30 probes, in float number.
fgVWLHealthCheckLinkVdom	.1.3.6.1.4.1.12356.101.4.9.2.1.10	The VDOM that the link monitor entry exists in. This name corresponds to the fgVdEntName used in fgVdTable.
fgVWLHealthCheckLinkBandwidthIn	.1.3.6.1.4.1.12356.101.4.9.2.1.11	The available bandwidth of incoming traffic detected by a health check on a specific member link, in Mbps,
fgVWLHealthCheckLinkBandwidthOut	.1.3.6.1.4.1.12356.101.4.9.2.1.12	The available bandwidth of outgoing traffic detected by a health check on a specific member link, in Mbps.
fgVWLHealthCheckLinkBandwidthBi	.1.3.6.1.4.1.12356.101.4.9.2.1.13	The available bandwidth of bi-direction traffic detected by a health check on a specific member link, in Mbps.
fgVWLHealthCheckLinkIfName	.1.3.6.1.4.1.12356.101.4.9.2.1.14	SD-WAN member interface name.

Example

This example shows an SD-WAN health check configuration and its collected statistics.

To configure the SD-WAN health check:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "port1"
            set gateway 192.168.2.1
        next
        edit 2
            set interface "MPLS"
            set zone "SD-Zone2"
```

```
            set cost 20
        next
        edit 3
            set interface "port2"
        next
    end
    config health-check
        edit "pingserver"
            set server "8.8.8.8"
            set sla-fail-log-period 10
            set sla-pass-log-period 20
            set members 2 1 3
            config sla
                edit 1
                    set link-cost-factor jitter packet-loss
                    set packetloss-threshold 2
                next
            end
        next
    end
end
```

The collected statistics:

fgVWLHealthCheckLinkID	.1.3.6.1.4.1.12356.101.4.9.2.1.1	1	2	3
fgVWLHealthCheckLinkName	.1.3.6.1.4.1.12356.101.4.9.2.1.2	pingserver	pingserver	pingserver
fgVWLHealthCheckLinkSeq	.1.3.6.1.4.1.12356.101.4.9.2.1.3	2	1	3
fgVWLHealthCheckLinkState	.1.3.6.1.4.1.12356.101.4.9.2.1.4	0	0	0
fgVWLHealthCheckLinkLatency	.1.3.6.1.4.1.12356.101.4.9.2.1.5	39.302	43.124	44.348
fgVWLHealthCheckLinkJitter	.1.3.6.1.4.1.12356.101.4.9.2.1.6	4.346	3.951	5.05
fgVWLHealthCheckLinkPacketSend	.1.3.6.1.4.1.12356.101.4.9.2.1.7	3657689	3657689	3657689
fgVWLHealthCheckLinkPacketRecv	.1.3.6.1.4.1.12356.101.4.9.2.1.8	3196258	3220258	3219466
fgVWLHealthCheckLinkPacketLoss	.1.3.6.1.4.1.12356.101.4.9.2.1.9	0	0	0
fgVWLHealthCheckLinkVdom	.1.3.6.1.4.1.12356.101.4.9.2.1.10	root	root	root

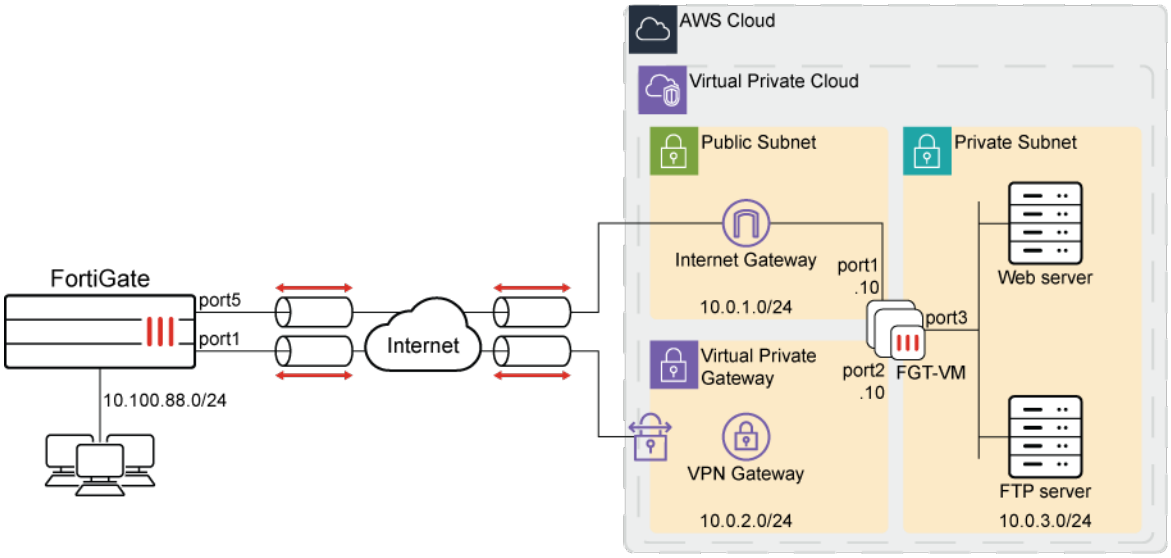
fgVWLHealthCheckLinkBandwidthIn	.1.3.6.1.4.1.12356.101.4.9.2.1.11	9999963	9999937	9999999
fgVWLHealthCheckLinkBandwidthOut	.1.3.6.1.4.1.12356.101.4.9.2.1.12	9999981	9999953	9999998
fgVWLHealthCheckLinkBandwidthBi	.1.3.6.1.4.1.12356.101.4.9.2.1.13	19999944	19999890	19999997
fgVWLHealthCheckLinkIfName	.1.3.6.1.4.1.12356.101.4.9.2.1.14	MPLS	port1	port2

SD-WAN cloud on-ramp

In this example, you configure a connection to a new cloud deployment that has some remote servers. SD-WAN is used to steer traffic through the required overlay tunnel.

The on-premise FortiGate has two internet connections, each with a single VPN connection. The two VPN gateways are configured on the cloud for redundancy, one terminating at the FortiGate-VM, and the other at the native AWS VPN Gateway.

This example uses AWS as the Infrastructure as a Service (IaaS) provider, but the same configuration can also apply to other services. A full mesh VPN setup is not shown, but can be added later if required.



To connect to the servers that are behind the cloud FortiGate-VM, virtual IP addresses (VIPs) are configured on port2 to map to the servers:

- VPN traffic terminating on port1 is routed to the VIP on port2 to access the web servers.
- VPN traffic terminating on the VPN gateway accesses the VIPs on port2 directly.

There are four major steps to configure this setup:

1. Configuring the VPN overlay between the HQ FortiGate and cloud FortiGate-VM on page 1568
2. Configuring the VPN overlay between the HQ FortiGate and AWS native VPN gateway on page 1572
3. Configuring the VIP to access the remote servers on page 1576
4. Configuring the SD-WAN to steer traffic between the overlays on page 1579

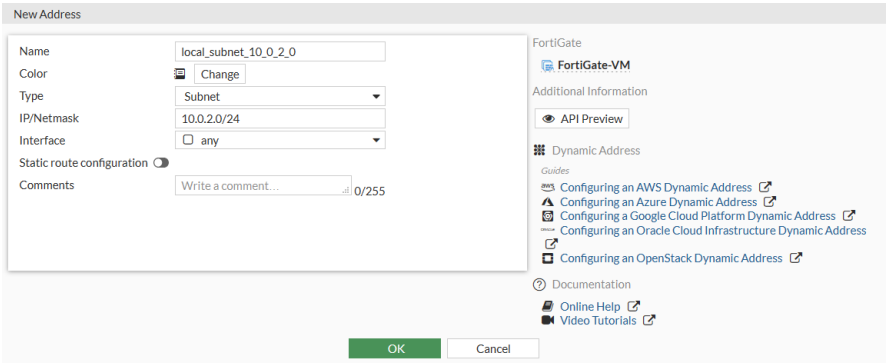
After the configuration is complete, verify the traffic to ensure that the configuration is working as expected, see Verifying the traffic on page 1582.

Configuring the VPN overlay between the HQ FortiGate and cloud FortiGate-VM

Configure the cloud FortiGate-VM

To create an address for the VPN gateway:

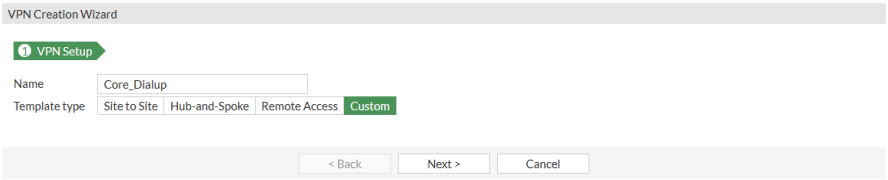
1. Go to *Policy & Objects* > *Addresses* and select *Address*.
2. Click *Create new*.
3. Set *Name* to *local_subnet_10_0_2_0*.
4. Set *IP/Netmask* to *10.0.2.0/24*.



5. Click *OK*.

To configure a custom IPsec VPN:

1. Go to *VPN* > *IPsec Wizard*.
2. Set *Name* to *Core_Dialup*.
3. Set *Template type* to *Custom*.

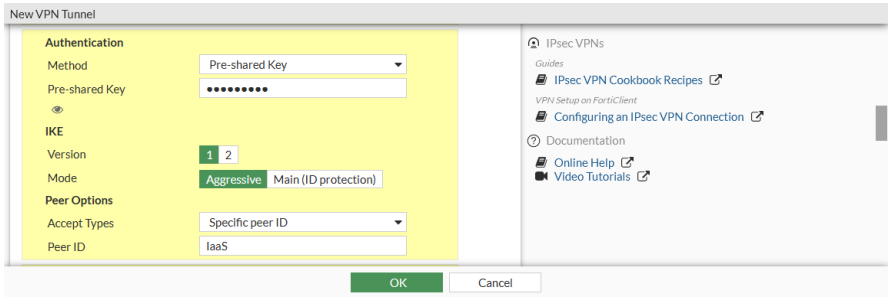


4. Click *Next*.
5. Configure *Network* settings:

Remote Gateway	Dialup User
Interface	port1
NAT Traversal	Enable

6. Configure *Authentication* settings:

Method	Pre-shared Key
Pre-shared Key	Enter the pre-shared key.
Version	1
Mode	Aggressive This setting allows the peer ID to be specified.
Accept Types	Specific peer ID
Peer ID	IaaS The other end of the tunnel needs to have its local ID set to IaaS.



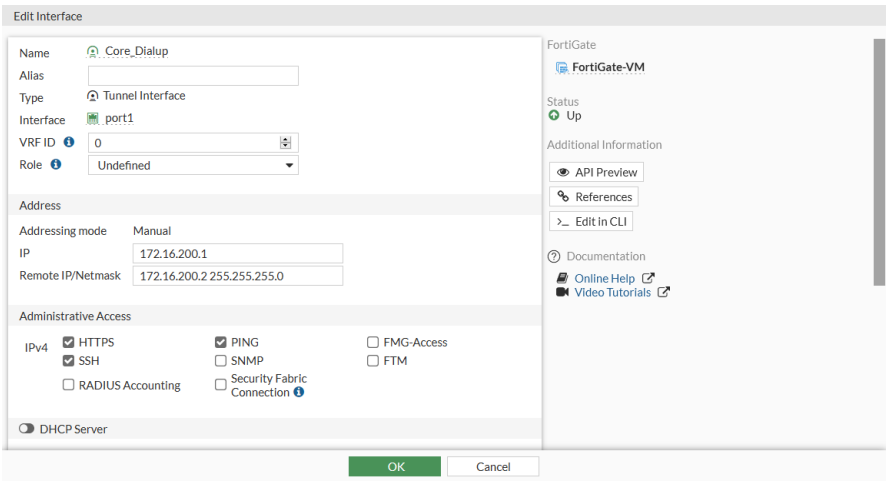
7. Leave the default *Phase 1 Proposal* settings and disable *XAUTH*.
8. Configure the *Phase 2 Selector* settings:

Name	Ent_Core
Local Address	Named Address - *local_subnet_10_0_2_0*
Remote Address	Named Address - *all* This setting allows traffic originating from both the remote subnet 10.100.88.0 and the health checks from the VPN interface on the remote FortiGate. For increased security, each subnet can be specified individually.

9. Click *OK*.

To configure remote and local tunnel IP addresses:

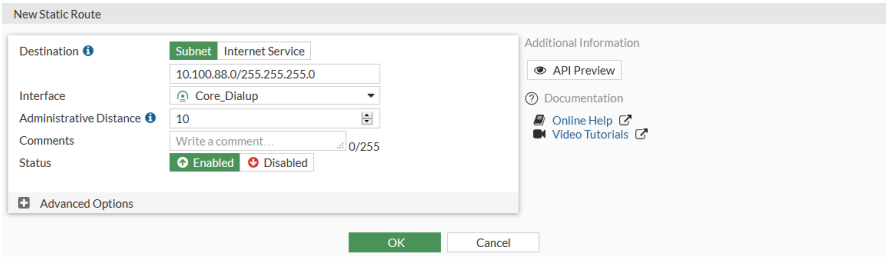
1. Go to *Network* > *Interfaces* and edit the *Core_Dialup* interface under *port1*.
2. Set *IP* to *172.16.200.1*.
3. Set *Remote IP/Netmask* to *172.16.200.2 255.255.255.0*. This is where remote health check traffic will come from.
4. Enable *Administrative access* for *HTTPS*, *PING*, and *SSH*.



5. Click *OK*.

To configure a route to the remote subnet through the tunnel:

1. Go to *Network* > *Static Routes* and click *Create New*.
2. Set *Destination* to *Subnet* and enter the IP address and netmask: *10.100.88.0/255.255.255.0*.
3. Set *Interface* to *Core_Dialup*.

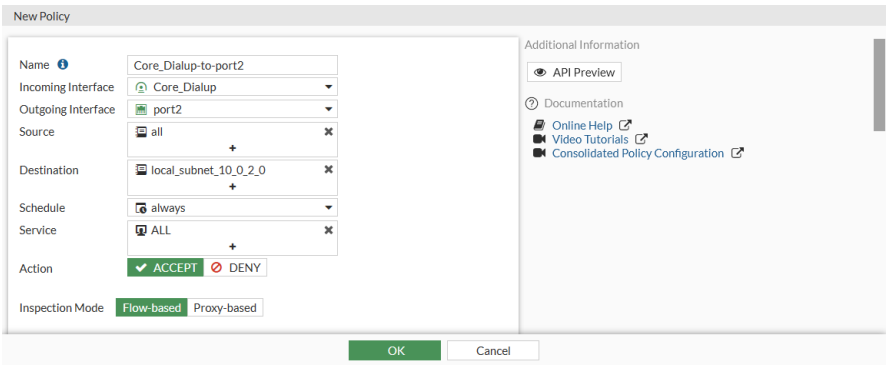


4. Click *OK*.

To configure a firewall policy to allow traffic from the tunnel to port2:

1. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
2. Configure the following:

Name	Core_Dialup-to-port2
Incoming Interface	Core_Dialup
Outgoing Interface	port2
Source	all
Destination	local_subnet_10_0_2_0
Schedule	always
Service	ALL
Action	ACCEPT



3. Configure the remaining settings as required.
4. Click *OK*.

Configure the HQ FortiGate

To create an address for the VPN gateway:

1. Go to *Policy & Objects* > *Addresses* and select *Address*.
2. Click *Create new*.
3. Set *Name* to *remote_subnet_10_0_2_0*.
4. Set *IP/Netmask* to *10.0.2.0/24*.
5. Click *OK*.

To configure a custom IPsec VPN:

1. Go to *VPN* > *IPsec Wizard*.
2. Set *Name* to *FGT_AWS_Tun*.
3. Set *Template type* to *Custom*.
4. Click *Next*.
5. Configure *Network* settings:

Remote Gateway	Static IP Address
IP Address	100.21.29.17
Interface	port5
NAT Traversal	Enable

6. Configure *Authentication* settings:

Method	Pre-shared Key
Pre-shared Key	Enter the pre-shared key.
Version	1
Mode	Aggressive This setting allows the peer ID to be specified.

Accept Types	Any peer ID

7. Leave the default *Phase 1 Proposal* settings, except set *Local ID* to *IaaS*.
8. Disable *XAUTH*.
9. Configure the *Phase 2 Selector* settings:

Name	FGT_AWS_Tun
Local Address	Named Address - *all* This setting allows traffic originating from both the local subnet 10.100.88.0 and the health checks from the VPN interface. For increased security, each subnet can be specified individually.
Remote Address	Named Address - *remote_subnet_10_0_2_0*

10. Click *OK*.

To configure local and remote tunnel IP addresses:

1. Go to *Network* > *Interfaces* and edit the *FGT_AWS_Tun* interface under *port5*.
2. Set *IP* to *172.16.200.2*.
3. Set *Remote IP/Netmask* to *172.16.200.1 255.255.255.0*.
4. Enable *Administrative access* for *HTTPS*, *PING*, and *SSH*.
5. Click *OK*.

Routing is defined when creating the SD-WAN interface. The firewall policy is created after the SD-WAN interface is defined.

Configuring the VPN overlay between the HQ FortiGate and AWS native VPN gateway

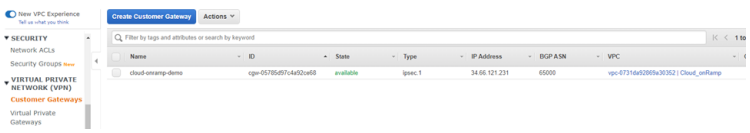
This example uses static routing. It is assumed that the AWS VPN Gateway is already configured, and that proper routing is applied on the corresponding subnet.

Verify the AWS configuration

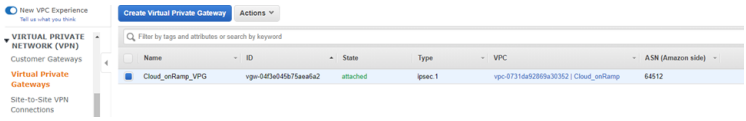
See Creating routing tables and associate subnets in the AWS Administration Guide for configuration details.

To check the AWS configuration:

1. Go to *Virtual Private Network (VPN)* > *Customer Gateways* to confirm that the customer gateway defines the FortiGate IP address as its Gateway IP address, in this case *34.66.121.231*.

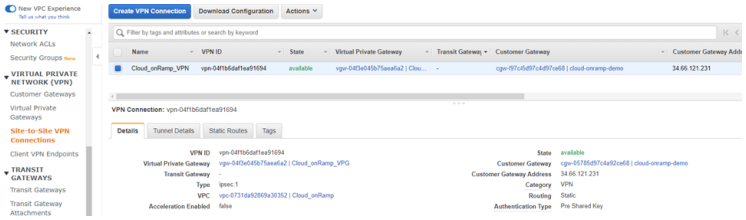


2. Go to *Virtual Private Network (VPN)* > *Virtual Private Gateways* to confirm that a virtual private gateway (VPG) has been created. In this case it is attached to the *Cloud_onRamp* VPC that contains the FortiGate and servers.

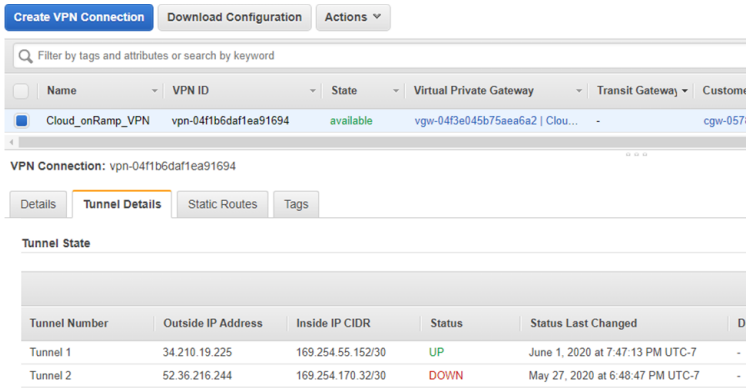


3. Go to *Virtual Private Network (VPN)* > *Site-to-Site VPN Connections* to confirm that site-to-site VPN connections have been created and attached to the customer gateway and virtual private gateway.

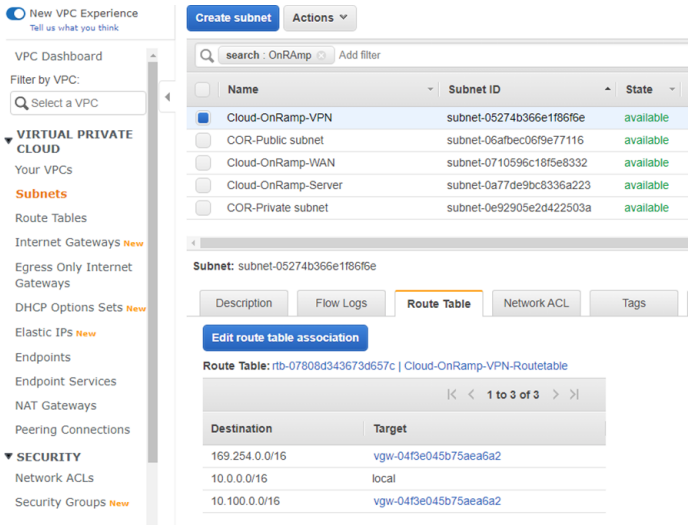
   If *Routing Options* is *Static*, the IP prefix of the remote subnet on the HQ FortiGate (10.100.88.0) is entered here.



   AWS site-to-site VPN always creates two VPN tunnels for redundancy. In this example, only Tunnel 1 is used.



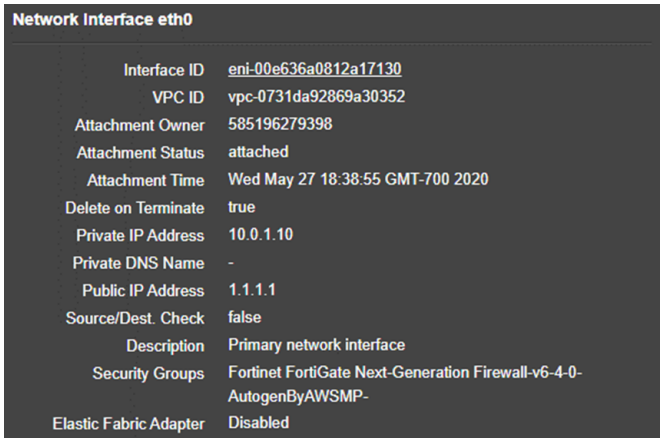
4. Click *Download Configuration* to download the FortiGate's tunnel configurations. The configuration can be referred to when configuring the FortiGate VPN.
5. The new VPG is attached to your VPC, but to successfully route traffic to the VPG, proper routing must be defined. Go to *Virtual Private Cloud* > *Subnets*, select the *Cloud-OnRamp-VPN*, and select the *Route Table* tab to verify that there are at least two routes to send traffic over the VPG.



- *169.254.0.0/24* defines the tunnel IP address. Health check traffic originating from the FortiGate will come from this IP range.
- *10.100.0.0/16* defines the remote subnet from the HQ FortiGate.
- Both routes point to the just created VPG *vgw-04xxxx*.

6. On the cloud FortiGate-VM EC2 instances, ensure that port1 and port2 both have *Source/Dest. Check* set to *false*. This allows the FortiGate to accept and route traffic to and from a different network.

   If you launched the instance from the AWS marketplace, this setting defaults to *true*.



Configure routing to the VPG on the cloud FortiGate-VM

To configure routing to the VPG on the cloud FortiGate-VM:

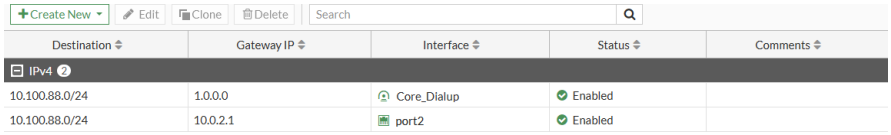
1. Go to *Network* > *Static Routes* and click *Create New*.
2. Set *Destination* to *Subnet* and enter the IP address and netmask: *10.100.88.0/255.255.255.0*.
3. Set *Gateway Address* to *Specify* and enter *10.0.2.1*.
4. Set *Interface* to *port2*.

   The new route must have the same *Administrative Distance* as the route that was created for traffic through the *Core_Dialup* tunnel to ensure that both routes are added to the routing table (see To configure a route

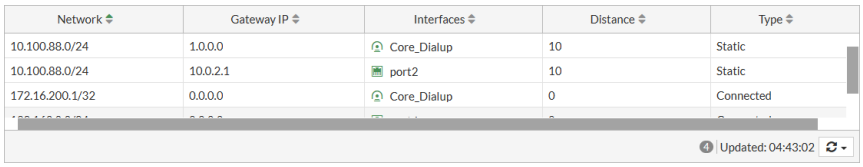
to the remote subnet through the tunnel).

The *Gateway Address* is arbitrarily set to 10.0.2.1. The VPG does not have an IP address, but the address defined here allows the FortiGate to route traffic out of port2, while AWS routes the traffic based on its routing table.

5. Click *OK*.
6. Go to *Network* > *Static Routes* to view the configured static routes:



7. Go to *Dashboard* > *Network Monitor* > *Routing* to view the routing table.



Configure IPsec VPN on the HQ FortiGate

To configure a custom IPsec VPN:

1. Go to *VPN* > *IPsec Wizard*.
2. Set *Name* to *AWS_VPG*.
3. Set *Template type* to *Custom*.
4. Click *Next*.
5. Configure *Network* settings:

Remote Gateway	Static IP Address
IP Address	34.210.19.225 This address is taken from the downloaded AWS configuration file.
Interface	port1
NAT Traversal	Enable

6. Configure *Authentication* settings:

Method	Pre-shared Key
Pre-shared Key	Enter the pre-shared key.
Version	1
Mode	Main

7. Configure the *Phase 1 Proposal* settings using information from the downloaded AWS configuration file.
8. Disable *XAUTH*.
9. Configure the *Phase 2 Selector* settings:

Name	AWS_VPG
Local Address	Named Address - *all* This setting allows traffic originating from both the local subnet 10.100.88.0 and the health checks from the VPN interface. For increased security, each subnet can be specified individually.
Remote Address	Named Address - *remote_subnet_10_0_2_0*

10. Click *OK*.

To configure local and remote tunnel IP addresses:

1. Go to *Network* > *Interfaces* and edit the *AWS_VPG* interface under *port1*.
2. Set *IP* to *169.254.55.154*.
3. Set *Remote IP/Netmask* to *169.254.55.153 255.255.255.0*.
4. Enable *Administrative access* for *HTTPS* and *PING*.
5. Click *OK*.

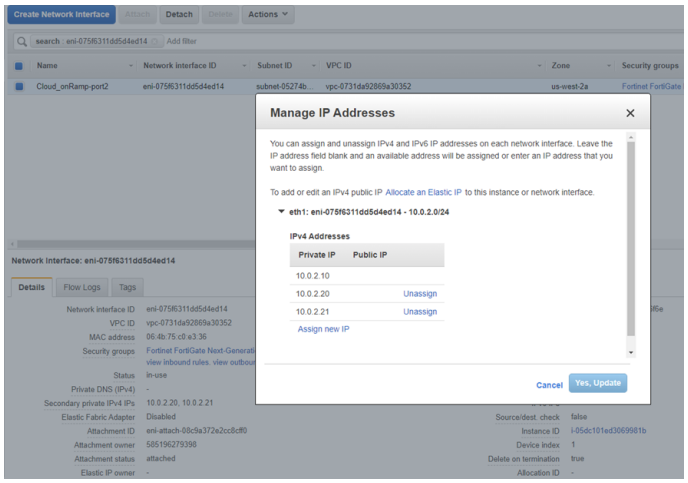
Routing is defined when creating the SD-WAN interface. The firewall policy is created after the SD-WAN interface is defined.

Configuring the VIP to access the remote servers

VIPs, interface IP addresses, and policies are created on the cloud FortiGate-VM to allow access to the remote servers.

To configure additional private IPs on AWS for the FortiGate VIP:

1. On the FortiGate EC2 instance, edit the *Elastic Network Interface* that corresponds to *port2*. In this example, Network Interface *eth1*.
2. Go to *Actions* > *Manage IP Addresses*.
3. Add two private IP address in the 10.0.2.0/24 subnet.
   These address will be used in the VIPs on the FortiGate. This ensures that traffic to these IP addresses is routed to the FortiGate by AWS.

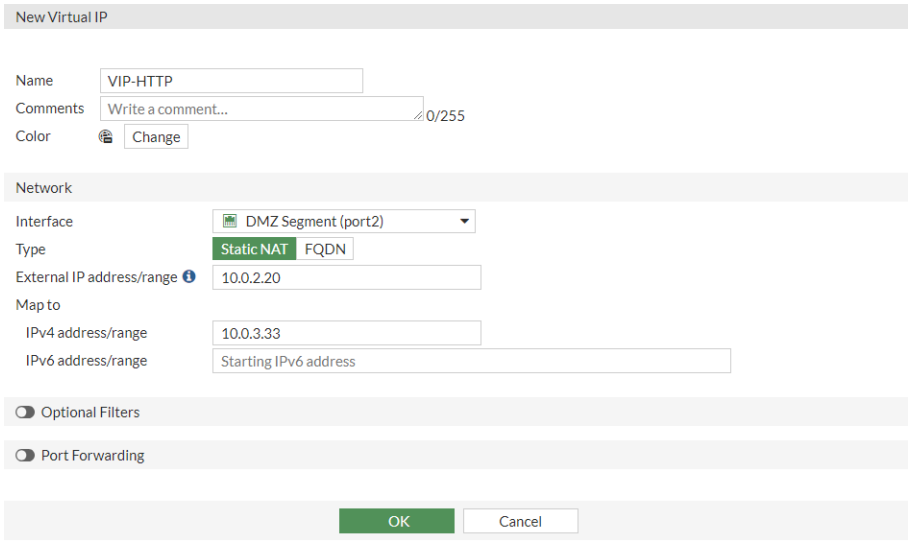


4. Click *Yes, Update*.

To configure VIPs on the cloud FortiGate-VM:

1. Go to *Policy & Objects* > *Virtual IPs* and select the *Virtual IP* tab.
2. Click *Create new*.
3. Configure the following:

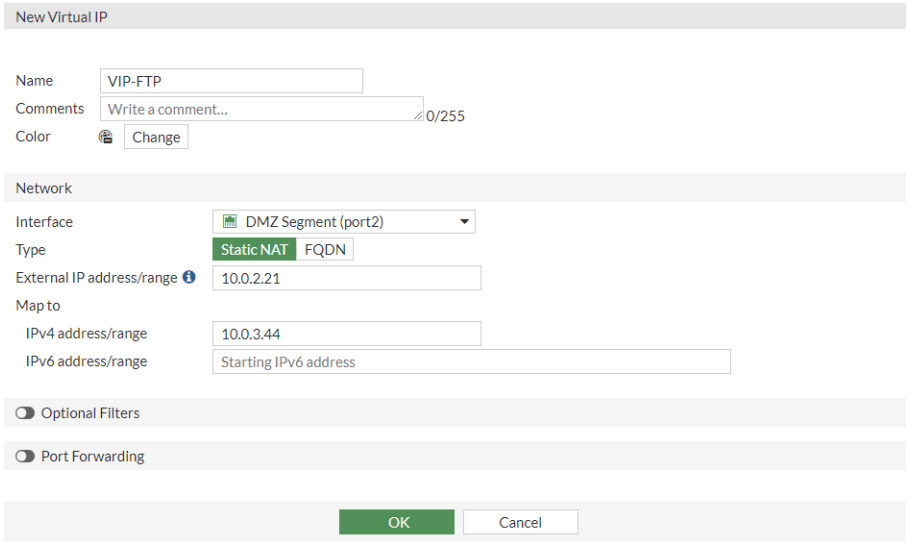
Name	VIP-HTTP
Interface	port2
External IP address/range	10.0.2.20
Map to IPv4 address/range	10.0.3.33



4. Click *OK*.
5. Create a second VIP for the FTP server with the following settings:

Name	VIP-FTP

Interface	port2
External IP address/range	10.0.2.21
Map to IPv4 address/range	10.0.3.44



To configure firewall policies to allow traffic from port2 to port3:

1. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
2. Configure the following:

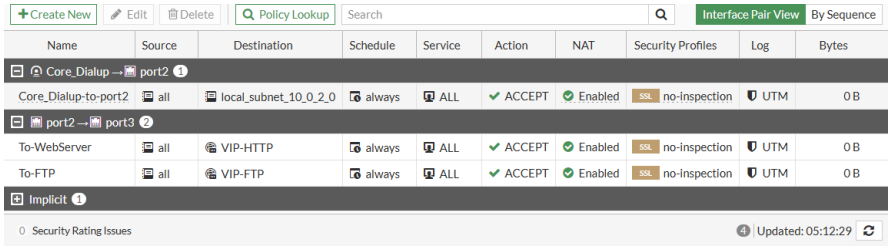
Name	To-WebServer
Incoming Interface	port2
Outgoing Interface	port3
Source	all
Destination	VIP-HTTP
Schedule	always
Service	ALL
Action	ACCEPT
NAT	Enabled

3. Configure the remaining settings as required.
4. Click *OK*.
5. Create a second policy for the FTP VIP with the following settings:

Name	To-FTP
Incoming Interface	port2

Outgoing Interface	port3
Source	all
Destination	VIP-FTP
Schedule	always
Service	ALL
Action	ACCEPT
NAT	Enabled

6. Click *OK*.



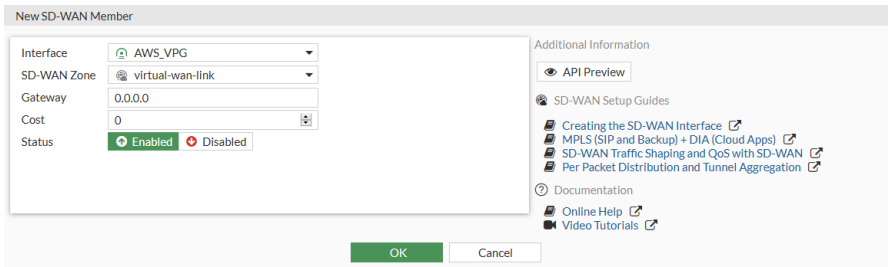
Configuring the SD-WAN to steer traffic between the overlays

Configure the HQ FortiGate to use two overlay tunnels for SD-WAN, steering HTTPS and HTTP traffic through the FGT_AWS_Tun tunnel, and SSH and FTP throguh the AWS_VPG tunnel.

1. Add SD-WAN member interfaces
2. Configure a route to the remote network
3. Configure firewall policies
4. Configure a health check
5. Configure SD-WAN rules

To add SD-WAN member interfaces:

1. Go to *Network* > *SD-WAN*, select the *SD-WAN Zones* tab, and click *Create New* > *SD-WAN Member*.
2. Set *Interface* to *AWS_VPG* then click *OK*.

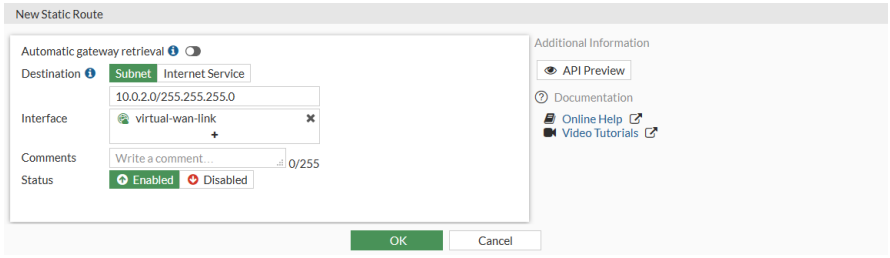


3. Click *Create New* > *SD-WAN Member* again.
4. Set *Interface* to *FGT_AWS_Tun*.

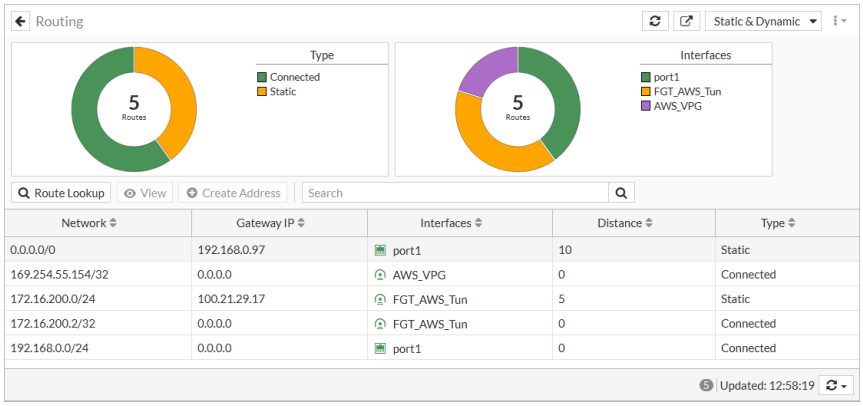
5. Set *Gateway* to *172.16.200.1*.
6. Click *OK*.

To configure a route to the remote network 10.0.2.0/24:

1. Go to *Network* > *Static Routes* and click *Create New*.
2. Set *Destination* to *Subnet* and enter the IP address and netmask: *10.0.2.0/255.255.255.0*.
3. Set *Interface* to *virtual-wan-link*.



4. Click *OK*.
   Individual routes to each tunnel are automatically added to the routing table with the same distance:



To configure firewall policies to allow traffic from the internal subnet to SD-WAN:

1. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
2. Configure the following:

Name	ISFW-to-IaaS
Incoming Interface	port3
Outgoing Interface	virtual-wan-link
Source	all

Destination	all
Schedule	always
Service	ALL
Action	ACCEPT
NAT	Enabled

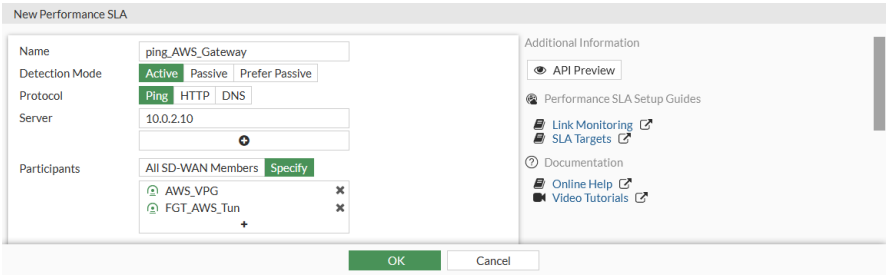
3. Configure the remaining settings as required.
4. Click *OK*.
   Once the firewall policies are configured, the VPN tunnels should come up when there is traffic.

To configure a health check to monitor the status of the tunnels:

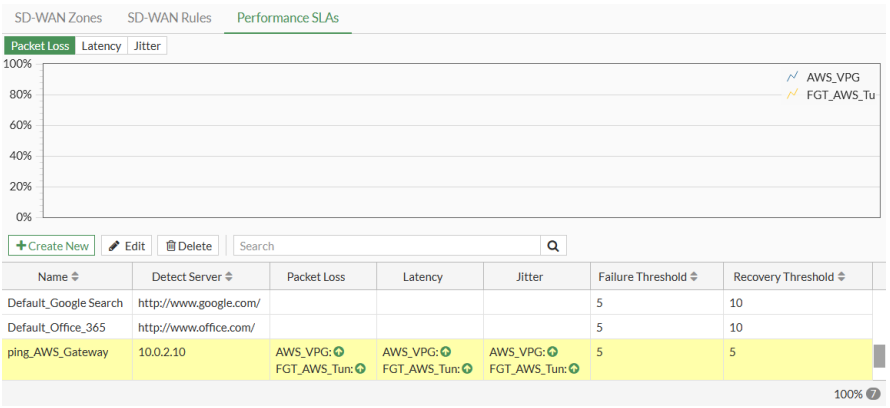
As you are accessing the servers on the 10.0.2.0/24 subnet, it is preferable to use the FortiGate port2 interface as the ping server for detection. This ensures that, if the gateway is not reachable in either tunnel, its routes are brought down and traffic continues on the other tunnel.

1. Go to *Network* > *SD-WAN*, select the *Performance SLAs* tab, and click *Create New*.
2. Configure the following:

Name	ping_AWS_Gateway
Protocol	Ping
Server	10.0.2.10
Participants	Specify Add AWS_VPG and FGT_AWS_Tun as participants.



3. Click *OK*.



Health check probes originate from the VPN interface's IP address. This is why the phase2 selectors are configured with *Local Address* set to *all*.

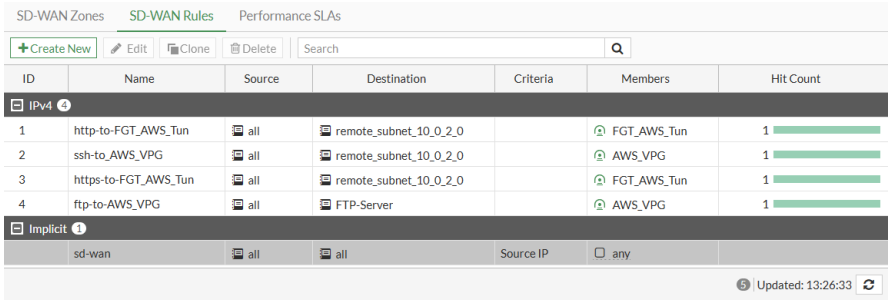
To configure SD-WAN rules to steer traffic:

HTTPS and HTTP traffic is steered to the FGT_AWS_Tun tunnel, and SSH and FTP traffic is steered to the AWS_VPG tunnel. The Manual algorithm is used in this example.

1. Go to *Network* > *SD-WAN*, select the *SD-WAN Rules* tab, and click *Create New*.
2. Configure the following:

Name	http-to-FGT_AWS_Tun
Source Address	all
Address	remote_subnet_10_0_2_0
Protocol	TCP
Port range	80 - 80
Outgoing Interfaces	Manual
Interface preference	FGT_AWS_Tun

3. Click *OK*.
4. Create other SD-WAN rules as required:



Verifying the traffic

To verify that pings are sent across the IPsec VPN tunnels

- On the HQ FortiGate, run the following CLI command:

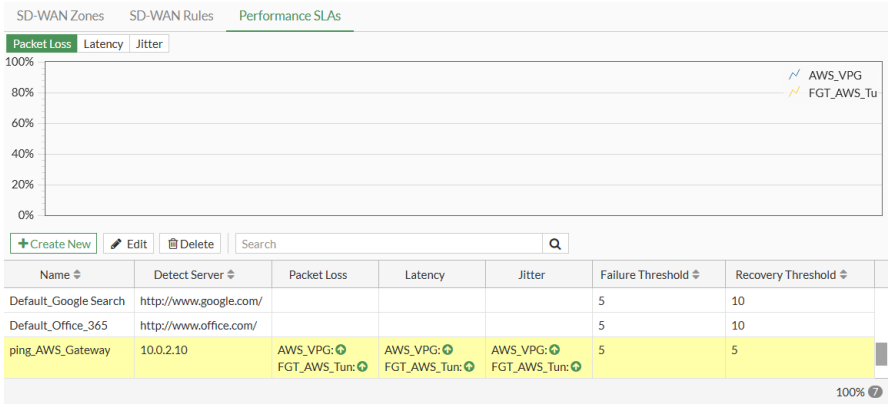
```
# diagnose sniffer packet any 'host 10.0.2.10' 4 0 1 interfaces=[any]
Using Original Sniffing Mode
interfaces=[any]
filters=[host 10.0.2.10]
pcap_snapshot: snaplen raised from 0 to 262144
2021-06-05 11:35:14.822600 AWS_VPG out 169.254.55.154 -> 10.0.2.10: icmp: echo request
2021-06-05 11:35:14.822789 FGT_AWS_Tun out 172.16.200.2 -> 10.0.2.10: icmp: echo request
2021-06-05 11:35:14.877862 FGT_AWS_Tun in 10.0.2.10 -> 172.16.200.2: icmp: echo reply
2021-06-05 11:35:14.878887 AWS_VPG in 10.0.2.10 -> 169.254.55.154: icmp: echo reply
```

- On the cloud FortiGate-VM, run the following CLI command:

```
# diagnose sniffer packet any 'host 10.0.2.10' 4 0 1 interfaces=[any]
Using Original Sniffing Mode
interfaces=[any]
filters=[host 10.0.2.10]
pcap_snapshot: snaplen raised from 0 to 262144
2021-06-05 11:37:57.176329 port2 in 169.254.55.154 -> 10.0.2.10: icmp: echo request
2021-06-05 11:37:57.176363 port2 out 10.0.2.10 -> 169.254.55.154: icmp: echo reply
2021-06-05 11:37:57.176505 Core_Dialup in 172.16.200.2 -> 10.0.2.10: icmp: echo request
2021-06-05 11:37:57.176514 Core_Dialup out 10.0.2.10 -> 172.16.200.2: icmp: echo reply
```

To verify the SLA health checks on the HQ FortiGate:

1. Go to *Network* > *SD-WAN* > *Performance SLAs*, select *Packet Loss*, and click the *ping_AWS_Gateway* SLA:

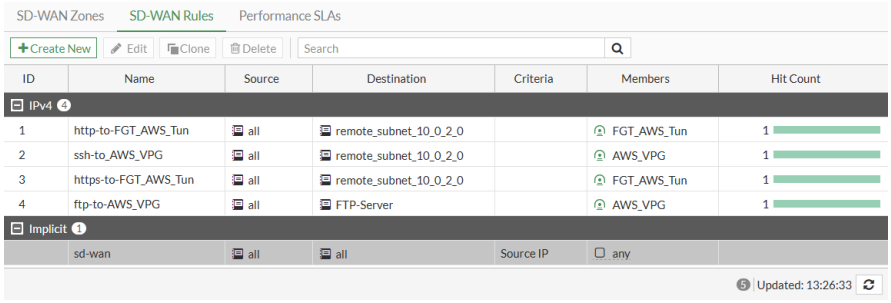


2. Run the following CLI command:

```
# diagnose sys sdwan health-check
…
Seq(1 AWS_VPG): state(alive), packet-loss(0.000%) latency(56.221), jitter(0.290) sla_map=0x0
Seq(2 FGT_AWS_Tun): state(alive), packet-loss(0.000%) latency(55.039), jitter(0.223) sla_
map=0x0
```

To verify service rules:

1. Go to *Network* > *SD-WAN* and select the *SD-WAN Rules* tab:



2. Run the following CLI command:

```
# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x0
  Gen(1), TOS(0x0/0x0), Protocol(6: 80->80), Mode(manual)
  Members:
    1: Seq_num(2 FGT_AWS_Tun), alive, selected
  Src address:
        0.0.0.0-255.255.255.255
  Dst address:
        10.0.2.0-10.0.2.255

Service(2): Address Mode(IPV4) flags=0x0
  Gen(1), TOS(0x0/0x0), Protocol(6: 22->22), Mode(manual)
  Members:
    1: Seq_num(1 AWS_VPG), alive, selected
  Src address:
        0.0.0.0-255.255.255.255
  Dst address:
        10.0.2.0-10.0.2.255

Service(3): Address Mode(IPV4) flags=0x0
  Gen(1), TOS(0x0/0x0), Protocol(6: 443->443), Mode(manual)
  Members:
    1: Seq_num(2 FGT_AWS_Tun), alive, selected
  Src address:
        0.0.0.0-255.255.255.255
  Dst address:
        10.0.2.0-10.0.2.255

Service(4): Address Mode(IPV4) flags=0x0
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
  Members:
    1: Seq_num(1 AWS_VPG), alive, selected
  Src address:
        0.0.0.0-255.255.255.255
  Dst address:
        10.0.2.21-10.0.2.21
```

To verify that sessions are going to the correct tunnel:

1. Run the following CLI command to verify that HTTPS and HTTP traffic destined for the Web server at 10.0.2.20 uses FGT_AWS_Tun:

```
# diagnose sys session filter dst 10.0.2.20
# diagnose sys session list

session info: proto=6 proto_state=11 duration=2 expire=3597 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=4
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=FGT_AWS_Tun/ vlan_cos=0/255
```

```
state=log may_dirty npu f00 csf_syncd_log app_valid
statistic(bytes/packets/allow_err): org=593/4/1 reply=3689/5/1 tuples=3
tx speed(Bps/kbps): 264/2 rx speed(Bps/kbps): 1646/13
orgin->sink: org pre->post, reply pre->post dev=0->18/18->0 gwy=172.16.200.1/0.0.0.0
hook=post dir=org act=snat 10.100.88.101:55589->10.0.2.20:80(172.16.200.2:55589)
hook=pre dir=reply act=dnat 10.0.2.20:80->172.16.200.2:55589(10.100.88.101:55589)
hook=post dir=reply act=noop 10.0.2.20:80->10.100.88.101:55589(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
src_mac=00:09:0f:00:03:01
misc=0 policy_id=32 auth_info=0 chk_client_info=0 vd=0
serial=00b7442c tos=ff/ff app_list=2000 app=34050 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id= ff000001 rpdb_svc_id=2154552596 ngfwid=n/a
npu_state=0x3041008

session info: proto=6 proto_state=66 duration=1 expire=3 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=4
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=FGT_AWS_Tun/ vlan_cos=0/255
state=log may_dirty ndr f00 csf_syncd_log
statistic(bytes/packets/allow_err): org=48/1/0 reply=40/1/1 tuples=3
tx speed(Bps/kbps): 26/0 rx speed(Bps/kbps): 22/0
orgin->sink: org pre->post, reply pre->post dev=5->18/18->5 gwy=172.16.200.1/10.100.88.101
hook=post dir=org act=snat 10.100.88.101:55621->10.0.2.20:443(172.16.200.2:55621)
hook=pre dir=reply act=dnat 10.0.2.20:443->172.16.200.2:55621(10.100.88.101:55621)
hook=post dir=reply act=noop 10.0.2.20:443->10.100.88.101:55621(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
src_mac=00:09:0f:00:03:01
misc=0 policy_id=32 auth_info=0 chk_client_info=0 vd=0
serial=00b74b50 tos=ff/ff app_list=2000 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id= ff000003 rpdb_svc_id=2154552596 ngfwid=n/a
npu_state=0x3041008
```

2. Run the following CLI command to verify that SSH and FTP traffic destined for the FTP server at 10.0.2.21 uses AWS_VPG:

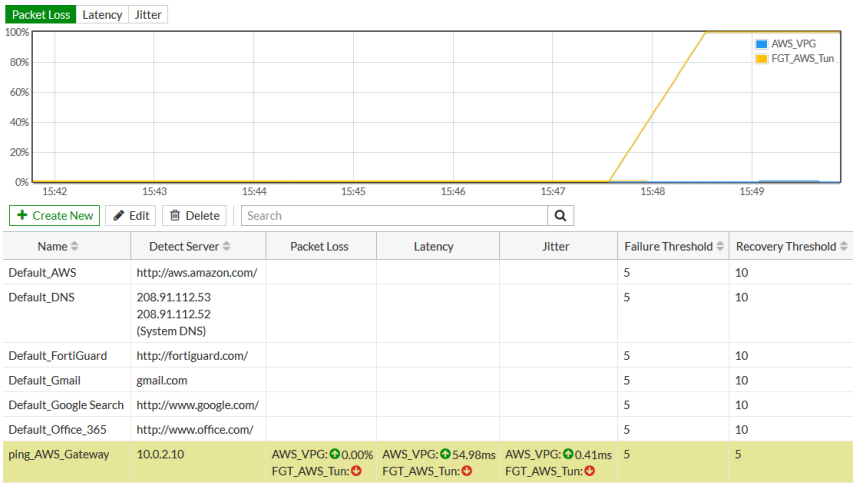
```
# diagnose sys session filter dst 10.0.2.20
# diagnose sys session list

session info: proto=6 proto_state=11 duration=197 expire=3403 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=4
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=AWS_VPG/ helper=ftp vlan_cos=0/255
state=log may_dirty ndr npu f00 csf_syncd_log app_valid
statistic(bytes/packets/allow_err): org=580/12/1 reply=863/13/1 tuples=3
tx speed(Bps/kbps): 2/0 rx speed(Bps/kbps): 4/0
orgin->sink: org pre->post, reply pre->post dev=5->17/17->5 gwy=169.254.55.153/10.100.88.101
```

```
hook=post dir=org act=snat 10.100.88.101:55528->10.0.2.21:21(169.254.55.154:55528)
hook=pre dir=reply act=dnat 10.0.2.21:21->169.254.55.154:55528(10.100.88.101:55528)
hook=post dir=reply act=noop 10.0.2.21:21->10.100.88.101:55528(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
src_mac=00:09:0f:00:03:01
misc=0 policy_id=32 auth_info=0 chk_client_info=0 vd=0
serial=00b72a5f tos=ff/ff app_list=2000 app=15896 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id= ff000004 rpdb_svc_id=2149689849 ngfwid=n/a
npu_state=0x3041008

session info: proto=6 proto_state=11 duration=3 expire=3596 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=4
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=AWS_VPG/ vlan_cos=0/255
state=log may_dirty ndr npu f00 csf_syncd_log app_valid
statistic(bytes/packets/allow_err): org=1496/6/1 reply=1541/5/1 tuples=3
tx speed(Bps/kbps): 416/3 rx speed(Bps/kbps): 429/3
orgin->sink: org pre->post, reply pre->post dev=5->17/17->5 gwy=169.254.55.153/10.100.88.101
hook=post dir=org act=snat 10.100.88.101:55644->10.0.2.21:22(169.254.55.154:55644)
hook=pre dir=reply act=dnat 10.0.2.21:22->169.254.55.154:55644(10.100.88.101:55644)
hook=post dir=reply act=noop 10.0.2.21:22->10.100.88.101:55644(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
src_mac=00:09:0f:00:03:01
misc=0 policy_id=32 auth_info=0 chk_client_info=0 vd=0
serial=00b75287 tos=ff/ff app_list=2000 app=16060 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id= ff000002 rpdb_svc_id=2149689849 ngfwid=n/a
npu_state=0x3041008
```

To simulate an issue on an overlay VPN tunnel:

On the cloud FortiGate-VM, disable the firewall policy allowing Core_Dialup to port2.

1. Health-checks through the FGT_AWS_Tun tunnel fail:
   a. Go to *Network > SD-WAN*, select the *Performance SLAs* tab, select *Packet Loss*, and click the *ping_AWS_Gateway* SLA:

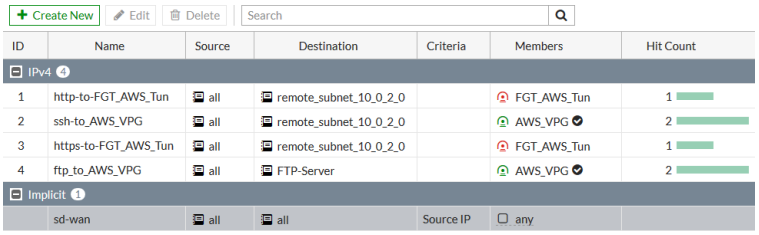


Name	Detect Server	Packet Loss	Latency	Jitter	Failure Threshold	Recovery Threshold
Default_AWS	http://aws.amazon.com/				5	10
Default_DNS	208.91.112.53 208.91.112.52 (System DNS)				5	10
Default_FortiGuard	http://fortiguard.com/				5	10
Default_Gmail	gmail.com				5	10
Default_Google Search	http://www.google.com/				5	10
Default_Office_365	http://www.office.com/				5	10
ping_AWS_Gateway	10.0.2.10	AWS_VPG: 0.00% FGT_AWS_Tun:	AWS_VPG: 54.98ms FGT_AWS_Tun:	AWS_VPG: 0.41ms FGT_AWS_Tun:	5	5

b. Run the following CLI command:

```
# diagnose sys sdwan health-check
…
Seq(1 AWS_VPG): state(alive), packet-loss(0.000%) latency(52.746), jitter(0.713) sla_
map=0x0
Seq(2 FGT_AWS_Tun): state(dead), packet-loss(19.000%) sla_map=0x0
```

2. Service rules show that the member is down:
   a. Go to *Network* > *SD-WAN* and select the *SD-WAN Rules* tab:



ID	Name	Source	Destination	Criteria	Members	Hit Count
IPv4 4						
1	http-to-FGT_AWS_Tun	all	remote_subnet_10_0_2_0		FGT_AWS_Tun	1
2	ssh-to_AWS_VPG	all	remote_subnet_10_0_2_0		AWS_VPG	2
3	https-to-FGT_AWS_Tun	all	remote_subnet_10_0_2_0		FGT_AWS_Tun	1
4	ftp_to_AWS_VPG	all	FTP-Server		AWS_VPG	2
Implicit 1						
	sd-wan	all	all	Source IP	any	

b. Run the following CLI command:

```
# diagnose sys sdwan service4

Service(1): Address Mode(IPV4) flags=0x0
  Gen(2), TOS(0x0/0x0), Protocol(6: 80->80), Mode(manual)
  Members:
    1: Seq_num(2 FGT_AWS_Tun), dead
  Src address:
        0.0.0.0-255.255.255.255
  Dst address:
        10.0.2.0-10.0.2.255

Service(2): Address Mode(IPV4) flags=0x0
  Gen(1), TOS(0x0/0x0), Protocol(6: 22->22), Mode(manual)
  Members:
    1: Seq_num(1 AWS_VPG), alive, selected
  Src address:
        0.0.0.0-255.255.255.255
```

```
  Dst address:
        10.0.2.0-10.0.2.255

Service(3): Address Mode(IPV4) flags=0x0
  Gen(2), TOS(0x0/0x0), Protocol(6: 443->443), Mode(manual)
  Members:
    1: Seq_num(2 FGT_AWS_Tun), dead
  Src address:
        0.0.0.0-255.255.255.255
  Dst address:
        10.0.2.0-10.0.2.255

Service(4): Address Mode(IPV4) flags=0x0
  Gen(1), TOS(0x0/0x0), Protocol(0: 1->65535), Mode(manual)
  Members:
    1: Seq_num(1 AWS_VPG), alive, selected
  Src address:
        0.0.0.0-255.255.255.255
  Dst address:
        10.0.2.21-10.0.2.21
```

3. Sessions are redirected to the working tunnel:
   a. Run the following CLI command:

```
# diagnose sys session list

session info: proto=6 proto_state=11 duration=3 expire=3596 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=4
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=AWS_VPG/ vlan_cos=0/255
state=log may_dirty ndr npu f00 csf_syncd_log app_valid
statistic(bytes/packets/allow_err): org=504/4/1 reply=620/3/1 tuples=3
tx speed(Bps/kbps): 150/1 rx speed(Bps/kbps): 184/1
orgin->sink: org pre->post, reply pre->post dev=0->17/17->0 gwy=169.254.55.153/0.0.0.0
hook=post dir=org act=snat 10.100.88.101:56373->10.0.2.20:80(169.254.55.154:56373)
hook=pre dir=reply act=dnat 10.0.2.20:80->169.254.55.154:56373(10.100.88.101:56373)
hook=post dir=reply act=noop 10.0.2.20:80->10.100.88.101:56373(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
src_mac=00:09:0f:00:03:01
misc=0 policy_id=32 auth_info=0 chk_client_info=0 vd=0
serial=00b87199 tos=ff/ff app_list=2000 app=34050 url_cat=0
rpdb_link_id= 80000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x3041008

session info: proto=6 proto_state=66 duration=3 expire=1 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=4
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=AWS_VPG/ vlan_cos=0/255
```

```
state=log may_dirty ndr f00 csf_syncd_log
statistic(bytes/packets/allow_err): org=48/1/0 reply=40/1/1 tuples=3
tx speed(Bps/kbps): 15/0 rx speed(Bps/kbps): 12/0
orgin->sink: org pre->post, reply pre->post dev=5->17/17->5
gwy=169.254.55.153/10.100.88.101
hook=post dir=org act=snat 10.100.88.101:56383->10.0.2.20:443(169.254.55.154:56383)
hook=pre dir=reply act=dnat 10.0.2.20:443->169.254.55.154:56383(10.100.88.101:56383)
hook=post dir=reply act=noop 10.0.2.20:443->10.100.88.101:56383(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
src_mac=00:09:0f:00:03:01
misc=0 policy_id=32 auth_info=0 chk_client_info=0 vd=0
serial=00b876bb tos=ff/ff app_list=2000 app=0 url_cat=0
rpdb_link_id= 80000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x3041008
total session 2
```

4. Routes to the *FGT_AWS_Tun* tunnel are removed:
   a. Go to *Dashboard* > *Network Monitor* > *Routing* to view the routing table.

Network	Gateway IP	Interfaces	Distance	IP Version	Type
IPv4 40					
0.0.0.0/0	10.100.64.254	Internet_A (port1)	1	IPv4	Static
0.0.0.0/0	10.100.65.254	Internet_B (port5)	1	IPv4	Static
10.0.2.0/24	169.254.55.153	AWS_VPG	1	IPv4	Static
10.0.10.0/24	0.0.0.0	VPN_A_Tunnel (Branch-HQ-A)	0	IPv4	Connected
10.0.10.1/32	0.0.0.0	VPN_A_Tunnel (Branch-HQ-A)	0	IPv4	Connected

   b. Run the following CLI command:

```
# get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [1/0] via 10.100.64.254, port1
                  [1/0] via 10.100.65.254, port5
S       10.0.2.0/24 [1/0] via 169.254.55.153, AWS_VPG
C       10.0.10.0/24 is directly connected, Branch-HQ-A
C       10.0.10.1/32 is directly connected, Branch-HQ-A
…
```

Zero Trust Network Access

This section includes information about ZTNA related features:

- Zero Trust Network Access introduction on page 1590
  - Basic ZTNA configuration on page 1593
  - Establish device identity and trust context with FortiClient EMS on page 1608
  - SSL certificate based authentication on page 1614
  - Full versus simple ZTNA policies on page 1616
  - Types of security posture tags on page 1623
  - ZTNA tag caching and direct tag synchronization on page 1628
  - Certificate verification for real servers on page 1629
- ZTNA advanced configurations on page 1635
- ZTNA configuration examples on page 1676
- ZTNA troubleshooting and debugging commands on page 1820
- ZTNA troubleshooting scenarios on page 1827
- ZTNA troubleshooting with error messages on page 1831

This feature is not supported on FortiGate models with 2 GB RAM or less. See Proxy-related features not supported on FortiGate 2 GB RAM models on page 126 for more information.

Zero Trust Network Access introduction

Zero Trust Network Access (ZTNA) is an access control method that uses client device identification, authentication, and security posture tags (formerly ZTNA tags) to provide role-based application access. It gives administrators the flexibility to manage network access for On-net local users and Off-net remote users. Access to applications is granted only after device verification, authenticating the user's identity, authorizing the user, and then performing context based posture checks using security posture tags.

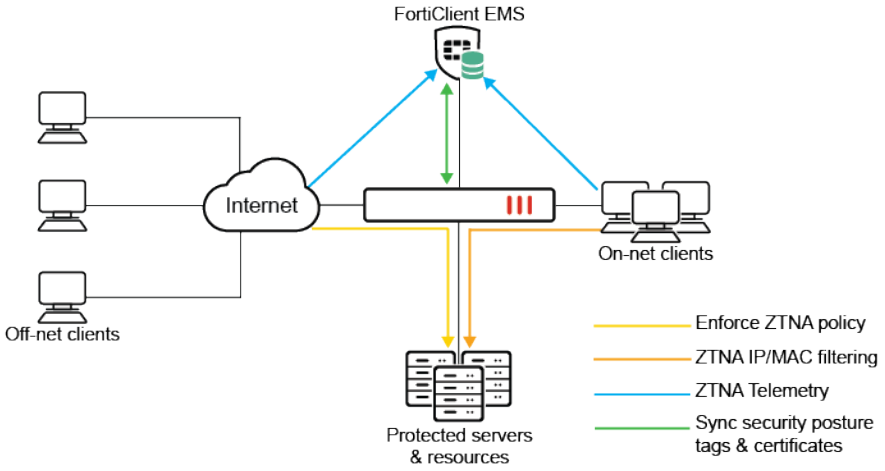
Traditionally, a user and a device have different sets of rules for on-net access and off-net VPN access to company resources. With a distributed workforce and access that spans company networks, data centers, and cloud, managing the rules can become complex. User experience is also affected when multiple VPNs are needed to get to various resources. ZTNA can improve this experience.

ZTNA application gateway and IP/MAC based access control

- ZTNA application gateway allows users to securely access resources through an SSL encrypted access proxy. This simplifies remote access by eliminating the use of VPNs.

- IP/MAC based access control combines IP/MAC with security posture tags for identification and security posture check to implement role-based zero trust access inline with regular firewall policies.

ZTNA telemetry, tags, and policy enforcement

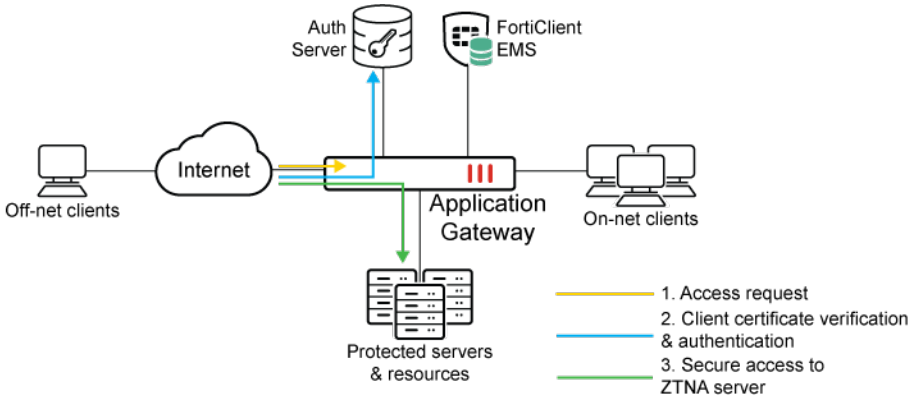


When On-net and Off-net FortiClient endpoints register to FortiClient EMS, device information, log on user information, and security posture are all shared over ZTNA telemetry with the EMS server. Clients also make a certificate signing request to obtain a client certificate from the EMS that is acting as the ZTNA Certificate Authority (CA).

Based on the client information, EMS applies matching Zero Trust tagging rules to tag the clients. These tags, and the client certificate information, are synchronized with the FortiGate in real-time. This allows the FortiGate to verify the client's identity using the client certificate, and grant access based on the security posture tags applied in the ZTNA policy.

For more information, see Establish device identity and trust context with FortiClient EMS on page 1608.

Application gateway



The FortiGate application gateway can proxy HTTP, SSH, RDP, SMB, FTP, and other TCP traffic as well as UDP traffic over secure connections with the client. This enables seamless access from the client to the protected servers, without needing to form IPsec tunnels.

ZTNA web proxy

The FortiGate ZTNA web proxy works as a reverse proxy for the HTTP server. When a client connects to a webpage hosted by the protected server, the address resolves to the FortiGate's access proxy VIP. The FortiGate proxies the connection and takes steps to authenticate the user. It prompts the user for their certificate on the browser, and verifies this against the ZTNA endpoint record that is synchronized from the EMS. If an authentication scheme, such as SAML authentication, is configured, the client is redirected to a captive portal for sign-on. If this passes, traffic is allowed based on ZTNA policies, and the FortiGate returns the webpage to the client.

For example configurations, see ZTNA web proxy example on page 1677, ZTNA web proxy with basic authentication example on page 1696, and ZTNA application gateway with SAML authentication example on page 1735.

ZTNA traffic forwarding proxy

The ZTNA traffic forwarding proxy (formerly TCP forwarding access proxy or TFAP) works as a special type of HTTPS reverse proxy. Instead of proxying traffic to a web server, TCP and UDP traffic is tunneled between the client and the access proxy over HTTPS, and forwarded to the protected resource. The FortiClient on the remote endpoint intercepts traffic destined for the protected resources and routes them to the FortiGate proxy gateway. An HTTPS connection is made to the FortiGate's access proxy VIP, where the client certificate is verified and access is granted based on the ZTNA policies. Traffic is forwarded from the FortiGate to the protected resource, and an end to end connection is established. To reduce overhead, you can disable access proxy encryption on the client, as some TCP protocols, like RDP, are already secure. The traffic forwarding proxy supports UTM scanning and deep inspection for HTTP, HTTPS, SMTP, SMTPS, IMAP, IMAPS, POP3, POP3S, SMB, and CIFS.

For an example configuration, see ZTNA traffic forwarding proxy example on page 1708.

SSH access proxy

The SSH access proxy provides some benefits to proxying SSH connections over traffic forwarding proxy, including allowing SSH deep inspection, performing optional SSH host-key validation, and allowing one-time user authentication to authenticate the ZTNA SSH access proxy connection and SSH server connection.

For an example configuration, see ZTNA SSH traffic forwarding proxy example on page 1726.

Basic ZTNA configuration components

The basic components required to configure ZTNA application gateway on the FortiGate are:

1. FortiClient EMS fabric connector and security posture tags.
2. FortiClient EMS running version 7.0.0 or later or FortiClient EMS Cloud.

3. FortiClient running 7.0.0 or later.
4. ZTNA port
5. ZTNA server (web server or traffic forwarding server)
6. ZTNA policy
7. (Optional) User authentication
8. (Optional) HA configurations

For configuration details, see Basic ZTNA configuration on page 1593.

This feature is not supported on FortiGate models with 2 GB RAM or less. See Proxy-related features not supported on FortiGate 2 GB RAM models on page 126 for more information.

ZTNA service connector

In certain environments, we may want to prevent users from accessing the FortiGate ZTNA application gateway directly or expose ports and services on the FortiGate to the public. In these scenarios, another layer of ZTNA proxy may be required to serve as the ZTNA Edge.

The FortiGate can act as a ZTNA service connector to reverse proxy service connections upstream to the ZTNA Edge. The ZTNA Edge may be a FortiPAM, FortiProxy, or even a FortiGate that acts as the ZTNA application gateway to the end users. The FortiGate, as the ZTNA service connector, forms a persistent control connection with ZTNA Edge. The ZTNA Edge forwards connection requests through the control tunnel to the FortiGate ZTNA service connector, which proxies the request to the protected server.

Basic ZTNA configuration

To deploy a ZTNA application gateway, configure the following components on the FortiGate:

1. Configure a FortiClient EMS connector on page 1594
2. Configure a ZTNA port on page 1596
3. Configure a ZTNA server on page 1597
4. Configure a ZTNA policy on page 1602
5. Optional authentication on page 1605
6. Optional HA configurations on page 1608

This feature is not supported on FortiGate models with 2 GB RAM or less. See Proxy-related features not supported on FortiGate 2 GB RAM models on page 126 for more information.

To configure ZTNA in the GUI, go to *System > Feature Visibility* and enable *Zero Trust Network Access*.

Configure a FortiClient EMS connector

To add an on-premise FortiClient EMS server in the GUI:

1. Go to *Security Fabric* > *Fabric Connectors* and double-click the *FortiClient EMS* card.
2. Set the *Status* to *Enabled* for one of the EMS servers.
3. Enter a name for the connector and the IP address or FQDN of the EMS.
4. Click *OK*.
5. A window appears to verify the EMS server certificate. Click *Accept*.
   See FortiClient EMS for more information.

To add an on-premise FortiClient EMS server in the CLI:

```
config endpoint-control fctems
    edit <ems-id>
        set status enable
        set name <name>
        set server <server IP or domain>
        set verifying-ca <certificate>
    next
end
```

To add FortiClient EMS Cloud in the GUI:

1. Go to *Security Fabric* > *Fabric Connectors* and double-click the *FortiClient EMS* card.
2. Set the *Status* to *Enabled* for one of the EMS servers.
3. Set the *Type* to *FortiClient EMS Cloud*.
4. Enter a name for the connector.
5. Click *OK*. A window appears to verify the EMS server certificate.
6. Click *Accept*.
   See FortiClient EMS for more information.

To add FortiClient EMS Cloud in the CLI:

```
config endpoint-control fctems
    edit <ems-id>
        set status enable
        set name <name>
        set fortinetone-cloud-authentication enable
        set certificate <string>
    next
end
```

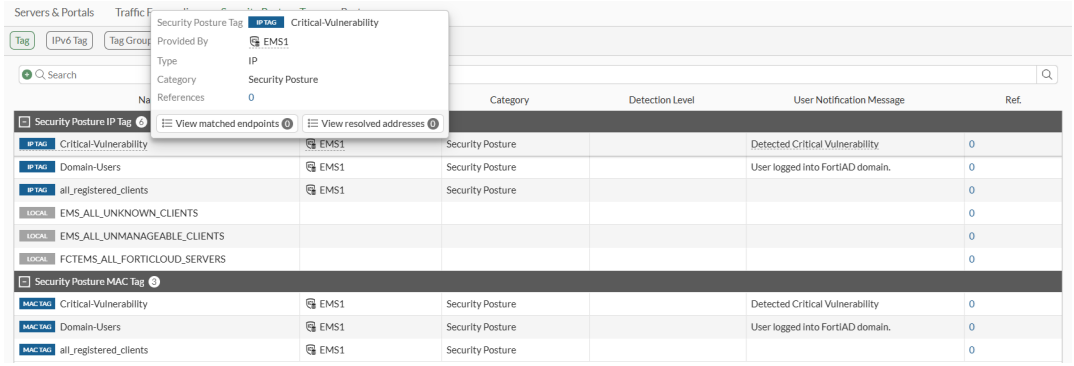
Security posture tags

After the FortiGate connects to the FortiClient EMS, it automatically synchronizes security posture tags. Security posture tags are generated from tagging rules configured on the FortiClient EMS. These tagging rules

are based on various posture checks that can be applied on the endpoints. See Endpoint Posture Check Reference.

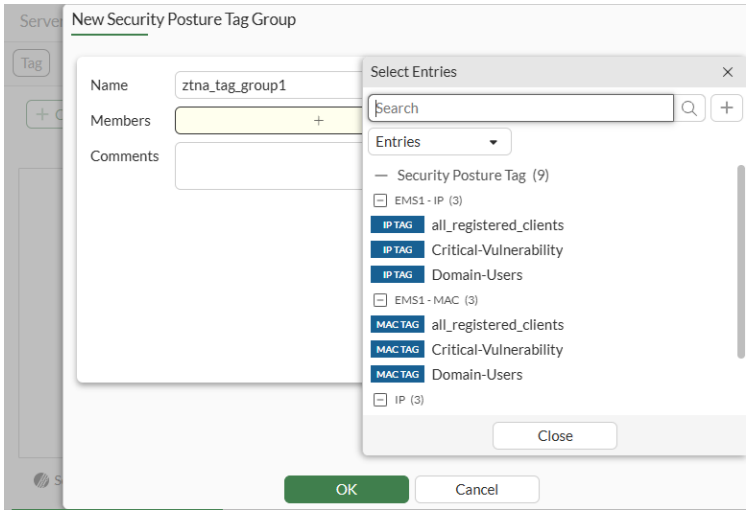
To view the synchronized security posture tags in the GUI:

1. Go to *Policy & Objects* > *ZTNA* and select the *Security Posture Tag* tab.
2. Hover the cursor over a tag name to view more information about the tag, such as its resolved addresses.



To create a security posture tag group in the GUI:

1. Go to *Policy & Objects* > *ZTNA* > *Security Posture Tags* and select the *Tag Group* sub-menu.
2. Click *Create New*.
3. Enter a name for the group and select the group members.



4. Click *OK*.

To view the synchronized security posture tags in the CLI:

```
# diagnose firewall dynamic address
# diagnose firewall dynamic list
```

To create a security posture tag group in the CLI:

```
config firewall addrgrp
    edit <group name>
        set category ztna-ems-tag
        set member <members>
    next
end
```

Configure a ZTNA port

The ZTNA port (formerly access proxy VIP) is the FortiGate ZTNA gateway that clients make HTTPS connections to. Starting in 8.0, the ZTNA port can be configured separately before being mapped to a server. This allows the ZTNA port to be decoupled from the ZTNA real-server(s) and be referenced by more than one set of servers.

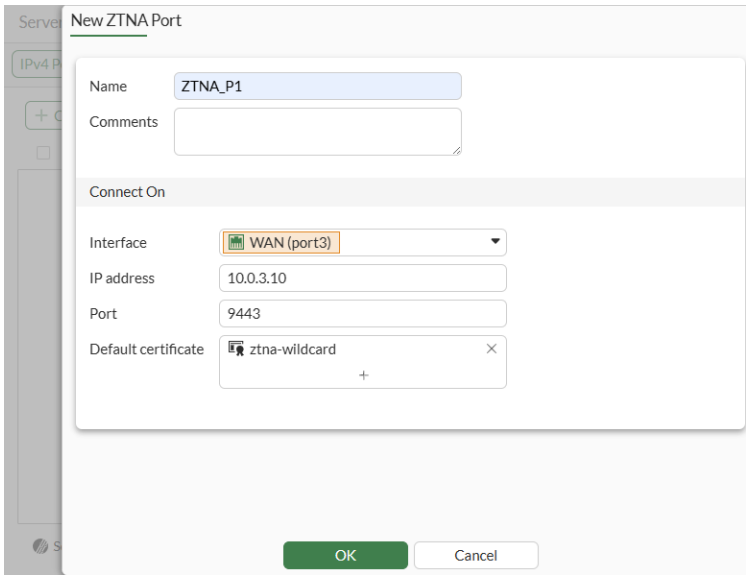
To create a ZTNA port in the GUI:

1. Go to *ZTNA* > *Ports*, and select *IPv4 Ports* or *IPv6 Ports* (visible when IPv6 is enabled in feature visibility).
2. Click *Create New*.
3. Enter a name for the ZTNA port.
4. Select an *Interface*.
5. Enter an IP address.
6. Enter the port to connect to.

   Verify that the IP address and port do not conflict with management access to the interface. Otherwise, change the IP address to another address on that subnet.

   You may specify the IP address as 0.0.0.0 if an interface other than Any is selected. The ZTNA Application Gateway IP address will dynamically use the primary and secondary IPv4 address of the interface as its external IP address.

7. Select the *Default certificate*. Clients will be presented with this certificate when they connect to the access proxy VIP.



8. Click *OK*.

To create a ZTNA port in the CLI:

```
config firewall vip
    edit <name>
        set type access-proxy
        set server-type https
        set extip <external IP>
        set extintf <external interface>
        set extport <external port>
        set empty-cert-action {accept | block | accept-unmanageable}
        set user-agent-detect {enable | disable}
        set client-cert {enable | disable}
        set ssl-certificate <certificate>
    next
end
```

Configure a ZTNA server

To configure a ZTNA server, map a ZTNA port to the real servers that clients will connect to. If a ZTNA port is not defined, you can also create a new one directly on the ZTNA server definition page.

A ZTNA server can be either a Web server or a traffic forwarding server. In the following example, we will create a ZTNA web server.

In the GUI, the web proxy server can be configured using basic or advanced configuration mode:

- **Basic mode (default)**: use this mode when you need to configure a single destination that is tied to this Web server object and virtual host.
- **Advanced mode**: use this mode when you need to configure multiple destinations tied to this Web server object and virtual host. For example, you plan to specify different paths for various destinations. You will also be able to customize the port for the server, which by default is 443.

To create a ZTNA web server for using the basic method in the GUI:

1. Go to *Policy & Objects* > *ZTNA* > *Servers & Portals*.
2. Go to *Web Server* sub-menu and click *Create New*.
3. Configure the following:

Setting	Description
Name	Name of the web server object.
IP version	Select IPv4 or IPv6. This option only appears if IPv6 is enabled on Feature Select.
Host*	Name of the virtual host that your end users are accessing.
ZTNA port	• *Use existing*: Select a previously configured ZTNA port. • *Create new*: Input settings for a new ZTNA port.
Internal server IP	• *DNS default*: Automatically resolves the virtual host and configure the real server to be the IP of the resolved host. • *Specify*: Configure the real server manually. Select Service (HTTP/HTTPS) and IP. By default, the port will be 443. To change the port, use Advanced mode.

* Host (aka Virtual Host) fields are mandatory, since multiple entries can be paired with the same ZTNA port. To differentiate between different servers, the virtual host field is used for matching.

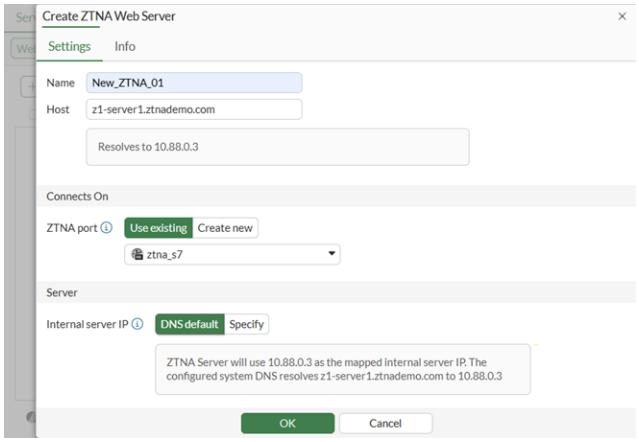
When a host is not specified, any request that resolves to the ZTNA port will be mapped to this ZTNA server. For instance, if both www.example1.com and www.example2.com resolve to the ZTNA port, then both requests are mapped to your real servers of this ZTNA server.

Note that only one ZTNA server entry can have an empty host, and it can only be configured from CLI.

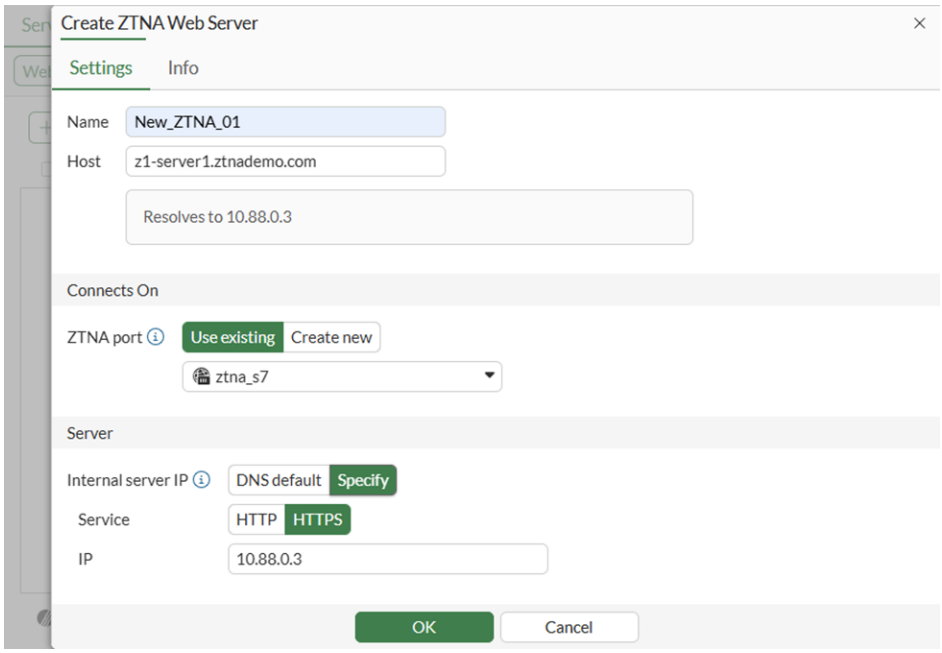
On the other hand, when you enter a name or IP address in the host field, then the request must match this host. For example, if www.example1.com is entered as the host, then only requests to www.example1.com will match this ZTNA server.

As a best practice, an external DNS should be used by remote clients to resolve the virtual-host to the ZTNA port.

Example using DNS default for Internal server IP



Manually specify Internal server IP



4. Click *OK* to save.

This method automatically configures a virtual host object in the CLI and references it in the ZTNA web-proxy object.

```
config firewall access-proxy-virtual-host
    edit "web-proxy_vh_mggshssu"
        set ssl-certificate "ztna-wildcard"
        set host "z1-server1.ztnademo.com"
    next
end
config ztna web-proxy
    edit "New_ZTNA_01"
        set vip "ztna_s7"
        set host "web-proxy_vh_mggshssu"
        config api-gateway
            edit 1
                config realservers
                    edit 1
                        set ip 10.88.0.3
                    next
                end
            next
        end
    next
end
```

For other advanced virtual host settings, use the CLI:

```
config firewall access-proxy-virtual-host
    edit <host>
        set host-type { substring | fqdn* | wildcard }
        set empty-cert-action { accept | block* | accept-unmanageable }
```

```
        set client-cert { enable* | disable }
    next
end
```

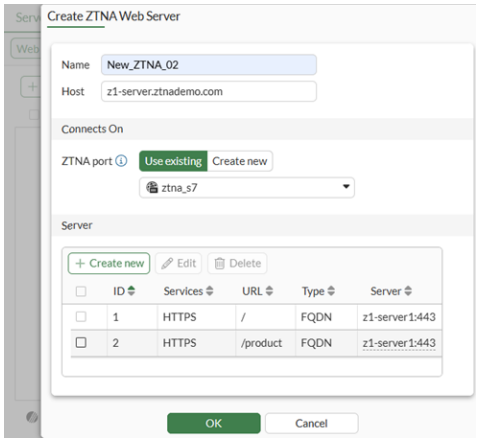
When using host-type fqdn, the host will only match when the subdomain matches the host. For example, the host test.com will match vhost1.test.com, but will not match test.com.vhost. On the other hand, host-type substring will match both.

To create a ZTNA web server using the advanced method in the GUI:

1. Go to *Policy & Objects* > *ZTNA* > *Servers & Portals*.
2. Go to the *Web Server* sub-menu and click *Create New*.
3. Enable *Advanced configuration mode*.
4. Enter the *Name*, *IP version*, *Host*, and *ZTNA port* settings as in the Basic method.
5. In the *Server* table, click *Create new* to add a new real server.

Setting	Description
Service	HTTP or HTTPS
Path*	Optional. Specify a specific path for matching. When creating more than one real server, the paths must be distinct.
Address Type	• *IP*: Input the IP and port of the real server. • *FQDN*: The address of the real server can be chosen from pre-configured FQDN addresses.

*The path can be matched by substring, wildcard, or regular expression when configured from the CLI. In the GUI, the default is substring. For example, if the virtual host is specified as www.example1.com, and the path substring is map1, then www.example1.com/map1 will be matched to this real server.



6. Click *OK* to save.

This method also automatically configures a virtual host object in the CLI and references it in the ZTNA web-proxy object:

```
config firewall access-proxy-virtual-host
        edit "web-proxy_vh_mggsn6e5"
        set ssl-certificate "ztna-wildcard"
        set host "z1-server.ztnademo.com"
    next
end
config ztna web-proxy
    edit "New_ZTNA_02"
        set vip "ztna_s7"
        set host "web-proxy_vh_mggsn6e5"
        config api-gateway
            edit 1
                config realservers
                    edit 1
                        set addr-type fqdn
                        set address "z1-server1"
                    next
                end
            next
            edit 2
                set url-map "/product"
                config realservers
                    edit 1
                        set addr-type fqdn
                        set address "z1-server1"
                    next
                end
            next
        end
    next
end
```

Starting in FortiOS 7.6.4, certificate verification of the real server is enabled by default. See Certificate verification for real servers on page 1629 for more details. You can manually adjust the setting using the following CLI command:

```
config ztna web-proxy
    edit <server>
        config api-gateway
            edit 1
                config realservers
                    edit 0
                        set ip <real server 2 address>
                        set port <real server 2 port>
                        set verify-cert {enable | disable}
                    next
                end
            next
        end
    next
end
```

Configure a ZTNA policy

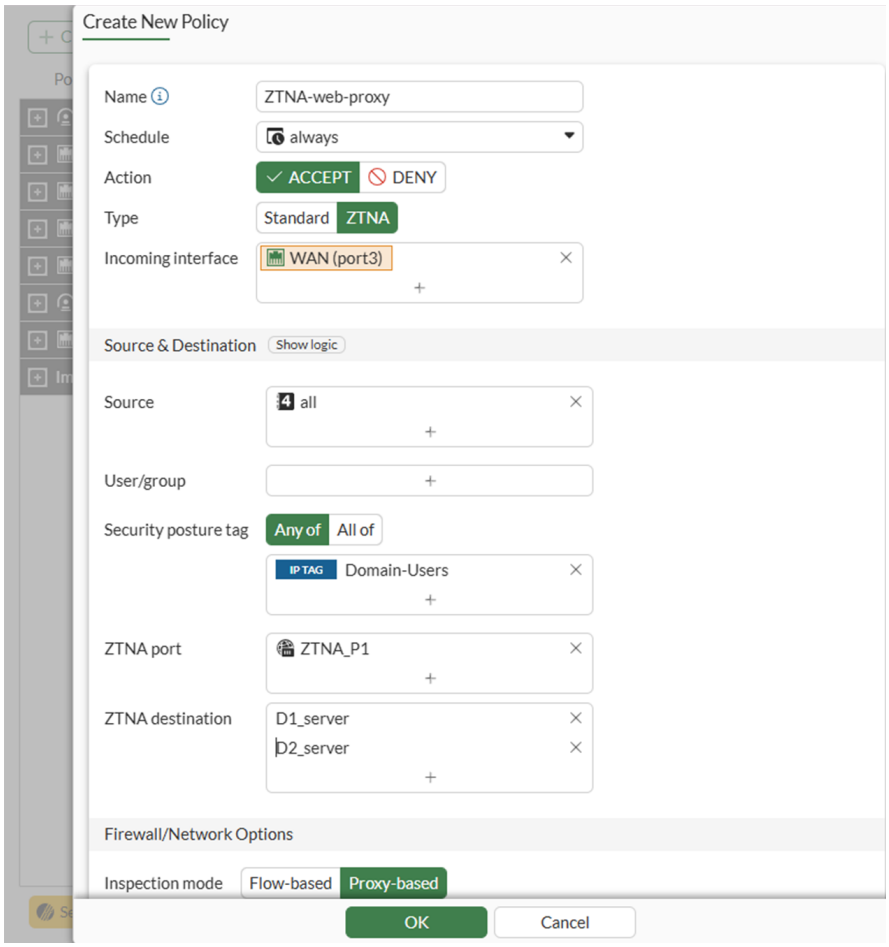
A ZTNA policy is used to enforce zero trust role based access control by using security posture tags or tag groups to verify a device's security posture. A ZTNA policy can also utilize security profiles to protect this traffic.

There are two ways to configure ZTNA rules in the GUI by using a full or simple ZTNA policy:

- Full ZTNA policy: The method uses ZTNA type proxy policies. You can granularly define destination addresses beyond the ZTNA server and destination you are allowing.
- Simple ZTNA policy: A simplified method for configuring a ZTNA policy using firewall policies. This method covers most functionality of a Full ZTNA policy, except it cannot control access based on destination interface or real server's destination address.

To configure a simple ZTNA policy in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Click *Create New*.
3. Enter a name for the ZTNA policy.
4. Select the *Schedule*.
5. Select the *Action*.
6. Set *Type* to *ZTNA*.
7. Select an *Incoming Interface*.
8. Set the *Source*.
9. Add the security posture tags or tag groups that are allowed access. If multiple tags are included, the matching method is set to *Any* by default.
10. Select the *ZTNA port*. By selecting the ZTNA port, all ZTNA servers associated with it become accessible.



11. 10.11. If traffic forwarding destinations are configured, select the destinations.
12. Configure the remaining options as needed.
13. Click OK.

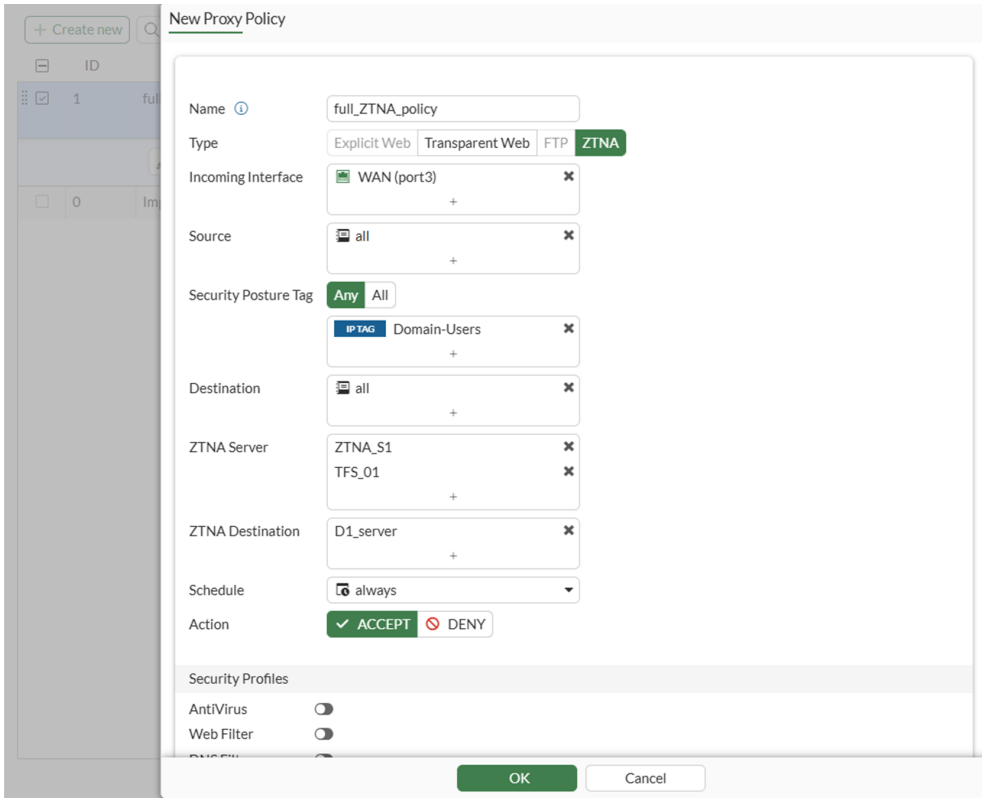
To configure a simple ZTNA policy in the CLI:

```
cconfig firewall policy
    edit 1
        set name <ZTNA policy name>
        set srcintf <interface>
        set dstintf "any"
        set action accept
        set srcaddr "all"
        set dstaddr "ZTNA port"
        set ztna-ems-tag <security posture tags>
               set ztna-destination <destination 1> <destination 2> …
        set schedule "always"
        set nat enable
    next
end
```

The `dstintf` command cannot be modified.

To configure a full ZTNA policy in the GUI:

1. Go to *System* > *Feature Visibility*.
2. Under *Security Features*, enable *Explicit Proxy*.
3. Click *Apply*.
4. Go to *Policy & Objects* > *Proxy Policy*.
5. Click *Create New*.
6. Enter a name for the rule.
7. Change *Type* to *ZTNA*.
8. Select an *Incoming Interface* and *Source*.
9. Add the security posture tags or tag groups that are allowed access. If multiple tags are included, the matching method is set to *Any* by default.
10. Select the *Destination*.
11. Select the *ZTNA Server*.
12. If a *Traffic Forwarding Server* is selected, then select a *ZTNA Destination* as well.
13. Select the *Action*.



14. Configure the remaining options as needed.
15. Click *OK*.

To configure a full ZTNA policy in the CLI:

```
config firewall proxy-policy
    edit 1
        set name <ZTNA policy name>
        set proxy ztna-proxy
        set ztna-proxy <proxy server 1> <proxy server 2> …
        set ztna-destination <destination 1> <destination 2> …
        set srcintf <interface>
        set srcaddr "all"
        set dstaddr "all"
        set ztna-ems-tag <security posture tag(s)>
        set action accept
        set schedule "always"
        set logtraffic all
    next
end
```

Optional authentication

To configure authentication to the ZTNA application gateway, you must configure an authentication scheme and authentication rule in the GUI or CLI. They are used to authenticate proxy-based policies, similar to configuring authentication for explicit and transparent proxy.

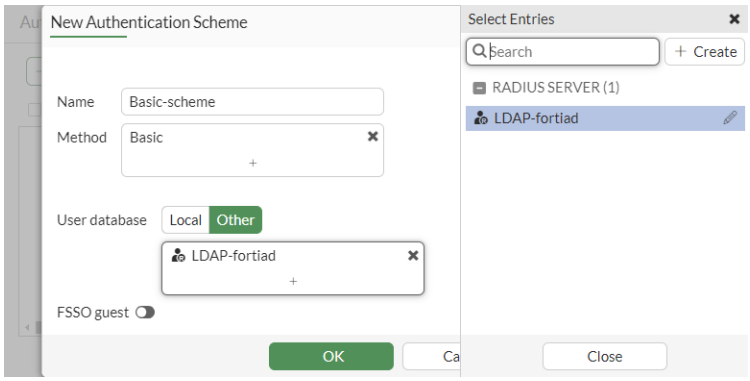
The authentication scheme defines the method of authentication that is applied. For ZTNA, basic HTTP and SAML methods are supported. Each method has additional settings to define the data source to check against. For example, with basic HTTP authentication, a user database can reference an LDAP server, RADIUS server, local database, or other supported authentication servers that the user is authenticated against.

The authentication rule defines the proxy sources and destinations that require authentication, and which authentication scheme to apply. For ZTNA, active authentication method is supported. The active authentication method references a scheme where users are actively prompted for authentication, like with basic authentication.

After the authentication rule triggers the method to authenticate the user, a successful authentication returns the groups that the user belongs to. In the ZTNA policy you can define a user or user group as the allowed source. Only users that match that user or group are allowed through the proxy policy.

To configure a basic authentication scheme in the GUI:

1. Go to *Policy & Objects* > *Authentication*.
2. Go to the *Authentication Schemes* tab.
3. Click *Create New*.
4. Enter a name for the scheme.
5. Set the *Method* as *Basic*.
6. Select the *User database* as required.



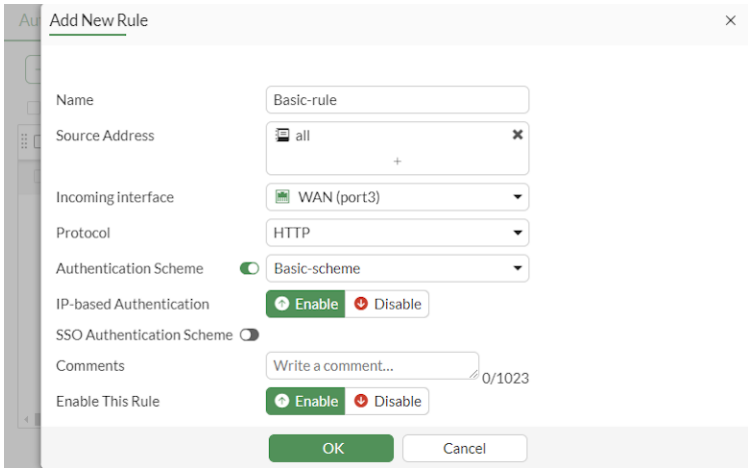
7. Click *OK*.

To configure a basic authentication scheme in the CLI:

```
config authentication scheme
    edit <name>
        set method basic
        set user-database <auth server>
    next
end
```

To configure an authentication rule in the GUI:

1. Go to *Policy & Objects* > *Authentication*.
2. Click *Create New*.
3. Enter a name for the rule.
4. Select the *Source Address*.
5. Select the *Incoming interface*.
6. Select the *Authentication Scheme*.



7. Click *OK*.

To configure an authentication rule in the CLI:

```
config authentication rule
    edit <name>
        set status enable
        set protocol http
        set srcintf <interface>
        set srcaddr <address>
        set dstaddr <address>
        set ip-based enable
        set active-auth-method <active auth scheme>
    next
end
```

To apply a user group to a full ZTNA policy in the GUI:

1. Go to *Policy & Objects* > *Proxy Policy*.
2. Edit an existing rule, or click *Create New* to create a new rule.
3. Click in the *Source* field, select the *User* tab, and select the users and user groups that will be allowed access.
4. Configure the remaining settings as required.
5. Click *OK*.

To apply a user group to a full ZTNA policy in the CLI:

```
config firewall proxy-policy
    edit 1
        set name <ZTNA policy name>
        set proxy ztna-proxy
        set ztna-proxy <proxy server 1> <proxy server 2> …
        set ztna-destination <destination 1> <destination 2> …
        set srcintf <interface>
        set srcaddr "all"
        set dstaddr "all"
        set ztna-ems-tag <security posture tag(s)>
        set action accept
        set schedule "always"
        set logtraffic all
        set groups <user group(s)>
    next
end
```

The authentication rule and scheme defines the method used to authenticate users. With basic HTTP authentication, a sign in prompt is shown after the client certificate prompt. After the authentication passes, the returned groups that the user is a member of are checked against the user groups that are defined in the ZTNA policy. If a group matches, then the user is allowed access after passing a posture check.

For basic setup information, see ZTNA web proxy with basic authentication example on page 1696.

For advanced setup information, see ZTNA application gateway with SAML authentication example on page 1735 and ZTNA application gateway with SAML and MFA using FortiAuthenticator example on page 1750.

Optional HA configurations

User information and TLS sessions are synchronized between HA members for ZTNA proxy sessions. When a failover occurs, the new primary unit will continue allowing sessions from the logged in users without asking for the client certificate and re-authentication again.

There are no special configurations for HA. Refer to HA active-passive cluster setup on page 4079 and HA active-active cluster setup on page 4081 to configure your HA cluster.

ZTNA web proxy vs traffic forwarding access proxy

In a ZTNA web proxy connection, there is no configurations needed on the client endpoint. Users can simply access the HTTP website on a browser by entering its URL. For traffic forwarding proxy (TFP), a ZTNA rule must be configured on the FortiClient endpoint. This rule instructs FortiClient to listen to traffic to the destination address and port and redirects the traffic to the FortiGate ZTNA application gateway over HTTPS.

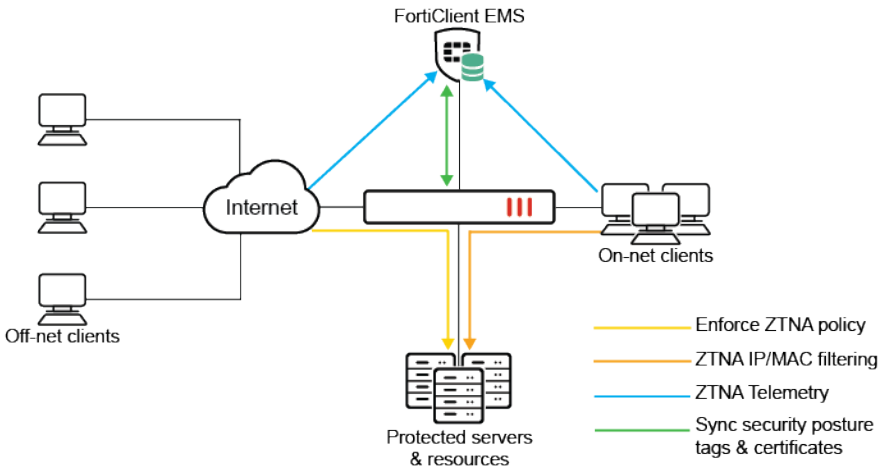
When deciding between using ZTNA web proxy or TFP for accessing web applications, consider the following.

- Use ZTNA web proxy when the protected web application address can be resolved by the remote users publicly.
- Use TFP when the protected application address can only be resolved on the internal network. TCP forwarding rules allow the FortiClient to intercept the request to the destination address and forward them to the application gateway.

For more information, see ZTNA traffic forwarding proxy example on page 1708.

Establish device identity and trust context with FortiClient EMS

How device identity is established through client certificates, and how device trust context is established between FortiClient, FortiClient EMS, and the FortiGate, are integral to ZTNA.



Device roles

FortiClient

FortiClient endpoints provide the following information to FortiClient EMS when they register to the EMS:

- Device information (network details, operating system, model, and others)
- Logged on user information
- Security posture (On-net/Off-net, antivirus software, vulnerability status, and others)

It also requests and obtains a client device certificate from the EMS ZTNA Certificate Authority (CA) when it registers to FortiClient EMS. The client uses this certificate to identify itself to the FortiGate.

FortiClient EMS

FortiClient EMS issues and signs the client certificate with the FortiClient UID, certificate serial number, and EMS serial number. The certificate is then synchronized to the FortiGate. EMS also shares its EMS ZTNA CA certificate with the FortiGate, so that the FortiGate can use it to authenticate the clients.

FortiClient EMS uses zero trust tagging rules to tag endpoints based on the information that it has on each endpoint. The tags are also shared with the FortiGate. See Endpoint Posture Check Reference for a list of the endpoint posture checks that EMS can perform.

Each security posture tag creates two firewall addresses in all VDOMs on a FortiGate. One firewall address is the IP address, and the other firewall address is the MAC address. Because each FortiGate model has a global limit and a per-VDOM limit for the maximum number of supported firewall addresses, the FortiGate model determines the maximum number of security posture tags allowable by that unit, which is the maximum number of firewall address divided by two. For each FortiGate model's limit, see the Maximum Values table.

FortiGate

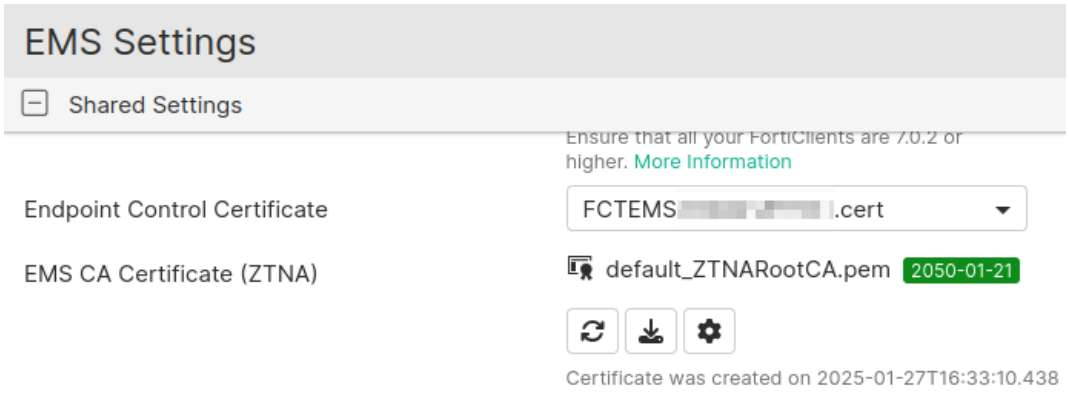
The FortiGate maintains a continuous connection to the EMS server to synchronize endpoint device information, including primarily:

- FortiClient UID
- Client certificate SN
- EMS SN
- Device credentials (user/domain)
- Network details (IP and MAC address and routing to the FortiGate)

When a device's information changes, such as when a client moves from on-net to off-net, or their security posture changes, EMS is updated with the new device information and then updates the FortiGate. The FortiGate's WAD daemon can use this information when processing ZTNA traffic. If an endpoint's security posture change causes it to no longer match the ZTNA policy criteria on an existing session, then the session is terminated.

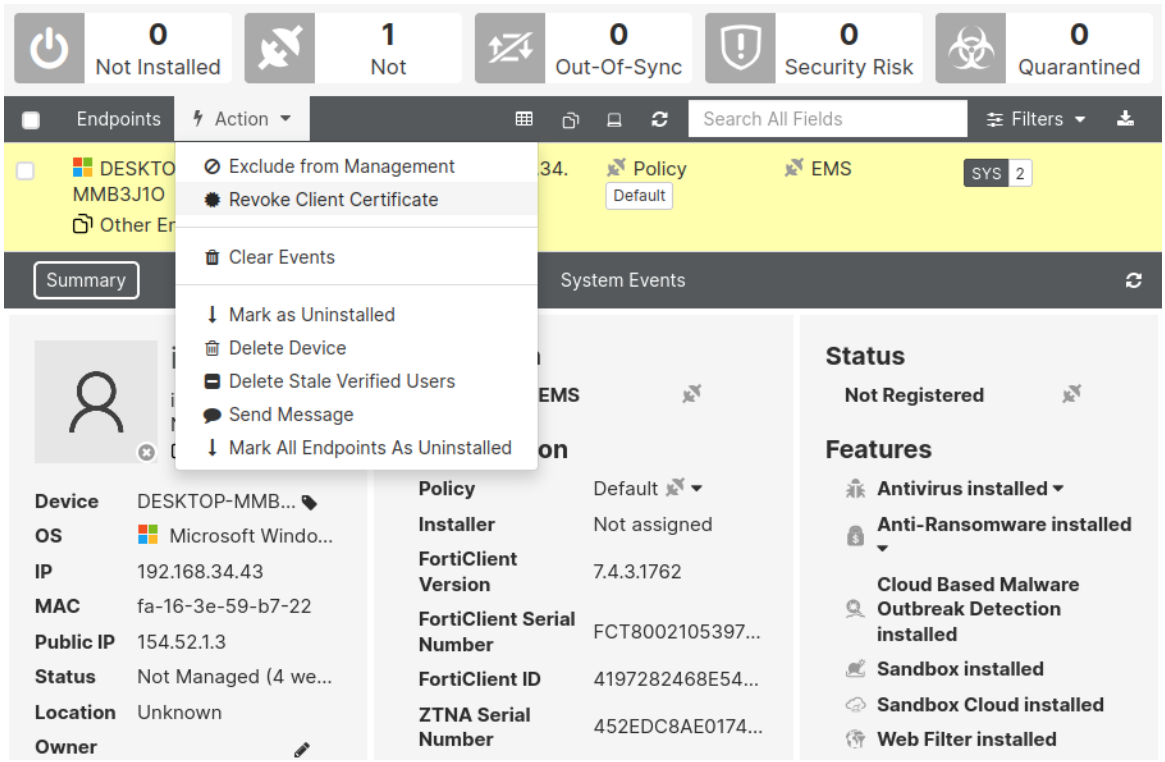
Certificate management on FortiClient EMS

FortiClient EMS has a *default_ZTNARootCA* certificate generated by default that the ZTNA CA uses to sign CSRs from the FortiClient endpoints. Clicking the refresh button revokes and updates the root CA, forcing updates to the FortiGate and FortiClient endpoints by generating new certificates for each client. Starting with FortiClient EMS 7.4.3, a custom root certificate and key can be uploaded to replace the default certificate.



Do not confuse the EMS CA certificate (ZTNA) with the SSL certificate. The latter is the server certificate that is used by EMS for HTTPS access and fabric connectivity to the EMS server.

EMS can also manage individual client certificates. To revoke the current client certificate that is used by the endpoint: go to *Endpoint > All Endpoints*, select the client, and click *Action > Revoke Client Certificate*.



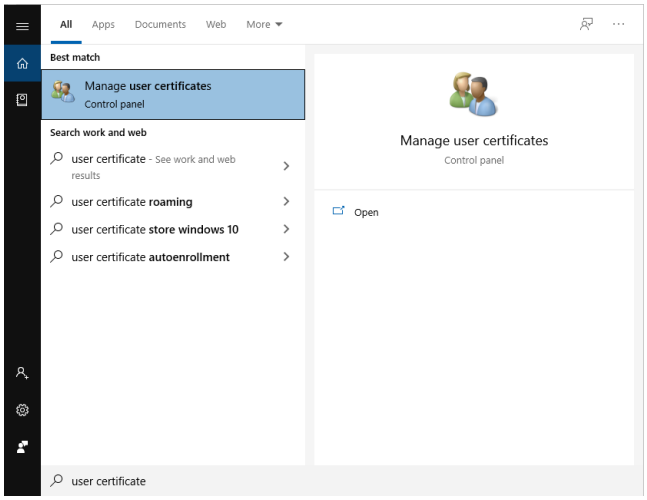
Locating and viewing the client certificate on an endpoint

In Windows, FortiClient automatically installs certificates into the certificate store. The certificate information in the store, such as certificate UID and SN, should match the information on EMS and the FortiGate.

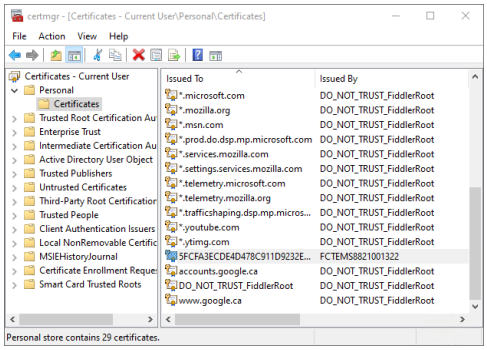
To locate certificates on other operating systems, consult the vendor documentation.

To locate the client certificate and EMS ZTNA CA certificate on a Windows PC:

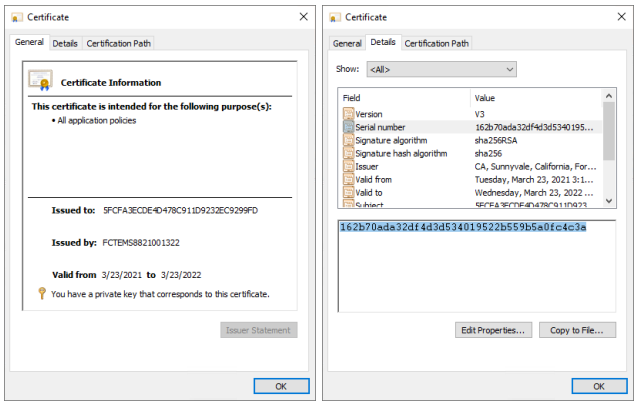
1. In the Windows search box, enter *user certificate* and click *Manage user certificates* from the results.



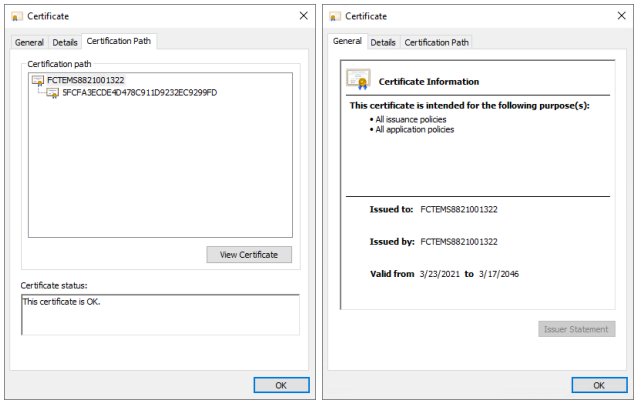
2. In the certificate manager, go to *Certificates - Current User* > *Personal* > *Certificates* and find the certificate that is issued by the FortiClient EMS.



3. Right-click on it and select Properties.
4. The *General* tab shows the client certificate UID and the issue and expiry dates. The *Details* tab show the certificate SN.



5. Go to the *Certificate Path* tab to see the full certificate chain.
6. Select the root CA and click *View Certificate* to view the details about the EMS ZTNA CA certificate.



Verifying that the client information is synchronized to the FortiGate

The following diagnose commands help to verify the presence of matching endpoint record, and information such as the client UID, client certificate SN, and EMS certificate SN on the FortiGate. If any of the information is missing or incomplete, client certificate authentication might fail because the corresponding endpoint entry is not found. More in-depth diagnosis would be needed to determine the reason for the missing records.

Command	Description
`# diagnose endpoint ec-shm list <ip> <mac> <EMS_serial_number> <EMS_tenant_id>`	Show the endpoint record list. Optionally, add filters.
`# diagnose wad dev query-by uid <uid> <ems sn> <ems tenant id>`	Query from WAD diagnose command by UID.
`# diagnose wad dev query-by ipv4 <ip>`	Query from WAD diagnose command by IP address.
`# diagnose test application fcnacd 7`	Check the FortiClient NAC daemon ZTNA and route cache.

Command	Description
`# diagnose test application fcnacd 8`	
`# diagnose test application fcnacd 5`	Force a sync with the FortiClient EMS server.

To check the endpoint record list for IP address 10.0.3.2:

```
# diagnose endpoint ec-shm list
Record #0:
                IP Address = 10.0.3.2
                MAC Address = 02:09:0f:00:03:03
                MAC list =
                VDOM =  (-1)
                EMS serial number: FCTEMS8825007166
                EMS tenant id: 00000000000000000000000000000000
                Client cert SN: 5341B8C01A11C7AB130A07718F1CBD20947E7253
                Public IP address: 148.230.50.179
                Quarantined: no
                Online status: online
                Registration status: registered
                On-net status: on-net
                Gateway Interface:
                FortiClient version: 7.4.3
                AVDB version: 1.0
                FortiClient app signature version: 35.152
                FortiClient vulnerability scan engine version: 3.8
                FortiClient UID: 9A016B5A6E914B42AD4168C066EB04CA
                Host Name: WIN10-01
                OS Type: WIN64
                OS Version: Microsoft Windows 10 Professional Edition, 64-bit (build 19045)
                Domain: fortiad.info
                Last Login User: tsmith
                …
                Number of Routes: (0)
```

To check the tags that are processed by the WAD daemon for a particular device:

```
# diagnose wad dev query-by uid 9A016B5A6E914B42AD4168C066EB04CA FCTEMS8825007166
00000000000000000000000000000
00000
Attr of type=0, length=83, value(ascii)=9A016B5A6E914B42AD4168C066EB04CA
Attr of type=7, length=8, value(ascii)=yii
Attr of type=4, length=13, value(ascii)=@fortiad.info
Attr of type=6, length=1, value(ascii)=true
Attr of type=9, length=1, value(ascii)=1
Attr of type=5, length=40, value(ascii)=5341B8C01A11C7AB130A07718F1CBD20947E7253
Attr of type=3, length=76, value(ascii)=ZTNA_all_registered_clients_
FCTEMS882500716600000000000000000000000000000000
Attr of type=3, length=66, value(ascii)=ZTNA_Domain-Users_
```

```
FCTEMS882500716600000000000000000000000000000000
Response termination due to no more data
```

To check the FortiClient NAC daemon cache:

```
# diagnose test application fcnacd 7
#ID:0
UID:     9A016B5A6E914B42AD4168C066EB04CA
EMS:     FCTEMS8825007166 00000000000000000000000000000000
State:   sysinfo:1, tag:1, tagsz:1, out-of-sync:0, ref:1
Owner:   @fortiad.info
Cert SN: 5341B8C01A11C7AB130A07718F1CBD20947E7253
online:  Yes
Sub TS:  1768534647
idx:0, ttdl:1   name:all_registered_clients
idx:1, ttdl:1   name:Domain-Users
```

SSL certificate based authentication

A client certificate is obtained when an endpoint registers to EMS. FortiClient automatically submits a CSR request and the FortiClient EMS signs and returns the client certificate. This certificate is stored in the operating system's certificate store for subsequent connections. The endpoint information is synchronized between the FortiGate and FortiClient EMS. When an endpoint disconnects or is unregistered from EMS, its certificate is removed from the certificate store and revoked on EMS. The endpoint obtains a certificate again when it reconnects to EMS.

By default, client certificate authentication is enabled on the access proxy, so when the HTTPS request is received the FortiGate's WAD process challenges the client to identify itself with its certificate. The FortiGate makes a decision based on the following possibilities:

1. If the client responds with the correct certificate that the client UID and certificate SN can be extracted from:
   - If the client UID and certificate SN match the record on the FortiGate, the client is allowed to continue with the ZTNA proxy rule processing.
   - If the client UID and certificate SN do not match the record on the FortiGate, the client is blocked from further ZTNA proxy rule processing.
2. If the client cancels and responds with an empty client certificate:
   - If `empty-cert-action` is set to `accept`, the client is allowed to continue with ZTNA proxy rule processing.
   - If `empty-cert-action` is set to `block`, the client is blocked from further ZTNA proxy rule processing.

To configure the client certificate actions:

```
config firewall vip
    edit <name>
        set client-cert {enable | disable}
        set empty-cert-action {accept | block | accept-unmanageable}
```

```
    next
end
```

To override these settings on a virtual host:

```
config firewall access-proy-virtual-host
    edit <host>
        set empty-cert-action {accept | block | accept-unmanageable}
        set client-cert {enable | disable}
    next
end
```

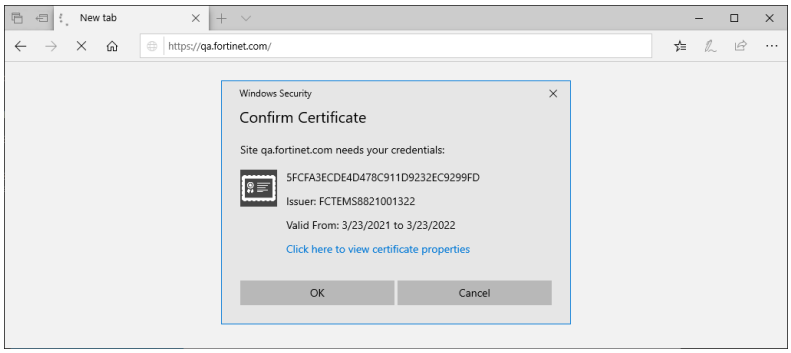
Example

In this example, a client connects to *qa.fortinet.com* and is prompted for a client certificate.

- `client-cert` is set to `enable`, and `empty-cert-action` is set to `block`.
- The ZTNA server is configured, and a ZTNA policy is set to allow this client.
- The domain resolves to the FortiGate access proxy VIP.

Scenario 1:

When prompted for the client certificate, the client clicks *OK* and provides a valid certificate that is verified by the FortiGate.

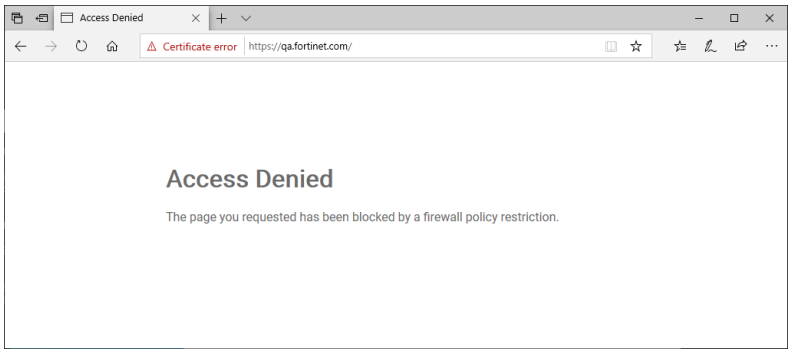


Result:

The client passes SSL certificate authentication and is allowed to access the website.

Scenario 2:

When prompted for the client certificate, the client clicks *Cancel*, resulting in an empty certificate response to the access proxy.



Result:

Because the certificate response is empty and `empty-cert-action` is set to `block`, the WAD daemon blocks the connection.

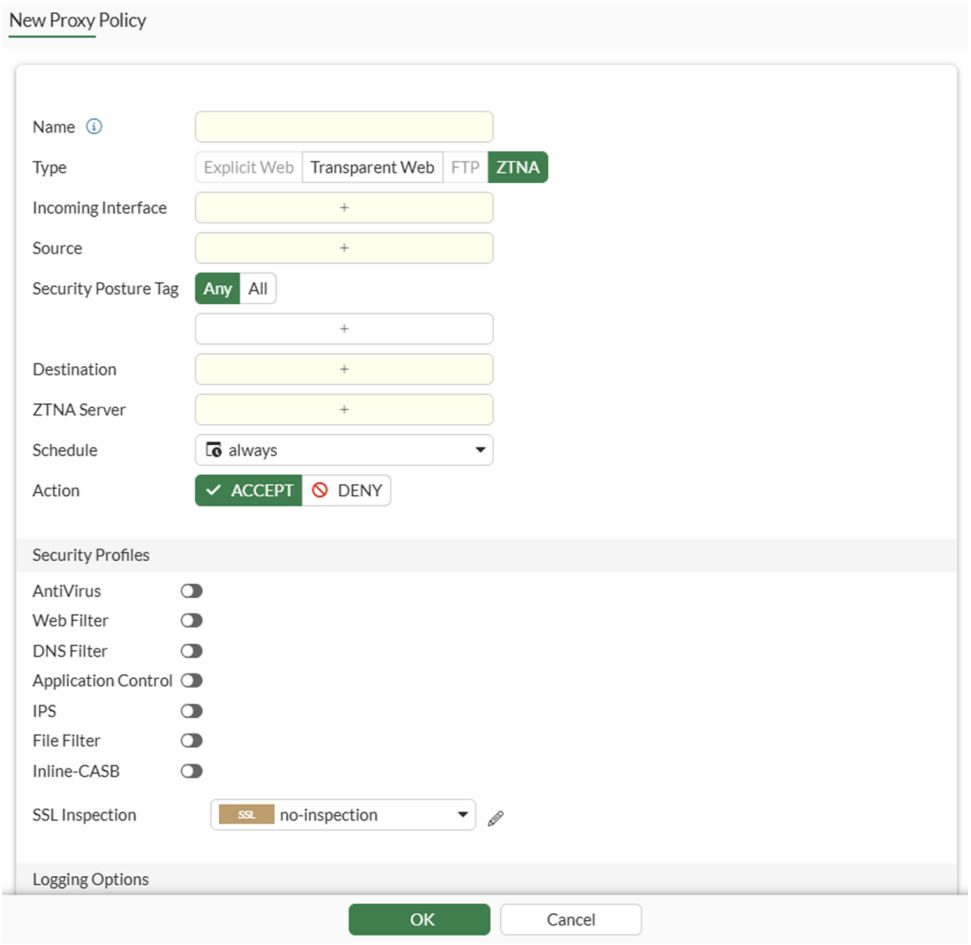
Currently, the Microsoft Edge, Google Chrome, and Safari browsers are supported by ZTNA.

Full versus simple ZTNA policies

There are two ways to configure ZTNA rules in the GUI by using a full or simple ZTNA policy.

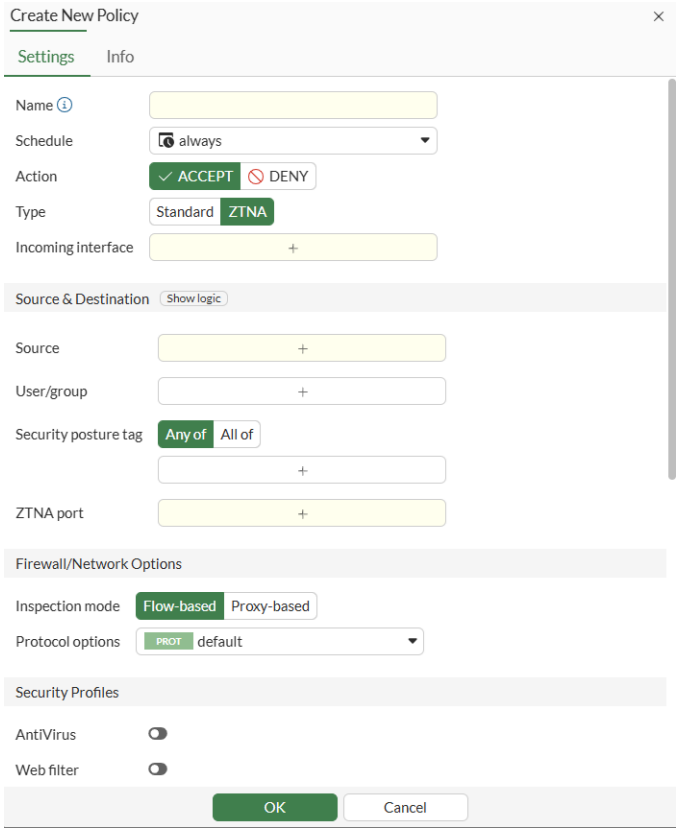
Full ZTNA policy

Administrators can configure ZTNA policies from the *Policy & Objects* > *Proxy Policy* page, and by setting the *Type* to *ZTNA*.



Simple ZTNA policy

In a simple ZTNA policy, a regular firewall policy is used for policy management. When creating a new firewall policy, administrators can configure a ZTNA policy by setting the *Type* to *ZTNA*.



A simple ZTNA policy cannot control access based on the destination interface or the real server's destination address. You can select a ZTNA TFS destination though. See the Examples on page 1619 section for detailed configurations.

Authentication for ZTNA policies

Authentication remains largely the same between both ZTNA policy configuration modes. You can specify user groups under *Source* to define the groups in which the access control applies to. However, the underlying authentication schemes and rules must still be in place to direct the traffic to the ZTNA application gateway.

Authentication for regular firewall policies

Authentication for regular firewall policies is traditionally handled by authd, which does not require an authentication scheme and rules to be configured in order to function. On a ZTNA policy, authentication is handled by WAD so that the authentication scheme and rules are used to determine the type of authentication and the traffic that requires authentication.

Redirecting a simple ZTNA policy to a full ZTNA policy

An option is available so that after matching a simple ZTNA policy, the traffic can be redirected for a full ZTNA policy match. This setting can only be configured from the CLI, and it is disabled by default.

```
config firewall policy
    edit <id>
        set ztna-policy-redirect {enable | disable}
    next
end
```

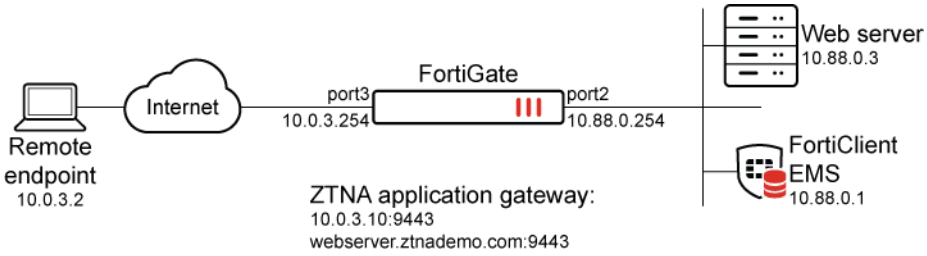
For example, a client has both tag A and tag B. In the simple ZTNA policy, the client matches a policy that requires tag A for a posture check. If they are using the `ztna-policy-redirect` option, then it will also require a full ZTNA policy match.

If a full ZTNA policy allows either tag A or tag B or all traffic in general, then the traffic is allowed. Otherwise, if a full ZTNA policy explicitly denies one of the tags, the traffic will be denied.

If no full ZTNA policy is matched, then the traffic is implicitly denied.

Examples

The following examples demonstrate how to configure a ZTNA policy using the full and simple ZTNA policy modes.



It is assumed that the following settings are already configured:

- EMS connection and EMS tags (Malicious-File-Detected and FortiAD.Info)
- ZTNA server configuration (ZTNA-webserver)
- Authentication scheme and rule

Configuring a full ZTNA policy

To configure a full ZTNA policy in the GUI:

1. Go to *Policy & Objects* > *Proxy Policy* and click *Create New*.
2. Configure the following settings:

Name	*ZTNA-webserver*
Type	*ZTNA*

Incoming Interface	*port3*
Source	*all*
Destinations	*Webserver1 (10.88.0.3/32)*
ZTNA Server	*ZTNA-webserver*
Schedule	*always*
Action	*ACCEPT*

3. Click *OK*.

To configure a full ZTNA policy in the CLI:

```
config firewall proxy-policy
    edit 1
        set name "ZTNA-webserver"
        set proxy ztna-proxy
        set ztna-proxy "ZTNA_S1"
        set srcintf "port3"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
    next
end
```

When traffic is allowed, the ZTNA logs show traffic passing through policy 1 on a policy called `ZTNA-webserver`, which is a proxy policy.

To verify the traffic logs:

```
# execute log filter category traffic
# execute log filter field subtype ztna
# execute log display
3: date=2026-02-11 time=17:33:27 eventtime=1770860006215822330 tz="-0800" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=28239
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.88.0.3
dstport=9043 dstintf="port2" dstintfrole="dmz" sessionid=9954 srcuuid="b458a65a-f759-51ea-d7df-
ef2e750026d1" dstuuid="b458a65a-f759-51ea-d7df-ef2e750026d1" service="tcp/9043"
proxyapptype="ztna-proxy" proto=6 action="accept" policyid=1 policytype="proxy-policy"
poluuid="bd19e37c-07b0-51f1-8a80-f206d714dad9" policyname="ZTNA-webserver" appcat="unscanned"
duration=126 gatewayid=1 realserverid=1 vip="ZTNA_P1" vipincomingip=10.0.3.10 accessproxy="ZTNA_
S1" clientdevicemanageable="manageable" clientcert="yes" wanin=302884 rcvdbyte=302884 wanout=4424
lanin=4155 sentbyte=4155 lanout=305667
```

Configuring a simple ZTNA policy

To configure a simple ZTNA policy in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
2. Configure the following settings:

Name	*ZTNA-webserver-fp*
Schedule	*always*
Action	*ACCEPT*
Type	*ZTNA*
Incoming Interface	*port3*
Source	*all*
ZTNA server	*ZTNA-P1*

3. Click *OK*.

To configure a simple ZTNA policy in the CLI:

```
config firewall policy
    edit 17
        set name "ZTNA-webserver-fp"
        set srcintf "port3"
        set dstintf "any"
        set action accept
        set srcaddr "all"
        set dstaddr "ZTNA_P1"
        set schedule "always"
        set logtraffic all
        set nat enable
    next
end
```

When traffic is allowed, the ZTNA logs show traffic passing through policy 9 on a policy called `ZTNA-webserver-fp`, which is a firewall policy.

To verify the traffic logs:

```
# execute log filter category traffic
# execute log filter field subtype ztna
# execute log display
14 logs found.
10 logs returned.
1: date=2026-02-11 time=17:47:54 eventtime=1770860873837986548 tz="-0800" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=28375
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.88.0.3
dstport=9043 dstintf="port2" dstintfrole="dmz" sessionid=10345 srcuuid="b458a65a-f759-51ea-d7df-
```

```
ef2e750026d1" service="tcp/9043" proxyapptype="ztna-proxy" proto=6 action="accept" policyid=17
policytype="policy" poluuid="8d902222-ecf0-51f0-8a9d-8b226df29093" policyname="ZTNA-webserver-fp"
appcat="unscanned" duration=128 gatewayid=1 realserverid=1 vip="ZTNA_P1" vipincomingip=10.0.3.10
accessproxy="ZTNA_S1" clientdevicemanageable="manageable" clientcert="yes" wanin=303036
rcvdbyte=303036 wanout=4424 lanin=4187 sentbyte=4187 lanout=305778
```

Configuring a ZTNA simple policy with security posture tags and authentication

In this example, a simple ZTNA policy uses the FortiAD.Info tag for a posture check and authentication against a pre-configured Active Directory server where the user tsmith resides. The authentication scheme and rule have already been configured as follows:

```
config authentication scheme
    edit "ZTNA-Auth-scheme"
        set method basic
        set user-database "LDAP-fortiad"
    next
end
```

```
config authentication rule
    edit "ZTNA-Auth-rule"
        set srcintf "port3"
        set srcaddr "all"
        set active-auth-method "ZTNA-Auth-scheme"
    next
end
```

To append security posture tag and authentication settings to the simple ZTNA policy:

1. Go to *Policy & Objects* > *Firewall Policy* and edit the *ZTNA-webserver-fp* policy.
2. For the *Source* field, click the *+* and add the user group named *LDAP-Remote-Allowed-Group*.
3. For the *Security Posture Tag* field, click the *+* and add the *Domain-Users tag*.
4. Click *OK*.

To verify the configuration:

1. Connect to the web server from a client.
2. After selecting the client certificate, the browser will prompt for a username and password. Enter the username (tsmith) and their password.

   Upon a successful authentication, the user can access the web server.
3. On the FortiGate, verify that the logs for the allowed traffic show the user tsmith and the tag Domain-Users:

```
# execute log filter field subtype ztna
# execute log display
18 logs found.
10 logs returned.
1: date=2026-02-11 time=18:06:50 eventtime=1770862009275387788 tz="-0800" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=28457
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.88.0.3
```

```
dstport=9043 dstintf="port2" dstintfrole="dmz" sessionid=10791 srcuuid="b458a65a-f759-51ea-
d7df-ef2e750026d1" dstuuid="06408606-ec26-51f0-9989-f2b848c420fd" service="tcp/9043"
proxyapptype="ztna-proxy" proto=6 action="accept" policyid=17 policytype="policy"
poluuid="8d902222-ecf0-51f0-8a9d-8b226df29093" policyname="ZTNA-webserver-fp"
appcat="unscanned" duration=126 user="tsmith" group="LDAP-Remote-Allowed-Group"
authserver="LDAP-fortiad" gatewayid=1 realserverid=1 vip="ZTNA_P1" vipincomingip=10.0.3.10
accessproxy="ZTNA_S1" clientdeviceid="9A016B5A6E914B42AD4168C066EB04CA"
clientdevicemanageable="manageable" clientdeviceems="FCTEMS8825007166"
clientdevicetags="Domain-Users/all_registered_clients" clientcert="yes" emsconnection="online"
wanin=302884 rcvdbyte=302884 wanout=4424 lanin=4187 sentbyte=4187 lanout=305743
fctuid="9A016B5A6E914B42AD4168C066EB04CA"
```

Types of security posture tags

Security posture tags are tags retrieved from FortiClient EMS once a FortiGate is connected to an EMS through the security fabric connector. FortiGate can retrieve different categories of tags from EMS:

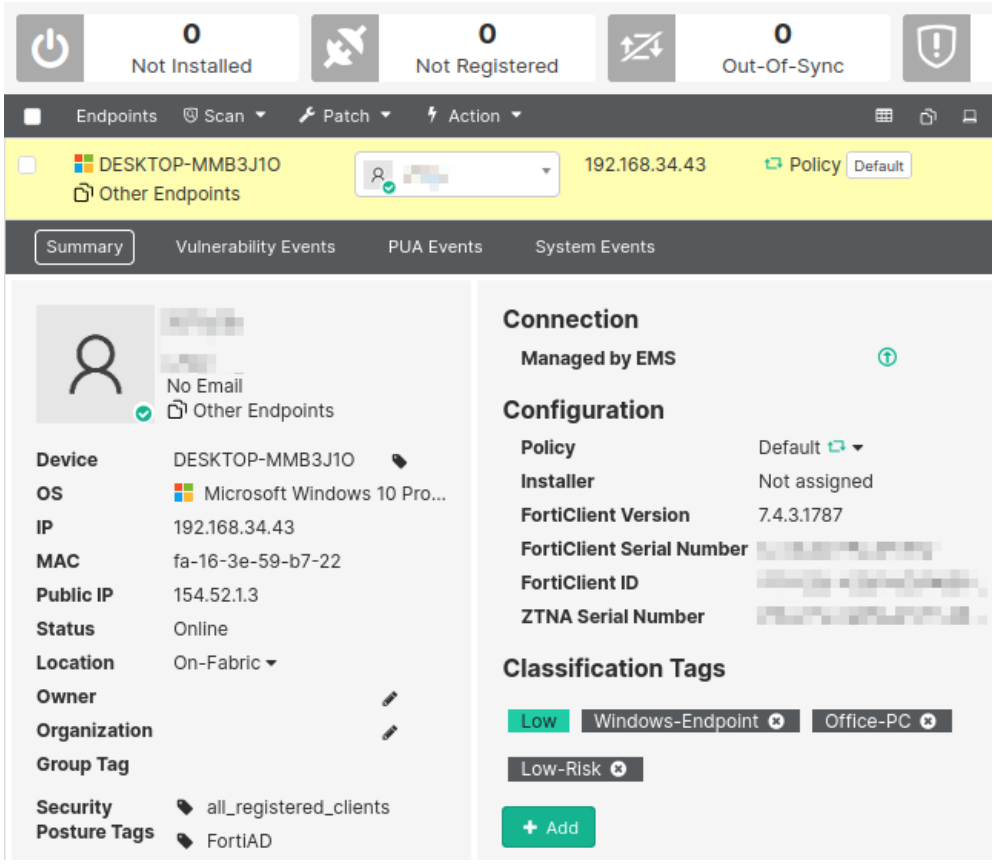
Category	Description
Security posture tags (Zero trust tags in earlier EMS versions)	Generated from tagging rules configured on FortiClient EMS. Tags are based on various posture checks that can be applied to the endpoints. Tags are synchronized from EMS to FortiGate by default.
Classification tags	Custom tags manually assigned by the EMS administrator to endpoints.
FortiGuard outbreak alert tags	Outbreak alerts inform EMS about vulnerabilities on an application or endpoint that is part of a recent outbreak. See FortiGuard Outbreak Alerts.
Fabric tags	Fabric tags are retrieved from FortiAnalyzer to EMS when endpoint logs trigger indicators of compromise (IOC) on a specific endpoint.

For more information about each type of tag on FortiClient EMS, see Fabric Devices.

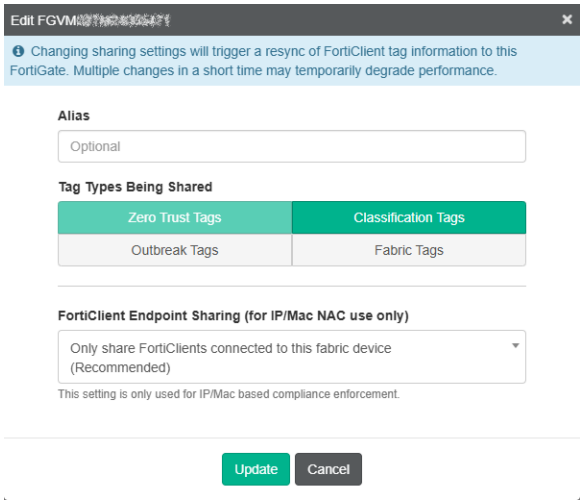
By default, only security posture tags/zero trust tags are synchronized from EMS to FortiGate. On EMS, you can enable other tags to be synchronized.

To synchronize classification tags from EMS:

1. From FortiClient EMS, go to *Endpoints* > *All Endpoints*.
2. Select an endpoint to view its summary.
3. Under *Classification Tags*, click *Add* to add a new custom tag.

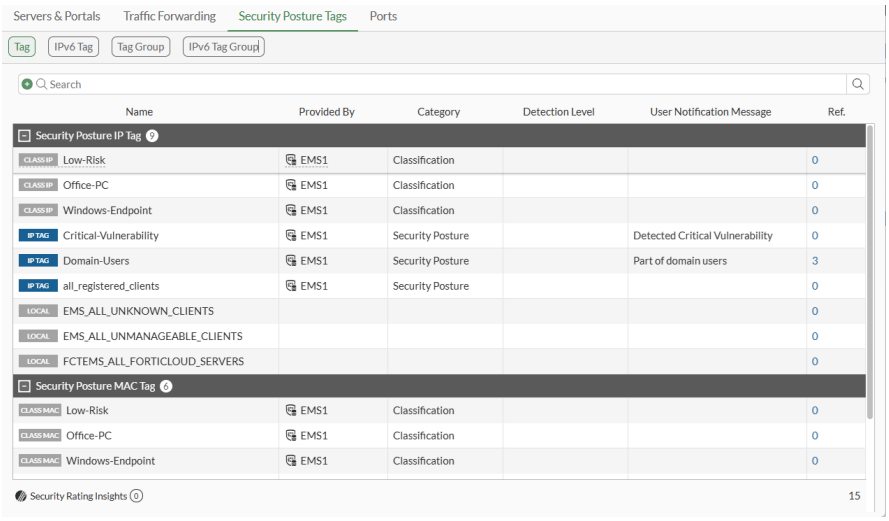


4. From *Fabric & Connectors* > *Fabric Devices*, find the FortiGate that requires the use of these tags.
5. Click *Edit* to update its synchronization settings.
6. Under *Tag Types Being Shared*, select all types that apply. In this example, *Classification Tags* is selected in addition to the default *Zero Trust Tags*.



7. Click *Update* to save.
8. On the FortiGate, go to *Policy & Objects* > *ZTNA*. Switch to the *Security Posture Tags* tab.

   Aside from the Security Posture type tags generated from tagging rules, Classification type tags are also retrieved.



IP versus MAC security posture tags

FortiOS utilizes security posture tags in ZTNA policies and standard firewall policies, however each policy type uses different information from within the tag.

ZTNA policy

ZTNA policies allow users to securely access resources through an encrypted access proxy. This simplifies remote access by reducing the need for remote access VPN and allowing only specific application access. ZTNA policies do not rely on the IP address or MAC address of the connecting clients. Instead, the aim of ZTNA is to provide device identification, user authentication, and role-based application access. This means that using an IP or MAC security posture tag will yield the same result.

Firewall policy

Firewall policies use ZTNA security posture tags in the source and destination fields to provide an additional factor for identification when implementing role-based zero trust access. This means that the tag address type (IP or MAC) is considered by the FortiGate and must be selected appropriately. These policies typically apply to local on-net users, and may also be used for remote users when paired with VPNs.

FortiGate will only learn the IP/MAC address information of directly connected FortiClient endpoints (for example, the gateway of the FortiClient). FortiGates may also learn the IP and MAC address for FortiClient endpoints that are directly connected to another FortiGate when EMS has been configured to share this tagging information with other FortiGates in a Security Fabric. See Configuring EMS to share tagging information with multiple FortiGates for more details.

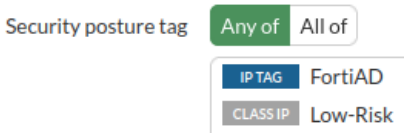
Using any versus all tags inside a ZTNA policy

When configuring the ZTNA policy to control access, the *Security posture tags* option can be set to *Any* or *All*:

- When set to *Any*, an endpoint must satisfy only one of the defined security posture tags to pass this check.
- When set to *All*, an endpoint must satisfy all of the defined security posture tags to pass this check.

Example 1: Using the Any of tag

1. A simple ZTNA policy is configured where the *Security posture tag* is set to *Any of*, with the tags *FortiAD* and *Low-Risk*.

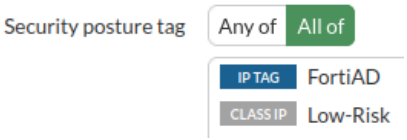


2. An endpoint with only the FortiAD tag connects.
3. The endpoint can connect to the server.
4. From the CLI, ZTNA logs indicate that the traffic was allowed.

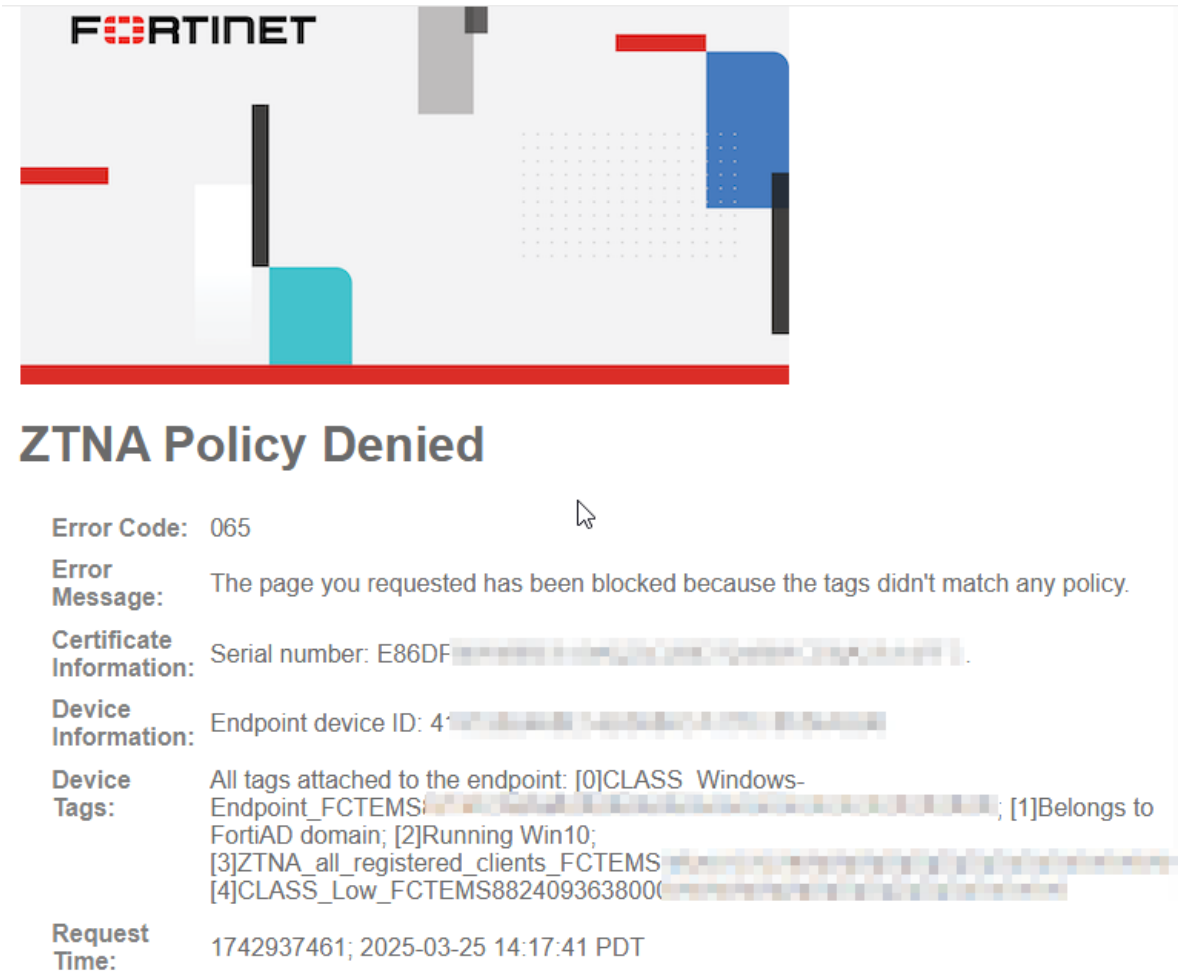
```
# execute log filter field subtype ztna
# execute log display
…
2: date=2025-01-07 time=10:57:00 eventtime=1736276220378088305 tz="-0800" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=17322
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.0.3.10
dstport=9043 dstintf="root" dstintfrole="undefined" sessionid=8876 srcuuid="b458a65a-f759-
51ea-d7df-ef2e750026d1" service="tcp/9043" proxyapptype="http" proto=6 action="accept"
policyid=17 policytype="policy" poluuid="8065fc36-cd28-51ef-cd5a-84f5ae253c13"
policyname="ZTNA-webserver-allow" appcat="unscanned" duration=0 vip="ZTNA-webserver"
accessproxy="ZTNA-webserver" clientdevicemanageable="unknown" clientcert="no" wanin=0
rcvdbyte=0 wanout=0 lanin=1764 sentbyte=1764 lanout=3049
```

Example 2: Using the All of tag

1. A simple ZTNA policy is configured where the *Security posture tag* is set to *All of*, with the tags *FortiAD* and *Low-Risk*.



2. An endpoint with the following tags connects:
   - Low (this is a different tag than Low-Risk, the tag configured in the policy)
   - Windows-Endpoint
   - FortiAD
   - Windows10
   - all_registered_clients
3. It is unable to connect to the server, and a denied message appears.



4. From the CLI, ZTNA logs indicate that the implicit policy denied traffic. Tags are logged under `clientdevicetags`, and it indicates the tags registered to the endpoint. *Low-Risk* is not associated with this endpoint.

```
# execute log filter field subtype ztna
# execute log display
5 logs found.
5 logs returned.

7: date=2025-03-25 time=14:17:42 eventtime=1742937461544682088 tz="-0700" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=192.168.34.43 srcport=56662
srcintf="port1" srcintfrole="undefined" dstcountry="Reserved" srccountry="Reserved"
dstip=192.168.34.181 dstport=443 dstintf="root" dstintfrole="undefined" sessionid=2811
service="HTTPS" proxyapptype="http" proto=6 action="deny" policyid=0 policytype="policy"
trandisp="dnat" tranip=192.168.34.181 tranport=443 appcat="unscanned" duration=0 vip="ZTNA-
server" accessproxy="ZTNA-server" clientdeviceid="4197282468E54B0080CA170C8536A596"
clientdevicemanageable="manageable" clientdeviceems="FCTEMS8824093638"
clientdevicetags="CLASS_Windows-Endpoint/FortiAD/Windows10/all_registered_clients/CLASS_Low"
clientcert="yes" emsconnection="online" msg="Traffic denied because failed to match a policy
or proxy-policy" wanin=0 rcvdbyte=0 wanout=0 lanin=3710 sentbyte=3710 lanout=7061
fctuid="4197282468E54B0080CA170C8536A596" crscore=30 craction=13107 crlevel="high"
```

ZTNA tag caching and direct tag synchronization

FortiOS supports resilient ZTNA enforcement by allowing tag evaluation to continue using cached endpoint information when communication with FortiClient EMS is temporarily unavailable. This improves reliability by preventing unnecessary access disruptions caused by EMS connectivity issues. The cache duration is configurable and has a default of 180s.

In addition, an optimized tag synchronization model reduces dependency on EMS for real-time tag resolution by enabling direct tag sharing between FortiClient and FortiGate. This streamlined approach minimizes latency, reduces complexity, and enhances overall system stability, while providing uninterrupted ZTNA service in the event of an EMS outage.

To configure the tag cache duration:

```
config endpoint-control fctems
    edit 0
        set out-of-sync-threshold <int>
    next
end
```

To see a listing of the ZTNA cache:

```
# diagnose test app fc 7
(...)
#ID:1
UID: 0D48C06390CA42D78158BE90C480A3F0
EMS: FCTEMS1234567890 00000000000000000000000000000000
State: sysinfo:1, tag:1, tagsz:1, out-of-sync:0, ref:0
Owner: (...)
Cert SN: EA997DBF4(...)184C608C62E2A40B
online: Yes
Sub TS: 0
Route IP:10.6.30.214, vdom:vdom1(1)
Route IP:10.1.100.214, vdom:vdom1(1)
Route IP:255.255.255.255, vdom:vdom1(1)
idx:0, ttdl:1 name:all_registered_clients
idx:4, ttdl:1 name:ZT_EMS_MGMT
idx:5, ttdl:1 name:Group-Membership-Domain-Use
idx:7, ttdl:1 name:ZT_AD_MGMT
idx:8, ttdl:1 name:ZT_OS_WIN
idx:9, ttdl:1 name:ZT_PO_AV_ENABLED
idx:10, ttdl:1 name:ZT_PO_VUL_CRITICAL
idx:11, ttdl:1 name:Group-Membership-Domain-U
idx:12, ttdl:1 name:Group-Membership-Domain-Users
idx:13, ttdl:1 name:ZT_FILE_TESTFILE
```

To see a listing of the gateway VPN cache:

```
# diagnose test app fc 33
Gateway VPN Cache:
0:
  uid: 0D48C06390CA42D78158BE90C480A3F0
    route#0:
    vdom: vdom1
    intf: ipsecvpn_0
    sn: FG181FTK12345678
    IP: 3.1.1.1
Gateway IPv4 Cache:
0: vdom1_3.1.1.1: 0D48C06390CA42D78158BE90C480A3F0
Gateway IPv6 Cache:
```

Certificate verification for real servers

When certificate verification (`verify-cert`) is enabled, FortiGate validates the backend server (real server) certificate against the trust anchors available on the FortiGate (CA certificates). Depending on how the real server is addressed and whether a virtual host (vhost) is defined, FortiGate may also validate that the certificate identity matches the expected server name.

The verification behavior depends on two inputs:

1. **vhost**: If you explicitly define the expected host name for the backend connection, often used for strict matching and for environments where multiple services share the same IP via SNI/virtual hosting.
2. **Real server address type**: If the backend is specified as an IP address or an FQDN.

Scenario 1: ZTNA access proxy with a virtual host using IP address for the real server access

```
config firewall access-proxy
    edit "ZTNA-Server"
        config api-gateway
            edit 1
                set virtual-host VH_ztna
                config realservers
                    edit 1
                        set ip 10.88.0.3
                        set verify-cert enable
                    next
                end
            next
        end
    next
end
```

```
config firewall access-proxy-virtual-host
    edit VH_ztna
        set ssl-certificate acme_wildcard
        set host webserver.ztnademo.com
```

```
    next
end
```

When a client makes a request to the ZTNA resource, FortiGate will use the vhost to match the request to this access proxy. Once matched, the hostname used in the request (webserver.ztnademo.com) will be checked against the certificate provided by the server. If the server does not contain the hostname (or a wildcard match, such as *.ztnademo.com), the connection will be refused. FortiGate will also verify that the real server certificate is signed by a trusted CA in the FortiGate certificate store.

Scenario 2: ZTNA access proxy with a virtual host using FQDN for real server

```
config firewall access-proxy
    edit "ZTNA-Server"
        config api-gateway
            edit 1
                set virtual-host VH_ztna
                config realservers
                    edit 1
                        set addr-type fqdn
                        set address "abc.example2.com"
                        set verify-cert enable
                    next
                end
            next
        end
    next
end
```

```
config firewall access-proxy-virtual-host
    edit VH_ztna
        set ssl-certificate acme_wildcard
        set host webserver.ztnademo.com
    next
end
```

When using a FQDN for the real server, ZTNA client will connect on webserver.ztnademo.com. FortiGate will then check the certificate provided by the real server for the configured real server FQDN. In this case `abc.example2.com`. FortiGate will also verify that the real server certificate is signed by a trusted CA in the FortiGate certificate store.

Scenario 3: No virtual host using IP address for real server

```
config firewall access-proxy
    edit "ZTNA-Server"
        config api-gateway
            edit 1
                config realservers
                    edit 1
                        set ip 10.1.1.1
                        set verify-cert enable
                    next
```

```
                end
            next
        end
    next
end
```

FortiGate will not verify that the IP address `10.1.1.1` is contained in the certificate of the ZTNA resource. FortiGate will only check that the certificate provided is signed by a trusted CA in the FortiGate certificate store.

Scenario 4: No virtual host using FQDN for real server

```
config firewall access-proxy
    edit "ZTNA-Server"
        config api-gateway
            edit 1
                config realservers
                    edit 1
                        set addr-type fqdn
                        set address "webserver.ztnademo.com"

                        set verify-cert enable
                    next
                end
            next
        end
    next
end
```

FortiGate will verify that the hostname `webserver.ztnademo.com` is contained in the certificate of the ZTNA resource. FortiGate will check that the certificate provided is signed by a trusted CA in the FortiGate certificate store.

Certificate verification failure

In the following example, a ZTNA server was created to provide access to a server hosted at 10.88.0.3:9043. This server uses a wildcard certificate: `*.ztnademo.com`. A virtual host is defined to match connections on `badexample.fortidemo.fortinet.com`, and is applied to the access proxy.

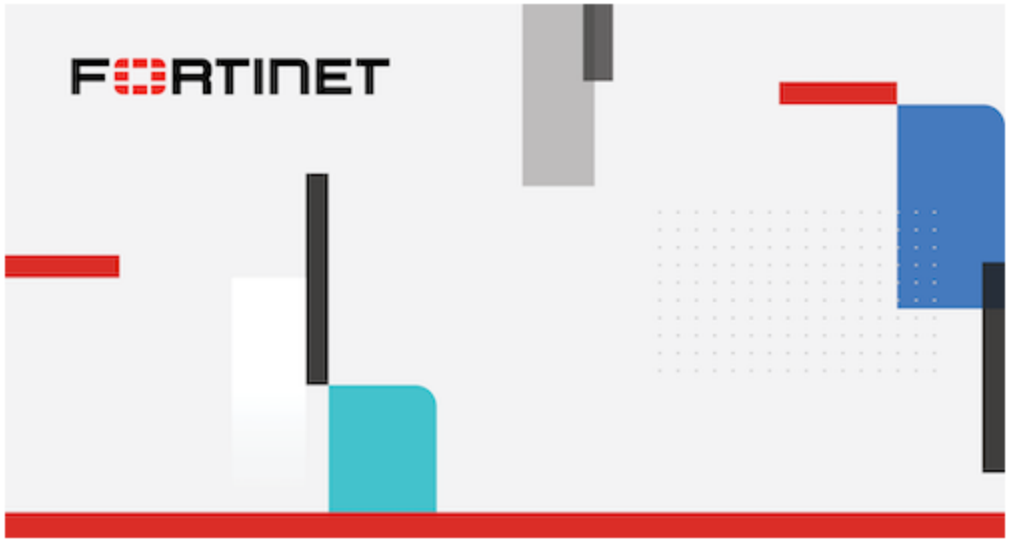
```
config firewall access-proxy
    edit "ZTNA-Server"
        set vip "webserver"
        config api-gateway
            edit 1
                set virtual-host VH_ztna
                config realservers
                    edit 1
                        set ip 10.88.0.3
                        set verify-cert enable
                    next
                end
```

```
            next
        end
    next
end
```

```
config firewall access-proxy-virtual-host
    edit VH_ztna
        set ssl-certificate acme_wildcard
        set host badexample.fortidemo.fortinet.com
    next
end
```

```
config firewall vip
    edit "webserver"
        set type access-proxy
        set server-type https
        set extip 10.0.3.10
        set extintf "port3"
        set extport 9043
        set ssl-certificate "ztna-wildcard"
    next
end
```

This configuration will result in a failure to connect when `verify-cert` is enabled, since the vhost does not match the provided certificate. The following ZTNA Connection Error page is displayed after the client attempts to access the ZTNA server by entering `https://badexample.fortidemo.fortinet.com:9043` in their browser. Note that this address resolves to 10.0.3.10.



ZTNA Connection Error

Error Code:	026
Error Message:	There is a connection issue when attempting to reach the destination website.
Certificate Information:	Serial number: F81D0522E362594B5F416A5D9A862D5F05B4D6F8
Device Information:	N/A.
EMS Information:	Serial number: FCTEMS8825007166; tenant id: 00000000000000000000000000000000.
Destination:	10.0.3.10:9043
Request Time:	1778702926; 2026-05-13 13:08:46 PDT

Output from debug showing the vhost matching the request:

```
[I][p:2233][s:11337]      wad_vs_ssl_port_caps_c2p_on_client_hello:11466 5:webserver: wsp
(0x7fa802b53048) handshake recv ClientHello record 3.1 client 3.3 supported 3.4
[V][p:2233][s:11337]      wad_vs_ssl_c2p_check_alpn         :24818 wsp=0x7fa802b53048, alpn=h2
[V][p:2233][s:11337]      wad_vs_ssl_c2p_check_alpn         :24827 wsp=0x7fa802b53048, vs server
set alpn http2
[V][p:2233][s:11337]      wad_vs_proxy_match_vhost          :4715  5:webserver: matching vhost by:
badexample.fortidemo.fortinet.com
[V][p:2233][s:11337]      wad_pattern_matcher_search        :1226  pattern-match
succ:badexample.fortidemo.fortinet.com
[I][p:2233][s:11337]      wad_vs_proxy_match_vhost          :4723  5:webserver: matched vhost
```

```
(auto-ztna-vhmp46fwrz 0x7fa80108dd68)
[I][p:2233][s:11337]      wad_vs_ssl_port_client_hello_validate_psk:10051 vsp(0x7fa802b53048)
```

Output from debug showing the client providing a valid certificate that it received from EMS:

```
[V][p:2233][s:11339]      wad_vs_ssl_port_caps_on_handshake_recv:11679 sp=0x7fa802b53048/10 recv
type=11 len=1062
[I][p:2233][s:11339]      wad_vs_ssl_access_proxy_on_clt_certs:13721 5:webserver: received certs
from the client.
[I][p:2233][s:11339]      wad_ssl_cert_auth_find            :177   Found auth_info (stable)
[V][p:2233][s:11339]      wad_vs_ssl_access_proxy_on_clt_certs:13735 5:webserver: vs, Found the
cert, and issued by:ems
[V][p:2233][s:11339]      wad_vs_ssl_access_proxy_on_clt_certs:13739 5:webserver: hit the cached
auth_result(7)
[I][p:2233][s:11339]      wad_vs_ssl_access_proxy_on_clt_certs:13776 5:webserver: Cert auth
success. issued_by: ems
[V][p:2233][s:11339]      wad_vs_ssl_port_caps_c2p_on_cert_verify:12830 wsp 0x7fa802b53048
CertificateVerify signature_scheme 0804[V][p:2233][s:11339]
```

Output from debug showing the FortiGate connecting to the server using its internal interface IP (10.88.0.254) as NAT is enabled in the firewall policy:

```
[V][p:2233][s:11339][r:84] wad_http_connect_srv             :894   [0x7fa802b43048] Connect to
server: 10.88.0.3:9043/10.88.0.3:9043
[I][p:2233][s:11339][r:84] wad_http_ses_ctx_find_conn_port  :668   ssi: 0x7fa80292f9c8 is looking
for conn_port by vf:0 from:any to:10.88.0.3:9043 outif:-1
[I][p:2233][s:11339][r:84] wad_tcp_port_alloc               :1488  alloc tcp_port=0x7fa802b802d8
[V][p:2233][s:11339][r:84] wad_tcp_port_connect_with_fd     :2253  oif =4, fwd_oif=0 src_addr_
unknown=1
[I][p:2233][s:11339][r:84] wad_tcp_port_connect_with_fd     :2300  TCP port=0x7fa802b802d8
sock=126 vrf=0 connecting 10.88.0.254:21228->10.88.0.3:9043
```

Output from debug showing the FortiGate connecting to the server using its internal interface IP (10.88.0.254) as NAT is enabled in the firewall policy:

```
[I][p:2233][s:11339]      wad_ssl_cert_auth_find            :177   Found auth_info (stable)
[I][p:2233][s:11339]      wad_ssl_cert_host_verify          :98    cert '' not-match
'badexample.fortidemo.fortinet.com'
[I][p:2233][s:11339]      wad_ssl_port_caps_1way_clt_on_srv_certs:15093 cert '' does not match
hostname 'badexample.fortidemo.fortinet.com'
[I][p:2233][s:11339]      wad_vs_ssl_port_caps_on_p2s_alert_sent:13344 fts send alert level=2
desc=certificate unknown
```

Compare this to accessing the server via a virtual host configured as follows:

```
config firewall access-proxy-virtual-host
    edit "auto-ztna-vhmp46fwrz"
        set ssl-certificate "ztna-wildcard"
        set host "webserver.ztnademo.com"
    next
end
```

Debug output showing certificate authentication success:

```
[I][p:2233][s:11665]      wad_vs_ssl_port_caps_p2s_on_server_hello:11624 wsp 0x7fa802b54420
handshake recv ServerHello 0303
[I][p:2233][s:11665]      wad_vs_ssl_p2s_check_alpn         :24859 wsp 0x7fa802b54420 select http2
[V][p:2233][s:11665]      wad_vs_ssl_port_caps_p2s_on_server_hello:11659 wsp 0x7fa802b54420
extended-master-secret
[V][p:2233][s:11665]      wad_ssl_port_caps_on_handshake_recv:13066 sp=0x7fa802b54420/11 recv
type=11 len=1407
[I][p:2233][s:11665]      wad_ssl_cert_auth_find    :177   Found auth_info (stable)
[I][p:2233][s:11665]      wad_ssl_cert_host_verify  :98    cert '' match
'webserver.ztnademo.com'matched name=*.ztnademo.com
[I][p:2233][s:11665]      wad_ssl_cert_get_auth_status      :2118  cert:status=fnbam result=good
stale=0 ocsp:status=good reason=unspecified cfg:expired=2 revoked=2 timeout=2 failure=2
untrusted=2
[I][p:2233][s:11665]      wad_ssl_port_caps_1way_clt_on_srv_certs:15153 sp=0x7fa802b54420/11 cert
auth success
```

ZTNA advanced configurations

This section includes the following ZTNA advanced configurations:

- Access control of unmanageable and unknown devices on page 1635
- HTTP2 connection coalescing and concurrent multiplexing for ZTNA on page 1641
- Fabric integration with FortiGSLB on page 1644
- ZTNA agentless web-based application access on page 1649
- ZTNA domain matching on page 1660
- ZTNA service connector NEW on page 1667
- ZTNA connector - reverse gateway and forwarder New on page 1674

Access control of unmanageable and unknown devices

The ZTNA application gateway can determine whether a client device that does not have FortiClient installed is a mobile device that is considered unmanageable, or is a non-mobile device that is considered unknown. The ZTNA application gateway tags the device as either `EMS_ALL_UNMANAGEABLE_CLIENTS` or `EMS_ALL_UNKNOWN_CLIENTS` respectively. The FortiGate WAD process achieves this by either matching device TLS fingerprints against a library or learning information from the HTTP User-Agent header if the `set user-agent-detect` setting is enabled.

Configuring the ZTNA access proxy and proxy policy

The `EMS_ALL_UNMANAGEABLE_CLIENTS` and `EMS_ALL_UNKNOWN_CLIENTS` tags allow for ZTNA access control of unmanageable and unknown devices using a proxy policy. The `accept-unmanageable` option for the `empty-cert-action` setting allows unmanageable clients to continue ZTNA proxy rule processing.

```
config firewall vip
    edit <name>
        set type access-proxy
        set client-cert enable
        set user-agent-detect {enable | disable}
        set empty-cert-action {accept | block | accept-unmanageable}
    next
end
```

Command	Description
`type access-proxy`	Configure an access proxy.
`client-cert enable`	Enable requesting client certificate.
`user-agent-detect {enable | disable}`	Enable/disable detecting the device type by HTTP User-Agent if no client certificate is provided (default = enable).
`empty-cert-action {accept | block | accept-unmanageable}`	Set the action for an empty client certificate: • `accept`: accept the SSL handshake if the client certificate is empty. • `block`: block the SSL handshake if the client certificate is empty. • `accept-unmanageable`: accept the SSL handshake only if the end point is unmanageable.

The `user-agent-detect` and `empty-cert-action` settings can only be configured in the CLI.

```
config firewall proxy-policy
    edit <id>
        set ztna-ems-tag {EMS_ALL_UNMANAGEABLE_CLIENTS | EMS_ALL_UNKNOWN_CLIENTS}
    next
end
```

Command	Description
`ztna-ems-tag {EMS_ALL_UNMANAGEABLE_CLIENTS | EMS_ALL_UNKNOWN_CLIENTS}`	Set the EMS tag names: • `EMS_ALL_UNMANAGEABLE_CLIENTS`: match any device that is unmanageable. • `EMS_ALL_UNKNOWN_CLIENTS`: match any device that is not recognized.

Consider the following use cases.

- Case 1: if a client device sends a TLS client hello in a mobile pattern, then WAD will try to match its TLS fingerprint with a WAD original library and mark it with an `EMS_ALL_UNMANAGEABLE_CLIENTS` tag.
- Case 2: if WAD cannot match the TLS fingerprint with an original library but `user-agent-detect` is enabled (under `config firewall access-proxy`), WAD will try to learn the device type from client request's User-Agent header. If it matches a mobile device, then it is still marked with an `EMS_ALL_UNMANAGEABLE_CLIENTS` tag.
- Case 3: if WAD cannot match the TLS fingerprint with an existing original or temporary library, or cannot learn it from User-Agent header, or `user-agent-detect` is disabled, then it will mark the device as `EMS_ALL_UNKNOWN_CLIENTS`.

In the access proxy settings, if `empty-cert-action` is set to `accept-unmanageable`, then only case 1 and 2 would go through the proxy policy. Case 3 would be denied, and a replacement message page would appear.

To configure ZTNA policy access control of unmanageable devices:

1. Configure the client certificate actions:

```
config firewall vip
    edit “ztna_server”
        set client-cert enable
        set user-agent-detect enable
        set empty-cert-action accept
    next
end
```

2. Configure the proxy policy with the ZTNA EMS tag to control device access:

```
config firewall proxy-policy
    edit 1
        set proxy ztna-proxy
        set ztna-proxy "ztna_server"
        set srcintf "port2"
        set srcaddr "all"
        set dstaddr "all"
        set ztna-ems-tag "EMS_ALL_UNMANAGEABLE_CLIENTS"
    next
end
```

Configuring dynamic address local tags

Like other security posture tags,`EMS_ALL_UNMANAGEABLE_CLIENTS` and `EMS_ALL_UNKNOWN_CLIENTS` are dynamic addresses on the FortiGate. The following diagnostic commands can be used to view local tag information:

- `diagnose firewall dynamic address`: a list of unmanageable and unknown clients' IP addresses associated with the `EMS_ALL_MANAGEABLE_CLIENTS` and `EMS_ALL_UNKNOWN_CLIENTS` dynamic addresses, respectively, is displayed.
- `diagnose user-device-store device memory list`: when device detection is enabled on a FortiGate interface that has a layer 2 connection to unmanageable and unknown device clients, then a client's device information is displayed.

To verify the list of dynamic firewall addresses in the CLI:

```
(vdom1) # diagnose firewall dynamic address
List all dynamic addresses:
IP dynamic addresses in VDOM vdom1(vfid: 1):
...
CMDB name: EMS_ALL_UNMANAGEABLE_CLIENTS
EMS_ALL_UNMANAGEABLE_CLIENTS: ID(101)
        ADDR(10.1.100.22)
Total IP dynamic range blocks: 1.
```

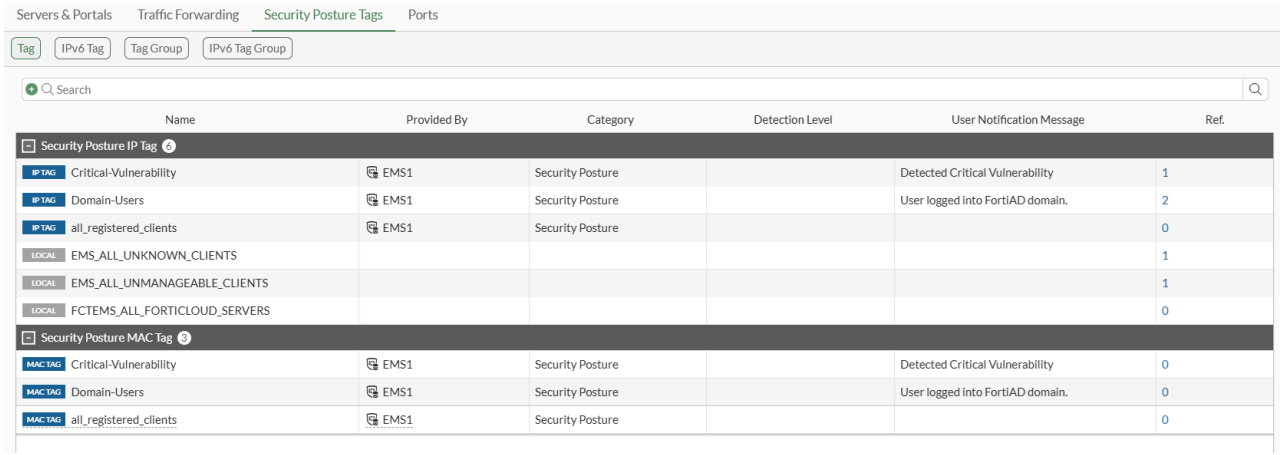
```
Total IP dynamic addresses: 1.
CMDB name: EMS_ALL_UNKNOWN_CLIENTS
EMS_ALL_UNKNOWN_CLIENTS: ID(154)
Total IP dynamic range blocks: 0.
Total IP dynamic addresses: 0.
...
```

To verify the client device information in the CLI:

```
(vdom1) # diagnose user-device-store device memory list
Record #1:
        ...
       device_info
              ...
              'is_online' = 'true'
              'is_ems_registered' = 'false'
              'active_start_time' = '1668811449'
              'is_fortiguard_src' = 'false'
              'tags' = 'EMS_ALL_UNMANAGEABLE_CLIENTS'
              ...
       interface_info
       ...
```

To view the local tag information in the GUI:

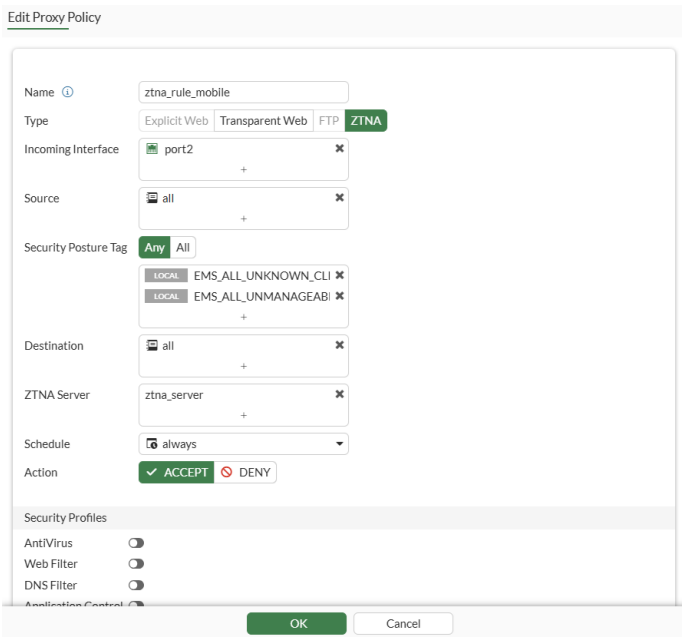
1. Go to *Policy & Objects* > *ZTNA* and select the *Security Posture Tag* tab.



2. Hover over a tag to view the tooltip, which displays matched endpoints and resolved addresses.

To apply a local tag in a full ZTNA policy:

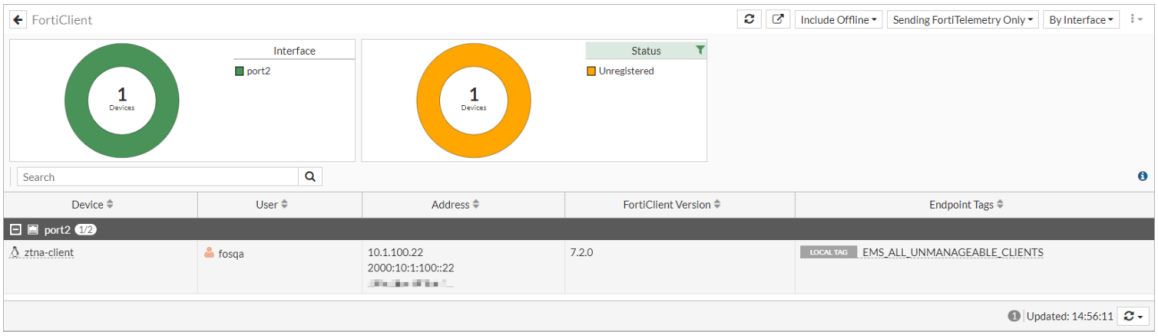
1. Go to *Policy & Objects* > *Proxy Policy*.
2. Click *Create New*, or select and edit an existing entry.
3. In the *Security Posture Tag* field, click the *+* to add tags. The local tags appear in the *IP* section.
4. Configure the other settings as needed.



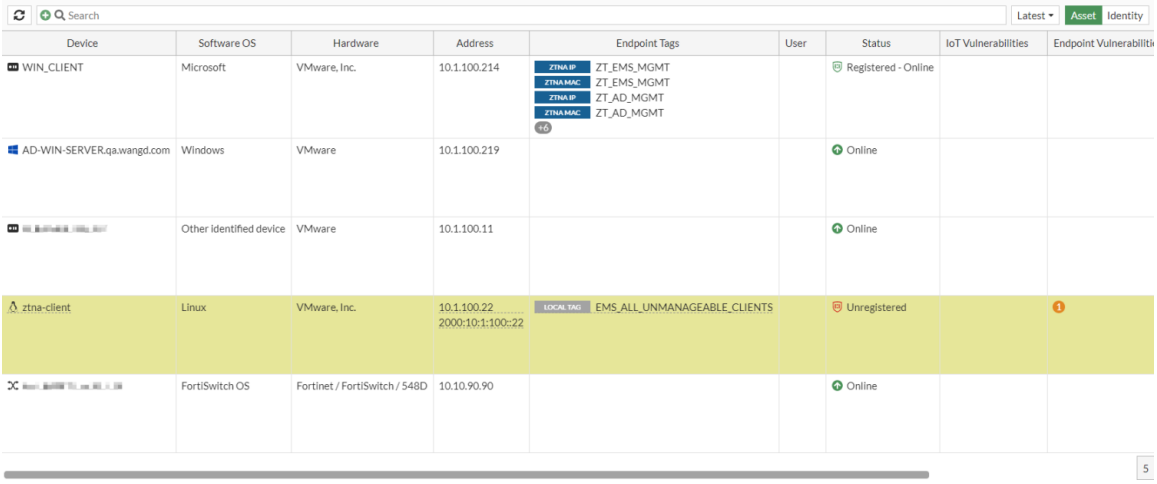
5. Click *OK*.

Local tag information is also available in the following GUI widgets and pages:

- *Dashboard* > *FortiClient* widget



- *Security Fabric* > *Asset Identity Center* page



Viewing ZTNA traffic logs

ZTNA traffic logs include the following fields related to unmanageable and unknown devices.

- Client connection status with EMS server with possible values of unknown, offline, or online:
  - CLI = `emsconnection`
  - GUI = *EMS Connection*
- Device manageability status with possible values of unknown, manageable, or unmanageable:
  - CLI = `clientdevicemanageable`
  - GUI = *Client Device Manageable*

The device manageability status can have one of the following values:

- Unknown: traffic from a client with an unknown TLS fingerprint and where the user agent information is not available for learning.
- Manageable: traffic from a non-mobile device (platform or operating system), with a known TLS fingerprint, or where the user agent information is available for learning.
- Unmanageable: traffic from a mobile device with a known mobile TLS fingerprint or user agent information is available for learning.

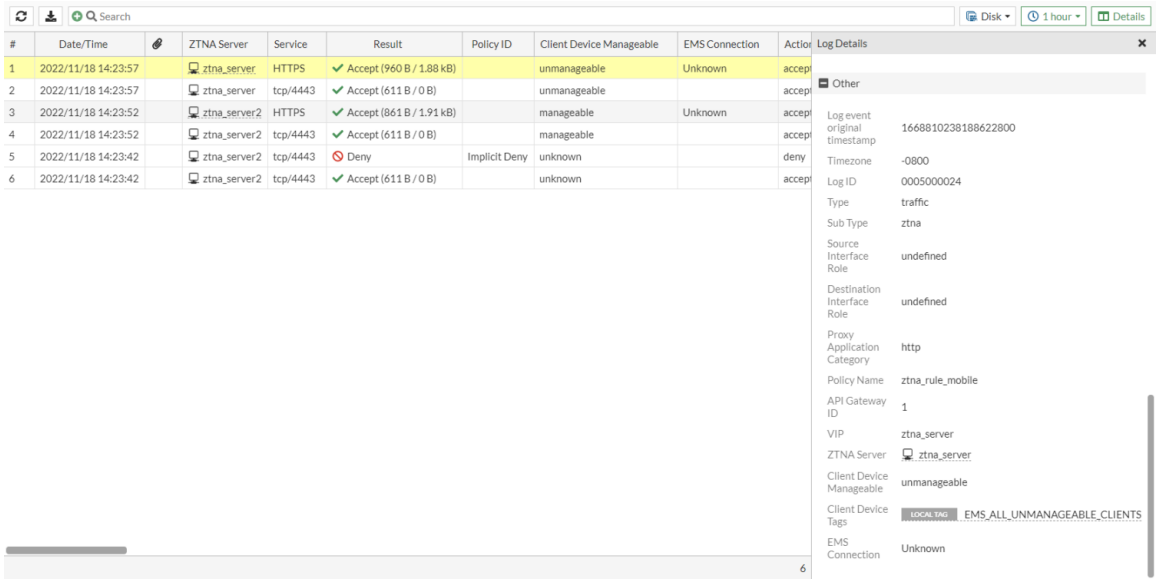
To view the ZTNA traffic logs in the CLI:

```
(vdom1)# execute log filter category 0
(vdom1)# execute log filter field subtype ztna
(vdom1)# execute log display

1: date=2022-11-18 time=14:23:57 eventtime=1668810238188622828 tz="-0800" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="vdom1" srcip=10.1.100.22 srcport=41400
srcintf="port2" srcintfrole="undefined" dstcountry="Reserved" srccountry="Reserved"
dstip=172.16.200.207 dstport=443 dstintf="port1" dstintfrole="undefined" sessionid=12147
service="HTTPS" proxyapptype="http" proto=6 action="accept" policyid=1 policytype="proxy-policy"
poluuid="03a79dd2-6775-51ed-19a0-444a0314f1a0" policyname="ztna_rule_mobile" duration=0
gatewayid=1 vip="ztna_server" accessproxy="ztna_server" clientdeviceid="pf-mobile;os-unknown;app-
safari" clientdevicemanageable="unmanageable" clientdevicetags="EMS_ALL_UNMANAGEABLE_CLIENTS"
emsconnection="unknown" wanin=1884 rcvdbyte=1884 wanout=833 lanin=960 sentbyte=960 lanout=3046
fctuid="pf-mobile;os-unknown;app-safari" appcat="unscanned"

3: date=2022-11-18 time=14:23:52 eventtime=1668810232937847134 tz="-0800" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="vdom1" srcip=10.1.100.22 srcport=46392
srcintf="port2" srcintfrole="undefined" dstcountry="Reserved" srccountry="Reserved"
dstip=172.16.200.209 dstport=443 dstintf="port1" dstintfrole="undefined" sessionid=12144
service="HTTPS" proxyapptype="http" proto=6 action="accept" policyid=2 policytype="proxy-policy"
poluuid="141b7db8-6785-51ed-32a5-58d696e60e2d" duration=0 gatewayid=1 vip="ztna_server2"
accessproxy="ztna_server2" clientdeviceid="pf-pc;os-unknown;app-curl"
clientdevicemanageable="manageable" clientdevicetags="EMS_ALL_UNKNOWN_CLIENTS"
emsconnection="unknown" wanin=1907 rcvdbyte=1907 wanout=699 lanin=861 sentbyte=861 lanout=3089
fctuid="pf-pc;os-unknown;app-curl" appcat="unscanned"

5: date=2022-11-18 time=14:23:42 eventtime=1668810222897968134 tz="-0800" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="vdom1" srcip=10.1.100.22 srcport=46390
srcintf="port2" srcintfrole="undefined" dstcountry="Reserved" srccountry="Reserved"
```

```
dstip=172.18.62.68 dstport=4443 dstintf="vdom1" dstintfrole="undefined" sessionid=12134
service="tcp/4443" proxyapptype="http" proto=6 action="deny" policyid=0 policytype="proxy-policy"
duration=0 vip="ztna_server2" accessproxy="ztna_server2" clientdevicemanageable="unknown"
msg="Denied: failed to match a proxy-policy" wanin=0 rcvdbyte=0 wanout=0 lanin=806 sentbyte=806
lanout=2661 appcat="unscanned" crscore=30 craction=131072 crlevel="high"
```

To view the ZTNA traffic logs in the GUI:

1. Go to *Log & Report > ZTNA Traffic*.
2. Select an entry and click *Details*.
3. Check the *Client Device Manageable* and *EMS Connection* fields.



HTTP2 connection coalescing and concurrent multiplexing for ZTNA

HTTP2 connection coalescing and concurrent multiplexing allows multiple HTTP2 requests to share the same TLS connection when the destination IP is the same, and the host names are compatible in the certificate. In ZTNA scenarios, the FortiGate application gateway may accept multiple HTTP2 requests to the same ZTNA server destined to different virtual hosts on the same real server. These HTTP2 requests can share the same TLS connection between the FortiGate and the real server so that the handshake does not need to be performed multiple times for multiple connections.

In order for the FortiGate to match the SNI (Server Name Indication), this SNI value must appear under the SAN extension on the server certificate. Configuring the SNI value under the CN alone will not work.

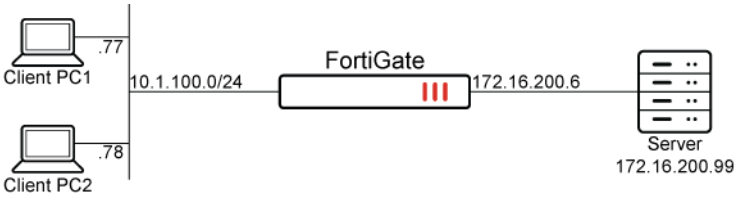
To configure the ZTNA web proxy:

```
config ztna web-proxy
    edit <name>
        set svr-pool-multiplex {enable | disable}
        set svr-pool-ttl <integer>
        set svr-pool-server-max-request <integer>
        set svr-pool-server-max-concurrent-request <integer>
    next
end
```

Command	Description
`svr-pool-multiplex {enable | disable}`	Enable/disable server pool multiplexing. When enabled, share the connected server in HTTP, HTTPS, and web portal API gateway.
`svr-pool-ttl <integer>`	Set the time-to-live in the server pool for idle connections to servers (in seconds, 0 - 2147483647, default = 15).
`svr-pool-server-max-request <integer>`	Set the maximum number of requests that servers in server pool handle before disconnecting (0 - 2147483647, default = 0).
`svr-pool-server-max-concurrent-request <integer>`	Set the maximum number of concurrent requests that servers in the server pool could handle (0 - 2147483647, default = 0).

Example

In this example, multiple clients submit requests in HTTP2. The requests hit the VIP address, and then FortiGate opens a session between itself (172.16.200.6) and the server (172.16.200.99). The coalescing occurs in this session as the multiple streams share the same TLS session to connect to the same destination server.



To configure connection coalescing and concurrent multiplexing with ZTNA:

1. Configure the VIP:

```
config firewall vip
    edit "vip-ztna"
        set type access-proxy
        set extip 10.1.100.223
        set extintf "port2"
        set server-type https
        set client-cert disable
        set h2-support enable
        set extport 443
```

```
        set ssl-certificate "Fortinet_SSL"
    next
end
```

2. Configure the ZTNA server and path mapping:

```
config ztna web-proxy
    edit "ztna"
        set vip "vip-ztna"
        set host “a.ftnt.com”
        set svr-pool-multiplex enable
        config api-gateway
            edit 1
                set url-map "/a"
                config realservers
                    edit 1
                        set ip 172.16.200.99
                    next
                end
            next
        end
    next
end
```

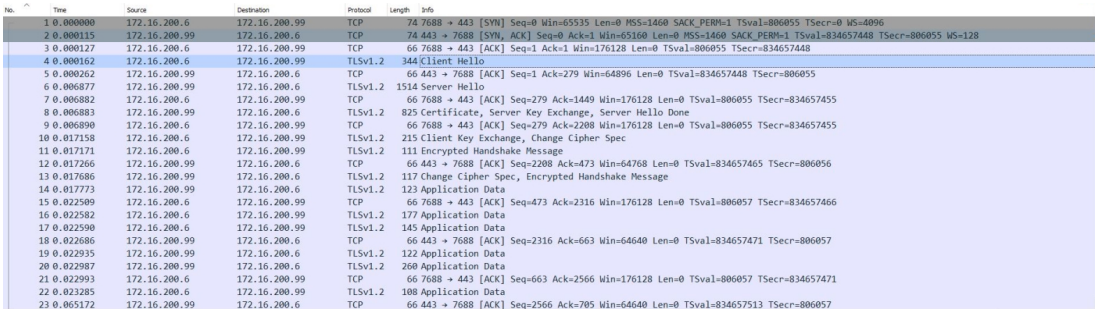
```
config ztna web-proxy
    edit "ztna_b"
        set vip "vip-ztna"
        set host “b.ftnt.com”
        set svr-pool-multiplex enable
        config api-gateway
            edit 1
                set url-map "/b"
                config realservers
                    edit 1
                        set ip 172.16.200.99
                    next
                end
            next
        end
    next
end
```

3. Configure the ZTNA policy:

```
config firewall proxy-policy
    edit 3
        set proxy ztna-proxy
        set ztna-proxy "ztna" "ztna_b"
        set srcintf "port2"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
```

```
        set schedule "always"
        set logtraffic all
        set utm-status enable
        set ssl-ssh-profile "deep-inspection-clone"
        set av-profile "av"
    next
end
```

4. Get the clients to access a.ftnt.com and b.ftnt.com. The clients share access with the same real server and certificate (CN=*.ftnt.com). The FortiGate shares the first TLS connection with second TLS connection.
5. Verify the sniffer packet capture on the FortiGate server side. There is one client hello.



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.16.200.6	172.16.200.99	TCP	74	7688 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1 TSval=806055 TSecr=0 WS=4096
2	0.000115	172.16.200.99	172.16.200.6	TCP	74	443 → 7688 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=834657448 TSecr=806055 WS=128
3	0.000127	172.16.200.6	172.16.200.99	TCP	66	7688 → 443 [ACK] Seq=1 Ack=1 Win=176128 Len=0 TSval=806055 TSecr=834657448
4	0.000162	172.16.200.6	172.16.200.99	TLSv1.2	344	Client Hello
5	0.000262	172.16.200.99	172.16.200.6	TCP	66	443 → 7688 [ACK] Seq=1 Ack=279 Win=64896 Len=0 TSval=834657448 TSecr=806055
6	0.006877	172.16.200.99	172.16.200.6	TLSv1.2	1514	Server Hello
7	0.006882	172.16.200.6	172.16.200.99	TCP	66	7688 → 443 [ACK] Seq=279 Ack=1449 Win=176128 Len=0 TSval=806055 TSecr=834657455
8	0.006883	172.16.200.99	172.16.200.6	TLSv1.2	825	Certificate, Server Key Exchange, Server Hello Done
9	0.006890	172.16.200.6	172.16.200.99	TCP	66	7688 → 443 [ACK] Seq=279 Ack=2208 Win=176128 Len=0 TSval=806055 TSecr=834657455
10	0.017158	172.16.200.6	172.16.200.99	TLSv1.2	215	Client Key Exchange, Change Cipher Spec
11	0.017171	172.16.200.6	172.16.200.99	TLSv1.2	111	Encrypted Handshake Message
12	0.017266	172.16.200.99	172.16.200.6	TCP	66	443 → 7688 [ACK] Seq=2208 Ack=473 Win=64768 Len=0 TSval=834657465 TSecr=806056
13	0.017686	172.16.200.99	172.16.200.6	TLSv1.2	117	Change Cipher Spec, Encrypted Handshake Message
14	0.017773	172.16.200.99	172.16.200.6	TLSv1.2	123	Application Data
15	0.022509	172.16.200.6	172.16.200.99	TCP	66	7688 → 443 [ACK] Seq=473 Ack=2316 Win=176128 Len=0 TSval=806057 TSecr=834657466
16	0.022582	172.16.200.6	172.16.200.99	TLSv1.2	177	Application Data
17	0.022590	172.16.200.6	172.16.200.99	TLSv1.2	145	Application Data
18	0.022686	172.16.200.99	172.16.200.6	TCP	66	443 → 7688 [ACK] Seq=2316 Ack=663 Win=64640 Len=0 TSval=834657471 TSecr=806057
19	0.022935	172.16.200.99	172.16.200.6	TLSv1.2	122	Application Data
20	0.022987	172.16.200.99	172.16.200.6	TLSv1.2	260	Application Data
21	0.022993	172.16.200.6	172.16.200.99	TCP	66	7688 → 443 [ACK] Seq=663 Ack=2566 Win=176128 Len=0 TSval=806057 TSecr=834657471
22	0.023285	172.16.200.6	172.16.200.99	TLSv1.2	108	Application Data
23	0.065172	172.16.200.99	172.16.200.6	TCP	66	443 → 7688 [ACK] Seq=2566 Ack=705 Win=64640 Len=0 TSval=834657513 TSecr=806057

6. Disable server pool multiplexing:

```
config ztna web-proxy
    edit "ztna"
        set vip "vip-ztna"
        set svr-pool-multiplex disable
    next
    edit "ztna_b"
        set vip "vip-ztna"
        set svr-pool-multiplex disable
    next
end
```

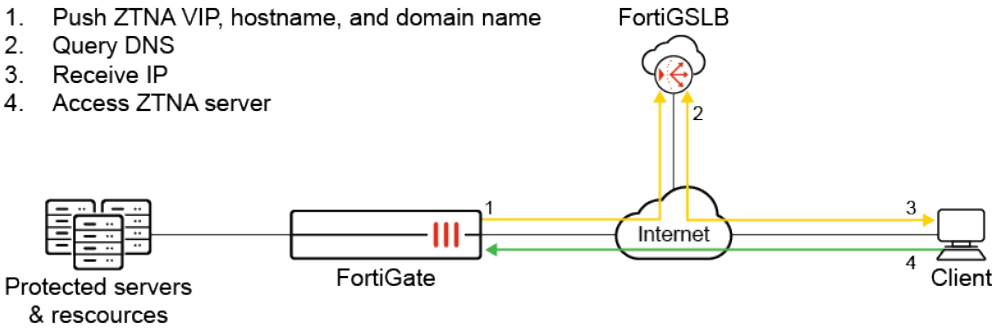
7. Verify the sniffer packet capture. This time, the FortiGate does not coalesce the TLS connection, so there are two client hellos.

No.	Time	Source	Destination	Protocol	Length	Info
27	3.209425	172.16.200.99	172.16.200.6	TLSv1.2	815	Certificate, Server Key Exchange, Server Hello Done
13	0.017192	172.16.200.99	172.16.200.6	TLSv1.2	119	Change Cipher Spec, Encrypted Handshake Message
32	3.219623	172.16.200.99	172.16.200.6	TLSv1.2	119	Change Cipher Spec, Encrypted Handshake Message
4	0.000160	172.16.200.6	172.16.200.99	TLSv1.2	346	Client Hello
23	3.201384	172.16.200.6	172.16.200.99	TLSv1.2	346	Client Hello
10	0.016623	172.16.200.6	172.16.200.99	TLSv1.2	217	Client Key Exchange, Change Cipher Spec
29	3.219146	172.16.200.6	172.16.200.99	TLSv1.2	217	Client Key Exchange, Change Cipher Spec
11	0.016635	172.16.200.6	172.16.200.99	TLSv1.2	113	Encrypted Handshake Message
30	3.219157	172.16.200.6	172.16.200.99	TLSv1.2	113	Encrypted Handshake Message
6	0.006115	172.16.200.99	172.16.200.6	TLSv1.2	1516	Server Hello
25	3.209418	172.16.200.99	172.16.200.6	TLSv1.2	1516	Server Hello

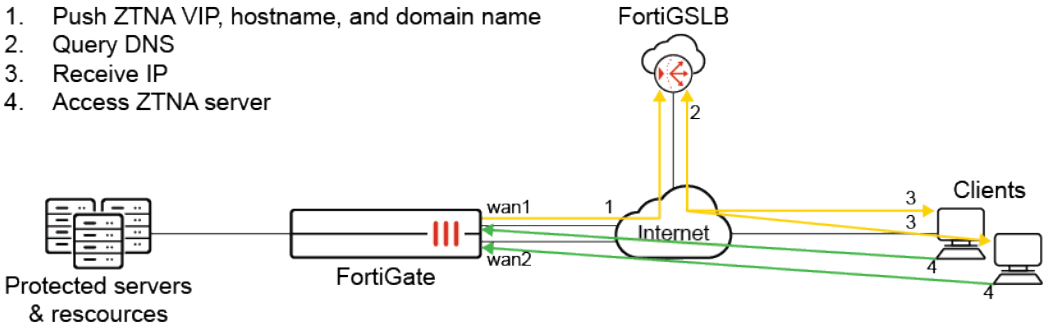
Fabric integration with FortiGSLB

Fabric integration between the FortiGate and FortiGSLB allows a FortiGate to publish custom host and domain names directly to FortiGSLB. This enables external IPs on VIPs used in ZTNA server objects to be published with the host and domain names directly to FortiGSLB, where its DNS service can provide nameserver lookups for the FQDNs.

In a basic use case, the hostname, domain name, and external IP of a ZTNA server can be published, and any subsequent updates to the address are immediately pushed to FortiGSLB.



In more advanced setup, an FQDN may map to different external IPs, which can be load balanced by FortiGSLB.



In addition, FortiGSLB can perform health checks on the external IPs, and then return the link with the better metrics. See How to add FortiGate SD-WAN Inbound Load Balancing to FortiGSLB in the FortiGSLB Handbook for more information.

This feature requires a valid FortiGSLB account contract (FGCS). If no valid FGCS contract is found, the CLI will return a warning message during configurations:

```
No license detected for FortiGSLB.
GSLB configuration and statistics will not be reported unless the account is licensed.
```

To enable VIP and ZTNA server integration with the FortiGSLB Cloud service:

```
config system global
    set fortigslb-integration {enable | disable}
end
```

To configure the FortiGSLB setting in the VIP:

```
config firewall vip
    edit <name>
        set one-click-gslb-server {enable | disable}
        set gslb-hostname <string>
        set gslb-domain-name <string>
        config gslb-public-ips
```

```
            edit <id>
                set ip <IP_address>
            next
        end
    next
end
```

Command	Description
`one-click-gslb-server {enable | disable}`	Enable/disable integration with FortiGSLB.
`gslb-hostname <string>`	Enter the hostname portion of the FQDN that will be used within the configured FortiGSLB domain.
`gslb-domain-name <string>`	Enter the domain name of the FQDN that will be used within the configured FortiGSLB domain.
`ip <IP_address>`	Enter the custom publicly accessible IP address that overrides the external IP address (`extip`). This setting is optional.

Example

In this example, a FortiGate has three WAN interfaces, each configured with different VIPs that are used in ZTNA server objects that point to the same real server. These VIPs are configured with the same GSLB hostname and domain name. As a result, the hostname and domain name are mapped to three different addresses and sent to FortiGSLB. FortiGSLB's default setting will perform load balancing and respond to DNS queries by returning the addresses in a round-robin fashion.

To configure FortiGSLB integration:

1. Enable integration with FortiGSLB in the global settings:

```
config system global
    set fortigslb-integration enable
end
```

2. Enable integration with FortiGSLB on each firewall VIP:

```
config firewall vip
    edit "ztna_vip1"
        set type access-proxy
        set server-type https
        set extip 172.18.62.66
        set extintf "port2"
        set one-click-gslb-server enable
        set gslb-hostname "qa.test"
        set gslb-domain-name "wangd.com"
        set extport 4443
        set ssl-certificate "default.test.com"
    next
```

```
    edit "ztna_vip2"
        set type access-proxy
        set server-type https
        set extip 172.18.62.67
        set extintf "port3"
        set one-click-gslb-server enable
        set gslb-hostname "qa.test"
        set gslb-domain-name "wangd.com"
        set extport 4443
        set ssl-certificate "default.test.com"
    next
    edit "ztna_vip3"
        set type access-proxy
        set server-type https
        set extip 172.18.62.68
        set extintf "port4"
        set one-click-gslb-server enable
        set gslb-hostname "qa.test"
        set gslb-domain-name "wangd.com"
        config gslb-public-ips
            edit 1
                set ip 172.18.62.69
            next
        end
        set extport 4443
        set ssl-certificate "default.test.com"
    next
end
```

3. Enable debugs:

```
# diagnose debug application cloudapid -1
# diagnose debug enable
```

A successful connection will produce output similar to the following:

```
<4234> 10 cloudapi_curl_debug()-19: CURL HEADER OUT: POST /api/v1.0/one-click-glb-
fgt/modifyconfig HTTP/2
Host: 1clickfgt.fortigslb-cloud.com
Accept: application/json
Content-Type: application/json
Content-Length: 553

<4234> 10 cloudapi_curl_debug()-19: CURL DATA OUT: {"members":[{"vdom_name":"vdom1","name_
key":"ztna_vip1","type":"ztna","ip_list":
["172.18.62.66"],"host":"qa.test","domain":"wangd.com"},{"vdom_name":"vdom1","name_key":"ztna_
vip2","type":"ztna","ip_list":["172.18.62.67"],"host":"qa.test","domain":"wangd.com"},{"vdom_
name":"vdom1","name_key":"ztna_vip3","type":"ztna","ip_list":
["172.18.62.69”],"host":"qa.test","domain":"wangd.com"}],"ha_cluster":
[{"sn":"FG181FTK22902632","host_name":"FGT1801F-ZTNA"},{"sn":"FG181FTK22902625","host_
name":"FGT1801F-ZTNA"}],"timestamp":"2023-11-23 00:28:43"}
```

Verification

Upon successfully passing the hostname, domain name, and IP address mappings to FortiGSLB, clients that are using FortiGSLB's DNS for DNS resolution can now get responses to their queries. Results on consecutive queries return the IP addresses in a round-robin fashion.

First query:

```
fosqa@ztna-client4:~/ztna_pytest$ dig @15.197.150.26 qa.test.wangd.com
; <<>> DiG 9.16.1-Ubuntu <<>> @15.197.150.26 qa.test.wangd.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 33860
;; flags: qr aa rd; QUERY: 1, ANSWER: 1, AUTHORITY: 1, ADDITIONAL: 2
;; WARNING: recursion requested but not available

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4096
;; QUESTION SECTION:
;qa.test.wangd.com.             IN      A

;; ANSWER SECTION:
qa.test.wangd.com.      5       IN      A       172.18.62.66

;; AUTHORITY SECTION:
wangd.com.              86400   IN      NS      defaultprimary.wangd.com.

;; ADDITIONAL SECTION:
defaultprimary.wangd.com. 86400 IN      A       15.197.150.26

;; Query time: 15 msec
;; SERVER: 15.197.150.26#53(15.197.150.26)
;; WHEN: Thu Nov 16 10:56:23 PST 2023
;; MSG SIZE  rcvd: 107
```

Second query:

```
fosqa@ztna-client4:~/ztna_pytest$ dig @15.197.150.26 qa.test.wangd.com
; <<>> DiG 9.16.1-Ubuntu <<>> @15.197.150.26 qa.test.wangd.com
…
;; QUESTION SECTION:
;qa.test.wangd.com.             IN      A

;; ANSWER SECTION:
qa.test.wangd.com.      5       IN      A       172.18.62.69
…
```

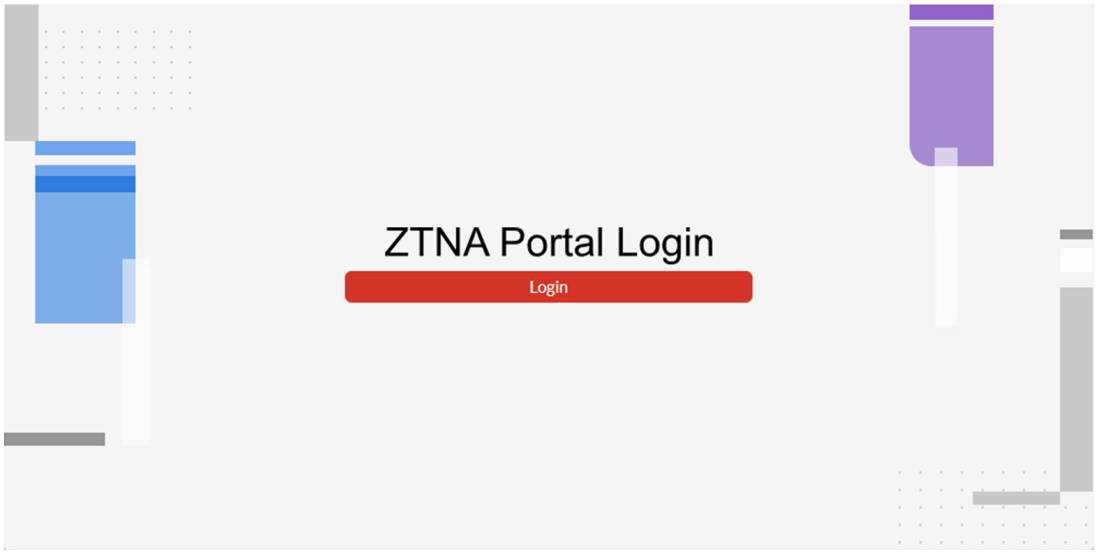
Third query:

```
fosqa@ztna-client4:~/ztna_pytest$ dig @15.197.150.26 qa.test.wangd.com
; <<>> DiG 9.16.1-Ubuntu <<>> @15.197.150.26 qa.test.wangd.com
…
;; QUESTION SECTION:
;qa.test.wangd.com.             IN      A

;; ANSWER SECTION:
qa.test.wangd.com.      5       IN      A       172.18.62.67
…
```

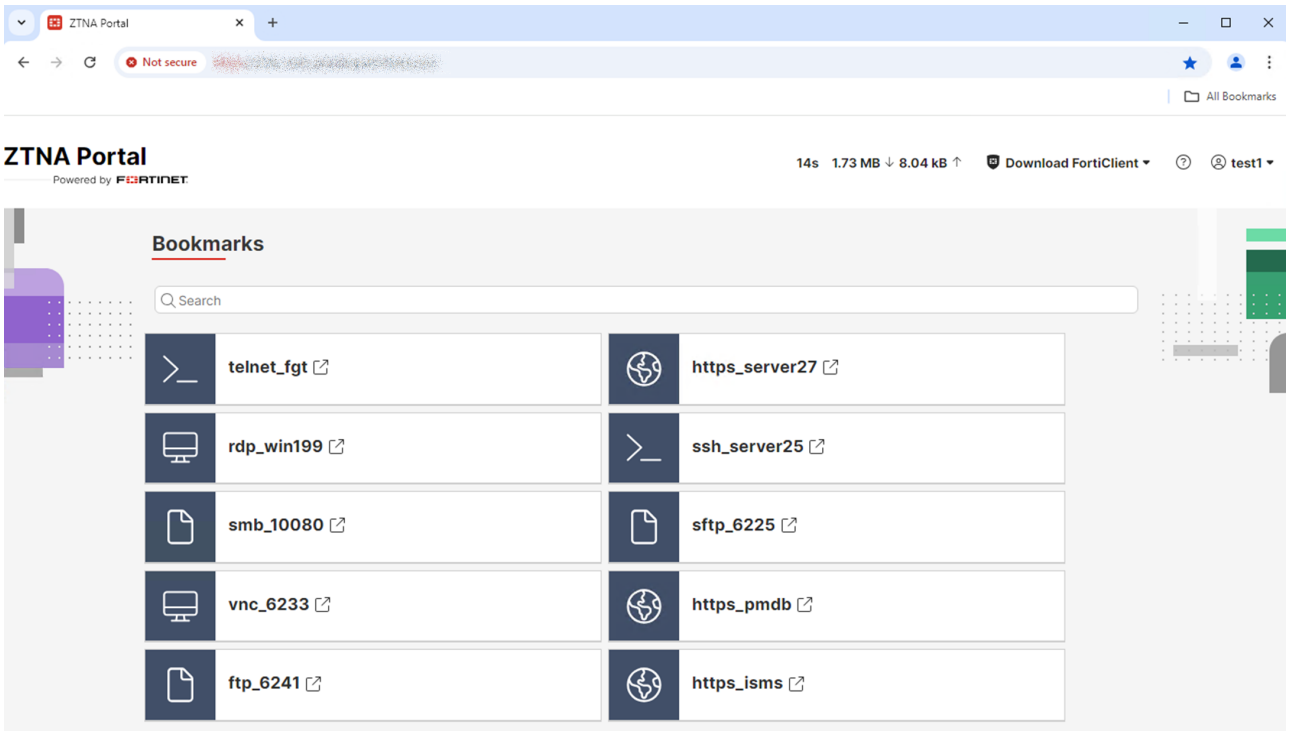
ZTNA agentless web-based application access

A ZTNA web portal is available to provide end-user access to applications without FortiClient or client certificate checks. The ZTNA portal handles authentication and authorization of traffic destined for the protected resources. It is implemented entirely in WAD.

When end-users connect to the ZTNA web portal, they are directed to a login page:



Once logged in, end-users can access bookmarks defined by the administrator:



For web applications, the client must resolve the FQDN to the ZTNA portal VIP. In addition, when accessing the web application, end users are presented with the certificate configured on the ZTNA portal VIP instead of the certificate on the end web server. Hence, the certificate requires the correct Subject Alternate Name(s) to avoid browser certificate errors.

If the ZTNA web portal is enabled without any configured bookmarks, the CLI displays the following warning, and users who access the portal will see a generic portal page with no bookmarks available:

```
Warning: Please ensure at least one bookmark is set in config ztna web-portal-bookmark
for ZTNA Web Portal to be functional.
```

Configuration

To configure a ZTNA port in the GUI:

1. Go to *Policy & Objects* > *ZTNA*, select the *Ports* tab, and click *Create New*.
2. Configure the port:

Option	Description
Name	Name of the ZTNA port
Interface	Listening interface
IP address	External address for the ZTNA Gateway
Port	External port for the ZTNA Gateway

Option	Description
Default certificate	Certificate used for the web portal

3. Click *OK*.

To configure a ZTNA port in the CLI:

Disable `client-cert` to disable the requirement for a client certificate so that agentless clients can connect.

```
config firewall vip
    edit <name>
        set type access-proxy
        set server-type https
        set extip <ip address>
        set extintf <interface>
        set client-cert disable
        set extport <port>
        set ssl-certificate <certificate>
    next
end
```

To create a web portal object in the GUI:

1. Go to *Policy & Objects* > *ZTNA*, select the *Servers & Portals* tab, select *Web Portal*, and click *Create New*.
2. Configure the web portal:

Option	Description
Name	Name of the web portal
Host	Virtual host name that end-users will connect to to access the ZTNA web portal
ZTNA port	Use existing. Select the ZTNA port created in the previous steps.
Authentication scheme	Enable. Create a new authentication scheme that applies to the web portal.
IP-based authentication	Enable to authenticate users per IP

3. Click *OK*.

To create a web portal object in the CLI:

1. Configure an access-proxy virtual host. Disable `client-cert` to allow agentless clients to connect to this destination to access the ZTNA web portal.

```
config firewall access-proxy-virtual-host
    edit "ztna-web-portal-fqdn"
        set host <web portal host name or ip>
        set client-cert disable
```

```
    next
end
```

2. After an authentication scheme is created, the authentication rule for the `ztna-portal` is created automatically.

```
config authentication rule
    edit <rule>
        set protocol ztna-portal
        set active-auth-method <auth scheme>
    next
end
```

3. ZTNA web portal settings:

```
config ztna web-portal
    edit <name>
        set vip <vip name>
        set host <virtual host name>
        set vip6 <virtual IPv6 name>
        set auth-rule <rule>
    next
end
```

Command	Description
`vip <vip name>`	The access-proxy VIP associated with this portal.
`host <virtual host name>`	The access-proxy virtual host object and FQDN defined for accessing this portal. This virtual host object should not conflict with other virtual host objects used in Traffic forwarding and ZTNA web proxy or with the SAML SP host.
`vip6 <virtual IPv6 name>`	The access-proxy VIP6 associated with the ZTNA server and applications that this portal is allowing.

To create a web portal bookmark:

```
config ztna web-portal-bookmark
    edit <name>
        set users <users>
        set groups <groups>
        config bookmarks
            edit <name>
                set apptype {ftp | rdp | sftp | smb | ssh | telnet | vnc | web}
                set url <string>
                set verify-cert {enable | disable}
                set host <name or IP>
                set description <description>
                set port <remote port>
                set sso {enable | disable}
            next
```

```
        end
    next
end
```

Command	Description
`users <users>`	The user(s) allowed to access the web-portal.
`groups <groups>`	The group(s) allowed to access the web-portal.
`apptype {ftp | rdp | sftp | smb | ssh | telnet | vnc | web}`	Supported types of bookmarks: ftp, rdp, sftp, smb, ssh, telnet, vnc, web.
`url <url>`	For different types of applications, the URL used to define the destination.
`verify-cert {enable | disable}`	Enable/disable certificate verification of the real server.
`host <name or ip>`	For different types of applications, the host name or IP used to define the destination.
`port <remote port>`	Where applicable, define the port for the service.
`sso {enable | disable}`	Enable/disable the SSO user login and credentials to connect to the application, where applicable.

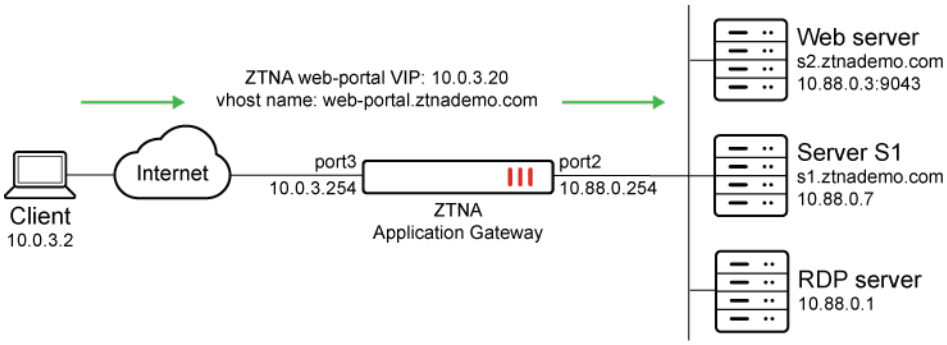
Not all options are listed. Some options are available only for certain types of applications.

To configure your proxy-policy to map to your web-portal:

```
config firewall proxy-policy
    edit <id>
        set proxy ztna-proxy
        set ztna-proxy <web-portal>
        …
    next
end
```

Example

This example demonstrates connecting to a ZTNA web portal to gain access to protected resources. Authentication is performed with LDAPS.



Prerequisites:

- The Client can resolve:
  - The hostname web-portal.ztnademo.com to the VIP address 10.0.3.20
  - The server s1.ztnademo.com to the VIP address 10.0.3.20
  - The server s2.ztnademo.com to the VIP address 10.0.3.20
- The FortiGate resolves:
  - s1.ztnademo.com to the real server IP of 10.88.0.7
  - s2.ztnademo.com to the real server IP of 10.88.0.3
- The FortiGate has an LDAPS connection and user group already created.

To configure from the GUI and CLI:

1. Configure a ZTNA port:
   a. Go to *Policy & Objects* > *ZTNA* >*Ports* and click *Create New*.
   b. Configure the port:

Option	Description
Name	ZTNA-web-proxy
Interface	port3
IP address	10.0.3.20
Port	443
Default certificate	ztna-wildcard

   c. Click *OK*.
   d. Disable client-cert in the CLI:

```
config firewall vip
    edit "ZTNA-web-proxy"
        set client-cert disable
    next
end
```

2. Create a Web Portal object:

a. Go to *Policy & Objects* > *ZTNA* > *Servers & Portals* > *Web Portal* and click *Create New*.
b. Configure the web portal:

Option	Description
Name	ztna-web-portal-ldap
Host	web-portal.ztnademo.com
ZTNA port	ZTNA-web-proxy

c. Enable *Authentication scheme* and, from the drop-down, click the *Create* button.
d. Configure the authentication scheme:

Option	Description
Name	ztna-web-portal-ldap
Method	Basic
User database	Other Select *LDAP-fortiad* from the options.

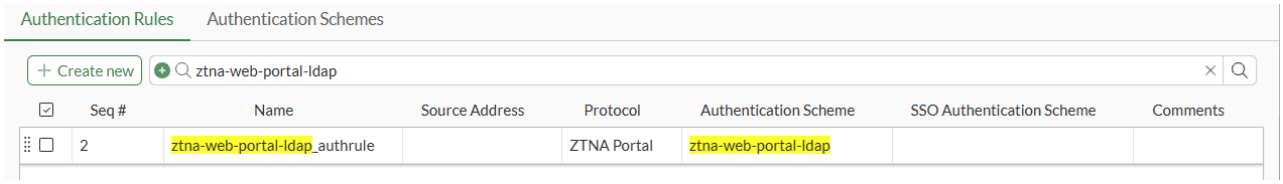
e. Click *OK*.
f. Set *IP-based authentication* to *Enabled* to authenticate users per IP.
g. Click *OK* to save the Web Portal.

3. Update the access-proxy virtual host to disable client-cert:

The configuration here defines the virtual host used to access the ZTNA web portal, as well as the server certificate for the portal. It overrides the settings in the VIP.

```
config firewall access-proxy-virtual-host
    edit "web-portal_vh_mkrbnyua"
        set host "web-portal.ztnademo.com"
        set client-cert disable
    next
end
```

4. Go to *Policy & Objects* > *Authentication* > *Authentication Rules* and note that a new authentication rule was automatically created with the same prefix as the authentication scheme.



5. In the CLI, create web-portal bookmarks that will point to your internal resources:

```
config ztna web-portal-bookmark
    edit "bookmark"
        set groups "LDAP-Remote-Allowed-Group"
        config bookmarks
            edit "Webserver"
                set url "https://s2.ztnademo.com:9043"
```

```
            next
            edit "Server-S1-Web"
                set sso enable
                set url "https://s1.ztnademo.com"
            next
            edit "Server-S1-SSh"
                set apptype ssh
                set sso enable
                set host "10.88.0.7"
                set logon-user "admin"
                set logon-password <password>
            next
            edit "FortiGate-Internal-SSH"
                set apptype ssh
                set host "10.88.0.254"
            next
            edit "RDP"
                set apptype rdp
                set host "10.88.0.1"
                set port 3389
            next
        end
    next
end
```

The groups setting defines the user group allowed access to the portal.

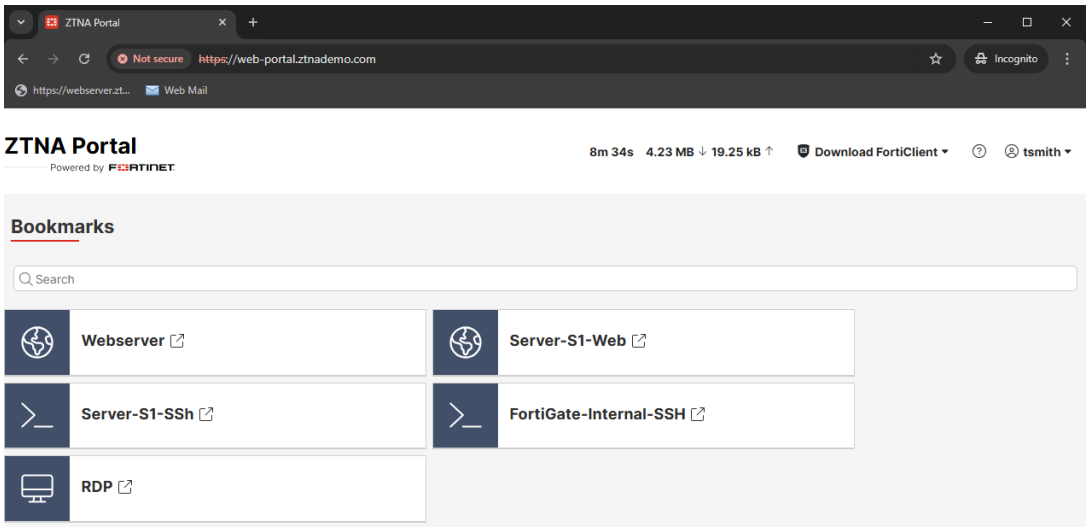
6. Create a full ZTNA policy to allow access to the new VIP:
   a. Go to *Policy & Object > Proxy Policy* and click *Create New*.
   b. Configure the policy:

Option	Description
Name	ZTNA-web-portal
Type	ZTNA
Incoming Interface	port3
Source	all
Destination	all
ZTNA Server	ztna-web-portal-ldap
Schedule	always
Action	Accept
Log Allowed Traffic	All Sessions

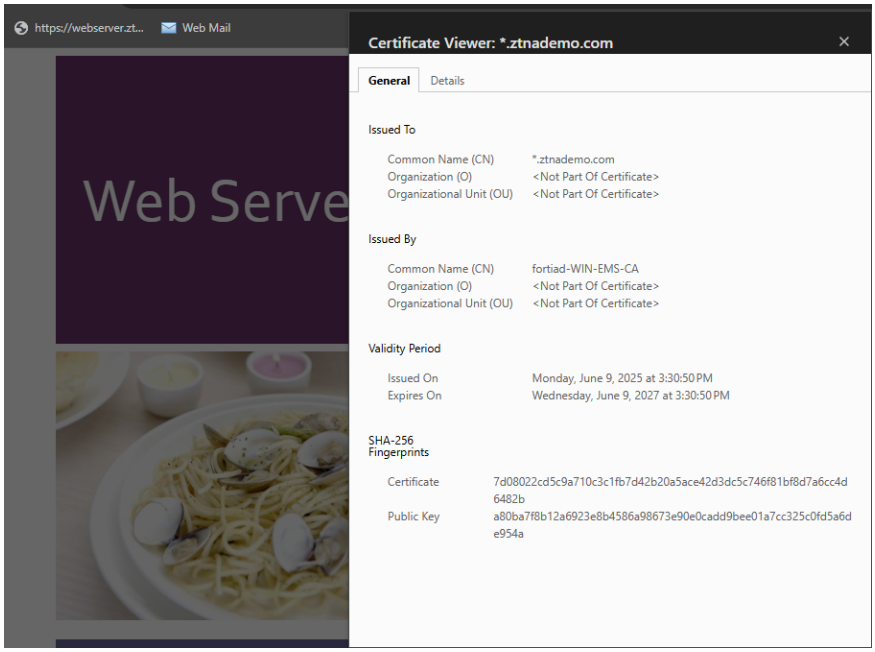
   c. Click *OK*.

Verification:

1. On the Client 10.0.3.2, access the portal on https://web-portal.ztnademo.com.
2. When prompted, enter the LDAP user credentials.
3. Once logged into the ZTNA Portal, choose the application bookmark to connect to.



4. Click *Webserver* to access *s2.ztnademo.com*. A new tab opens up to the web page.
5. Check the certificate to confirm it is signed by the CA certificate used in the VIP configurations.

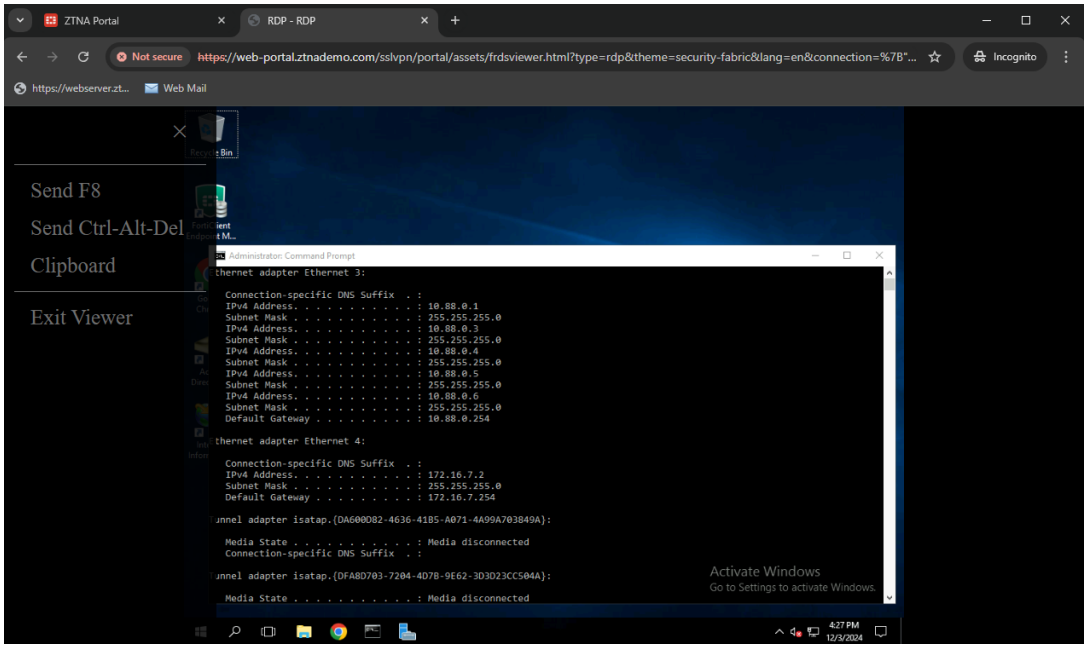


6. On the FortiGate, go to *Log & Report > ZTNA Traffic* to view the latest traffic log. Alternatively, use these commands to view the logs from CLI:

```
# execute log filter field subtype ztna

# execute log display
32 logs found.
```

```
10 logs returned.

1: date=2026-02-18 time=19:30:46 eventtime=1771471846623889083 tz="-0800" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=51677
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.88.0.3
dstport=9043 dstintf="port2" dstintfrole="dmz" sessionid=175737 srcuuid="b458a65a-f759-51ea-
d7df-ef2e750026d1" dstuuid="b458a65a-f759-51ea-d7df-ef2e750026d1" service="tcp/9043"
proxyapptype="ztna-proxy" proto=6 action="accept" policyid=2 policytype="proxy-policy"
poluuid="3d457470-f899-51f0-a91f-9e9a2262cd66" policyname="ZTNA-web-portal" appcat="unscanned"
duration=165 user="tsmith" authserver="LDAP-fortiad" gatewayid=1 vip="ZTNA-web-proxy"
vipincomingip=10.0.3.20 clientdevicemanageable="unknown" clientcert="no" wanin=302808
rcvdbyte=302808 wanout=4720 lanin=10315 sentbyte=10315 lanout=1942557
```

7. On the ZTNA Portal, connect to RDP. When prompted enter the credentials.
8. Once successfully connected, users can press F8 for additional controls.



9. Review the logs:

```
# execute log filter field subtype ztna

# execute log display
35 logs found.
10 logs returned.

1: date=2026-02-18 time=19:36:32 eventtime=1771472192179627219 tz="-0800" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=51692
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.88.0.1
dstport=3389 dstintf="port2" dstintfrole="dmz" sessionid=175964 srcuuid="b458a65a-f759-51ea-
d7df-ef2e750026d1" dstuuid="b458a65a-f759-51ea-d7df-ef2e750026d1" service="RDP"
proxyapptype="ztna-proxy" proto=6 action="accept" policyid=2 policytype="proxy-policy"
poluuid="3d457470-f899-51f0-a91f-9e9a2262cd66" policyname="ZTNA-web-portal" appcat="unscanned"
duration=0 gatewayid=1 vip="ZTNA-web-proxy" vipincomingip=10.0.3.20
```

```
clientdevicemanageable="unknown" clientcert="yes" wanin=0 rcvdbyte=0 wanout=0 lanin=3306
sentbyte=3306 lanout=463
```

Additional troubleshooting and debugs:

If an issue occurs, you can troubleshoot by running these commands:

```
# diagnose wad debug enable category all
# diagnose wad debug enable level verbose
# diagnose debug enable
```

A working connection will output the debugs indicating the web-portal matched the proper gateway. Then accessing the bookmark will output debugs for matching the bookmark.

```
GET /XX/YY/ZZ/webservice?bmgroup=bookmark&bmname=Webserver&cookie=9DC408DF3F32C8805385A1204DED7F81
HTTP/1.1
Host: s2.ztnademo.com
upgrade-insecure-requests: 1
user-agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/131.0.0.0 Safari/537.36
accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,a
pplication/signed-exchange;v=b3;q=0.7
sec-fetch-site: same-site
sec-fetch-mode: navigate
sec-fetch-user: ?1
sec-fetch-dest: document
sec-ch-ua: "Google Chrome";v="131", "Chromium";v="131", "Not_A Brand";v="24"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
referer: https://web-portal.ztnademo.com/
accept-encoding: gzip, deflate, br, zstd
accept-language: en-US,en;q=0.9
cookie: ZTPORTAL-EP-w8ICkTaqA8_EMoSCNDe_gyTAXEUw=430AAAAAAIAAAAAAAAAKF1h1WrrQAbAnodk0tYbtu1vnZZ-
qlaOhL_y-UZIXxPCwqIplM3ujQ==DAAAAFB0iNTviC6zS6nUx9VNZ3yWCbkG
priority: u=0, i

[V][p:3472][s:6905][r:815] wad_http_marker_uri               :1533  path=/XX/YY/ZZ/webservice
len=20
[V][p:3472][s:6905][r:815] wad_http_parse_host              :1911  host_len=15
[I][p:3472][s:6905][r:815] wad_http_parse_host              :1948  host=[15]s2.ztnademo.com
[I][p:3472][s:6905][r:815] wad_http_str_canonicalize        :2468  enc=0
path=/XX/YY/ZZ/webservice len=20 changes=0
[I][p:3472][s:6905][r:815] wad_http_str_canonicalize        :2470  end=5
path=bmgroup=bookmark&bmname=Webserver&cookie=9DC408DF3F32C8805385A1204DED7F81 len=
73 changes=0
[V][p:3472][s:6905][r:815] wad_http_normalize_uri           :2643  host_len=15 path_len=20 query_
len=73
[I][p:3472][s:6905][r:815] wad_http_req_detect_special      :14025 captive_portal detected:
false, preflight=(null)
[I][p:3472][s:6905][r:815] wad_vs_proxy_match_gwy           :4534  16059:ZTNA-web-proxy: matching
gwy with vhost(_def_virtual_host_)
```

```
[V][p:3472][s:6905][r:815] wad_vs_proxy_match_vhost          :4653  16059:ZTNA-web-proxy: matching
vhost by: s2.ztnademo.com
[V][p:3472][s:6905][r:815] wad_vs_proxy_match_vhost          :4656  16059:ZTNA-web-proxy: no host
matched.
[V][p:3472][s:6905][r:815] wad_vs_web_portal_cookie_lookup   :8784  decode cookie_
str=9DC408DF3F32C8805385A1204DED7F81 as cookie_val=1870837449, vd_id=0, vs_
id=16059, gwy_id=1
[I][p:3472][s:6905][r:815] wad_vs_proxy_match_gwy            :4578  16059:ZTNA-web-proxy: Matched
gwy(1) type(ztna-portal) via cookie in query 0x7fdeabcd2448
.
…
[I][p:3472][s:6905][r:815] wad_ztna_portal_match_user        :79    matched cached grp:LDAP-
Remote-Allowed-Group
…
[V][p:3472][s:6905][r:815] wad_http_req_exec_act             :13264 response is ready!
[V][p:3472][s:6905][r:815] wad_http_msg_start_setup_proc     :2280  msg(0x7fdeaa161048) proc-setup
started from: req_resp_ready.
[V][p:3472][s:6905][r:815] wad_http_def_proc_msg_plan        :2242  msg(0x7fdeaa161048) setting up
processor(req_resp_ready)
[V][p:3472][s:6905][r:815] wad_http_msg_start_setup_proc     :2280  msg(0x7fdeaa4f4cf0) proc-setup
started from: resp_forward.
[V][p:3472][s:6905][r:815] wad_http_def_proc_msg_plan        :2242  msg(0x7fdeaa4f4cf0) setting up
processor(resp_forward)
[I][p:3472][s:6905][r:815] wad_dump_fwd_http_resp            :3040  hreq=0x7fdeaa161048 Forward
response from Internal:

HTTP/1.1 303 See Other
X-Content-Type-Options: nosniff
X-Frame-Options: SAMEORIGIN
X-XSS-Protection: 1; mode=block
Strict-Transport-Security: max-age=31536000
Content-Security-Policy: frame-ancestors 'self'; object-src 'self'; script-src 'self' https:
'unsafe-eval' 'unsafe-inline' blob:;
Connection: close
Content-Type: text/html
Cache-Control: no-cache
Location: https://s2.ztnademo.com:443
Content-Length: 133
Set-Cookie: ZTNAWebPortal=9DC408DC3F32C8805385A1204DED7F81; Max-Age=3600; Path=/; HTTPOnly; Secure
Set-Cookie: ZTNAWEB=bmgroup=bookmark&bmname=Webserver; Path=/; HTTPOnly; Secure
```

ZTNA domain matching

A domain filter can be used within ZTNA real server object definitions in the absence of an address, or when you need to match multiple servers in a domain that are not defined in an address or address group. Domain matching is helpful when used with TCP forwarding, where the destinations are configured on the FortiClient but they are not individually configured on the FortiGate ZTNA server configurations. When using domain matching, the destination must be resolvable by the FortiGate.

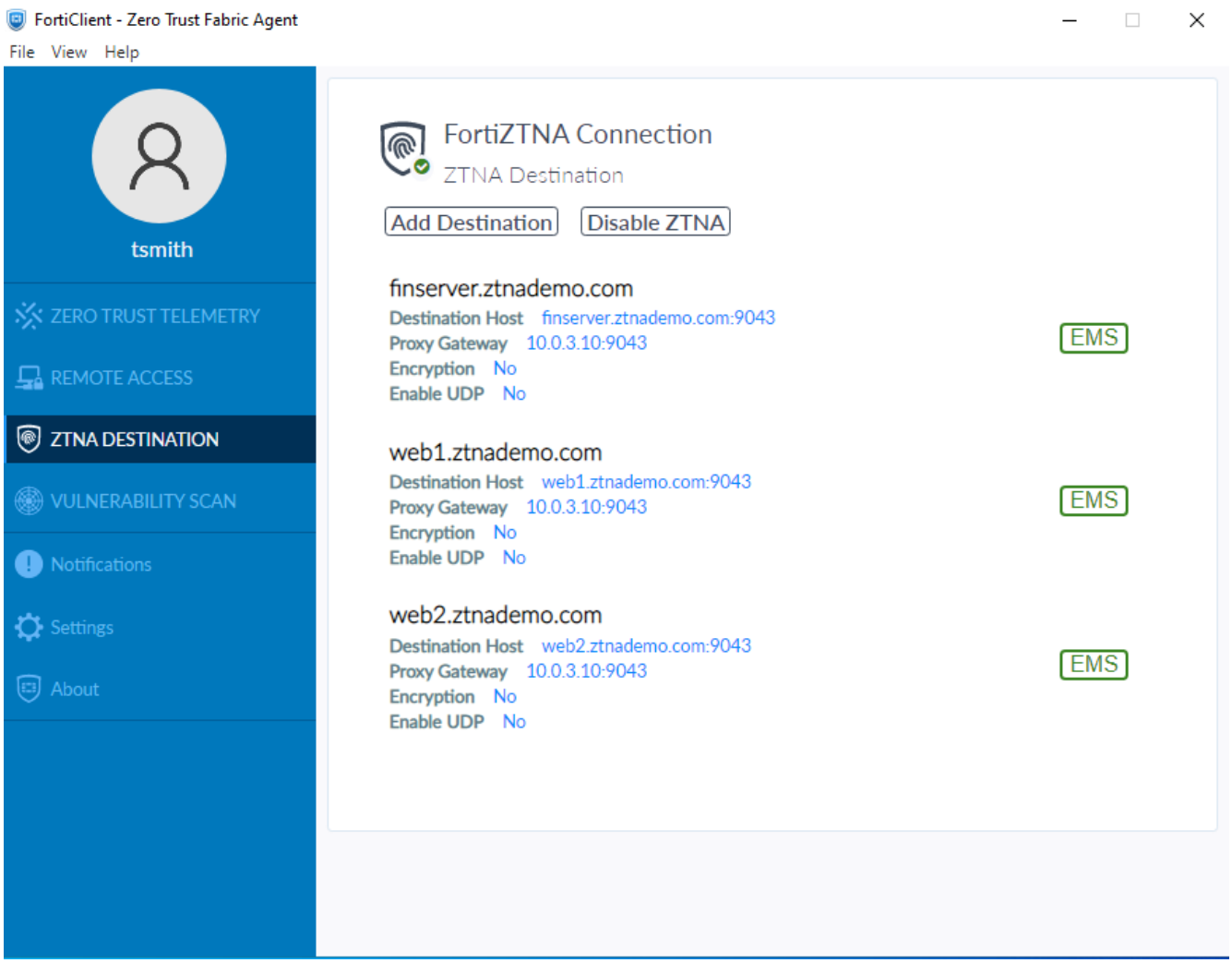
```
config firewall access-proxy
    edit <name>
        config api-gateway
            edit <ID>
                config realservers
                    edit 1
                        set domain <domain>
                    next
                end
            next
        end
    next
end
```

Where `<domain>` can be a substring of the actual domain. For example, `st.com` will match `test.com` or `first.com`, but will not match `trait.com`.

Example 1: Wildcard domain match ztnademo.com

In this example, several TCP forwarding entries have been added on the EMS server and pushed to the FortiClient endpoint:

- finserver.ztnademo.com
- web1.ztnademo.com
- web2.ztnademo.com



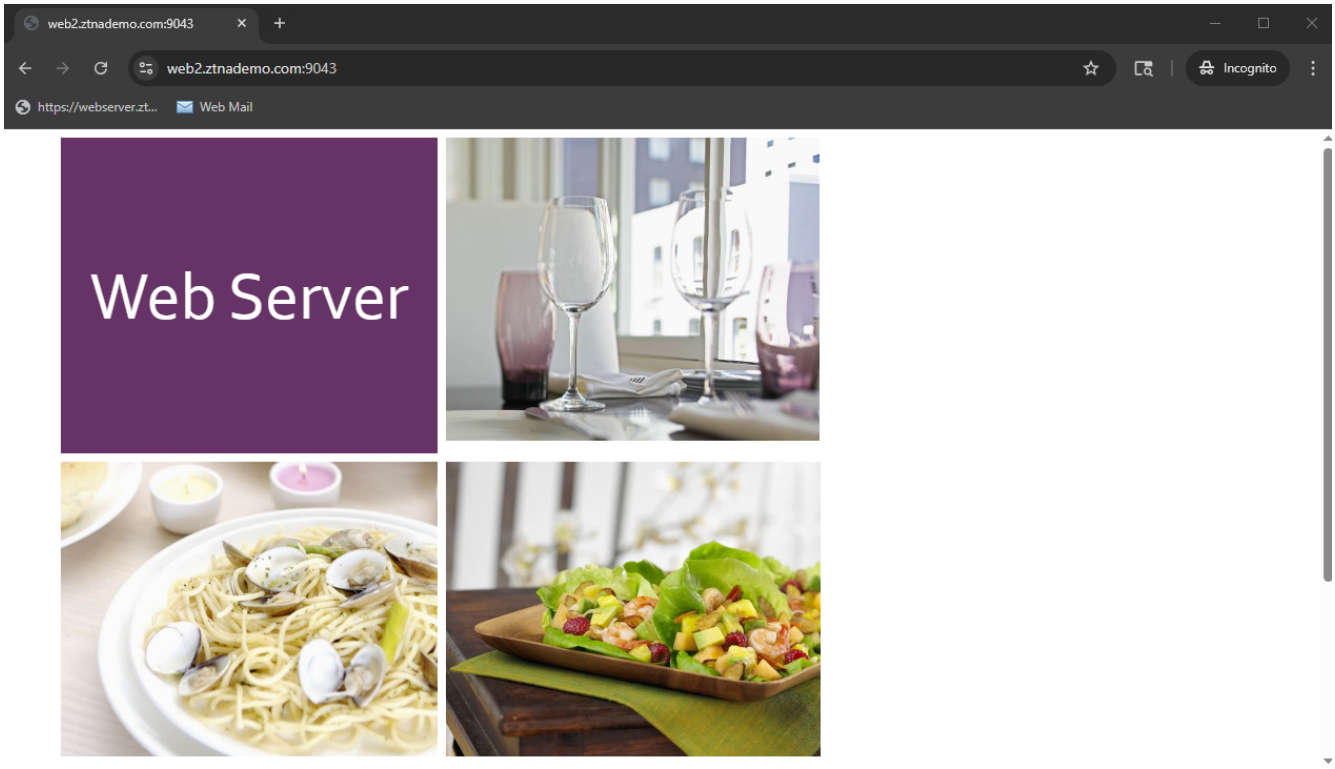
All 3 servers are resolvable on the FortiGate.

On the ZTNA real server definition on the FortiGate, a wildcard domain filter is used to match everything under `ztnademo.com`. No address object is defined.

```
config firewall access-proxy
    edit "ZTNA-server"
        set vip "ZTNA-server"
        config api-gateway
            edit 1
                set url-map "/tcp"
                set service tcp-forwarding
                config realservers
                    edit 1
                        set domain "ztnademo.com"
                        set mappedport 9043
                    next
                end
            next
        end
```

```
    next
end
```

When accessing these servers from the remote endpoint, each of these can be accessed.




```
# execute log filter field subtype ztna
# execute log display
164 logs found.
10 logs returned.

1: date=2025-07-29 time=15:37:59 eventtime=1753828678973202823 tz="-0700" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=62031
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.0.3.10
dstport=9043 dstintf="port2" dstintfrole="dmz" sessionid=15859 srcuuid="b458a65a-f759-51ea-d7df-
ef2e750026d1" service="tcp/9043" proxyapptype="ztna-proxy" proto=6 action="accept" policyid=17
policytype="policy" poluuid="8caee736-6cc5-51f0-837b-5bc10419a710" policyname="ZTNA-traffic"
trandisp="dnat" tranip=10.88.0.4 tranport=9043 appcat="unscanned" duration=87 gatewayid=1
vip="ZTNA-server" accessproxy="ZTNA-server" clientdevicemanageable="manageable" clientcert="yes"
wanin=303035 rcvdbyte=303035 wanout=2964 lanin=4943 sentbyte=4943 lanout=306164

2: date=2025-07-29 time=15:35:20 eventtime=1753828520045168898 tz="-0700" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=62016
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.0.3.10
dstport=9043 dstintf="port2" dstintfrole="dmz" sessionid=15807 srcuuid="b458a65a-f759-51ea-d7df-
ef2e750026d1" service="tcp/9043" proxyapptype="ztna-proxy" proto=6 action="accept" policyid=17
policytype="policy" poluuid="8caee736-6cc5-51f0-837b-5bc10419a710" policyname="ZTNA-traffic"
trandisp="dnat" tranip=10.88.0.5 tranport=9043 appcat="unscanned" duration=21 gatewayid=1
vip="ZTNA-server" accessproxy="ZTNA-server" clientdevicemanageable="manageable" clientcert="yes"
```

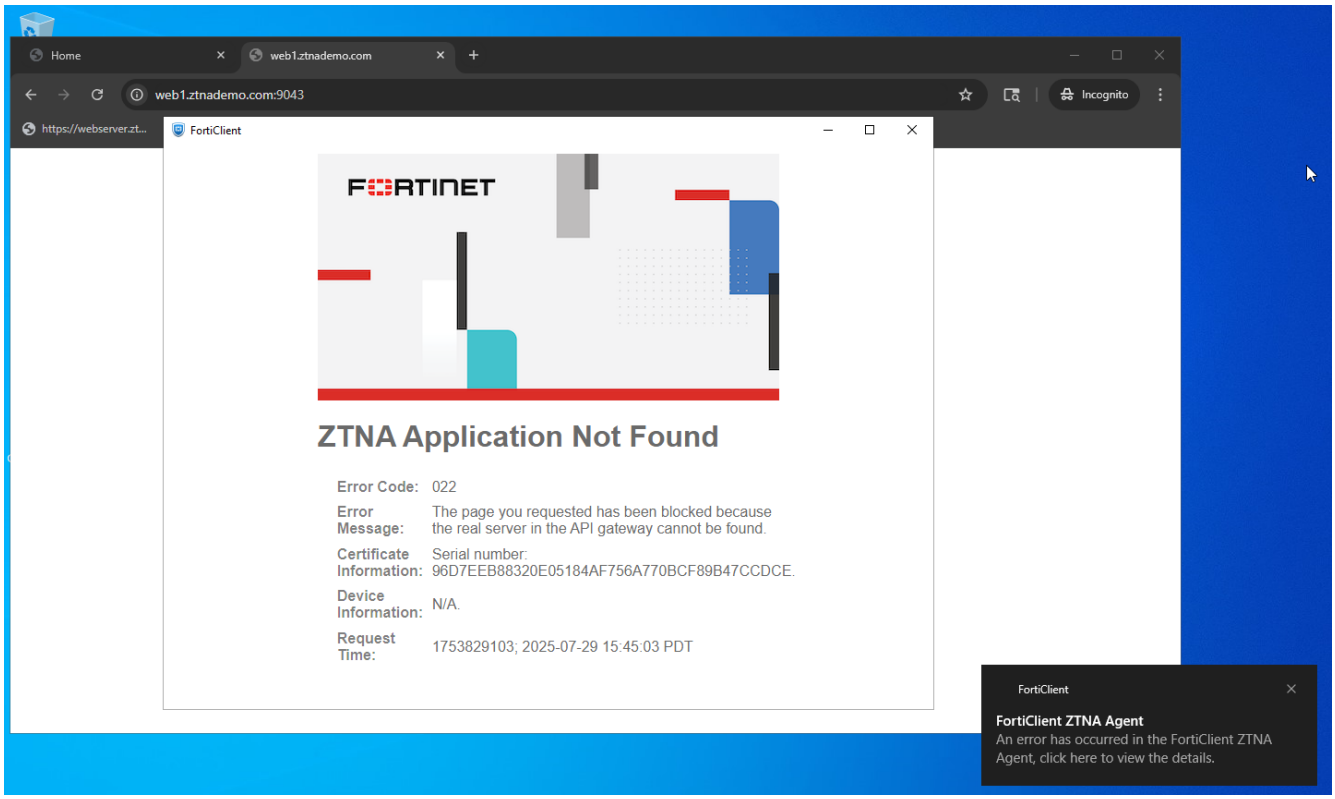
```
wanin=31581 rcvdbyte=31581 wanout=3699 lanin=5683 sentbyte=5683 lanout=34710

3: date=2025-07-29 time=15:35:20 eventtime=1753828520046436756 tz="-0700" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=62019
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.0.3.10
dstport=9043 dstintf="port2" dstintfrole="dmz" sessionid=15812 srcuuid="b458a65a-f759-51ea-d7df-
ef2e750026d1" service="tcp/9043" proxyapptype="ztna-proxy" proto=6 action="accept" policyid=17
policytype="policy" poluuid="8caee736-6cc5-51f0-837b-5bc10419a710" policyname="ZTNA-traffic"
trandisp="dnat" tranip=10.88.0.3 tranport=9043 appcat="unscanned" duration=6 gatewayid=1
vip="ZTNA-server" accessproxy="ZTNA-server" clientdevicemanageable="manageable" clientcert="yes"
wanin=302887 rcvdbyte=302887 wanout=2896 lanin=4875 sentbyte=4875 lanout=306016
```

Example 2: Wildcard domain match server.ztnademo.com

In this example, the same servers are pushed to the FortiClient endpoint. However, a more specific domain filter is applied to only allow `*server.ztnademo.com`.

```
config firewall access-proxy
    edit "ZTNA-server"
        set vip "ZTNA-server"
        config api-gateway
            edit 1
                set url-map "/tcp"
                set service tcp-forwarding
                config realservers
                    edit 1
                        set domain "server.ztnademo.com"
                        set mappedport 9043
                    next
                end
            next
        end
    next
end
```

When accessing these servers, only `finserver.ztnademo.com` can be accessed. Other servers will return with an error.




```
# execute log filter field subtype ztna
# execute log display
189 logs found.
10 logs returned.

1: date=2025-07-29 time=15:46:20 eventtime=1753829180317849279 tz="-0700" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=62060
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.0.3.10
dstport=9043 dstintf="port2" dstintfrole="dmz" sessionid=15987 srcuuid="b458a65a-f759-51ea-d7df-
ef2e750026d1" service="tcp/9043" proxyapptype="ztna-proxy" proto=6 action="accept" policyid=17
policytype="policy" poluuid="8caee736-6cc5-51f0-837b-5bc10419a710" policyname="ZTNA-traffic"
trandisp="dnat" tranip=10.88.0.5 tranport=9043 appcat="unscanned" duration=95 gatewayid=1
vip="ZTNA-server" accessproxy="ZTNA-server" clientdevicemanageable="manageable" clientcert="yes"
wanin=31581 rcvdbyte=31581 wanout=3624 lanin=5608 sentbyte=5608 lanout=34710

2: date=2025-07-29 time=15:45:56 eventtime=1753829155650041262 tz="-0700" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=62122
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.0.3.10
dstport=9043 dstintf="root" dstintfrole="undefined" sessionid=16026 srcuuid="b458a65a-f759-51ea-
d7df-ef2e750026d1" service="tcp/9043" proxyapptype="http" proto=6 action="deny" policyid=17
policytype="policy" poluuid="8caee736-6cc5-51f0-837b-5bc10419a710" policyname="ZTNA-traffic"
trandisp="dnat" tranip=10.0.3.10 tranport=9043 appcat="unscanned" duration=0 vip="ZTNA-server"
accessproxy="ZTNA-server" clientdevicemanageable="manageable" clientcert="yes" msg="Traffic denied
because failed to find a server: reason: Cannot find the real server in the API gateway.,
hostname: 10.0.3.10" wanin=0 rcvdbyte=0 wanout=0 lanin=1963 sentbyte=1963 lanout=3014 crscore=30
craction=131072 crlevel="high"
```

Example 3: Domain match precedence

In this example, an address group is specified in the real server definition, which includes all three server's FQDN. A domain filter is configured to only filter on web.

```
config firewall addrgrp
    edit "ztnademo-group"
        set member "web1.ztnademo.com" "web2.ztnademo.com" "finserver.ztnademo.com"
    next
end
```

```
config firewall access-proxy
    edit "ZTNA-server"
        set vip "ZTNA-server"
        config api-gateway
            edit 1
                set url-map "/tcp"
                set service tcp-forwarding
                config realservers
                    edit 1
                        set address "ztnademo-group"
                        set domain "web"
                        set mappedport 9043
                    next
                end
            next
        end
    next
end
```

When accessing web1.ztnademo.com or web2.ztnademo.com, wad debug shows that domain matching takes precedence over address matching:

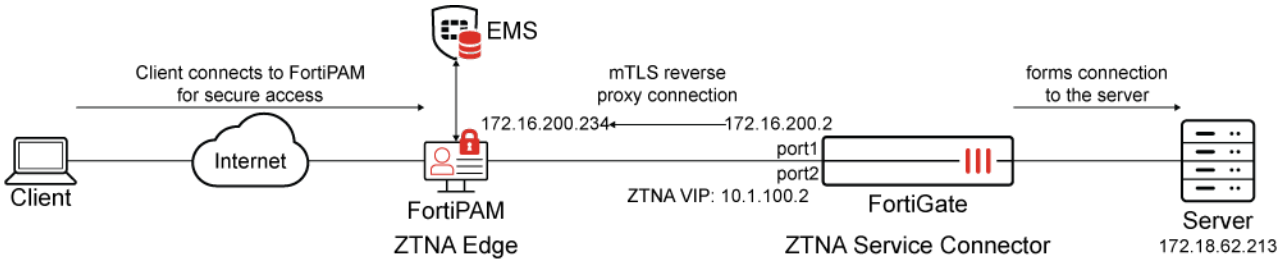
```
# diagnose wad debug enable category all
Debug messages will be on for 30 minutes.
# diagnose wad debug enable level verbose
Debug messages will be on for 30 minutes.
# diagnose debug enable

GET /tcp?address=web1.ztnademo.com&port=9043&tls=0 HTTP/1.1
Host: 10.0.3.10:9043
User-Agent: Forticlient
Accept: */*
Authorization:
Connection: Upgrade
Cookie:
Upgrade: tcp-forwarding/1.0
X-Fos-Auth:

…
[I][p:2195][s:1334][r:10] wad_vs_gwy_tcp_dst_ovrd           :3392  16363:ZTNA-server:1: req
(0x7f4122a79048) found real server by matching domain(web).
```

```
[W][p:2195][s:1334][r:10] wad_vs_proxy_dns_resolve          :3280  req(0x7f4122a79048) vs DNS
request name=web1.ztnademo.com len=17 type/pref=0/0
[I][p:2195][s:1334][r:10] __wad_dns_send_query              :863   0:0: sending DNS request for
remote peer web1.ztnademo.com id=0 IPv4
…
[I][p:2195][s:1334][r:10] wad_vs_proxy_dns_request_done     :3229  req(0x7f4122a79048) vs DNS
resolved: 10.88.0.3
…

The following debug indicates DNS was able to resolve the requested destination, and domain
matching was successful. Address matching was not attempted.

[I][p:2195][s:1334][r:10] wad_http_req_exec_on_vs_dns_ready :12216 req(0x7f4122a79048) vs DNS
ready: dns_resolved(1), domain_matched(1), addr_matched(0)
[V][p:2195][s:1334][r:10] wad_get_dst_intf_idx2             :174   Matched route cache.local=0,
oif=4
[V][p:2195][s:1334][r:10] wad_http_req_get_dst_intf         :11982 vd=0 dst=10.88.0.3 ifidx=4

On the other hand, when accessing finserver.ztnademo.com, domain matching is unsuccessful, but
address matching was successful.

GET /tcp?address=finserver.ztnademo.com&port=9043&tls=0 HTTP/1.1
Host: 10.0.3.10:9043
User-Agent: Forticlient
Accept: */*
Authorization:
Connection: Upgrade
Cookie:
Upgrade: tcp-forwarding/1.0
X-Fos-Auth:

…
[I][p:2195][s:1347][r:11] wad_http_req_exec_on_vs_dns_ready :12216 req(0x7f4122a79048) vs DNS
ready: dns_resolved(1), domain_matched(0), addr_matched(1)
[V][p:2195][s:1347][r:11] wad_get_dst_intf_idx2             :174   Matched route cache.local=0,
oif=4
[V][p:2195][s:1347][r:11] wad_http_req_get_dst_intf         :11982 vd=0 dst=10.88.0.5 ifidx=4
```

ZTNA service connector NEW

In certain environments, we may want to prevent users from accessing the FortiGate ZTNA access proxy directly or expose ports and services on the FortiGate to the public. In these scenarios, another layer of ZTNA proxy may be required to serve as the ZTNA Edge.

The FortiGate can act as a ZTNA service connector to reverse proxy service connections upstream to the ZTNA Edge. In this scenario, the ZTNA Edge may be a FortiPAM or FortiProxy which acts as the ZTNA access proxy to the end users. The FortiGate, as the ZTNA service connector, forms a persistent control connection with ZTNA Edge. The ZTNA Edge forwards connection requests through the control tunnel to the FortiGate ZTNA service connector, which proxies the request to the protected server.



The ZTNA Edge in this scenario completes the ZTNA checks and enforces the ZTNA policy. If the connection is allowed, the request is forwarded to the ZTNA service connector.

The ZTNA service connector is responsible for proxying the connection and establishing a data connection back to the ZTNA Edge to forward the response. This effectively forms a proxy chain where the FortiGate service connector provides reverse proxy connection back to the ZTNA Edge.

Usage

CLI syntax

The following CLI settings configure the connection with the ZTNA Edge:

```
config ztna reverse-connector
    edit <name>
        set address <IP or FQDN address>
        set port <port number>
        set certificate <local certificate>
        set trusted-server-ca <ca certificate>
        set source-ip IP address <ip address>
        set source-ip-interface interface <interface>
        set interface-select-method {auto | sdwan | specify}
        set vrf-select <integer>
    next
end
```

Field	Description
address <IP or FQDN address>	Address of the ZTNA Edge
port <port number>	The port number for connecting to the ZTNA Edge
certificate <local certificate>	The local client certificate used for mTLS negotiation
trusted-server-ca <ca certificate>	The root CA certificate to authenticate the mTLS connection
source-ip <ip address>	FortiGate IPv4 address to be used for ZTNA reverse-connector connection.
source-ip-interface <interface>	Source interface to be used for ZTNA reverse-connector connection.
interface-select-method {auto | sdwan | specify}	Specify how to select outgoing interface to reach server.
vrf-select <integer>	VRF ID used for connection to server (0 - 511).

GUI settings

1. Go to *System* > *Feature Visibility* and enable *Zero Trust Network Access* and *ZTNA Reverse Proxy Connector*.
2. Go to *Policy & Objects* > *ZTNA* and select the *Traffic Forwarding* tab.
3. Select the *Reverse Proxy Connector* sub-menu and click *Create New*.
4. Configure the following, then click *OK*:

Field	Description
Name	Name of the connector
Address	IP or FQDN Address of the ZTNA Edge
Port	The port number for connecting to the ZTNA Edge
CA certificate	The root CA certificate to authenticate the mTLS connection
Default certificate	The local client certificate used for mTLS negotiation

Example

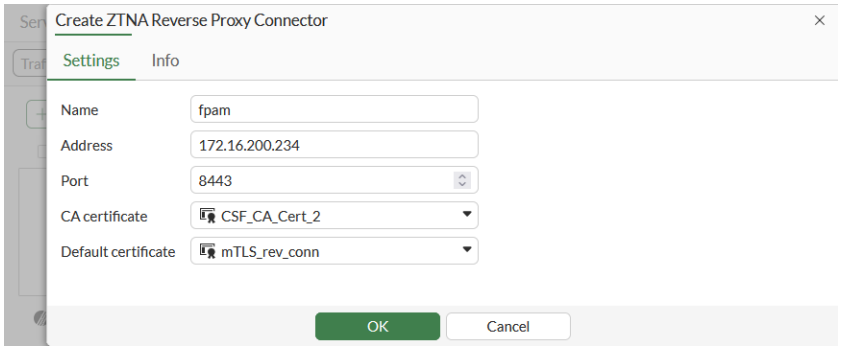
In the example topology shown in the introduction, we will configure a TCP forwarding destination on the FortiGate service connector to allow RDP connection to the Server (172.18.62.213). We will also configure a ZTNA reverse-connector to connect to the upstream FortiPAM.

The FortiPAM will follow the FortiPAM server (ZTNA Edge) configuration as illustrated in the Configuring FortiPAM/FortiGate as reverse gateway example. We will configure a secret on the FortiPAM server to allow users to RDP into the Server.

Alternatively, if deploying using a FortiProxy on the ZTNA Edge, follow the FortiProxy as reverse proxy to service connector example in the FortiProxy admin guide.

To configuring the reverse proxy connection in the FortiOS GUI:

1. Go to *System* > *Feature Visibility* and enable *Zero Trust Network Access* and *ZTNA Reverse Proxy Connector*.
2. Go to *Policy & Objects* > *ZTNA* and select the *Traffic Forwarding* tab.
3. Select the *Reverse Proxy Connector* sub-menu and click *Create New*.
4. Configure the following:



Field	Description
Name	Fpam
Address	172.16.200.234
Port	8443
CA certificate	CSF_CA_Cert_2
Default certificate	mTLS_rev_conn

5. Click *OK*

To configure the ZTNA reverse-connector in the FortiOS CLI:

```
config ztna reverse-connector
    edit "fpam"
        set address "172.16.200.234"
        set port 8443
        set certificate "mTLS_rev_conn"
        set trusted-server-ca "CSF_CA_Cert__2"
    next
end
```

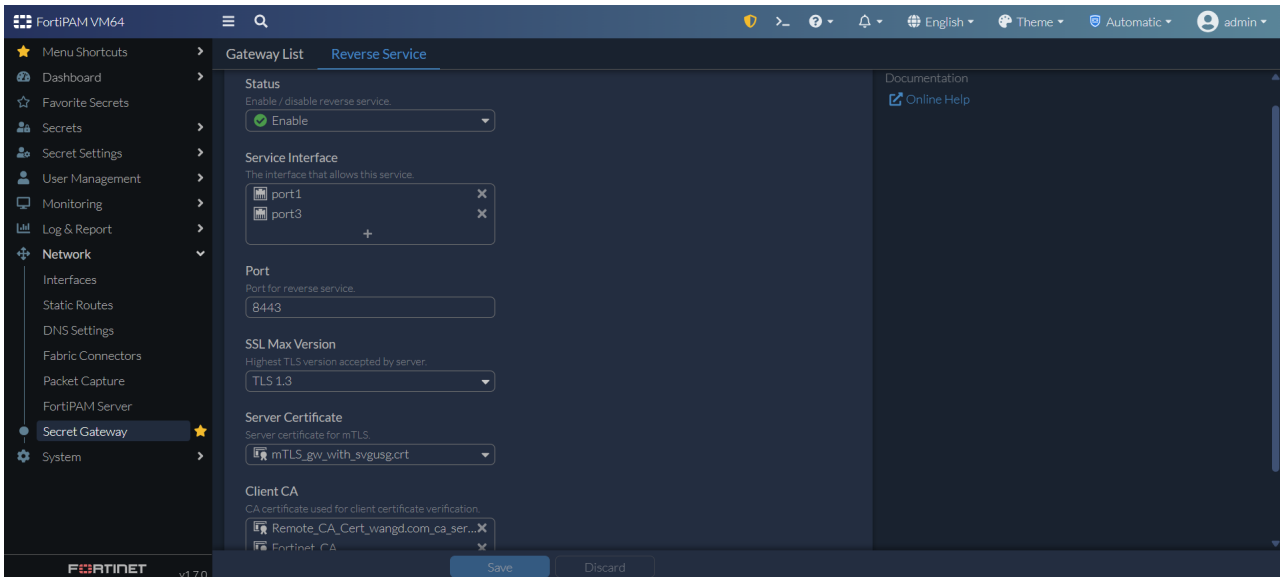
To configure the TCP forwarding proxy, destination, and rules in the FortiOS CLI:

```
config firewall vip
    edit "vip"
        set type access-proxy
        set server-type https
        set extip 10.1.100.2
        set extintf "port2"
        set client-cert disable
        set extport 4443
        set ssl-certificate "Fortinet_SSL"
    next
end
config ztna traffic-forward-proxy
    edit "ztfp"
        set vip "vip"
    next
end
config firewall address
    edit "to_server_213"
        set subnet 172.18.62.213 255.255.255.255
    next
end
config ztna destination
    edit "RDP_213"
        set address "to_server_213"
        set mappedport 3389
    next
```

```
end
config firewall proxy-policy
    edit 1
        set proxy ztna-proxy
        set ztna-proxy "ztfp"
        set ztna-destination "RDP_213"
        set srcintf "any"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set logtraffic all
        set srcaddr6 "all"
        set dstaddr6 "all"
        set utm-status enable
        set profile-protocol-options "protocol"
    next
end
```

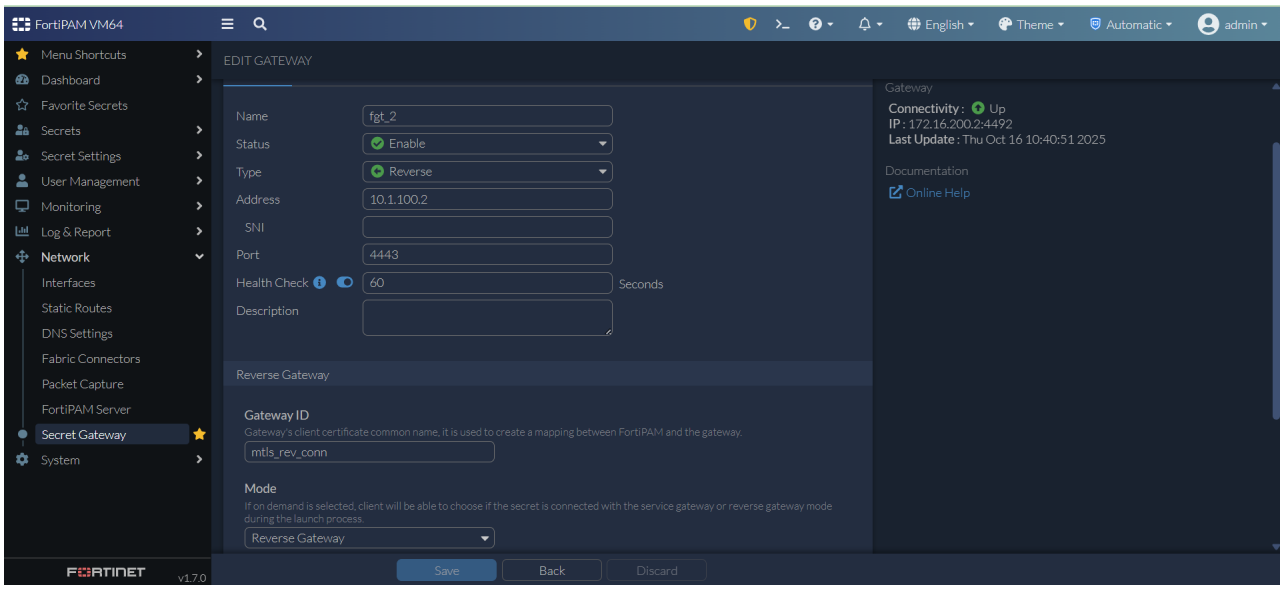
To configure the FortiPAM ZTNA Edge:

1. On the FortiPAM, go to *Network* > *Secret Gateway* > *Reverse Service*.
2. Enable the service and enter the server settings.

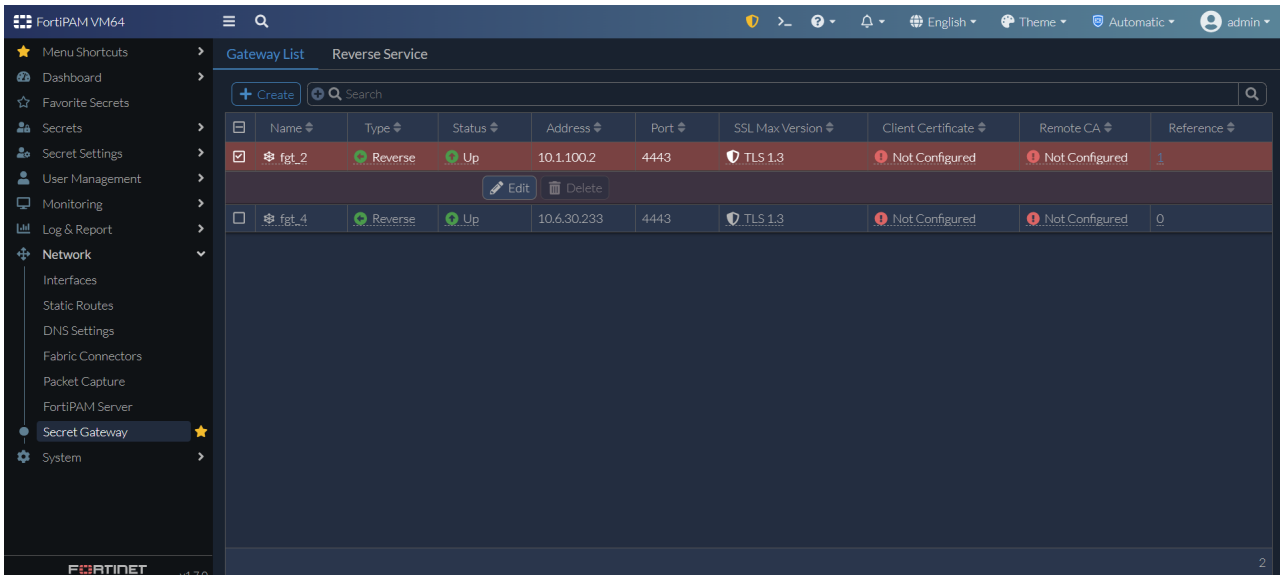


3. Click *Save*.
4. Under *Network* > *Secret Gateway*, create a new *Gateway*.
5. Enter the settings below and input the FortiGate VIP address to identify the desired gateway.

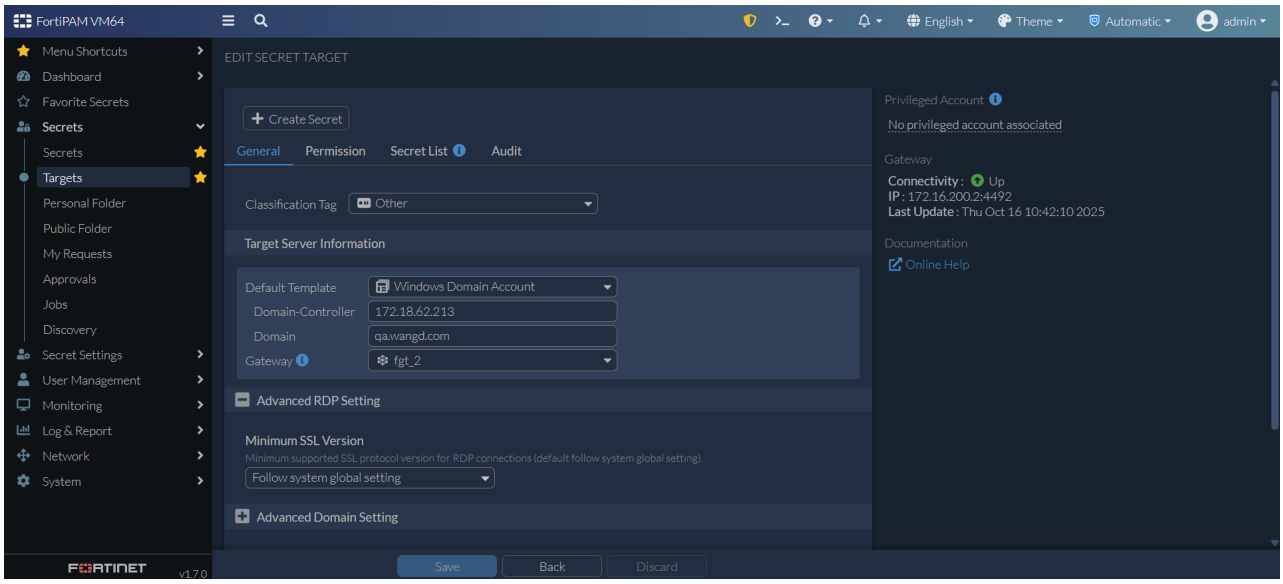
   For the *Gateway ID*, enter the CN string from the FortiGate client certificate (mTLS_rev_conn in this example).



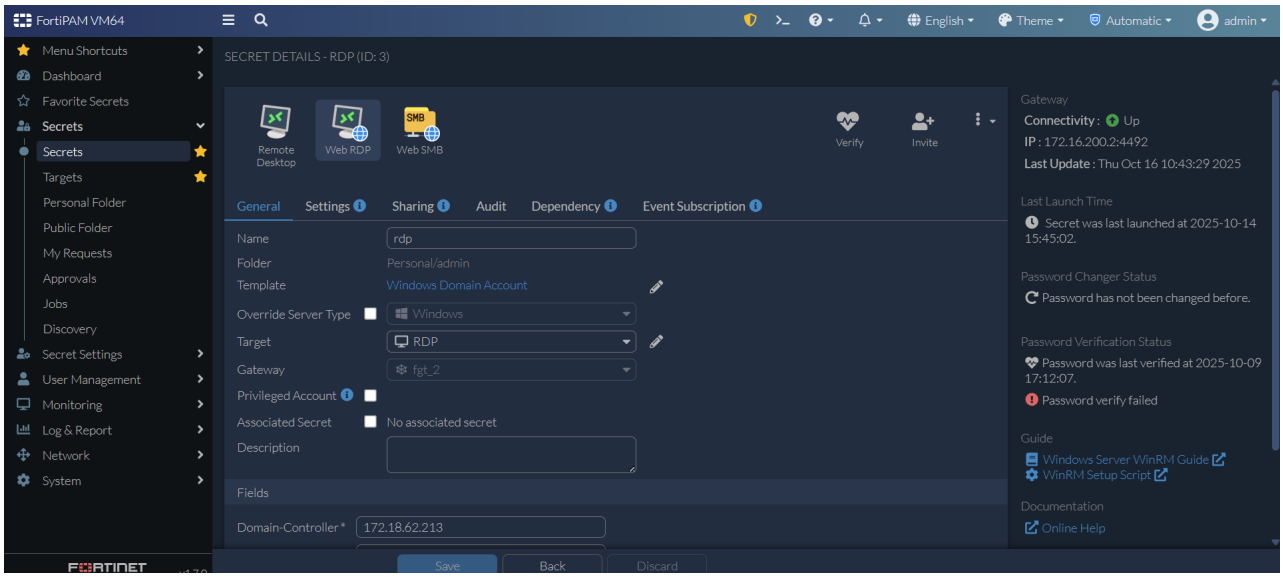
6. Save and view the *Gateway* from the *Gateway List*.



7. Go to *Secrets* > *Targets* and define a *Target* using this *Gateway*. Enter the address of the *Server* as the *Domain-Controller* and select the *Gateway* that was created earlier.



8. Go to *Secrets* > *Secrets* and define a *Secret* using this *Target*.

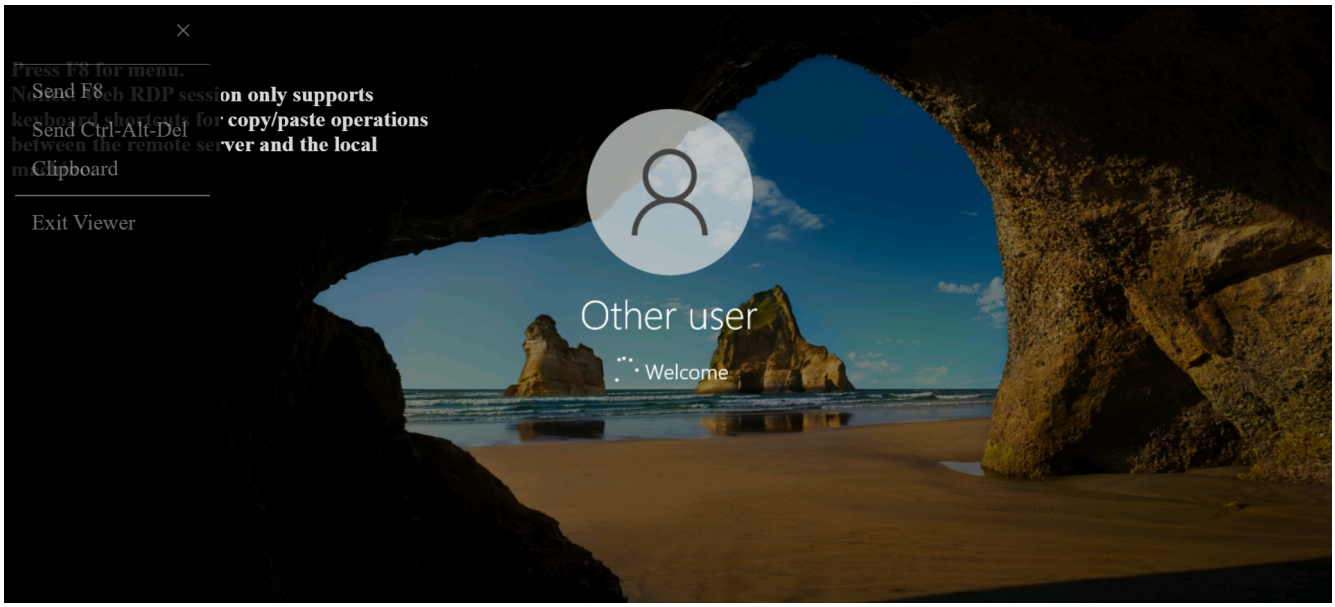


9. Save the settings.

To access the server through RDP:

On the Windows Client, you can login to the FortiPAM as a user, then access a secret from the FortiPAM portal. The client itself will not directly connect to the server itself. In fact, the FortiPAM will not initiate a connection to the server or the ZTNA service connector. It is the FortiGate ZTNA service connector which initiates the control channel and data channel to the FortiPAM to facilicate the communications.

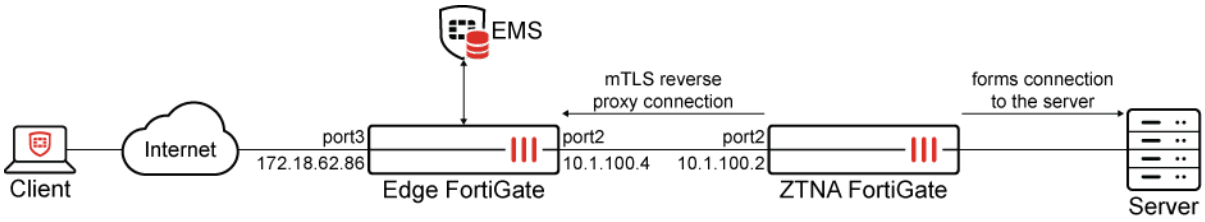
On the client, use the Web RDP button from the FortiPAM GUI to RDP to the server.



ZTNA connector - reverse gateway and forwarder NEW

Continuing on from the ZTNA service connector NEW example, a FortiGate can be used as the ZTNA edge device in place of the FortiPAM. When acting as the edge device, the FortiGate may act as a reverse gateway just as the FortiPAM did, or it may function as a ZTNA query forwarder, directly forward client's ZTNA query to next hop.

FortiGate as a reverse gateway



To configure the ZTNA Edge FortiGate as a reverse gateway:

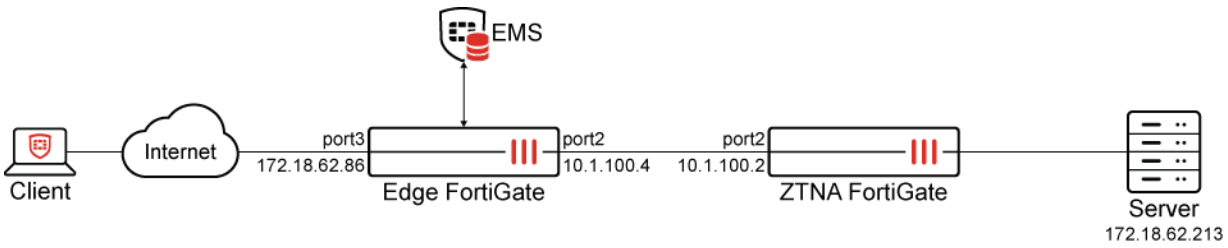
```
config ztna connector-edge
    set status enable
    set interface "port2"
    set port 8443
    set server-cert "mTLS_rev_conn"
    set trusted-client-ca "CSF_CA_Cert_7edfcd73755d4c909d8057433329ef2b_1"
end
```

```
config ztna service-connector
    edit "svc-conn"
        set connection-mode reverse
        set forward-address "10.1.100.2"
        set forward-port 4443
        set certificate "Fortinet_Factory"
        set trusted-ca "Fortinet_Sub_CA"
        set relay-dev-info enable
        set relay-user-info enable
        set health-check-interval 300
        set forward-destination-cn "mtls_gw"
    next
end
```

```
config ztna traffic-forward-proxy
    edit "ztfp"
        set vip "ztna2"
        set host "qa.ftnttest.com"
        config url-route
            edit "route"
                set service-connector "svc-conn"
            next
        end
    next
end
```

```
config firewall vip
    edit "ztna2"
        set vip-id 1
        set type access-proxy
        set server-type https
        set extip 172.18.62.86
        set extintf "port3"
        set extport 4443
        set ssl-certificate "Fortinet_SSL"
    next
end
```

FortiGate as a forwarder



To configure the ZTNA Edge FortiGate as a ZTNA forwarder:

```
config ztna service-connector
    edit "fwd-svc"
        set forward-address "10.1.100.2"
        set forward-port 4443
        set certificate "Fortinet_Factory"
        set trusted-ca "Fortinet_Sub_CA"
        set relay-dev-info enable
    next
end
```

```
config ztna traffic-forward-proxy
    edit "ztfp"
        set vip "ztna2"
        set host "qa.ftnttest.com"
        config url-route
            edit "route"
                set service-connector "fwd-svc"
            next
        end
    next
end
```

```
config firewall vip
    edit "ztna2"
        set vip-id 1
        set type access-proxy
        set server-type https
        set extip 172.18.62.86
        set extintf "port3"
        set extport 4443
        set ssl-certificate "Fortinet_SSL"
    next
end
```

ZTNA configuration examples

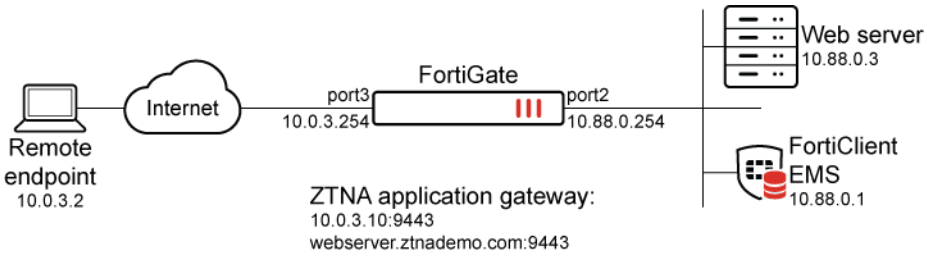
This section includes the following ZTNA configuration examples:

- ZTNA web proxy example on page 1677
- ZTNA web proxy with load balancing example on page 1691
- ZTNA web proxy with basic authentication example on page 1696
- ZTNA traffic forwarding proxy example on page 1708
- ZTNA traffic forwarding proxy with FQDN example on page 1719
- ZTNA SSH traffic forwarding proxy example on page 1726
- ZTNA application gateway with SAML authentication example on page 1735

- ZTNA application gateway with SAML and MFA using FortiAuthenticator example on page 1750
- Secure LDAP connection from FortiAuthenticator with zero trust tunnel example on page 1776
- ZTNA IP MAC based access control example on page 1776
- ZTNA IPv6 examples on page 1788
- ZTNA Zero Trust application gateway example on page 1795
- ZTNA SaaS application access control with Inline CASB on page 1795
- ZTNA application gateway with KDC to access shared drives on page 1801
- Custom replacement message for ZTNA virtual hosts on page 1808
- ZTNA single sign-on with Entra ID on page 1811

ZTNA web proxy example

In this example, a web proxy is configured to demonstrate its function as a reverse proxy on behalf of the web server it is protecting. It verifies user identity, device identity, and trust context, before granting access to the protected source.



This example shows access control that allows or denies traffic based on security posture tags. Traffic is allowed when the FortiClient endpoint is tagged as *Low Importance* using Classification tags, and denied when the endpoint is tagged with *Malicious-File-Detected*.

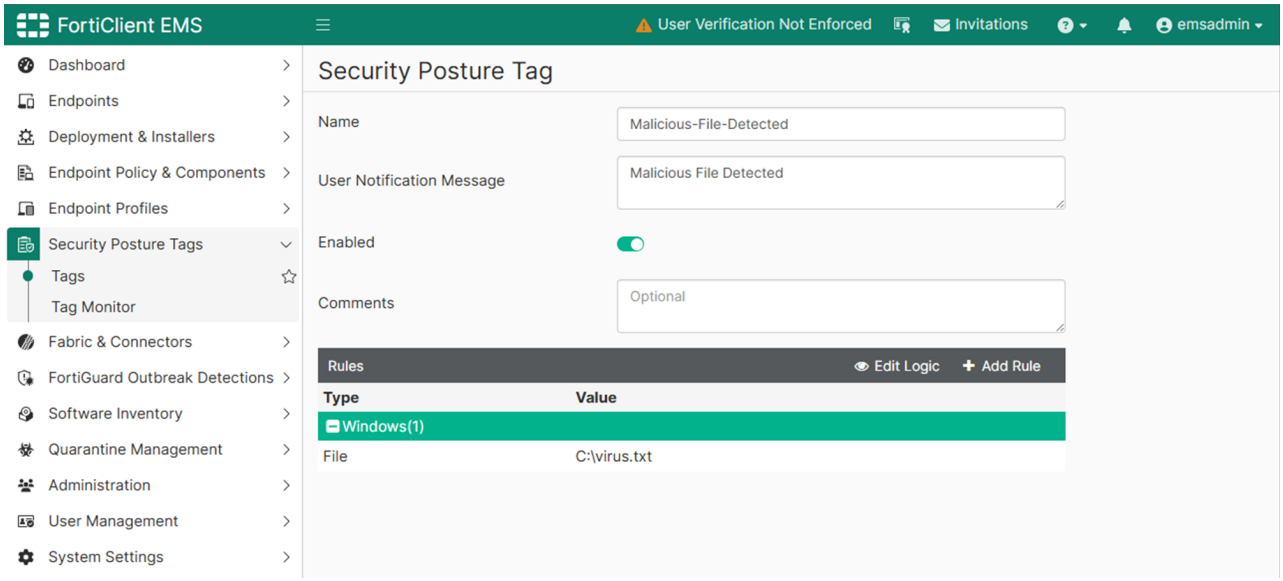
This example uses EMS 7.4.5, and assumes that the FortiGate EMS fabric connector is already successfully connected.

To configure ZTNA in the GUI, go to *System > Feature Visibility* and enable *Zero Trust Network Access*.

To configure a Zero Trust tagging rule on the FortiClient EMS:

1. Log in to the FortiClient EMS.
2. Go to *Security Posture Tags > Tags*, and click *Create*.
3. In the *Name* field, enter *Malicious-File-Detected*.
4. In the *Tag Endpoint As* dropdown list, select *Malicious-File-Detected*.
   EMS uses this tag to dynamically group together endpoints that satisfy the rule, as well as any other rules that are configured to use this tag.
5. Specify a *User Notification Message*, such as *Malicious File Detected*.
6. Click *Add Rule* then configure the rule:

a. For *OS*, select *Windows*.
b. From the *Rule Type* dropdown list, select *File*.
c. Enter a file name, such as *C:\virus.txt*.
d. Click *Save*.

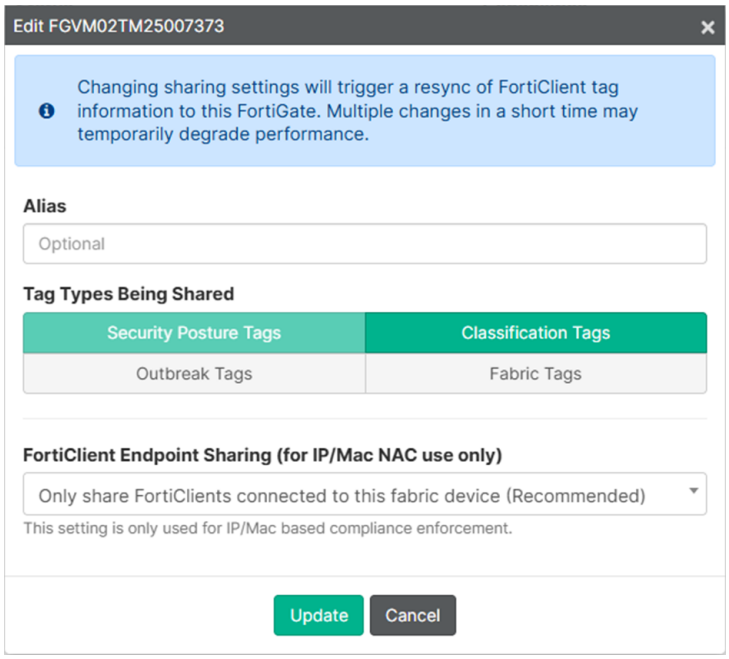


7. Click *Save*.

To configure FortiClient EMS to share classification tags:

This is required for the Low Importance tag to be sent to the FortiGate.

1. Go to *Fabric & Connectors* > *Fabric Devices*.
2. Expand *Standalone devices* and hover over the FortiGate device to select the box with a pencil icon.
3. Click *Edit*.
4. Under *Tag Types Being Shared*, add *Classification Tags*.



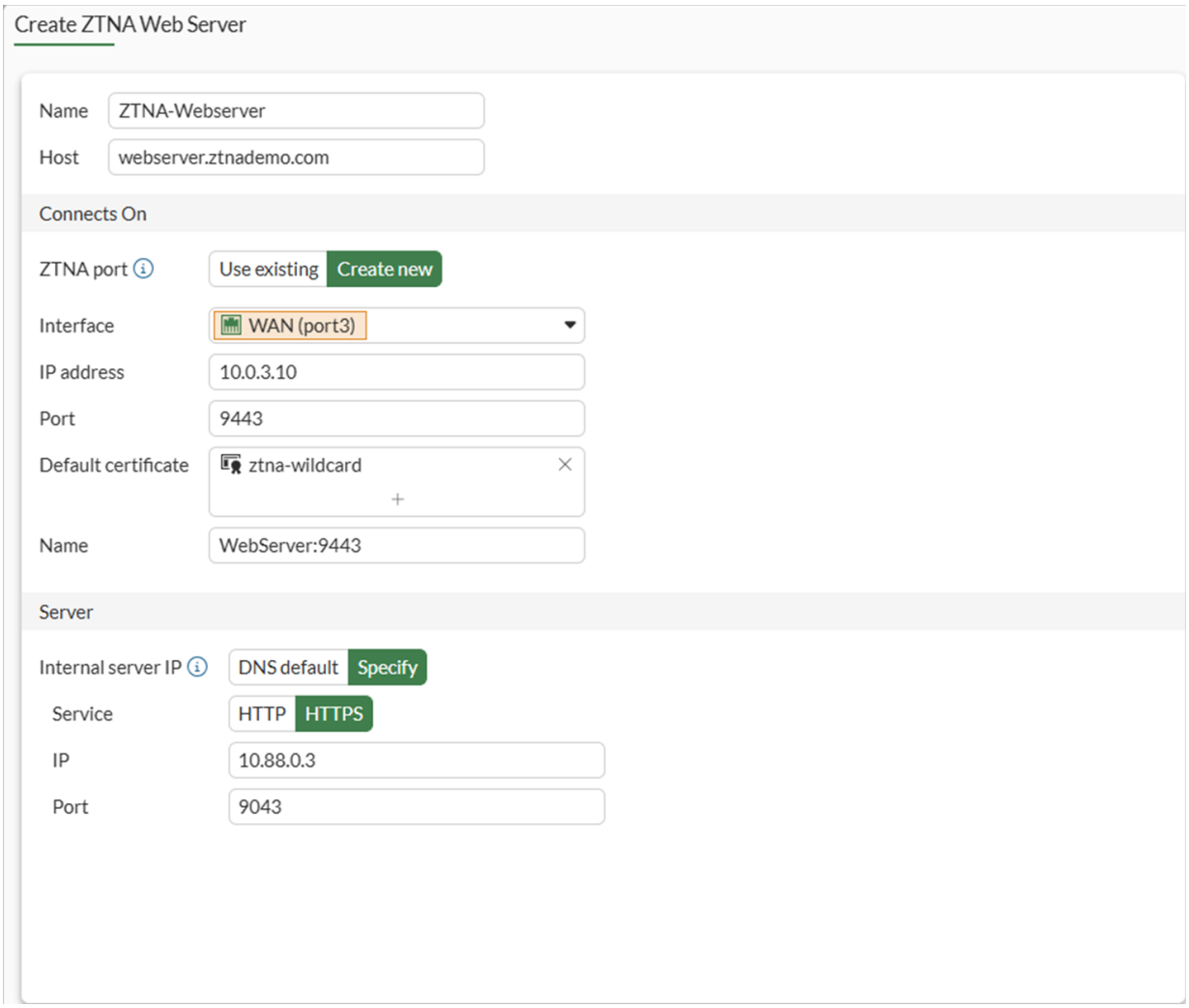
5. Click *Update*.

To configure a ZTNA server for HTTPS access proxy in the FortiGate GUI:

1. Go to *Policy & Objects* > *ZTNA* > *Servers & Portals* > *Web server*.
2. Click *Create New*.
3. Set *Name* to *ZTNA-Webserver*.
4. Set *Host* to the IP address or FQDN that will be used to access the webserver: webserver.ztnademo.com.
5. Under *Connects On*, select *Create new* for ZTNA port to provide the following details:
   a. Set *Interface* to *port3*.
   b. Set *External IP* to *10.0.3.10*.
   c. Set *External port* to *9443*.

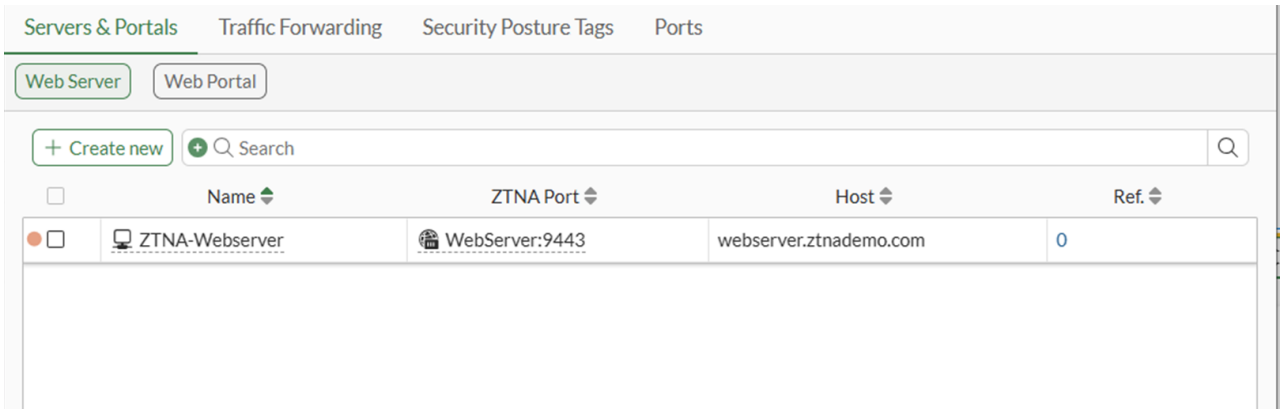
   Verify that the IP address and port do not conflict with management access to the interface. Otherwise, change the IP address to another address on that subnet.
6. Select a certificate. Clients will be presented with this certificate when they connect to the ZTNA port.

   This example uses a wildcard certificate with a Common Name (CN): *.ztnademo.com. This certificate is trusted by the endpoints which will access the ZTNA port.
7. Enter a name for the ZTNA port, such as *WebServer:9443*.
8. In the *Server* section, configure the internal server settings:
   a. Switch the *Internal server IP* to *Specify*.
   b. Set *Service* to *HTTPS*.
   c. Set *IP* to the internal IP for the web server: *10.88.0.3*.
   d. Set *Port* to *9043*.



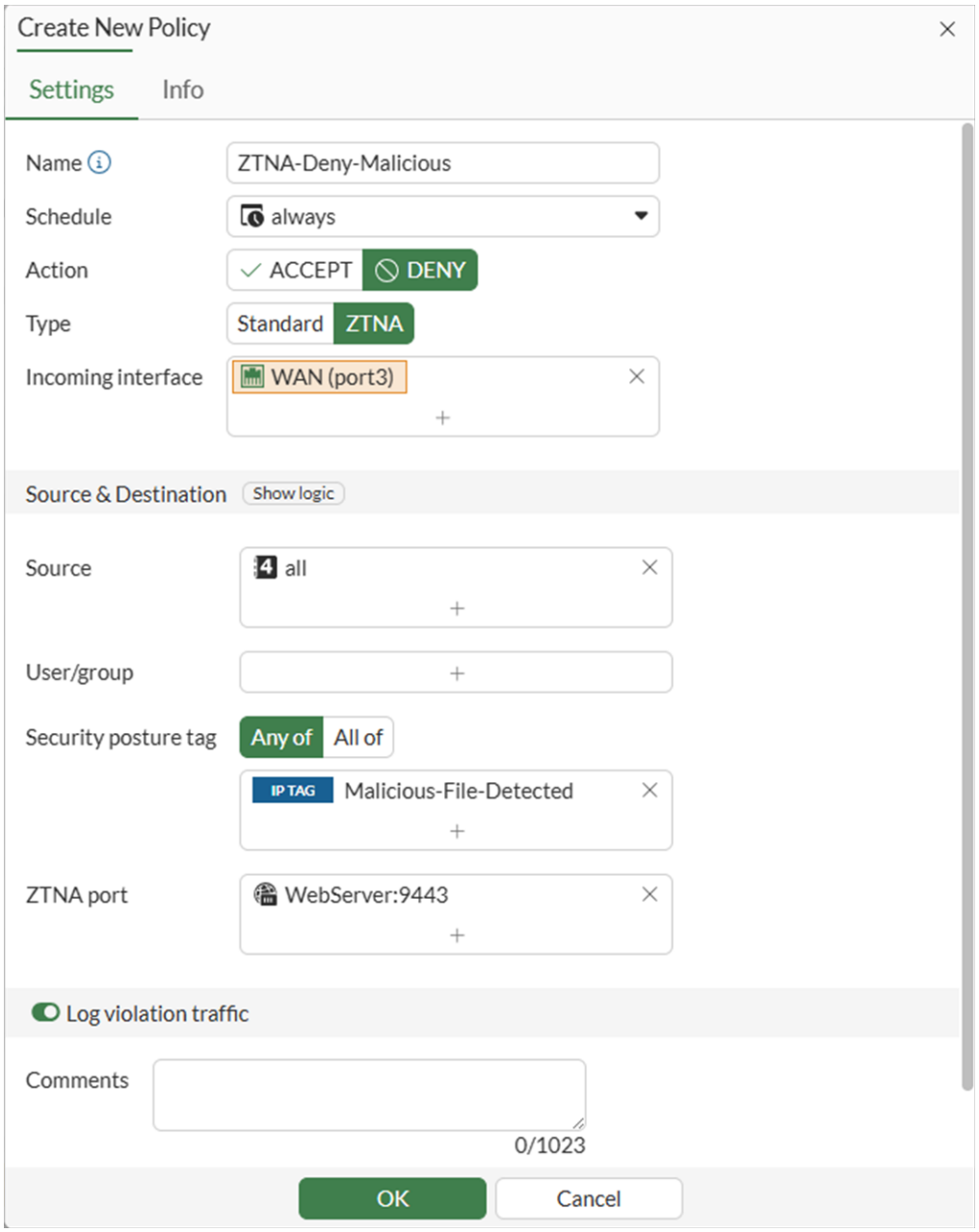
The WebServer service is running on port 9043. This configuration will translate clients accessing TCP/9443 to TCP/9043.

9. Click *OK*. The Web server object is listed.



To configure simple ZTNA policies to allow and deny traffic based on security posture tags in the GUI:

1. Go to *Policy & Objects > Firewall Policy*.
2. Create a rule to deny traffic:
   a. Click *Create New*.
   b. Set *Name* to *ZTNA-Deny-Malicious*.
   c. Set *Action* to *DENY*.
   d. Set *Type* to *ZTNA*.
   e. Set *Incoming Interface* to *port3*.
   f. Set *Source* to *all*.
   g. Keep the default of Any of, and add the security posture tag *IP Malicious-File-Detected*.
   This tag is dynamically retrieved from EMS when you first created the Zero Trust Tagging Rule.
   h. Set *ZTNA port* to the *WebServer:9443* port created earlier.
   i. Enable *Log Violation Traffic*.



j. Click *OK*.

3. Create a rule to allow traffic:
   a. Click *Create New*.
   b. Set *Name* to *ZTNA-Allow-Simple*.
   c. Set *Type* to *ZTNA*.
   d. Set *Incoming Interface* to *port3*.
   e. Set *Source* to *all*. This can also be set to specific IP addresses to only allow those addresses to connect to this HTTPS access proxy.
   f. Keep the default of *Any of*, and add the class tag IP Low.
   g. Select the *ZTNA port WebServer:9443*.

h. Configure the remaining options as needed.
i. Click *OK*.

4. On the firewall policy list, make sure that the deny rule (*ZTNA-Deny-Malicious*) is above the allow rule (*ZTNA-Allow-Simple*).

To configure HTTPS access in the CLI:

1. Configure the access proxy VIP:

```
config firewall vip
    edit "WebServer:9443"
        set type access-proxy
        set server-type https
        set extip 10.0.3.10
        set extintf "port3"
        set extport 9443
        set ssl-certificate "ztna-wildcard"
    next
end
```

2. Configure the virtual host:

```
config firewall access-proxy-virtual-host
    edit "WebServer9443"
        set ssl-certificate “ztna-wildcard”
        set host “webserver.ztnademo.com”
    next
end
```

3. Configure the web proxy:

```
config ztna web-proxy
    edit "ZTNA-Webserver"
        set vip "Webserver:9443"
        set host WebServer9443
        config api-gateway
            edit 1
                config realservers
                    edit 1
                        set ip 10.88.0.3
                        set port 9043
                    next
                end
            next
        end
    next
end
```

4. Configure ZTNA policies:

```
config firewall policy
    edit 1
        set name "ZTNA-Deny-Malicious"
        set srcintf "port3"
        set dstintf "any"
        set srcaddr "all"
        set dstaddr "WebServer:9443"
        set ztna-ems-tag "EMS1_ZTNA_Malicious-File-Detected"
        set schedule "always"
        set logtraffic all
    next
    edit 2
        set name "ZTNA-Allow-Simple"
        set srcintf "port3"
        set dstintf "any"
        set action accept
        set srcaddr "all"
        set dstaddr "WebServer:9443"
        set ztna-ems-tag "EMS1_CLASS_Low"
        set schedule "always"
        set logtraffic all
    next
end
```

Testing the remote access to the HTTPS access proxy

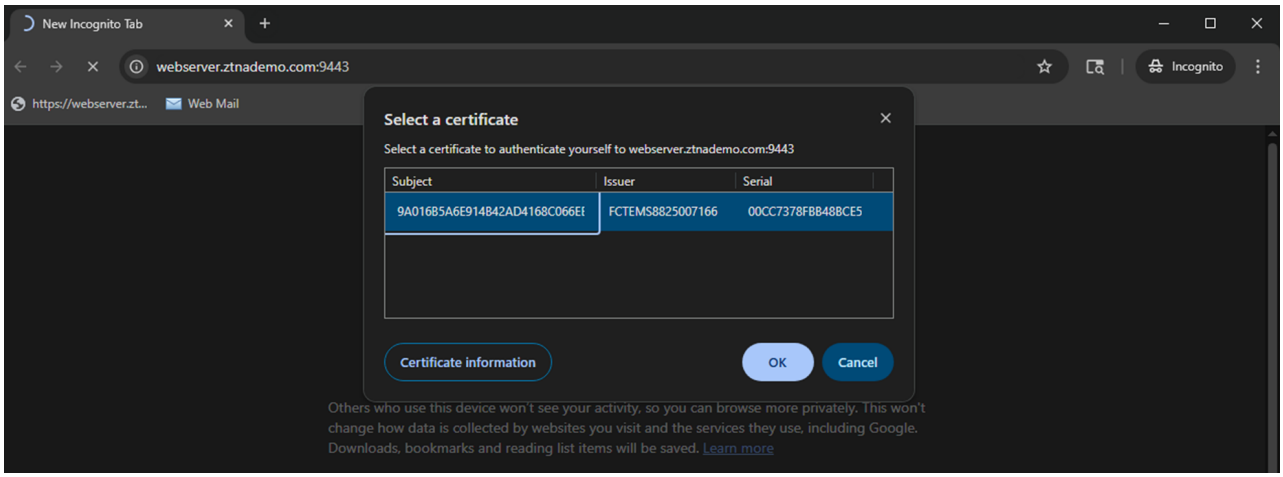
After FortiClient EMS and FortiGate are configured, the HTTPS access proxy remote connection can be tested.

Access allowed:

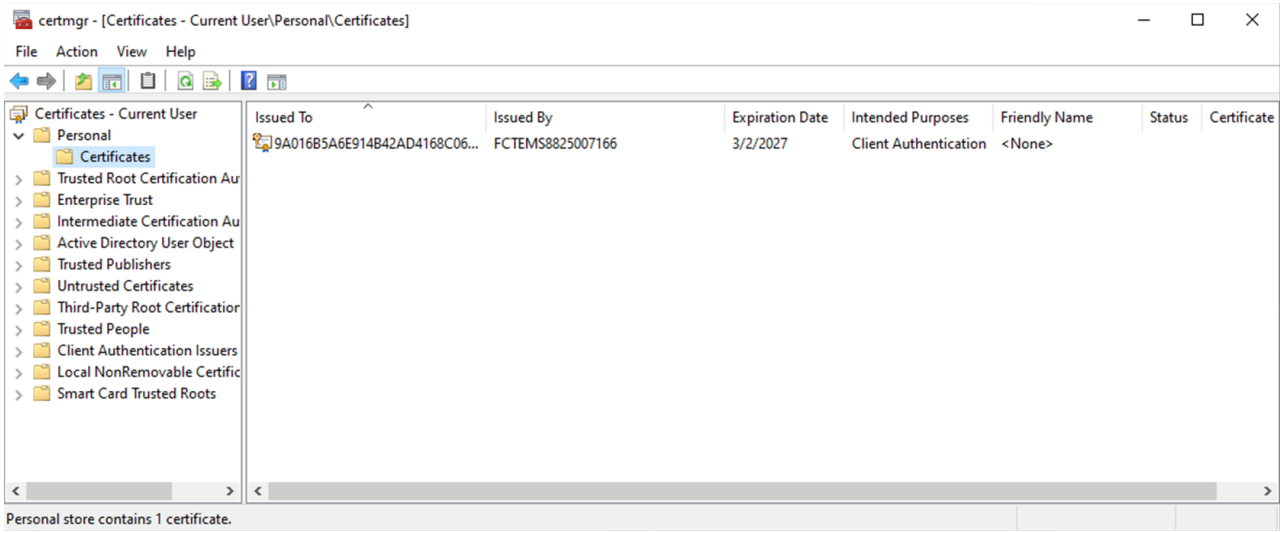
1. On the remote Windows PC, open FortiClient.
2. On the *Zero Trust Telemetry* tab, make sure that you are connected to the EMS server.

   It is not necessary to configure a *ZTNA Destination* on FortiClient for the HTTPS access proxy use case. In fact, configuring a *ZTNA Destination* rule for the website may interfere with its operation.

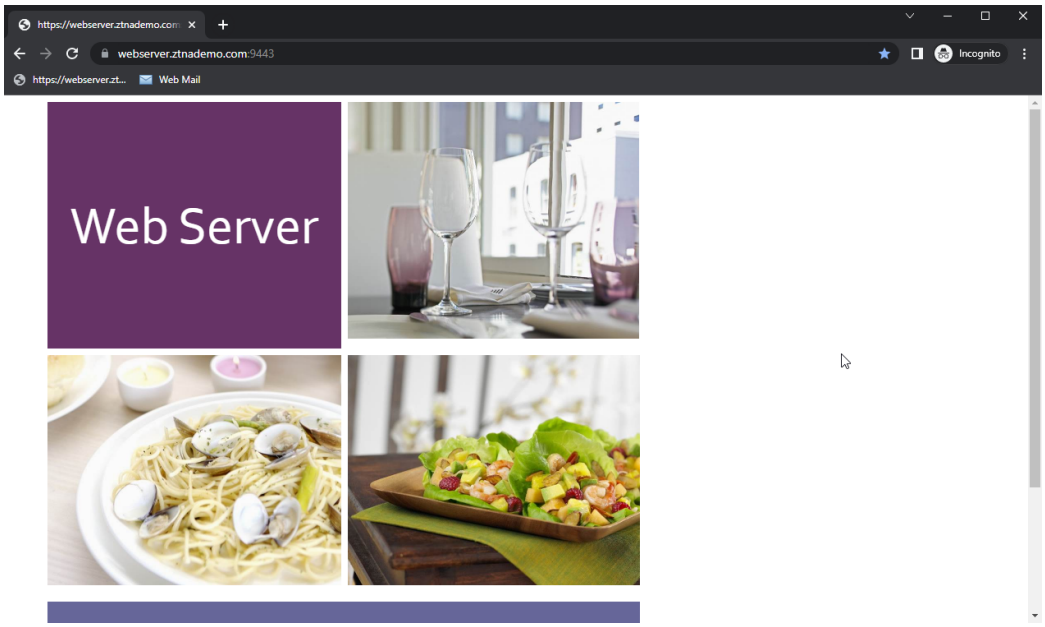
3. Open a browser and enter the address of the server and the access port. When entering the FQDN, make sure that the DNS can resolve the address to the IP address of the FortiGate. In this example, webserver.ztnademo.com resolves to 10.0.3.10.
4. The browser prompts for the client certificate to use. Select the EMS signed certificate, then click *OK*.



The certificate is in the *User Configuration* store, under *Personal* > *Certificates*. The details show the SN of the certificate, which matches the record on the FortiClient EMS and the FortiGate.

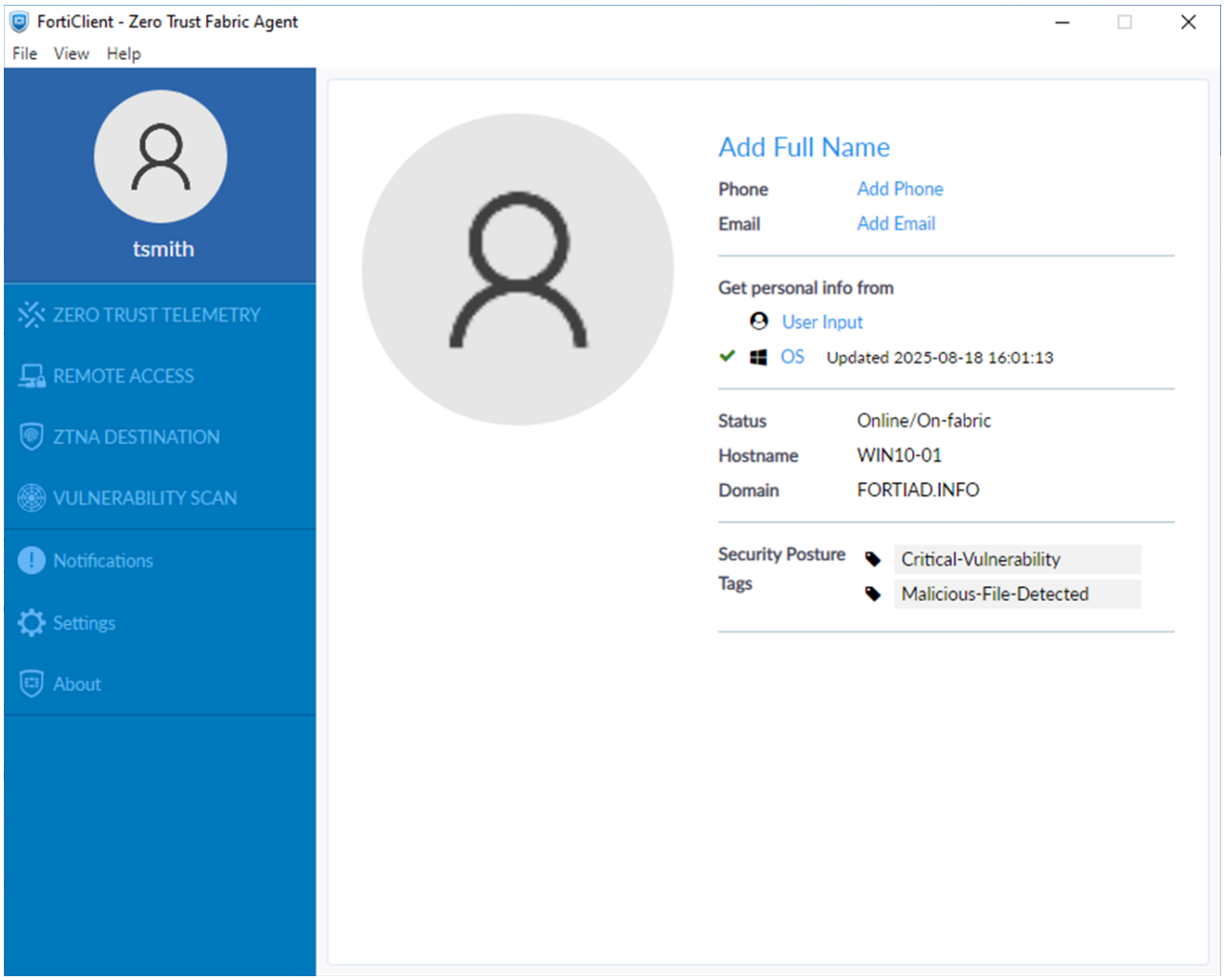


5. The client is verified by the FortiGate to authenticate your identity.
6. The FortiGate matches your security posture by verifying your security posture tag and/or classification tag. The matching ZTNA rule allows access to the web server.

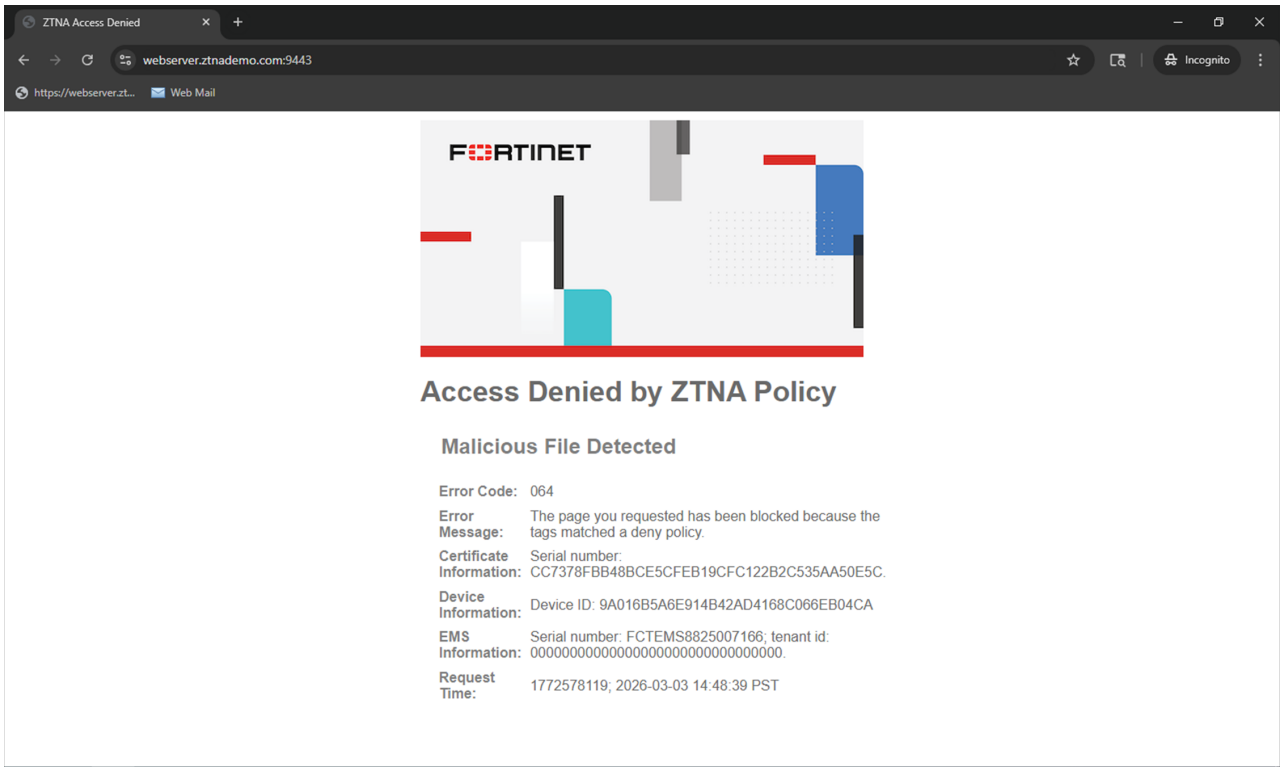


Access denied:

1. On the remote Windows PC, trigger the Zero Trust Tagging Rule by creating the file in C:\virus.txt.
2. Verify that FortiClient has received the *Malicious-File-Detected* tag by opening FortiClient and clicking the portrait in the top left.



3. Open a browser and enter the address http://webserver.ztnademo.com:9443.
4. The client is verified by the FortiGate to authenticate your identity.
5. FortiGate checks your security posture. Because EMS has tagged the PC with the *Malicious-File-Detected* tag, it matches the *ZTNA-Deny-malicious* rule.
6. You are denied access to the web server.



Logs and debugs

Access allowed:

```
# diagnose endpoint ec-shm list
Record #0:
                IP Address = 10.0.3.2
                MAC Address = 02:09:0f:00:03:03
                MAC list =
                VDOM =  (-1)
                EMS serial number: FCTEMS8825007166
                EMS tenant id: 00000000000000000000000000000000
                Client cert SN: CC7378FBB48BCE5CFEB19CFC122B2C535AA50E5C
                Public IP address: 148.230.50.203
                Quarantined: no
                Online status: online
                Registration status: registered
                On-net status: off-net
                Gateway Interface:
                FortiClient version: 7.4.3
                AVDB version: 1.0
                FortiClient app signature version: 35.177
                FortiClient vulnerability scan engine version: 3.8
                FortiClient UID: 9A016B5A6E914B42AD4168C066EB04CA
                Host Name: WIN10-01
                OS Type: WIN64
                OS Version: Microsoft Windows 10 Professional Edition, 64-bit (build 19045)
```

```
                Domain: fortiad.info
                Last Login User: tsmith
                Owner: @fortiad.info
                Host Model: Standard PC (i440FX + PIIX, 1996)
                Host Manufacturer: QEMU
                CPU Model: Intel(R) Xeon(R) CPU @ 2.20GHz
                Memory Size: 8095
                AV Feature: 1
                FW Feature: 1
                WF Feature: 1
                AS Feature: 0
                VS Feature: 1
                VN Feature: 1
                Last vul message received time: Tue Mar  3 12:36:42 2026
                Last vul scanned time: Tue Mar  3 17:38:36 2026
                Last vul statistic: critical=0, high=0, medium=0, low=1, info=0
                Avatar fingerprint:
                Avatar source username: Tom Smith
                Avatar source email:
                Avatar source: OS
                Phone number:
                Issued time in seconds: 1772570201
                JWT expiry time in seconds: 0 (Current time in seconds: 1772577332)
                Subscribed only: true
                Sendall: disabled
                Number of Routes: (0)
```

```
# diagnose test application fcnacd 7
#ID:0
UID:     9A016B5A6E914B42AD4168C066EB04CA
EMS:     FCTEMS8825007166 00000000000000000000000000000000
State:   sysinfo:1, tag:1, tagsz:1, out-of-sync:0, ref:1
Owner:   @fortiad.info
Cert SN: CC7378FBB48BCE5CFEB19CFC122B2C535AA50E5C
online:  Yes
Sub TS:  1772570201
idx:0, ttdl:1   name:all_registered_clients
idx:3, ttdl:2   name:Low
```

```
# diagnose wad dev query-by uid 9A016B5A6E914B42AD4168C066EB04CA FCTEMS8825007166
00000000000000000000000000000000
Attr of type=0, length=83, value(ascii)=9A016B5A6E914B42AD4168C066EB04CA
Attr of type=7, length=8, value(ascii)=Gi
Attr of type=4, length=13, value(ascii)=@fortiad.info
Attr of type=6, length=1, value(ascii)=true
Attr of type=9, length=1, value(ascii)=1
Attr of type=5, length=40, value(ascii)=CC7378FBB48BCE5CFEB19CFC122B2C535AA50E5C
Attr of type=3, length=76, value(ascii)=ZTNA_all_registered_clients_
FCTEMS8825007166000000000000000000000000000000
Attr of type=3, length=58, value(ascii)=CLASS_Low_FCTEMS8825007166000000000000000000000000000000
Response termination due to no more data
```

```
# execute log filter category 0
# execute log filter field subtype ztna
# execute log display

1: date=2026-03-03 time=14:46:14 eventtime=1772577974130406404 tz="-0800" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=64377
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.88.0.3
dstport=9043 dstintf="port2" dstintfrole="dmz" sessionid=9174 srcuuid="b458a65a-f759-51ea-d7df-
ef2e750026d1" dstuuid="01b612e6-174a-51f1-1a3d-b3293e8efa12" service="tcp/9043"
proxyapptype="ztna-proxy" proto=6 action="accept" policyid=2 policytype="policy"
poluuid="693c3e3e-174c-51f1-3e13-715130a1a107" policyname="ZTNA-Allow-Simple" appcat="unscanned"
duration=129 gatewayid=1 realserverid=1 vip="WebServer:9443" vipincomingip=10.0.3.10
accessproxy="ZTNA-Webserver" clientdeviceid="9A016B5A6E914B42AD4168C066EB04CA"
clientdevicemanageable="manageable" clientdeviceems="FCTEMS8825007166" clientdevicetags="CLASS_
Low/ all_registered_clients" clientcert="yes" emsconnection="online" wanin=303009 rcvdbyte=303009
wanout=4424 lanin=4220 sentbyte=4220 lanout=305716 fctuid="9A016B5A6E914B42AD4168C066EB04CA"
```

Access denied:

```
# diagnose test application fcnacd 7
#ID:0
UID:     9A016B5A6E914B42AD4168C066EB04CA
EMS:     FCTEMS8825007166 00000000000000000000000000000000
State:   sysinfo:1, tag:1, tagsz:1, out-of-sync:0, ref:1
Owner:   @fortiad.info
Cert SN: CC7378FBB48BCE5CFEB19CFC122B2C535AA50E5C
online:  Yes
Sub TS:  1772570201
idx:0, ttdl:1   name:all_registered_clients
idx:1, ttdl:1   name:Malicious-File-Detected
idx:3, ttdl:2   name:Low
```

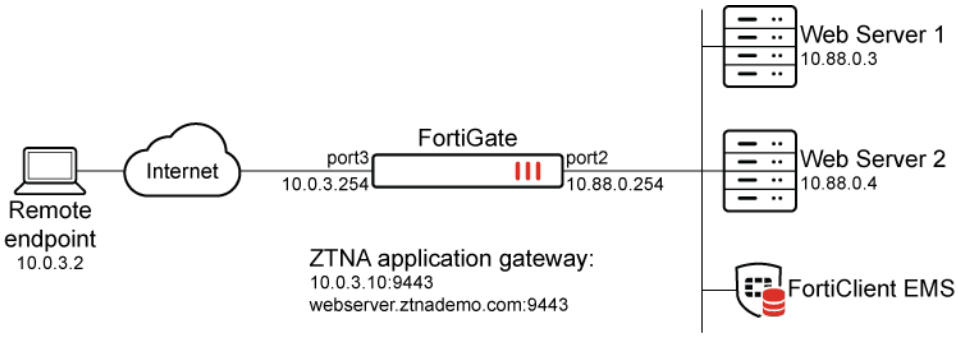
```
# diagnose wad dev query-by uid 9A016B5A6E914B42AD4168C066EB04CA FCTEMS8825007166
00000000000000000000000000000000
Attr of type=0, length=83, value(ascii)=9A016B5A6E914B42AD4168C066EB04CA
Attr of type=7, length=8, value(ascii)=di
Attr of type=4, length=13, value(ascii)=@fortiad.info
Attr of type=6, length=1, value(ascii)=true
Attr of type=9, length=1, value(ascii)=1
Attr of type=5, length=40, value(ascii)=CC7378FBB48BCE5CFEB19CFC122B2C535AA50E5C
Attr of type=3, length=76, value(ascii)=ZTNA_all_registered_clients_
FCTEMS882500716600000000000000000000000000000000
Attr of type=3, length=77, value(ascii)=ZTNA_Malicious-File-Detected_
FCTEMS8825007166000000000000000000000000000000000
Attr of type=3, length=58, value(ascii)=CLASS_Low_FCTEMS882500716600000000000000000000000000000000
Response termination due to no more data
```

```
# execute log filter category 0
# execute log filter field subtype ztna
# execute log display
```

```
1: date=2026-03-03 time=14:48:40 eventtime=1772578119728445929 tz="-0800" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=64396
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.0.3.10
dstport=9443 dstintf="root" dstintfrole="undefined" sessionid=9280 srcuuid="b458a65a-f759-51ea-
d7df-ef2e750026d1" dstuuid="01b612e6-174a-51f1-1a3d-b3293e8efa12" service="tcp/9443"
proxyapptype="ztna-proxy" proto=6 action="deny" policyid=1 policytype="policy" poluuid="393a561c-
174c-51f1-cdd1-442746c0ffc3" policyname="ZTNA-Deny-Malicious" appcat="unscanned" duration=0
vip="WebServer:9443" vipincomingip=10.0.3.10 clientdeviceid="9A016B5A6E914B42AD4168C066EB04CA"
clientdevicemanageable="manageable" clientdeviceems="FCTEMS8825007166" clientdevicetags="CLASS_
Low/Malicious-File-Detected/ all_registered_clients" clientcert="yes" emsconnection="online"
wanin=0 rcvdbyte=0 wanout=0 lanin=3773 sentbyte=3773 lanout=4524
fctuid="9A016B5A6E914B42AD4168C066EB04CA" msg="Traffic denied because proxy-policy action is deny.
Matched tag: EMS1_ZTNA_Malicious-File-Detected" crscore=30 craction=131072 crlevel="high"
```

ZTNA web proxy with load balancing example

This example expands on the ZTNA web proxy example on page 1677 example, adding a second webserver that is to be load balanced for webserver.ztnademo.com requests and has health checks enabled. Configure the ZTNA web proxy example before continuing.



To add a second webserver as a real server in the existing web proxy:

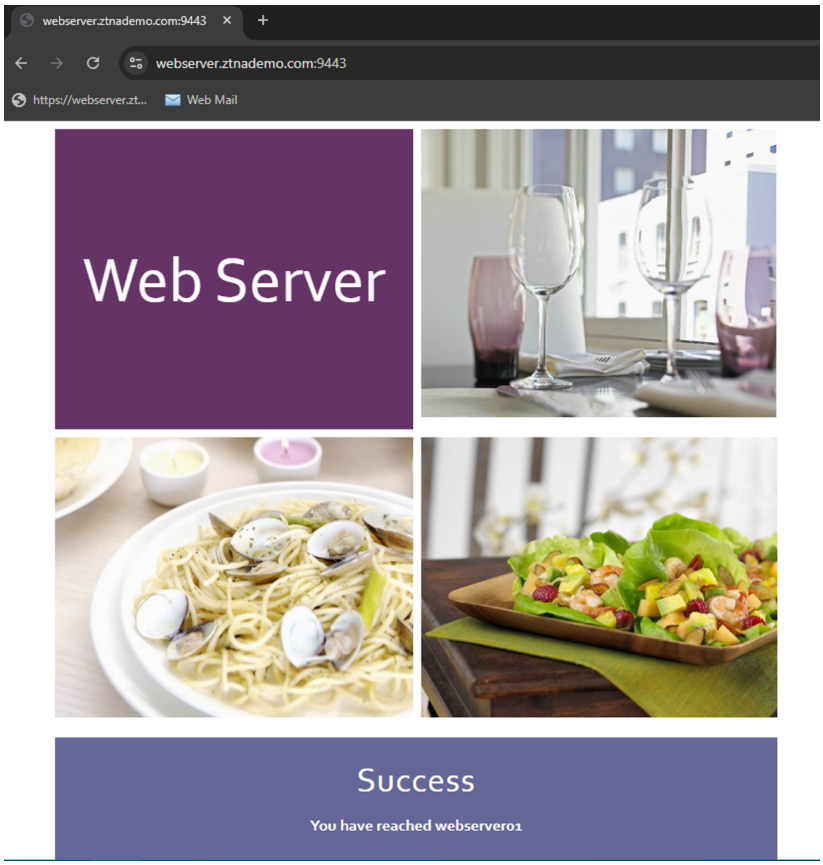
> It is not possible to configure a second real server for a web proxy in the GUI and, after configuring the second real server in the CLI, you cannot make further edits to the server in the GUI.

```
config ztna web-proxy
    edit "ZTNA-Webserver"
        set vip "Webserver:9443"
        set host "WebServer9443"
        config api-gateway
            edit 1
                config realservers
                    edit 1
                        set health-check enable
                    next
```

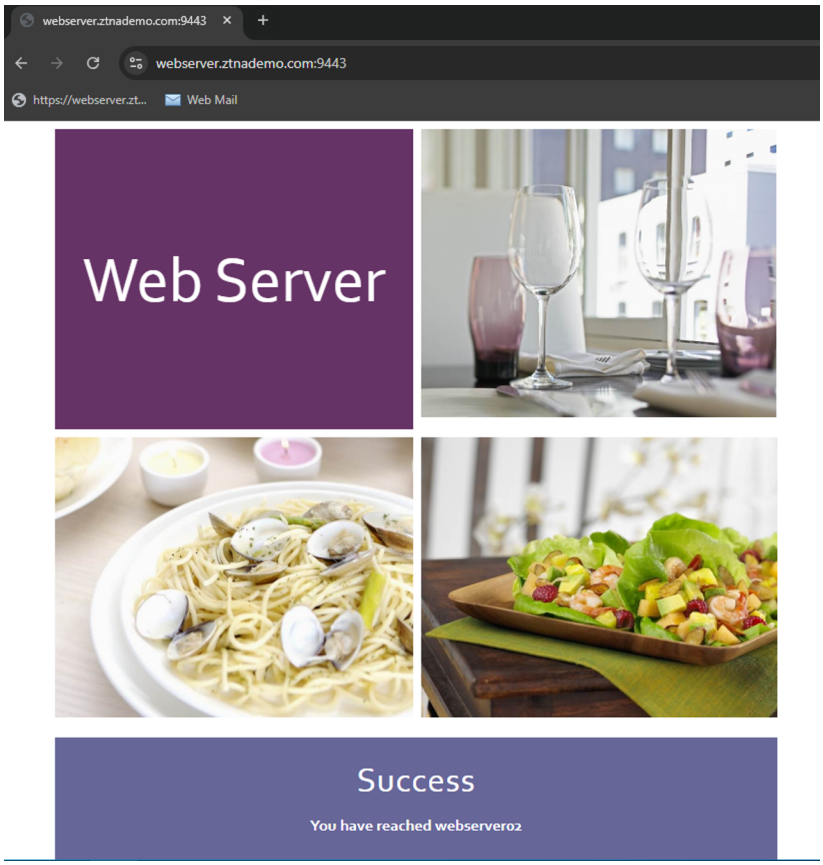
```
                    edit 2
                        set ip 10.88.0.4
                        set port 9043
                        set health-check enable
                    next
                end
                set ldb-method round-robin
            next
        end
    next
end
```

To verify that load balancing is working :

1. On the remote Windows PC, open a browser and enter the address of the server and the access port.



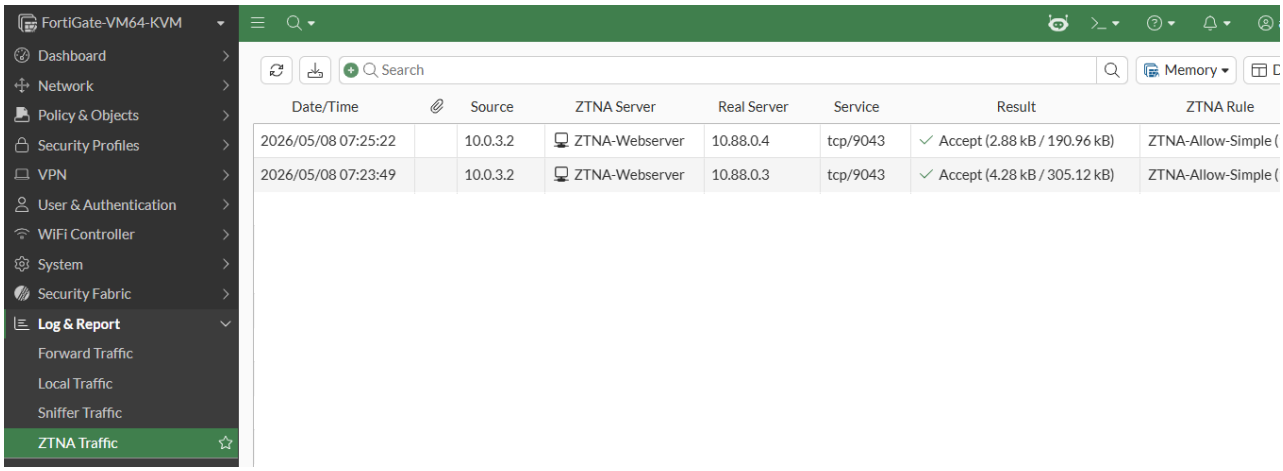
2. Refresh the page to load the second server.



These webservers are configured to display their hostname at the bottom of the page. You may need to verify server load balancing using packet capture on the FortiGate or by reviewing the ZTNA logs.

3. Close the browser to complete the session and generate a log.

   Note that only one log will be generated, and it will reflect the second server accessed. To generate a second log, connect to the ZTNA web proxy a second time.



```
execute log filter category 0
execute log filter field subtype ztna
execute log display

1: date=2026-05-08 time=07:25:22 eventtime=1778211337861311393 tz="-0700" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=52396
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.88.0.4
dstport=9043 dstintf="port2" dstintfrole="dmz" sessionid=36144 srcuuid="b458a65a-f759-51ea-
d7df-ef2e750026d1" service="tcp/9043" proxyapptype="ztna-proxy" proto=6 action="accept"
policyid=2 policytype="policy" poluuid="b77e7abe-4a8a-51f1-54f1-b73a574438e0"
policyname="ZTNA-Allow-Simple" appcat="unscanned" duration=136 gatewayid=1 realserverid=2
vip="WebServer:9443" vipincomingip=10.0.3.10 accessproxy="ZTNA-Webserver" ztnaproxytype="ztna-
web" clientdevicemanageable="manageable" clientcert="yes" wanin=513933 rcvdbyte=513933
wanout=8068 lanin=4551 sentbyte=4551 lanout=514704

2: date=2026-05-08 time=07:23:49 eventtime=1778211775251735907 tz="-0700" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=52425
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.88.0.3
dstport=9043 dstintf="port2" dstintfrole="dmz" sessionid=36410 srcuuid="b458a65a-f759-51ea-
d7df-ef2e750026d1" service="tcp/9043" proxyapptype="ztna-proxy" proto=6 action="accept"
policyid=2 policytype="policy" poluuid="b77e7abe-4a8a-51f1-54f1-b73a574438e0"
policyname="ZTNA-Allow-Simple" appcat="unscanned" duration=131 gatewayid=1 realserverid=1
vip="WebServer:9443" vipincomingip=10.0.3.10 accessproxy="ZTNA-Webserver" ztnaproxytype="ztna-
web" clientdevicemanageable="manageable" clientcert="yes" wanin=503831 rcvdbyte=503831
wanout=7485 lanin=4476 sentbyte=4476 lanout=504533
```

To check the health status of both servers:

```
# diagnose wad ztna-web-proxy health-check status

Link Monitor: ZP-ZTNA-Webserver-1-1, Status: alive, Server num(1), cfg_version=0 Flags=0x1 init,
Create time: Fri May  8 07:30:57 2026
VRF: 0
Interval: 1000 ms
Service-detect: disable
Diffservcode: 000000
Class-ID: 0
Transport-Group: 0
Class-ID: 0
  Peer: 10.88.0.3(10.88.0.3)
        protocol: ping(9043), state: alive
                Latency(Min/Max/Avg): 0.640/2.001/0.854 ms
                Jitter(Min/Max/Avg): 0.003/1.229/0.211 ms
                Packet lost: 0.000%
                MOS: 4.404
                Number of out-of-sequence packets: 0
                Fail Times(0/5)
                Packet sent: 729, received: 729, Sequence(sent/rcvd/exp): 730/730/731

Link Monitor: ZP-ZTNA-Webserver-1-2, Status: alive, Server num(1), cfg_version=0 Flags=0x1 init,
Create time: Fri May  8 07:30:57 2026
VRF: 0
```

```
Interval: 1000 ms
Service-detect: disable
Diffservcode: 000000
Class-ID: 0
Transport-Group: 0
Class-ID: 0
  Peer: 10.88.0.4(10.88.0.4)
        protocol: ping(9043), state: alive
                Latency(Min/Max/Avg): 0.522/3.145/0.812 ms
                Jitter(Min/Max/Avg): 0.001/2.470/0.328 ms
                Packet lost: 0.000%
                MOS: 4.404
                Number of out-of-sequence packets: 0
                Fail Times(0/5)
                Packet sent: 729, received: 729, Sequence(sent/rcvd/exp): 730/730/731
```

To verify that the health check status will remove dead servers from the load balancing:

1. Make one of the servers unavailable.
   In this example, a blackhole route for that host is configured:

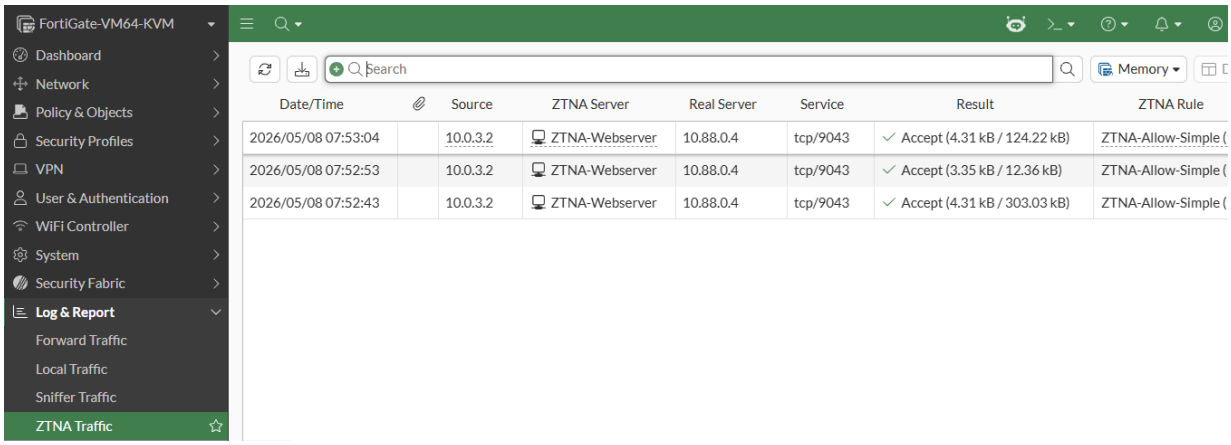
```
config router static
    edit 0
        set dst 10.88.0.3/32
        set blackhole enable
    next
end
```

2. Review the health-check status for server status:

```
# diagnose wad ztna-web-proxy health-check status

Link Monitor: ZP-ZTNA-Webserver-1-1, Status: dead, Server num(1), cfg_version=0 Flags=0x1
init, Create time: Fri May  8 07:30:57 2026
VRF: 0
Interval: 1000 ms
Service-detect: disable
Diffservcode: 000000
Class-ID: 0
Transport-Group: 0
Class-ID: 0
  Peer: 10.88.0.3(10.88.0.3)
        protocol: ping(9043), state: dead
                Packet lost: 5.000%
                MOS: 4.041
                Number of out-of-sequence packets: 0
                Recovery times(0/5) Fail Times(0/5)
                Packet sent: 941, received: 935, Sequence(sent/rcvd/exp): 942/936/937

Link Monitor: ZP-ZTNA-Webserver-1-2, Status: alive, Server num(1), cfg_version=0 Flags=0x1
init, Create time: Fri May  8 07:30:57 2026
VRF: 0
```

```
Interval: 1000 ms
Service-detect: disable
Diffservcode: 000000
Class-ID: 0
Transport-Group: 0
Class-ID: 0
  Peer: 10.88.0.4(10.88.0.4)
        protocol: ping(9043), state: alive
                Latency(Min/Max/Avg): 0.516/2.041/0.801 ms
                Jitter(Min/Max/Avg): 0.005/1.172/0.213 ms
                Packet lost: 0.000%
                MOS: 4.404
                Number of out-of-sequence packets: 0
                Fail Times(0/5)
                Packet sent: 941, received: 941, Sequence(sent/rcvd/exp): 942/942/943
```

3. Verify that webserver01 is no longer an option for ZTNA users to access via load balancing:
   a. Connect to the webserver.ztnademo.com:9443 as you did previously. After accepting the certificate, webserver02 will load.
   b. After refreshing several times, only webserver02 is displayed. Similarly, opening new connections will always return webserver02.
4. Review the logs to confirm your findings, particularly if the webpage served does not uniquely identify the server that provided it:
   a. This example shows multiple connections to webserver.ztnademo.com. Note that all the connections are to 10.88.0.4:

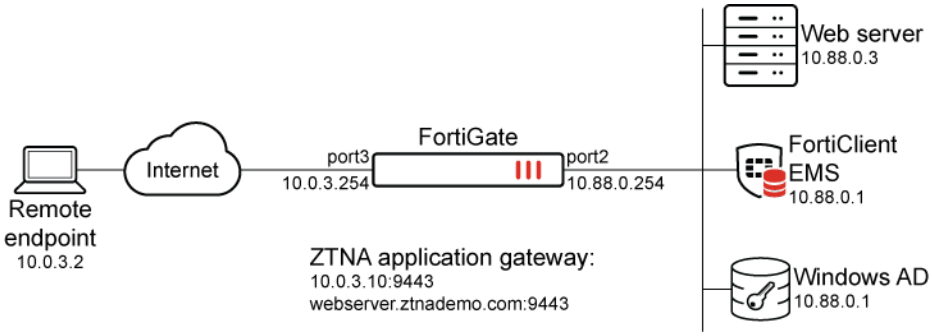


   b. If not for the health-check removing the server from load-balancing, users would receive an unreachable/gateway timeout error (504).

ZTNA web proxy with basic authentication example

This example expands on the ZTNA web proxy example on page 1677 example, adding LDAP authentication to the ZTNA rule. Users are allowed based on passing the client certificate authentication check, user authentication, and security posture check.

Users that are in the AD security group *ALLOWED-VPN* are allowed access to the access proxy. Users that are not part of this security group are not allowed access.



This example assumes that the FortiGate EMS fabric connector is already successfully connected.

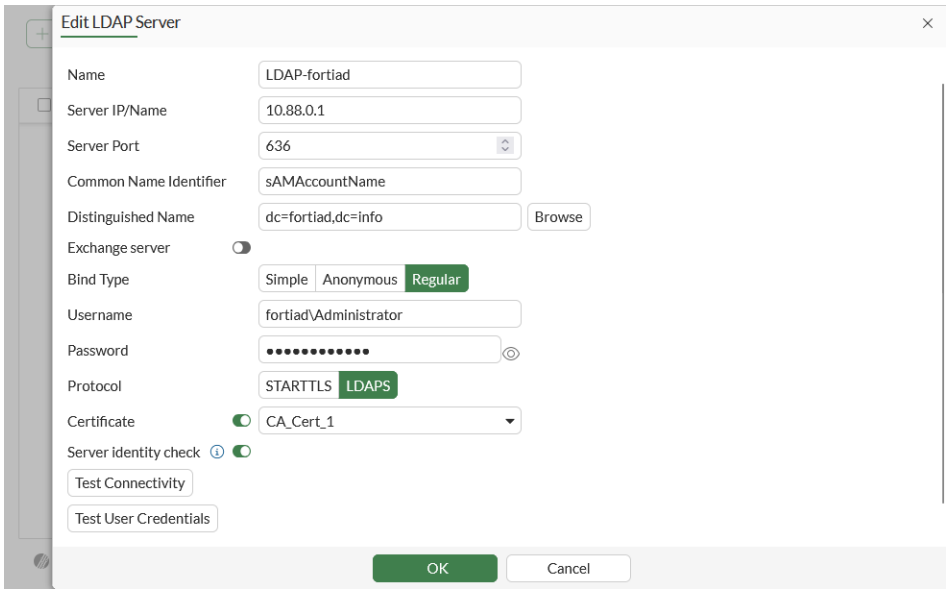
LDAP/Active Directory Users and Groups:

- Domain: fortiad.info
- Users (Groups):
  - tsmith (Domain Users, Remote-Allowed)
  - lhansen (Domain Users)

To configure a secure connection to the LDAP server in the GUI:

1. Go to *User & Authentication* > *LDAP Servers* and click *Create New*.
2. Configure the following settings:

Name	LDAP-fortiad
Server IP/Name	10.88.0.1
Server Port	636
Common Name Identifier	sAMAccountName
Distinguished Name	dc=fortiad,dc=info
Exchange server	Disabled
Bind Type	Regular Enter the *Username* and *Password* for LDAP binding and lookup.
Protocol	LDAPS
Certificate	Enable and select the CA certificate to validate the server certificate.
Server identity check	Optionally, enable to verify the domain name or IP address against the server certificate.



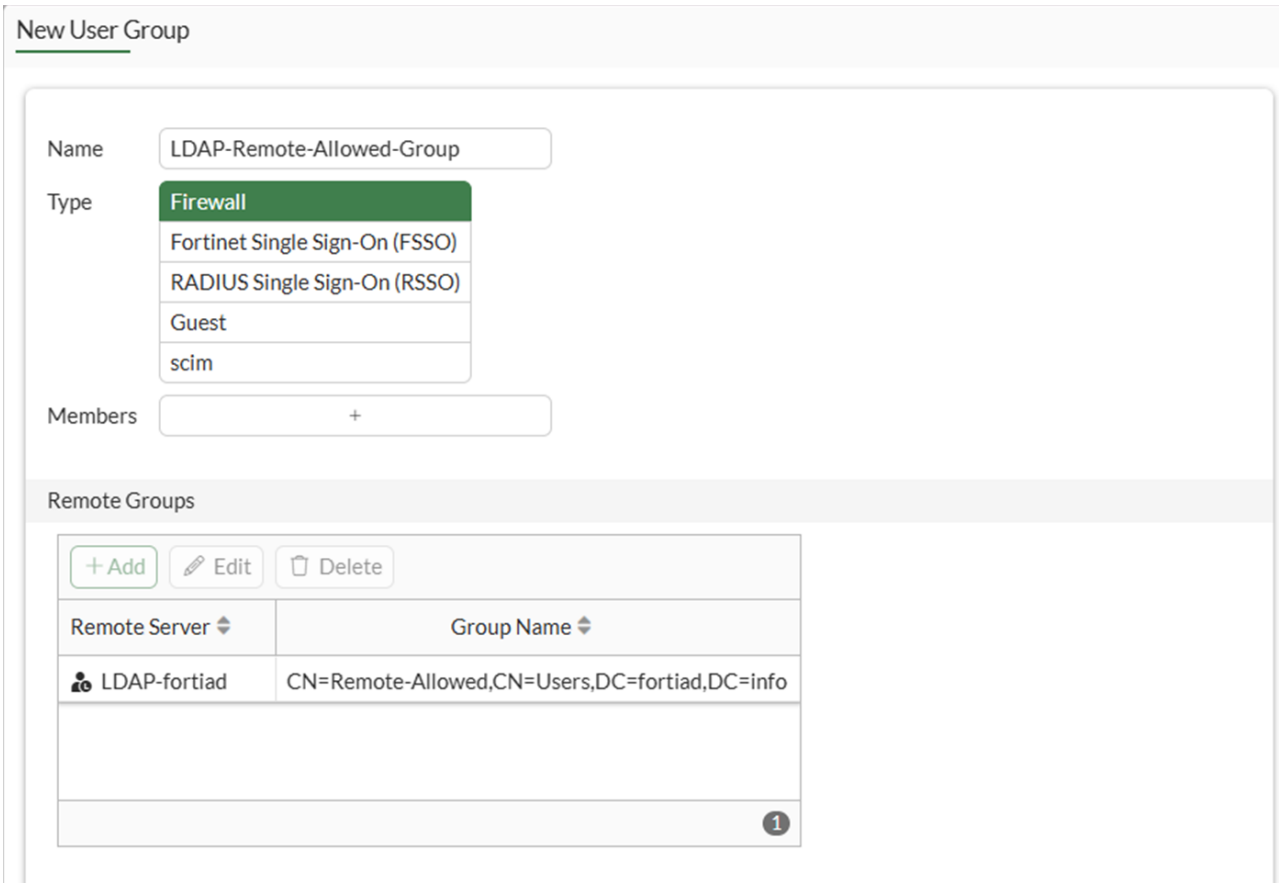
3. Click *Test Connectivity* to verify the connection to the server.
4. Click *OK*.

To configure a secure connection to the LDAP server in the CLI:

```
config user ldap
    edit "LDAP-fortiad"
        set server "10.88.0.1"
        set cnid "sAMAccountName"
        set dn "dc=fortiad,dc=info"
        set type regular
        set username "fortiad\\Administrator"
        set password <password>
        set secure ldaps
        set ca-cert "CA_Cert_1"
        set port 636
    next
end
```

To configure a remote user group from the LDAP server in the GUI:

1. Go to *User & Authentication* > *User Groups* and click *Create New*.
2. Set the name to *LDAP-Remote-Allowed-Group*.
3. Set *Type* to *Firewall*.
4. In the *Remote Groups* table click *Add*:
   a. Set *Remote Server* to *LDAP-fortiad*.
   b. Locate the *Remote-Allowed* group, right-click on it, and click *Add Selected*.
   c. Click *OK*.



5. Click *OK*.

To configure a remote user group from the LDAP server in the CLI:

```
config user group
    edit "LDAP-Remote-Allowed-Group"
        set member "LDAP-fortiad"
        config match
            edit 1
                set server-name "LDAP-fortiad"
                set group-name "CN=Remote-Allowed,CN=Users,DC=fortiad,DC=info"
            next
        end
    next
end
```

Authentication scheme and rules

After the LDAP server and user group have been configured, an authentication scheme and rule must be configured.

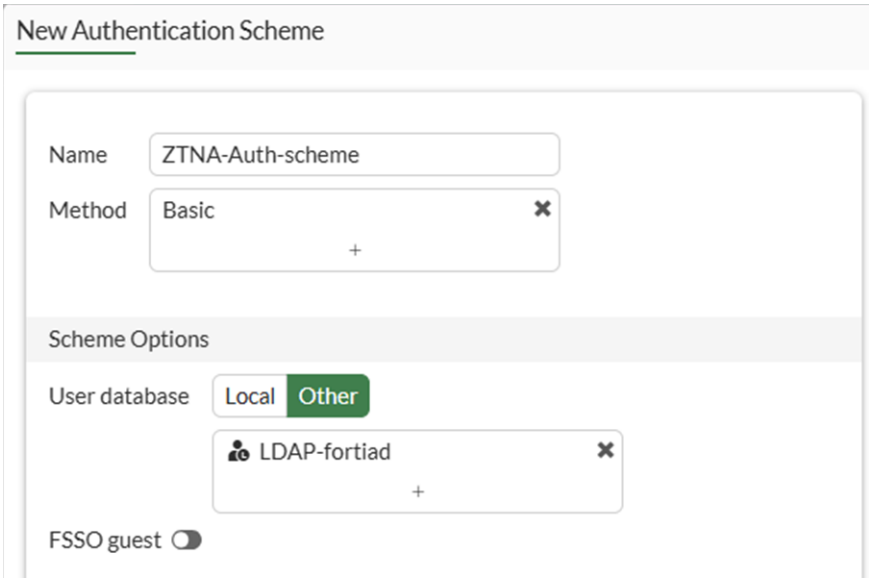
To configure authentication schemes and rules in the GUI, go to *System > Feature Visibility* and enable *Explicit Proxy*.

Authentication scheme

The authentication scheme defines the method of authentication that is applied. In this example, basic HTTP authentication is used so that users are prompted for a username and password the first time that they connect to a website through the HTTPS access proxy.

To configure an authentication scheme in the GUI:

1. Go to *Policy & Objects > Authentication* and select the *Authentication Schemes* tab.
2. Click *Create New*.
3. Set the name to *ZTNA-Auth-scheme*.
4. Set *Method* to *Basic*.
5. Set *User database* to *Other* and select *LDAP-fortiad* as the LDAP server.



6. Click *OK*.

To configure an authentication scheme in the CLI:

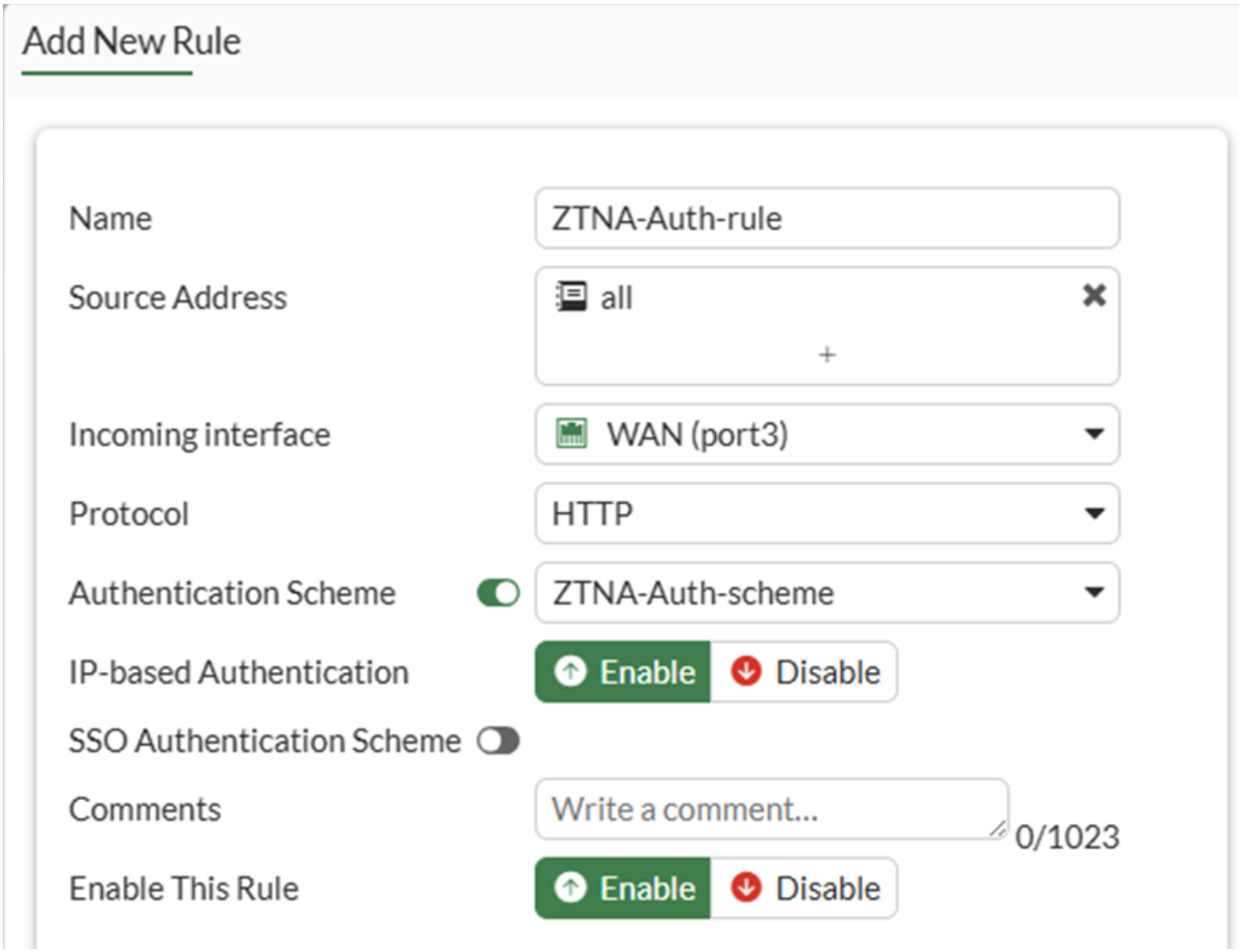
```
config authentication scheme
    edit "ZTNA-Auth-scheme"
        set method basic
        set user-database "LDAP-fortiad"
    next
end
```

Authentication rule

The authentication rule defines the proxy sources and destination that require authentication, and what authentication scheme is applied. In this example, active authentication through the basic HTTP prompt is used and applied to all sources.

To configure an authentication rule in the GUI:

1. Go to *Policy & Objects* > *Authentication* > *Authentication Rules* and click *Create New*.
2. Set the name to *ZTNA-Auth-rule*.
3. Set *Source Address* to *all*.
4. Set *Incoming interface* to *port3*.
5. Set *Protocol* to *HTTP*.
6. Enable *Authentication Scheme* and select *ZTNA-Auth-scheme*.



7. Click *OK*.

To configure an authentication rule in the CLI:

```
config authentication rule
    edit "ZTNA-Auth-rule"
        set srcintf "port3"
        set srcaddr "all"
        set active-auth-method "ZTNA-Auth-scheme"
    next
end
```

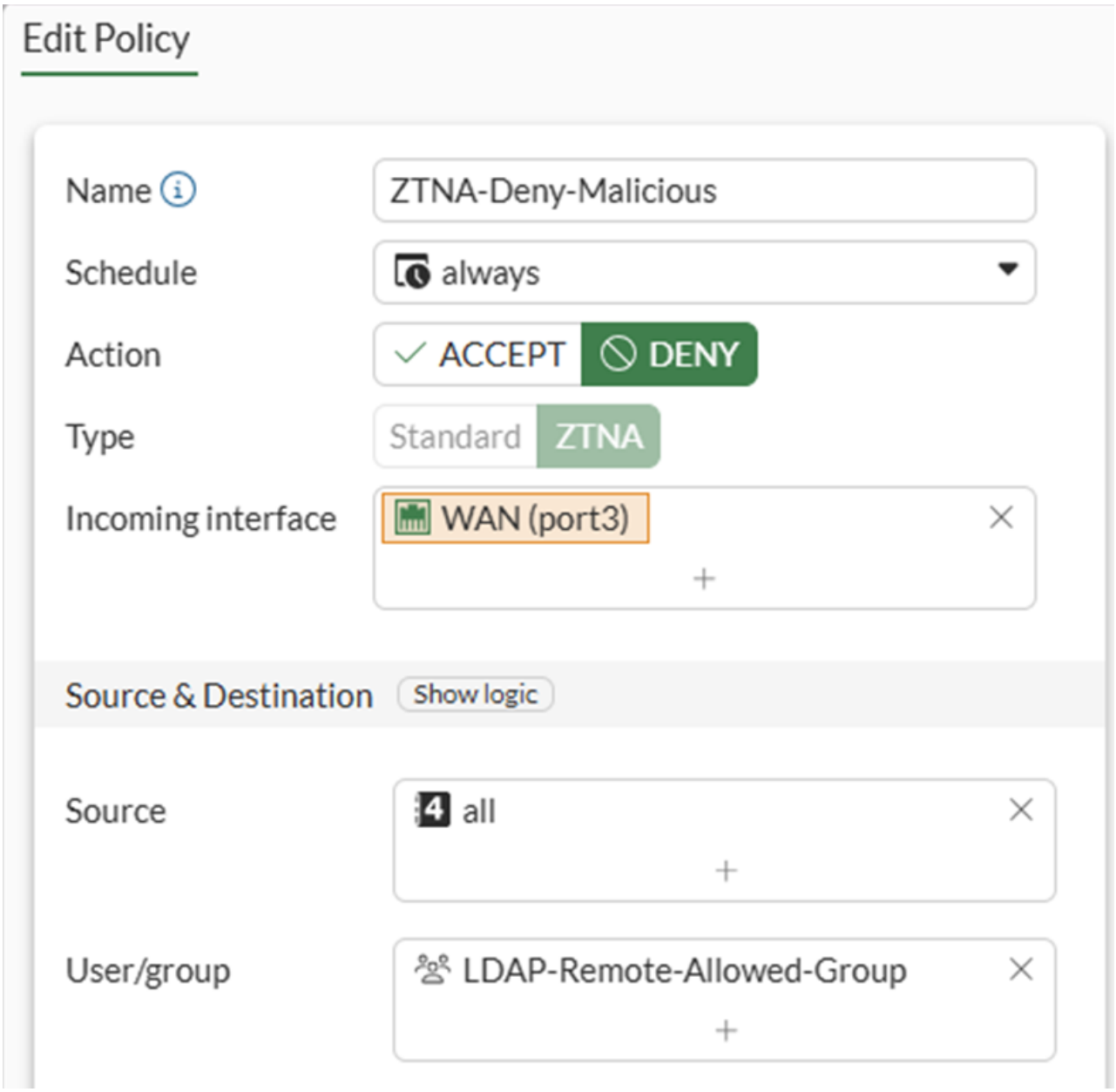
Applying the user group to a ZTNA policy

A user or user group must be applied to the ZTNA policy that you need to control user access to. The authenticated user from the authentication scheme and rule must match the user or user group in the ZTNA policy.

In this example, the user group is applied to the two simple ZTNA policies that were configured in ZTNA web proxy example on page 1677.

To apply a user group to the simple ZTNA policies in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Edit the *ZTNA-Deny-Malicious* rule.
3. In the *Source & Destination* section, set *User/group* to *LDAP-Remote-Allowed-Group*.
4. Click *OK*.
5. Edit the *ZTNA-Allow-Simple* rule.
6. In the *Source & Destination* section, set *User/group* to *LDAP-Remote-Allowed-Group*.



7. Click *OK*.

To apply a user group to the simple ZTNA policies in the CLI:

```
config firewall policy
     edit 1
        set name "ZTNA-Deny-Malicious"
        set srcintf "port3"
        set dstintf "any"
        set srcaddr "all"
        set dstaddr "WebServer:9443"
        set ztna-ems-tag "EMS1_ZTNA_Malicious-File-Detected"
        set schedule "always"
```

```
        set logtraffic all
        set groups "LDAP-Remote-Allowed-Group"
    next
    edit 2
        set name "ZTNA-Allow-Simple"
        set srcintf "port3"
        set dstintf "any"
        set action accept
        set srcaddr "all"
        set dstaddr "WebServer:9443"
        set ztna-ems-tag "EMS1_CLASS_Low"
        set schedule "always"
        set logtraffic all
        set groups "LDAP-Remote-Allowed-Group"
    next
end
```

For configuration examples using full ZTNA policy, see Configure a ZTNA policy on page 1602.

Testing remote access to the HTTPS access proxy with user authentication

Scenario 1: access allowed - user tsmith

1. On a remote Windows PC, open the FortiClient app, select the *Zero Trust Telemetry* tab, and confirm that you are connected to the EMS server.

   > It is not necessary to configure a *ZTNA Destination* on the FortiClient for the HTTPS access proxy use case. In fact, configuring a *ZTNA Destination* rule for the website may interfere with its operation.

2. In a browser, enter the address of the server and the access port.

   If entering an FQDN, make sure that DNS can resolve the address to the IP address of the FortiGate. In this example, *webserver.ztnademo.com* resolves to 10.0.3.10.
3. When the browser asks for the client certificate to use, select the EMS signed certificate, then click *OK*.

   The client certificate is verified by the FortiGate to authenticate your identity.
4. When prompted, enter the username *tsmith* and the password, and click *Sign in*.

   As *tsmith* is a member of the *Remote-Allowed-Group* group in Active Directory, it will match the *LDAP-Remote-Allowed-Group* user group. After the user authentication passes, the FortiGate performs a posture check on the ZTNA group. When that passes, you are allowed access to the website.

To verify the results:

```
#  diagnose firewall auth list

 10.0.3.2, tsmith
        type: fw, id: 0, duration: 31, idled: 13
        expire: 287, allow-idle: 300
```

```
        packets: in 8 out 7, bytes: in 1164 out 1080
        group_id: 4 3
        group_name: LDAP-Administrator LDAP-Remote-Allowed-Group

----- 1 listed, 0 filtered ------
```

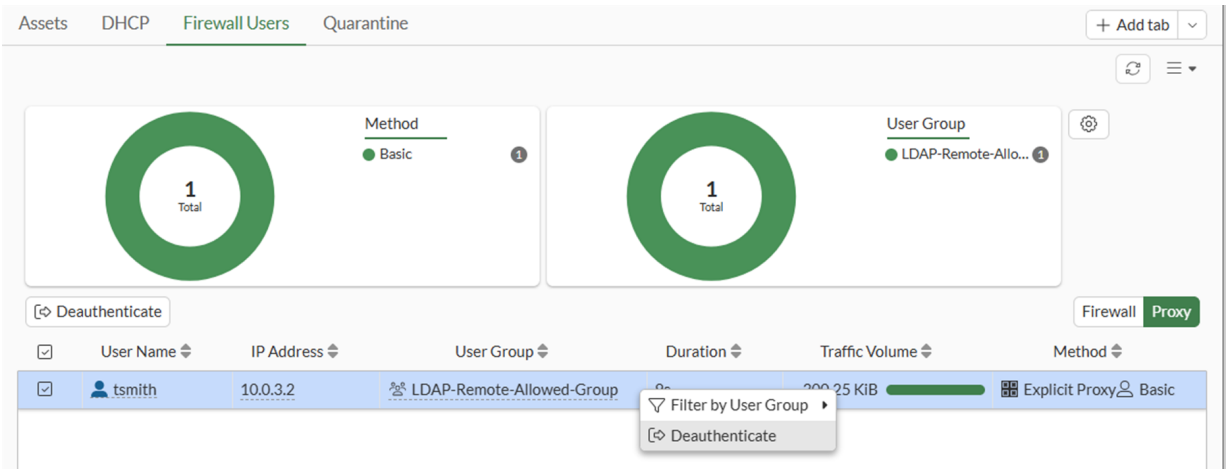
```
# diagnose test app fcnacd 7
#ID:0
UID:     9A016B5A6E914B42AD4168C066EB04CA
EMS:     FCTEMS8825007166 00000000000000000000000000000000
State:   sysinfo:1, tag:1, tagsz:1, out-of-sync:0, ref:1
Owner:   @fortiad.info
Cert SN: CC7378FBB48BCE5CFEB19CFC122B2C535AA50E5C
online:  Yes
Sub TS:  1772570201
idx:0, ttdl:1   name:all_registered_clients
idx:1, ttdl:1   name:Critical-Vulnerability
idx:3, ttdl:2   name:Low
```

The `user_name` is the windows log in username learned by FortiClient. It might not match the username used in firewall user authentication.

```
# execute log filter category 0
# execute log filter field subtype ztna
# execute log display

1: date=2026-03-04 time=07:34:42 eventtime=1772638482258291129 tz="-0800" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=1314 srcintf="port3"
srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.88.0.3 dstport=9043
dstintf="port2" dstintfrole="dmz" sessionid=22480 srcuuid="b458a65a-f759-51ea-d7df-ef2e750026d1"
dstuuid="01b612e6-174a-51f1-1a3d-b3293e8efa12" service="tcp/9043" proxyapptype="ztna-proxy"
proto=6 action="accept" policyid=2 policytype="policy" poluuid="693c3e3e-174c-51f1-3e13-
715130a1a107" policyname="ZTNA-Allow-Simple" appcat="unscanned" duration=131 user="tsmith"
group="LDAP-Remote-Allowed-Group" authserver="LDAP-fortiad" gatewayid=1 realserverid=1
vip="WebServer:9443" vipincomingip=10.0.3.10 accessproxy="ZTNA-Webserver"
clientdeviceid="9A016B5A6E914B42AD4168C066EB04CA" clientdevicemanageable="manageable"
clientdeviceems="FCTEMS8825007166" clientdevicetags="CLASS_Low/all_registered_clients"
clientcert="yes" emsconnection="online" wanin=303009 rcvdbyte=303009 wanout=4469 lanin=4247
sentbyte=4247 lanout=340700 fctuid="9A016B5A6E914B42AD4168C066EB04CA"
```

Scenario 2: access denied – user lhansen

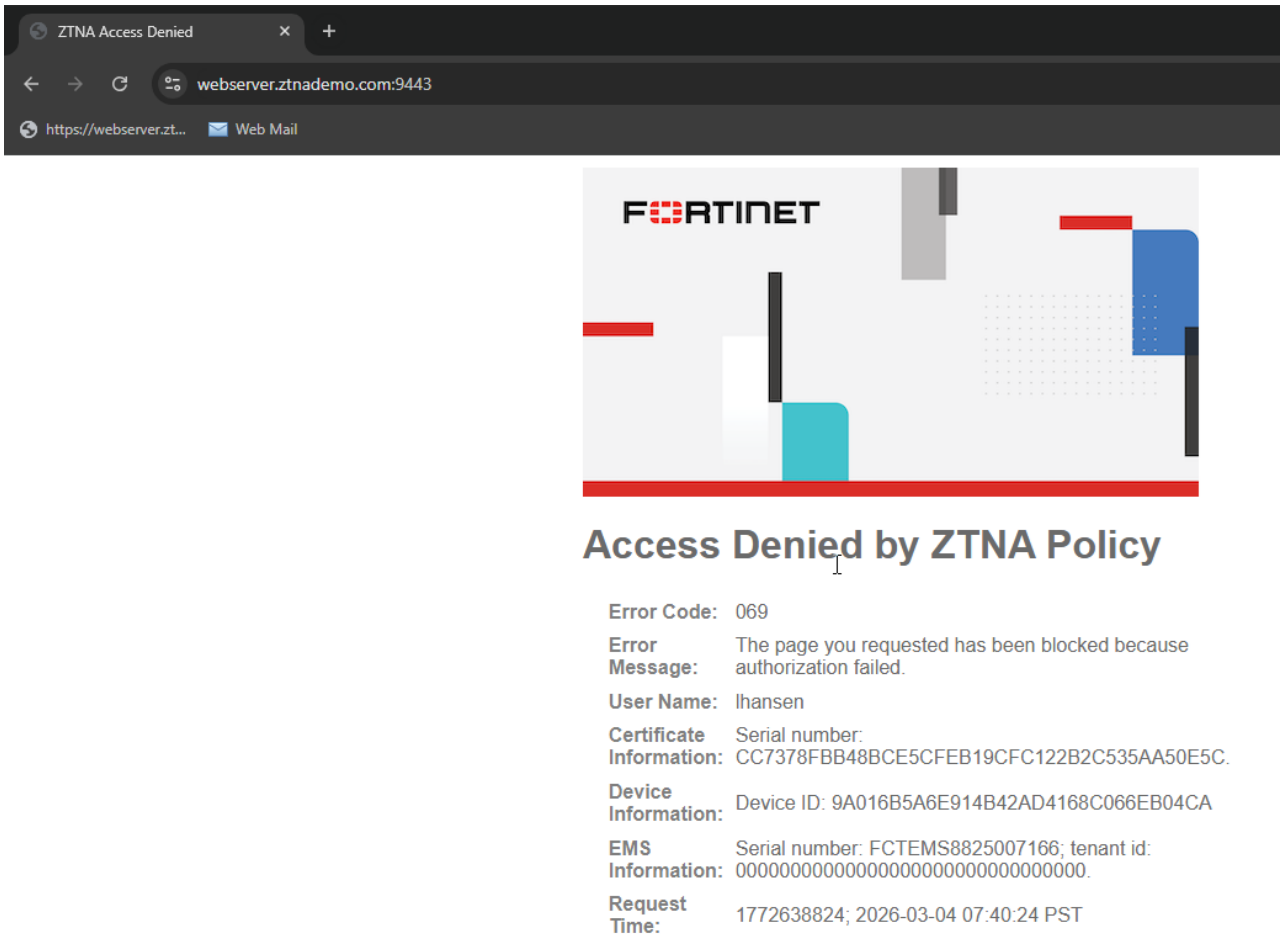
1. If scenario 1 has just been tested, log in to the FortiGate and deauthenticate the user:
   a. Go to *Dashboard > Assets & Identities > Firewall Users* and select the *Proxy* table.
   b. Right-click on the user *tsmith* and select deauthenticate.



2. On a remote Windows PC, open the FortiClient app, select the *Zero Trust Telemetry* tab, and confirm that you are connected to the EMS server.
3. In a browser, enter the address *webserver.ztnademo.com*.
4. When the browser asks for the client certificate to use, select the EMS signed certificate, then click *OK*. This option might not appear if you have already selected the certificate when testing scenario 1.

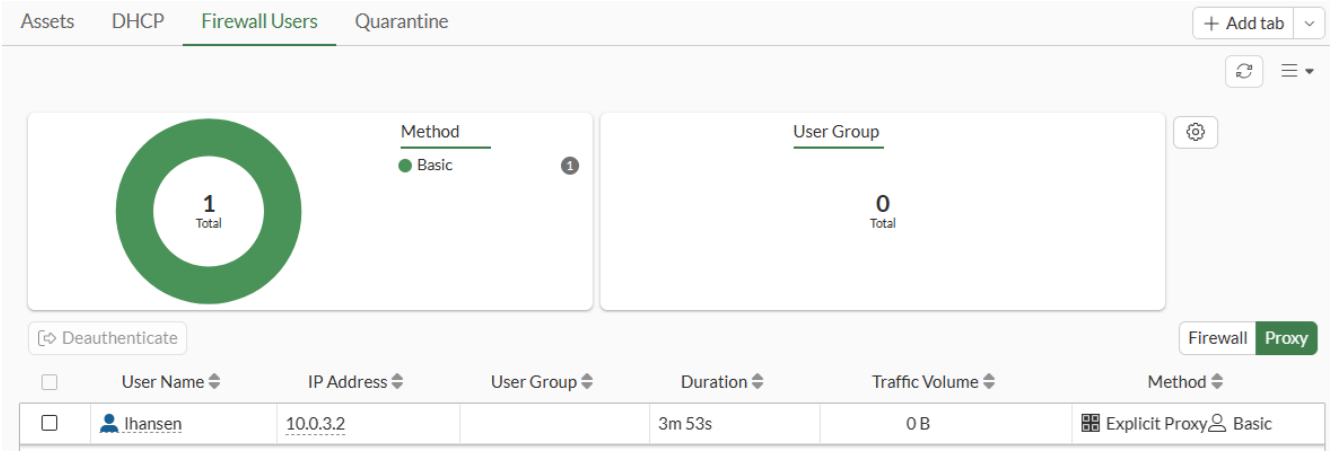
   The client certificate is verified by the FortiGate to authenticate your identity.
5. When prompted, enter the username *lhansen* and the password, and click *Sign in*.

   As *lhansen* is not a member of the *Remote-Allowed* group in Active Directory, it will not match the *LDAP-Remote-Allowed-Group* user group. Because no other policies are matched, this user is implicitly denied



To verify the results:

Go to *Dashboard* > *Assets & Identities* > *Firewall Users* and select the *Proxy* table. Confirm that user *lhansen* is listed, but no applicable user group is returned.



```
# execute log display

1: date=2026-03-04 time=07:41:25 eventtime=1772638885574732337 tz="-0800" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=1355 srcintf="port3"
srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.0.3.10 dstport=9443
dstintf="root" dstintfrole="undefined" sessionid=22618 srcuuid="b458a65a-f759-51ea-d7df-
ef2e750026d1" dstuuid="01b612e6-174a-51f1-1a3d-b3293e8efa12" service="tcp/9443"
proxyapptype="ztna-proxy" proto=6 action="deny" policyid=0 policytype="proxy-policy"
appcat="unscanned" duration=61 user="lhansen" authserver="LDAP-fortiad" vip="WebServer:9443"
vipincomingip=10.0.3.10 clientdeviceid="9A016B5A6E914B42AD4168C066EB04CA"
clientdevicemanageable="manageable" clientdeviceems="FCTEMS8825007166" clientdevicetags="CLASS_
Low/all_registered_clients" clientcert="yes" emsconnection="online" wanin=0 rcvdbyte=0 wanout=0
lanin=3866 sentbyte=3866 lanout=39507 fctuid="9A016B5A6E914B42AD4168C066EB04CA" msg="Traffic
denied because failed to match a policy or proxy-policy" crscore=30 craction=131072 crlevel="high"
```

ZTNA traffic forwarding proxy example

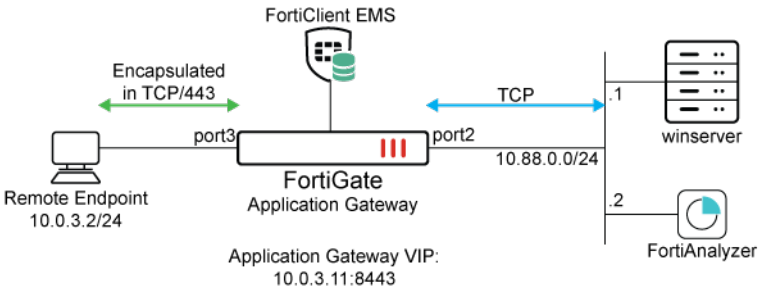
In this example, a traffic forwarding proxy (TFP) is configured to demonstrate an HTTPS reverse proxy that forwards TCP traffic to the designated resource. The access proxy tunnels TCP traffic between the client and the FortiGate over HTTPS, and forwards the TCP traffic to the protected resource. It verifies user identity, device identity, and trust context, before granting access to the protected source.

The TFP supports communication between the client and the access proxy without SSL/TLS encryption. The connection still begins with a TLS handshake. The client uses the HTTP 101 response to switch protocols and remove the HTTPS stack. Further end to end communication between the client and server are encapsulated in the specified TCP port, but not encrypted by the access proxy. This improves performance by reducing the overhead of encrypting an already secured underlying protocol, such as RDP, SSH, or FTPS. Users should still enable the encryption option for end to end protocols that are insecure.

In this example, RDP (Remote Desktop Protocol) and SMB (Server Message Block) protocol access are configured to one server, and SSH access to the other server. Encryption is disabled for RDP and SSH, and enabled for SMB.

FortiClient (Windows) must be running 7.0.3 or later to detect SMB.

You cannot use ZTNA connection rules and TCP forwarding on a Windows 7 endpoint.



This example assumes that the FortiGate EMS fabric connector is already successfully connected.

To configure the ZTNA server for TFP in the GUI:

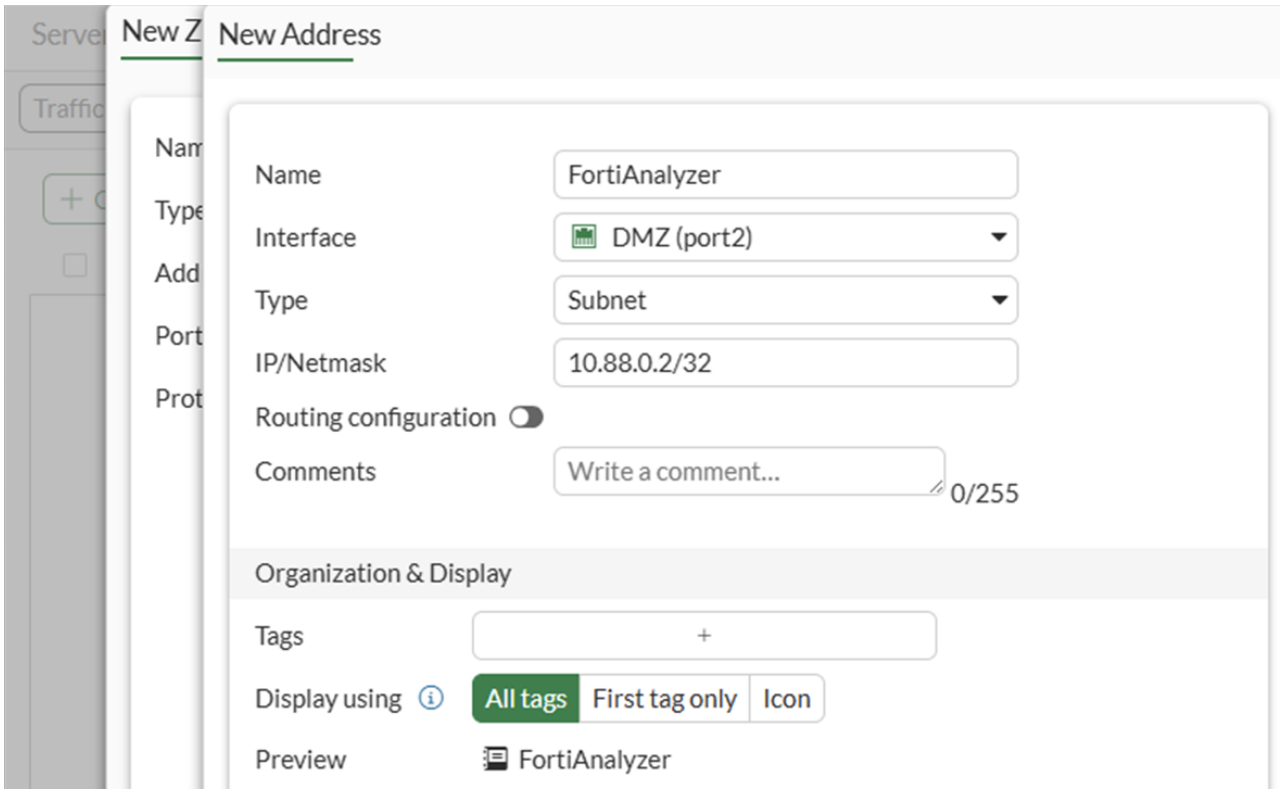
1. Go to *Policy & Objects* > *ZTNA* > *Traffic Forwarding* > *Traffic Forwarding Server*.
2. Click *Create new*.
3. Set *Name* to *ZTNA-tcp-server*.
4. Set *Host* to *10.0.3.11*.
5. Set *ZTNA port* to *Create new* and configure the port:

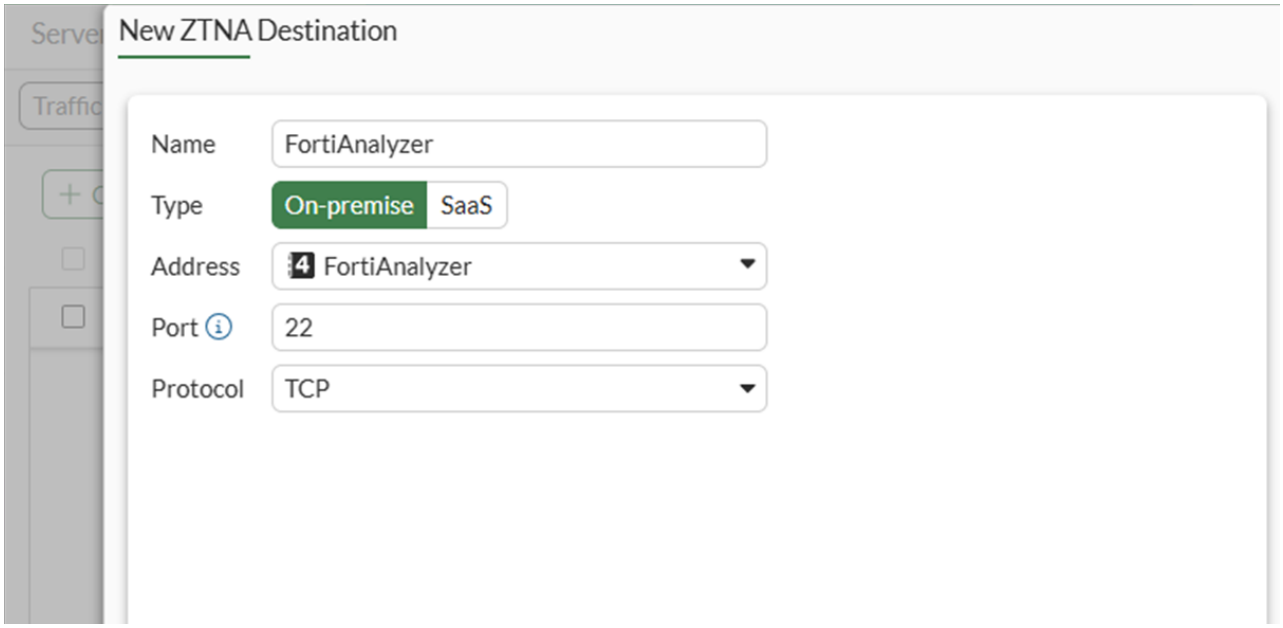
Option	Value
Interface	port3
IP address	10.0.3.11
Port	8443
Default certificate	This example uses a wildcard certificate with a Common Name (CN): *.ztnademo.com. This certificate is trusted by the endpoints which will access the ZTNA port.
Name	TCPserver:8443

Verify that the IP address and port do not conflict with management access to the interface. Otherwise, change the IP address to another address on that subnet.

6. Click *OK* to save the traffic forwarding server.
7. Select the *Destination* view to define a destination.
8. Click *Create new*.
9. Configure the ZTNA destination:

Option	Value
Name	FortiAnalyzer
Type	On-premise
Address	Use the dropdown and then the + button to define a new address object with: 10.88.0.2/32.
Port	22
Protocol	TCP

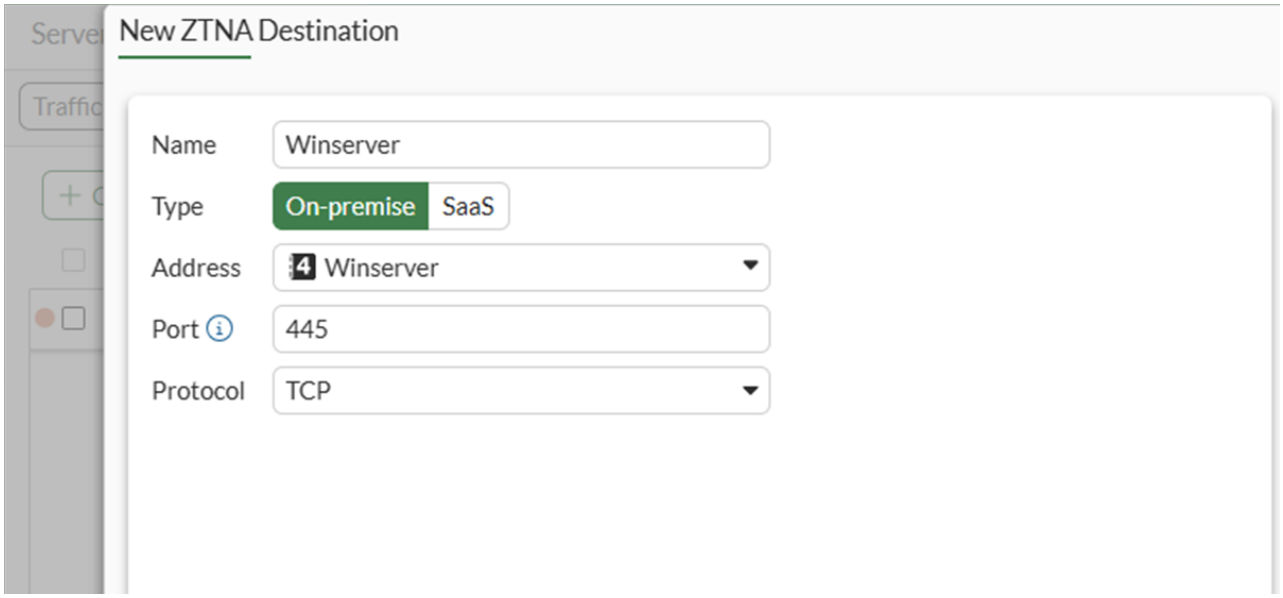




10. Click *OK* to save the ZTNA destination.
11. Click *Create New* again to define a second destination for the winserver server at 10.88.0.1 with ports 445 and 3389 to correspond to SMB and RDP:

Option	Value
Name	Winserver

Option	Value
Type	On-premise
Address	Use the dropdown and then the + button to define a new address object with: 10.88.0.1/32.
Port	445 It is not possible to specify two non-consecutive ports for a destination in the GUI. The second port, 3389, must be appended using the CLI.
Protocol	TCP



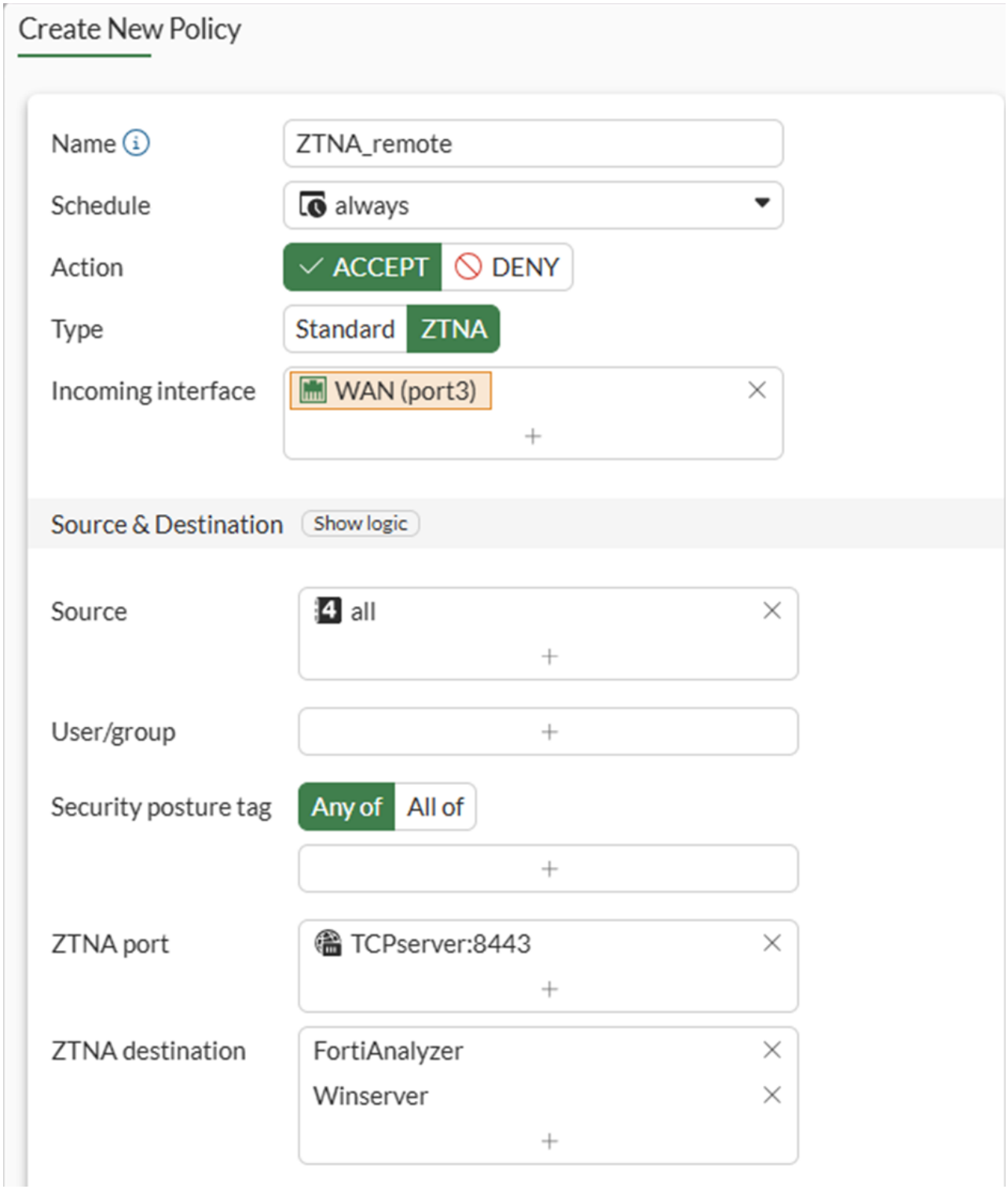
12. Click *OK* to save the ZTNA destination.
13. Use the CLI to add the second port:

```
config ztna destination
     edit "Winserver"
        set mappedport 445 3389
     next
 end
```

To configure a simple ZTNA policy to allow traffic to the TCP access proxy in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy*, and click *Create New*.
2. Set *Name* to *ZTNA_remote*.
3. Set *Type* to *ZTNA*.
4. Set *Incoming Interface* to *port3*.
5. Set *Source* to *all*.
6. Set *ZTNA port* to *TCPserver:8443*.

7. Set *ZTNA destination* to *FortiAnalyzer* and *Winserver*.
8. Set *Log allowed traffic* to *All sessions*.
9. Configure the remaining options as needed.



10. Click *OK*.

To configure a ZTNA traffic forwarding proxy in the CLI:

1. Configure the access proxy VIP:

```
config firewall vip
    edit "TCPserver:8443"
        set type access-proxy
        set server-type https
        set extip 10.0.3.11
        set extintf "port3"
        set extport 8443
        set ssl-certificate "ztna-wildcard"
    next
end
```

2. Configure the destinations:

```
config ztna destination
    edit "Winserver"
        set address "Winserver"
        set mappedport 445 3389
        set protocol TCP
    next
    edit "FortiAnalyzer"
        set address "FortiAnalyzer"
        set mappedport 22
        set protocol TCP
    next
end
```

3. Configure the virtual host:

```
config firewall access-proxy-virtual-host
     edit "10.0.3.11_host"
         set ssl-certificate “ztna-wildcard”
         set host “10.0.3.11”
     next
 end
```

> The virtual host is configured automatically when specifying the Host field during the Traffic Forwarding Server creation.

4. Configure the traffic forward proxy:

```
config ztna traffic-forward-proxy
    edit "ZTNA-tcp-server"
        set vip "TCPserver:8443"
        set host "10.0.3.11_host"
    next
end
```

5. Configure Firewall Address Objects and ZTNA policy:

```
config firewall address
     edit "FortiAnalyzer"
         set subnet 10.88.0.2 255.255.255.255
     next
     edit "Winserver"
         set subnet 10.88.0.1 255.255.255.255
     next
end
```

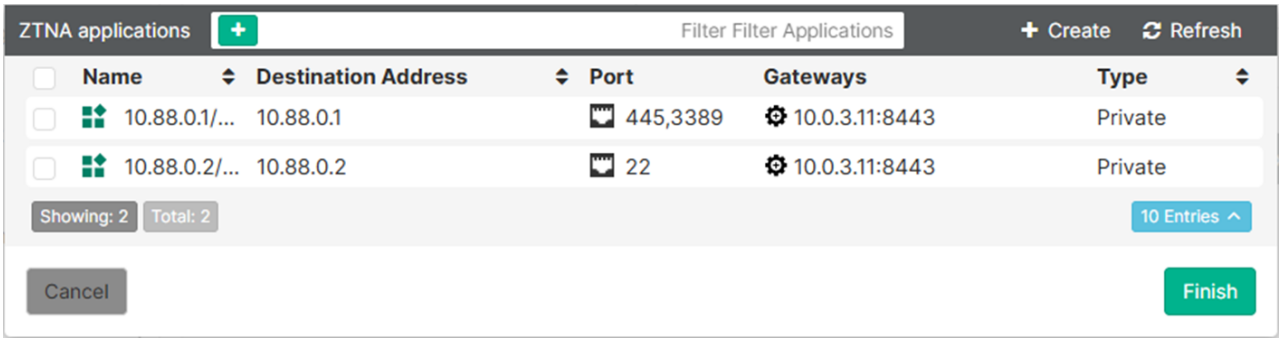
```
config firewall policy
    edit 1
        set name "ZTNA_remote"
        set srcintf "port3"
        set dstintf "any"
        set action accept
        set srcaddr "all"
        set dstaddr "TCPserver:8443"
        set ztna-destination "FortiAnalyzer" "Winserver"
        set schedule "always"
        set logtraffic all
        set nat enable
    next
end
```

Test the connection to the access proxy

Before connecting, users must have a ZTNA connection rule in FortiClient. This will be configured using EMS.

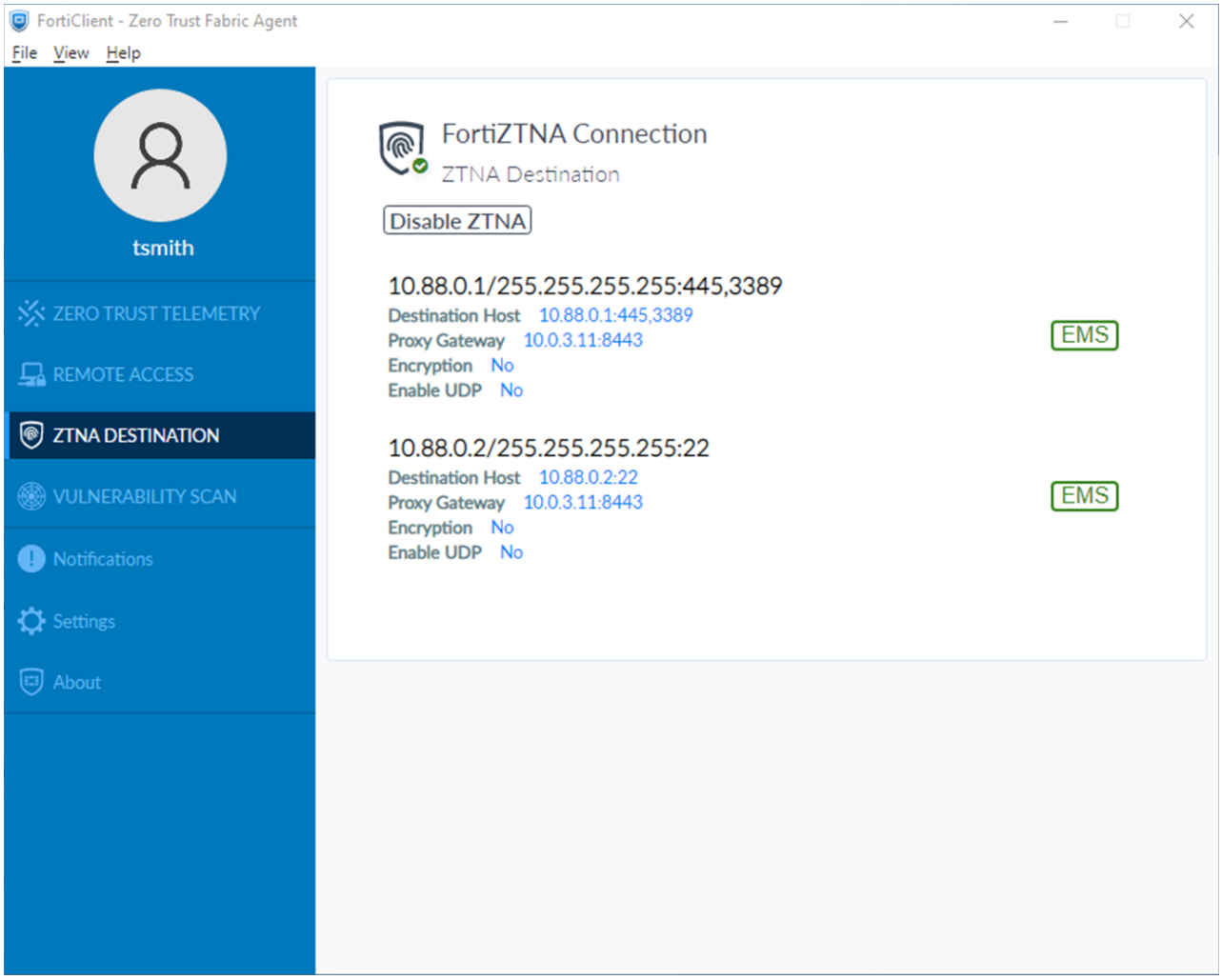
To create a ZTNA Destination in EMS:

1. Go to *Endpoint Profiles* > *ZTNA Destinations*.
2. Select the required profile, or create a new one.
3. Set *Name* to the desired name.
4. In the *Rules* section click *Add*.
   Two ZTNA applications should populate that were learned from the FortiGate.



5. Select both applications then click *Finish*.
   The two destinations now appear as rule entries.
6. Click *Save* to save the ZTNA Destinations profile.

7. Ensure that this profile is applied to the endpoint policy assigned to the remote endpoint.
   After some time, the FortiClient will receive the profile update and the configuration will appear under the ZTNA DESTINATION tab on FortiClient.

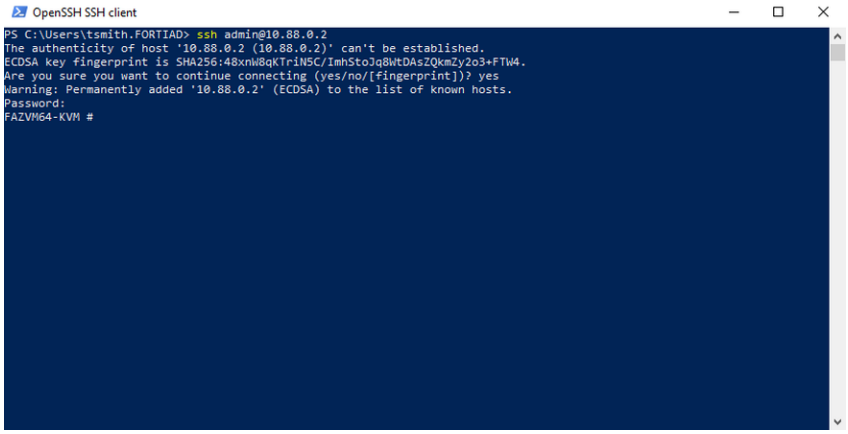


To test the access proxy:

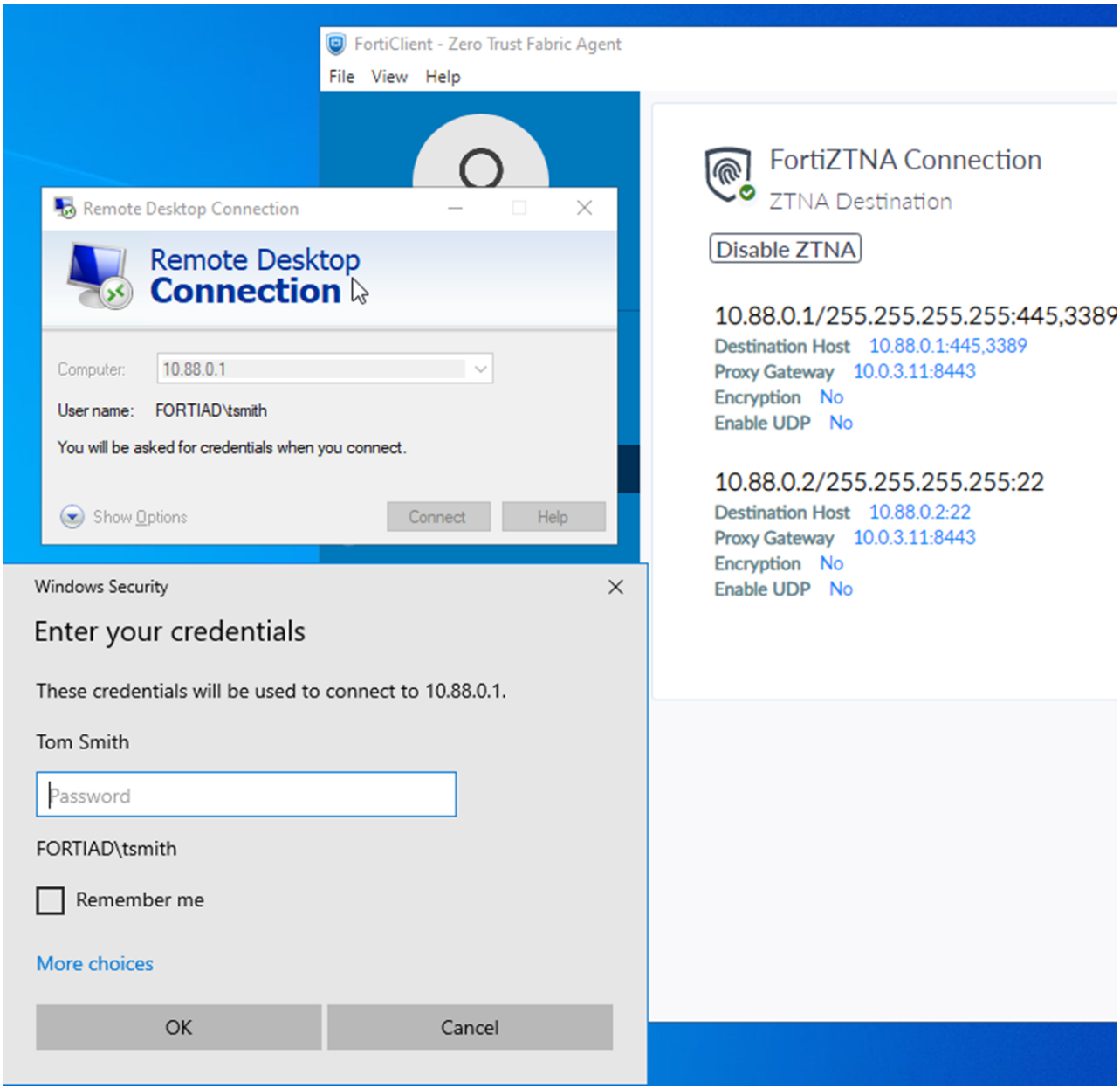
1. Test SSH connectivity to the FortiAnalyzer by opening a SSH client, such as PowerShell:

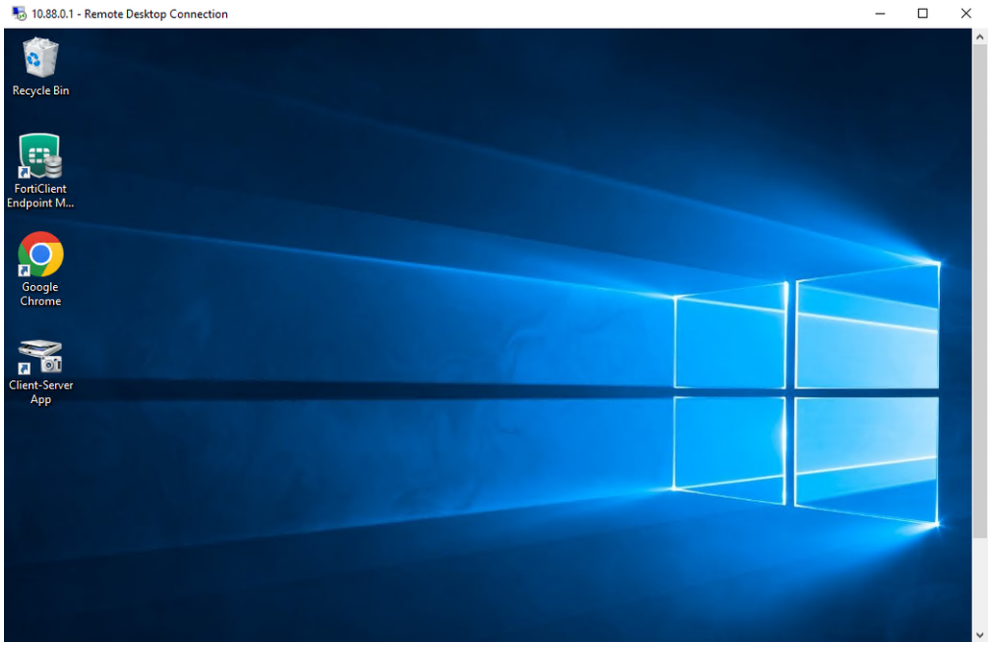
```
ssh admin@10.88.0.2
```

2. If it is the first time that you are connecting, accept the ECDSA key fingerprint of the FortiAnalyzer.

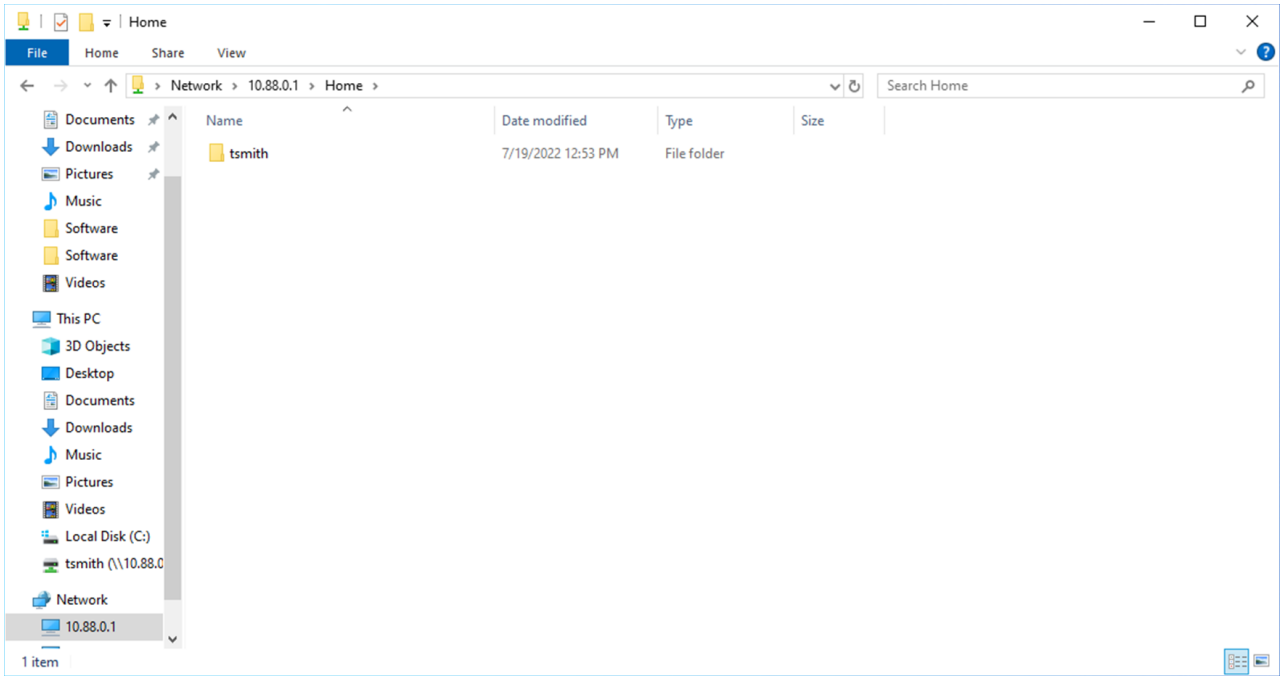


3. Test RDP by opening the Remote Desktop Connection application and providing the address to RDP to (10.88.0.1) and the username and password:


FortiClient - Zero Trust Fabric Agent
File View Help
FortiZTNA Connection
ZTNA Destination
Disable ZTNA
10.88.0.1/255.255.255.255:445,3389
Destination Host 10.88.0.1:445,3389
Proxy Gateway 10.0.3.11:8443
Encryption No
Enable UDP No
10.88.0.2/255.255.255.255:22
Destination Host 10.88.0.2:22
Proxy Gateway 10.0.3.11:8443
Encryption No
Enable UDP No
Remote Desktop Connection
Remote Desktop Connection
Computer: 10.88.0.1
User name: FORTIAD\tsmith
You will be asked for credentials when you connect.
Show Options
Connect
Help
Windows Security
Enter your credentials
These credentials will be used to connect to 10.88.0.1.
Tom Smith
Password
FORTIAD\tsmith
Remember me
More choices
OK
Cancel



4. Test SMB access by opening File Explorer and entering `\\10.88.0.1\Home`:



Logs

```
# execute log filter category 0
# execute log filter field subtype ztna
# execute log display
```

SSH:

```
1: date=2026-03-05 time=10:05:32 eventtime=1772733931608530952 tz="-0800" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=3638 srcintf="port3"
srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.88.0.2 dstport=22
dstintf="port2" dstintfrole="dmz" sessionid=40549 srcuuid="b458a65a-f759-51ea-d7df-ef2e750026d1"
dstuuid="488b7d46-17ea-51f1-0341-09c9959801a7" service="SSH" proxyapptype="ztna-proxy" proto=6
action="accept" policyid=1 policytype="policy" poluuid="33483cfe-18bb-51f1-bede-ba49af71cc72"
policyname="ZTNA_remote" appcat="unscanned" duration=50 vip="TCPserver:8443"
vipincomingip=10.0.3.11 accessproxy="ZTNA-tcp-server" ztnadestination="FortiAnalyzer"
clientdevicemanageable="manageable" clientcert="yes" wanin=2917 rcvdbyte=2917 wanout=2301
lanin=4270 sentbyte=4270 lanout=6046
```

Logs will not generate until the session is complete. Exit the SSH session by pressing *Ctrl+C*.

RDP:

```
11: date=2026-03-05 time=10:10:54 eventtime=1772734254005541566 tz="-0800" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=3649 srcintf="port3"
srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.88.0.1 dstport=3389
dstintf="port2" dstintfrole="dmz" sessionid=40642 srcuuid="b458a65a-f759-51ea-d7df-ef2e750026d1"
dstuuid="488b7d46-17ea-51f1-0341-09c9959801a7" service="RDP" proxyapptype="ztna-proxy" proto=6
action="accept" policyid=1 policytype="policy" poluuid="33483cfe-18bb-51f1-bede-ba49af71cc72"
policyname="ZTNA_remote" appcat="unscanned" duration=39 vip="TCPserver:8443"
vipincomingip=10.0.3.11 accessproxy="ZTNA-tcp-server" ztnadestination="Winserver"
clientdevicemanageable="manageable" clientcert="yes" wanin=320490 rcvdbyte=320490 wanout=73972
lanin=75943 sentbyte=75943 lanout=323619
```

SMB:

```
2: date=2026-03-05 time=10:41:51 eventtime=1772736111610076560 tz="-0800" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=3728 srcintf="port3"
srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.88.0.1 dstport=445
dstintf="port2" dstintfrole="dmz" sessionid=41351 srcuuid="b458a65a-f759-51ea-d7df-ef2e750026d1"
dstuuid="488b7d46-17ea-51f1-0341-09c9959801a7" service="SMB" proxyapptype="ztna-proxy" proto=6
action="accept" policyid=1 policytype="policy" poluuid="33483cfe-18bb-51f1-bede-ba49af71cc72"
policyname="ZTNA_remote" appcat="unscanned" duration=0 vip="TCPserver:8443"
vipincomingip=10.0.3.11 accessproxy="ZTNA-tcp-server" ztnadestination="Winserver"
clientdevicemanageable="manageable" clientcert="yes" wanin=0 rcvdbyte=0 wanout=0 lanin=1970
sentbyte=1970 lanout=3129
```

ZTNA traffic forwarding proxy with FQDN example

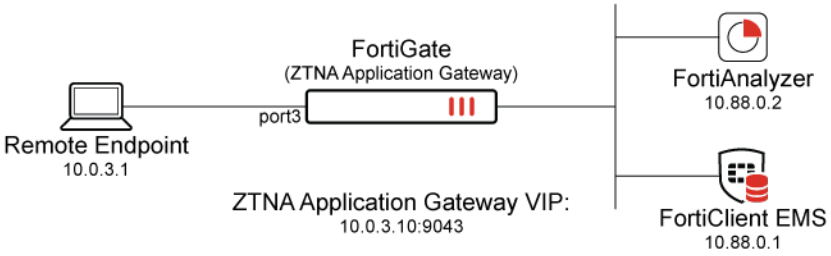
When defining ZTNA connection rules on FortiClient for TCP forwarding, it is sometimes desirable to configure the destination host address as an FQDN address instead of an IP address. Since the real servers are often servers in the corporate network, this layer of obfuscation prevents internal IPs from easily leaking to the public, and also makes the destination more easily recognizable by the end users.

One obstacle to overcome is getting remote hosts to resolve an internal FQDN that is typically only resolvable by an internal DNS in the corporate network. This can be solved with the following:

1. When an FQDN address is added in FortiClient's ZTNA Destination, FortiClient will intercept connections destined for the FQDN address and replace the destination with a special IP address (such as 10.235.0.1).
2. FortiClient listens to any traffic destined for the FQDN and its port and forwards the traffic using the TCP forwarding URL with FQDN to the ZTNA application gateway.
3. The ZTNA application gateway will resolve the FQDN, matching the traffic to the ZTNA real server configuration with the same domain and address.
4. If a valid ZTNA real server entry is found, traffic is forwarded to the real server.

Example

In this example, a FortiAnalyzer in the internal network is added to the FortiGate access proxy for TCP forwarding. A ZTNA Destination is configured on the FortiClient, with the destination host field pointing to the FQDN addresses of the internal servers. The FQDN address is also resolvable by the FortiGate and the same FQDN is used in the real server mapping.



This example assumes that the FortiGate EMS Fabric connector is already successfully connected.

This features requires a minimum FortiClient and FortiClient EMS version of 7.0.3.

To configure the TCP forwarding proxy:

1. Go to *Policy & Objects* > *ZTNA* > *Traffic Forwarding* > *Traffic Forwarding Server*.
2. Click *Create new*.
3. Set *Name* to *10.0.3.11*.
4. Set *Host* to *10.0.3.11*.
   This address must exactly match the Proxy Gateway configured on the FortiClient. It is not possible to use a FQDN or hostname that would resolve to the Proxy Gateway IP address.
5. Set *ZTNA port* to *Create new* and configure the port:

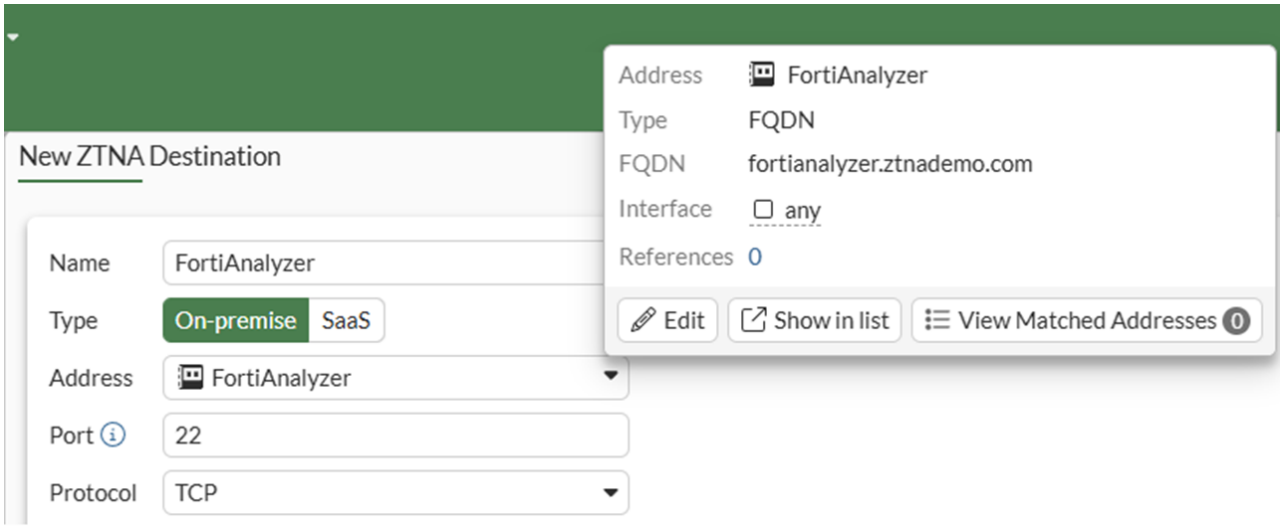
Option	Value
Interface	port3
IP address	10.0.3.11
Port	9043
Default certificate	This example uses a wildcard certificate with a Common Name (CN): **.ztnademo.com*.

Option	Value
	This certificate is trusted by the endpoints which will access the ZTNA port.
Name	10.0.3.11:9043

Verify that the IP address and port do not conflict with management access to the interface. Otherwise, change the IP address to another address on that subnet.

6. Click *OK* to save the traffic forwarding server.
7. Select the *Destination* view to define a destination.
8. Click *Create new*.
9. Configure the ZTNA destination:

Option	Value
Name	FortiAnalyzer
Type	On-premise
Address	Use the dropdown and then the + button to define a new address object: • *Type*: *FQDN* • *FQDN*: *FortiAnalyzer*
Port	22
Protocol	TCP



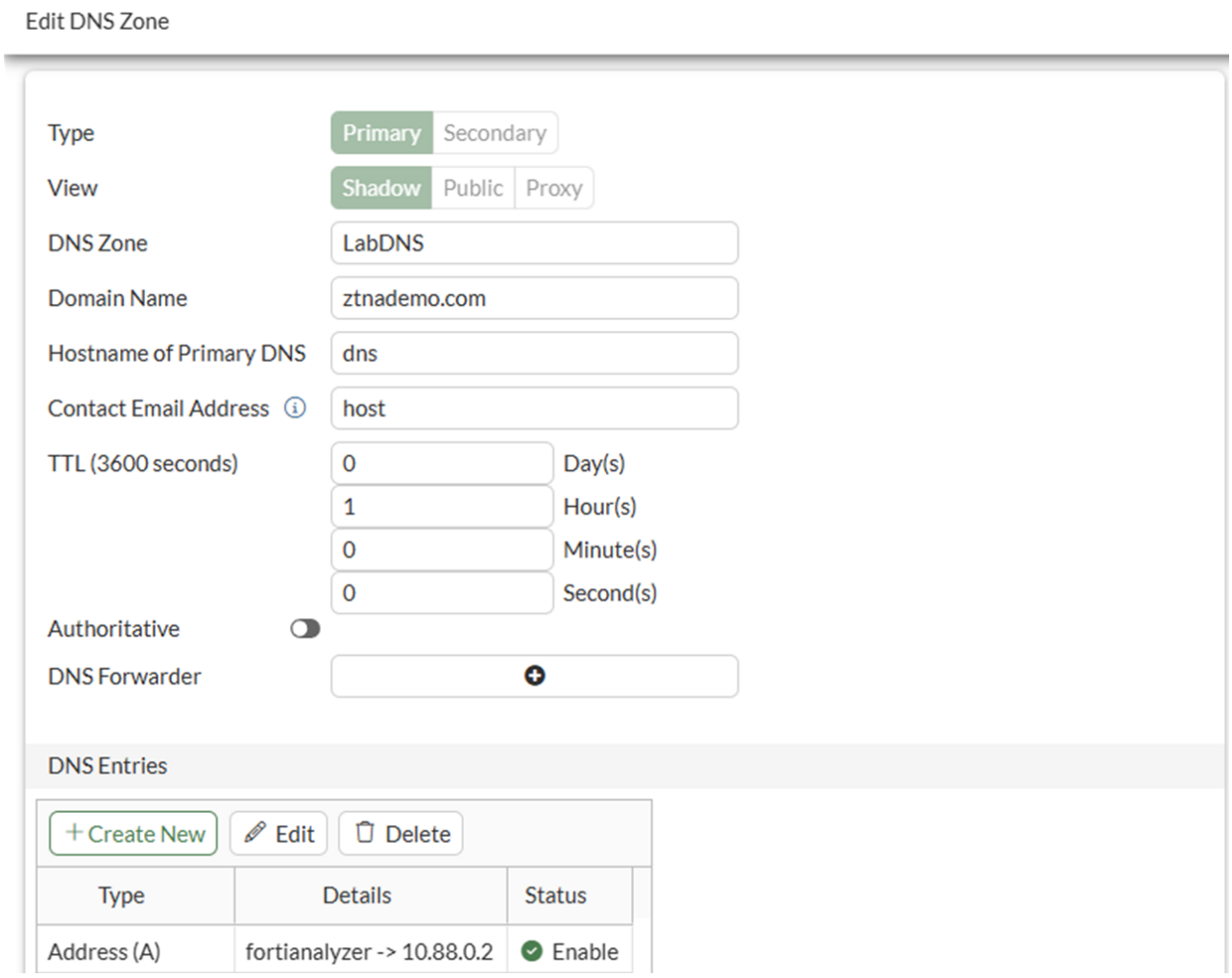
10. Click *OK* to save the ZTNA destination.

To configure the FQDN resolution:

The FQDN *fortianalyzer.ztnademo.com* does not currently resolve to anything. A DNS entry is required for the FortiGate to resolve the FQDN to the FortiAnalyzer IP address of 10.88.0.2. In this example, the FortiGate itself is functioning as the DNS server and a DNS entry must be added to map the FQDN to an IP address.

1. Go to *Network* > *DNS Servers*.
2. Edit the ztnademo.com domain DNS database. If necessary, create a new database for the domain.
3. In the *DNS Entries* table click *Create New*.
4. Configure the DNS entry:

Option	Value
Type	Address (A)
Hostname	fortianalyzer
Fully Qualified Domain Name (FQDN)	(auto-populates)
IP Address	10.88.0.2



5. Click *OK* to save the DNS entry.
6. Click *OK* to save the DNS zone.
7. Verify the resolution using the CLI:

```
# execute ping fortianalyzer.ztnademo.com
PING fortianalyzer.ztnademo.com (10.88.0.2): 56 data bytes
64 bytes from 10.88.0.2: icmp_seq=0 ttl=64 time=0.5 ms
64 bytes from 10.88.0.2: icmp_seq=1 ttl=64 time=0.5 ms
^C
--- fortianalyzer.ztnademo.com ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 0.5/0.5/0.5 ms
```

To configure the ZTNA rule:

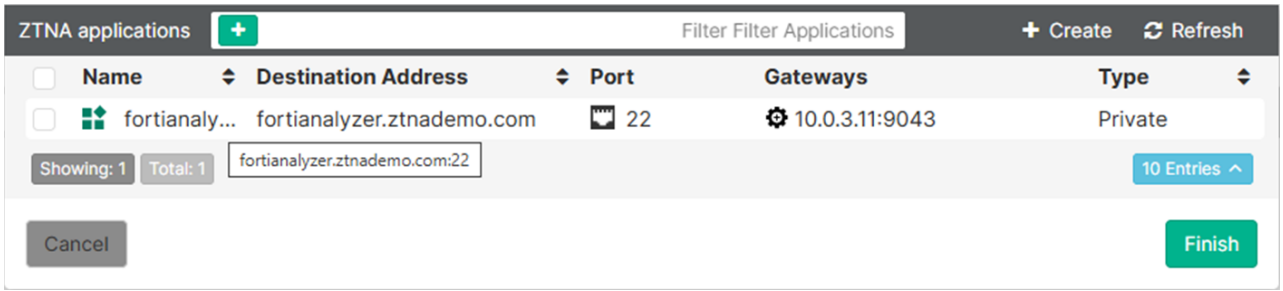
1. Go to *Policy & Objects* > *Firewall Policy* and click *Create new*.
2. Set *Name* to *ZTNA-Admin-Access*.
3. Set *Type* to *ZTNA*.
4. Set *Incoming Interface* to *WAN (port3)*.
5. Set *Source* to *all*.
6. Set *ZTNA port* to *10.0.3.11:9043*.
7. Set *ZTNA destination* to *FortiAnalyzer*.
8. Set *Log allowed traffic* to *All sessions*.
9. Configure the remaining options as needed.
10. Click *OK*.

Testing the connection to the TCP forwarding access proxy

Before connecting, users must have a ZTNA Destination in FortiClient.

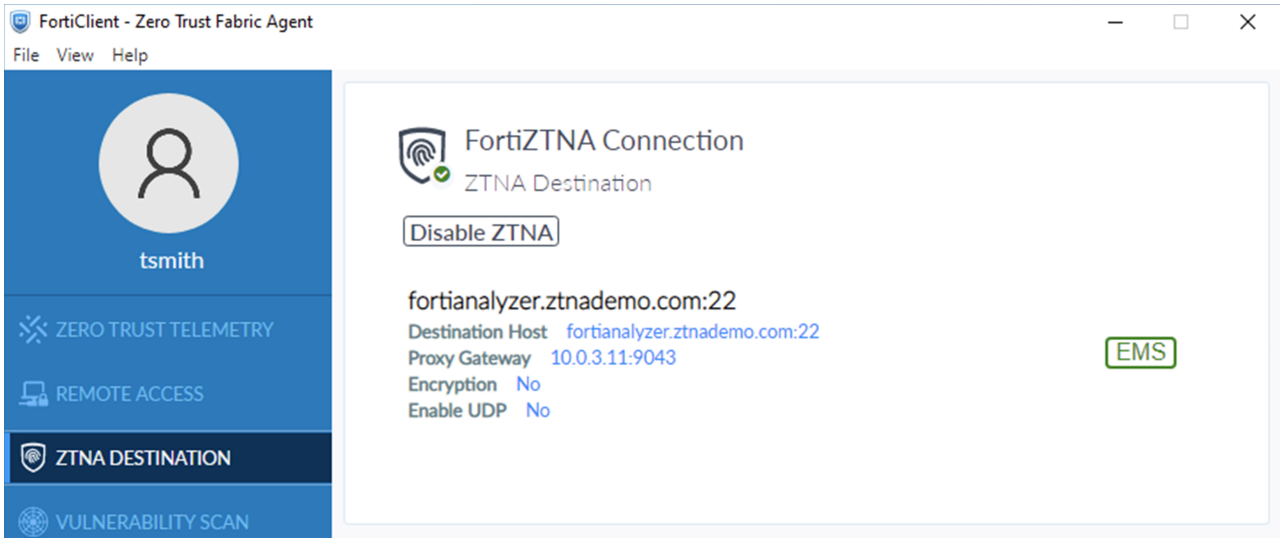
To create the ZTNA rules in EMS:

1. Go to *Endpoint Profiles* > *ZTNA Destinations*.
2. Select the required profile, or create a new one.
3. Set *Name* to the desired name.
4. In the *Rules* section click *Add*.
   The ZTNA applications will display the fortianalyzer.ztnademo.com:22 destination created earlier on the FortiGate. You can hover over the name field to show the full name.



5. Select the entry and click *Finish*.
   The destination now appears as a rule entry.
6. Click *Save* to save the ZTNA Destinations profile.

7. Ensure that this profile is applied to the endpoint policy assigned to the remote endpoint.
   After some time, the FortiClient will receive the profile update and the configuration will appear under the ZTNA DESTINATION tab on FortiClient.



To test the access proxy:

1. Verify the resolution for fortianalyzer.ztnademo.com on the remote endpoint using ping:

```
>ping fortianalyzer.ztnademo.com

Pinging fortianalyzer.ztnademo.com [10.235.0.1] with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 10.235.0.1:
       Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),
```

   The FortiClient is resolving fortianalyzer.ztnademo.com to a special address, overriding any DNS resolution for this host. FortiClient will listen to the traffic to this FQDN and forward it to the Proxy Gateway.
2. Test SSH access from the remote endpoint using an SSH client, such as PowerShell:

```
ssh admin@fortianalyzer.ztnademo.com
```

3. Enter the password when prompted.
4. After successfully connecting, ensure that you end the session to generate a log on the FortiGate.

FortiGate logs

On the FortiGate, go to *Log & Report > ZTNA Traffic* to view the logs. Alternatively, use the CLI to display the most recent ZTNA logs:

```
# execute log filter category 0
# execute log filter field subtype ztna
# execute log display
```

Successful log

```
1: date=2026-03-06 time=11:04:00 eventtime=1772823839937353369 tz="-0800" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=20933
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.88.0.2
dstport=22 dstintf="port2" dstintfrole="dmz" sessionid=61153 srcuuid="b458a65a-f759-51ea-d7df-
ef2e750026d1" dstuuid="88ae5cb6-196d-51f1-c58c-322876d641bb" service="SSH" proxyapptype="ztna-
proxy" proto=6 action="accept" policyid=1 policytype="policy" poluuid="6d795be4-1972-51f1-e6ab-
ed4c5318bd15" policyname="ZTNA-Admin-Access" appcat="unscanned" duration=3 vip="10.0.3.11:9043"
vipincomingip=10.0.3.11 accessproxy="ZTNA-10.0.3.11" ztnadestination="FortiAnalyzer"
clientdevicemanageable="manageable" clientcert="yes" wanin=1629 rcvdbyte=1629 wanout=1693
lanin=3679 sentbyte=3679 lanout=4758
```

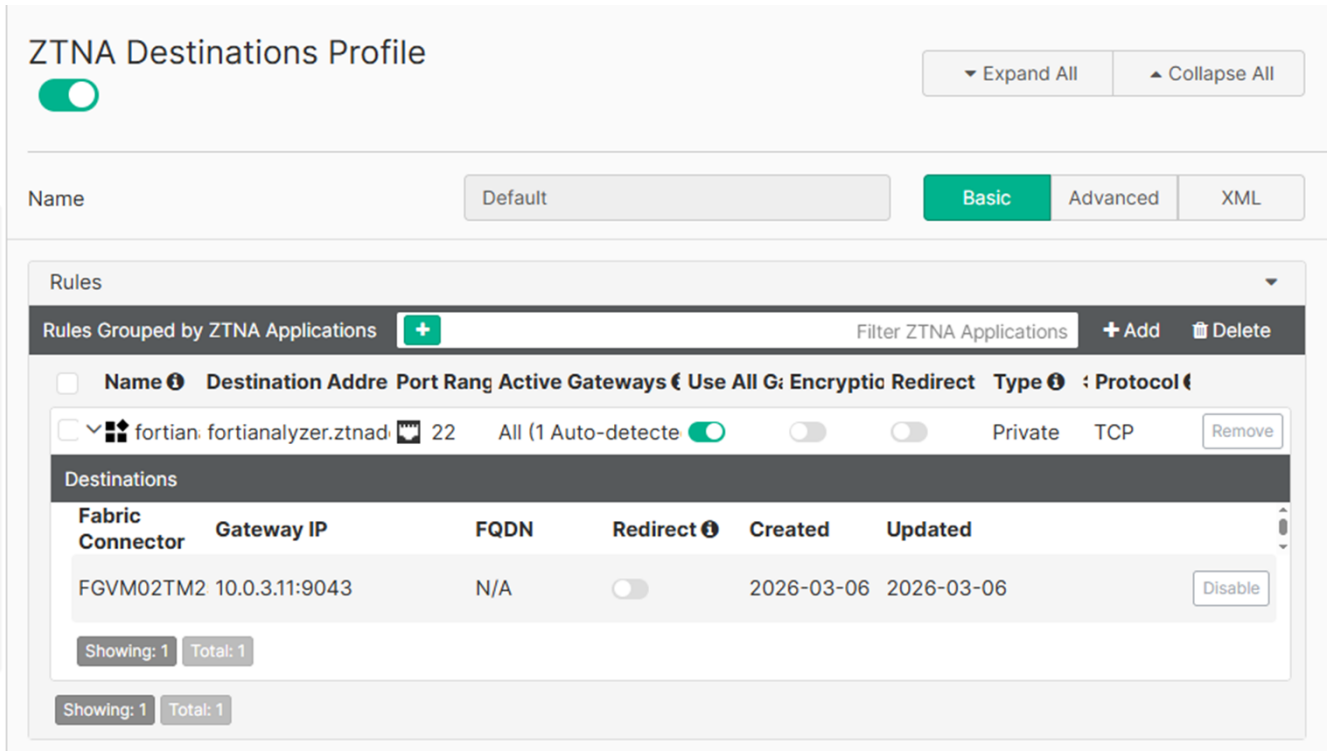
Failed log

This failure is as a result of setting the wrong host when configuring the TCP forwarding proxy. In this case, the host field was set to *fortianalyzer* instead of *10.0.3.11*.

```
1: date=2026-03-06 time=11:13:48 eventtime=1772824427849248121 tz="-0800" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=20949
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.0.3.11
dstport=9043 dstintf="root" dstintfrole="undefined" sessionid=61384 srcuuid="b458a65a-f759-51ea-
d7df-ef2e750026d1" service="tcp/9043" proxyapptype="ztna-proxy" proto=6 action="deny" policyid=1
policytype="policy" poluuid="6d795be4-1972-51f1-e6ab-ed4c5318bd15" policyname="ZTNA-Admin-Access"
appcat="unscanned" duration=0 vip="10.0.3.11:9043" vipincomingip=10.0.3.11
clientdevicemanageable="manageable" clientcert="yes" wanin=0 rcvdbyte=0 wanout=0 lanin=1970
sentbyte=1970 lanout=3014 msg="Traffic denied because HTTP url
(https://10.0.3.11/tcp?address=fortianalyzer.ztnademo.com&port=22&tls=0) failed to match an API-
gateway with vhost(name/hostname:_def_virtual_host_/_def_virtual_host_)" crscore=30
craction=131072 crlevel="high"
```

This log indicates that the FortiClient sends the request `https://10.0.3.11/tcp?address=fortianalyzer.ztnademo.com&port=22&tls=0`, which does not match the configured virtual host (traffic forwarding server): `https://fortianalyzer/tcp?address=fortianalyzer.ztnademo.com&port=22&tls=0`.

The FortiClient request is defined by the Gateway IP in the *ZTNA Destinations Profile > Rules > Destinations > Gateway IP* on EMS. In this case: 10.0.3.11.



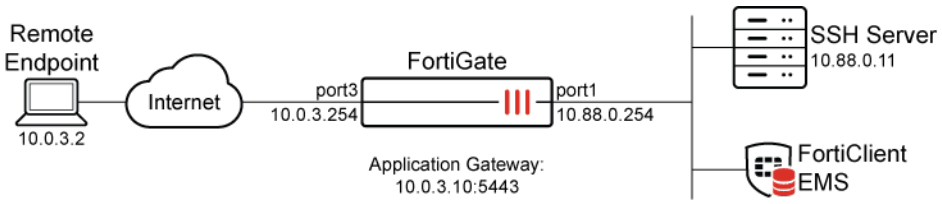
The Gateway IP is defined by the IP address and port combination as configured in the ZTNA port, *ZTNA-10.0.3.11* on the FortiGate. The port must be used in a firewall policy before it is synchronized to the EMS with the fabric connector.

ZTNA SSH traffic forwarding proxy example

ZTNA can be configured with SSH traffic forwarding proxy to provide a seamless SSH connection to the server.

Advantages of using an SSH access proxy instead of a TCP forwarding access proxy include:

- Establishing device trust context with user identity and device identity checks.
- Applying SSH deep inspection to the traffic through the SSH related profile.
- Performing optional SSH host-key validation of the server.
- Using one-time user authentication to authenticate the ZTNA SSH access proxy connection and the SSH server connection.



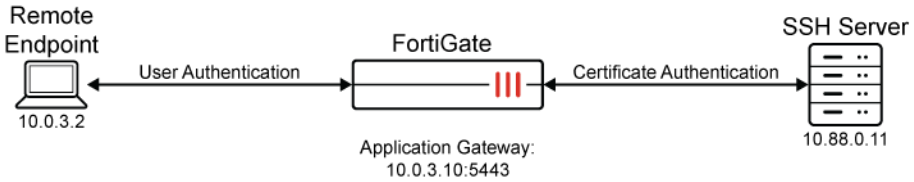
This example assumes that the FortiGate EMS fabric connector is already successfully connected.

Perform SSH host-key validation of the server

To act as a reverse proxy for the SSH server, the FortiGate must perform SSH host-key validation to verify the identity of the SSH server. The FortiGate does this by storing the public key of the SSH server in its SSH host-key configurations. When a connection is made to the SSH server, if the public key matches one that is used by the server, then the connection is established. If there is no match, then the connection fails.

One-time user authentication

SSH access proxy allows user authentication to occur between the client and the access proxy, while using the same user credentials to authenticate with the SSH server. The following illustrates how this works:



1. The remote endpoint registers to FortiClient EMS and receives the client certificate.
2. The remote endpoint tries to connect to the SSH access proxy. It must use the same username that is later used for access proxy authentication.
3. The FortiGate challenges the endpoint with device identity validation.
4. The remote endpoint provides the EMS issued certificate for device identification.
5. The FortiGate challenges the endpoint with user authentication. This example uses an LDAP group to control access.
6. The users enters their credentials on the remote endpoint.
7. The FortiGate authenticates the user and collects the username.
8. Using the FortiGate's CA or the customer's CA certificate, the FortiGate signs an SSH certificate and embeds the username in its principal.
9. The FortiGate attempts to connect to the SSH server using the certificate authentication.
10. The SSH server verifies the authenticity of the certificate, and matches the username principal against its authorized_keys file.
11. If the username matches a record in the file, then the SSH connection is established. If no match is found, then the SSH connection fails.

Example

In this example, an SSH connection is established using SSH access proxy with host-key validation and one-time authentication.

- The SSH server is a Linux based server that uses sshd to provide remote access.
- For SSH host-key validation, the public key of the SSH server has been imported into the FortiGate.
- For one-time authentication using certificate authentication:

- The SSH server must allow certificate authentication.
- The SSH server must have the proper entry in its authorized_keys file that contains the user principal and the FortiGate CA's public key.
- The entry is present in the user directory corresponding to the user that is trying to log in.

To pre-configure the Linux SSH server:

1. Retrieve the public key used for host-key validation:
   a. Locate the public key files in the SSH server:

```
$ ls -la /etc/ssh/*.pub
-rw-r--r-- 1 root root 186 Mar 29  2020 /etc/ssh/ssh_host_ecdsa_key.pub
-rw-r--r-- 1 root root 106 Mar 29  2020 /etc/ssh/ssh_host_ed25519_key.pub
-rw-r--r-- 1 root root 406 Mar 29  2020 /etc/ssh/ssh_host_rsa_key.pub2
```

   b. Choose the public key file based on the hash type (in this case, ECDSA), and show it's content:

```
$ cat /etc/ssh/ssh_host_ecdsa_key.pub
ecdsa-sha2-nistp256 AAAAE2*********IpEik=
```

   This key will be used when configuring the FortiGate.
2. Retrieve the FortiGate CA's public key from the FortiGate:

```
# show full firewall ssh local-ca Fortinet_SSH_CA
config firewall ssh local-ca
    edit "Fortinet_SSH_CA"
        set password <hidden password>
        set private-key "-----BEGIN OPENSSH PRIVATE KEY-----
<hidden private key>
-----END OPENSSH PRIVATE KEY-----"
        set public-key "ssh-rsa AAAAB3**********JLXlxj3"
        set source built-in
    next
end
```

3. On the Linux server, enable the SSH service to use the authorized_keys file:
   a. Locate and edit the */etc/ssh/sshd_config* file.
   b. Ensure that the *AuthorizedKeysFile* line is uncommented, for example:

```
AuthorizedKeysFile      .ssh/authorized_keys .ssh/authorized_keys2
```

4. Allow remote SSH log in with certificate authentication and principal name:
   a. Log in to the SSH server using the account that will be granted remote SSH access (in this example: *tsmith*):
   b. Locate the account's authorized_keys file in the *~/.ssh* directory:

```
$ ls -la ~/.ssh
 total 12
 drwxrwxr-x 2 tsmith tsmith 4096 Aug 10 19:14 .
 drwxr-xr-x 5 tsmith tsmith 4096 Aug 10 19:13 ..
 -rw-rw-r-- 1 tsmith tsmith 419 Aug 10 19:14 authorized_keys
```

c. If the directory and file do not exist, create the directory:

```
$ mkdir ~/.ssh
```

d. Create an entry containing the following keywords and add them to the authorized_keys file:

```
echo 'cert-authority,principals="tsmith" ssh-rsa AAAAB3**********JLXlxj3' >> authorized_
keys
```

Where:

- `cert-authority` - indicates that this entry is used in certificate authentication by validating the certificate using the public key provided in this entry.
- `principals="tsmith"` - indicates the user that must match with the username embedded in the SSH certificate.
- `ssh-rsa AAAAB3**********JLXlxj3` - indicates the FortiGate CA's public key that is used to validate the SSH certificate.

5. Restart the sshd service:

```
$ sudo systemctl stop sshd
$ sudo systemctl start sshd
```

The SSH server can now accept SSH connection from *tsmith@<server IP>*, where the SSH certificate used by the FortiGate to log in contains tsmith embedded as a principal.

When a user connects from a SSH client using *<username>@<server IP>*, sshd will locate the authorized_keys file in the directory */home/<username>/.ssh/authorized_keys*. If the authorized_keys is not in that directory, authentication will fail on the SSH server side.

If you suspect that authentication is failing on the SSH server, use the following commands to manually start sshd in debug mode to troubleshoot:

```
$ sudo systemctl stop sshd
```

```
$ /usr/sbin/sshd -ddd -p 22
```

To configure the FortiGate :

1. Configure a new VIP to allow access to the SSH access proxy over 10.0.3.10:5443:

```
config firewall vip
    edit "ZTNA_SSH"
        set type access-proxy
        set server-type https
        set extip 10.0.3.10
        set extintf "port3"
        set extport 5443
        set ssl-certificate "Fortinet_SSL"
    next
end
```

2. Configure the address object for the SSH server:

```
config firewall address
    edit "SSH_server"
        set subnet 10.88.0.11 255.255.255.255
    next
end
```

3. Configure the host-key that will be used to authenticate the SSH server. The public-key was retrieved during pre-configure the Linux SSH server (step 1b).

```
config firewall ssh host-key
    edit "ed25519"
        set type ECDSA
        set usage access-proxy
        set public-key "AAAAE2**********IpEik="
    next
end
```

4. Configure the access proxy SSH client certificate:

   A CA certificate is assigned to sign the SSH certificate that will be used in the SSH authentication. The SSH certificate will have the username embedded in the certificate principal.

```
config firewall access-proxy-ssh-client-cert
    edit "ssh-access-proxy"
        set source-address enable
        set auth-ca "Fortinet_SSH_CA"
    next
end
```

5. Configure the traffic forward proxy virtual host settings:

```
config firewall access-proxy-virtual-host
    edit "10.0.3.10_host"
        set ssl-certificate “ztna-wildcard”
        set host “10.0.3.10”
    next
end
```

   The virtual host is configured automatically when specifying the Host field during the Traffic Forwarding Server creation in the GUI.

6. Configure the traffic forward proxy:

```
config ztna traffic-forward-proxy
    edit "ZTNA_SSH"
        set vip "ZTNA_SSH"
        set host "10.0.3.10_host"
    next
end
```

7. Configure the destination:

```
config ztna destination
    edit "SSH_Server"
        set address "SSH_server"
        set mappedport 22
        set protocol TCP
        set conn-type ssh
        set ssh-client-cert "ssh-access-proxy"
        set ssh-host-key-validation enable
        set ssh-host-key "ed25519"
    next
end
```

8. Configure the LDAP setting, user setting, and user group to apply user authentication to the access proxy connection using LDAP:

```
config user ldap
    edit "LDAP-fortiad"
        set server "10.88.0.1"
        set cnid "sAMAccountName"
        set dn "dc=fortiad,dc=info"
        set type regular
        set username "fortiad\\Administrator"
        set password *
        set secure ldaps
        set port 636
    next
end
```

```
config user group
    edit "LDAP-Remote-Allowed-Group"
        set member "LDAP-fortiad"
        config match
            edit 1
                set server-name "LDAP-fortiad"
                set group-name "CN=Remote-Allowed,CN=Users,DC=fortiad,DC=info"
            next
        end
    next
end
```

9. Create the authentication scheme and rule to perform the authentication:

```
config authentication scheme
    edit "basic_auth"
        set method basic
        set user-database "LDAP-fortiad"
    next
end
```

```
config authentication rule
    edit "ztna_basic"
        set srcintf "port3"
```

```
        set srcaddr "all"
        set ip-based disable
        set active-auth-method "basic_auth"
        set web-auth-cookie enable
    next
end
```

10. Configure the full ZTNA policy to allow traffic to the SSH server, and apply user authentication, posture check, and a security profile where necessary:

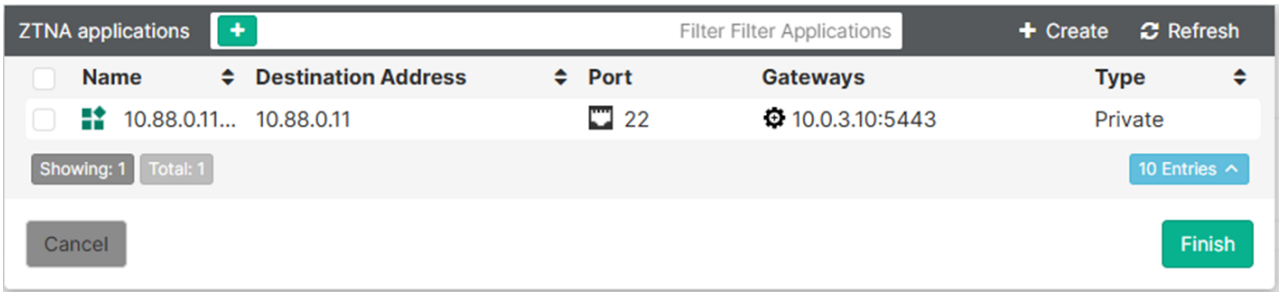
```
config firewall policy
    edit 18
        set name "SSH-proxy"
        set srcintf "port3"
        set dstintf "any"
        set action accept
        set srcaddr "all"
        set dstaddr "ZTNA_SSH"
        set ztna-destination "SSH_Server"
        set schedule "always"
        set ssl-ssh-profile "custom-deep-inspection"
        set logtraffic all
        set nat enable
        set groups "LDAP-Remote-Allowed-Group"
    next
end
```

Results

Before connecting, users must have a ZTNA Destination in FortiClient.

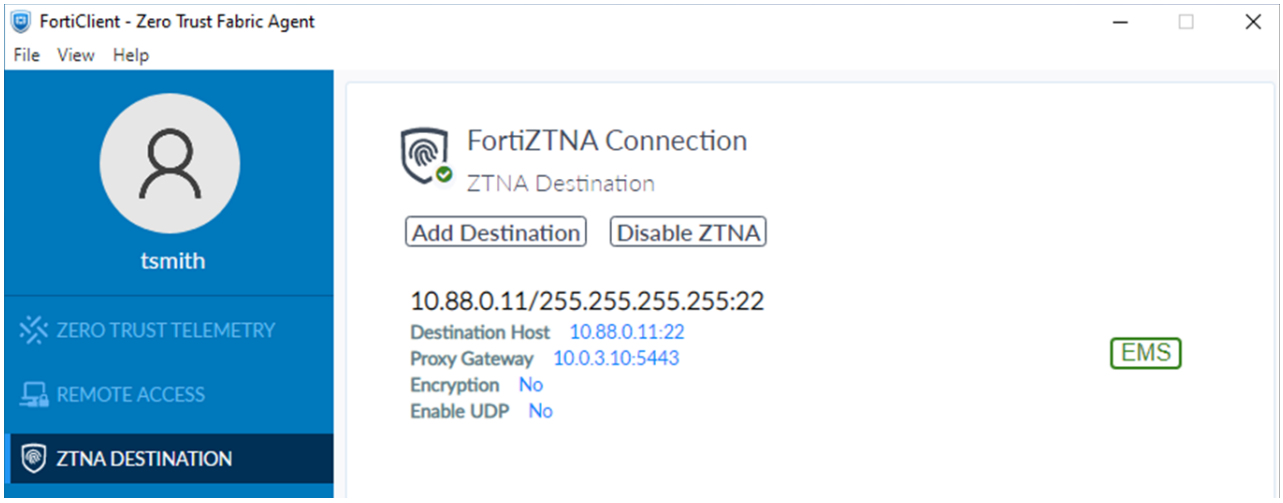
To create the ZTNA rules in EMS:

1. Go to *Endpoint Profiles* > *ZTNA Destinations*.
2. Select the required profile, or create a new one.
3. Set *Name* to the desired name.
4. In the *Rules* section click *Add*.
   The ZTNA destination appears as learned from the FortiGate through the fabric connector.



5. Select the entry and click *Finish*.
   The destination now appears as a rule entry.
6. Click *Save* to save the ZTNA Destinations profile.

7. Ensure that this profile is applied to the endpoint policy that is assigned to the user's endpoint.
   After some time, the FortiClient will receive the profile update and the configuration will appear under the ZTNA DESTINATION tab on FortiClient.

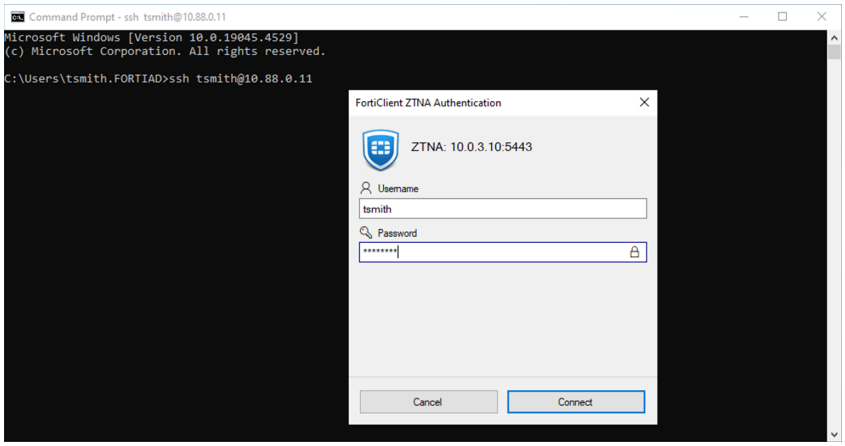


To test the access proxy:

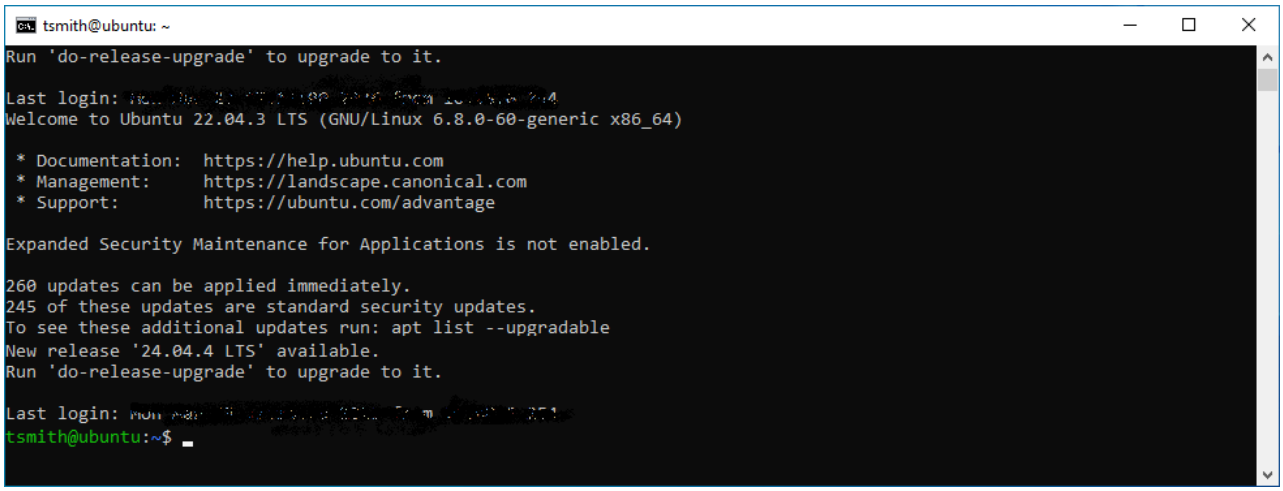
1. Open an SSH client, such as Windows command prompt, and make an SSH connection to tsmith@10.88.0.11. SSH uses port 22 by default.

```
ssh tsmith@10.88.0.11
```

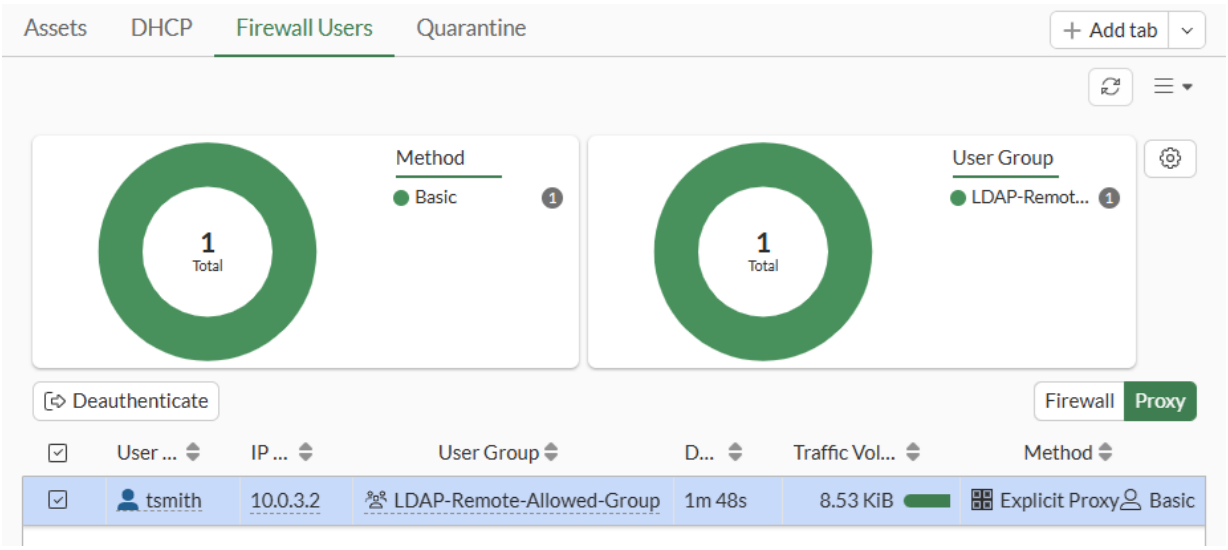
2. After device authentication is performed and passes in the background, FortiClient prompts the user to sign in.
   Enter the username, *tsmith*, and password, then click *Sign in*.



   After successful user authentication, the SSH connection is established without an additional log in.



3. On the FortiGate, check the logged in user:
   a. Go to *Dashboard* > *Assets & Identities* > *Firewall Users* and select the *Proxy* table.



   b. Check the WAD proxy user list:

```
# diagnose wad user list
 ID: 1, VDOM: root, IPv4: 10.0.3.2
  user name    : tsmith
  worker       : 0
  duration     : 200
  auth_type    : Session
  auth_method  : Basic
  pol_id       : 18
  g_id         : 3
  user_based   : 0
  expire       : no
  LAN:
    bytes_in=4727 bytes_out=7494
  WAN:
    bytes_in=4586 bytes_out=4149
```

4. The successful connection is logged in the forward traffic logs after the SSH connection has disconnected. There are two logs as there is a pause for the user to authenticate.

```
# execute log filter category 0
# execute log filter field subtype ztna
# execute log display

1: date=2026-03-23 time=10:27:44 eventtime=1774286864266735556 tz="-0700" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=4568
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.88.0.11
dstport=22 dstintf="port2" dstintfrole="dmz" sessionid=477 srcuuid="b458a65a-f759-51ea-d7df-
ef2e750026d1" service="SSH" proxyapptype="ztna-proxy" proto=6 action="accept" policyid=18
policytype="policy" poluuid="97082ece-2483-51f1-df8b-f33fd5ffae6d" policyname="SSH-proxy"
appcat="unscanned" duration=1 user="tsmith" group="LDAP-Remote-Allowed-Group"
authserver="LDAP-fortiad" vip="ZTNA_SSH" vipincomingip=10.0.3.10 accessproxy="ZTNA_SSH"
ztnadestination="SSH_Server" clientdevicemanageable="manageable" clientcert="yes" wanin=0
rcvdbyte=0 wanout=0 lanin=2094 sentbyte=2094 lanout=3328

2: date=2026-03-23 time=10:27:43 eventtime=1774286863705959985 tz="-0700" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=4567
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.0.3.10
dstport=5443 dstintf="root" dstintfrole="undefined" sessionid=472 srcuuid="b458a65a-f759-51ea-
d7df-ef2e750026d1" service="tcp/5443" proxyapptype="ztna-proxy" proto=6 action="pending"
policyid=18 policytype="policy" poluuid="97082ece-2483-51f1-df8b-f33fd5ffae6d"
policyname="SSH-proxy" appcat="unscanned" duration=17 vip="ZTNA_SSH" vipincomingip=10.0.3.10
clientdevicemanageable="manageable" clientcert="yes" wanin=0 rcvdbyte=0 wanout=0 lanin=2083
sentbyte=2083 lanout=38001
```

5. Review the SSH login attempt on the Linux server by reviewing the auth.log file:

```
root:~$ cat /var/log/auth.log
Mar 23 18:27:44 ubuntu sshd[3345190]: Accepted publickey for tsmith from 10.88.0.254 port
15910 ssh2: RSA-CERT SHA256:6aIamnV…7SY ID tsmith (serial 0) CA RSA SHA256:DD/As8Y…k3animbo
Mar 23 18:27:44 ubuntu sshd[3345190]: pam_unix(sshd:session): session opened for user tsmith
(uid=1001) by (uid=0)
Mar 23 18:27:44 ubuntu systemd-logind[630]: New session 1475 of user tsmith.
Mar 23 18:31:22 ubuntu sshd[3345268]: Received disconnect from 10.88.0.254 port 15910:11:
disconnected by user
Mar 23 18:31:22 ubuntu sshd[3345268]: Disconnected from user tsmith 10.88.0.254 port 15910
Mar 23 18:31:22 ubuntu sshd[3345190]: pam_unix(sshd:session): session closed for user tsmith
Mar 23 18:31:22 ubuntu systemd-logind[630]: Session 1475 logged out. Waiting for processes to
exit.
Mar 23 18:31:22 ubuntu systemd-logind[630]: Removed session 1475.
```

ZTNA application gateway with SAML authentication example

SAML can be used with ZTNA as an authentication method. In this setup, user credentials are stored in Active Directory, which is accessed by the Identity Provider (IdP). FortiAuthenticator serves as the IdP, retrieving user identity information from Active Directory, while FortiGate acts as the Service Provider (SP) delivering ZTNA

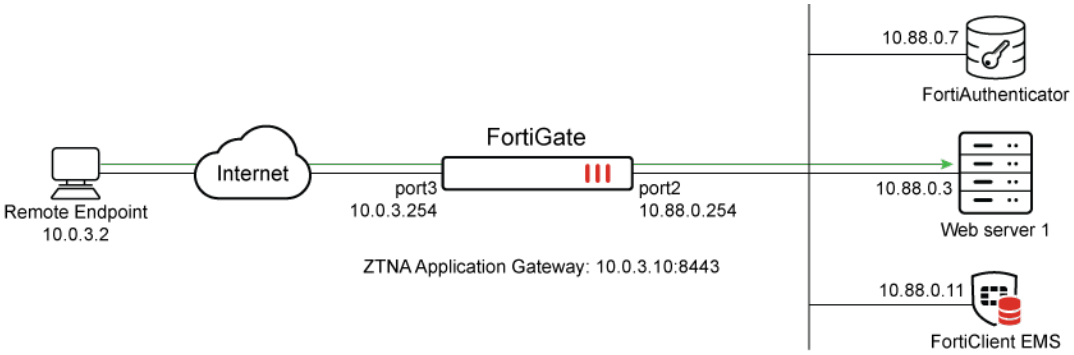
services alongside its security functions. When a user attempts to access a protected resource, the FortiGate redirects them to FortiAuthenticator for authentication. After the user successfully authenticates, FortiAuthenticator returns a SAML assertion to the FortiGate, which then acts as the trust broker to enforce policies and authorize access based on the validated identity and attributes.

For a basic configuration:

1. Configure ZTNA for web server access.
2. Configure a SAML SSO object on the FortiGate.
3. Update the IdP to reflect the SP settings.
4. Apply the SAML SSO object to an authentication scheme.
5. Create a user group that uses the SAML SSO object as its remote authentication server.
6. Create a ZTNA policy that uses the web server ZTNA configuration and SAML user group.

Example

In this example, an HTTPS access proxy is configured, and SAML authentication is applied to authenticate the client. The FortiGate acts as the SAML SP and a SAML authenticator serves as the IdP. In addition to verifying the user and device identity with the client certificate, the user is also authorized based on user credentials to establish a trust context before granting access to the protected resource. A VIP and Firewall policy is created to enable the Remote Endpoint to access the FortiAuthenticator for IdP authentication.



This example assumes that the FortiGate EMS fabric connector is already successfully connected.

Configure ZTNA for the web server

These steps will create a ZTNA application gateway of 10.0.3.10:8443 to proxy connections to 10.88.0.3:9043.

To configure ZTNA for the web server in the GUI:

1. Go to *Policy & Objects* > *ZTNA* and select the *Servers and Portals* tab. Web Server is selected by default.
2. Click *Create New*.
3. Set *Name* to *ZTNA-Webserver*.
4. Set *Host* to the IP address or FQDN that will be used to access the web server: webserver.ztnademo.com.

> Endpoints accessing the web server must resolve this FQDN to the ZTNA Application Gateway address (10.0.3.10 in this example).

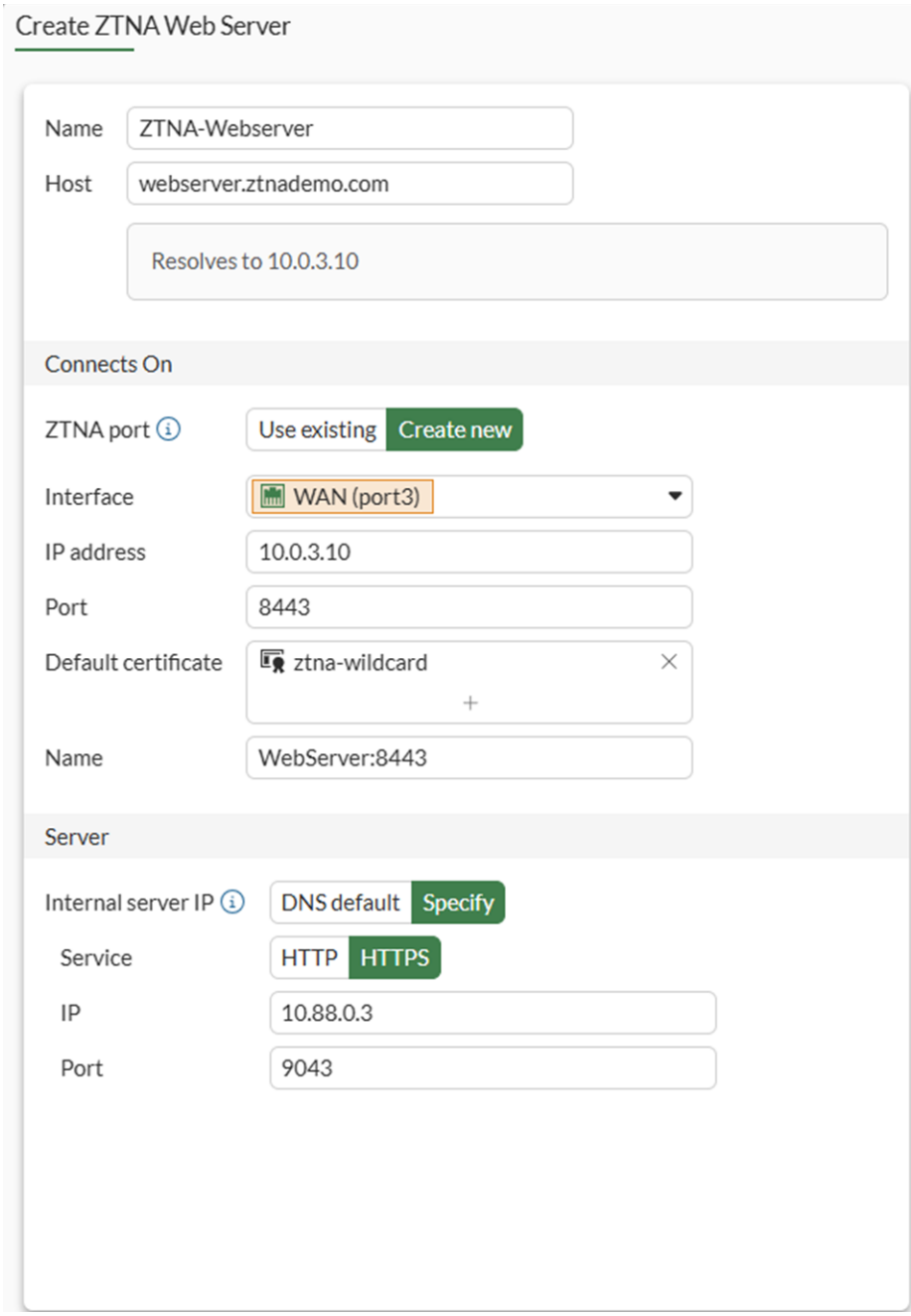
5. Under *Connects On*, select *Create new* for ZTNA port to provide the following details:
   a. Set *Interface* to *port3*.
   b. Set *External IP* to *10.0.3.10*.
   c. Set *External port* to *8443*.

   > Verify that the IP address and port do not conflict with management access to the interface. Otherwise, change the IP address to another address on that subnet.

6. Select a *certificate*. Clients will be presented with this certificate when they connect to the ZTNA port.

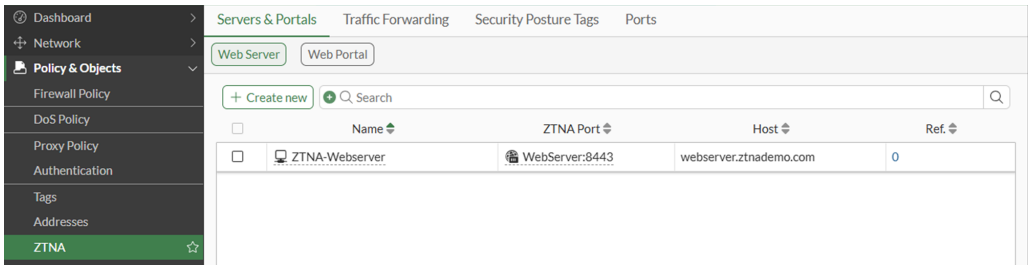
   > This example uses a wildcard certificate with a Common Name (CN): *.ztnademo.com. This certificate is trusted by the endpoints which will access the ZTNA port.

7. Provide a name for the ZTNA port, such as WebServer:8443.
8. In the Server section, configure the internal server settings as follows:
   a. Switch the Internal server IP to *Specify*.
   b. Set *Service* to *HTTPS*.
   c. Set *IP* to the internal IP for the web server: 10.88.0.3.
   d. Set *Port* to *9043*.



The WebServer service is running on port 9043. This configuration will translate clients accessing TCP/9443 to TCP/9043.

e. Click *OK*. The Web Server object is listed.



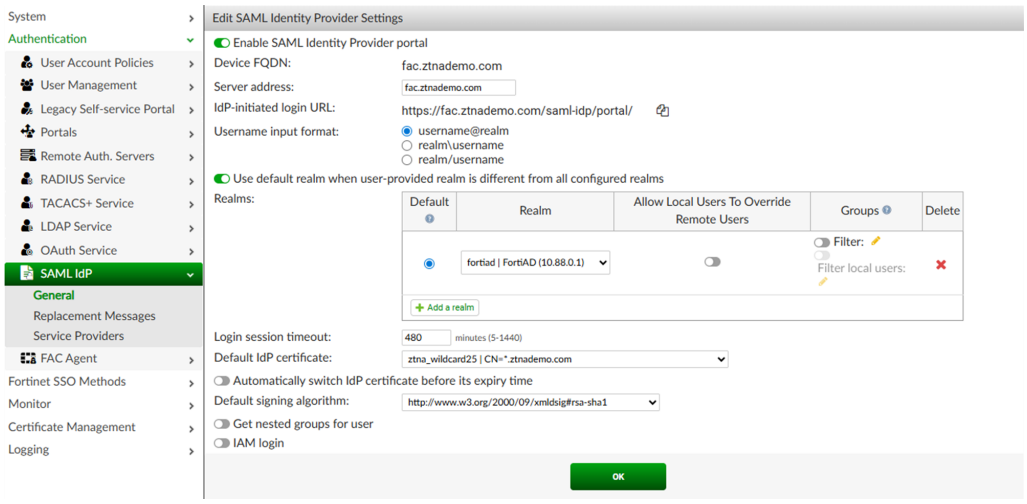
To configure ZTNA for the web server in the CLI:

```
config firewall vip
        edit "WebServer:8443"
        set uuid 55f5f710-1d56-51f1-f6cb-1be820301ca5
        set type access-proxy
        set server-type https
        set extip 10.0.3.10
        set extintf "port3"
        set extport 8443
        set ssl-certificate "ztna-wildcard"
    next
end
```

```
config ztna web-proxy
    edit "ZTNA-Webserver"
        set vip "WebServer:9443"
        set host " WebServer9443"
        config api-gateway
            edit 1
                config realservers
                    edit 1
                        set ip 10.88.0.3
                        set port 9043
                    next
                end
            next
        end
    next
end
```

Configuring SAML server

This must match the settings on the IdP. For this example, the FortiAuthenticator (IdP) has been configured as follows:



To configure FortiGate as the Service Provider:

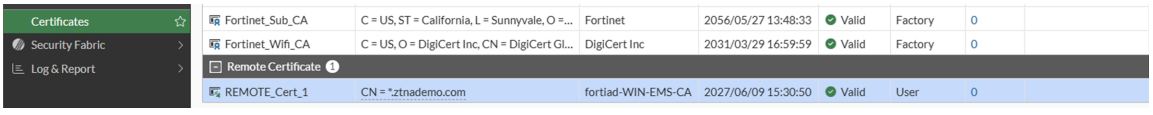
1. Go to *User & Authentication* > *Single Sign-On* and select *Create New*.
2. Set *Name* to saml_ztna.
3. Set *Address* to webserver.ztnademo.com:8443.

   The Entity ID, Assertion consumer service URL and Single logout service URL will be updated. You will need to enter these URLs into FAC in a later step.
4. Enable *Certificate*, then select the certificate used for the client.

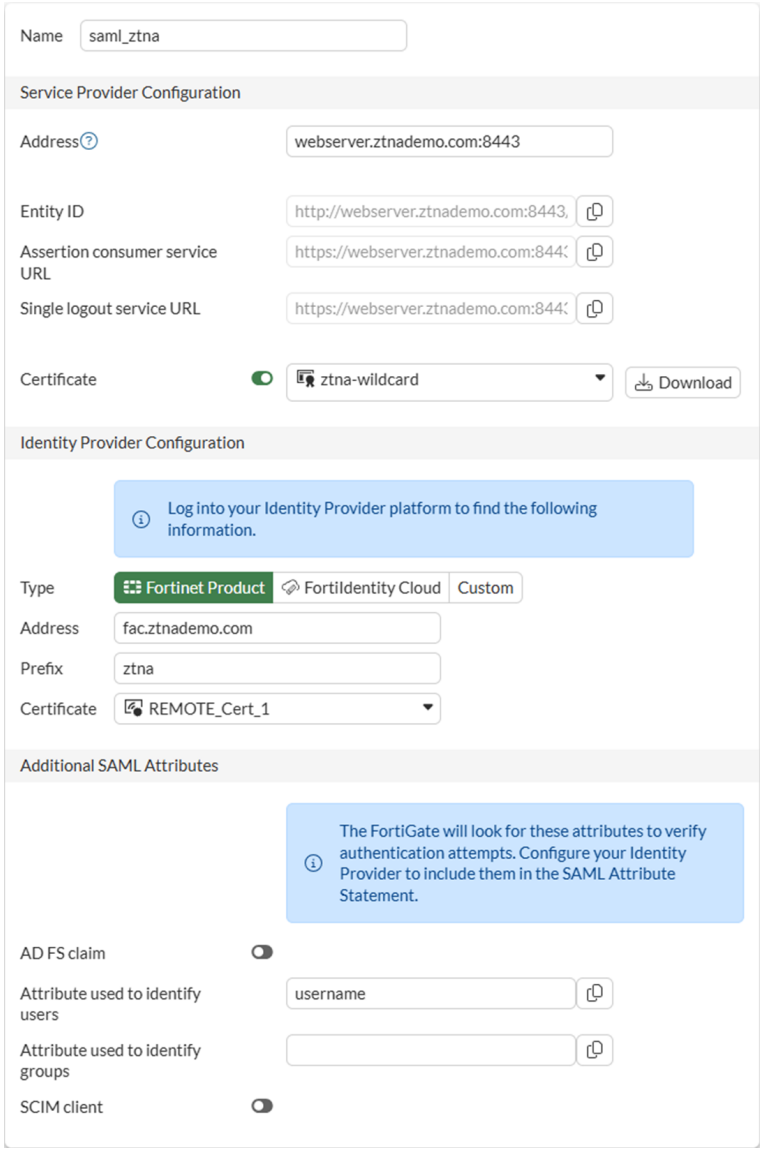
   In this example, the ztna-wildcard certificate is a local certificate that is used to sign SAML messages that are exchanged between the client and the FortiGate SP.
5. Use the settings from the FortiAuthenticator to fill the custom Identity Provider Details. In this example, we select Fortinet Product and fill in the following fields:

Address	fac.ztnademo.com
Prefix	ztna
IdP certificate	REMOTE_Cert_1

   The certificate REMOTE_Cert_1 is the certificate used by the IdP (fac.ztnademo.com). In this example, a wildcard certificate is used.



6. Set Attribute used to identify users to username. Attributes to identify users and groups are case sensitive.



7. Click *OK* to save the settings

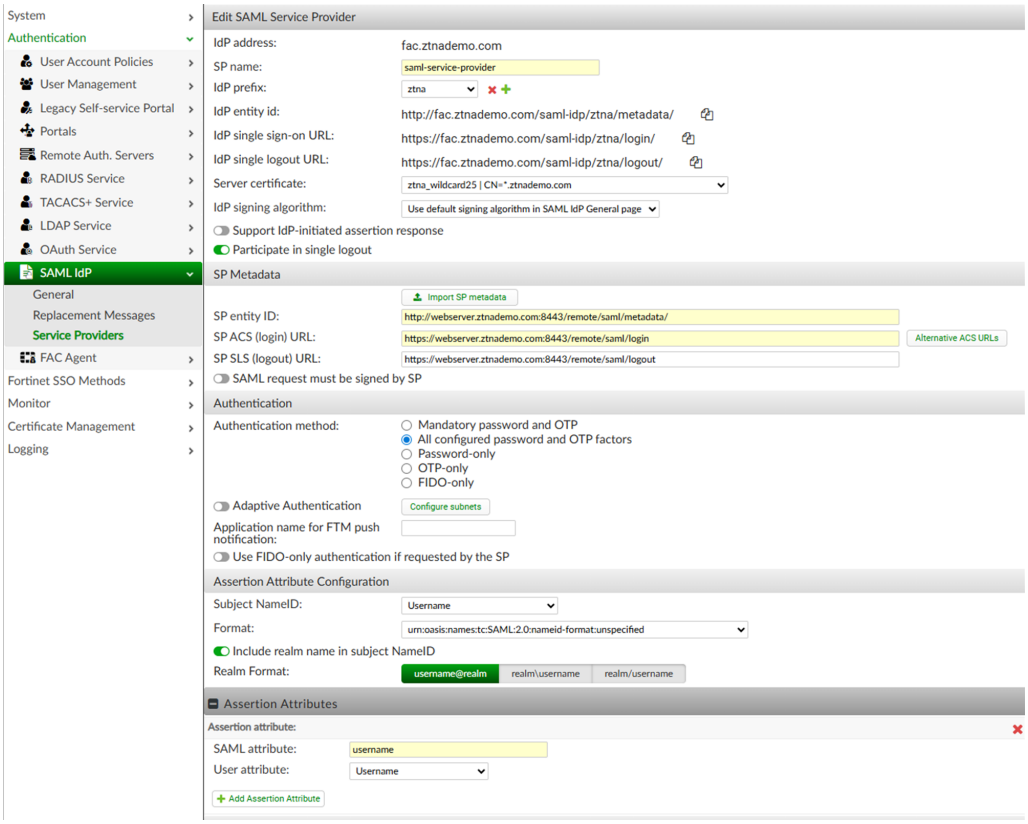
To configure a SAML server in the CLI:

```
config user saml
    edit "saml_ztna"
        set cert "ztna-wildcard"
        set idp-cert "REMOTE_Cert_1"
        set user-name "username"
        set digest-method sha256
        set entity-id "http://webserver.ztnademo.com:8443/remote/saml/metadata/"
        set single-sign-on-url "https://webserver.ztnademo.com:8443/remote/saml/login"
        set single-logout-url "https://webserver.ztnademo.com:8443/remote/saml/logout"
        set idp-entity-id "http://fac.ztnademo.com/saml-idp/ztna/metadata/"
        set idp-single-sign-on-url "https://fac.ztnademo.com/saml-idp/ztna/login/"
        set idp-single-logout-url "https://fac.ztnademo.com/saml-idp/ztna/logout/"
```

```
    next
end
```

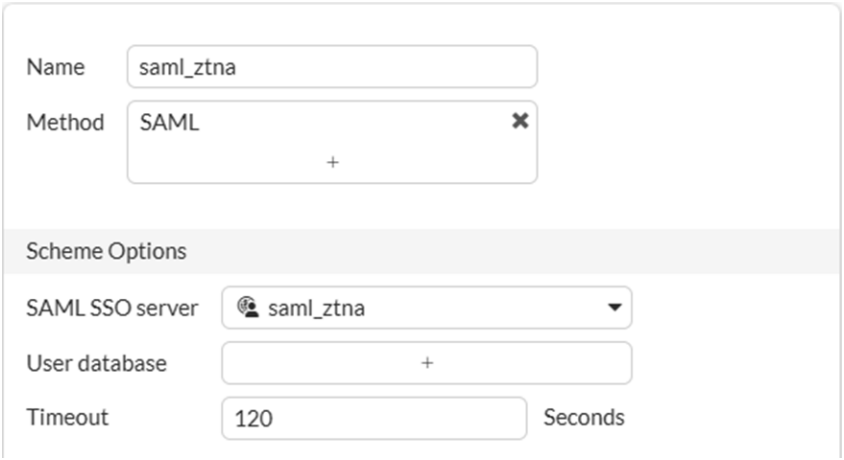
Apply the SP settings in the IdP

The settings configured on the SP must be added to the IdP.

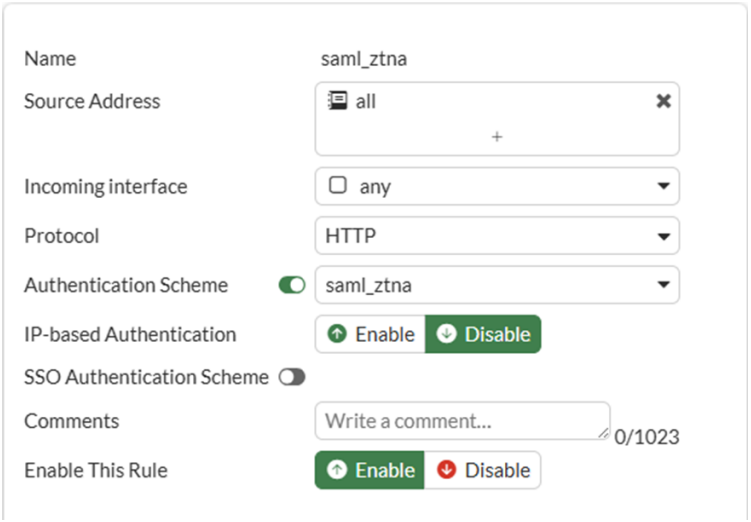


To apply the SAML server to proxy authentication in the GUI:

1. Go to *Policy & Objects* > *Authentication*.
   a. Create a new authentication scheme:
   b. Go to the *Authentication Scheme* tab and click *Create New*.
   c. Set the *Name* to saml_ztna.
   d. Set the *Method* to *SAML*.
   e. Set *SAML SSO server* to saml_ztna.
   f. Click *OK*.



2. Create a new authentication rule:
   a. Select *Authentication Rules* from the top menu bar.
   b. Click *Create New*.
   c. Set the *Name* to saml_ztna.
   d. Set the *Source address* to all.
   e. Set *Incoming interface* to port3.
   f. Set *Protocol* to *HTTP*.
   g. Enable *Authentication Scheme*, and select saml_ztna.
   h. Set *IP-based Authentication* to *Disable*.
   i. Click *OK* to save the rule.



To apply the SAML server to proxy authentication from the CLI:

```
config authentication scheme
     edit "saml_ztna"
         set method saml
         set saml-server "saml_ztna"
```

```
    next
end
```

```
config authentication rule
    edit "saml_ztna"
        set srcintf "port3"
        set srcaddr "all"
        set ip-based disable
        set active-auth-method "saml_ztna"
        set web-auth-cookie enable
    next
end
```

To create a user group for the SAML user object:

1. Go to *User & Authentication > User Groups*, and click *Create New*.
2. Set *Name* to saml_group.
3. Under *Remote Groups*, click *Add*.
4. For *Remote Server*, select the SAML user object: saml_ztna and click *OK*.
5. Click *OK* again to save the user group.

To create a user group for the SAML user object in the CLI:

```
config user group
    edit "saml_group"
        set member "saml_ztna"
    next
end
```

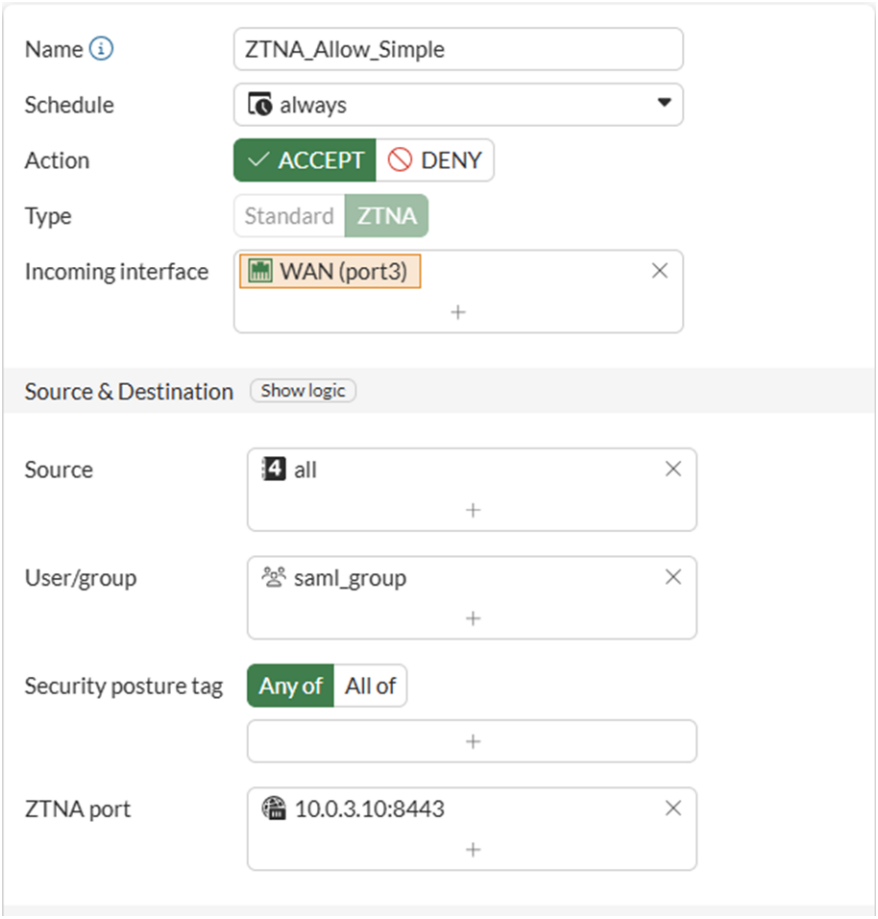
To enable user group authentication in a ZTNA firewall policy:

A simple ZTNA policy is created to apply the SAML authentication.

1. Go to *Policy & Objects > Firewall Policy* and select *Create New*.
2. Configure the following:

Name	ZTNA_Allow_Simple
Action	Accept
Type	ZTNA
Incoming Interface	port3
Source	All
User/group	saml_group
ZTNA port	WebServer:8443
Log allowed traffic	All sessions

3. Click *OK* to save the policy.



To enable user group authentication in a ZTNA firewall policy in the CLI:

```
config firewall policy
    edit 1
        set name "ZTNA_Allow_Simple"
        set srcintf "port3"
        set dstintf "any"
        set action accept
        set srcaddr "all"
        set dstaddr "WebServer:8443"
        set ztna-destination "Webserver:9043"
        set schedule "always"
        set logtraffic all
        set nat enable
        set groups "saml_group"
    next
end
```

To configure a VIP and firewall policy to forward IdP authentication traffic to the FortiAuthenticator

Remote clients connect to the FortiAuthenticator IdP behind the FortiGate using a VIP. In this example, users connect to the FQDN fac.ztnademo.com which resolves to the VIP's external IP address.

1. Configure the VIP to forward traffic to the FortiAuthenticator:
   a. Go to *Policy & Objects* > *Virtual IPs*. The *Virtual IP* tab is selected.
   b. Click *Create New* to define the following:

Name	FAC-VIP
Interface	any
External IP address/range	10.0.3.7
Map to > IPv4 address/range	10.88.0.7

   Additional settings may be configured to limit traffic to TCP/443.
   a. Click *OK* to save the VIP.
2. Configure a firewall policy to allow the VIP traffic:
   a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
   b. Configure the following:

Name	WAN_to_FAC
Action	Accept
Type	Standard
Incoming Interface	port3
Outgoing Interface	port2
Source	All
Destination	FAC-VIP
Service	All
NAT	Disabled

   c. Click *OK* to save the policy

To configure a VIP and firewall policy to forward IdP authentication traffic to FortiAuthenticator in the CLI:

```
config firewall vip
    edit "FAC-VIP"
        set uuid a43372f8-06af-51ec-8f86-30b451b2ebe0
        set extip 10.0.3.7
        set mappedip "10.88.0.7"
        set extintf "any"
    next
end
```

```
config firewall policy
    edit 2
        set name "WAN_to_FAC"
        set srcintf "port3"
```

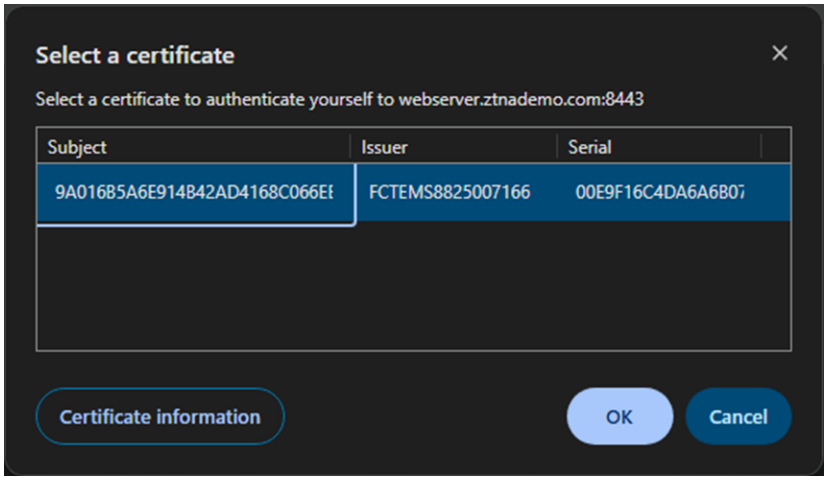
```
        set dstintf "port2"
        set action accept
        set srcaddr "all"
        set dstaddr "FAC-VIP"
        set schedule "always"
        set service "ALL"
        set logtraffic all
    next
end
```

Testing the connection

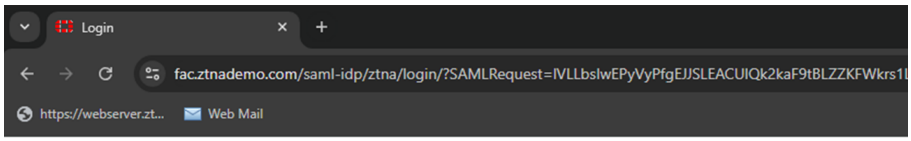
To test the connection:

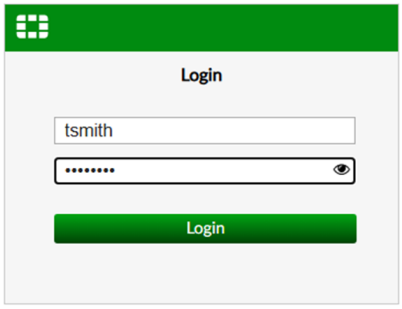
> It is not necessary to configure a *ZTNA Destination* on the FortiClient for the HTTPS access proxy use case. In fact, configuring a *ZTNA Destination* rule for the website may interfere with its operation.

1. On the remote Windows computer, open FortiClient and register to the EMS server.
2. Device authentication prompts the user for their device certificate. Select the certificate issued by EMS and click OK.

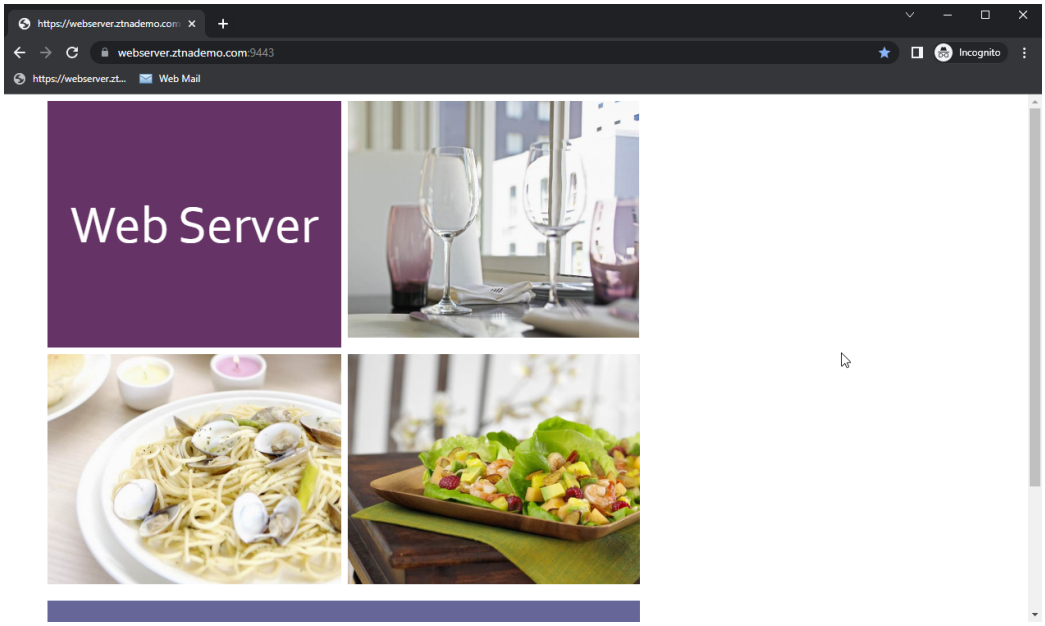


3. The user traffic matches the ZTNA policy and the user is redirected to the IdP for authentication.





4. Enter your user credentials. The SAML server authenticates and sends a SAML assertion response message to the FortiGate.
5. The user is proxied to the webpage on the real web server.



Logs and debugs

Use the following command to check the user information after the user has been authenticated:

```
# diagnose wad user list
ID: 3, VDOM: root, IPv4: 10.0.3.2
  user name    : tsmith
```

```
  worker      : 0
  duration    : 69
  auth_type   : Session
  auth_method : SAML
  pol_id      : 17
  g_id        : 8
  user_based  : 0
  expire      : no
  LAN:
    bytes_in=28146 bytes_out=309763
  WAN:
    bytes_in=303009 bytes_out=4474
```

ZTNA traffic log

```
# execute log filter category 0
# execute log filter field subtype ztna
# execute log display
1: date=2026-03-11 time=08:49:50 eventtime=1773244190067470581 tz="-0700" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=9417 srcintf="port3"
srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.88.0.3 dstport=9043
dstintf="port2" dstintfrole="dmz" sessionid=97173 srcuuid="b458a65a-f759-51ea-d7df-ef2e750026d1"
dstuuid="55f5f710-1d56-51f1-f6cb-1be820301ca5" service="tcp/9043" proxyapptype="ztna-proxy"
proto=6 action="accept" policyid=1 policytype="policy" poluuid="dfc01cba-1c88-51f1-cf94-
b9c10b070486" policyname="ZTNA_Allow_Simple" appcat="unscanned" duration=250 user="tsmith"
group="saml_group" authserver="saml_ztna" gatewayid=1 realserverid=1 vip="WebServer:8443"
vipincomingip=10.0.3.10 accessproxy="ZTNA-Webserver" clientdevicemanageable="manageable"
clientcert="yes" wanin=303009 rcvdbyte=303009 wanout=4474 lanin=28146 sentbyte=28146 lanout=309787
```

```
2: date=2026-03-11 time=08:44:09 eventtime=1773243849152689824 tz="-0700" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=9399 srcintf="port3"
srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.0.3.10 dstport=8443
dstintf="root" dstintfrole="undefined" sessionid=96814 srcuuid="b458a65a-f759-51ea-d7df-
ef2e750026d1" service="tcp/8443" proxyapptype="ztna-proxy" proto=6 action="pending" policyid=1
policytype="policy" poluuid="dfc01cba-1c88-51f1-cf94-b9c10b070486" policyname="ZTNA_Allow_Simple"
appcat="unscanned" duration=0 vip="WebServer:8443" vipincomingip=10.0.3.10
clientdevicemanageable="unknown" clientcert="no" wanin=0 rcvdbyte=0 wanout=0 lanin=1764
sentbyte=1764 lanout=4141
```

Forward traffic log

```
# execute log filter category 0
# execute log filter field policyid 1
# execute log filter field subtype forward
# execute log display
1: date=2026-03-11 time=08:45:40 eventtime=1773243940074849332 tz="-0700" logid="0000000010"
type="traffic" subtype="forward" level="notice" vd="root" srcip=10.0.3.2 srcport=9417
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.0.3.10
dstport=8443 dstintf="root" dstintfrole="undefined" sessionid=97173 srcuuid="b458a65a-f759-51ea-
d7df-ef2e750026d1" service="tcp/8443" proxyapptype="web-proxy" proto=6 action="accept" policyid=1
policytype="proxy-policy" poluuid="dfc01cba-1c88-51f1-cf94-b9c10b070486" policyname="ZTNA_Allow_
Simple" appcat="unscanned" duration=0 vip="WebServer:8443" vipincomingip=10.0.3.10
```

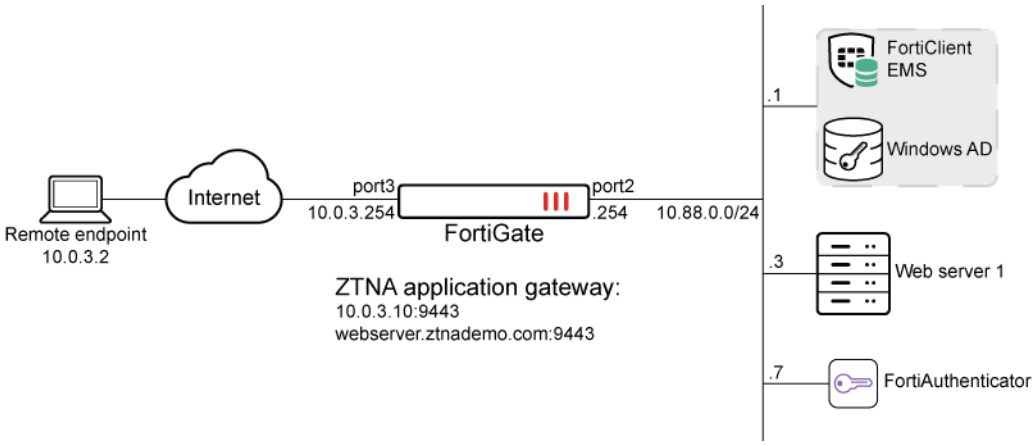
```
clientdevicemanageable="manageable" clientcert="yes" wanin=0 rcvdbyte=0 wanout=0 lanin=3810
sentbyte=3810 lanout=4582 msg="Traffic is redirected to SAML authentication."
```

ZTNA application gateway with SAML and MFA using FortiAuthenticator example

ZTNA application gateway supports device verification using device certificates that are issued by EMS. To authenticate users, administrators can use either basic or SAML authentication. An advantage of SAML authentication is that multi-factor authentication (MFA) can be provided by the SAML Identity Provider (IdP).

In these examples, a FortiAuthenticator is used as the IdP, and MFA is applied to user authentication for remote users accessing the web, RDP, and SSH resources over the ZTNA application gateway. It is assumed that the FortiGate EMS fabric connector has already been successfully connected.

- Configuring FortiAuthenticator on page 1751
- Configuring the FortiGate SAML settings on page 1755
- Example 1 - Applying SAML and MFA to ZTNA HTTPS access proxy on page 1760
- Example 2 - Applying SAML and MFA to a ZTNA TCP forwarding access proxy for RDP connections on page 1767



DNS resolutions:

- webserver.ztnademo.com:9443 -> 10.0.3.10:9443
- fac.ztnademo.com -> 10.0.3.7

The FortiAuthenticator (FAC) integrates with Active Directory (AD) on the Windows Domain Controller, which is also acting as the EMS server. Users are synchronized from the AD to the FAC, and remote users are configured with token-based authentication. SAML authentication is configured on the FortiGate, pointing to the FAC as the SAML IdP. The SAML server is applied to the ZTNA application gateway authentication scheme and rule, to provide the foundation for applying user authentication on individual ZTNA policies.

Configuring FortiAuthenticator

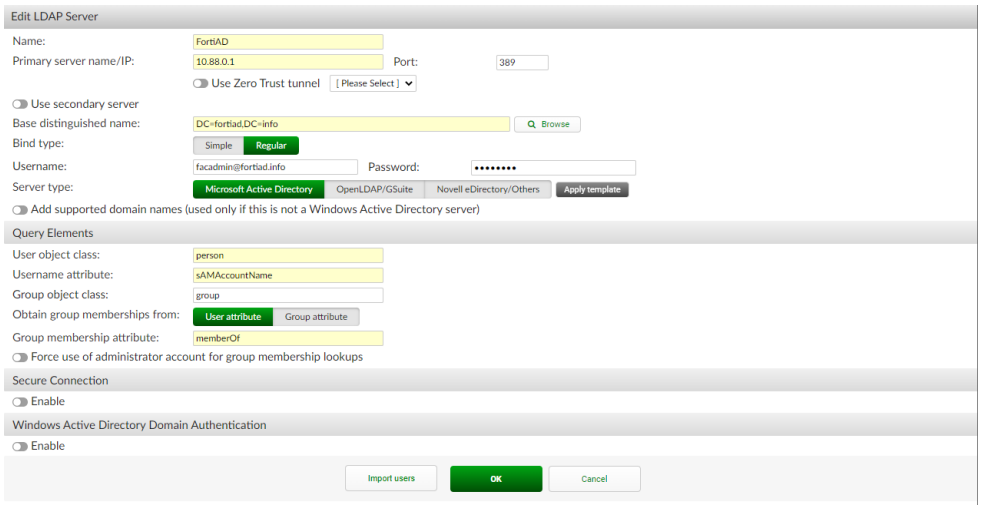
First configure FortiAuthenticator to synchronize users from AD using LDAP, apply MFA to individual remote users, and be the IdP.

To create a remote authentication server pointing to the Windows AD:

1. Go to *Authentication* > *Remote Auth. Servers* > *LDAP* and click *Create New*.
2. Configure the following:

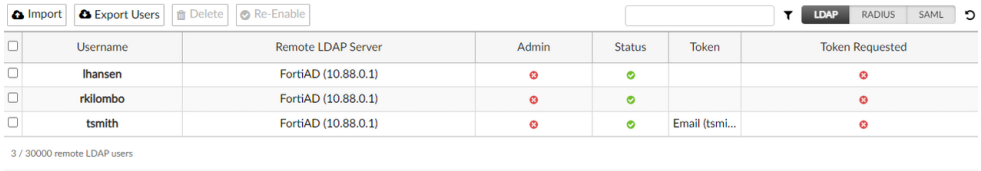
Name	FortiAD
Primary server name / IP	10.88.0.1
Port	389 (or another port if using LDAPS)
Based distinguished name	DC=fortiad,DC=info
Bind type	Regular
Username	<user account used for LDAP bind>
Password	<password of user>
Server Type	Microsoft Active Directory
User object class	person (default)
Username attribute	sAMAccountName (default)
Group object class	group (default)
Obtain group membership from	User attribute
Group membership attribute	memberOf (default)
Secure connection	Enable if using LDAPS or STARTTLS

3. Click *OK*.
4. Edit the *FortiAD* entry.
5. At the bottom, click *Import users*.



6. On the *Import Remote LDAP User* screen, configure the following settings:
   a. For *Remote LDAP Server*, select *FortiAD*.
   b. For *Action*, select *Import users*.
      i. Click *Go*.
      ii. Select the users to import.
      iii. Click *OK*.

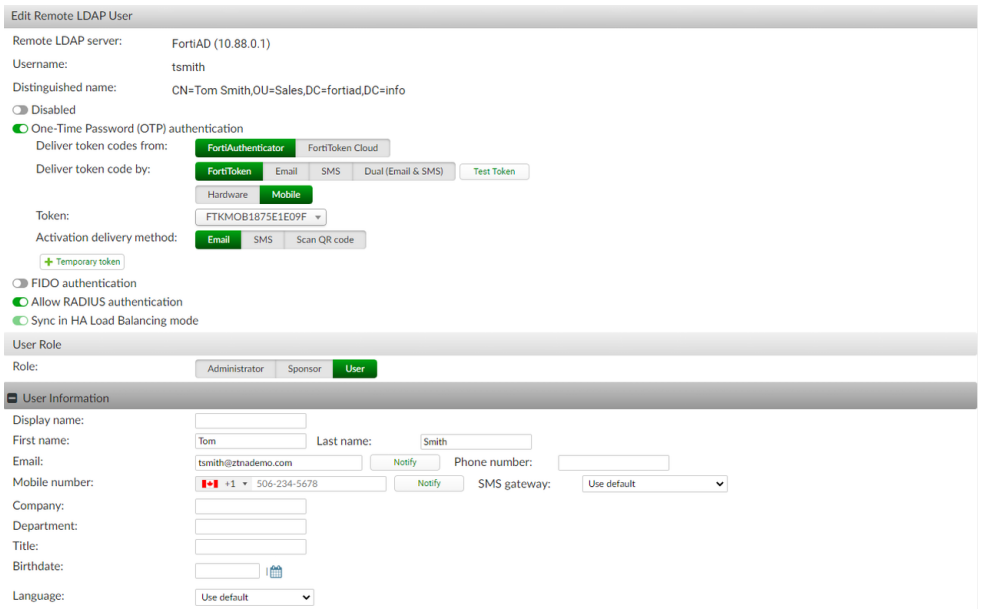
   The screen displays the users that are imported.



   For more details, see LDAP in the FortiAuthenticator Administration Guide.

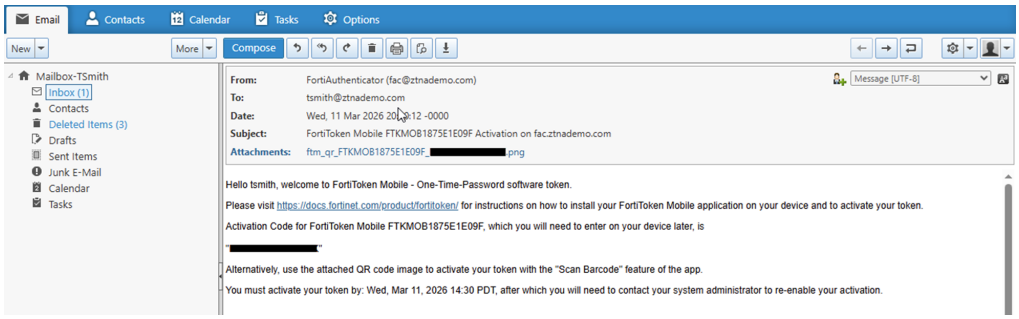
To configure a remote LDAP user to use MFA:

1. Go to *Authentication > User Management > Remote Users*, and edit a user.
2. Enable *One-Time Password (OTP) authentication*.
   a. Set *Deliver token codes from* to *FortiAuthenticator*.
   b. Set *Deliver token by*:
      i. Select *FortiToken > Mobile*.
      ii. Select *Token* from the drop-down list.
      iii. Set the *Activation delivery method* to email.
3. In the *User Information* section, add the email address that will be used for the FortiToken activation.

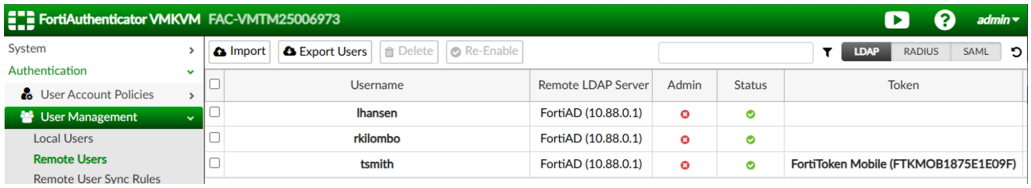


4. Click *OK*.

   An activation email is sent to the user that they can use to install the token to their FortiToken Mobile app.



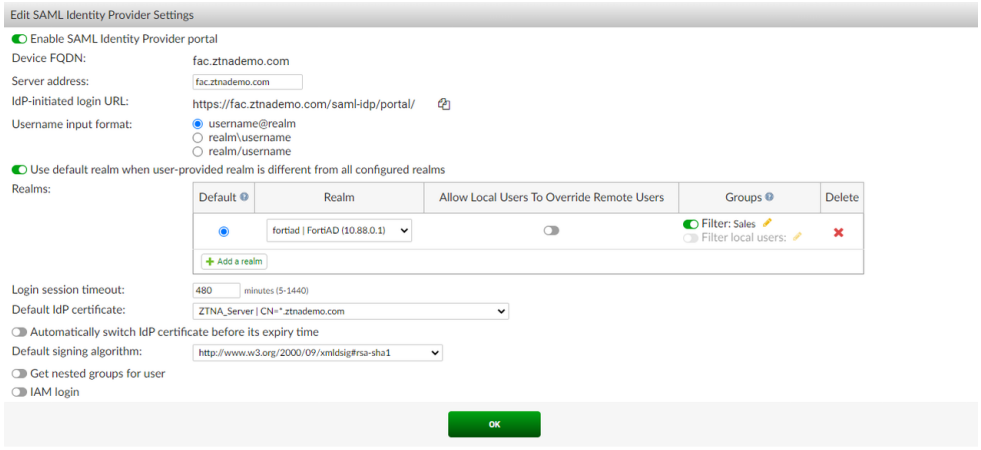
   This guide assumes the token has been activated on a mobile device for user Tsmith.



   For more details, see Remote users in the FortiAuthenticator Administration Guide.

To configure SAML IdP:

1. Go to *Authentication > SAML IdP > General* and enable *Enable SAML Identity Provider portal*.
2. The *Server address* is the device FQDN or IP address (configured in the System Information widget at *System > Dashboard > Status*). In this example, it is *fac.ztnademo.com*.
3. Set *Username input format* to *username@realm*.
4. Click *Add a realm* in the *Realms* table:
   a. Set *Realm* to the just created LDAP realm (*FortiAD*).
   b. Optionally, enable *Filter* and select the required users groups. In this example, *Sales* is configured.
5. Set *Default IdP certificate* to the certificate that will be used in the HTTPS connection to the IdP portal.

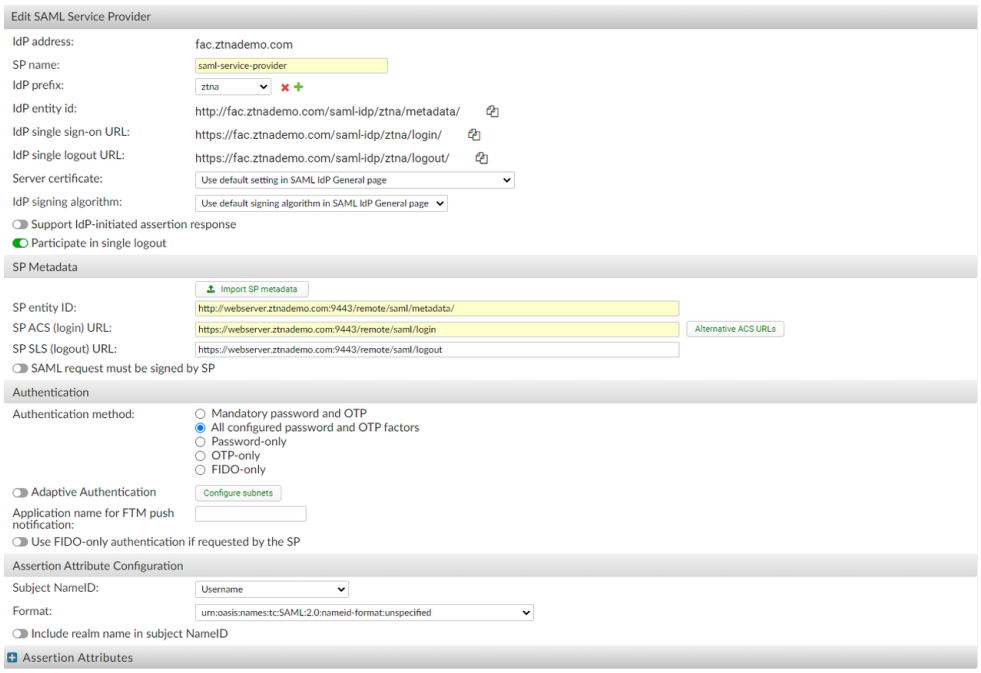


6. Click *OK*.
7. Go to *Authentication > SAML IdP > Service Providers*, and click *Create New* to create a service provider (SP) for the FortiGate SP.
8. Configure the following, which must match what will be configured on the FortiGate:

SP name	saml-service-provider
IdP prefix	ztna
Server certificate	Use default setting in *SAML IdP > General* page.
SP entity ID	http://webserver.ztnademo.com:9443/remote/saml/metadata/
SP ACS (login) URL	https://webserver.ztnademo.com:9443/remote/saml/login
SP SLS (logout) URL	https://webserver.ztnademo.com:9443/remote/saml/logout
Participate in single logout	Enable

Where the *SP entity ID*, *SP ACS (login) URL*, and *SP SLS (logout) URL* break down as follows:

- *webserver.ztnademo.com* - The FQDN that resolves to the FortiGate SP.
- *9443* - The port that is used to map to the FortiGate's SAML SP service.
- */remote/saml* - The custom, user defined fields.
- */metadata*, */login*, and */logout* - The standard convention used to identify the SP entity, log in portal, and log out portal.



9. Click *OK*.
10. Edit the just created SP object and, under *Assertion Attribute*, click *Add Assertion Attribute*.
11. Set *SAML attribute* to the username and set *User attribute* to *Username*, then click *OK*.
12. Click *OK*.

For more details, see Configuring SAML settings in the SAML Interoperability Guide.

Configuring the FortiGate SAML settings

On FortiGate, a SAML user is used to define the SAML SP and IdP settings. This user is then applied to the ZTNA proxy using an authentication scheme, rule, and settings. A ZTNA server is then created to allow access to the SAML SP server so that end users can reach the FortiGate SP's captive portal. The SAML user must then be added to a ZTNA policy to trigger authentication when accessing the ZTNA application gateway.

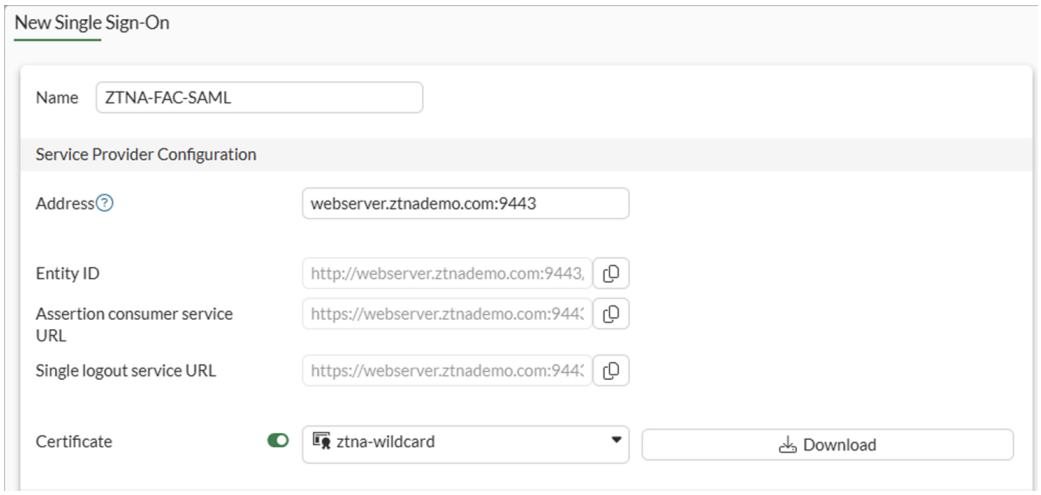
To create a new SAML user/server from the GUI:

1. On the FortiGate, go to *User & Authentication* > *Single Sign-On*.
2. Click *Create New*.
3. Set *Name* to *ZTNA-FAC-SAML*.
4. Set *Address* to *webserver.ztnademo.com:9443*.

   The *Entity ID*, *Assertion consumer service URL* and *Single logout service URL* will be updated. These should be the same URLs you configured on FortiAuthenticator.
5. Enable *Certificate*, then select the certificate used for the client.

   This example uses a wildcard certificate with a Common Name (CN):

   *.ztnademo.com to sign SAML messages that are exchanged between the client and the FortiGate SP.

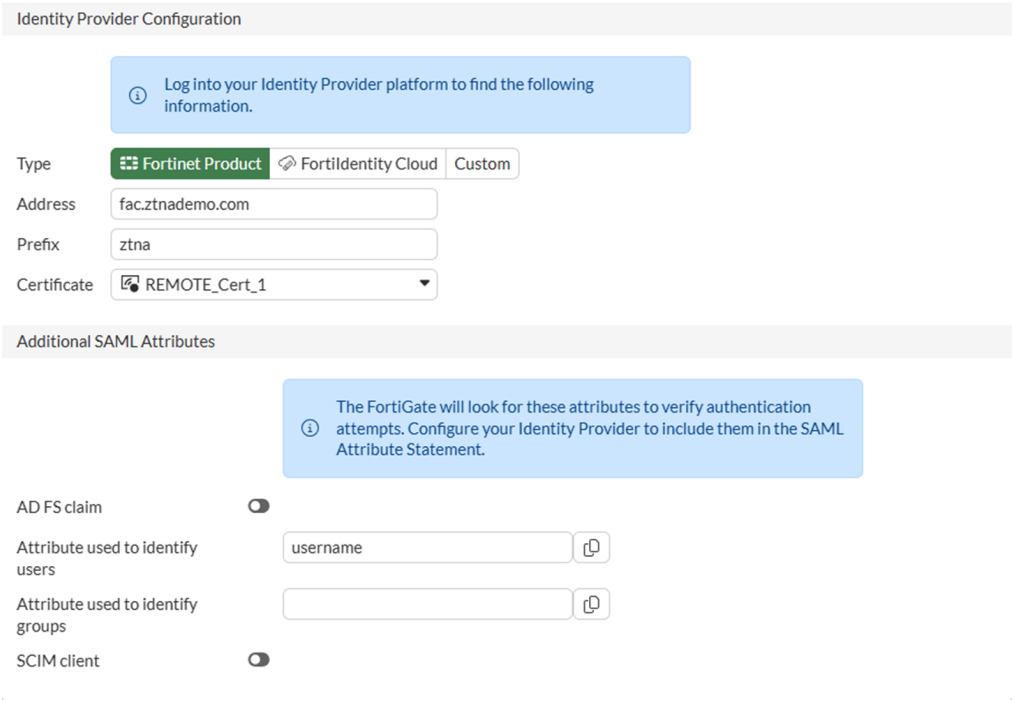


6. Use the settings from the Identity Provider to fill the custom *Identity Provider Details*. In this example, we select *Fortinet Product*, and fill in the fields as follows:

Address	fac.ztnademo.com
Prefix	ztna
IdP certificate	REMOTE_Cert_1

The certificate REMOTE_Cert_1 is the certificate used by the IdP (fac.ztnademo.com). In this example, a wildcard certificate is used.

7. Set *Attribute used to identify users* to *username*. Attributes to identify users and groups are case sensitive.



8. Click *OK* to save the settings.

To create a new SAML user/server from the CLI:

```
config user saml
    edit "ZTNA-FAC-SAML"
        set cert "ztna-wildcard"
        set idp-cert "REMOTE_Cert_1"
        set user-name "username"
        set digest-method sha256
        set entity-id "http://webserver.ztnademo.com:9443/remote/saml/metadata/"
        set single-sign-on-url "https://webserver.ztnademo.com:9443/remote/saml/login"
        set single-logout-url "https://webserver.ztnademo.com:9443/remote/saml/logout"
        set idp-entity-id "http://fac.ztnademo.com/saml-idp/ztna/metadata/"
        set idp-single-sign-on-url "https://fac.ztnademo.com/saml-idp/ztna/login/"
        set idp-single-logout-url "https://fac.ztnademo.com/saml-idp/ztna/logout/"
    next
end
```

To create a user group for the SAML user object:

1. Under *User & Authentication* > *User Groups*, click *Create New*.
2. Set *Name* to *ztna-saml-users*.
3. Under *Remote Groups*, click *Add*.
4. For *Remote Server*, select *ZTNA-FAC-SAML*.
5. Click *OK*.
6. Click *OK* again to save.

To apply the SAML server to proxy authentication from the GUI:

1. Go to *Policy & Objects > Authentication*.
2. Go to the *Authentication Schemes* tab and click *Create New*.
3. Set the name to *ZTNA-SAML-scheme*.
4. Set *Method* to *SAML*.
5. Set *SAML SSO server* to *ZTNA-FAC-SAML*.
6. Click *OK*.
7. Remaining in *Policy & Objects > Authentication*, select the *Authentication Rules* tab in the top menu.
8. Click *Create New*.
9. Set the *Name* to *ZTNA-SAML-rule*.
10. Set *Source Address* to *all*.
11. Set *Incoming interface* to *port3*.
12. Set *Protocol* to *HTTP*.
13. Enable *Authentication Scheme* and select *ZTNA-SAML-scheme*.
14. Set *IP-based Authentication* to *Disable*.
15. Click *OK*.

To apply the SAML server to proxy authentication from the CLI:

```
config authentication scheme
    edit "ZTNA-SAML-scheme"
        set method saml
        set saml-server "ZTNA-FAC-SAML"
    next
end
config authentication rule
    edit "ZTNA-SAML-rule"
        set srcintf "port3"
        set srcaddr "all"
        set ip-based disable
        set active-auth-method "ZTNA-SAML-scheme"
        set web-auth-cookie enable
    next
end
```

Assign an active authentication scheme and captive portal to serve the log in page for the SAML requests.

To configure the active authentication scheme and captive portal from the GUI:

1. Go to *User & Authentication > Authentication Settings*.
2. Enable *Authentication scheme*.
3. Select *ZTNA-SAML-scheme*.
4. Set *Captive portal type* to *FQDN*.
5. Enable *Captive portal*.
6. Select the firewall address *webserver.ztnademo.com*. If this has not be created, create this firewall address.

7. Click *Apply* to save.

To configure the active authentication scheme and captive portal from the CLI:

```
config firewall address
    edit "webserver.ztnademo.com"
        set type fqdn
        set fqdn "webserver.ztnademo.com"
    next
end
config authentication setting
    set active-auth-scheme "ZTNA-SAML-scheme"
    set captive-portal "webserver.ztnademo.com"
end
```

To configure a VIP and a firewall policy to forward IdP authentication traffic to the FortiAuthenticator:

Remote clients connect to the FortiAuthenticator IdP behind the FortiGate using a VIP. In this example, users connect to the FQDN fac.ztnademo.com that resolves to the VIP's external IP address.

1. Configure the VIP to forward traffic to the FortiAuthenticator:
   a. Go to *Policy & Objects* > *Virtual IPs* and navigate to the *Virtual IP* tab.
   b. Click *Create new* .
   c. Configure the following:

Name	FAC-VIP
Interface	Any
External IP address/range	10.0.3.7
Map to > IPv4 address/range	10.88.0.7

   d. Click *OK*.
2. Configure a firewall policy to allow VIP:
   a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
   b. Configure the following:

Name	WAN_to_FAC
Schedule	Always
Action	Accept
Type	Standard
Incoming Interface	port3
Outgoing Interface	port2
Source	All

ZTNA server	FAC-VIP
Service	HTTP. HTTPS
NAT	disabled

c. Click *OK*.

To configure a VIP and firewall policy to forward IdP authentication traffic to the FortiAuthenticator in the CLI:

```
config firewall vip
    edit "FAC-VIP"
        set extip 10.0.3.7
        set mappedip "10.88.0.7"
        set extintf "any"
    next
end
```

```
config firewall policy
    edit 2
        set name "WAN_to_FAC"
        set srcintf "port3"
        set dstintf "port2"
        set action accept
        set srcaddr "all"
        set dstaddr "FAC-VIP"
        set schedule "always"
        set service "ALL"
        set logtraffic all
    next
end
```

Example 1 - Applying SAML and MFA to ZTNA HTTPS access proxy

In this HTTPS web proxy example, a web proxy is used to access a web server. The ZTNA rule verifies device identity and applies the SAML user group to enforce user authentication against the IdP (FortiAuthenticator).

To configure the ZTNA server for HTTPS access proxy:

1. Go to *Policy & Objects* > *ZTNA* and select the *Servers & Portals* tab.
2. Select *Create New*.
3. Set *Name* to *ZTNA-Webserver*.
4. Set *Host* to the IP address or FQDN that will be used to access the web server: webserver.ztnademo.com.

   Endpoints accessing the web server must resolve this FQDN to the ZTNA Application Gateway address (10.0.3.10 in this example).

5. Under *Connects on*, select *Create new for ZTNA port* to provide the following details:

a. Set *Interface* to *port3*.
b. Set *External IP* to *10.0.3.10*.
c. Set *External port* to *9443*.

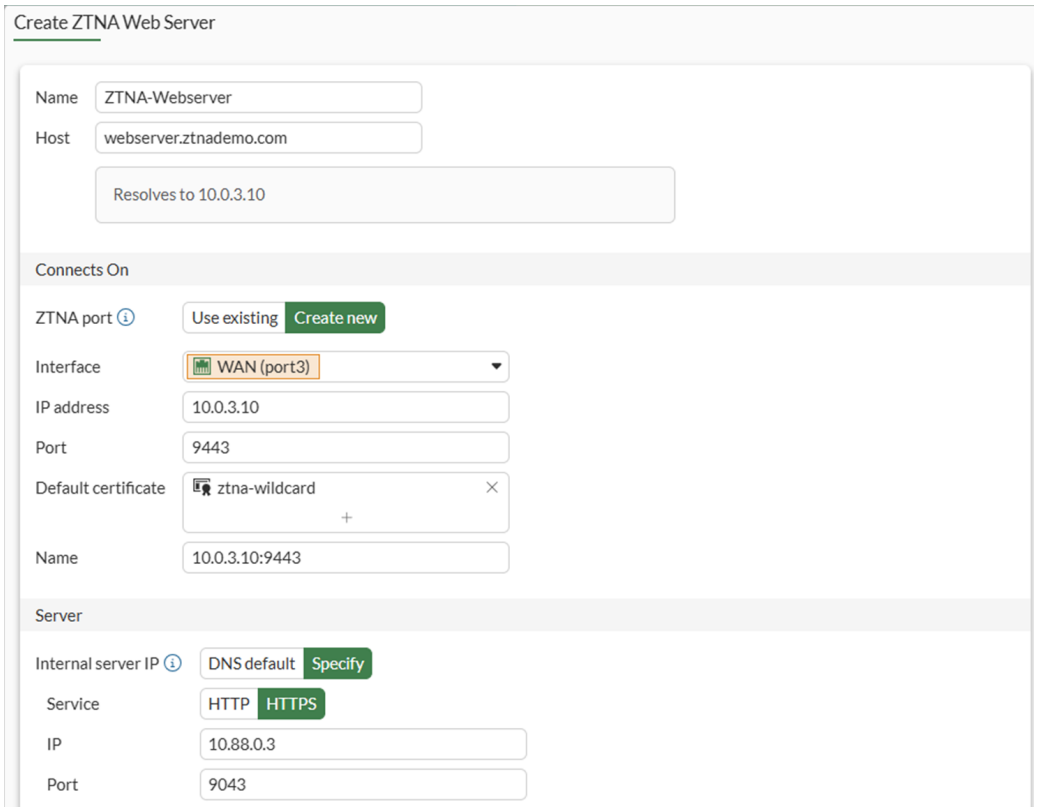
> Verify that the IP address and port do not conflict with management access to the interface. Otherwise, change the IP address to another address on that subnet.

6. Select a *certificate*. Clients will be presented with this certificate when they connect to the ZTNA port.

> This example uses a wildcard certificate with a Common Name (CN): *.ztnademo.com. This certificate is trusted by the endpoints which will access the ZTNA port.

7. Provide a name for the ZTNA port, such as 10.0.3.10:9443.
8. In the *Server* section, configure the internal server settings as follows:
   a. Switch the *Internal server IP* to *Specify*.
   b. Set *Service* to *HTTPS*.
   c. Set *IP to the internal IP for the web server: 10.88.0.3*.
   d. Set *Port* to *9443*.

> The WebServer service is running on port 9043. This configuration will translate clients accessing TCP/9443 to TCP/9043.



9. Click *OK* to save the ZTNA web server.

To configure ZTNA for the web server in the CLI:

```
config firewall vip
    edit "10.0.3.10:9443"
        set type access-proxy
        set server-type https
        set extip 10.0.3.10
        set extintf "port3"
        set extport 9443
        set ssl-certificate "ztna-wildcard"
    next
end
```

```
config firewall access-proxy-virtual-host
    edit "WebServer9443"
        set ssl-certificate "ztna-wildcard"
        set host "webserver.ztnademo.com"
    next
end
```

```
config ztna web-proxy
    edit "ZTNA-Webserver"
        set vip "10.0.3.10:9443"
        set host " WebServer9443"
        config api-gateway
            edit 1
                config realservers
                    edit 1
                        set ip 10.88.0.3
                        set port 9043
                    next
                end
            next
        end
    next
end
```

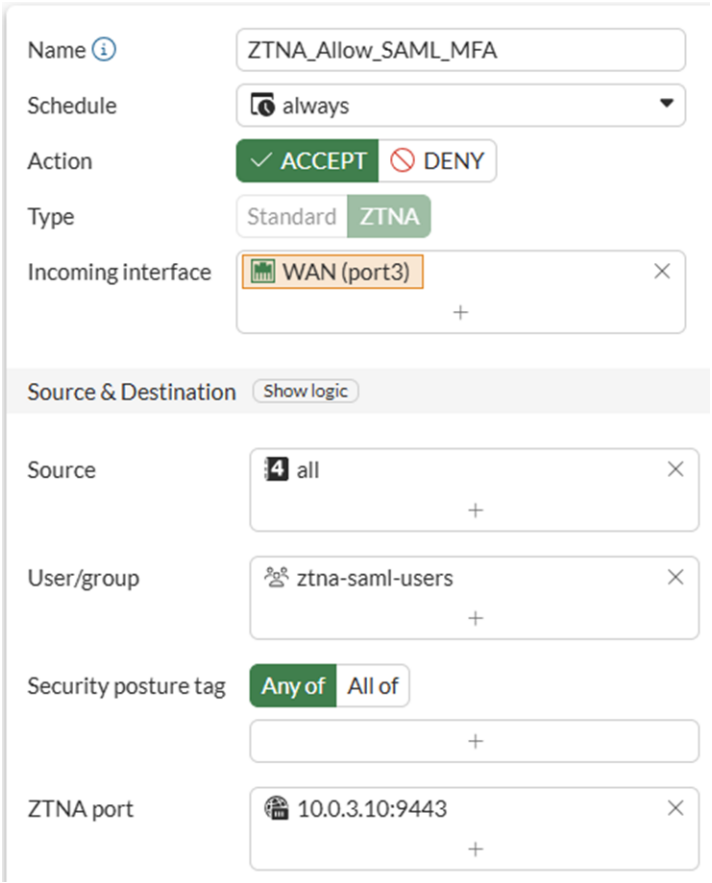
To enable user group authentication in a ZTNA firewall policy:

A simple ZTNA policy is created to apply the SAML authentication.

1. Go to *Policy & Objects > Firewall Policy* and select *Create New*.
2. Configure the following:

Name	ZTNA_Allow_SAML_MFA
Action	Accept
Type	ZTNA
Incoming Interface	port3

Source	All
User/group	ztna-saml-users
ZTNA port	10.0.3.10:9443
Log allowed traffic	All sessions



3. Click *OK* to save the policy.

To enable user group authentication in a ZTNA firewall policy in the CLI:

```
Config firewall policy
    edit 1
        set name "ZTNA_Allow_SAML_MFA"
        set srcintf "port3"
        set dstintf "any"
        set action accept
        set srcaddr "all"
        set dstaddr "10.0.3.10:9443"
        set ztna-destination "Webserver:9043"
        set schedule "always"
        set logtraffic all
        set nat enable
        set groups "ztna-saml-users"
```

```
    next
end
```

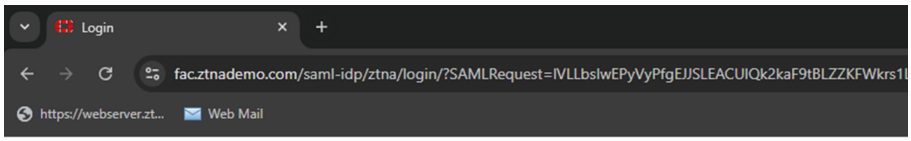
Testing and verification:

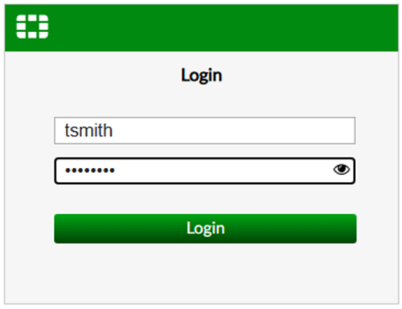
From the remote endpoint, user Tom Smith attempts to connect to the web server over ZTNA:

1. On the remote Windows computer, open FortiClient and register to the EMS server.

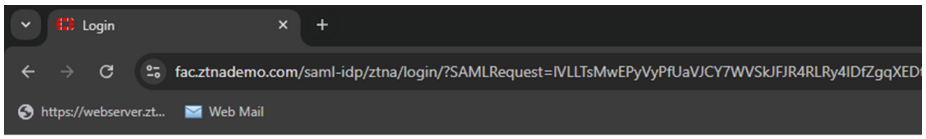
   > It is not necessary to configure a *ZTNA Destination* on the FortiClient for the HTTPS access proxy use case. In fact, configuring a *ZTNA Destination* rule for the website may interfere with its operation.

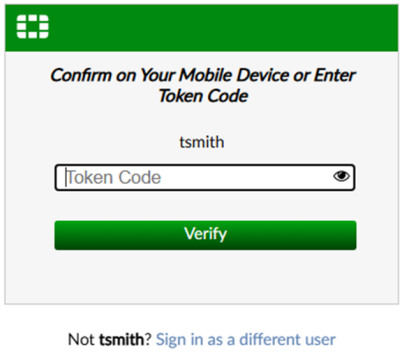
2. Open a browser and attempt to connect to the web server at *https://webserver.ztnademo.com:9443*.
3. Device authentication prompts the user for their device certificate. Select the certificate issued by EMS and click *OK*.
4. FortiGate receives the SAML request and redirects the user to the IdP login screen. Enter the username and password for Tom Smith and click *Login*.



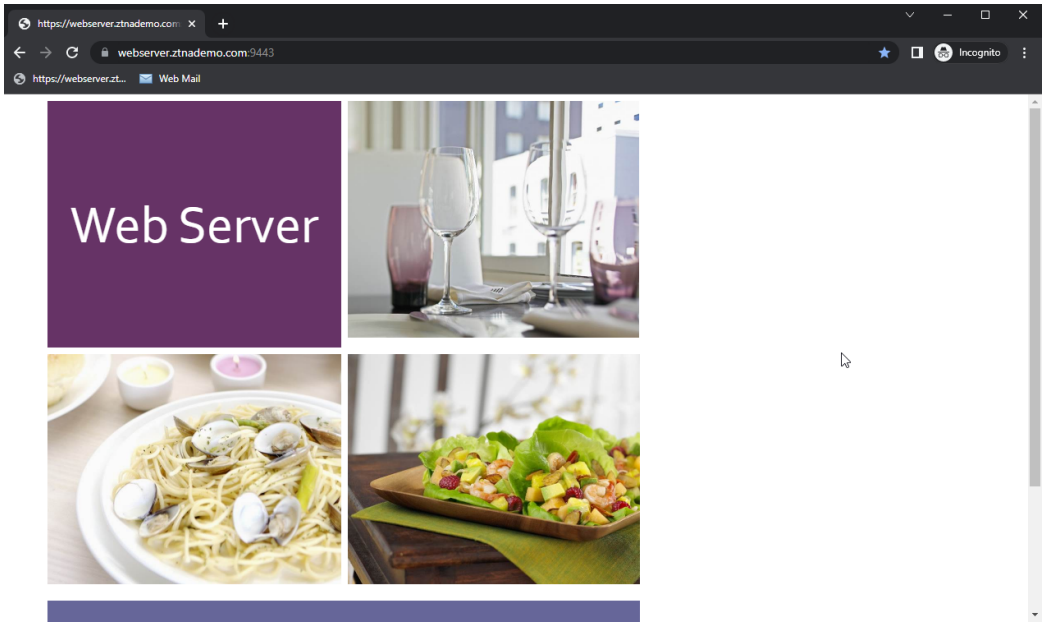


5. A second prompt opens asking for the *Token Code*. Enter the code then click *Verify*.





6. The FortiAuthenticator IdP verifies the login, then sends the SAML assertion back to the user.
7. The browser redirects the assertion to the FortiGate SP, which decides if the user is allowed access.
8. On a successful log in, FortiGate redirects the user to the web page that they are trying to access.



Logs and debugs:

On the FortiGate, a successful connection can be seen in *Log & Report > Forward Traffic* and *Log & Report > ZTNA Traffic*. The CLI can also be used:

```
# execute log filter category 0
# execute log filter field srcip 10.0.3.2
# execute log display

1: date=2026-03-12 time=08:03:36 eventtime=1773327816464678481 tz="-0700" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=12503
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.88.0.3
dstport=9043 dstintf="port2" dstintfrole="dmz" sessionid=282912 srcuuid="b458a65a-f759-51ea-d7df-
ef2e750026d1" dstuuid="0ffed9c4-1e1f-51f1-2bc7-7bff227fae3a" service="tcp/9043"
proxyapptype="ztna-proxy" proto=6 action="accept" policyid=1 policytype="policy"
poluuid="dfc01cba-1c88-51f1-cf94-b9c10b070486" policyname="ZTNA_Allow_SAML_MFA" appcat="unscanned"
duration=188 user="tsmith" group="ztna-saml-users" authserver="ZTNA-FAC-SAML" gatewayid=1
realserverid=1 vip="10.0.3.10:9443" vipincomingip=10.0.3.10 accessproxy="ZTNA-Webserver"
clientdevicemanageable="manageable" clientcert="yes" wanin=303009 rcvdbyte=303009 wanout=4474
lanin=16315 sentbyte=16315 lanout=307926

2: date=2026-03-12 time=08:02:11 eventtime=1773327731253532980 tz="-0700" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="root" srcip=10.0.3.2 srcport=12511
srcintf="port3" srcintfrole="wan" dstip=10.0.3.7 dstport=443 dstintf="port2" dstintfrole="dmz"
srcuuid="b458a65a-f759-51ea-d7df-ef2e750026d1" dstuuid="a43372f8-06af-51ec-8f86-30b451b2ebe0"
srccountry="Reserved" dstcountry="Reserved" sessionid=282931 proto=6 action="client-rst"
policyid=2 policytype="policy" poluuid="2e840d4e-1184-51ec-63b9-9b805a8b7344" policyname="WAN_to_
FAC" service="HTTPS" trandisp="dnat" tranip=10.88.0.7 tranport=443 appcat="unscanned" duration=90
sentbyte=21479 rcvdbyte=14073 sentpkt=49 rcvdpkt=36

3: date=2026-03-12 time=08:00:28 eventtime=1773327628511837705 tz="-0700" logid="0000000010"
type="traffic" subtype="forward" level="notice" vd="root" srcip=10.0.3.2 srcport=12503
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.0.3.10
dstport=9443 dstintf="root" dstintfrole="undefined" sessionid=282912 srcuuid="b458a65a-f759-51ea-
d7df-ef2e750026d1" service="tcp/9443" proxyapptype="web-proxy" proto=6 action="accept" policyid=1
policytype="proxy-policy" poluuid="dfc01cba-1c88-51f1-cf94-b9c10b070486" policyname="ZTNA_Allow_
SAML_MFA" appcat="unscanned" duration=0 vip="10.0.3.10:9443" vipincomingip=10.0.3.10
clientdevicemanageable="manageable" clientcert="yes" wanin=0 rcvdbyte=0 wanout=0 lanin=3778
sentbyte=3778 lanout=4524 msg="Traffic is redirected to SAML authentication."

4: date=2026-03-12 time=08:00:25 eventtime=1773327625893533073 tz="-0700" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=12499
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.0.3.10
dstport=9443 dstintf="root" dstintfrole="undefined" sessionid=282909 srcuuid="b458a65a-f759-51ea-
d7df-ef2e750026d1" service="tcp/9443" proxyapptype="ztna-proxy" proto=6 action="pending"
policyid=1 policytype="policy" poluuid="dfc01cba-1c88-51f1-cf94-b9c10b070486" policyname="ZTNA_
Allow_SAML_MFA" appcat="unscanned" duration=0 vip="10.0.3.10:9443" vipincomingip=10.0.3.10
clientdevicemanageable="unknown" clientcert="no" wanin=0 rcvdbyte=0 wanout=0 lanin=1828
sentbyte=1828 lanout=4141
```

Log number four shows that authentication first hits the ZTNA policy and has an action of **pending**.

Log number three shows that the ZTNA policy redirects the traffic to SAML authentication.

Log number two shows that the WAN_to_FAC policy is hit, indicating the remote user is sending traffic to the FortiAuthenticator.

Log number one shows that the ZTNA policy accepts the traffic and the remote user is permitted to access the ZTNA destination. Notice how the dstip and dstport both change from the application gateway (10.0.3.10:9443) to the real destination (10.88.0.3:9043).

Use the following command to show if the FortiGate's WAD process has an active record of the SAML user login:

```
# diagnose wad user list
ID: 5, VDOM: root, IPv4: 10.0.3.2
  user name   : tsmith
  worker      : 0
  duration    : 6
  auth_type   : Session
  auth_method : SAML
  pol_id      : 17
  g_id        : 8
  user_based  : 0
  expire      : no
  LAN:
    bytes_in=16414 bytes_out=308116
  WAN:
    bytes_in=303036 bytes_out=4474
```

Example 2 - Applying SAML and MFA to a ZTNA TCP forwarding access proxy for RDP connections

In this TCP forwarding access proxy example, RDP connections are allowed to be forwarded to the Windows/EMS server. Traffic to TCP/3389 is allowed through the ZTNA proxy. There are 2 steps to this example: Configure the ZTNA server and Configure the ZTNA policy.

To configure the ZTNA server for TCP forwarding on TCP/3389:

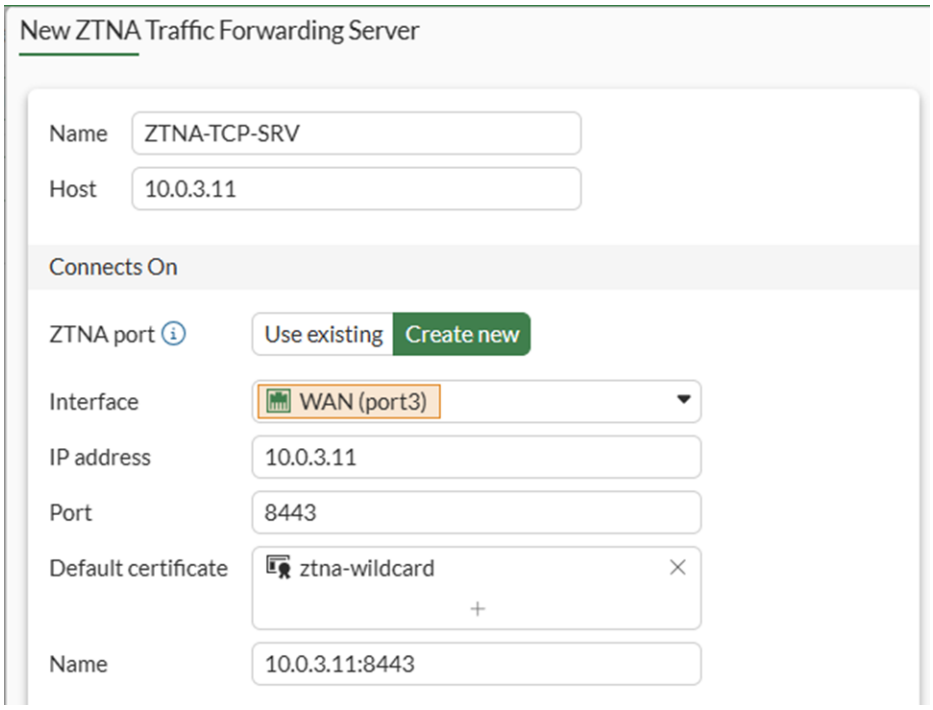
1. Create a firewall address for the Windows/EMS server:
   a. Go to *Policy & Objects* > *Addresses* and select *Address*.
   b. Click *Create new*.
   c. Configure the following:

Name	winserver
Interface	any
Type	Subnet
IP/Netmask	10.88.0.1/32

   d. Click *OK*.
2. Go to *Policy & Objects* > *ZTNA* and select the *Traffic Forwarding* tab. Traffic Forwarding Server is selected by default.
3. Select *Create New*.
4. Set *Name* to *ZTNA-TCP-SRV*.
5. Set *Host* to *10.0.3.11*.

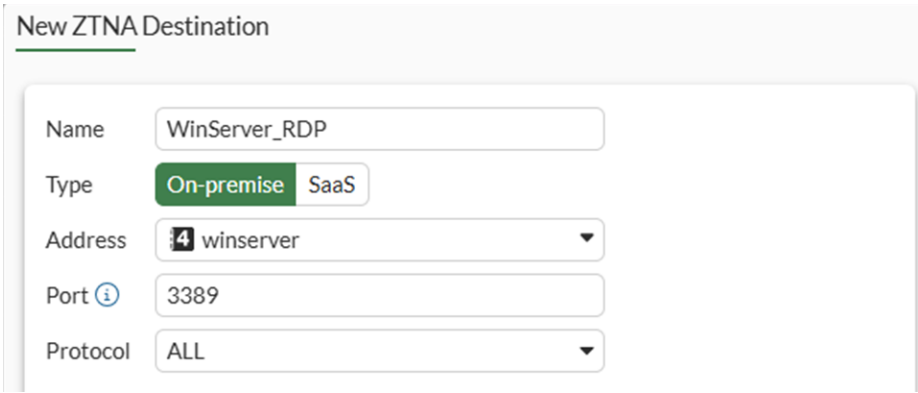
6. For *ZTNA port*, select *Create New* and enter the following details:

Interface	port3
IP address	10.0.3.11
Port	8443
Default certificate	This example uses a wildcard certificate with a Common Name (CN): *.ztnademo.com. This certificate is trusted by the endpoints which will access the ZTNA port.
Name	10.0.3.11:8443



7. Click *OK*.
8. Define a destination for the winserver:
   a. Remaining in the *Traffic Forwarding* section of ZTNA, select *Destination* from the top menu bar.
   b. Select *Create new* to define a new Destination for winserver access:

Name	WinServer_RDP
Type	On-premise
Address	Select the previously created winserver address object
Port	3389
Protocol	TCP[CK1]



c. Click *OK* to save the destination.

To configure the ZTNA server for TCP forwarding on TCP/3389 in the CLI:

1. Configure the access proxy VIP:

```
config firewall vip
    edit "10.0.3.11:8443"
        set type access-proxy
        set server-type https
        set extip 10.0.3.11
        set extintf "port3"
        set extport 8443
        set ssl-certificate "ztna-wildcard"
    next
end
```

2. Configure the destination:

```
config ztna destination
    edit "WinServer_RDP"
        set address "winserver"
        set mappedport 3389
    next
end
```

3. Configure the virtual host:

```
config firewall access-proxy-virtual-host
    edit "10.0.3.11_host"
        set ssl-certificate "ztna-wildcard"
        set host "10.0.3.11"
    next
end
```

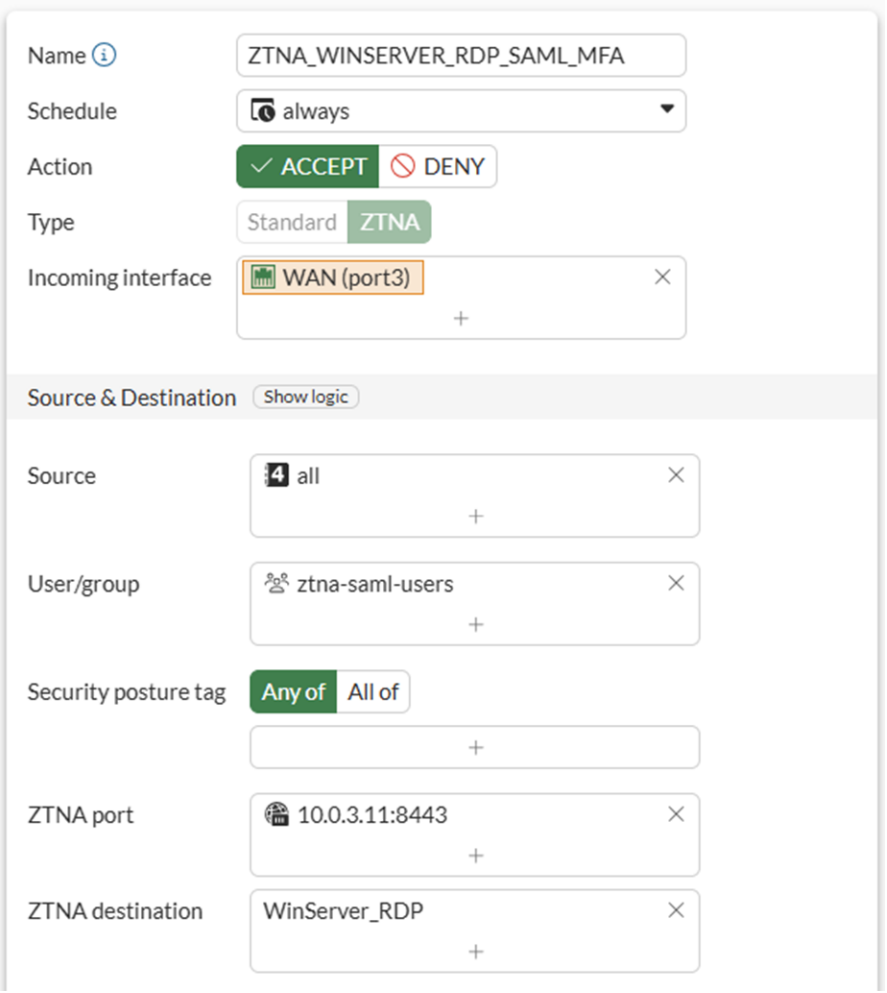
4. Configure the traffic forward proxy:

```
config ztna traffic-forward-proxy
    edit "ZTNA-TCP-SRV"
        set vip "10.0.3.11:8443"
```

```
        set host "10.0.3.11_host"
    next
end
```

To configure the ZTNA policy for TCP forwarding on TCP/3389:

1. Go to *Policy & Objects* > *Firewall Policy*, and click *Create New*.
2. Set *Name* to ZTNA_WINSERVER_RDP_SAML_MFA.
3. Set *Type* to *ZTNA*.
4. Set *Incoming Interface* to *port3*.
5. Set *Source* to *all*.
6. Set the *User/group* to ztna-saml-users.
7. Set the *ZTNA port*: 10.0.3.11:8443.
8. Set the *ZTNA destination*: WinServer_RDP
9. Configure *Log allowed traffic* to log All sessions.
10. Configure the remaining settings as necessary.



11. Click *OK*.

To configure the ZTNA policy for TCP forwarding on TCP/3389 using the CLI:

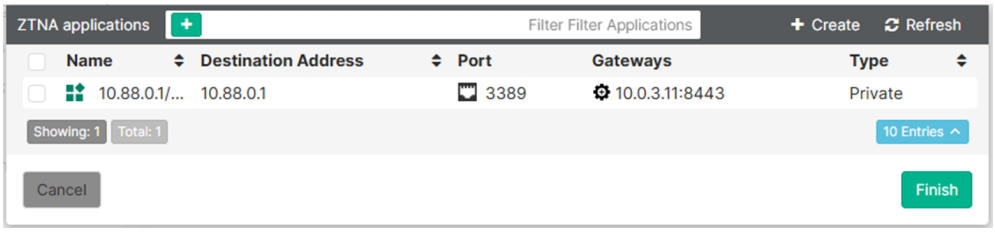
```
config firewall policy
    edit 1
        set name "ZTNA_WINSERVER_RDP_SAML_MFA"
        set srcintf "port3"
        set dstintf "any"
        set action accept
        set srcaddr "all"
        set dstaddr "10.0.3.11:8443"
        set ztna-destination "WinServer_RDP"
        set schedule "always"
        set logtraffic all
        set nat enable
        set groups "ztna-saml-users"
    next
end
```

Testing and verification:

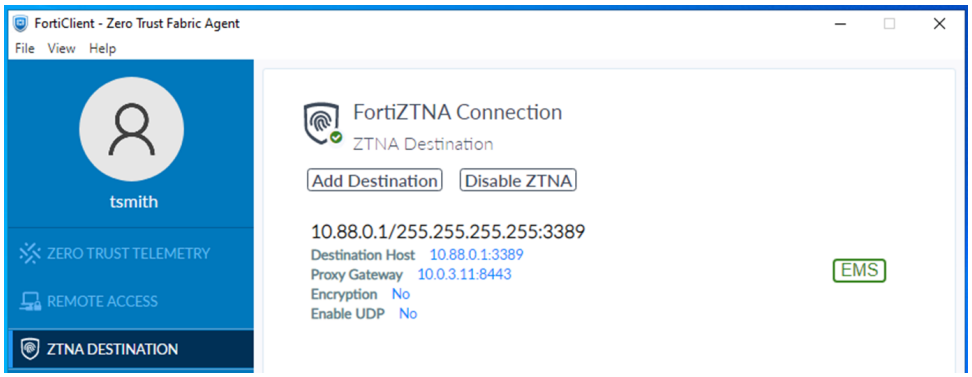
Before connecting, users must have a ZTNA connection rule in FortiClient. This will be configured using Enterprise Management Server (EMS).

To create a ZTNA Destination in EMS:

1. Go to *Endpoint Profiles > ZTNA Destinations*. Select the desired profile or create a new one.
2. In the *Name* field, enter the desired name.
3. In the *Rules* section, click *Add*.
4. The ZTNA applications is populated with the destination configured on the FortiGate.



5. Select the application, then click *Finish*.
6. The destination now appears as a rule entry.
7. Click *Save* to save the ZTNA Destinations profile.
8. Ensure this profile is applied to the endpoint policy assigned to the remote endpoint.
9. After some time, the FortiClient will receive the profile update and the configuration will appear under the *ZTNA DESTINATION* tab on FortiClient.

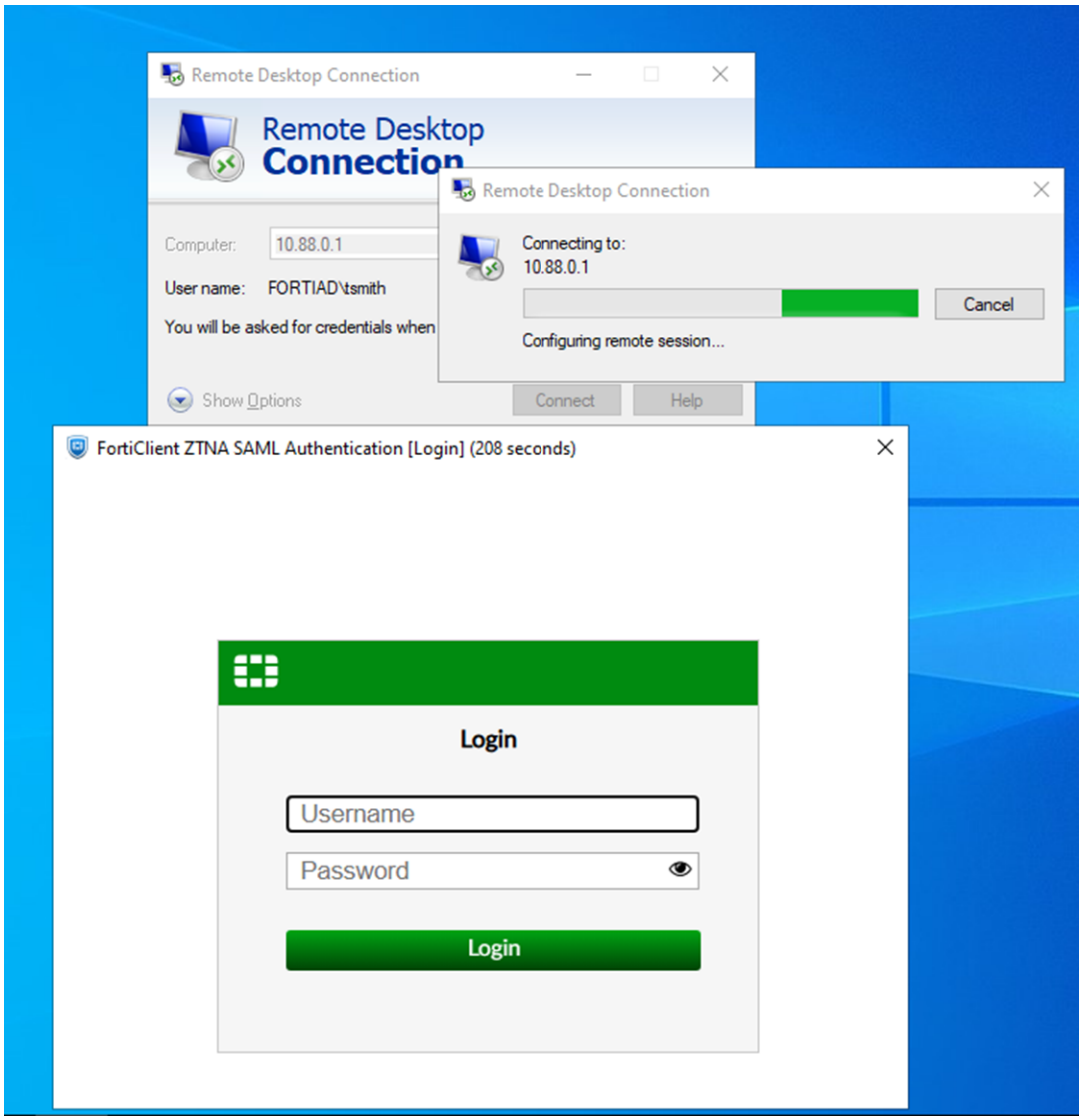


To test the RDP access:

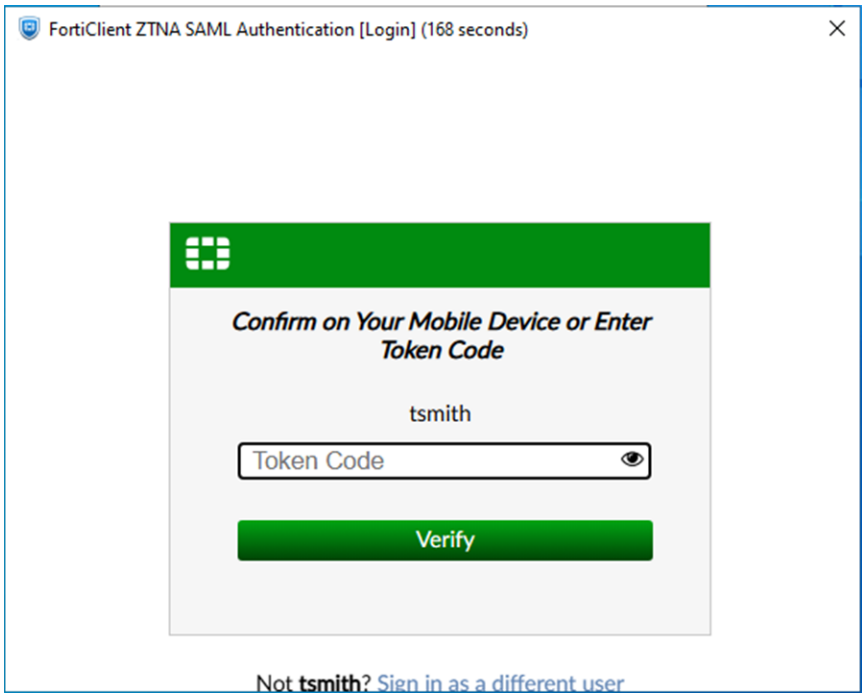
1. On the remote PC, open a new RDP connection.
2. Enter the IP address 10.88.0.1. By default, RDP session use port 3389.

   When the connection to the ZTNA application gateway is established, FortiGate will redirect the SAML login request to the FortiAuthenticator IdP.

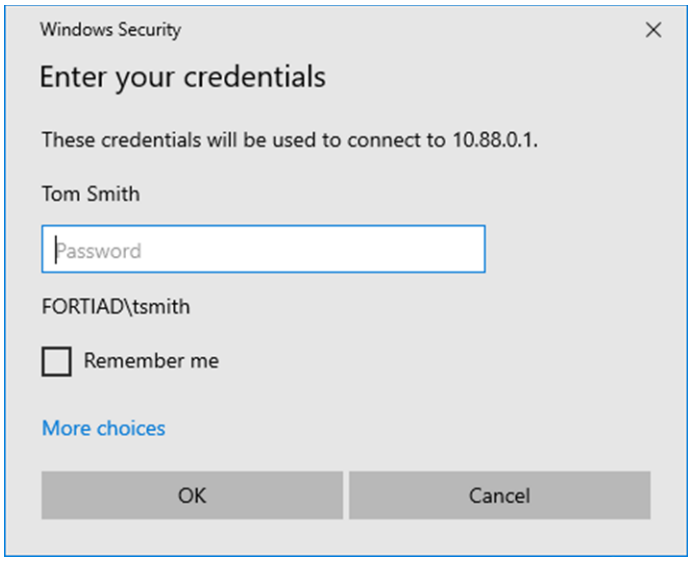
   A FortiClient prompt will open with the FortiAuthenticator login screen.



3. Enter the username and password then click *Login*.
4. A second prompt opens asking for the *Token Code*. Enter the code from your FortiToken app, then click *Verify*.



5. FortiAuthenticator verifies the token code, determines if the login is successful, then sends the SAML assertion back to the client.
6. The client redirects the response back to the FortiGate SP.
7. If the log in was successful, the user can now log on to the RDP session.



Logs and debugs:

On the FortiGate, a successful connection can be seen in *Log & Report* > *Forward Traffic* and *Log & Report* > *ZTNA Traffic*. The CLI can also be used:

```
# execute log filter category 0
# execute log filter field srcip 10.0.3.2
```

```
# execute log display

1: date=2026-03-12 time=11:06:49 eventtime=1773338809031245180 tz="-0700" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=13002
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.88.0.1
dstport=3389 dstintf="port2" dstintfrole="dmz" sessionid=287556 srcuuid="b458a65a-f759-51ea-d7df-
ef2e750026d1" dstuuid="cbf3c8b0-1e33-51f1-8d38-aad44938f061" service="RDP" proxyapptype="ztna-
proxy" proto=6 action="accept" policyid=1 policytype="policy" poluuid="dfc01cba-1c88-51f1-cf94-
b9c10b070486" policyname="ZTNA_WINSERVER_RDP_SAML_MFA" appcat="unscanned" duration=23
user="tsmith" group="ztna-saml-users" authserver="ZTNA-FAC-SAML" vip="10.0.3.11:8443"
vipincomingip=10.0.3.11 accessproxy="ZTNA-TCP-SRV" ztnadestination="WinServer_RDP"
clientdevicemanageable="manageable" clientcert="yes" wanin=0 rcvdbyte=0 wanout=0 lanin=4234
sentbyte=4234 lanout=5459

2: date=2026-03-12 time=11:06:44 eventtime=1773338803583557253 tz="-0700" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="root" srcip=10.0.3.2 srcport=13004
srcintf="port3" srcintfrole="wan" dstip=10.0.3.7 dstport=443 dstintf="port2" dstintfrole="dmz"
srcuuid="b458a65a-f759-51ea-d7df-ef2e750026d1" dstuuid="a43372f8-06af-51ec-8f86-30b451b2ebe0"
srccountry="Reserved" dstcountry="Reserved" sessionid=287559 proto=6 action="client-rst"
policyid=2 policytype="policy" poluuid="2e840d4e-1184-51ec-63b9-9b805a8b7344" policyname="WAN_to_
FAC" service="HTTPS" trandisp="dnat" tranip=10.88.0.7 tranport=443 appcat="unscanned" duration=10
sentbyte=4873 rcvdbyte=4373 sentpkt=13 rcvdpkt=13

3: date=2026-03-12 time=11:06:26 eventtime=1773338786165014542 tz="-0700" logid="0000000010"
type="traffic" subtype="forward" level="notice" vd="root" srcip=10.0.3.2 srcport=13002
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.0.3.11
dstport=8443 dstintf="root" dstintfrole="undefined" sessionid=287556 srcuuid="b458a65a-f759-51ea-
d7df-ef2e750026d1" service="tcp/8443" proxyapptype="web-proxy" proto=6 action="accept" policyid=1
policytype="proxy-policy" poluuid="dfc01cba-1c88-51f1-cf94-b9c10b070486" policyname="ZTNA_
WINSERVER_RDP_SAML_MFA" appcat="unscanned" duration=0 vip="10.0.3.11:8443" vipincomingip=10.0.3.11
clientdevicemanageable="manageable" clientcert="yes" wanin=0 rcvdbyte=0 wanout=0 lanin=3910
sentbyte=3910 lanout=4582 msg="Traffic is redirected to SAML authentication."

4: date=2026-03-12 time=11:06:26 eventtime=1773338786073230647 tz="-0700" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=12998
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.0.3.11
dstport=8443 dstintf="root" dstintfrole="undefined" sessionid=287552 srcuuid="b458a65a-f759-51ea-
d7df-ef2e750026d1" service="tcp/8443" proxyapptype="ztna-proxy" proto=6 action="pending"
policyid=1 policytype="policy" poluuid="dfc01cba-1c88-51f1-cf94-b9c10b070486" policyname="ZTNA_
WINSERVER_RDP_SAML_MFA" appcat="unscanned" duration=0 vip="10.0.3.11:8443" vipincomingip=10.0.3.11
clientdevicemanageable="unknown" clientcert="no" wanin=0 rcvdbyte=0 wanout=0 lanin=1827
sentbyte=1827 lanout=4141
```

Log number four shows that authentication first hits the ZTNA policy and has an action of **pending**.

Log number three shows that the ZTNA policy redirects the traffic to SAML authentication.

Log number two shows that the WAN_to_FAC policy is hit, indicating the remote user is sending traffic to the FortiAuthenticator.

Log number one shows that the ZTNA policy accepts the traffic and the remote user is permitted to access the ZTNA destination. Notice how the dstip and dstport both change from the application gateway (10.0.3.11:8443) to the real destination (10.88.0.1:3389).

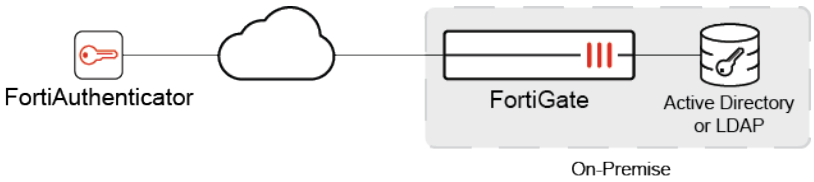
Use the following command to show if the FortiGate's WAD process has an active record of the SAML user login:

```
# diagnose wad user list

ID: 6, VDOM: root, IPv4: 10.0.3.2
  user name    : tsmith
  worker       : 0
  duration     : 192
  auth_type    : Session
  auth_method  : SAML
  pol_id       : 17
  g_id         : 8
  user_based   : 0
  expire       : no
  LAN:
    bytes_in=95081 bytes_out=317409
  WAN:
    bytes_in=303552 bytes_out=72039
```

Secure LDAP connection from FortiAuthenticator with zero trust tunnel example

When an on-premise Active Directory or LDAP server must be accessed remotely, a secure connection is important. LDAP without security should not be exposed on the internet, as user information and passwords are transferred in plain text.

This example describes how to configure a secure LDAP connection from a remote FortiAuthenticator to an on-premise AD server.



The on-premise FortiGate acts as a ZTNA application gateway to allow the FortiAuthenticator access to the AD using a TCP forwarding access proxy. Traditionally, this requires endpoints to have FortiClient installed with ZTNA destinations configured in order to connect and authenticate with a client certificate. In this scenario, the FortiAuthenticator will operate a local certificate authority, which generates a client certificate for the connection. The local root CA certificate is exported and installed on the FortiGate in order to authenticate and trust the client connection. This replaces the need for a FortiClient EMS server to manage the client certificate and act as the certificate authority.

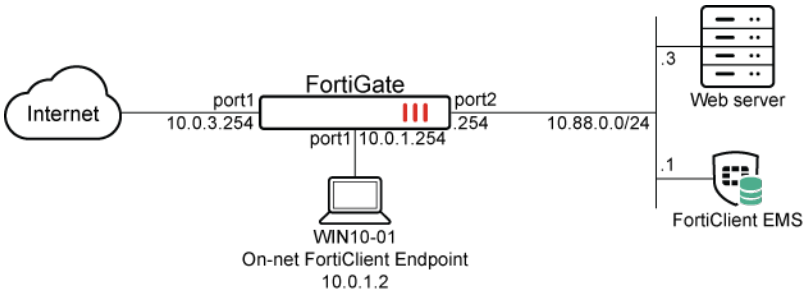
For detailed configuration steps, see Setting up a zero trust tunnel on the FortiAuthenticator page.

ZTNA IP MAC based access control example

In this example, firewall policies are configured that use security posture tags to control access between on-net devices and an internal web server. This mode does not require the use of the access proxy, and only uses security posture tags for access control. Traffic is passed when the FortiClient endpoint meets two conditions.

1. It is tagged with the *Domain-Users* security posture tag, identifying the device as logged on to the Domain.
2. It has the *High* importance classification tag indicating the device is High importance and low risk.

Traffic is denied when the FortiClient endpoint is tagged with *Malicious-File-Detected*.

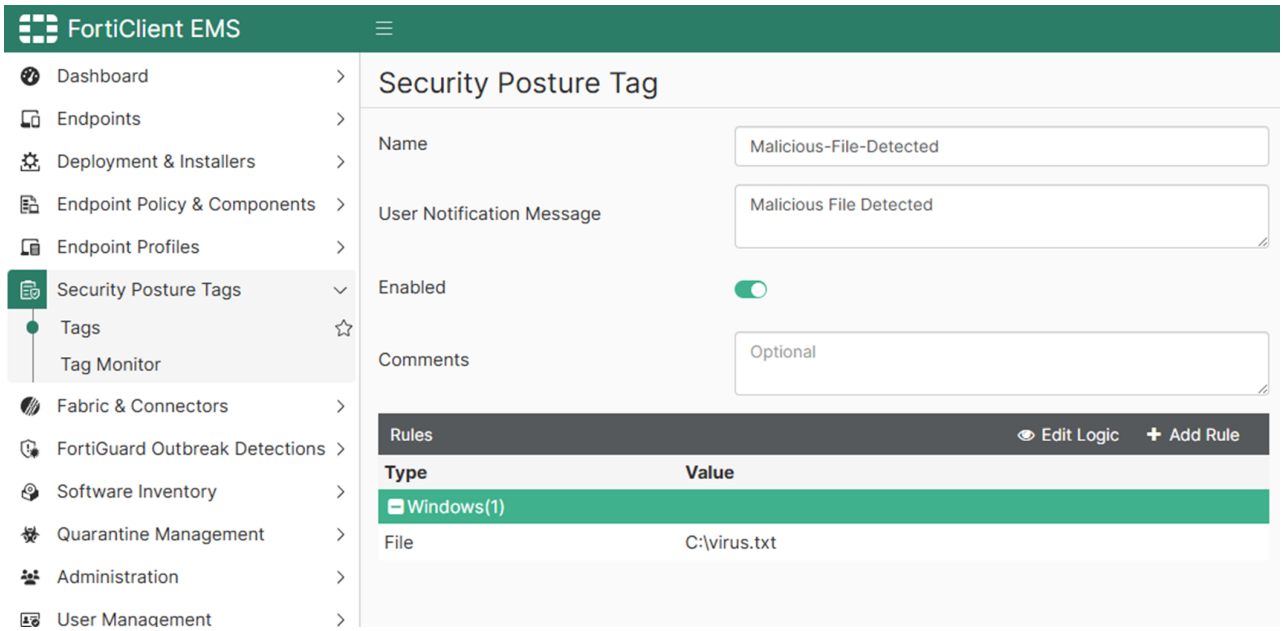


This example assumes that the FortiGate EMS fabric connector is already successfully connected, and a domain has been added to the FortiClient EMS. See Security Fabric connectors on page 4301 and Adding endpoints using an AD domain server

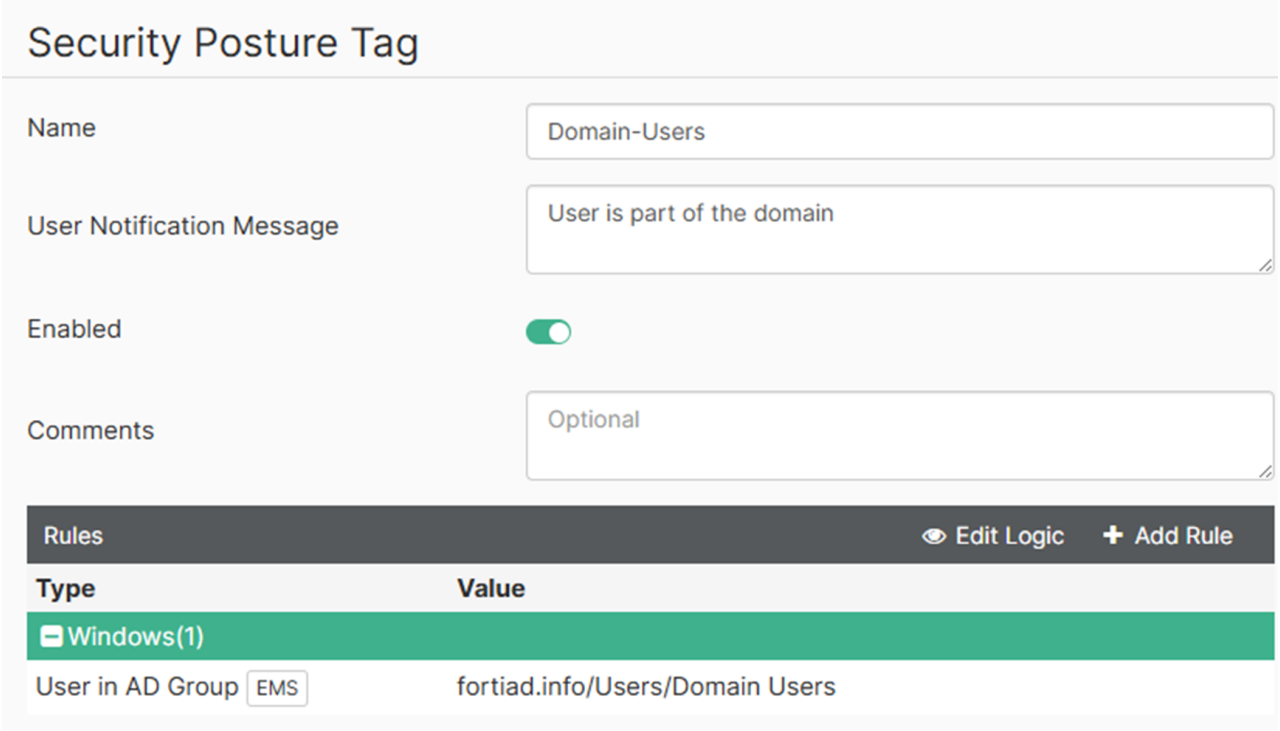
> To configure ZTNA in the GUI, go to *System > Feature Visibility* and enable *Zero Trust Network Access*.

To configure Zero Trust tagging rules on the FortiClient EMS:

1. Log in to the FortiClient EMS.
2. Go to *Security Posture Tags > Tags*, and click *Create*.
3. In the *Name* field, enter *Malicious-File-Detected*.
4. Specify a *User Notification Message*, such as *Malicious File Detected*.
5. Click *Add Rule* then configure the rule:
   a. For *OS*, select *Windows*.
   b. From the *Rule Type* dropdown list, select *File*.
   c. Enter a file name, such as *C:\virus.txt*.
   d. Click *Save*.



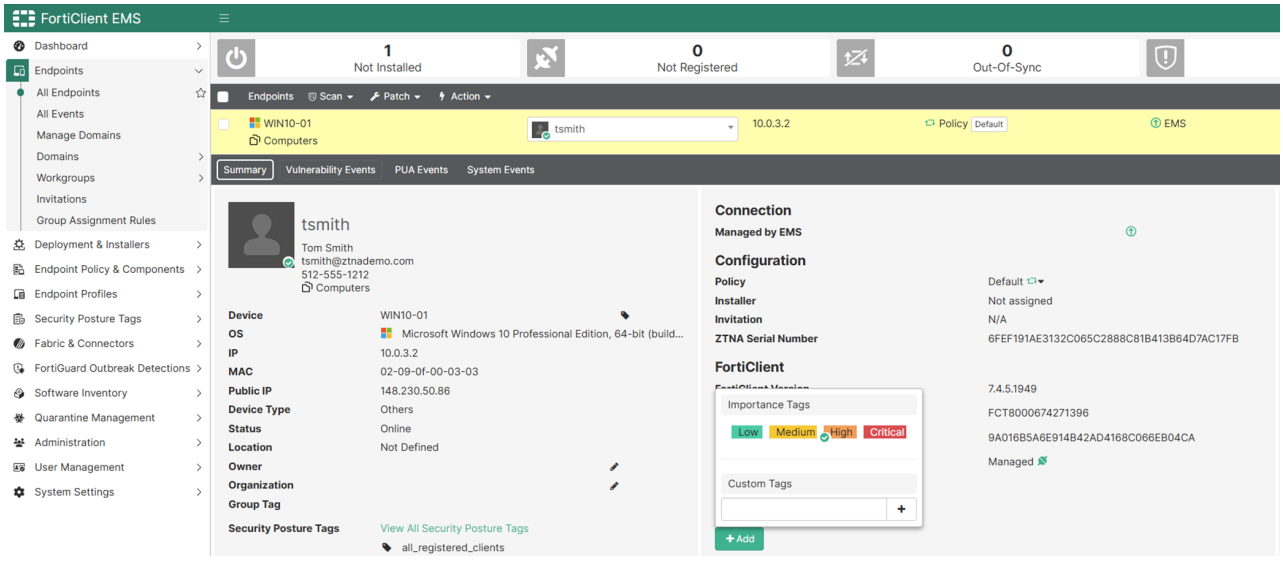
6. Click *Save*.
7. Click *Create* again.
8. In the *Name* field, enter *Domain-Users*.
9. Specify a *User Notification Message*, such as *User is part of the domain*.
10. Click *Add Rule* then configure the rule:
    a. For *OS*, select *Windows*.
    b. From the *Rule Type* dropdown list, select *User in AD Group*.
    c. For *AD Group*, select the *Domain-Users AD group*.
    d. Click *Save*.



11. Click *Save*.

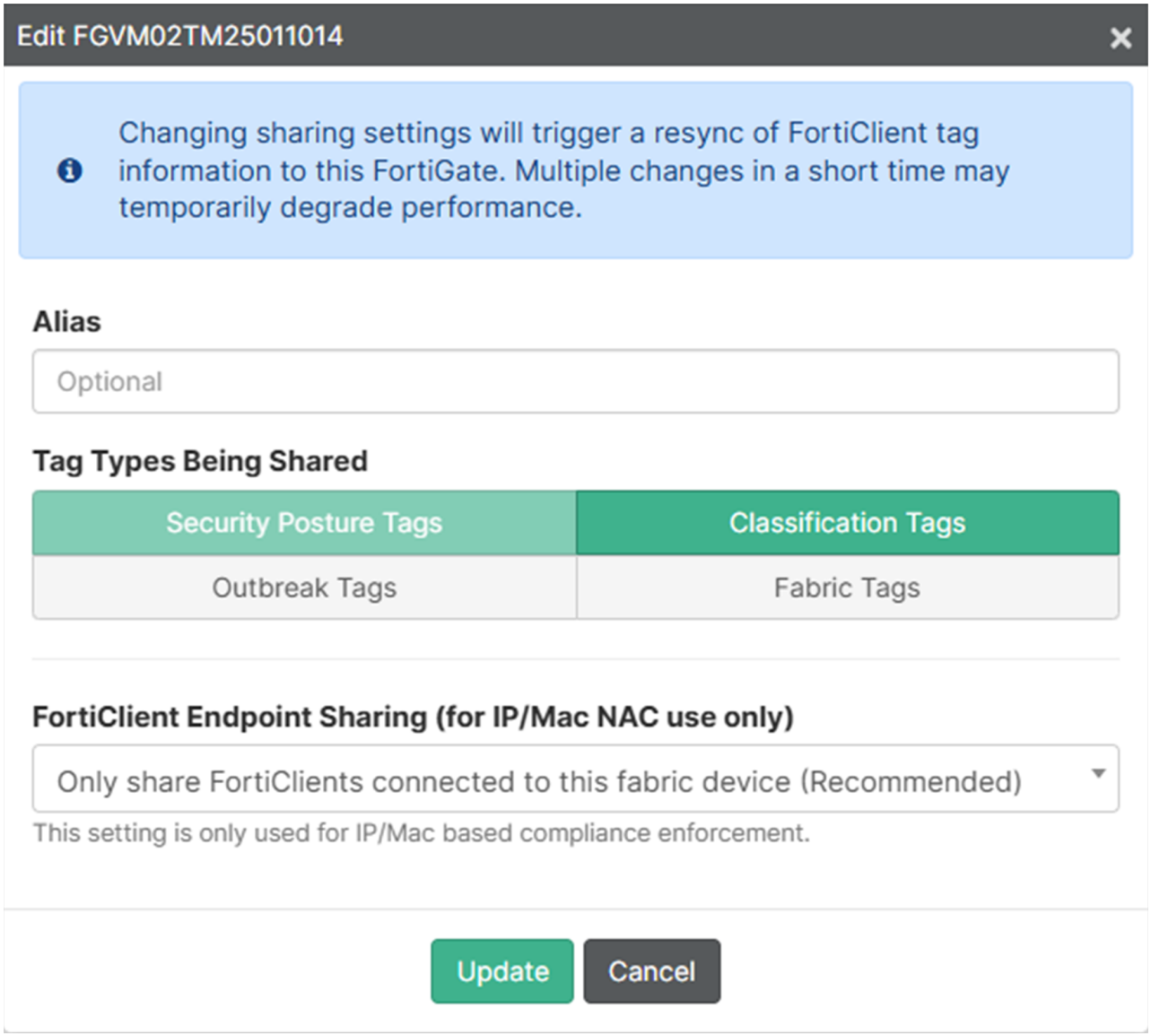
To configure a classification tag on the FortiClient EMS:

1. Go to *Endpoint* > *All Endpoints*.
2. Select the computer that will be granted access. This computer should be already registered to FortiClient EMS.
3. In the *Summary* tab, under *Classification Tags*, click *Add* and then set to High Importance.



4. Go to *Fabric & Connectors* > *Fabric Devices*.

5. Select the connecting FortiGate, then click the icon with the square and pencil to edit the FortiGate connector.
6. Under *Tag Types Being Shared*, add *Classification Tags*.

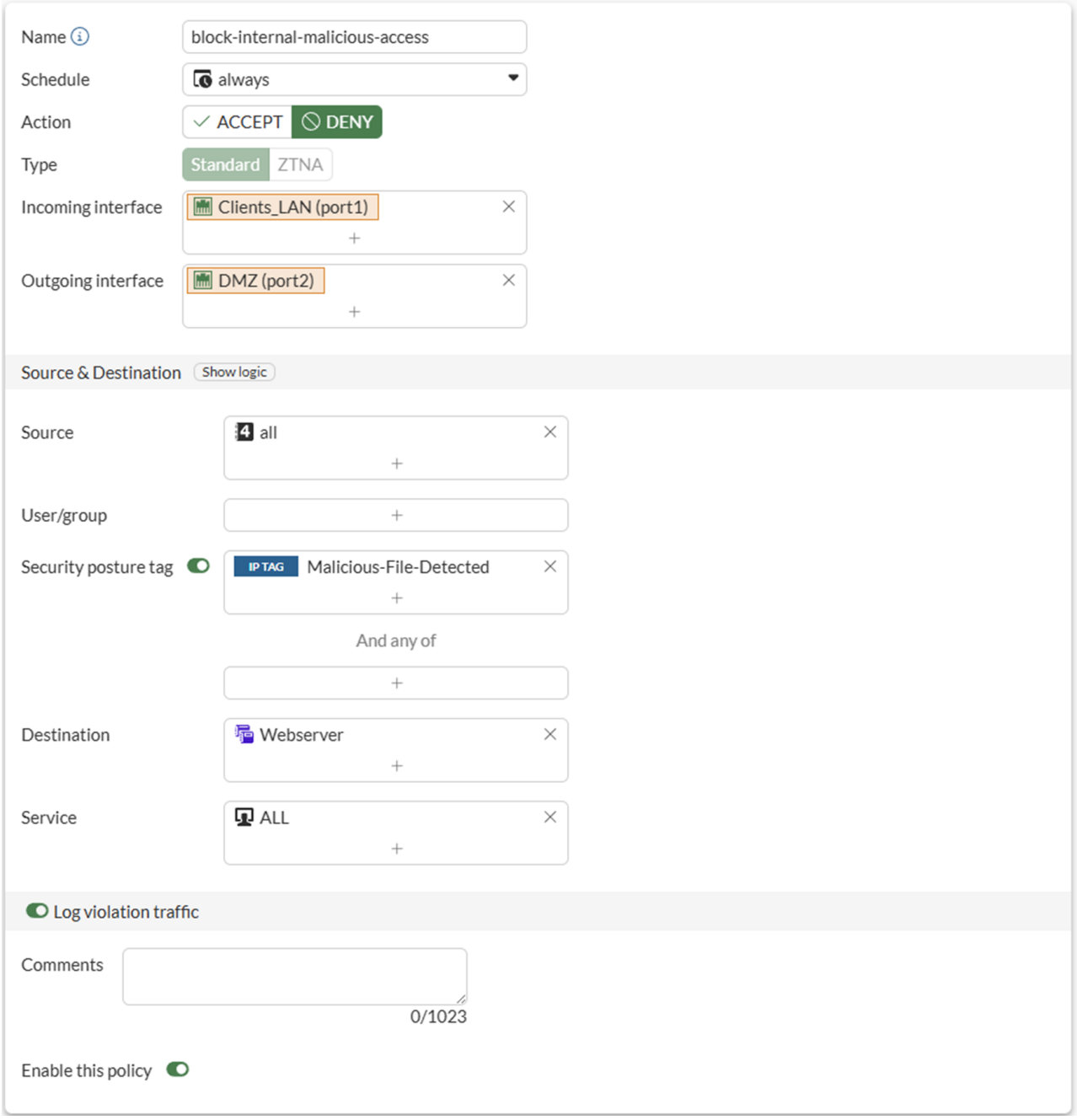


7. Click *Update*. The connector should reflect two black tags to indicate two tag types being shared.

To configure a firewall policy with IP/MAC based access control to deny traffic in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
2. Set *Name* to *block-internal-malicious-access*.
3. Set *Action* to *DENY*.
4. Set *Type* to *Standard*.
5. Set *Incoming Interface* to *port1*.
6. Set *Outgoing Interface* to *port2*.
7. Set *Source* to *all*.

8. Enable *Security posture tag* and select *[IP TAG] Malicious-File-Detected*.
9. Set *Destination* to the address of the Web server. If no address is created, create a new address object for 10.88.0.3/32.
10. Set *Service* to *ALL*.
11. Enable *Log Violation Traffic*.
12. Configuring the remaining settings as needed.



13. Click *OK*.

To configure a firewall policy with IP/MAC based access control to allow access in the GUI:

1. Go to *Policy & Objects > Firewall Policy* and click *Create New*.
2. Set *Name* to *allow-internal-access*.
3. Set *Action* to *ACCEPT*.
4. Set *Type* to *Standard*.
5. Set *Incoming interface* to *port1*.
6. Set *Outgoing interface* to *port2*.
7. Set *Source* to *all*.
8. Enable *Security posture tag* and select *[MAC TAG] Domain-Users*, and set the *And any of* field to *[CLASS MAC] High*.

   The secondary group of tags is used with a logical And operator. The secondary group tags can be same or different types than the primary group.
9. Set *Destination* to the address of the Web server.
10. Set *Service* to *ALL*.
11. Enable *Log allowed traffic* and set it to *All Sessions*.
12. Configuring the remaining settings as needed.
13. Click *OK*.

To configure firewall policies with IP/MAC based access control to block and allow access in the CLI:

```
config firewall policy
    edit 17
        set name "block-internal-malicious-access"
        set srcintf "port1"
        set dstintf "port2"
        set ztna-status enable
        set srcaddr "all"
        set dstaddr "Webserver"
        set ztna-ems-tag "EMS1_ZTNA_Malicious-File-Detected"
        set schedule "always"
        set service "ALL"
        set logtraffic all
    next
    edit 19
        set name "allow-internal-access"
        set srcintf "port1"
        set dstintf "port2"
        set action accept
        set ztna-status enable
        set srcaddr "all"
        set dstaddr "Webserver"
        set ztna-ems-tag "MAC_EMS1_ZTNA_Domain-Users"
        set ztna-ems-tag-secondary "MAC_EMS1_CLASS_High"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set nat enable
```

```
    next
end
```

When multiple tags are selected with `set ztna-ems-tag <tags>`, matching occurs using a logical OR operator. Any single tag that matches will return true.

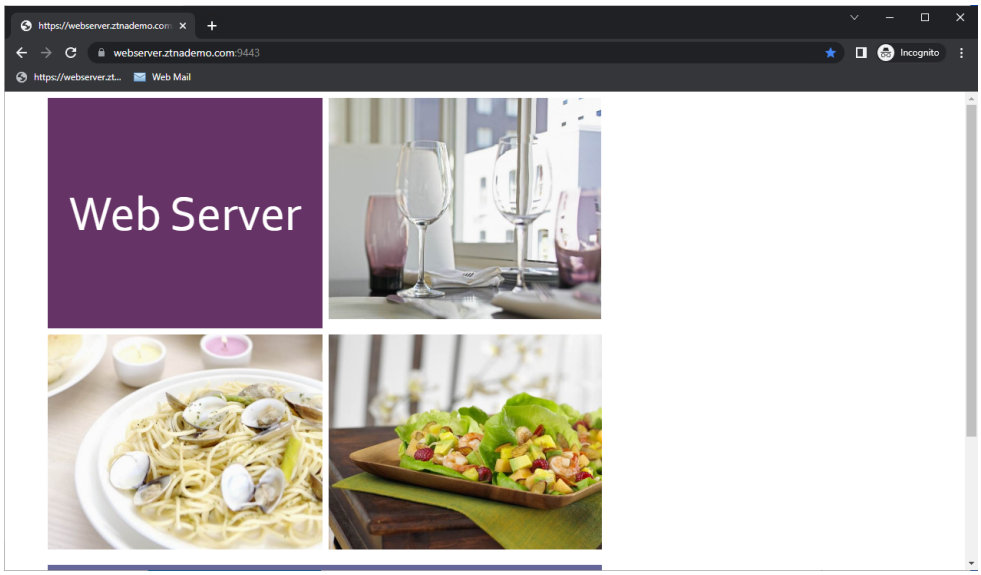
The `set ztna-ems-tag-secondary <tags>` command allows a second group of tags to be specified that is used with a logical And operator. The secondary group tags can be same or different types than the primary group.

The `set ztna-tags-match-logic {and | or}` command does not change the logical operator. It is applied by WAD to tags selected for ZTNA proxy-policy.

Testing the access to the web server from the on-net client endpoint

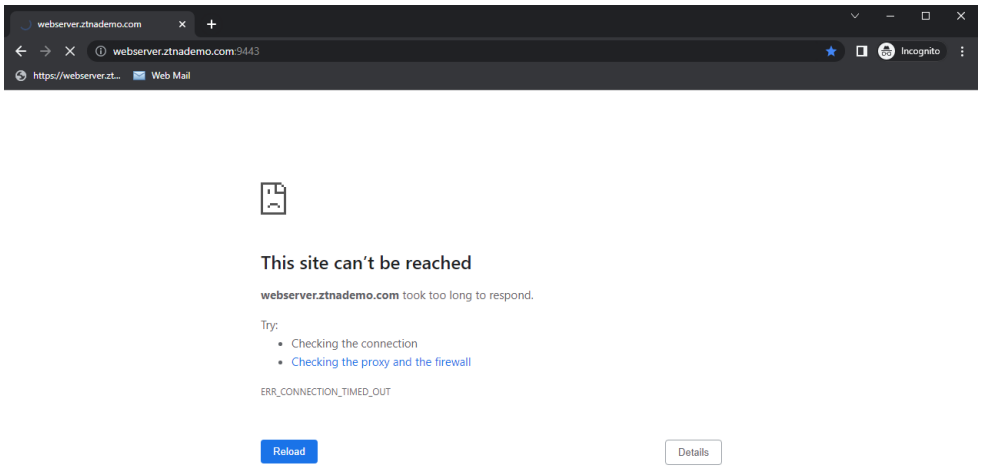
Access allowed:

1. On the internal PC, open FortiClient.
2. On the *Zero Trust Telemetry* tab, make sure that you are connected to the EMS server.
3. Open a browser and enter the address of the server.
4. The FortiGate matches your security posture by verifying your security posture tags and matching the corresponding `allow-internal-access` firewall policy, and you are allowed access to the web server.



Access denied:

1. On the internal PC, trigger the Zero Trust Tagging Rule by creating the file in C:\virus.txt.
2. Open a browser and enter the address of the server.
3. FortiGate checks your security posture. Because EMS has tagged the PC with the *Malicious-File-Detected* tag, it matches the *block-internal-malicious-access* firewall policy.
4. You are denied access to the web server.



Logs and debugs

Access allowed:

```
# diagnose endpoint ec-shm list
Record #0:
IP Address = 10.0.1.2
                MAC Address = 02:09:0f:00:01:02
                MAC list =
                VDOM = root (0)
                EMS serial number: FCTEMS8825007166
                EMS tenant id: 00000000000000000000000000000000
                Client cert SN: 6FEF191AE3132C065C2888C81B413B64D7AC17FB
                Public IP address: 148.230.50.86
                Quarantined: no
                Online status: online
                Registration status: registered
                On-net status: off-net
                Gateway Interface: port1
                FortiClient version: 7.4.5
                AVDB version: 1.0
                FortiClient app signature version: 35.189
                FortiClient vulnerability scan engine version: 3.15
                FortiClient UID: 9A016B5A6E914B42AD4168C066EB04CA
                Host Name: WIN10-01
                OS Type: WIN64
                OS Version: Microsoft Windows 10 Professional Edition, 64-bit (build 19045)
                Domain: fortiad.info
                Last Login User: tsmith
                Owner: @fortiad.info
                Host Model: Standard PC (i440FX + PIIX, 1996)
                Host Manufacturer: QEMU
                CPU Model: Intel Xeon Processor (SapphireRapids)
                Memory Size: 8095
                AV Feature: 1
                FW Feature: 1
```

```
                WF Feature: 1
                AS Feature: 0
                VS Feature: 1
                VN Feature: 1
                Last vul message received time: Tue Mar 24 12:10:47 2026
                Last vul scanned time: Sun Mar 22 07:00:03 2026
                Last vul statistic: critical=0, high=0, medium=0, low=3, info=0
                Avatar fingerprint:
                Avatar source username: Tom Smith
                Avatar source email:
                Avatar source: OS
                Phone number:
                Issued time in seconds: 1774379409
                JWT expiry time in seconds: 0 (Current time in seconds: 1774380150)
                Subscribed only: false
                Sendall: disabled
                Number of Routes: (1)
                        Gateway Route #0:
                                - IP:10.0.1.2, MAC:02:09:0f:00:01:02, VPN: no
                                - Interface:port1, VDOM:root (0), SN: FGVM02TM25011014
                                - Gateway IP:0.0.0.0, MAC:02:09:0f:00:03:01
```

```
# diagnose wad dev query-by uid 9A016B5A6E914B42AD4168C066EB04CA FCTEMS8825007166
00000000000000000000000000000000
Attr of type=0, length=83, value(ascii)=9A016B5A6E914B42AD4168C066EB04CA
Attr of type=7, length=8, value(ascii)=i
Attr of type=4, length=13, value(ascii)=@fortiad.info
Attr of type=6, length=1, value(ascii)=true
Attr of type=9, length=1, value(ascii)=1
Attr of type=5, length=40, value(ascii)=6FEF191AE3132C065C2888C81B413B64D7AC17FB
Attr of type=3, length=76, value(ascii)=ZTNA_all_registered_clients_
FCTEMS8825007166000000000000000000000000000000000000
Attr of type=3, length=66, value(ascii)=ZTNA_Domain-Users_
FCTEMS882500716600000000000000000000000000000000
Attr of type=3, length=59, value(ascii)=CLASS_High_
FCTEMS88250071660000000000000000000000000000000
Response termination due to no more data
```

```
# diagnose firewall dynamic list
 List all dynamic addresses:
 ...
MAC dynamic addresses in VDOM root(vfid: 0):
 ...
CMDB name: MAC_EMS1_ZTNA_Domain-Users
TAG name: Domain-Users
MAC_EMS1_ZTNA_Domain-Users: ID(153)
        MAC(02:09:0f:00:01:02)
Total MAC dynamic addresses: 1.
 ...
CMDB name: MAC_EMS1_CLASS_High
TAG name: High
MAC_EMS1_CLASS_High: ID(52)
```

```
        MAC(02:09:0f:00:01:02)
Total MAC dynamic addresses: 1.
 ...
CMDB name: MAC_EMS1_ZTNA_Domain-Users
TAG name: Domain-Users
MAC_EMS1_ZTNA_Domain-Users: ID(153)
        MAC(02:09:0f:00:01:02)
Total MAC dynamic addresses: 1.
```

```
# diagnose test application fcnacd 7
#ID:0
UID:     9A016B5A6E914B42AD4168C066EB04CA
EMS:     FCTEMS8825007166 00000000000000000000000000000000
State:   sysinfo:1, tag:1, tagsz:1, out-of-sync:0, ref:1
Owner:   @fortiad.info
Cert SN: 6FEF191AE3132C065C2888C81B413B64D7AC17FB
online:  Yes
Sub TS:  1774380310
Route IP:10.0.1.2, vdom:root(0)
idx:0, ttdl:1   name:all_registered_clients
idx:2, ttdl:1   name:Domain-Users
idx:3, ttdl:2   name:High
```

```
# execute log filter field srcip 10.0.1.2
# execute log display
1: date=2026-03-24 time=12:15:29 eventtime=1774379729258821761 tz="-0700" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="root" srcip=10.0.1.2 srcport=8290
srcintf="port1" srcintfrole="undefined" dstip=10.88.0.3 dstport=9043 dstintf="port2"
dstintfrole="dmz" srcuuid="b458a65a-f759-51ea-d7df-ef2e750026d1" dstuuid="592dfb72-0775-51ec-aa79-
94bd9894388c" emstag="MAC_EMS1_ZTNA_Domain-Users" emstag2="MAC_EMS1_CLASS_High"
srccountry="Reserved" dstcountry="Reserved" sessionid=41193 proto=6 action="close" policyid=19
policytype="policy" poluuid="6006181c-27b4-51f1-02e2-dbc9951e19dd" policyname="allow-internal-
access" service="tcp/9043" trandisp="snat" transip=10.88.0.254 transport=8290 appcat="unscanned"
duration=3 sentbyte=4095 rcvdbyte=311977 sentpkt=30 rcvdpkt=225
fctuid="9A016B5A6E914B42AD4168C066EB04CA" unauthuser="tsmith" unauthusersource="forticlient"
srcdomain="fortiad.info" srcremote=148.230.50.86
```

Access denied:

```
# diagnose wad dev query-by uid 9A016B5A6E914B42AD4168C066EB04CA FCTEMS8825007166
00000000000000000000000000000000
Attr of type=0, length=83, value(ascii)=9A016B5A6E914B42AD4168C066EB04CA
Attr of type=7, length=8, value(ascii)=i
Attr of type=4, length=13, value(ascii)=@fortiad.info
Attr of type=6, length=1, value(ascii)=true
Attr of type=9, length=1, value(ascii)=1
Attr of type=5, length=40, value(ascii)=6FEF191AE3132C065C2888C81B413B64D7AC17FB
Attr of type=3, length=76, value(ascii)=ZTNA_all_registered_clients_
FCTEMS8825007166000000000000000000000000000000000
Attr of type=3, length=76, value(ascii)=ZTNA_Critical-Vulnerability_
FCTEMS8825007166000000000000000000000000000000000
```

```
Attr of type=3, length=66, value(ascii)=ZTNA_Domain-Users_
FCTEMS8825007166000000000000000000000000000000000
Attr of type=3, length=59, value(ascii)=CLASS_High_
FCTEMS8825007166000000000000000000000000000000000
Attr of type=3, length=77, value(ascii)=ZTNA_Malicious-File-Detected_
FCTEMS8825007166000000000000000000000000000000000
Response termination due to no more data
```

```
# diagnose firewall dynamic list
  List all dynamic addresses:
  ...
 CMDB name: EMS1_ZTNA_Malicious-File-Detected
TAG name: Malicious-File-Detected
EMS1_ZTNA_Malicious-File-Detected: ID(205)
        ADDR(10.0.1.2)
Total IP dynamic ipblocks: 1.
Total IP dynamic range blocks: 0.
Total IP dynamic addresses: 1.
```

```
# diagnose test application fcnacd 7
 #ID:0
UID:     9A016B5A6E914B42AD4168C066EB04CA
EMS:     FCTEMS8825007166 00000000000000000000000000000000
State:   sysinfo:1, tag:1, tagsz:1, out-of-sync:0, ref:1
Owner:   @fortiad.info
Cert SN: 6FEF191AE3132C065C2888C81B413B64D7AC17FB
online:  Yes
Sub TS:  1774380310
Route IP:10.0.1.2, vdom:root(0)
idx:0, ttdl:1   name:all_registered_clients
idx:2, ttdl:1   name:Domain-Users
idx:3, ttdl:2   name:High
idx:4, ttdl:1   name:Malicious-File-Detected
```

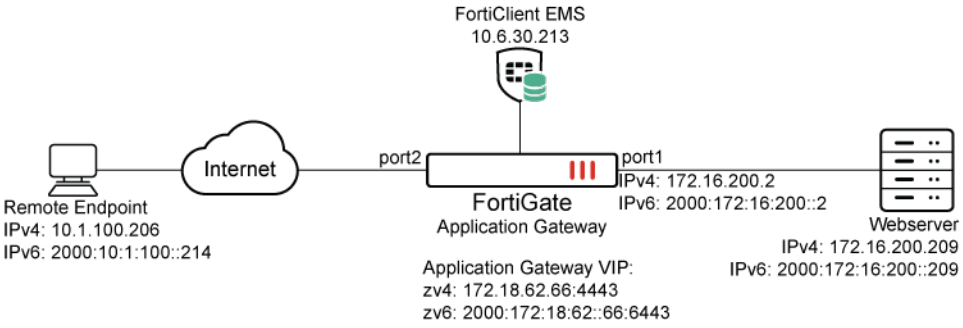
```
# execute log filter field srcip 10.0.1.2
# execute log filter field policyid 17
# execute log display

2: date=2026-03-25 time=06:10:27 eventtime=1774444227233285909 tz="-0700" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="root" srcip=10.0.1.2 srcport=10313
srcintf="port1" srcintfrole="undefined" dstip=10.88.0.3 dstport=9043 dstintf="port2"
dstintfrole="dmz" srcuuid="b458a65a-f759-51ea-d7df-ef2e750026d1" dstuuid="592dfb72-0775-51ec-aa79-
94bd9894388c" emstag="EMS1_ZTNA_Malicious-File-Detected" srccountry="Reserved"
dstcountry="Reserved" sessionid=68514 proto=6 action="deny" policyid=17 policytype="policy"
poluuid="551cbf70-27a3-51f1-619d-0258bb4090b8" policyname="block-internal-malicious-access"
service="tcp/9043" trandisp="noop" appcat="unscanned" duration=0 sentbyte=0 rcvdbyte=0 sentpkt=0
rcvdpkt=0 fctuid="9A016B5A6E914B42AD4168C066EB04CA" unauthuser="tsmith"
unauthusersource="forticlient" srcdomain="fortiad.info" srcremote=148.230.50.86 crscore=30
craction=131072 crlevel="high"
```

ZTNA IPv6 examples

IPv6 can be configured in ZTNA in several scenarios:

- IPv6 Client — IPv6 Access Proxy — IPv6 Server
- IPv6 Client — IPv6 Access Proxy — IPv4 Server
- IPv4 Client — IPv4 Access Proxy — IPv6 Server
- IPv6 EMS tags in dual stack firewall policies

These examples show the basic configuration for each scenario. It is assumed that the EMS fabric connector is already successfully connected.



Example 1: IPv6 Client — IPv6 Access Proxy — IPv6 Server

To configure the FortiGate:

1. Configure the IPv6 access proxy VIP:

```
config firewall vip6
    edit "zv6"
        set type access-proxy
        set extip 2000:172:18:62::66
        set server-type https
        set extport 6443
        set ssl-certificate "cert"
    next
end
```

2. Configure a virtual host:

```
config firewall access-proxy-virtual-host
    edit "vhost_ipv6"
        set ssl-certificate "cert"
        set host "webserver.test.com"
    next
end
```

The client uses this address to connect to the access proxy.

3. Configure an IPv6 web proxy and IPv6 api-gateway, apply the VIP6 and virtual host to it, and assign an IPv6 address to the realserver:

```
config ztna web-proxy
    edit "zs6"
        set vip "zv6"
        set host "vhost_ipv6"
        config api-gateway6
            edit 1
                config realservers
                    edit 1
                        set ip 2000:172:16:200::209
                    next
                end
            next
        end
    next
end
```

4. Configure a ZTNA policy:

```
config firewall policy
    edit 1
        set name "ztna_rule"
        set srcintf "port2"
        set dstintf "any"
        set action accept
        set srcaddr6 "all"
        set dstaddr6 "zs6"
        set schedule "always"
        set logtraffic all
    next
end
```

To test the configuration:

1. On an IPv6 client, ensure that the address webserver.test.com resolves to the IPv6 VIP address of 2000:172:18:62::66.
2. In a browser, connect to https://webserver.test.com:6443.
3. After device certificate verification, the browser will open up the webpage on the IPv6 real server.
4. In the Forward Traffic Log, the following log is available:

```
1: date=2026-03-03 time=14:46:14 eventtime=1772577974130406404 tz="-0800" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=2000:10:1:100::214 srcport=55957
srcintf="port2" srcintfrole="undefined" dstcountry="Reserved" srccountry="Reserved"
dstip=2000:172:16:200::209 dstport=443 dstintf="port1" dstintfrole="undefined" sessionid=9174
srcuuid="b458a65a-f759-51ea-d7df-ef2e750026d1" dstuuid="01b612e6-174a-51f1-1a3d-b3293e8efa12"
service="HTTPS" proxyapptype="ztna-proxy" proto=6 action="accept" policyid=1
policytype="policy" poluuid="693c3e3e-174c-51f1-3e13-715130a1a107" policyname="ztna_rule"
appcat="unscanned" duration=129 gatewayid=1 realserverid=1 vip="zv6"
vipincomingip=2000:172:18:62::66 accessproxy="zs6"
```

```
clientdeviceid="9A016B5A6E914B42AD4168C066EB04CA" clientdevicemanageable="manageable"
clientdeviceems="FCTEMS8825007166" clientdevicetags="CLASS_Low/ all_registered_clients"
clientcert="yes" emsconnection="online" wanin=303009 rcvdbyte=303009 wanout=4424 lanin=4220
sentbyte=4220 lanout=305716 fctuid="9A016B5A6E914B42AD4168C066EB04CA"
```

Example 2: IPv6 Client — IPv6 Access Proxy — IPv4 Server

To configure the FortiGate:

1. Configure the IPv6 access proxy VIP:

```
config firewall vip6
    edit "zv6"
        set type access-proxy
        set extip 2000:172:18:62::66
        set server-type https
        set extport 6443
        set ssl-certificate "cert"
    next
end
```

2. Configure a virtual host:

```
config firewall access-proxy-virtual-host
    edit "vhost_ipv6"
        set ssl-certificate "cert"
        set host "webserver.test.com"
    next
end
```

   The client uses this address to connect to the access proxy.

3. Configure an IPv6 web proxy and api-gateway, apply the VIP6 and virtual host to it, and assign an IPv4 address to the realserver:

```
config ztna web-proxy
    edit "zs6"
        set vip "zv6"
        set host "vhost_ipv6"
        config api-gateway
            edit 1
                config realservers
                    edit 1
                        set ip 172.16.200.209
                    next
                end
            next
        end
    next
end
```

4. Configure a ZTNA policy:

```
config firewall policy
    edit 1
        set name "ztna_rule"
        set srcintf "port2"
        set dstintf "any"
        set action accept
        set srcaddr6 "all"
        set dstaddr6 "zs6"
        set schedule "always"
        set logtraffic all
    next
end
```

To test the configuration:

1. On an IPv6 client, ensure that the address webserver.test.com resolves to the IPv6 VIP address of 2000:172:18:62::66.
2. In a browser, connect to https://webserver.test.com:6443.
3. After device certificate verification, the browser will open up the webpage on the IPv4 real server.
4. In the Forward Traffic Log, the following log is available:

```
1: date=2026-03-03 time=14:46:14 eventtime=1772577974130406404 tz="-0800" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=2000:10:1:100::214 srcport=55957
srcintf="port2" srcintfrole="undefined" dstcountry="Reserved" srccountry="Reserved"
dstip=172.16.200.209 dstport=443 dstintf="port1" dstintfrole="undefined" sessionid=9174
srcuuid="b458a65a-f759-51ea-d7df-ef2e750026d1" dstuuid="01b612e6-174a-51f1-1a3d-b3293e8efa12"
service="HTTPS" proxyapptype="ztna-proxy" proto=6 action="accept" policyid=1
policytype="policy" poluuid="693c3e3e-174c-51f1-3e13-715130a1a107" policyname="ztna_rule"
appcat="unscanned" duration=129 gatewayid=1 realserverid=1 vip="zv6"
vipincomingip=2000:172:18:62::66 accessproxy="zs6"
clientdeviceid="9A016B5A6E914B42AD4168C066EB04CA" clientdevicemanageable="manageable"
clientdeviceems="FCTEMS8825007166" clientdevicetags="CLASS_Low/ all_registered_clients"
clientcert="yes" emsconnection="online" wanin=303009 rcvdbyte=303009 wanout=4424 lanin=4220
sentbyte=4220 lanout=305716 fctuid="9A016B5A6E914B42AD4168C066EB04CA"
```

Example 3: IPv4 Client — IPv4 Access Proxy — IPv6 Server

To configure the FortiGate:

1. Configure the IPv4 access proxy VIP:

```
config firewall vip
    edit "zv4"
        set type access-proxy
        set extip 172.18.62.66
        set extintf "port2"
        set server-type https
        set extport 4443
        set ssl-certificate "cert"
```

```
    next
end
```

2. Configure a virtual host:

```
config firewall access-proxy-virtual-host
    edit "vhost_ipv4"
        set ssl-certificate "cert"
        set host "qa.test.com"
    next
end
```

The client uses this address to connect to the access proxy.

3. Configure an IPv4 web proxy and IPv6 api-gateway, apply the VIP and virtual host to it, and assign an IPv6 address to the realserver:

```
config ztna web-proxy
    edit "zs4"
        set vip "zv4"
        set host "vhost_ipv4"
        config api-gateway6
            edit 1
                config realservers
                    edit 1
                        set ip 2000:172:16:200::209
                    next
                end
            next
        end
    next
end
```

4. Configure a ZTNA policy:

```
config firewall policy
    edit 1
        set name "ztna_rule"
        set srcintf "port2"
        set dstintf "any"
        set action accept
        set srcaddr "all"
        set dstaddr "zs4"
        set schedule "always"
        set logtraffic all
    next
end
```

To test the configuration:

1. On an IPv4 client, ensure that the address webserver.test.com resolves to the IPv4 VIP address of 172.18.62.66.
2. In a browser, connect to https://webserver.test.com:6443.

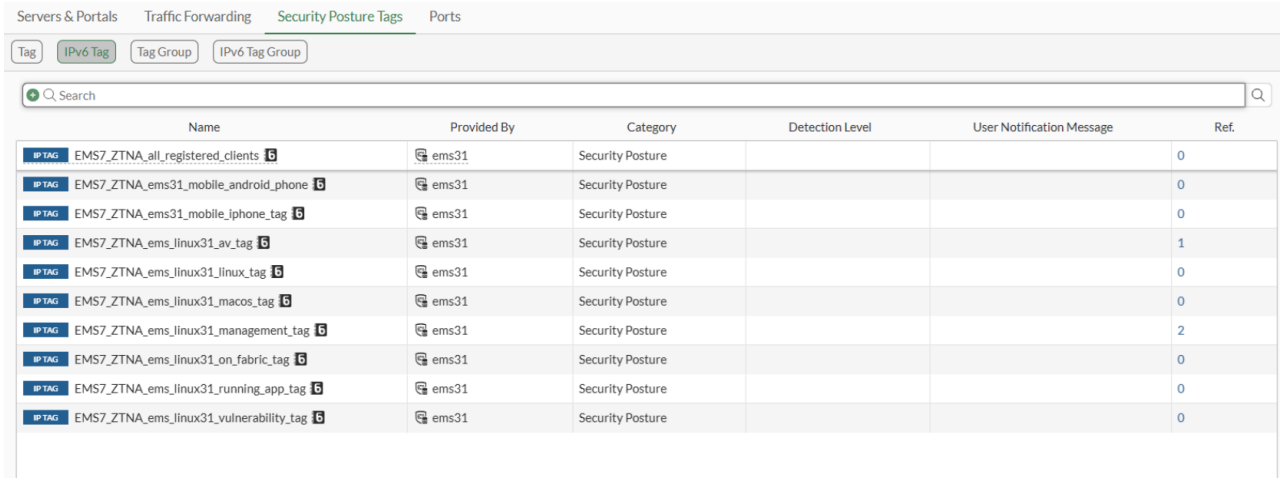
3. After device certificate verification, the browser will open up the webpage on the IPv6 real server.
4. In the Forward Traffic Log, the following log is available:

```
1: date=2026-03-03 time=14:46:14 eventtime=1772577974130406404 tz="-0800" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.1.100.206 srcport=55957
srcintf="port2" srcintfrole="undefined" dstcountry="Reserved" srccountry="Reserved"
dstip=2000:172:16:200::209 dstport=443 dstintf="port1" dstintfrole="undefined" sessionid=9174
srcuuid="b458a65a-f759-51ea-d7df-ef2e750026d1" dstuuid="01b612e6-174a-51f1-1a3d-b3293e8efa12"
service="HTTPS" proxyapptype="ztna-proxy" proto=6 action="accept" policyid=1
policytype="policy" poluuid="693c3e3e-174c-51f1-3e13-715130a1a107" policyname="ztna_rule"
appcat="unscanned" duration=129 gatewayid=1 realserverid=1 vip="zv4"
vipincomingip=172.18.62.66 accessproxy="zs4" clientdeviceid="9A016B5A6E914B42AD4168C066EB04CA"
clientdevicemanageable="manageable" clientdeviceems="FCTEMS8825007166"
clientdevicetags="CLASS_Low/ all_registered_clients" clientcert="yes" emsconnection="online"
wanin=303009 rcvdbyte=303009 wanout=4424 lanin=4220 sentbyte=4220 lanout=305716
fctuid="9A016B5A6E914B42AD4168C066EB04CA"
```

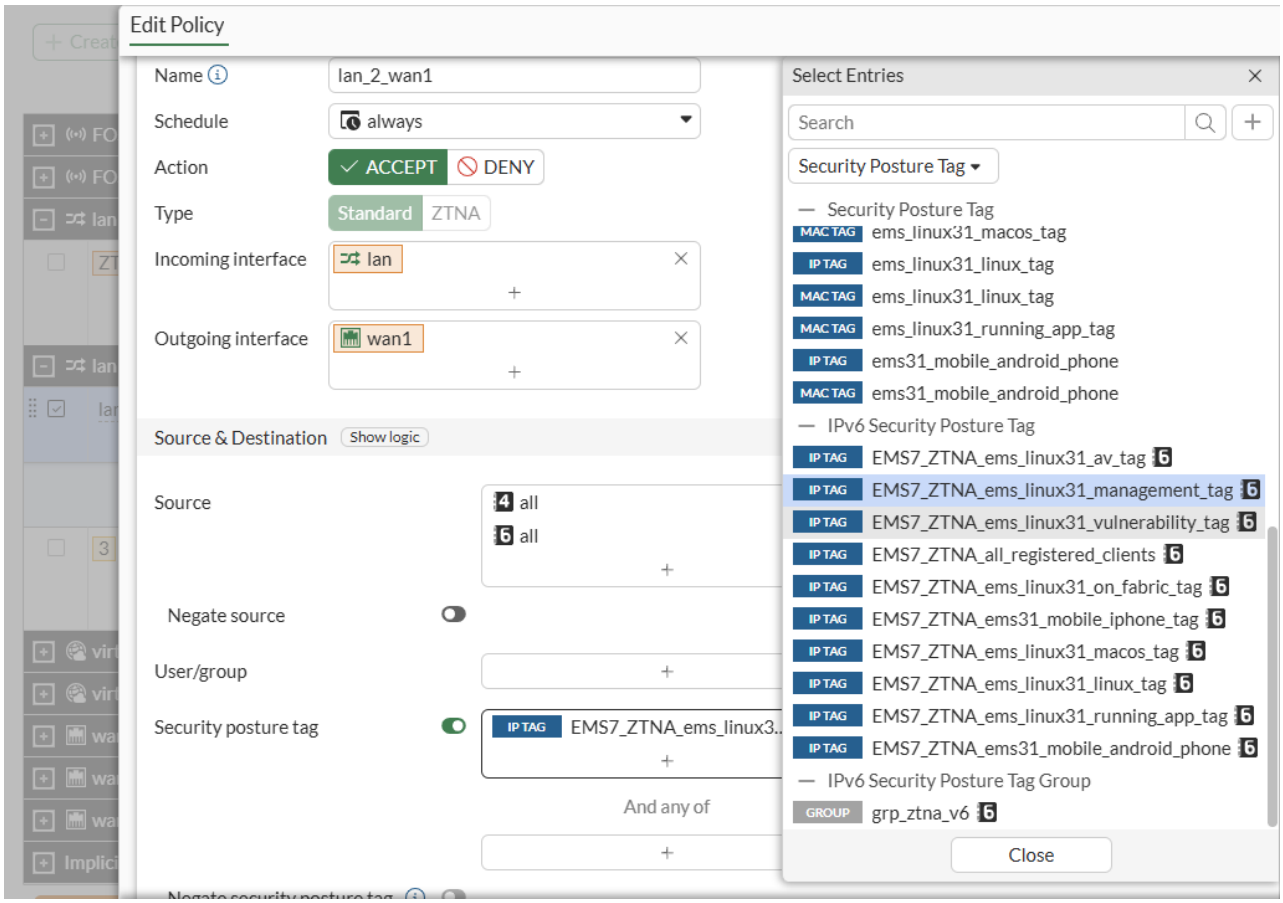
Example 4: IPv6 EMS tags in dual stack firewall policies

To configure the FortiGate:

1. Go to *Policy & Objects* > *ZTNA* > *Security Posture Tags* > *IPv6 Tag* to verify that FortiGate has IPv6 tags from FortiClient EMS:



2. Go to *Policy & Objects* > *Firewall Policy*, create or edit a firewall policy with IPv4 and IPv6 source addresses, and add IPv6 EMS security posture tag:



To configure the FortiGate in the CLI:

1. Verify the EMS tag in the CLI:

```
# show firewall address6
config firewall address6
    ...
    edit "EMS7_ZTNA_ems_linux31_management_tag"
        set uuid 880ef31c-fcba-51f0-e348-1f01acd82c22
        set type dynamic
        set sub-type ems-tag
        set dirty clean
        set obj-tag "ems_linux31_management_tag"
        set tag-type "zero_trust"
    next
    ...
end
```

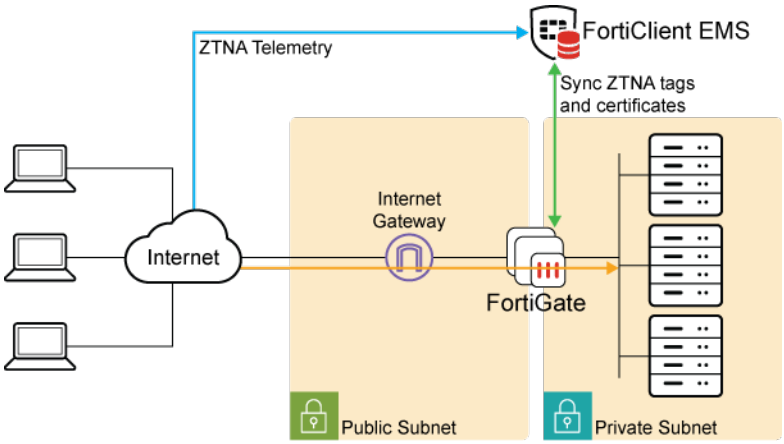
2. Apply the tag in a firewall policy:

```
config firewall policy
    edit 1
        set name "lan_2_wan1"
```

```
        set srcintf "lan"
        set dstintf "wan1"
        set action accept
        set ztna-status enable
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set ztna-ems-tag6 "EMS7_ZTNA_ems_linux31_management_tag"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set nat enable
    next
end
```

ZTNA Zero Trust application gateway example

The Zero Trust application gateway (ZTAG) for SaaS applications delivers Zero Trust Network Access (ZTNA) to companies that deploy SaaS applications and services in the cloud. A Zero Trust application gateway is deployed in the cloud, protecting resources on the private subnets by enforcing security and access control.



A Zero Trust application gateway can be deployed directly on AWS, Azure, and VMware. The deployment template can configure the necessary ZTNA settings to get the ZTNA configurations started.

For more information, see the Zero Trust Application Gateway Admin Guide.

ZTNA SaaS application access control with Inline CASB

The FortiGate ZTNA application gateway can be configured to act as an inline cloud access security broker (CASB) by providing access control to software-as-a-service (SaaS) traffic using ZTNA access control rules and an inline CASB profile. A CASB sits between users and their cloud service to enforce security policies as they access cloud-based resources.

The following components are required to use the ZTNA inline CASB feature:

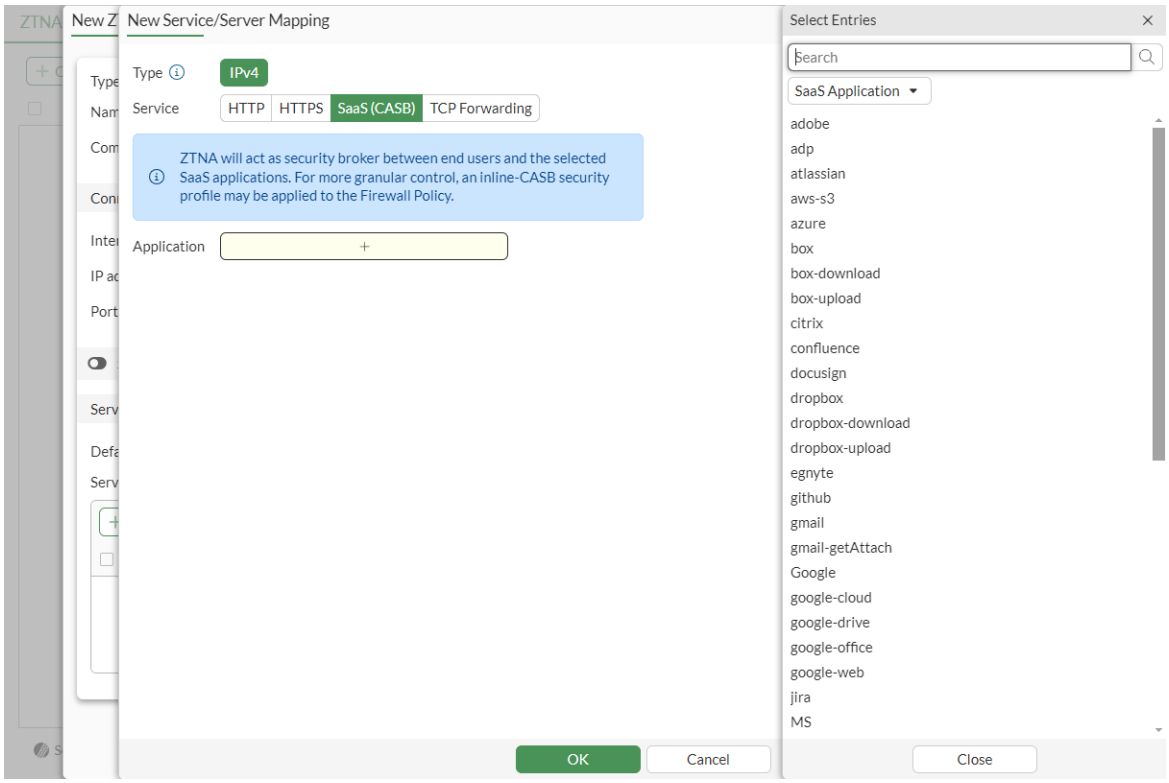
- The FortiGuard Inline CASB Database (ICDB) used by the FortiGate and FortiClient EMS.

  This database includes all FQDNs related to specific SaaS applications and corresponding FortiGuard packages for FortiOS and FortiClient.
- A FortiGate ZTNA TCP forwarding access proxy configuration that specifies SaaS application destinations using application names defined in the ICDB.
- ZTNA connection rules for SaaS traffic that are provisioned using FortiClient EMS (version 7.2.2 and later).
- FortiClient (version 7.2.0 and later) installed on the user's machine to receive the ZTNA connection rules for SaaS traffic from FortiClient EMS.

Syntax

Users can configure the ZTNA application gateway using the SaaS proxy access type and conveniently specify SaaS application destinations by application name or by application group name without needing to manually search for and enter FQDNs specific to each SaaS application. This can be configured in the GUI or in the CLI.

To configure a ZTNA service/server mapping to use SaaS in the GUI:



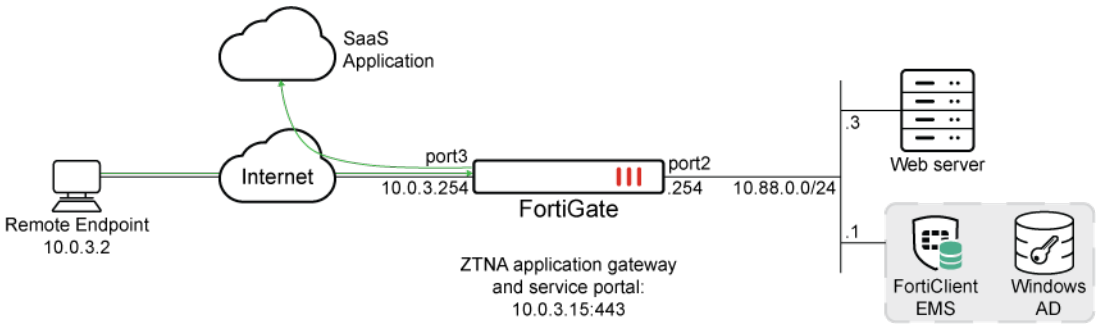
To configure a ZTNA application gateway to use SaaS in the CLI:

```
config firewall access-proxy
    edit <name>
        config api-gateway
            edit <ID>
                set url-map "/saas"
```

```
                set service saas
                set application <app 1> [app 2] ...
            next
        end
    next
end
```

Example

In this example, the FortiGate is configured as a ZTNA application gateway with a VIP of 10.0.3.15 and uses the SaaS access proxy type. SaaS application Gmail is allowed, but the action of uploading an attachment on Gmail is blocked.



To configure the ZTNA server object in the GUI:

1. Go to *Policy & Objects* > *ZTNA* > *Traffic Forwarding* > *Traffic Forwarding Server*.
2. Click *Create new*.
3. Set *Name* to *ZTNA-SaaS-Access*.
4. Set *Host* to *10.0.3.15*
5. Set *ZTNA port* to *Create new* and configure the following:

Option	Value
Interface	port3
IP Address	10.0.3.15
Port	443
Default Certificate	This example uses a wildcard certificate with a Common Name (CN): *.ztnademo.com. This certificate is trusted by the endpoints which will access the ZTNA port.
Name	SaaS:443

Verify that the IP address and port do not conflict with management access to the interface. Otherwise, change the IP address to another address on that subnet.

6. Click *OK*.

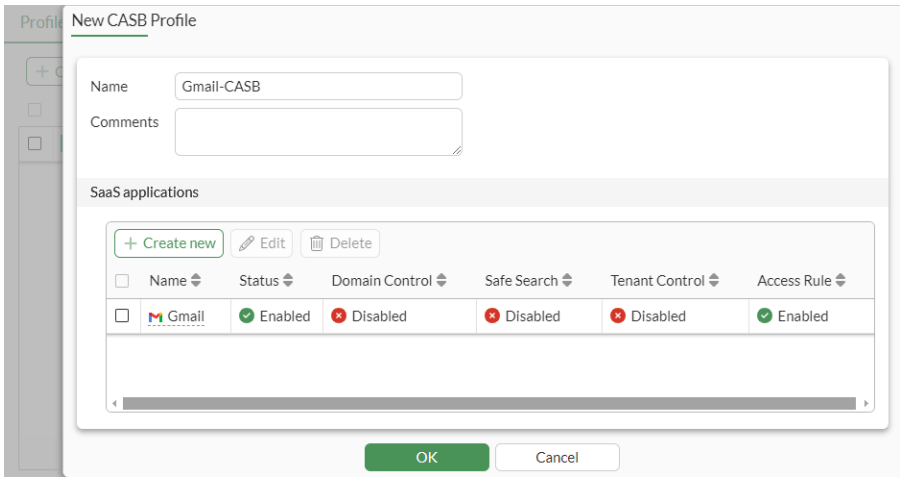
7. Go to *Policy & Objects* > *ZTNA* > *Traffic Forwarding* > *Destination*.
8. Click Create new and configure the following:

Option	Value
Name	SaaS
Type	SaaS
SaaS Application	gmail
Protocol	ALL

9. Click *OK*.

To configure the Inline CASB profile in the GUI:

1. Go to *System* > *Feature Visibility*.
2. Under *Security Features*, enable *Inline-CASB*.
3. Go to *Security Profiles* > *Inline-CASB* and click *Create new*.
4. Name the new profile *Gmail-CASB*.
5. In the *SaaS Applications* table, click *Create new*.
6. In *Select application*, select *Gmail*.
7. Click *Next*.
8. In *Privilege control*, right-click the entry and select *Block*.
9. Click *OK*.



10. Click *OK*.

To configure the ZTNA policy in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Click *Create new* and configure the following:

Option	Value
Name	ZTNA-SaaS-Access
Type	ZTNA
Incoming Interface	WAN (port3)
Source	All
User/group	<include, if needed>
Security posture tag	<include, if needed>
ZTNA port	SaaS:443
ZTNA destination	SaaS
Inline-CASB	Gmail-CASB
SSL inspection	custom-deep-inspection
Log allowed traffic	All sessions

3. Click *OK*.
4. Click *OK* to confirm the use of full SSL Inspection. Ensure that your clients trust the CA that is used in the SSL Inspection profile.

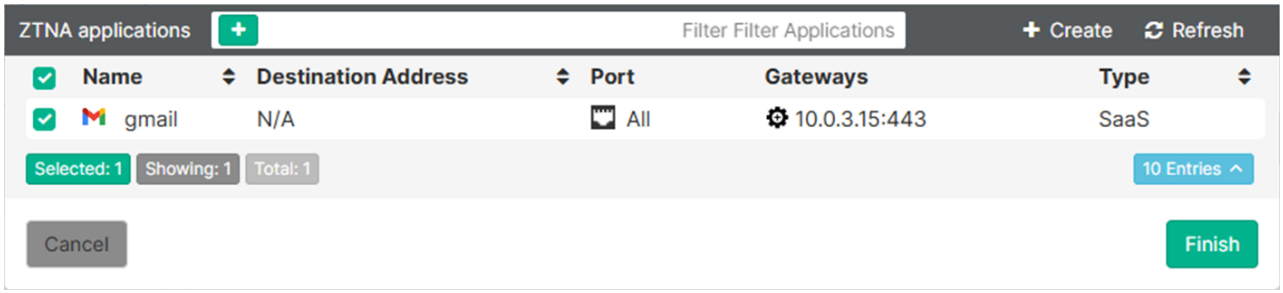
Optionally, if user authentication is configured, the ZTNA rule (`set users` or `set groups`), configure the authentication scheme and rule (see Applying user authentication in the ZTNA Deployment guide).

Before connecting, the users must have corresponding ZTNA Destinations in FortiClient. In FortiClient EMS, configure SaaS applications in *Endpoint Profiles > ZTNA Destinations* and push application destinations to FortiClient.

To configure the FortiClient EMS:

This step assumes the FortiGate and FortiClient EMS connector has been configured. See Fabric devices in the EMS Administration Guide and Configuring FortiClient EMS on page 4322.

1. On FortiClient EMS, go to *Endpoint Profiles > ZTNA Destinations*.
2. Edit the *Default* profile.
3. Next to *Name*, click *Advanced*.
4. Click the eye icon beside *ZTNA Destination Profile* to enable this profile to be viewed on the FortiClient.
5. Under *Rules*, click *Add* . The *ZTNA applications* dialog is displayed.
6. Select the gmail application as defined on the FortiGate.



7. Click *Finish*.

8. Click *Save*. The FortiClient endpoints will synchronize the destination from EMS.

Testing and results

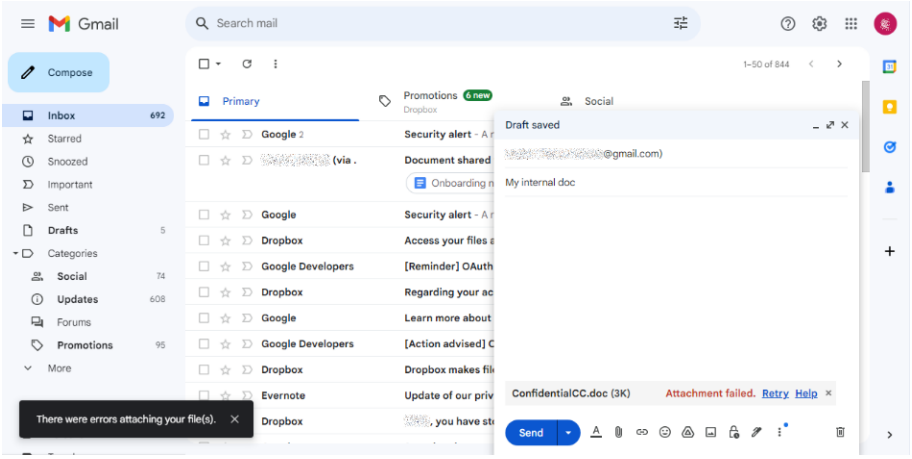
Once connected, the FortiClient retrieves the list of hosted ZTNA services, including the SaaS service, and adds corresponding ZTNA connection rules for the configured SaaS applications.

Connect to Gmail from a browser. The traffic is allowed.

After closing the session, the corresponding traffic logs can be viewed from the FortiGate GUI or from the CLI.

```
# execute log filter field subtype ztna
# exec log display
1: date=2026-03-31 time=09:06:21 eventtime=1774973182173477239 tz="-0700" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=55150
srcintf="port3" srcintfrole="wan" dstcountry="United States" srccountry="Reserved"
dstip=142.251.186.95 dstport=443 dstintf="port3" dstintfrole="wan" sessionid=1956
srcuuid="b458a65a-f759-51ea-d7df-ef2e750026d1" service="HTTPS" proxyapptype="ztna-proxy" proto=6
action="accept" policyid=17 policytype="policy" poluuid="59192848-2d11-51f1-b37c-286a7f89f5af"
policyname="ZTNA-SaaS-Access" appcat="unscanned" duration=1 vip="SaaS:443" vipincomingip=10.0.3.15
accessproxy="ZTNA-SaaS-Access" clientdevicemanageable="manageable" clientcert="yes" wanin=0
rcvdbyte=0 wanout=1509 lanin=3852 sentbyte=3852 lanout=3129
```

Connect to Gmail from a browser again. This time, compose an email and attach a file. This action will be blocked.



After closing the session, the corresponding *UTM > CASB* logs can be viewed from the FortiGate GUI or from the CLI.

```
# exec log filter reset
# exec log filter category utm-casb
# exec log display
1: date=2026-03-31 time=09:08:42 eventtime=1774973321505383780 tz="-0700" logid="2500010000"
type="utm" subtype="casb" eventtype="casb" level="warning" vd="root" policyid=17
poluuid="59192848-2d11-51f1-b37c-286a7f89f5af" policytype="policy" sessionid=2007 srcip=10.0.3.2
dstip=142.251.116.19 srcport=55181 dstport=443 srcintf="port3" srcintfrole="wan"
srcuuid="b458a65a-f759-51ea-d7df-ef2e750026d1" dstintf="port3" dstintfrole="wan" proto=6
url="https://mail.google.com/_/upload?authuser=0&dcp=asu-n" action="block" profile="Gmail-CASB"
saasapp="google-gmail" useractivity="google-gmail-upload-local-file" subaction="monitor"
```

```
tenantmatch="missed" activitycategory="activity-control" msg="CASB access was blocked because it
contained banned activity."
```

ZTNA application gateway with KDC to access shared drives

Fortinet technical support cannot troubleshoot third-party applications involved in this solution.

Kerberos Key Distribution Center (KDC) proxy protocol can be used to remotely authenticate domain users and issue Kerberos tickets. A remote user can use the same Kerberos ticket to authenticate when remotely accessing shared drives through ZTNA application gateway.

The following options are available when setting up ZTNA application gateway on TCP 445 for remote access to mapped drives:

- Create the mapped drive using the server's IP address, which uses Windows NT LAN Manager (NTLM) to authenticate users.
- Create the mapped drive using the server's FQDN (or DNS name), which requires Kerberos authentication.

The preferred way to create mapped drives is using the server's DNS name. KDC proxy protocol can be a reliable way to initiate Kerberos authentication for remote users to seamlessly access mapped drives and requires the following steps:

1. Set up the KDC proxy service on a server.
2. Set up the ZTNA application gateway for mapped drives.

The KDC Proxy only works for domain-joined workstations.

Setting up the KDC proxy service

The following steps are required to set up the KDC proxy service on a server:

1. Deploy a server that is joined to the domain and create a trusted certificate.
2. Configure the KDC proxy service on the server.
3. Configure clients to use the KDC proxy.
4. If client machines cannot retrieve group policy updates, configure registry keys on the clients.

Deploying a server and creating a certificate

Deploy a server that is joined to the domain. At a minimum, the server must have a public network interface with a domain name pointed to it. Clients will connect to the server over the internet. For networking, you can forward port 443 if necessary.

Create a trusted certificate for the public domain name of the proxy endpoint:

- Domain: fortitest.net
- KDC proxy: kdcproxy.fortitest.net

The KDC proxy service is present on the server, but not configured.

Configuring the KDC proxy service on the server

The KDC proxy service must be configured before it can be started.

To configure the server:

1. Start an elevated command prompt, and configure a URL ACL for the endpoint:

```
netsh http add urlacl url=https://+:443/KdcProxy user="NT authority\Network Service"
```

2. Associate the certificate created previously with the endpoint:

```
netsh http add sslcert ipport=0.0.0.0:443 certhash=<thumbprint of your certificate> appid=
{generated Guid}
```

This association instructs `HTTP.SYS` to use the certificate when connections occur over HTTPS.

For example: `netsh http add sslcert ipport=0.0.0.0:443 certhash=aaaabbbbccccddddeeeeffff0000111122223333 appid={aaaaaaaa-bbbb-cccc-dddd-aaaabbbbcccc}`

> For `appid` you should generate a random GUID here. From PowerShell try `[Guid]::NewGuid()`. The `certhash` is the thumbprint of your certificate.
>
> If the certificate contains extensions for CRL or AIA, those URLs should be reachable from the remote client.

3. If not using smart card certificates (or Windows Hello) for authentication, disable the certificate authentication requirement:

```
REG ADD "HKLM\SYSTEM\CurrentControlSet\Services\KPSSVC\Settings" /v HttpsClientAuth /t REG_
DWORD /d 0x0 /f
```

4. Additionally if not using certificates, enable password authentication:

```
REG ADD "HKLM\SYSTEM\CurrentControlSet\Services\KPSSVC\Settings" /v
DisallowUnprotectedPasswordAuth /t REG_DWORD /d 0x0 /f
```

5. Configure the KDC Proxy Service (KPSSVC) to start automatically:

```
sc config kpssvc start=auto
```

6. Start the service:

```
net start kpssvc
```

Configuring clients to use a KDC proxy

Clients must be configured to use a KDC proxy. This can be done through GPO or through modifying the registry directly. The GPO path is:

```
Computer Configuration\Policies\Administrative Templates\System\Kerberos\Specify KDC proxy servers
for Kerberos clients
```

Enabling this policy requires setting a realm-to-value mapping. The realm is your domain name, starting with a period (for example ".fortitest.net"), or a "*" to include all domains in a wildcard format. The value is specially crafted:

```
<https kdcproxy.fortitest.net:kdcproxy />
```

Deploy the policy, and reboot the client. Wait a while for caching to complete, and then log in. You should find users are now authenticating over the KDC proxy endpoint.

To verify from a logged in session, launch a command prompt and try the following command:

```
klist get krbtgt
```

Configuring registry keys on clients

If you are trying to deploy these settings on a client machine that cannot retrieve group policy updates, manually configure the registry keys for the client:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\Kerberos]
"KdcProxyServer_Enabled"=dword:00000001
```

```
[HKEY_LOCAL_
MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\Kerberos\KdcProxy\ProxyServers]
"*"="<https kdcproxy.fortitest.net />" or ".fortitest.net"="<https kdcproxy.fortitest.net />"
```

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System\Kerberos\Parameters]
"NoRevocationCheck"=dword:00000000
```

Setting up the ZTNA application gateway for mapped drives

The following steps are required to set up ZTNA application gateway for mapped drives on the FortiGate:

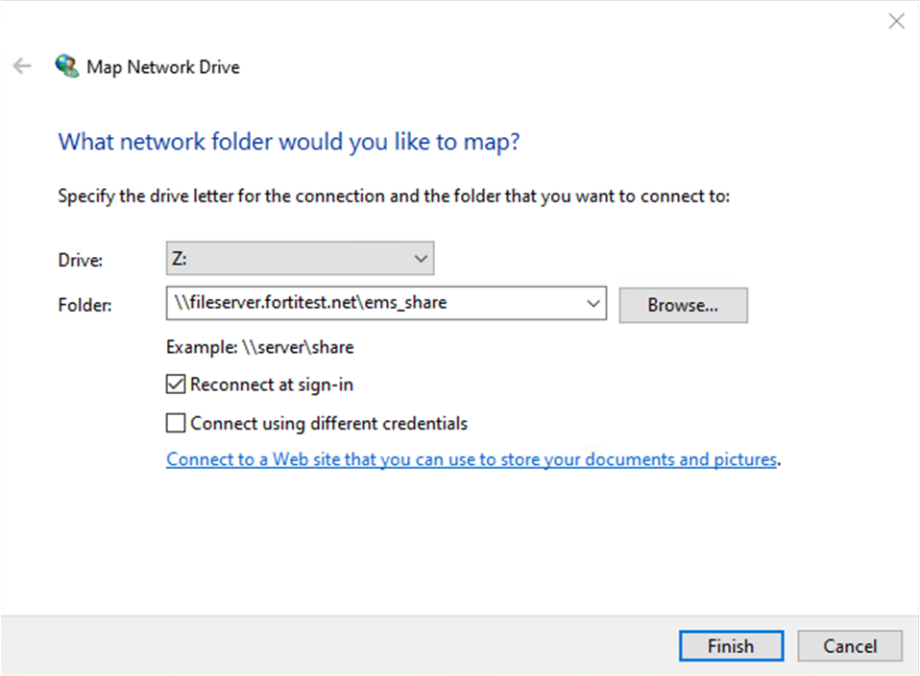
1. Configure FortiClient to have the ZTNA destination rule.
2. Configure a ZTNA server and proxy policy on FortiGate.

Configuring a ZTNA destination rule on FortiClient

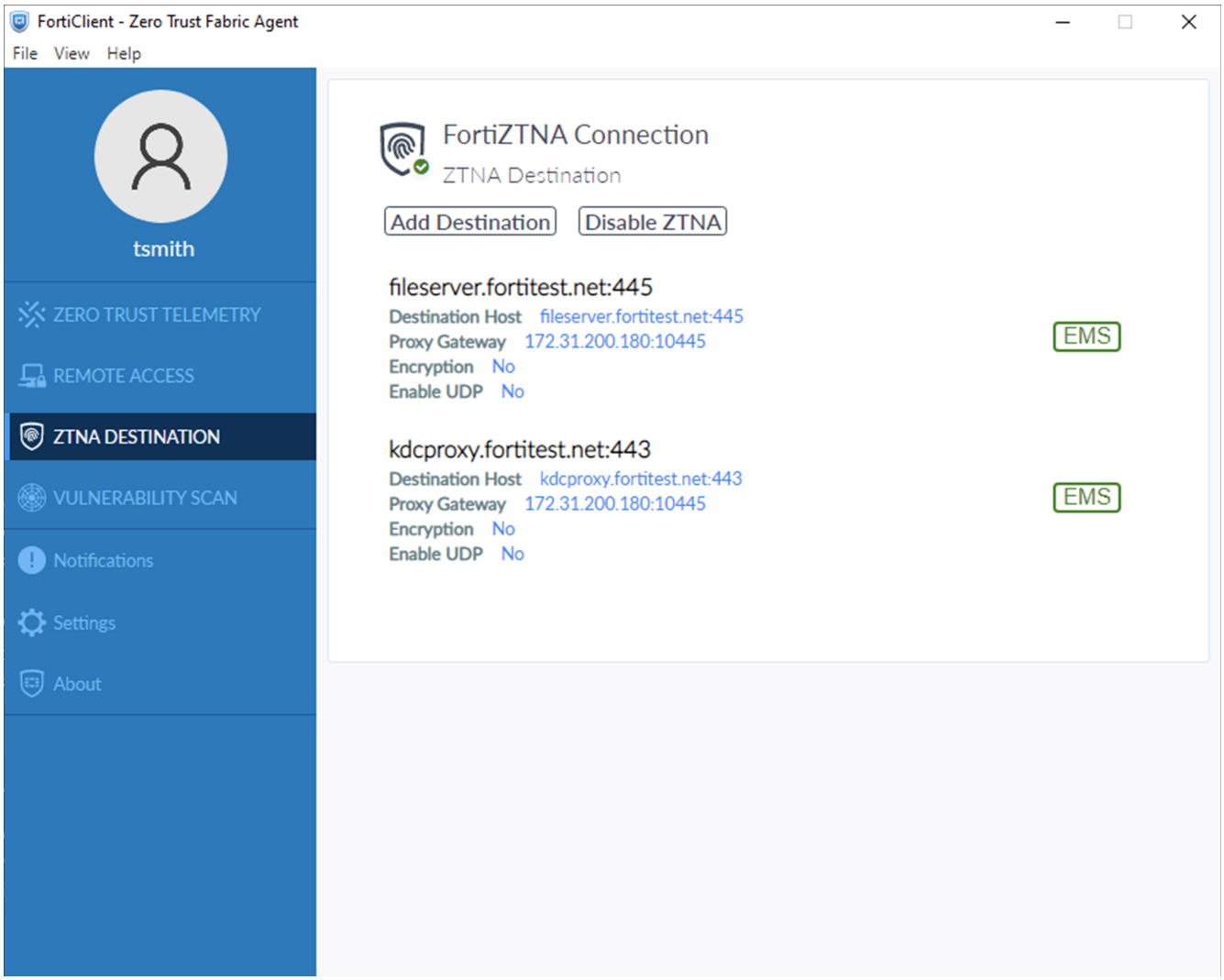
The ZTNA destination rule can be configured by using FortiClient EMS or by manually configuring FortiClient. The destination host matches the server's FQDN on port 445, and the proxy gateway should be the FortiGate WAN port on the chosen external port.

Another rule is required for the KDC proxy communication with the destination host defined as the FQDN of the KDC proxy server and port 443.

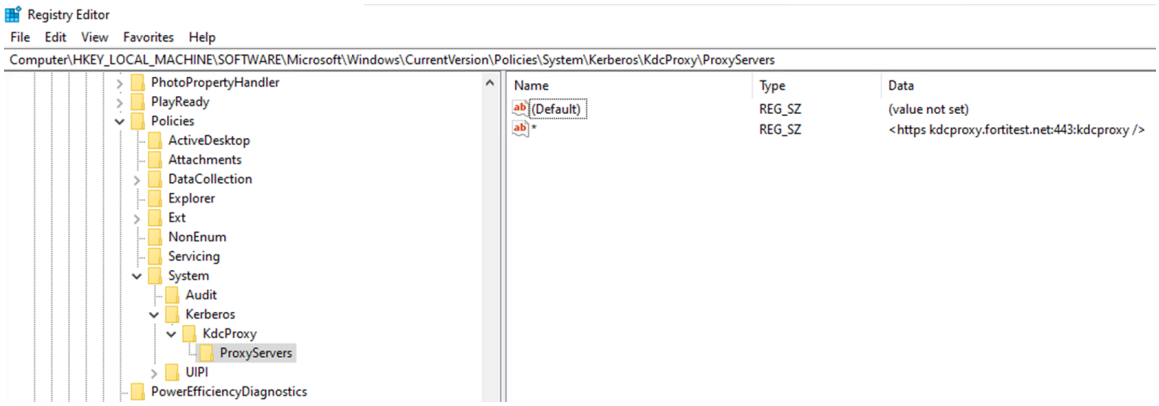
The map network drive configuration should be the last step after configuring the FortiGate. The following example shows the mapped drive:



The following example shows a ZTNA destination configured in FortiClient with the *Proxy Gateway*:



The following example shows the registry settings:



Configuring FortiGate

A ZTNA server and a proxy policy are required to verify and forward the incoming requests on port 10445 to the file server on port 445 as well as to the KDC proxy server on port 443.

To configure the ZTNA server:

1. Go to *Policy & Objects > ZTNA > Traffic Forwarding > Traffic Forwarding Server*.
2. Click *Create new*.
3. Set the *Name* to *Mapped Drive Using KDC*.
4. Set the *Host* to *172.31.200.180*.
5. In the ZTNA port table, click *Create New* and configure the following:

Option	Value
Interface	port3
IP address	172.31.200.180
Port	10445
Default certificate	Select the certificate which endpoints will be presented when accessing the ZTNA port.
Name	KDC:10445

6. Click *OK*.
7. Go to *Policy & Objects > ZTNA > Traffic Forwarding > Destination*.
8. Click *Create new* to define a new destination for the file server access:

Option	Value
Name	File-Server
Type	On-premise
Address	Use the dropdown and then the + button to define a new address object: • Name: fileserver • Type: FQDN • FQDN: fileserver.fortitest.net
Port	445
Protocol	TCP

9. Click *OK*.
10. Click *Create new* to define another new destination for the KDC proxy server:

Option	Value
Name	KDC_Proxy_Server
Type	On-premise

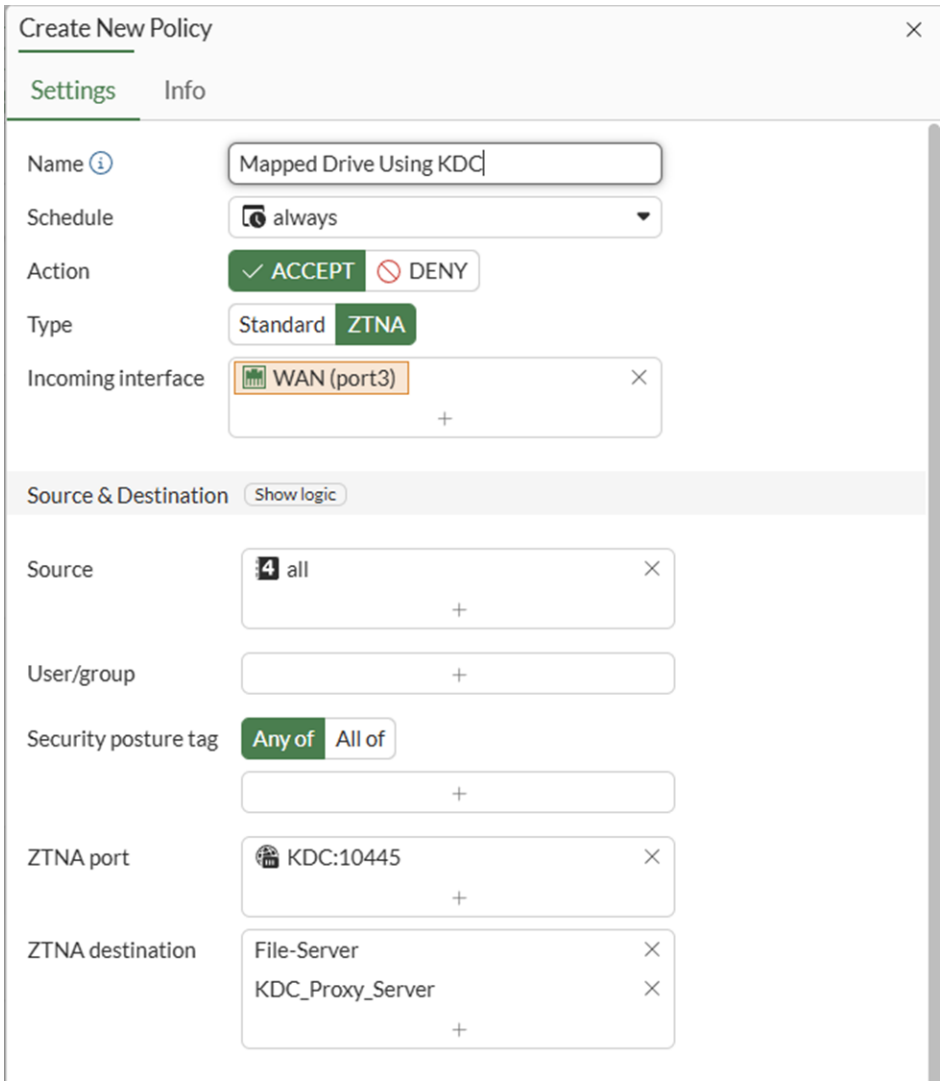
Option	Value
Address	Use the dropdown and then the + button to define a new address object: • Name: kdcproxy • Type: FQDN • FQDN: kdcproxy.fortitest.net
Port	443
Protocol	TCP

11. Click *OK*.

To configure the proxy policy:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Click *Create new*.
3. Configure the following:

Name	Mapped Drive Using KDC
Type	ZTNA
Incoming Interface	FGT-WAN (port1)
Source	all
Destination	FileServer KDC_Proxy_Server
ZTNA Port	KDC:10445
ZTNA Destination	File-Server KDC_Proxy_Server
Schedule	always
Action	Accept
Log Allowed Traffic	Enabled All Sessions
Enable this policy	Enabled



4. Click *OK*.

Custom replacement message for ZTNA virtual hosts

Each ZTNA virtual host can be configured to display messages from a custom replacement message group.

First create a replacement message group, and customize one ore more messages in the group. Then configure one or more ZTNA virtual hosts to use the replacement message group.

```
config firewall access-proxy-virtual-host
    edit <host name>
        set replacemsg-group <replacemsg group>
    next
end
```

When a client fails a ZTNA check with the virtual host, the replacement message is displayed.

Example

In this example, a ZTNA virtual host named *server1.ztna.local* is mapped to a replacement message group named *test-vhost*, and the group includes a customized *ZTNA Empty Certificate Error Page* message. The message is customized with a Company Y logo.

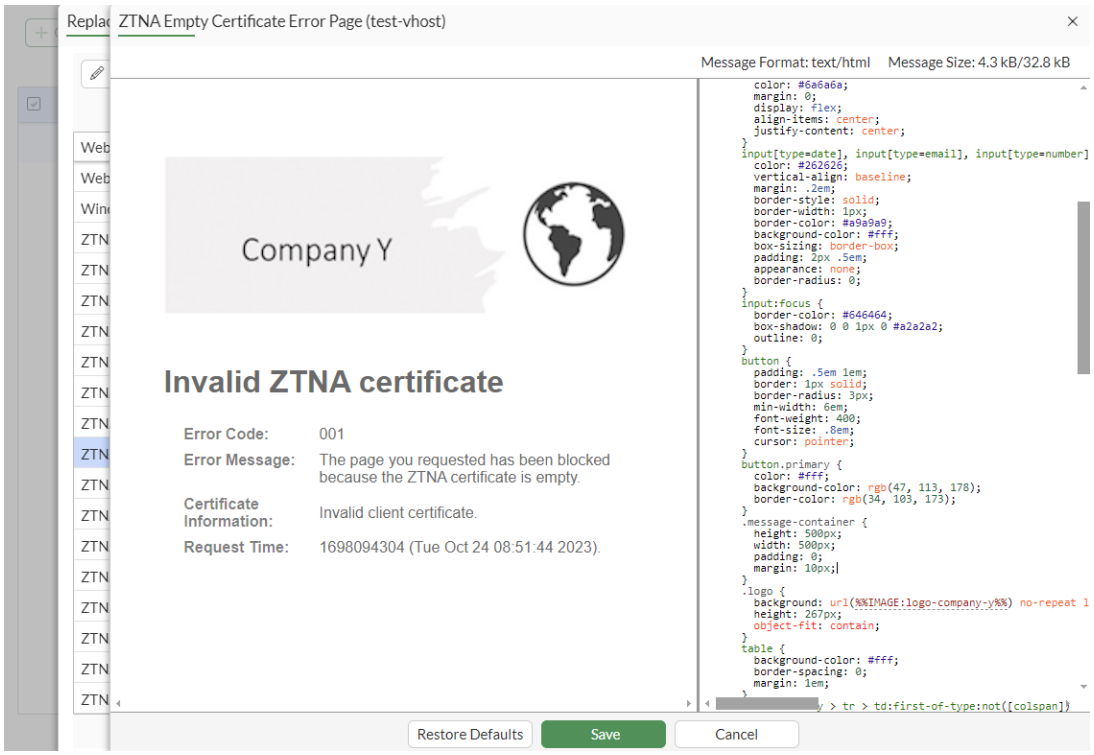
When clients fail a ZTNA check with the ZTNA virtual host (server1.ztna.local) because of an empty certificate, the custom replacement message is displayed.

Go to *System* > *Feature Visibility* and enable *Replacement Message Groups*. See Feature visibility on page 3952 for more information.

To customize replacement messages for ZTNA virtual hosts:

1. Upload a logo to the FortiGate to use in replacement messages:
   a. Go to *System* > *Replacement Messages* and click *Manage Images*.
   b. Click *Create New*.
   c. Name and upload an image file.
   d. Click *OK*. The logo is uploaded to the FortiGate.
2. Create a replacement message group named, for example, *test-vhost*:
   a. Go to *System* > *Replacement Message Groups* and click *Create New*.
   b. Specify a name for the group, such as *test-vhost*.
   c. Set *Group Type* to *Security*.
   d. Click *OK*.
3. Customize one or more messages in the *test-vhost* group:
   In this example, the *ZTNA Empty Certificate Error Page* message is edited to add a custom logo.
   a. Double-click the *test-vhost* replacement message group to open it for editing.
   b. Select the *ZTNA Empty Certificate Error Page* message and click *Edit*.
   c. In the right pane, edit the URL for the `.logo` section by typing the logo name to select the uploaded logo, for example, *logo-company-y*.

```
...
}
.logo {
  background: url(%%IMAGE:logo-company-y%%) no-repeat left center;
  height: 267px;
  object-fit: contain;
}
...
```



d. Click *Save*. A green checkmark is displayed in the *Modified* column to indicate a customized message.

4. Configure a ZTNA server with a ZTNA virtual host named *server1.ztna.local*. See Configure a ZTNA server on page 1597.

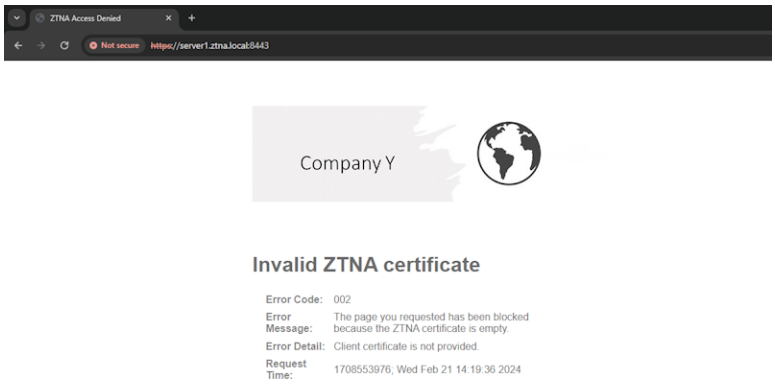
   In the *Service/server* mapping, be sure to set *Virtual Host* to *Specify*, and enter the name or IP address of the host that the request must match. For example, if server1.ztna.local is entered as the host, then only requests to server1.ztna.local will match.

5. Map the ZTNA virtual host to the replacement message group in the CLI.

   In this example, the ZTNA virtual host named `server1.ztna.local` is configured to use the `test-vhost` replacement message group.

```
config firewall access-proxy-virtual-host
    edit "server1.ztna.local"
        set replacemsg-group "test-vhost"
    next
end
```

6. Create a ZTNA policy to allow traffic to the ZTNA server. See Configure a ZTNA policy on page 1602.
7. When a client fails to access the ZTNA virtual host named *server1.ztna.local* because of an empty certificate error, the following custom replacement message with the Company Y logo is displayed.



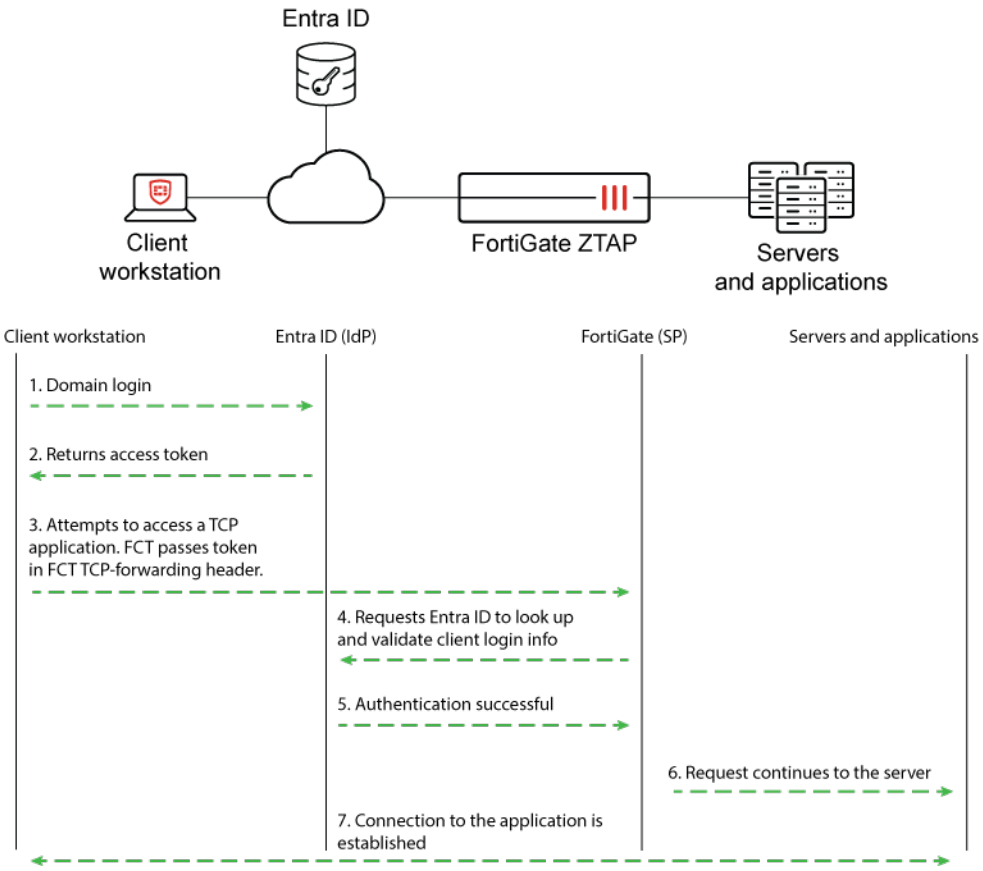
ZTNA single sign-on with Entra ID

In this enhancement, Windows users signed in to their workstations using Microsoft Entra ID domain are automatically allowed access to ZTNA-protected TCP resources by using the same IdP login information. FortiGate queries Entra ID using the client's login token to look up and validate the user. This allows single sign-on (SSO) and eliminates the extra step for each user to authenticate when they access a TCP application.

Prerequisites include:

- FortiClient 7.4.3 and later
- FortiOS 7.6.1 and later
- Domain configured on Microsoft Entra ID
- Enterprise application configured on Microsoft Entra ID
- Various permissions enabled in Microsoft Entra ID
- Windows clients that are able to join to the Entra ID domain

Topology and communications

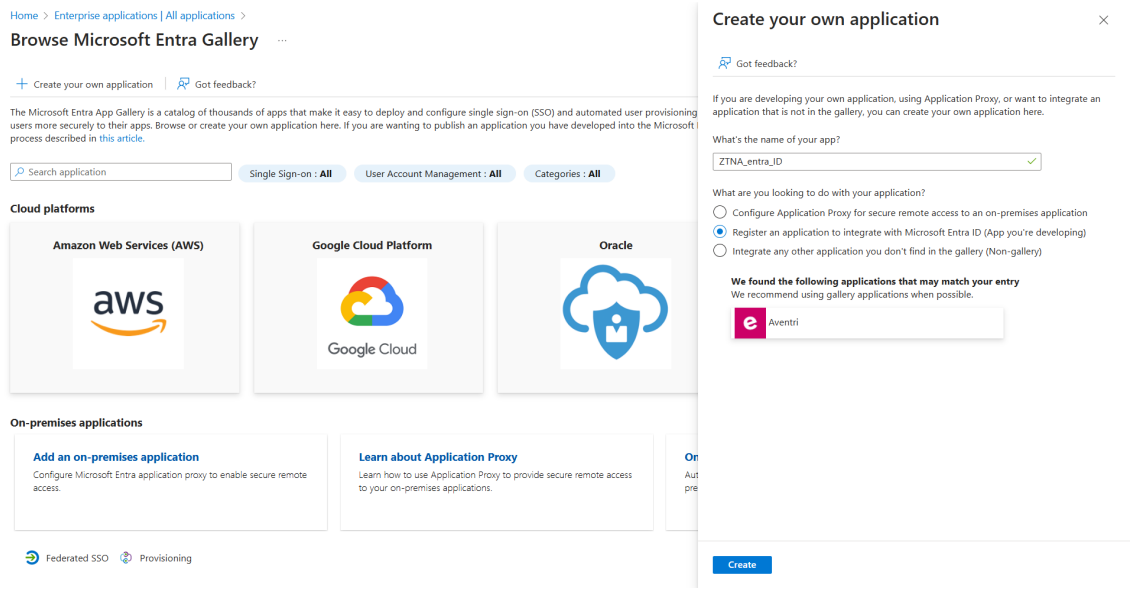


Entra ID configurations

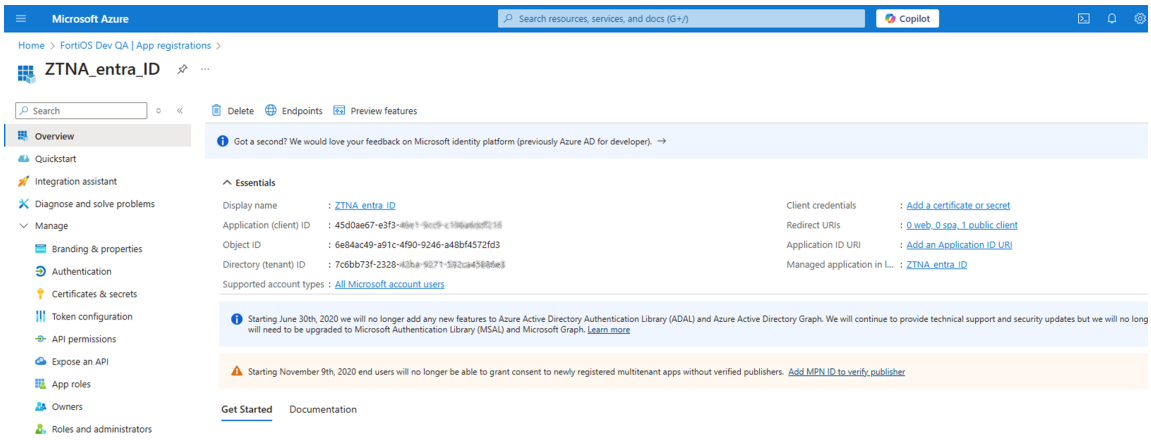
Create an Azure enterprise application to obtain Entra ID values.

To create an Azure enterprise application:

1. Click *Create your own application*, and select the *Register an application to integrate with Microsoft Entra ID (App you're developing)* option.



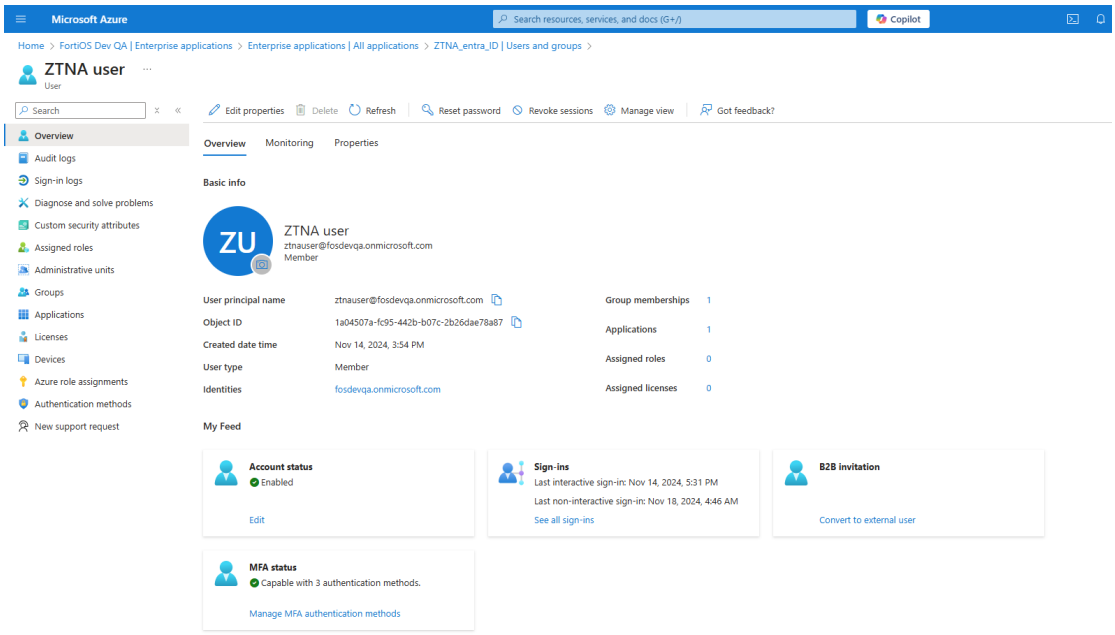
2. Open the *App registrations* page, and locate your app. On the *Overview* page, you can obtain the *Application (client) ID* and the *Directory (tenant) ID*.



In the this example, the values from the *ZTNA_entra_ID* app can be applied to the FortiClient EMS ZTNA Destinations profile in later steps:

Entra ID portal field	Corresponding FortiClient XML profile element	Value
Application (client) ID	<client_id>	45d0ae67-e3f3-xxxx-xxxx-xxxxxxxxxxxx
Directory (tenant) ID	<tenant_name>	7c6bb73f-2328-xxxx-xxxx-xxxxxxxxxxxx

Users and groups can be assigned to the enterprise application. In this example, the user ztnauser@fosdevqa.onmicrosoft.com belongs to a group assigned to the ZTNA_entra_ID app.



The user belongs to the Entra ID domain that is used by the user workstation to join the domain.

FortiClient EMS configurations

In FortiClient EMS 7.4.3 and later, the client ID and tenant ID can be specified in the `azure_app` attribute in XML.

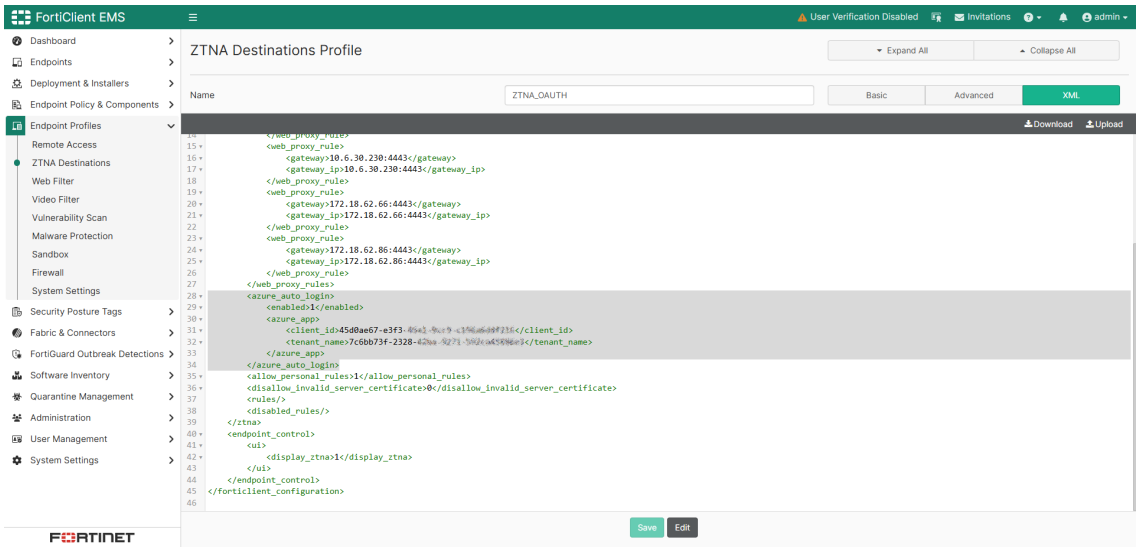
To configure FortiClient EMS:

1. Go to *Endpoint Profiles* > *ZTNA Destinations*.
2. Create a new profile or edit an existing one.
3. Click *XML*, and then click *Edit*.
4. Configure the profile as follows:

   Configure `<client_id>` and `<tenant_name>` with the Application (client) ID and Directory (tenant) ID values for the enterprise application in the Entra ID portal.

   Bold lines in the XML sample highlight key fields to support this feature:

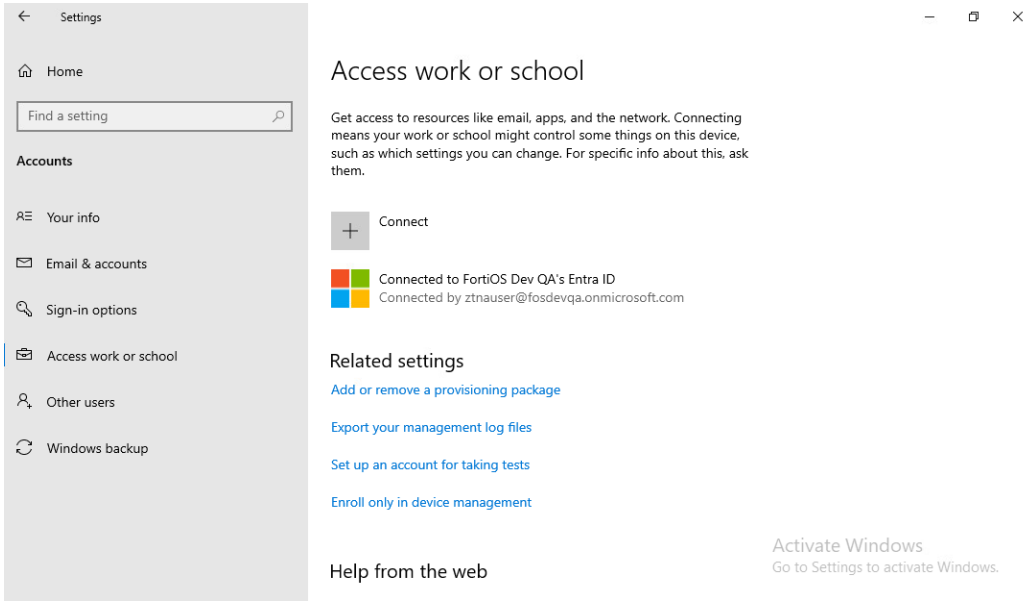
```
<forticlient_configuration>
    <ztna>
            …
        <azure_auto_login>
            <enabled>1</enabled>
            <azure_app>
                <client_id>45d0ae67-e3f3-xxxx-xxxx-xxxxxxxxxxxx</client_id>
                <tenant_name>7c6bb73f-2328-xxxx-xxxx-xxxxxxxxxxxx</tenant_name>
            </azure_app>
        </azure_auto_login>
        …
    </ztna>
<forticlient_configuration>
```



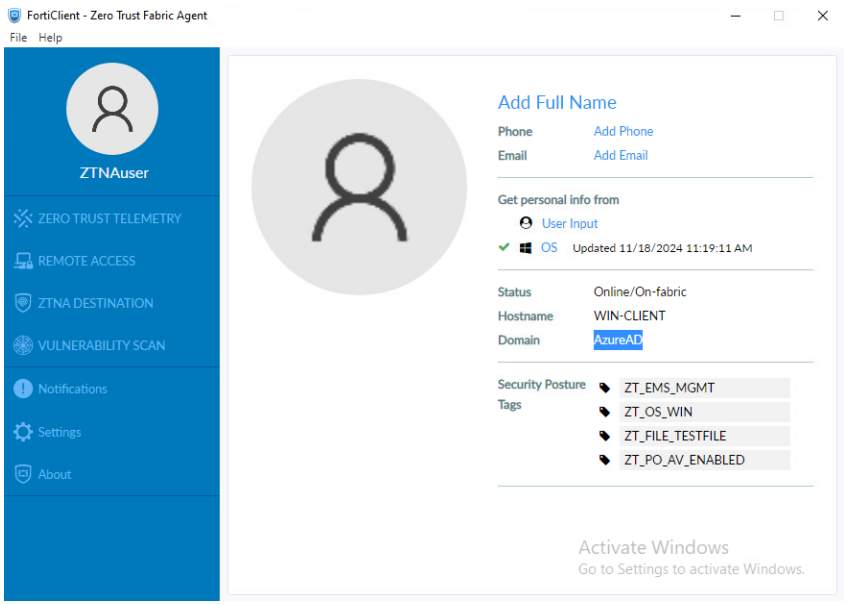
Not shown here, but TCP forwarding destinations are defined in the ZTNA destination profile.

Windows endpoint configuration

The endpoint client should already be joined to the Entra ID domain.



Once logged in and FortiClient is registered to EMS, click the user avatar to display information about the user, including the login domain name.

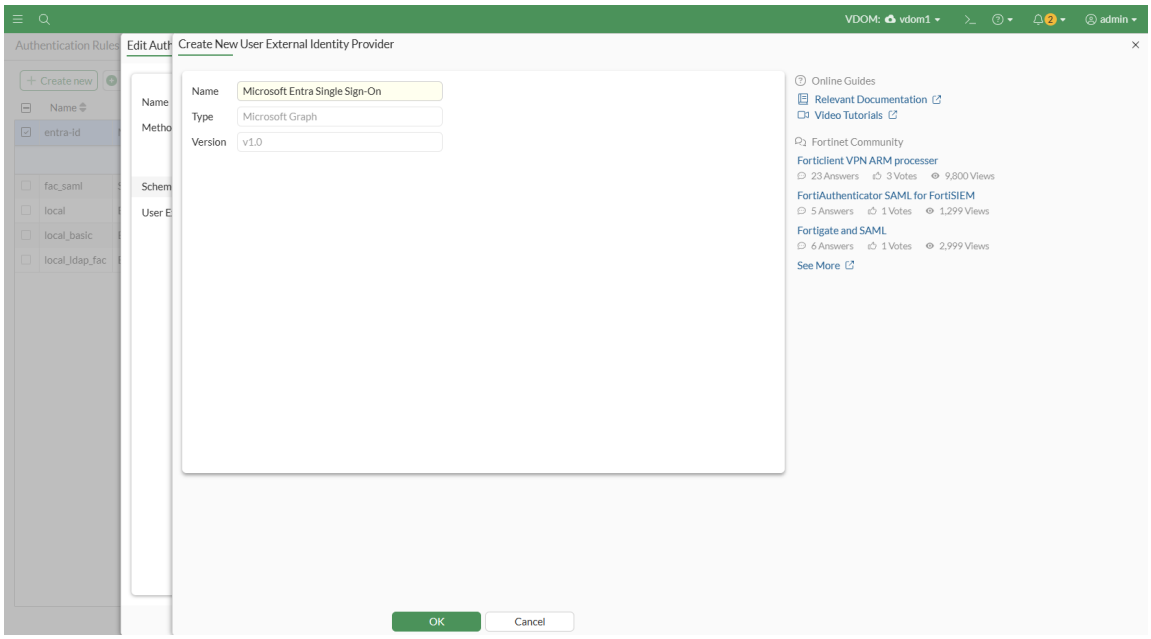


FortiGate configuration

FortiGate ZTNA server configurations remain the same. EMS connectivity and ZTNA server configurations are not explained in this topic. The major difference in FortiGate configurations is in the authentication scheme, rule, and user group. Therefore, the configuration process starts with authentication scheme and rule.

To configure the authentication scheme and rule:

1. Go to *Policy & Objects* > *Authentication*, and click the *Authentication Schemes* tab.
2. Click *Create new*.
3. Enter the name.
4. For method, click *+* and select the method *Microsoft Entra Single Sign-On*.
5. Create a new *User External Identity Provider*:
   a. Click the drop-down and click *Create*.
   b. Enter a name.
   c. Leave the default values of *Microsoft Graph* and *v1.0*, which are enabled by default.



d. Click *OK*.

6. Select the newly created External Identity Provider.
7. Click *OK* to save.
8. Click the *Authentication Rule* tab. Click *Create new*.
   a. Configure the following:

Field	Value
Name	Name of the rule
Source Address	*all* to apply this rule to all source traffic
Incoming Interface	*any* to apply this on any interface
Protocol	HTTP
Authentication Scheme	Disabled
IP-based Authentication	Enabled
SSO Authentication Scheme	Enabled. Set to the name of the new authentication scheme
Enable this rule	Enabled

   b. Click *OK* to save.

Next, you must configure the user group that is authorized to access the protected TCP application. The user group is queried against Entra ID using Microsoft Graph v1.0.

To configure the user group:

1. Go to *User & Authentication* > *User Groups*, and click *Create new*.
2. Enter a name.
3. Keep type as default (Firewall).

4. Under *Remote Groups*, click *Add*.
   a. For *Remote Server*, select the User External Identity Provider that was previously created.
   b. Optionally for *Groups*, specify the group name/ID corresponding to the user group object ID on Entra ID to filter by group.
   c. Click *OK*.
5. Click *OK* to save.

Finally, you must apply the user group to a ZTNA policy. In this example, a full ZTNA policy is configured.

To configure a full ZTNA policy:

1. Go to *Policy & Objects* > *Proxy Policy*, and click *Create new*.
2. Configure the following:

Field	Value
Name	Name of the policy
Type	ZTNA
Incoming Interface	Interface used by FortiGate to listen for the ZTNA connections.
Source	Address: Specify the source, or select *all* to allow all traffic sources. User: select the user group that was created in the previous step.
Security Posture Tag	Choose any tags that may be applicable.
Destination	Specify a destination server, or select *all* for all servers.
ZTNA Server	Specify the ZTNA server to allow access to.
Schedule	always
Action	ACCEPT

3. Configure other optional settings as needed.
4. Click *OK* to save.

Sample CLI configurations

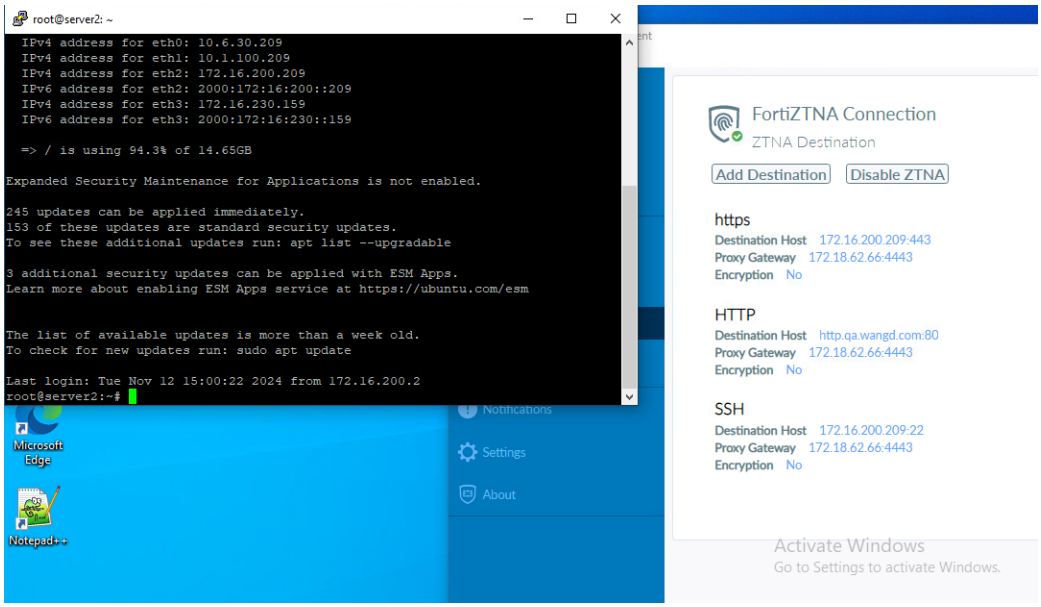
```
config user external-identity-provider
    edit "entra_id"
        set type ms-graph
        set version v1.0
    next
end

config authentication scheme
    edit "entra_id"
        set method entra-sso
        set external-idp "entra_id"
    next
end
```

```
config authentication rule
    edit "entra_id"
        set srcaddr "all"
        set dstaddr "all"
        set ip-based disable
        set sso-auth-method "entra_id"
        set web-auth-cookie enable
    next
end

config user group
    edit "aad_group"
        set member "entra_id"
        config match
            edit 1
                set server-name "entra_id"
                set group-name "22e10685-49d3-4af8-b75b-57f22cb8aa32"
            next
        end
    next
end

config firewall proxy-policy
    edit 3
        set name "fzt"
        set proxy access-proxy
        set access-proxy "ztna_4"
        set srcintf "port2"
        set srcaddr "all"
        set dstaddr "all"
        set ztna-ems-tag "EMS7_ZTNA_ZT_OS_LINUX" "EMS7_ZTNA_ZT_OS_MAC" "EMS7_ZTNA_ZT_OS_WIN"
        set action accept
        set schedule "always"
        set logtraffic all
        set srcaddr6 "all"
        set dstaddr6 "all"
        set groups "aad_group"
        set ssl-ssh-profile "custom-deep-inspection"
    next
end
```

Testing and verification

In this example, a user logged in to Windows as ztnauser@fosdevqa.onmicrosoft.com in the AzureAD domain attempts to access a host 172.16.200.209:22 through SSH.

As the screenshot shows, the user can connect to the server through SSH without logging in again.



From the FortiGate, go to *Log & Report > ZTNA Traffic* to view corresponding traffic logs. Alternatively, use the CLI to run:

```
# execute log filter field subtype ztna
# execute log display
1: date=2024-11-18 time=14:46:53 eventtime=1731970013042933838 tz="-0800" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="vdom1" srcip=10.1.100.214 srcport=51765
srcintf="port2" srcintfrole="lan" dstcountry="Reserved" srccountry="Reserved" dstip=172.18.62.66
dstport=4443 dstintf="port1" dstintfrole="lan" sessionid=100630 service="tcp/4443"
proxyapptype="ztna-proxy" proto=6 action="accept" policyid=3 policytype="proxy-policy"
poluuid="5a72570a-932a-51ef-4e9e-a2d01c654769" policyname="fzt" trandisp="dnat"
tranip=172.16.200.209 tranport=22 appcat="unscanned" duration=121
user="ztnauser@fosdevqa.onmicrosoft.com" group="aad_group" gatewayid=2 vip="ztna_4"
accessproxy="ztna_4" clientdeviceid="0D48C06390CA42D78158BE90C480A3F0"
clientdevicemanageable="manageable" clientdeviceems="FCTEMS8821001322" clientdevicetags="ZT_PO_AV_
ENABLED/ZT_OS_WIN/ZT_FILE_TESTFILE/ZT_EMS_MGMT/all_registered_clients" clientcert="yes"
emsconnection="online" wanin=1729 rcvdbyte=1729 wanout=1412 lanin=6820 sentbyte=6820 lanout=4279
fctuid="0D48C06390CA42D78158BE90C480A3F0" unauthuser="ZTNAuser" unauthusersource="forticlient"
srcdomain="fosdevqa.onmicrosoft.com" srcremote=204.101.161.19
```

ZTNA troubleshooting and debugging commands

The following debug commands can be used to troubleshoot ZTNA issues:

Command	Description
`# diagnose endpoint fctems test-connectivity <EMS>`	Verify FortiGate to FortiClient EMS connectivity.

Command	Description
`# execute fctems verify <EMS>`	Verify the FortiClient EMS's certificate.
`# diagnose test application fcnacd 2`	Dump the EMS connectivity information.
`# diagnose debug app fcnacd -1` `# diagnose debug enable`	Run real-time FortiClient NAC daemon debugs.
`# diagnose endpoint ec-shm list <ip> <mac> <EMS_serial_number> <EMS_tenant_id>`	Show the endpoint record list. Optionally, add filters.
`# diagnose endpoint lls-comm send ztna find-uid <uid> <EMS_serial_number> <EMS_tenant_id>`	Query endpoints by client UID, EMS serial number, and EMS tenant ID.
`# diagnose endpoint lls-comm send ztna find-ip-vdom <ip> <vdom>`	Query endpoints by the client IP-VDOM pair.
`# diagnose wad dev query-by uid <uid> <EMS_serial_number> <EMS_tenant_id>`	Query from WAD diagnose command by UID, EMS serial number, and EMS tenant ID.
`# diagnose wad dev query-by ipv4 <ip>`	Query from WAD diagnose command by IP address.
`# diagnose firewall dynamic list`	List EMS security posture tags and all dynamic IP and MAC addresses.
`# diagnose test application fcnacd 7` `# diagnose test application fcnacd 8`	Check the FortiClient NAC daemon ZTNA and route cache.
`# diagnose wad worker policy list`	Display statistics associated with application gateway rules.

`# diagnose wad debug enable category all`
`# diagnose wad debug enable level verbose`
`# diagnose debug enable`
	Run real-time WAD debugs.

`# diagnose debug reset`	Reset debugs when completed.
`# diagnose wad user list`	List the ZTNA/proxy users.
`# diagnose wad user clear <id> <ip> <vdom>`	Clear a single ZTNA/proxy user.
`# diagnose wad user clear`	Clear all ZTNA/proxy users.
`# diagnose wad ztna-web-proxy health-check status`	Query realserver health-check information under ZTNA web-proxy.

The WAD daemon handles proxy related processing. The FortiClient NAC daemon (fcnacd) handles FortiGate to EMS connectivity.

Troubleshooting usage and output

1. Verify the FortiGate to EMS connectivity and EMS certificate:

```
# diagnose endpoint fctems test-connectivity WIN10-EMS
Connection test was successful:
```

```
# execute fctems verify WIN10-EMS
Server certificate already verified.
```

```
# diagnose test application fcnacd 2
EMS context status:
FortiClient EMS number 1:
        name: WIN10-EMS confirmed: yes
        fetched-serial-number: FCTEMS0000109188
Websocket status: connected
```

2. If fcnacd does not report the proper status, run real-time fcnacd debugs:

```
# diagnose debug app fcnacd -1
# diagnose debug enable
```

3. Verify the following information about an endpoint:
   - Network information
   - Registration information
   - Client certificate information
   - Device information
   - Vulnerability status
   - Relative position with the FortiGate

```
# diagnose endpoint ec-shm list 10.6.30.214
Record 0:
                IP Address = 10.6.30.214
                MAC Address = 00:0c:29:ba:1e:61
                MAC list = 00:0c:29:ba:1e:61;00:0c:29:ba:1e:6b;
                VDOM = root (0)
                EMS serial number: FCTEMS8821001322
                EMS tenant id: 00000000000000000000000000000000
                Client cert SN: 17FF6595600A1AF53B87627AB4EBEDD032593E64
                Quarantined: no
                Online status: online
                Registration status: registered
                On-net status: on-net
                Gateway Interface: port2
                FortiClient version: 7.0.0
                AVDB version: 84.778
                FortiClient app signature version: 18.43
                FortiClient vulnerability scan engine version: 2.30
                FortiClient UID: 5FCFA3ECDE4D478C911D9232EC9299FD
                Host Name: ADPC
```

```
    …
                Number of Routes: (1)
                        Gateway Route #0:
                                - IP:10.1.100.214, MAC: 00:0c:29:ba:1e:6b, Indirect: no
                                - Interface:port2, VFID:0, SN: FG5H1E5819902474
online records: 1; offline records: 0; quarantined records: 0
```

4. Query the endpoint information, include security posture tags, by UID or IP address:

```
# diagnose endpoint lls-comm send ztna find-uid 5FCFA3ECDE4D478C911D9232EC9299FD
FCTEMS8821001322 00000000000000000000000000000000
UID: 5FCFA3ECDE4D478C911D9232EC9299FD
EMS Fabric ID: FCTEMS8821001322:00000000000000000000000000000000
        status code:ok
        Domain: qa.wangd.com
        User: user1
        Cert SN:17FF6595600A1AF53B87627AB4EBEDD032593E64
        EMS SN: FCTEMS8821001322
        Routes(1):
         - route[0]: IP=10.1.100.214, VDom=root
        Tags(3):
         - tag[0]: name=ZT_OS_WIN
         - tag[1]: name=all_registered_clients
         - tag[2]: name=Medium
```

```
# diagnose endpoint lls-comm send ztna find-ip-vdom 10.1.100.214 root
UID: 5FCFA3ECDE4D478C911D9232EC9299FD
        status code:ok
        Domain: qa.wangd.com
        User: user1
        Cert SN:17FF6595600A1AF53B87627AB4EBEDD032593E64
        EMS SN: FCTEMS8821001322
        Routes(1):
         - route[0]: IP=10.1.100.214, VDom=root
        Tags(3):
         - tag[0]: name=ZT_OS_WIN
         - tag[1]: name=all_registered_clients
         - tag[2]: name=Medium
```

5. Query endpoint information from WAD by UID or IP address:

```
# diagnose wad dev query-by uid 5FCFA3ECDE4D478C911D9232EC9299FD FCTEMS8821001322
00000000000000000000000000000000
Attr of type=0, length=32, value(ascii)=5FCFA3ECDE4D478C911D9232EC9299FD
Attr of type=4, length=30, value(ascii)=MAC_FCTEMS8821001322_ZT_OS_WIN
Attr of type=4, length=26, value(ascii)=FCTEMS8821001322_ZT_OS_WIN
Attr of type=4, length=43, value(ascii)=MAC_FCTEMS8821001322_all_registered_clients
Attr of type=4, length=39, value(ascii)=FCTEMS8821001322_all_registered_clients
Attr of type=4, length=27, value(ascii)=MAC_FCTEMS8821001322_Medium
Attr of type=4, length=23, value(ascii)=FCTEMS8821001322_Medium
Attr of type=5, length=18, value(ascii)=FOSQA@qa.wangd.com
Attr of type=6, length=40, value(ascii)=17FF6595600A1AF53B87627AB4EBEDD032593E64
```

```
# diagnose wad dev query-by ipv4 10.1.100.214
Attr of type=0, length=32, value(ascii)=5FCFA3ECDE4D478C911D9232EC9299FD
Attr of type=4, length=30, value(ascii)=MAC_FCTEMS8821001322_ZT_OS_WIN
Attr of type=4, length=26, value(ascii)=FCTEMS8821001322_ZT_OS_WIN
Attr of type=4, length=43, value(ascii)=MAC_FCTEMS8821001322_all_registered_clients
Attr of type=4, length=39, value(ascii)=FCTEMS8821001322_all_registered_clients
Attr of type=4, length=27, value(ascii)=MAC_FCTEMS8821001322_Medium
Attr of type=4, length=23, value(ascii)=FCTEMS8821001322_Medium
Attr of type=5, length=18, value(ascii)=FOSQA@qa.wangd.com
Attr of type=6, length=40, value(ascii)=17FF6595600A1AF53B87627AB4EBEDD032593E64
```

6. List all the dynamic ZTNA IP and MAC addresses learned from EMS:

```
# diagnose firewall dynamic list
List all dynamic addresses:
FCTEMS0000109188_all_registered_clients: ID(51)
        ADDR(172.17.194.209)
        ADDR(192.168.40.8)
…
FCTEMS0000109188_Low: ID(78)
        ADDR(172.17.194.209)
        ADDR(192.168.40.8)
…
FCTEMS0000109188_Malicious-File-Detected: ID(190)
        ADDR(172.17.194.209)
        ADDR(192.168.40.8)
…
```

7. Check the FortiClient NAC daemon ZTNA and route cache:

```
# diagnose test application fcnacd 7
ZTNA Cache:
-uid 5FCFA3ECDE4D478C911D9232EC9299FD: { "tags": [ "ZT_OS_WIN", "all_registered_clients",
"Medium" ], "domain": "qa.wangd.com", "user_name": "user1", "client_cert_sn":
"17FF6595600A1AF53B87627AB4EBEDD032593E64", "owner": "FOSQA@qa.wangd.com", "gateway_route_
list": [ { "gateway_info": { "fgt_sn": "FG5H1E5819902474", "interface": "port2", "vdom":
"root" }, "route_info": [ { "ip": "10.1.100.214", "mac": "00-0c-29-ba-1e-6b", "route_type":
"direct" } ] } ], "ems_sn": "FCTEMS8821001322" }
```

```
# diagnose test application fcnacd 8
IP-VfID Cache:
IP: 10.1.100.206, vfid: 0, uid: 3DED29B54386416E9888F2DCBD2B9D21
IP: 10.1.100.214, vfid: 0, uid: 5FCFA3ECDE4D478C911D9232EC9299FD
```

8. Troubleshoot WAD with real-time debugs to understand how the proxy handled a client request:

```
# diagnose wad debug enable category all
# diagnose wad debug enable level verbose
# diagnose debug enable

[0x7fbd7a46bb60] Received request from client: 10.10.10.20:56312
GET / HTTP/1.1 Host: 192.168.2.86:8443 Connection: keep-alive Cache-Control: max-age=0
```

```
Upgrade-Insecure-Requests: 1 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/89.0.4389.90 Safari/537.36 Edg/89.0.774.57
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,applicat
ion/signed-exchange;v=b3;q=0.9 Sec-Fetch-Site: none Sec-Fetch-Mode: navigate Sec-Fetch-User:
?1 Sec-Fetch-Dest: document Accept-Encoding: gzip, deflate, br Accept-Language: en-US,en;q=0.9
[p:29957][s:458767][r:1] wad_http_marker_uri(1269): path=/ len=1
[p:29957][s:458767][r:1] wad_http_parse_host(1641): host_len=17
[p:29957][s:458767][r:1] wad_http_parse_host(1677): len=12
[p:29957][s:458767][r:1] wad_http_parse_host(1686): len=4
[p:29957][s:458767][r:1] wad_http_str_canonicalize(2180): path=/ len=1 changes=0
[p:29957][s:458767][r:1] wad_http_str_canonicalize(2189): path=/ len=1 changes=0
[p:29957][s:458767][r:1] wad_http_normalize_uri(2232): host_len=12 path_len=1 query_len=0
[p:29957][s:458767][r:1] wad_vs_proxy_match_gwy(2244): 6:WIN2K16-P1: matching gwy with vhost(_
def_virtual_host_)
[p:29957][s:458767][r:1] wad_vs_proxy_match_vhost(2293): 6:WIN2K16-P1: matching vhost by:
192.168.2.86
[p:29957][s:458767][r:1] wad_vs_matcher_map_find(477): Empty matcher!
[p:29957][s:458767][r:1] wad_vs_proxy_match_vhost(2296): 6:WIN2K16-P1: no host matched.
[p:29957][s:458767][r:1] wad_vs_proxy_match_gwy(2263): 6:WIN2K16-P1: matching gwy by (/) with
vhost(_def_virtual_host_).
[p:29957][s:458767][r:1] wad_pattern_matcher_search(1210): pattern-match succ:/
[p:29957][s:458767][r:1] wad_vs_proxy_match_gwy(2271): 6:WIN2K16-P1: Matched gwy(1) type
(https).
[p:29957][s:458767][r:1] wad_http_vs_check_dst_ovrd(776): 6:WIN2K16-P1:1: Found server:
192.168.20.6:443
[p:29957][s:458767][r:1] wad_http_req_exec_act(9296): dst_addr_type=3 wc_nontp=0 sec_web=1
web_cache=0 req_bypass=0
[p:29957][s:458767][r:1] wad_http_req_check_policy(8117): starting policy matching(vs_pol=
1):10.10.10.20:56312->192.168.20.6:443
[p:29957][s:458767][r:1] wad_fw_addr_match_ap(1524): matching ap:WIN2K16(7) with vip
addr:WIN2K16-P1(10)
[p:29957][s:458767][r:1] wad_fw_addr_match_ap(1524): matching ap:WIN2K16-P1(10) with vip
addr:WIN2K16-P1(10)
[p:29957][s:458767][r:1] wad_http_req_policy_set(6811): match pid=29957 policy-id=2 vd=0 in_
if=3, out_if=7 10.10.10.20:56312 -> 192.168.20.6:443
[p:29957][s:458767][r:1] wad_cifs_profile_init(93): CIFS Profile 0x7fbd7a5bf200 [] of type 0
created
[p:29957][s:458767][r:1] wad_http_req_proc_policy(6622): web_cache(http/https=0/0, fwd_
srv=<nil>.
[p:29957][s:458767][r:1] wad_auth_inc_user_count(1668): increased user count, quota:128000, n_
shared_user:2, vd_used: 2, vd_max: 0, vd_gurantee: 0
[p:29957][s:458767][r:1] __wad_fmem_open(563): fmem=0xaaee3e8, fmem_name='cmem 336 bucket',
elm_sz=336, block_sz=73728, overhead=20, type=advanced
[p:29957][s:458767][r:1] __wad_hauth_user_node_hold(2107): wad_hauth_user_node_alloc (1568):
holding node 0x7fbd76d48060
mapping user_node:0x7fbd76d48060, user_ip:0x7fbd7a57b408(0), user:0x7fbd7a5cf420(0)
[p:29957][s:458767][r:1] __wad_hauth_user_node_hold(2107): wad_user_node_stats_hold (483):
holding node 0x7fbd76d48060
[p:29957][s:458767][r:1] __wad_hauth_user_node_hold(2107): wad_http_session_upd_user_node
(4813): holding node 0x7fbd76d48060
```

```
[p:29957][s:458767][r:1] wad_http_req_proc_policy(6698): policy result:vf_id=0:0 sec_
profile=0x7fbd7a5bef00 set_cookie=0
[p:29957][s:458767][r:1] wad_http_urlfilter_check(381): uri_norm=1 inval_host=0 inval_url=0
scan-hdr/body=1/0 url local=0 block=0 user-cat=0 allow=0 ftgd=0 keyword=0 wisp=0
[p:29957][s:458767][r:1] wad_http_req_proc_waf(1309): req=0x7fbd7a46bb60 ssl.deep_scan=1
proto=10 exempt=0 waf=(nil) body_len=0 ua=Chrome/89.0.4389.90 skip_scan=0
[p:29957][s:458767][r:1] wad_http_req_proc_antiphish(5376): Processing antiphish request
[p:29957][s:458767][r:1] wad_http_req_proc_antiphish(5379): No profile
[p:29957][s:458767][r:1] wad_http_connect_server(4696): http session 0x7fbd7a532ac8
req=0x7fbd7a46bb60
[p:29957][s:458767][r:1] wad_http_srv_still_good(4575): srv((nil)) nontp(0) dst_type(3)
req: dst:192.168.20.6:443, proto:10)
hcs: dst:N/A:0, proto:1)
```

9. Verify the health status of a real server:

```
config ztna web-proxy
    edit "ZTNA-Webserver"
        set vip "webserver-VIP
        set host webserver-VHOST
        config api-gateway
            edit 1
                config realservers
                    edit 1
                        set ip 10.88.0.4
                        set health-check enable
                    next
                end
            next
        end
    next
end
```

```
# diagnose wad ztna-web-proxy health-check status
Link Monitor: ZP-ZTNA-Webserver-1-2, Status: alive, Server num(1), cfg_version=0 Flags=0x1
init, Create time: Thu May  7 20:08:00 2026
VRF: 0
Interval: 1000 ms
Service-detect: disable
Diffservcode: 000000
Class-ID: 0
Transport-Group: 0
Class-ID: 0
  Peer: 10.88.0.4(10.88.0.4)
        protocol: ping(9043), state: alive
                Latency(Min/Max/Avg): 0.487/0.932/0.638 ms
                Jitter(Min/Max/Avg): 0.003/0.328/0.109 ms
                Packet lost: 0.000%
                MOS: 4.404
                Number of out-of-sequence packets: 0
                Fail Times(0/5)
                Packet sent: 2436, received: 2436, Sequence(sent/rcvd/exp): 2437/2437/2438
```

Always reset the debugs after using them:

```
# diagnose debug reset
```

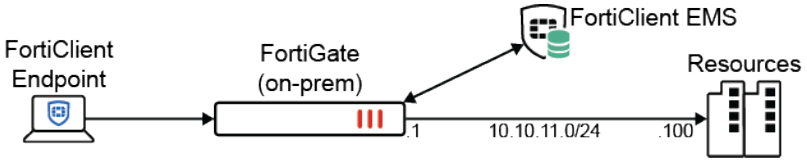
ZTNA troubleshooting scenarios

This topic describes how to troubleshoot common FortiClient endpoint IP/MAC access control issues for the following topologies:

- ZTNA access control on page 1827
- IP/MAC based access control on page 1828

ZTNA access control

In this topology, FortiClient endpoints use an SSL encrypted connection to the FortiGate application gateway to access protected resources. FortiGate works with FortiClient EMS to use a combination of IP/MAC addresses and security posture tags to control FortiClient endpoint access to resources.



This section describes how to handle the following errors:

- Invalid ZTNA certificate on page 1827
- ZTNA policy mismatch on page 1828

Invalid ZTNA certificate

When FortiClient attempts to access a server protected by ZTNA, an *Invalid ZTNA certificate* error is shown. This error often appears when the serial number for the ZTNA certificate differs between the endpoint and the FortiGate.

1. Check the serial number for the ZTNA certificate on the endpoint and the FortiGate:
   a. On the endpoint, check the serial number for the certificate.
   b. On the FortiGate, check the serial number for the client certificate by running the following command:

```
# diagnose endpoint ec-shm list
```

2. If the serial number for the ZTNA certificate differs between the endpoint and the FortiGate, and the serial number on the FortiGate is comprised of zeros, check the following:
   a. For FortiClient, make sure that the endpoint is running FortiClient 7.0 or later. FortiClient versions earlier than 7.0 do not support ZTNA.

   b. For FortiClient EMS, make sure that ZTNA is enabled. Check the profile on EMS and the endpoint's summary information.
   c. For licensing, make sure that you have a ZTNA agent license entitlement. Only some license types support ZTNA.
3. If the serial numbers still do not match, deregister FortiClient from EMS, and then connect FortiClient to EMS again to trigger a new certificate signing request.

ZTNA policy mismatch

In most cases, FortiGate denies incoming ZTNA requests because the endpoint FortiClient does not meet the tagging criteria configured in the ZTNA rule and is considered a policy mismatch.

1. On the FortiGate, look at the ZTNA event logs and the forwarded logs.
2. Run the following commands on the ZTNA server:

```
# diagnose wad debug enable category policy
```

```
# diagnose wad debug enable level verbose
```

```
# diagnose debug enable
```

   The command output contains incoming ZTNA requests and the FortiGate process for matching the connection to a ZTNA rule.
3. Verify the zero trust tags for the endpoint:
   - On FortiClient, verify the applied tags. Click the avatar to view the zero trust tags.
   - On FortiClient EMS, verify the endpoint's tags. Go to the endpoint list and click the endpoint.
   - On FortiGate, verify the tags using the following commands:
     - Display ZTNA cache data for all endpoints:

       ```
       # diagnose test application fcnacd 7
       ```

     - Display ZTNA cache data for an individual endpoint:

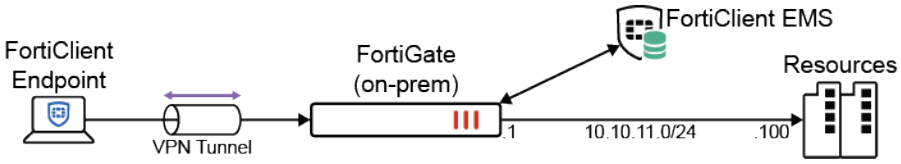
       ```
       # diagnose wad dev query-by uid <UID> <EMS S/N> <tenant ID>
       ```

4. If the tagging information differs between FortiGate and EMS, examine the EMS tag exchange communication between FortiGate and EMS by looking at the cmNotify and python logs in the debug diagnostics for EMS.
   For more information about FortiClient EMS diagnostics, see Generate Diagnostic Log in the FortiClient EMS Administration Guide.

IP/MAC based access control

In the following ZTNA topology, FortiClient endpoints use VPN to access resources. FortiGate works with FortiClient EMS to use a combination of IP/MAC addresses and security posture tags to control FortiClient endpoint access to resources.

For more information, see ZTNA IP MAC based access control example on page 1776.



Security posture tag information missing on the FortiGate

If the IP address for the FortiClient endpoint is not associated with a security posture tag on the FortiGate, a firewall policy mismatch occurs, and the FortiGate denies network access to the FortiClient endpoint.

The following workflow summarizes how FortiGate retrieves the IP address and tags for the FortiClient endpoint to help you better understand how to troubleshoot the situation:

1. FortiClient establishes a VPN connection to the FortiGate.
2. FortiGate uses the API to pass FortiClient's UUID and VPN IP address to FortiClient EMS.
3. FortiGate requests system information and tags from FortiClient based on the response from EMS.

Based on the workflow, start troubleshooting before the FortiClient endpoint attempts to establish a VPN connection to FortiGate. On FortiGate, run the following commands:

```
# diagnose debug application fcnacd -1
```

```
# diagnose debug console timestamp enable
```

```
# diagnose endpoint filter show-large-data yes
```

```
# diagnose debug enable
```

The following outputs illustrate how to examine the command output. The output can differ between environments. The outputs help illustrate how to understand the communication between FortiGate and FortiClient EMS.

In the following output, FortiGate's VPN daemon sends FortiClient's UUID and the VPN IP address to FortiClient EMS using the API. The NAC daemon makes the API call to send the details to FortiClient EMS:

```
2022-10-17 08:50:41 [fcems_call_vpn_client_gateway_call:1147] VPN act connect (UID:
3358095CFDCB414B9EDA49ADE79AF428, Interface: port1, IP: 10.212.134.200, VDom: root,
 FortiGate-SN: FGVM02TM22018374) added to EMS FortiClientEMS
(FCTEMS8821003330:00000000000000000000000000000000)
2022-10-17 08:50:41 [ec_ez_worker_base_prep_resolver:373] Outgoing interface index 0 for 2
(FortiClientEMS).
2022-10-17 08:50:41 [ec_ez_worker_prep_data_url:98] request (206):
"""
{"sn_list":["FGVM02TM22018374"],"uid_list":
[{"uid":"3358095CFDCB414B9EDA49ADE79AF428","ip":"10.212.134.200","is_
delete":false,"vdom":"root","interface":"port1","sn":"F
GVM02TM22018374"}],"is_snapshot":false}
"""

2022-10-17 08:50:41 [ec_ez_worker_prep_data_url:176] Full URL:
https://172.31.200.183/api/v1/fgt/gateway_details/vpn
```

```
2022-10-17 08:50:41 [ec_ems_context_submit_work:498] Call submitted successfully.
    obj-id: 7, desc: REST API to send updated regarding VPN updates., entry: api/v1/fgt/gateway_
details/vpn.

2022-10-17 08:50:41 [ec_daemon_submit_sock_call:49] sent 244,244
2022-10-17 08:50:42 [_renew_resolver:219] called.

2022-10-17 08:50:42 [ec_ez_worker_process:347] Processing call for obj-id: 7, entry:
"api/v1/fgt/gateway_details/vpn"
2022-10-17 08:50:42 [ec_ez_worker_process:366] reply:
"""
{"result": {"retval": 1, "message": "FortiGate VPN connection details updated successfully"}}
"""
```

The following example from the fcmNotify.log file on FortiClient EMS shows how FortiClient EMS interprets the information sent from FortiGate:

```
2022-10-26 11:59:37,817 DEBUG ems_logger 6 7 [VPN Gateway Details]: Request made with params:
{'is_snapshot': False, 'sn_list': ['FG10E0TB20903081', 'FG10E0TB20903034'], 'uid_list': [{'uid':
'D997B2A7A78E4E6F832309FF97FC2215', 'vdom': 'root', 'interface': 'EXT', 'sn': 'FG10E0TB20903081',
'ip': '10.1.18.61', 'is_delete': False}]}.
2022-10-26 11:59:38,281 DEBUG ems_logger 6 7 [Sysinfo c44cc74b1185431491f71c133c097f00 Certificate
user: FG10E0TB20903081]: Request with SN [FG10E0TB20903034,FG10E0TB20903081] success. Returned 1
endpoints. uid_offset: D997B2A7A78E4E6F832309FF97FC2215, updated_after: 2022-10-26
15:59:37.8237471, is_final: True
2022-10-26 11:59:38,543 DEBUG ems_logger 6 7 [UID-Tags e6ecc42c058e48b2b71cf7d65ecd432c
Certificate user: FG10E0TB20903081]: Request with SN [FG10E0TB20903034,FG10E0TB20903081] success.
uid_offset: D997B2A7A78E4E6F832309FF97FC2215, updated_after: 2022-10-26 15:59:37.8227461, is_
final: True
```

FortiGate uses the information from FortiClient EMS to make a targeted API call to FortiClient EMS to retrieve both system information and tag information (with the means of uid_offset and updated_after parameters) for the endpoint. The following is the API call to retrieve the tags from FortiClient EMS:

```
https://172.31.200.182/api/v1/report/fct/uid_tags?sn_list[]=FGVM02TM22018374&updated_after=2022-
10-17 15:59:37.8227461&uid_offset=3358095CFDCB414B9EDA49ADE79AF428
```

The following in an example of the API call and subsequent communication between FortiGate and FortiClient EMS to retrieve tags for the FortiClient endpoint IP address:

```
2022-10-17 08:50:42 [ec_ez_worker_base_prep_resolver:373] Outgoing interface index 0 for 2
(FortiClientEMS).
2022-10-17 08:50:42 [ec_ez_worker_prep_data_url:98] request (26):
"""
sn_list[]=FGVM02TM22018374
"""

2022-10-17 08:50:42 [ec_ez_worker_prep_data_url:176] Full URL:
https://172.31.200.183/api/v1/report/fct/uid_tags?sn_list[]=FGVM02TM22018374
2022-10-17 08:50:42 [ec_ems_context_submit_work:498] Call submitted successfully.
    obj-id: 13, desc: REST API to get updates of tags associated with FCT UID., entry:
api/v1/report/fct/uid_tags.
```

```
2022-10-17 08:50:43 [ec_ez_worker_process:347] Processing call for obj-id: 12, entry:
"api/v1/report/fct/tags"
2022-10-17 08:50:43 [ec_ez_worker_process:366] reply:
"""
{"result": {"retval": 1, "message": "Returned FCT incremental tags information."}, "data": {"tag_
uid_offset": "F200BAC5-352C-41AD-9BC2-C6D177D391B1", "updated_after":"2022-10-17
15:52:20.4951668", "is_zipped": true, "is_final": true, "unzipped_size": 3508, "data":
"eJzFl0tv4zYUhf9KoXVuwadIZsfnYBYTFEgwsygKQbGYVKgsGZKcJg3mv/c66SNAa04BF87GgERa59Mhe ...
BLXtfB2IRdYii2Qx9/uI6+scMw/XrUzcMp/B3U5zwm"}}
"""

2022-10-17 08:50:43 [fcems_json_unzip:285] unzipped:
"""
{"command_version":2,"serial":"FCTEMS8821003330","device_type":"fortiems","commands":
[{"command":"update","addresses":[{"uuid":"814CA385-A346-4028-91FE-06011FFBC8A1","tag_properties":
{"name":"vul_enabled","type":"zero_trust"},"type":"ipblock","values":[]},{"uuid":"814CA385-A346-
4028- ... -93B7-E15BB3007AEC","tag_properties":{"name":"FortiESNAC.exe","type":"zero_trust"}},
{"uuid":"82DF3EC6-9D1B-4200-A3C6-366D9AFF4ED0","tag_properties":{"name":"IPSEC_
Allowed","type":"zero_trust"}}]}]}
"""
```

Other useful CLI commands

Output the JSON-formatted list of FortiGate interfaces (gateways) with IP and MAC addresses. This is the list that FortiGate sends to EMS so that EMS can identify the endpoints that are directly connected to the firewall:

```
# diagnose endpoint fctems json gateway-mac-request
```

Makes EMS execute API calls to the EMS API endpoints on demand:

```
# diagnose test application fcnacd 5
```

Send the gateway list to EMS on demand. It could be useful to execute `diagnose test application fcnacd 5` right after command during troubleshooting, as EMS will have an updated list of firewall interfaces:

```
# diagnose test application fcnacd 99
```

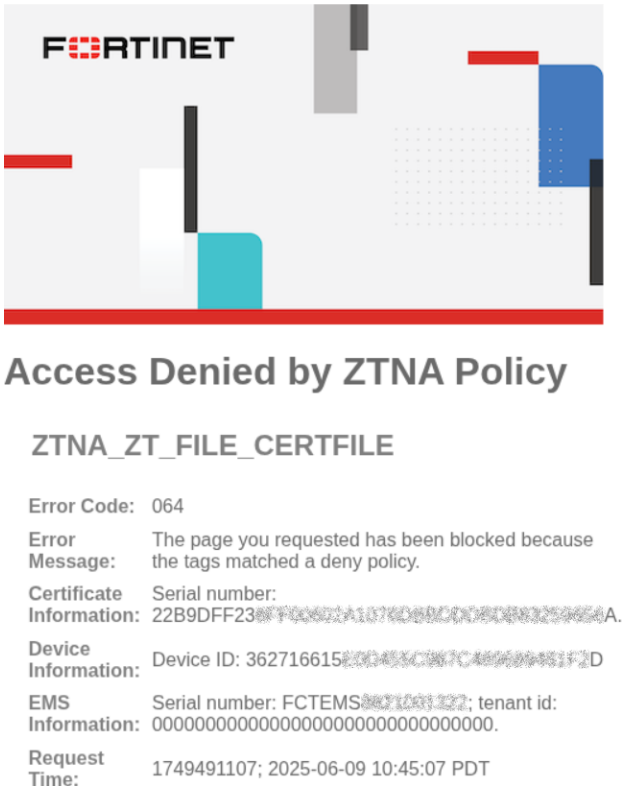
For more commands, see ZTNA troubleshooting and debugging commands on page 1820.

ZTNA troubleshooting with error messages

As part of the troubleshooting process, one essential step is to utilize the error messages displayed on the client endpoints to deduce the issue encountered. This is particularly important when accessing a resource from a web browser.

When a problem is encountered, the user is likely to see a replacement message.

For example, if a FortiGate is configured with a deny policy when matching a specific security posture tag called ZTNA_ZT_FILE_CERTFILE, and the endpoint client hit the deny policy because it matched this tag, an access denied message is would be shown.



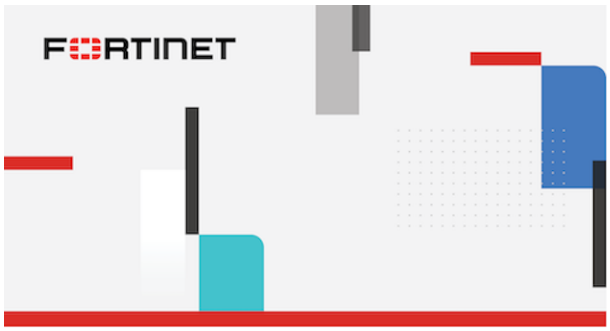
The replacement message includes all or a subset of the following information:

Field	Description
Title	Short summary of the issue
Subtitle	Additional details of the issue
Error Code	Code based on Error codes and replacement messages
Error Message	Specific error behavior encountered
User name	If the policy has user group defined, display the username
Certificate Information	The SN of the FortiClient endpoint certificate
Device Information	The Device ID of the FortiClient endpoint
Destination	The destination in which the client is trying to reach
EMS Information	Information about the EMS server including: • Server SN • Tenant ID
Request Time	When the request was made

Correlating with FortiGate ZTNA event logs

The replacement message information helps administrators correlate the event with the ZTNA event logs. If multiple EMS servers or tenants are used, the information helps pinpoint the relevant EMS server or tenant. Administrators can find the specific firewall policy that was matched to determine if the violation is legitimate, or if there is a misconfiguration.

For example, the following replacement message and log show that a user named mjohnson attempted to match a policy on the FortiGate against the ZTNA-webserver access proxy but was unable to match a policy that authorized its access due to no user group matched.



ZTNA Policy Denied

Error Code:	069
Error Message:	The page you requested has been blocked because authorization failed.
User Name:	mjohnson
Certificate Information:	Serial number: 230D5F9BF[redacted]6.
Device Information:	N/A.
Request Time:	1752624366; Wed Jul 16 00:06:06 2025

```
# execute log filter field subtype ztna
# execute log display
1 logs found.
1 logs returned.
1: date=2025-07-15 time=17:06:06 eventtime=1752624366166875160 tz="-0700" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=50177
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.0.3.10
dstport=9043 dstintf="root" dstintfrole="undefined" sessionid=13512 srcuuid="1a4694b6-6106-51f0-
d0c6-2c91f714c2ec" service="tcp/9043" proxyapptype="http" proto=6 action="deny" policyid=0
policytype="proxy-policy" appcat="unscanned" duration=0 user="mjohnson" vip="ZTNA-webserver"
accessproxy="ZTNA-webserver" clientdevicemanageable="manageable" clientcert="yes" msg="Traffic
denied because failed to match a policy or proxy-policy" wanin=0 rcvdbyte=0 wanout=0 lanin=2051
sentbyte=2051 lanout=3014 crscore=30 craction=131072 crlevel="high"
```

Additional Examples

Issue attempting to reach a destination

A client tries to access the destination 172.16.200.208:443, but the destination cannot be reached from the FortiGate. Before this enhancement, this resulted in a non-ZTNA replacement message, stating *504 Gateway Timeout: remote server did not respond to the proxy*.

After this enhancement, a *ZTNA Connection Error* replacement message with *Error Code 024* is displayed:

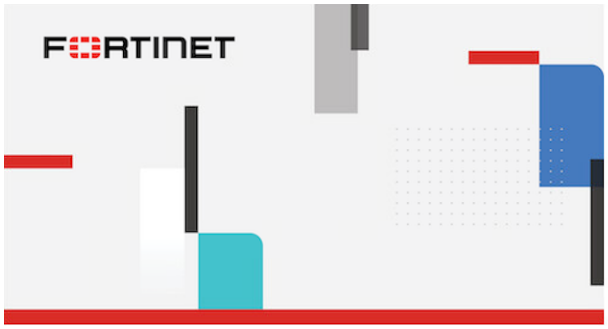
DNS resolution when matching a wildcard domain

A FortiGate has a real server definition for TCP forwarding using a wildcard domain match:

```
config firewall access-proxy
    edit "ztna"
        set vip "ztna"
            edit 2
                set url-map "/tcp"
                set service tcp-forwarding
                config realservers
                    edit 1
                        set address "ftnttest.com"
                        set domain "*.ftnttest.com"
                    next
                end
            next
```

```
        end
    next
end
```

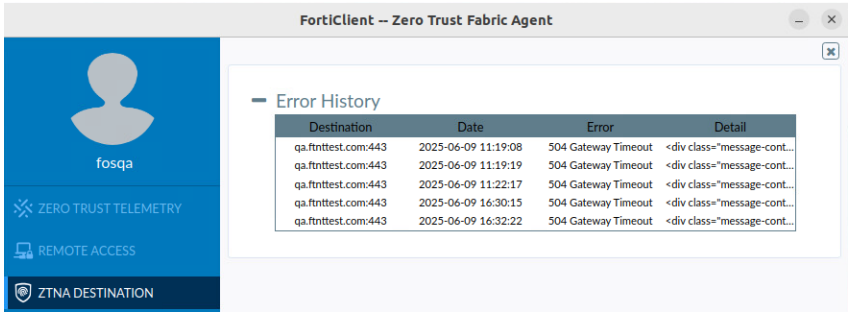
When the endpoint client requests the TCP forwarding resource that matches the wildcard domain, but this address cannot be resolved by the FortiGate, a *ZTNA Connection Error* replacement message with *Error Code 025* is displayed:



ZTNA Connection Error

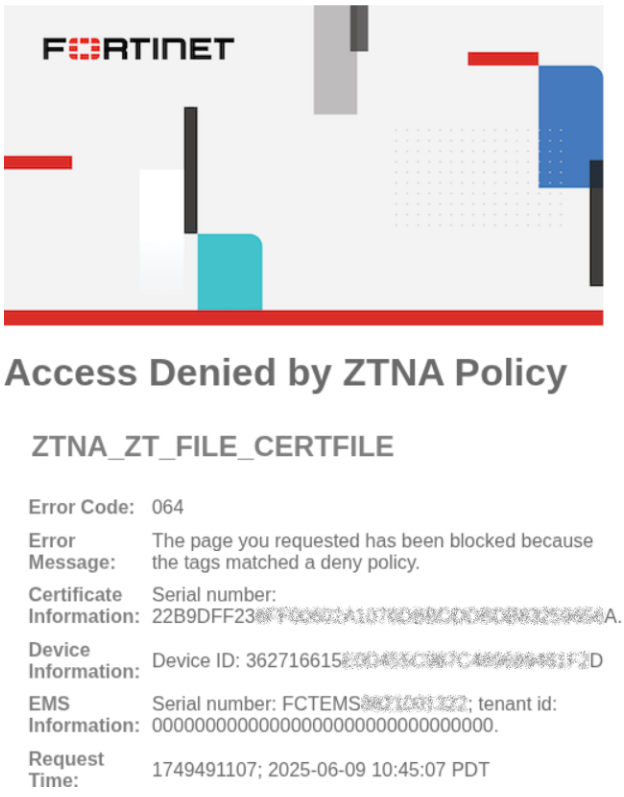
Error Code:	025
Error Message:	ZTNA connection failed due to a DNS issue.
Certificate Information:	Serial number: 22B9DFF23[redacted]A.
Device Information:	N/A.
Destination:	(N/A)
Request Time:	1749493337; 2025-06-09 11:22:17 PDT

Additionally, on FortiClient, the *Error History* indicates a *504 Gateway Timeout* when accessing the resource:



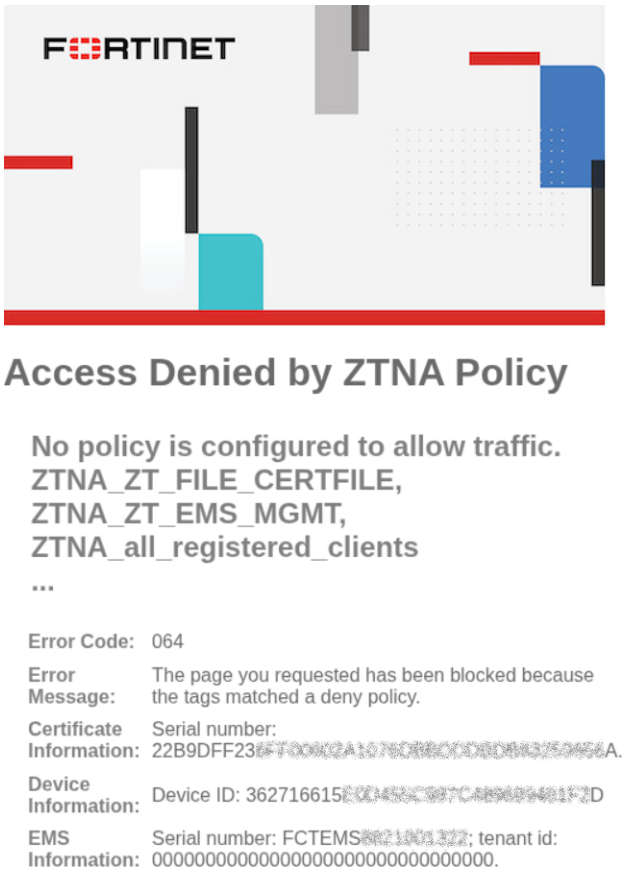
Tag(s) matched a deny policy

A FortiGate is configured with a deny policy when matching a specific security posture tag called ZTNA_ZT_FILE_CERTFILE. An *Access Denied by ZTNA policy* replacement message with *Error Code 064* is displayed:



Tag(s) matched one or more deny policy

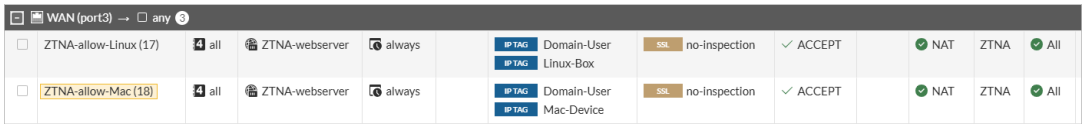
A FortiGate is configured with one or more deny policies, without any allow policies for ZTNA traffic. Client matches one of more of the deny policies based on its tags. An *Access Denied by ZTNA policy* replacement message with *Error Code 064* is displayed:



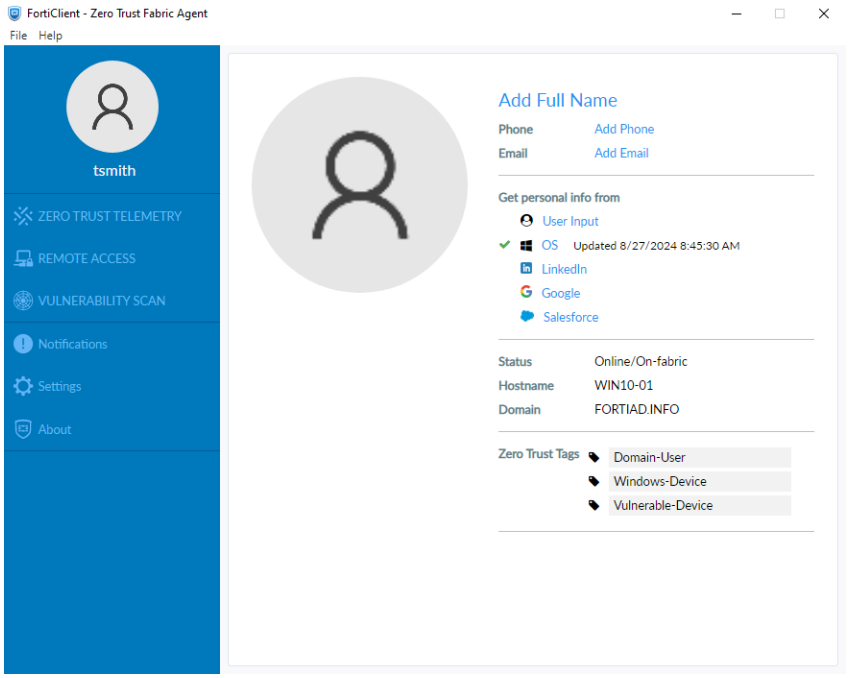
Client tags do not match any allowed ZTNA policy

One or more allow ZTNA policies are configured, with tags that are required for each policy. Client tags do not match the tag criteria in the allowed ZTNA policy, resulting in an implicit deny. An *Access Denied by ZTNA policy* replacement message with *Error Code 065* is displayed, along with the missing attributes/tags that would have matched a policy.

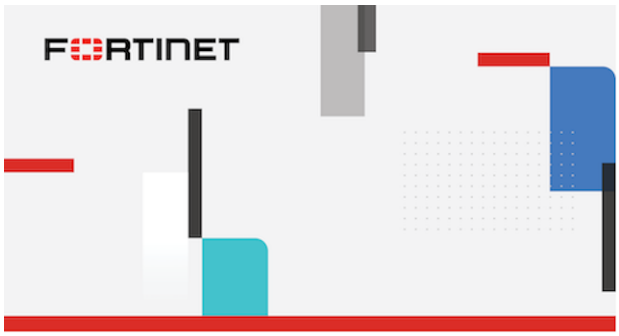
ZTNA policy configuration:



FortiClient tags:



Replacement Message:



Access Denied by ZTNA Policy

Missing attributes:
Detected Ubuntu Box,
Detected Mac Device

Error Code:	065
Error Message:	The page you requested has been blocked because the tags didn't match any policy.
Certificate Information:	Serial number: 18CC843F1[redacted]B.
Device Information:	Device ID: 9A016B5A6[redacted]A
EMS Information:	Serial number: FCTEMS[redacted]; tenant id: 00000000000000000000000000000000.
Request Time:	1749753906; 2025-06-12 11:45:06 PDT

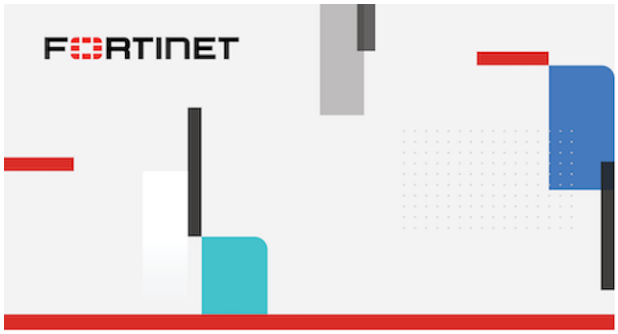
Log message:

```
1: date=2025-06-12 time=11:45:06 eventtime=1749753906319108035 tz="-0700" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=20444
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.0.3.10
dstport=9043 dstintf="root" dstintfrole="undefined" sessionid=1989 srcuuid="b458a65a-f759-51ea-
d7df-ef2e750026d1" dstuuid="617b9fe2-4753-51f0-781e-613b9b51f53e" service="tcp/9043"
proxyapptype="ztna-proxy" proto=6 action="deny" policyid=0 policytype="policy" appcat="unscanned"
duration=0 vip="ZTNA-webserver" vipincomingip=10.0.3.10 accessproxy="ZTNA-webserver"
clientdeviceid="9A016B5A6E914B42AD4168C066EB04CA" clientdevicemanageable="manageable"
clientdeviceems="FCTEMS8824006771" clientdevicetags="Vulnerable-Device/Windows-Device/all_
registered_clients/Domain-User" clientcert="yes" emsconnection="online" msg="Traffic denied
because failed to match a policy or proxy-policy" wanin=0 rcvdbyte=0 wanout=0 lanin=3720
sentbyte=3720 lanout=3452 fctuid="9A016B5A6E914B42AD4168C066EB04CA" crscore=30 craction=131072
crlevel="high"
```

Client tags match a negated tag in an allowed ZTNA policy

One or more allow ZTNA policies are configured, with negated tag(s) that are required for each policy. For example, the negated Vulnerable-Device, Mac-Device, and Linux-box tags are applied to a ZTNA policy:

The same client from Example 5 with a Vulnerable-Device tag attempts to connect to a ZTNA resource. An *Access Denied by ZTNA policy* replacement message with *Error Code 065* is displayed, along with the tag matching the negate tag on the allowed policy.



Access Denied by ZTNA Policy

ZTNA_Vulnerable-Device

Error Code:	065
Error Message:	The page you requested has been blocked because the tags didn't match any policy.
Certificate Information:	Serial number: 18CC843F1 B.
Device Information:	Device ID: 9A016B5A6 A
EMS Information:	Serial number: FCTEMS ; tenant id: 00000000000000000000000000000000.
Request Time:	1749754406; 2025-06-12 11:53:26 PDT

Log message:

```
1: date=2025-06-12 time=11:53:26 eventtime=1749754406771205939 tz="-0700" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.0.3.2 srcport=20560
srcintf="port3" srcintfrole="wan" dstcountry="Reserved" srccountry="Reserved" dstip=10.0.3.10
dstport=9043 dstintf="root" dstintfrole="undefined" sessionid=2095 srcuuid="b458a65a-f759-51ea-
d7df-ef2e750026d1" dstuuid="617b9fe2-4753-51f0-781e-613b9b51f53e" service="tcp/9043"
proxyapptype="ztna-proxy" proto=6 action="deny" policyid=0 policytype="policy" appcat="unscanned"
duration=0 vip="ZTNA-webserver" vipincomingip=10.0.3.10 accessproxy="ZTNA-webserver"
clientdeviceid="9A016B5A6E914B42AD4168C066EB04CA" clientdevicemanageable="manageable"
clientdeviceems="FCTEMS8824006771" clientdevicetags="Vulnerable-Device/Windows-Device/all_
registered_clients/Domain-User" clientcert="yes" emsconnection="online" msg="Traffic denied
because failed to match a policy or proxy-policy" wanin=0 rcvdbyte=0 wanout=0 lanin=3624
sentbyte=3624 lanout=3452 fctuid="9A016B5A6E914B42AD4168C066EB04CA" crscore=30 craction=131072
crlevel="high"
```

Policy and Objects

This section contains topics on configuring policies and traffic shaping:

- Policies on page 1841
- Address objects on page 2031
- Custom tags NEW on page 2089
- Protocol options on page 2101
- Traffic shaping on page 2108
- Internet Services on page 2178

Policies

The firewall policy is the axis around which most features of the FortiGate revolve. Many firewall settings end up relating to or being associated with the firewall policies and the traffic they govern. Any traffic going through a FortiGate has to be associated with a policy. These policies are essentially discrete compartmentalized sets of instructions that control the traffic flow going through the firewall. These instructions control where the traffic goes, how it is processed, if it is processed, and whether or not it is allowed to pass through the FortiGate.

When the firewall receives a connection packet, it analyzes the source address, destination address, and service (by port number). It also registers the incoming interface, the outgoing interface it needs to use, and the time of day. Using this information, the FortiGate firewall attempts to locate a security policy that matches the packet. If a policy matches the parameters, then the FortiGate takes the required action for that policy. If it is *Accept*, the traffic is allowed to proceed to the next step. If the action is *Deny* or a match cannot be found, the traffic is not allowed to proceed.

The two basic actions at the initial connection are either *Accept* or *Deny*:

- If the action is *Accept*, the policy permits communication sessions. There may be other packet processing instructions, such as requiring authentication to use the policy or restrictions on the source and destination of the traffic.
- If the action is *Deny*, the policy blocks communication sessions, and you can optionally log the denied traffic. If no security policy matches the traffic, the packets are dropped. A *Deny* security policy is needed when it is required to log the denied traffic, also called *violation traffic*.

One other action can be associated with the policy:

- *IPsec*: this is an *Accept* action that is specifically for IPsec VPNs.

Each field in a firewall policy that accepts multiple inputs, such as srcaddr and dstaddr, can accept as many inputs as there are unique objects created. The maximum number of objects depends on the model. See the Maximum Values Table for more details.

The following topics provide information on the available types of policies and configuration instructions:

- Firewall policy on page 1842
- NGFW policy on page 1868
- Local-in policy on page 1886
- DoS policy on page 1901
- Telemetry policy on page 1909
- Access control lists on page 1911
- Interface policies on page 1912

The following topics provide instructions on configuring policies:

- Source NAT on page 1913
- Destination NAT on page 1939
- Examples and policy actions on page 1965

Firewall policy

The firewall policy is the axis around which most of the other features of the FortiGate firewall revolve. A large portion of the settings in the firewall at some point will end up relating to or being associated with the firewall policies and the traffic that they govern. Any traffic going through a FortiGate unit has to be associated with a policy. These policies are essentially discrete compartmentalized sets of instructions that control the traffic flow going through the firewall. These instructions control where the traffic goes, how it's processed, if it's processed, and even whether or not it's allowed to pass through the FortiGate.

The following topics provide information on the firewall policy and configuration:

- Firewall policy parameters on page 1842
- Configurations in the GUI on page 1843
- Configurations in the CLI on page 1853
- Policy views on page 1858
- Policy match on page 1863
- Services on page 1863

Firewall policy parameters

For traffic to flow through the FortiGate firewall, there must be a policy that matches its parameters:

- Incoming interface(s)
- Outgoing interface(s)
- Source address(es)
- Destination address(es)
- User(s) identity
- Service(s)
- Schedule

Traffic parameters are checked against the configured policies for a match. If the parameters do not match any configured policies, the traffic is denied.

Traffic flow initiated from each direction requires a policy, that is, if sessions can be initiated from both directions, each direction requires a policy.

Just because packets can go from point A to point B on port X does not mean that the traffic can flow from point B to point A on port X. A policy must be configured for each direction.

When designing a policy, there is often reference to the traffic flow, but most communication is two-way so trying to determine the direction of the flow might be confusing. If traffic is HTTP web traffic, the user sends a request to the website, but most of the traffic flow will be coming from the website to the user or in both directions? For the purposes of determining the direction for a policy, the important factor is the direction of the initiating communication. The user is sending a request to the website, so this is the initial communication; the website is responding so the traffic is from the user's network to the Internet.

FortiOS does not perform a reverse-path check on reply traffic that matches an allowed session based on the IP tuple. The request traffic can be sent on one interface and the reply traffic could return on another interface.

Configurations in the GUI

Firewall policies can be created in the GUI by configuring the necessary parameters.

Schedule	The time frame that is applied to the policy. This can be something as simple as a time range that the sessions are allowed to start, such as between 8:00 AM and 5:00 PM. Something more complex like business hours that include a break for lunch and time of the session's initiation may need a schedule group because it will require multiple time ranges to make up the schedule.
Incoming interface (s)	This is the interface or interfaces by which the traffic is first connected to the FortiGate unit. The exception being traffic that the FortiGate generates itself. This is not limited to the physical Ethernet ports found on the device. The incoming interface can also be a logical or virtual interface such as a VPN tunnel, a Virtual WAN link, or a wireless interface. The FortiLink interface cannot be added into a firewall policy in the GUI. Instead, when you need to apply the FortiLink interface as a source, such as when you want to allow access to a syslog or SNMP server, create the firewall policy from the CLI.
Outgoing interface (s)	This is the interface or interfaces used by traffic leaving a port once it has been processed by the firewall. Similar to incoming interfaces, it is not limited to only physical interfaces.
Source address (es)	The addresses that a policy can receive traffic from can be wide open or tightly controlled. For a public web server that the world at large should be able to access, the best choice will be *all*. If the destination is a private web server that only the branch offices of a company should be able to access, or a list of internal computers that are the only ones allowed to access an external resource, then a group of preconfigured addresses is the better strategy

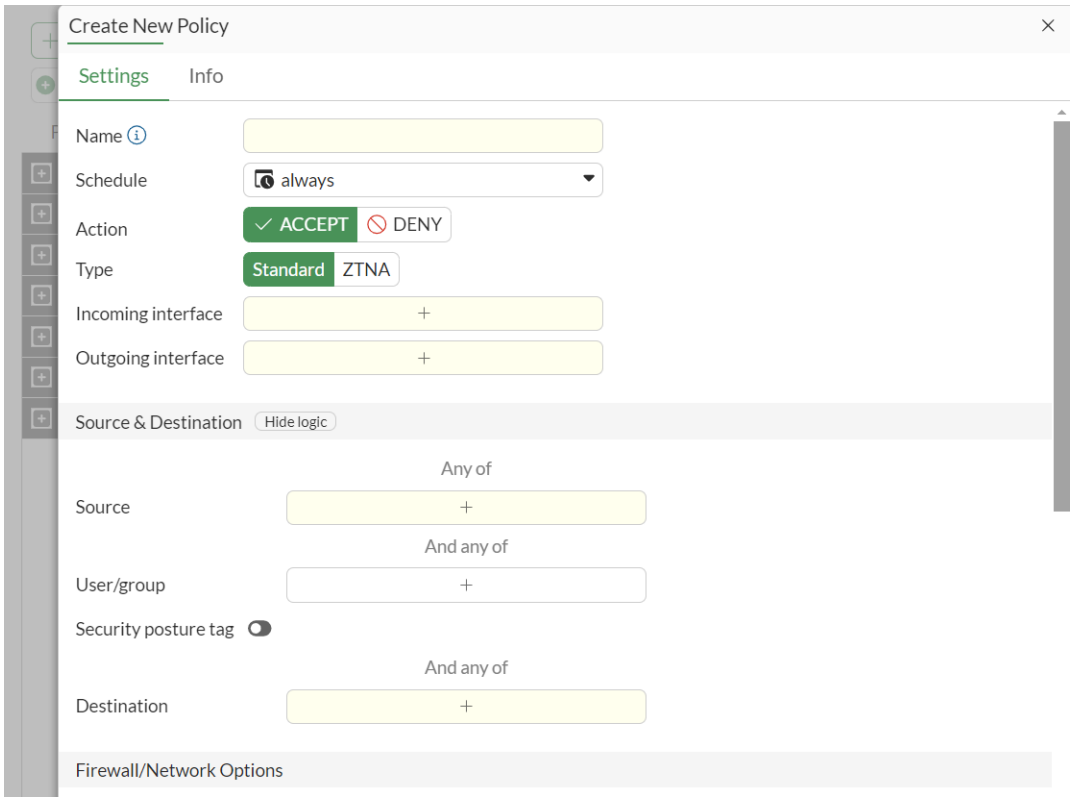
User/group	This parameter is based on a user identity that can be from a number of authentication authorities. It will be an account or group that has been set up in advance that can be selected from the drop down menu. The exception to this is the feature that allows the importing of LDAP Users. When the feature is used, a small wizard window will appear to guide the user through the setup. The caveat is that the LDAP server object in the *User & Authentication > LDAP Servers* section has to be already configured to allow the use of this import feature.
Destination address(es)	In the same way that the source address may need to be limited, the destination address can be used as a traffic filter. When the traffic is destined for internal resources, the specific address of the resource can be defined to better protect the other resources on the network. One of the specialized destination address options is to use a Virtual IP address. If the destination address doesn't need to be internal, you can define policies that are only for connecting to specific addresses on the Internet.
Internet service(s)	In this context, an Internet service is a combination of one or more addresses and one or more services associated with a service found on the Internet such as an update service for software.
Service	The services chosen represent the TCP/IP suite port numbers that will most commonly be used to transport the named protocols or groups of protocols. This is different than *Application Control* which looks more closely at the packets to determine the actual protocol used to create them. A case where either side can initiate the communication, like between two internal interfaces on the FortiGate unit, would be a more likely situation to require a policy for each direction.

Displaying logic between firewall policy objects in the GUI

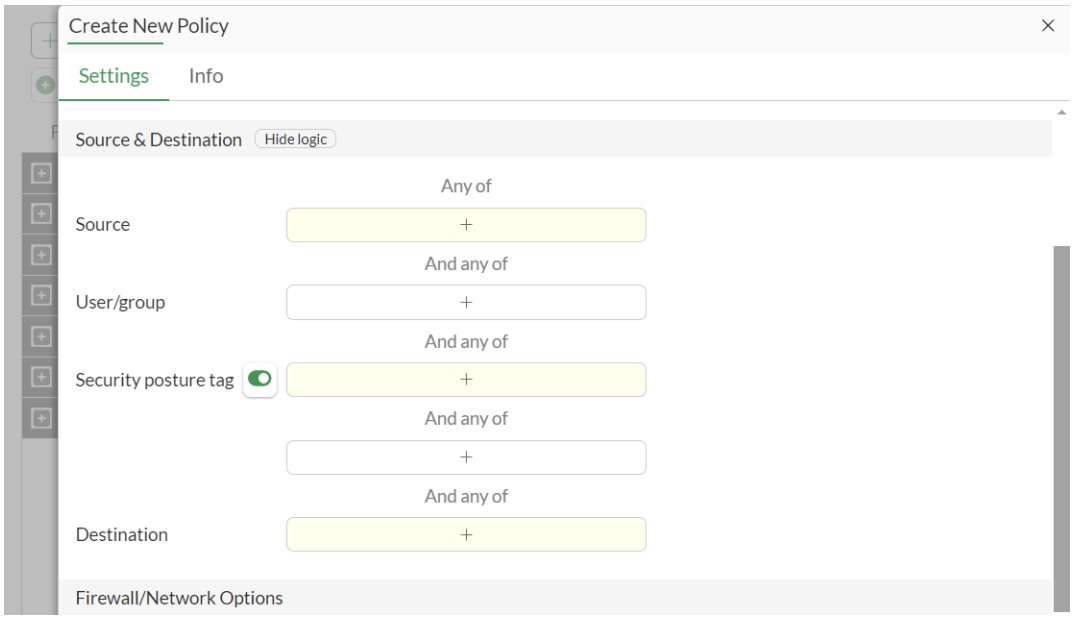
The FortiOS GUI can display the logical AND relationship in firewall policies between source and destination objects, including security posture tags when enabled, to help you configure firewall policies.

To display the logic in firewall policies in the GUI:

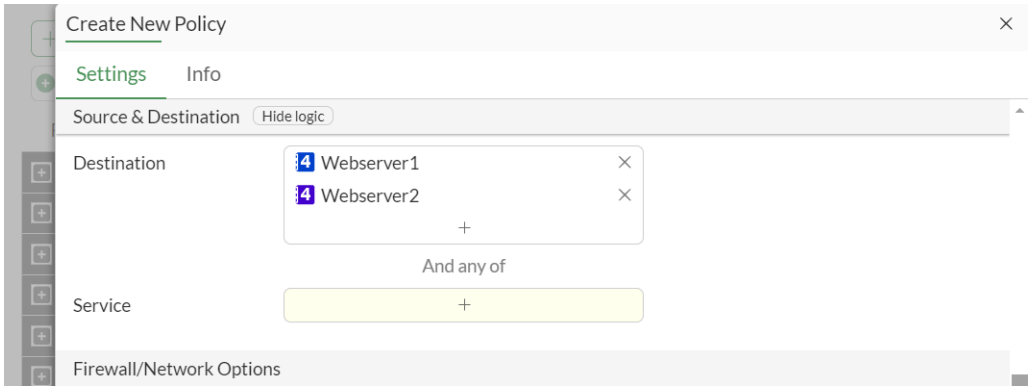
1. Go to *Policy & Objects > Firewall Policy*, and click *Create New*. The *Create New Policy* pane opens.
2. In the *Source & Destination* section, click *Show Logic*.
   - The labels *Any of* and *And any of* appear to clarify how the objects are used to match traffic. The *And any of* label indicates a logical AND relationship between the objects.



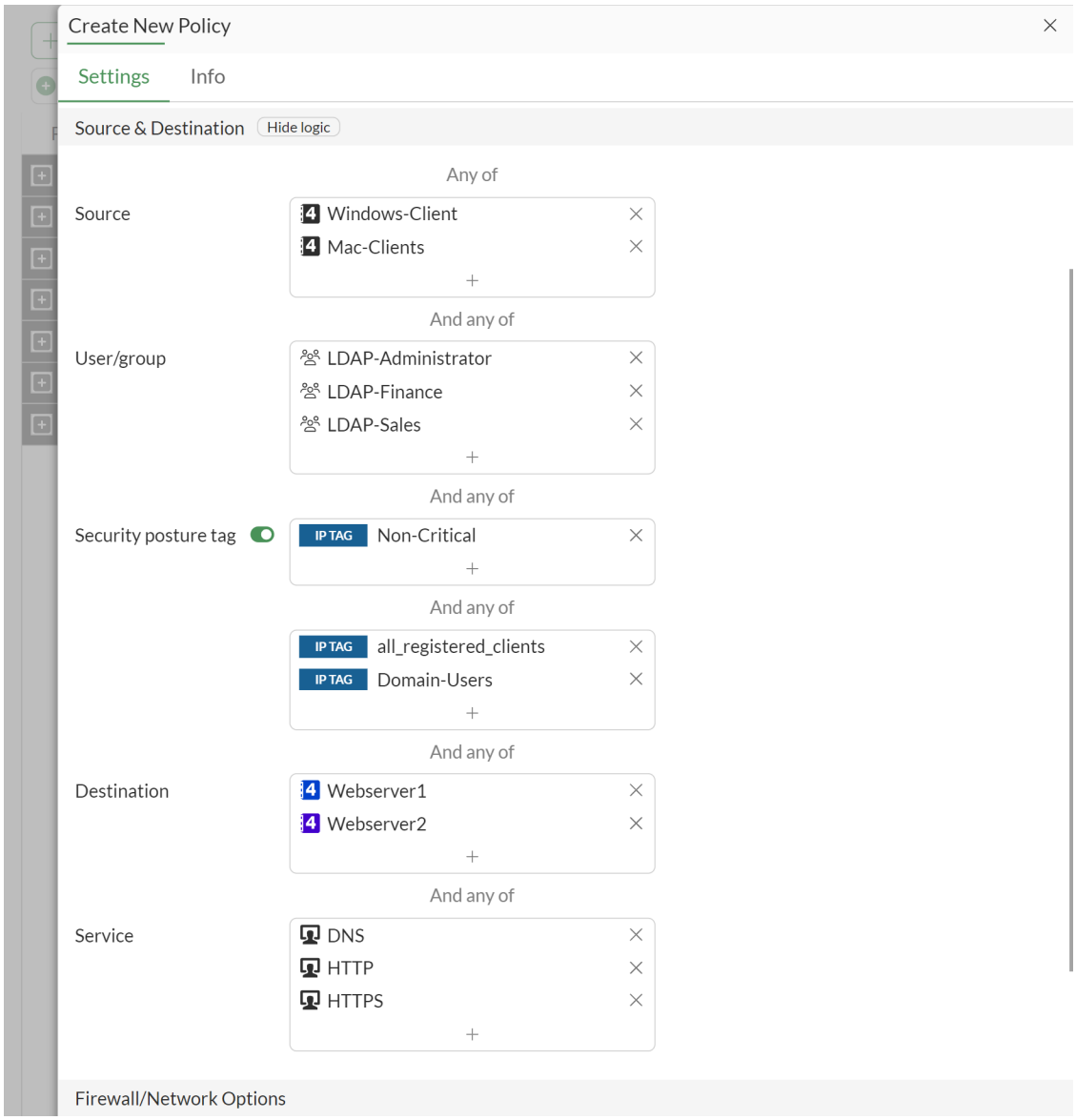
3. Enable *Security posture tag*:
   - The labels *And any of* appear, indicating a logical AND relationship between the two security posture tag objects.
   - The first security posture tag group is required when *Security posture tag* is enabled and is considered the primary tag group.
   - A second security posture tag group is optional when *Security posture tag* is enabled and is considered the secondary tag group.
   - Within each of the tag groups, there is a logical OR relationship between the tags when multiple tags are used.



4. In the *Destination* list, select one or more destinations to display *And any of* between the *Destination* and *Service* fields.



Following is the logic of using the mandatory and optional fields in the *Source & Destination* section, based on the following configurations:



Traffic must match all criteria below:

- Any *Source*: either *Windows-Client* OR *Mac-Clients*
- Any *User/group*: either *LDAP-Administrator*, *LDAP-Finance*, OR *LDAP-Sales*
- The primary *Security posture tag*: *Non-Critical*
- Any secondary security posture tags: either *all_registered_clients* OR *Domain-Users*
- Any *Destination*: either *Webserver1* OR *Webserver2*
- Any *Service*: Either *DNS*, *HTTP*, OR *HTTPS*

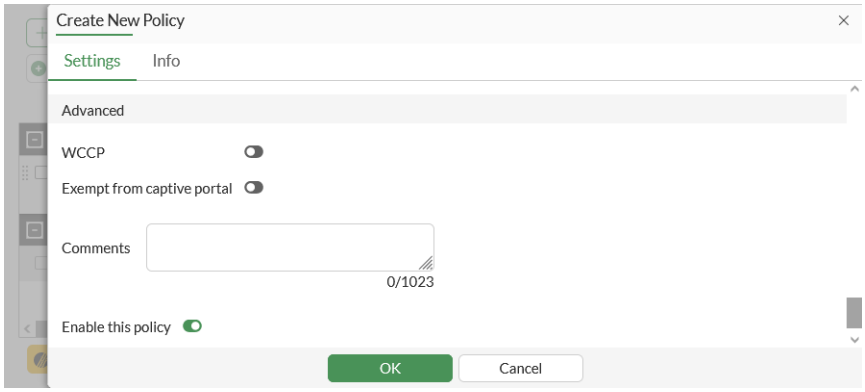
Enabling advanced policy options in the GUI

Advanced policy options can be enabled so that you can configure the options in the GUI.

To enable advanced policy options:

```
config system settings
    set gui-advanced-policy enable
end
```

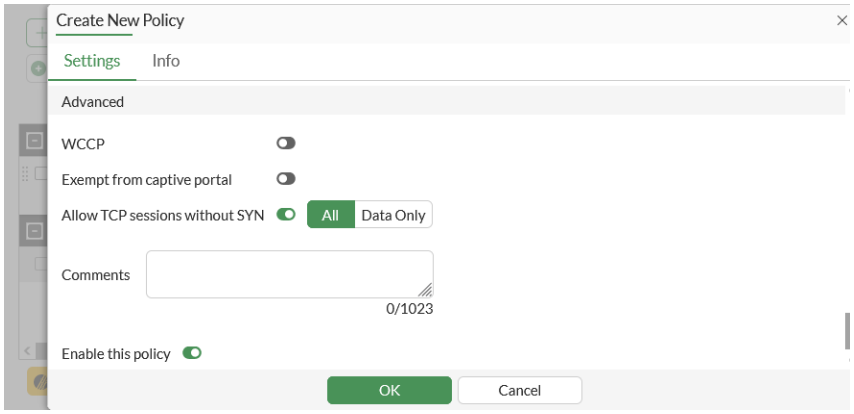
Advanced policy options are now available when creating or editing a policy in the GUI:



To enable configuring TCP sessions without SYN:

```
config system settings
    set tcp-session-without-syn enable
end
```

TCP sessions without SYN can now be configured when creating or editing a policy in the GUI:



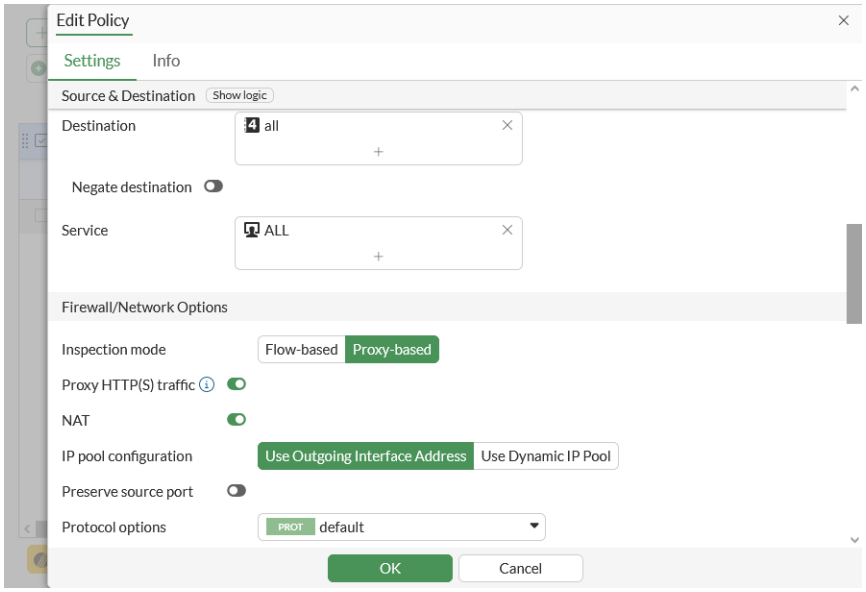
Inspection mode per policy

When configuring a firewall policy, you can select a *Flow-based* or *Proxy-basedInspection Mode*. The default setting is *Flow-based*.

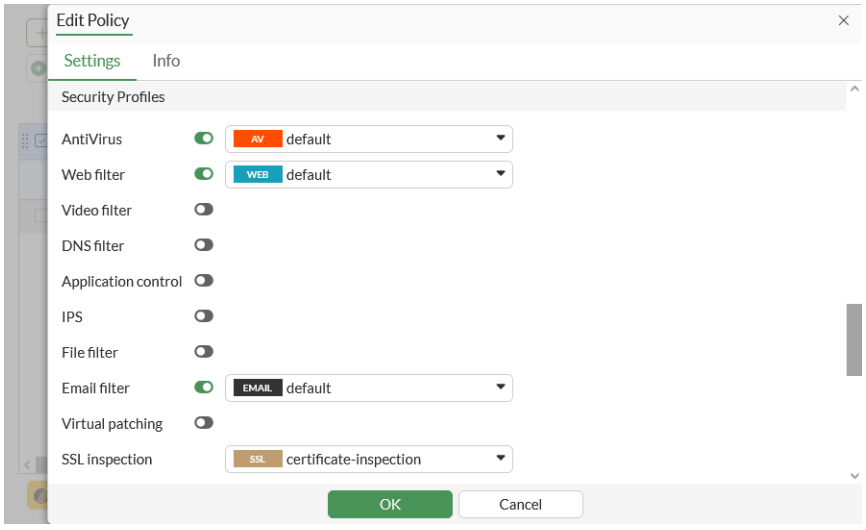
To configure inspection mode in a policy:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Create a new policy, or edit an existing policy.

3. Configure the policy as needed.
   a. If you change the *Inspection Mode* to *Proxy-based*, the *Proxy HTTP(S) traffic* option displays.



   b. In the *Security Profiles* section, if no security profiles are enabled, the default *SSL Inspection* is *no-inspection*. If you enable any security profile, the *SSL Inspection* should be changed *certificate-inspection*, *deep-inspection*, or a custom inspection profile.



To see the inspection mode changes using the CLI:

```
config firewall policy
    edit 1
        set srcintf "wan2"
        set dstintf "wan1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
```

```
        set service "ALL"
        set inspection-mode proxy
        set nat enable
    next
end
```

To see the HTTP and SSH policy redirect settings when inspection mode is set to proxy using the CLI:

```
config firewall policy
    edit 1
        set srcintf "wan2"
        set dstintf "wan1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set inspection-mode proxy
        set http-policy-redirect enable
        set ssh-policy-redirect enable
        set nat enable
    next
end
```

To see the default SSL-SSH policy set to no inspection using the CLI:

```
config firewall policy
    edit 1
       show fu | grep ssl-ssh-profile
        set ssl-ssh-profile "no-inspection"
    next
end
```

Add Policy change summary and Policy expiration to Workflow Management

Two options, *Policy change summary* and *Policy expiration*, are included in *Workflow Management*. *Policy change summary* enforces an audit trail for changes to firewall policies. *Policy expiration* allows administrators to set a date for the firewall policy to be disabled.

There are three states for the *Policy change summary*:

- *Disable*: users will not be prompted to add a summary when editing a policy.
- *Required*: the *Policy change summary* will be enabled and will require users to add a summary when editing or creating a firewall policy.
- *Optional*: the *Policy change summary* will be enabled but users can leave the summary empty, if preferred, when editing or creating a firewall policy.

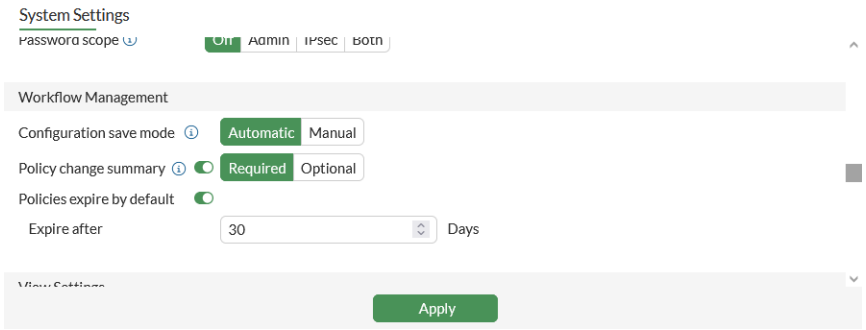
There are three states for *Policy expiration*:

- *Disable*: the firewall policy will not expire. This is the default setting for *Policy expiration*.
- *Default*: the firewall policy will expire after the default number of days.
- *Specify*: the firewall policy will expire at a set date and time.

The default value for *Policy expiration* is 30 days. This number can be changed in the CLI or in *System > Settings > Preference* in the GUI to any value between zero and 365 days. If the default value is set to zero, the *Default* state will disable the *Policy expiration*.

To configure the firewall policy change summary and default expiration in the GUI:

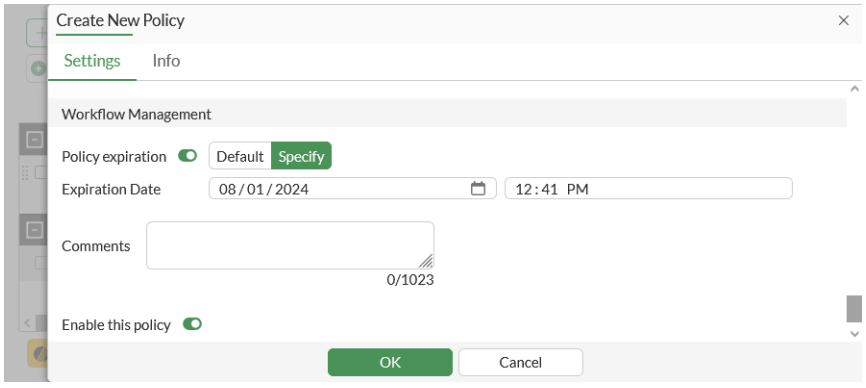
1. Go to *System > Feature Visibility*.
2. Enable *Workflow Management*.
3. Click *Apply*.
4. Go to *System > Settings > Preference*.
5. In the *Workflow Management* section, set *Policy change summary* to *Required*. *Policies expire by default* is enabled by default with an *Expire after* value of *30*.



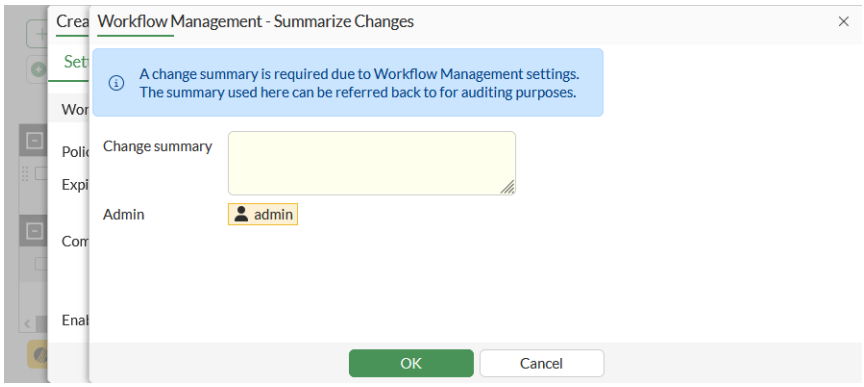
6. Click *Apply*.

To configure firewall policy expiration in the GUI:

1. Go to *Policy & Objects > Firewall Policy* and click *Create New*.
2. Name the policy and configure the necessary parameters.
3. Set *Policy expiration* to *Specify*.



4. Select the date and time for the policy to expire from the *Expiration date* fields.
5. Click *OK*. The *Workflow Management - Summarize Changes* pane opens.



6. In the *Change summary* field, enter details about the changes made to the policy. These details can be referred to later for auditing purposes.
7. Click *OK*.

To configure the firewall policy change summary in the CLI:

```
config system settings
    set gui-enforce-change-summary {disable | require | optional}
end
```

To configure the policy expiration default value in the CLI:

```
config system settings
    set default-policy-expiry-days <integer>
end
```

To configure firewall policy expiration in the CLI:

```
config firewall policy
    edit <id>
        set policy-expiry {enable | disable}
        set policy-expiry-date <YYYY-MM-DD HH:MM:SS>
    next
end
```

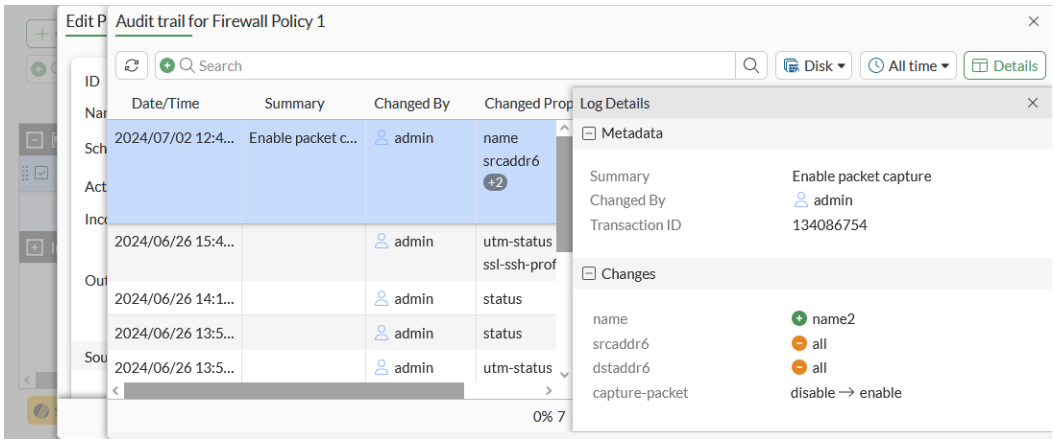
Policy change summaries are used to track changes made to a firewall policy. The *Audit Trail* allow users to review the policy change summaries, including the date and time of the change and which user made the change.

The *Audit Trail* is only supported by FortiGate models with disk logging.

To review the audit trail in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Select the policy you want to review and click *Edit*.

3. In the right-side banner (or on the *Info* tab if shown), click *Audit Trail*. The *Audit trail for Firewall Policy* pane opens and displays the policy change summaries for the selected policy.



4. Select an entry to review the details of the change made.
5. When you are done reviewing the *Audit Trail*, click *Close*.
6. Click *Cancel* to exit the *Edit Policy* page.

Configurations in the CLI

Firewall policies can be created in the CLI by configuring the necessary parameters. See Configurations in the GUI on page 1843 for more information on the various parameters.

Parameter	Definition
srcintf	Incoming (ingress) interface.
dstintf	Outgoing (egress) interface.
srcaddr	Source IPv4 address and address group names.
dstaddr	Destination IPv4 address and address group names.
internet-service	Enable/disable use of Internet Services for this policy. If enabled, destination address and service are not used.
schedule	Schedule name.
service	Service and service group names.
anti-replay	Enable/disable checking of TCP flags per policy.
match-vip	Enable/disable matching of VIPs when used in a policy with a deny action.
auto-asic-offload	Enable/disable hardware acceleration. Available on select FortiGate models with Secure Processing Unit (SPU) hardware only.
tcp-mss-sender	Sender TCP maximum segment size (MSS).
tcp-mss-receiver	Receiver TCP maximum segment size (MSS).
session-ttl	Time-to-live (TTL) in seconds for session accepted by this policy.

Firewall anti-replay option per policy

When the global anti-replay option is disabled, the FortiGate does not check TCP flags in packets. The per policy anti-replay option overrides the global setting. This allows you to control whether or not TCP flags are checked per policy.

To enable the anti-replay option so TCP flags are checked using the CLI:

```
config firewall policy
    edit 1
        set name "policyid-1"
        set srcintf "wan2"
        set dstintf "wan1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set anti-replay enable
        set logtraffic all
        set nat enable
    next
end
```

If the source IP address is either on the same network as or on a different subnet from the firewall's interface that will do the traffic redirection:

- If `allow-traffic-redirect` is enabled, then the traffic will be redirected without the need for a policy, based only on the routing decision.
- If `allow-traffic-redirect` is disabled, then the traffic must be matched by an IPv4 policy before being forwarded over the same interface that it entered on. If no IPv4 policy matches the traffic, then it will match the implicit deny policy and will be dropped.

```
config system global
    set allow-traffic-redirect {enable | disable}
end
```

When an IPv4 policy is needed to forward the traffic over the same interface that it came from, then `anti-replay` must be disabled for TCP traffic so that the traffic will not be dropped as replayed traffic.

Starting in FortiOS 7.6.5, `allow-traffic-redirect` is disabled by default. After upgrading, the setting will be disabled, even if it was enabled before the upgrade.

Deny matching with a policy with a virtual IP applied

Preventing hosts with specific source addresses from accessing a server behind the FortiGate may be required in some cases. For this scenario, you should have previously configured a firewall policy with a virtual IP (VIP) object applied to it to allow such access. See Destination NAT on page 1939 for details.

When denying traffic destined for a typical firewall policy without a VIP applied, you would simply configure a new firewall policy with an action of `deny` and with specific source addresses above the firewall policy that you

want to prevent these hosts from accessing. However, the FortiGate matches firewall policies with VIPs applied differently than typical firewall policies. Policies with VIPs applied have priority over typical firewall policies.

Therefore, to block specific source traffic destined for a firewall policy specified with an action of `accept` and with a VIP applied, you should configure `set match-vip enable` on the firewall policy with a `deny` action that has been configured to match traffic before the firewall policy with the VIP applied. By default, new `deny` action firewall policies have `match-vip` enabled.

If the policy action is set to `accept`, `match-vip` cannot be enabled.

To block VIP traffic in a deny policy:

```
config firewall policy
    edit 1
        set name "deny-policy-1"
        set srcintf "wan1"
        set dstintf "lan1"
        set srcaddr "src-hosts-to-deny-access"
        set dstaddr "all"
        set action "deny"
        set schedule "always"
        set service "all"
        set match-vip enable
    next
    edit 2
        set name "vip-policy-1"
        set srcintf "wan1"
        set dstintf "lan1"
        set srcaddr "all"
        set dstaddr "vip-object-1"
        set action "accept"
        set schedule "always"
        set service "ALL"
    next
end
```

Alternatively, to block access to a firewall policy with a VIP applied, you can configure a new VIP object configured with `set src-filter <range>`. Configure a new firewall policy with a `deny` action and with this new VIP applied, and then configure this policy to match traffic before the firewall policy with the same VIP applied with an action of `accept`. In this case, the firewall policy can simply have `set match-vip disable`.

To specify a VIP with source addresses specified with a deny policy:

```
config firewall vip
    edit “vip-with-srcaddr-to-deny”
        set extip "10.1.100.199"
        set extintf "wan1"
        set mappedip "172.16.200.55"
        set src-filter "1.1.1.1/24"
    next
end
```

```
config firewall policy
    edit 3
        set name "deny-policy-3"
        set srcintf "wan1"
        set dstintf "lan1"
        set srcaddr "all"
        set dstaddr "vip-with-srcaddr-to-deny"
        set action "deny"
        set match-vip disable
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set name "vip-policy-1"
        set srcintf "wan1"
        set dstintf "lan1"
        set srcaddr "all"
        set dstaddr "vip-object-1"
        set action "accept"
        set match-vip disable
        set schedule "always"
        set service "ALL"
    next
end
```

Hardware acceleration

Hardware acceleration is supported on select FortiGate devices and is enabled by default on all firewall policies to ensure optimal performance when processing network traffic traversing the FortiGate. See the Hardware Acceleration Reference Manual for details.

Typically, hardware acceleration on a specific firewall policy is disabled for one of two purposes:

- To allow CLI commands such as the packet sniffer and debug flow to display all traffic matching the policy since traffic offloaded by SPU hardware on a FortiGate device is not visible by those CLI tools.
- To troubleshoot any possible issues arising by using hardware acceleration.

To disable hardware acceleration in an IPv4 firewall policy:

```
config firewall policy
    edit 1
        set auto-asic-offload disable
    next
end
```

To disable hardware acceleration in an IPv6 firewall policy:

```
config firewall policy6
    edit 1
        set auto-asic-offload disable
```

```
    next
end
```

To disable hardware acceleration in a multicast firewall policy:

```
config firewall multicast-policy
    edit 1
        set auto-asic-offload disable
    next
end
```

TCP Maximum Segment Size (MSS)

The TCP maximum segment size (MSS) is the maximum amount of data that can be sent in a TCP segment. The MSS is the MTU size of the interface minus the 20 byte IP header and 20 byte TCP header. By reducing the TCP MSS, you can effectively reduce the MTU size of the packet.

The TCP MSS can be configured in a firewall policy, or directly on an interface. See Interface MTU packet size on page 232 for details on configuring TCP MSS directly on an interface.

To configure TCP MSS in a firewall policy:

```
config firewall policy
    edit <policy ID>
        set srcintf "internal"
        set dstintf "wan1"
        set srcaddr "10.10.10.6"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set tcp-mss-sender 1448
        set tcp-mss-receiver 1448
    next
end
```

Adjusting session time-to-live (TTL)

A session is a communication channel between two devices or applications across the network. Sessions allow FortiOS to inspect and act on a sequential group of packets in a session all at once instead of inspecting each packet individually. Each session has an entry in the session table that includes important information about the session.

The session time-to-live (TTL) parameter determines how long a session of a particular protocol such as TCP, UDP, or ICMP remains in the session table. To ensure proper operation of some devices or applications, the session TTL parameter may need to be increased or decreased to allow sessions to remain active in the session table for a longer or shorter duration, respectively.

To configure a modified session TTL in a firewall policy:

```
config firewall policy
    edit <policy ID>
        set srcintf "internal"
        set dstintf "wan1"
        set srcaddr "10.10.10.6"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set session-ttl 1800
    next
end
```

The session TTL can be set to zero or `never` to ensure a session never times out. See No session timeout on page 1992 for details.

Session TTL should only be set to zero or `never` after careful consideration of:

- The connected device's or application's requirements for sessions to always stay alive
- The expectation that a connected device or application will use the same session determined by traffic using a fixed source port, fixed destination port, fixed source IP address, and fixed destination IP address.

> ⚠ When session TTL is set to zero or `never`, then sessions will not be cleared from the session table or expire after a specified time unless the CLI commands `diagnose system session filter <filter>` and `diagnose system session clear` are used.
>
> If this setting is used in the case when traffic through a firewall policy can generate numerous unique sessions, then this may have unintended consequences to the FortiGate's memory usage and performance due to the session table constantly growing and not clearing out idle sessions.

To disable session TTL in a firewall policy:

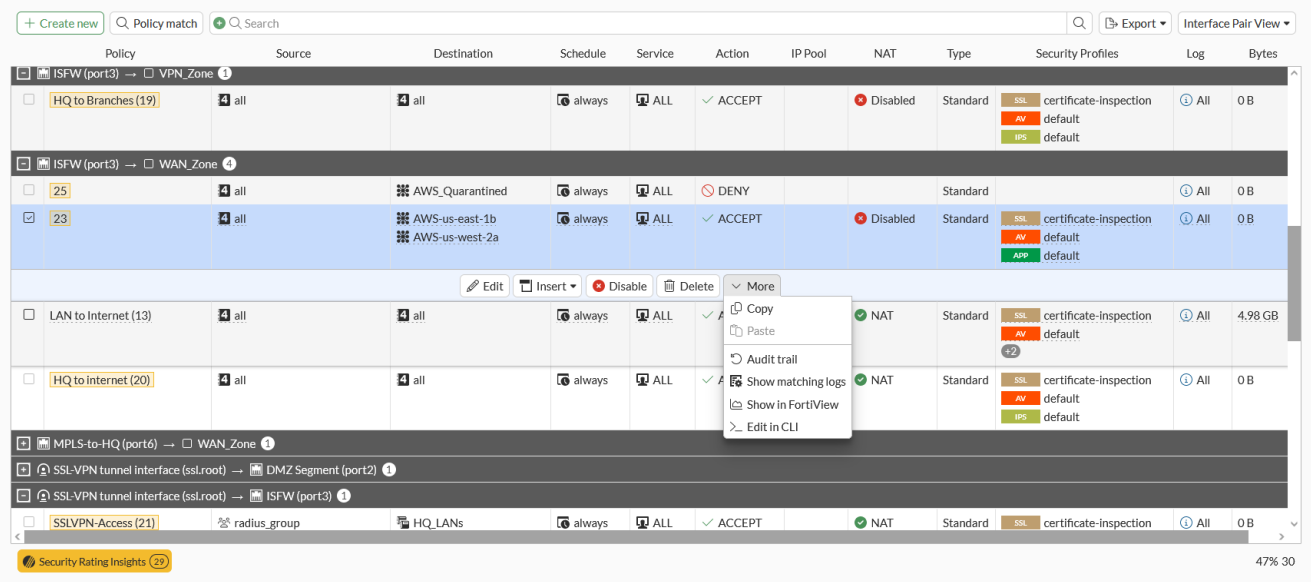
```
config firewall policy
    edit <policy ID>
        set srcintf "internal"
        set dstintf "wan1"
        set srcaddr "10.10.10.6"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set session-ttl never
    next
end
```

Policy views

In *Policy & Objects* policy list pages, there are three policy views: *Interface Pair View*, *Sequence Grouping View*, and *By Sequence*. For large policy tables (thousands of policies), the *By Sequence* view will load the fastest.

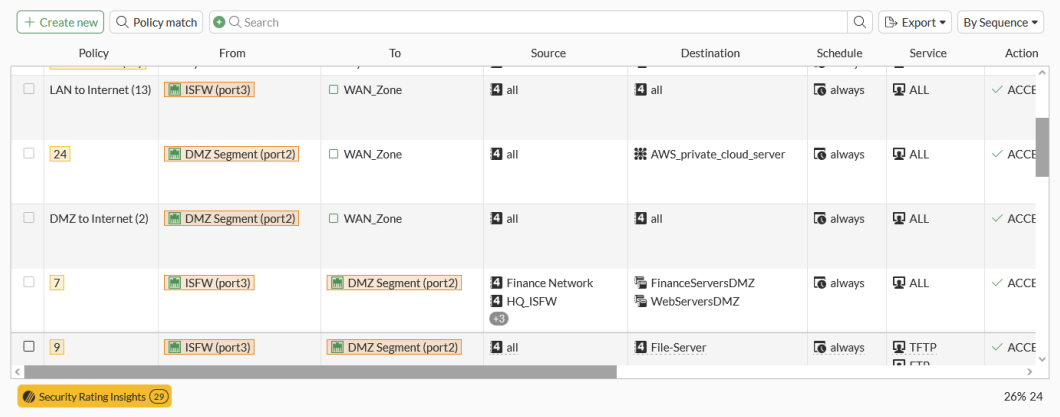
Interface Pair View displays the policies in the order that they are checked for matching traffic, grouped by the pairs of incoming and outgoing interfaces in collapsible sections. The *Interface Pair View* can be used when a policy is configured with multiple interfaces. Right-click on any pair name and click *Expand All* or *Collapse All* to expand or collapse all of the pairs.

When a policy is inserted, the Incoming and Outgoing interfaces are automatically filled in, and you can confirm the location of the new policy in the right-side gutter before clicking *OK* to insert the policy.



When a single policy is selected, an inline menu opens below the row. The *More* dropdown menu includes the same expanded list of options that are available in the right-click menu. When multiple policies are selected, the top menu bar changes to show buttons that are applicable to the multiple selections and you can click View selected to show only the selected policies on the page.

By Sequence displays policies in the order that they are checked for matching traffic without any grouping.

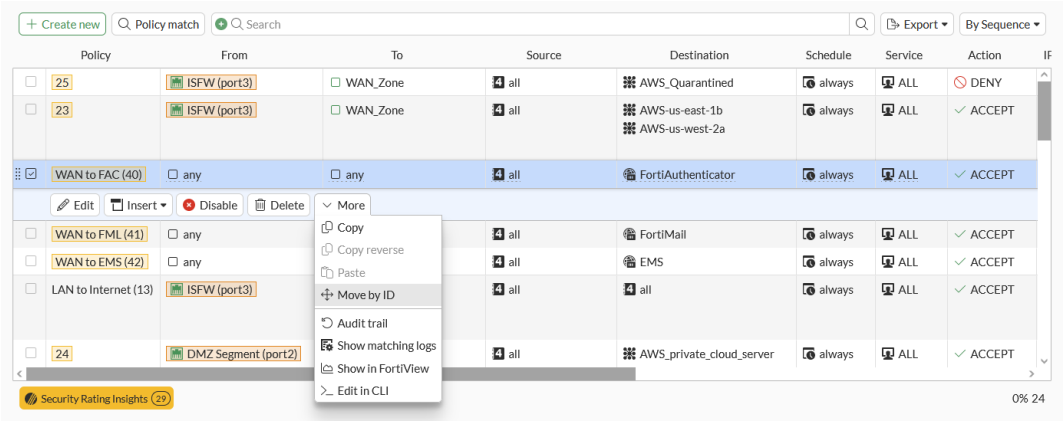


Policies can then be moved by their policy ID before or after another specified policy ID.

> Moving policies by ID is only available when viewing the *Firewall Policy* page in *By Sequence* or *Sequence Grouping View*.

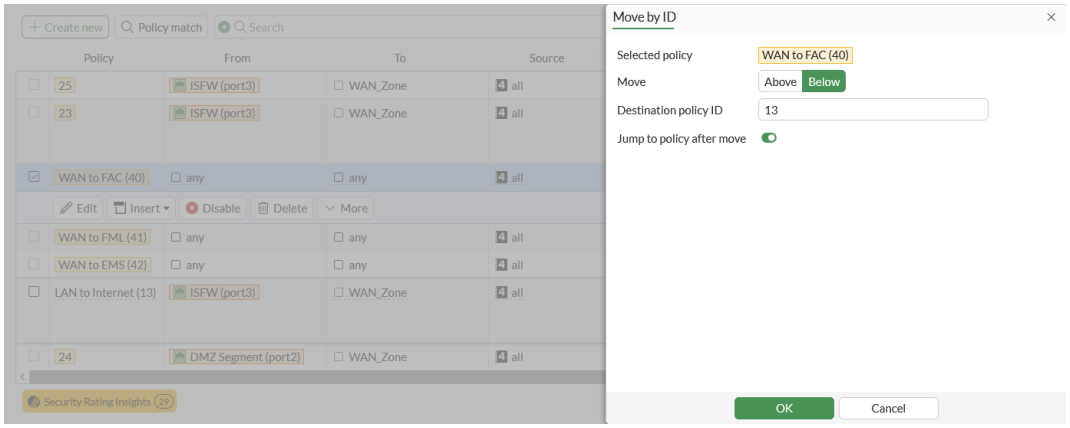
To move a policy by policy ID:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Select the policy you want to move.
3. Select *More* > *Move by ID*.



The *Move by ID* pane is displayed.

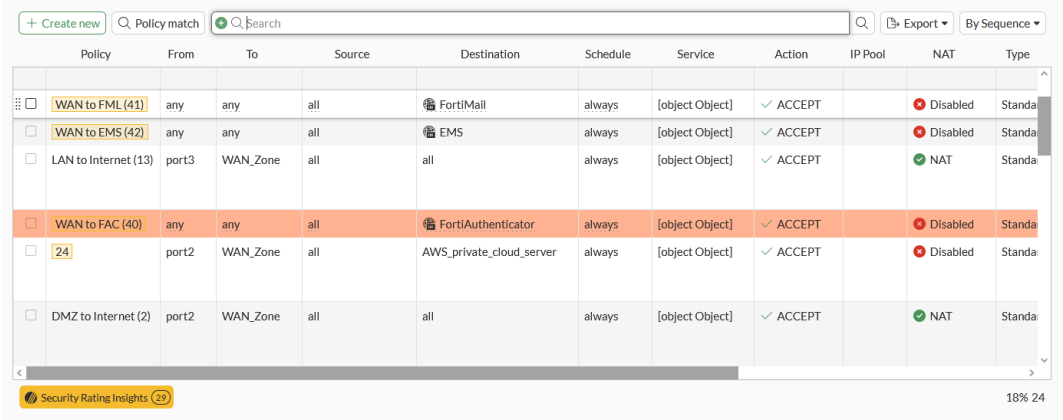
4. Define the new location of the policy:
   a. Select whether the policy should be moved *Above* or *Below* the policy ID you will define in the next step.
   b. In the *Destination policy ID* field, enter the ID of the destination policy or select it from the dropdown menu.



5. If you do not want to automatically view the new location of the policy, disable *Jump to policy after move*. This feature is enabled by default.
6. Click *OK*.

> If *Workflow Management* is enabled in *System* > *Feature Visibility*, the *Workflow Management - Summarize Changes* pane is displayed. Enter a *Change summary* then click *OK* to continue.

The policy will be moved to the new location.



Filtering on policy list statistics

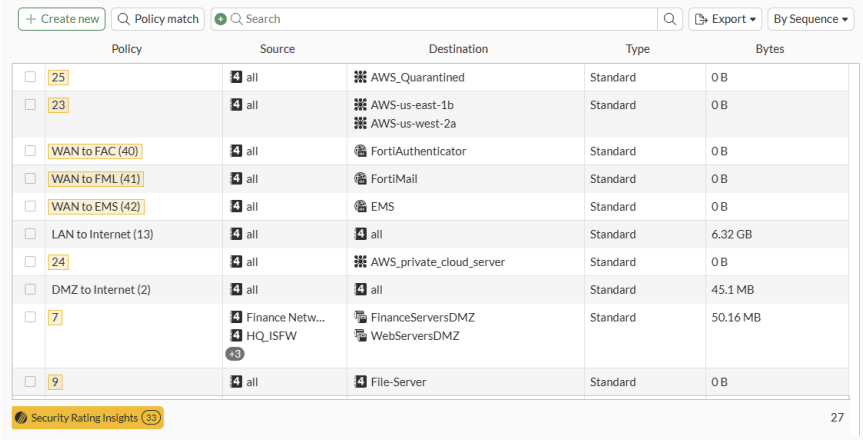
Statistics filtering on mutable policy lists allows users to refine policies based on key metrics such as *Bytes*, *Packets*, *Hit Count*, and *Last Used*. Statistics filters have been implemented in the *Firewall Policy*, *SSL Inspection & Authentication Policy*, and *Security Policy* pages. However, the available filters differ depending on the page.

This provides more precise control for identifying high-impact or frequently used policies, improving efficiency in policy management and troubleshooting.

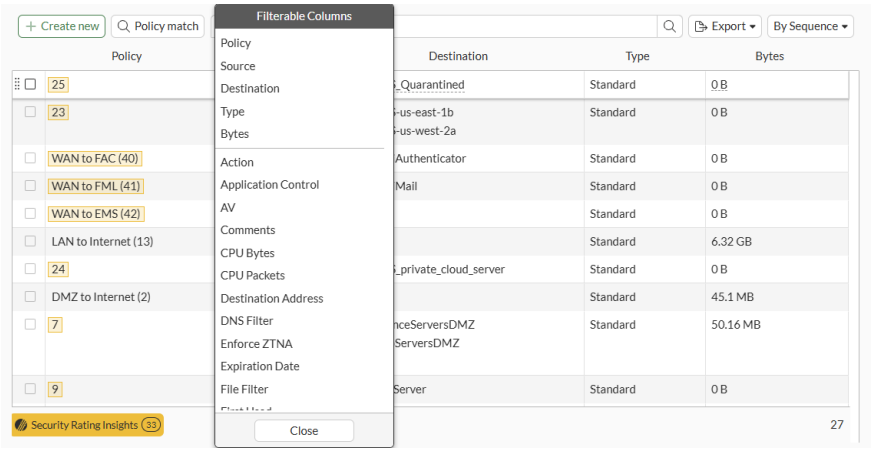
The following example demonstrates filtering the firewall policies table so that only policies with traffic greater than or equal to 40 MB are displayed.

To filter the policy list based on policy statistics:

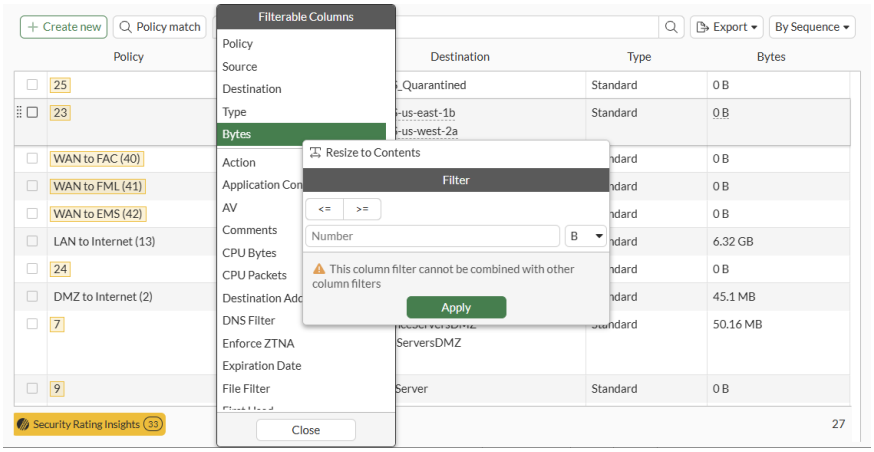
1. Go to *Policy & Objects* > *Firewall Policy*.



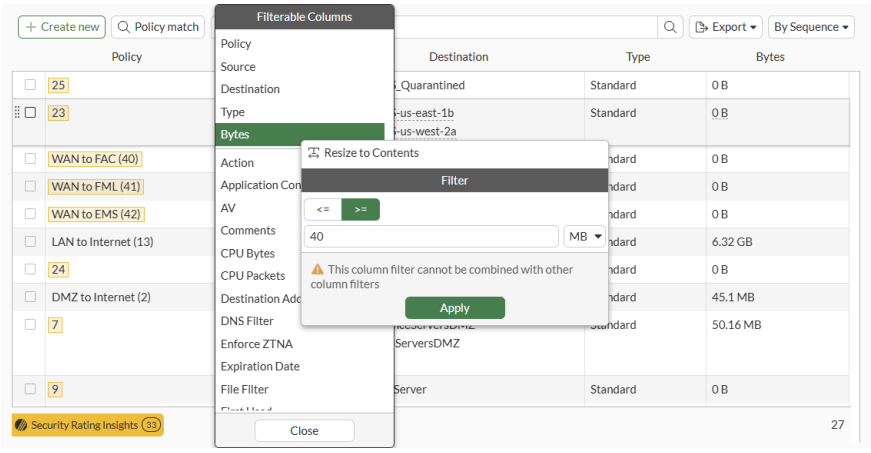
2. In the *Search* field, select the *+*. The *Filterable Columns* list is displayed.



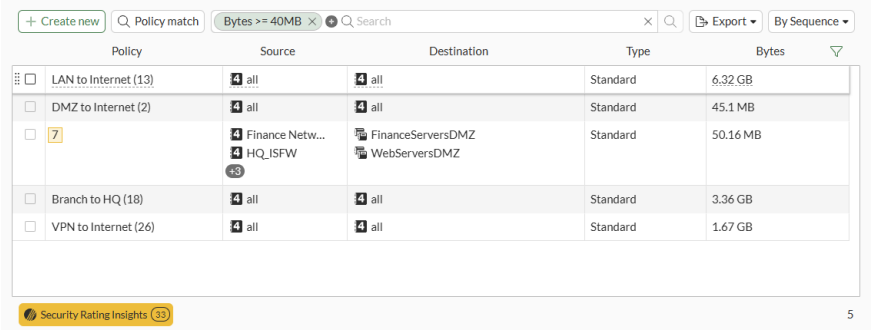
3. Select *Bytes*. A filter definitions dialog is displayed.



4. Define the filter parameters:
   a. Select >= to define the parameter range.
   b. Enter 40 into the value field.
   c. Select the unit dropdown list and select *MB*.



5. Click *Apply*. The filter parameters will be applied to the table.



Policy match

Firewall policy match is based on the `Source_interfaces/Protocol/Source_Address/Destination_Address` that matches the `source-port` and `dst-port` of the protocol. Use this tool to find out which policy matches specific traffic from a number of policies. After completing the match, the matching firewall policy is highlighted on the policy list page.

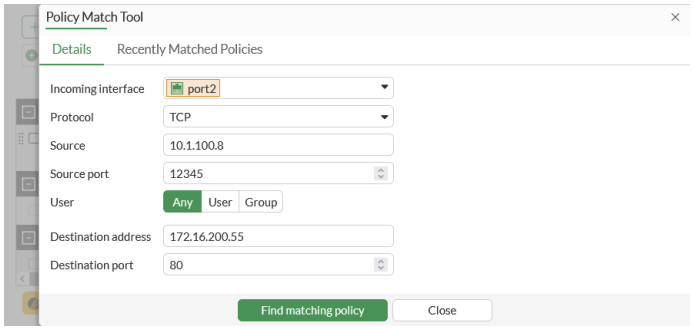
The *Policy match* tool has the following requirements:

- Transparent mode does not support policy match function.
- When executing the policy match, you need to confirm whether the relevant route required for the policy work already exists.

Sample configuration

This example uses the TCP protocol to show how policy match works:

1. On a *Policy & Objects* policy list page, click *Policy match* and enter the traffic parameters.



2. Click *Find matching policy* to display the policy match results.

Services

Services represent typical traffic types and application packets that pass through the FortiGate. Services include the service protocol type (TCP, UDP, ICMP, and so on), address, category, and logical destination port. Services can then be applied in a firewall policy to represent the TCP/IP suite port numbers that will most commonly be used to transport the named protocols or groups of protocols. Likewise, security profiles use service definitions to match session types.

The following services are available:

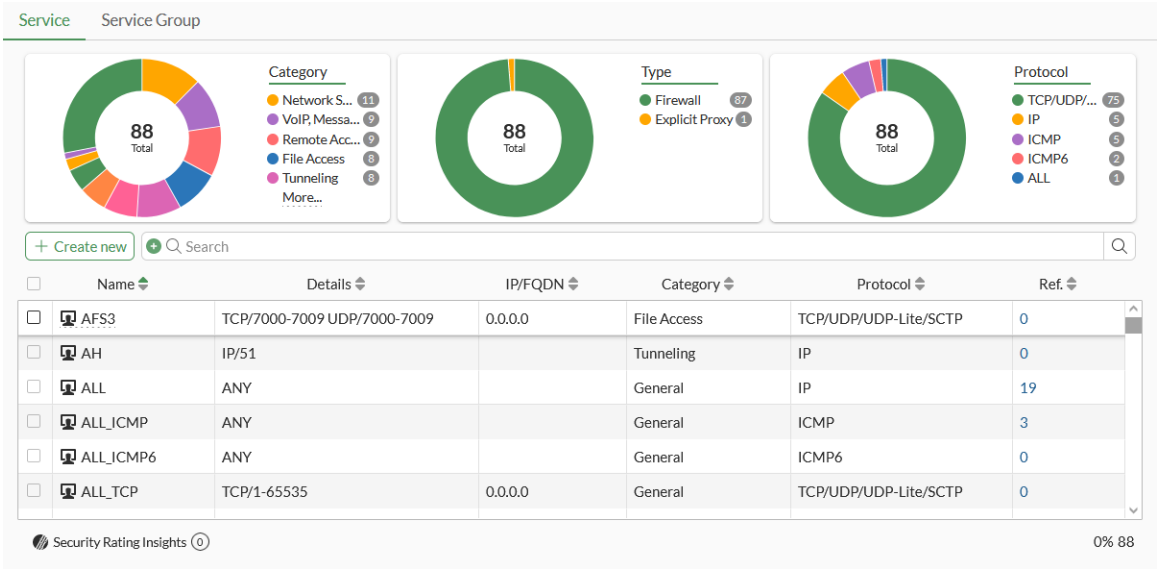
- Predefined services on page 1864
- Custom services on page 1866
- Service groups on page 1867

Predefined services

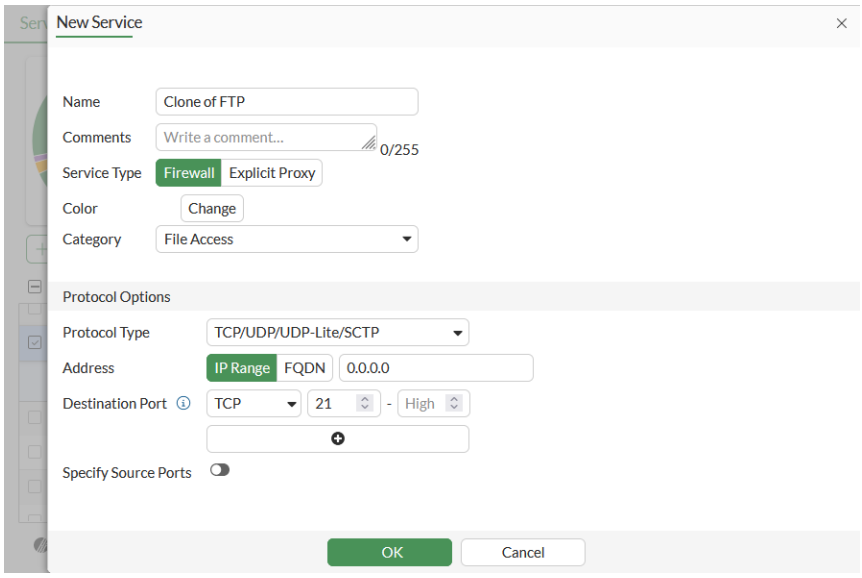
Firewall policies can be configured with default, predefined services that have been created for common traffic types. Predefined services can be edited, cloned, and deleted from the *Policy & Objects > Services* list. Cloning a services allows you to create a copy of the service parameters and edit it to create a similar service while still maintaining the existing service.

To clone a service:

1. Go to *Policy & Objects > Services*.
2. Go to the *Services* tab.



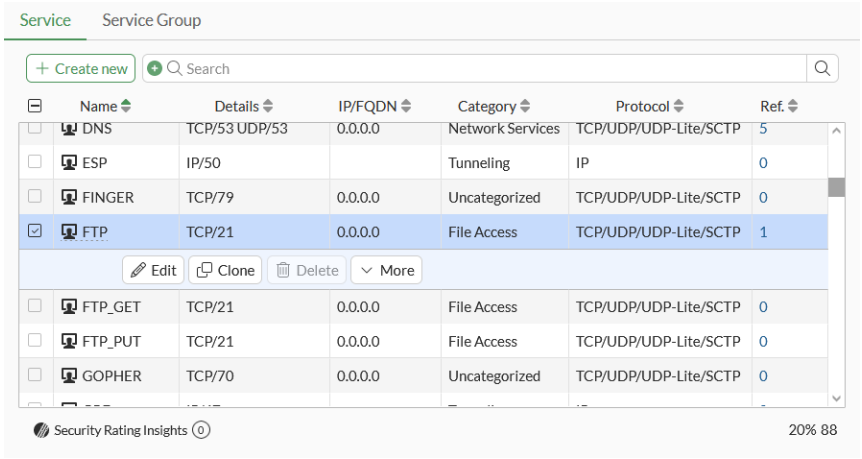
3. Select the service you want to clone.
4. Click *Clone*. The *New Service* page is displayed.



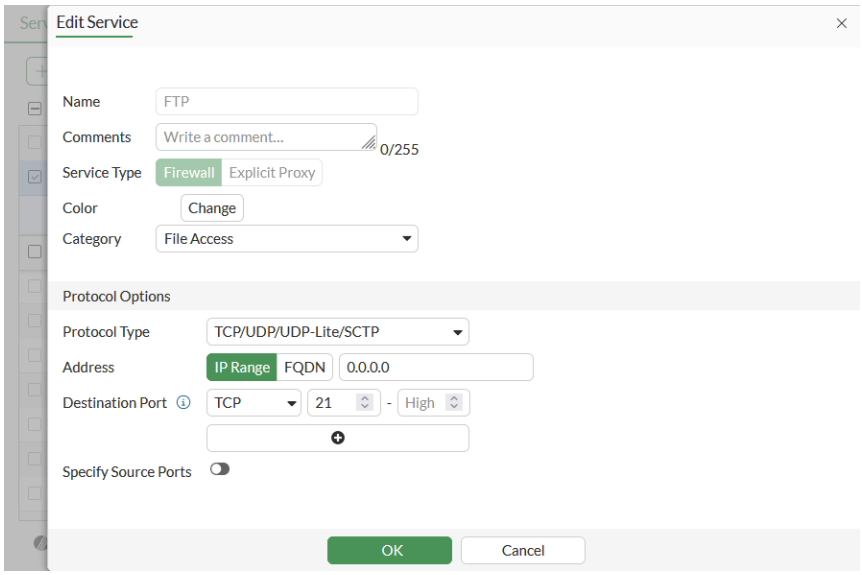
5. Edit the service details as needed.
6. Click *OK*.

To edit a service:

1. Go to *Policy & Objects* > *Services*.
2. Go to the *Services* tab and select the service that you need to edit.



3. Click *Edit*. The *Edit Service* page is displayed.



4. Edit the service details as needed.
5. Click *OK*.

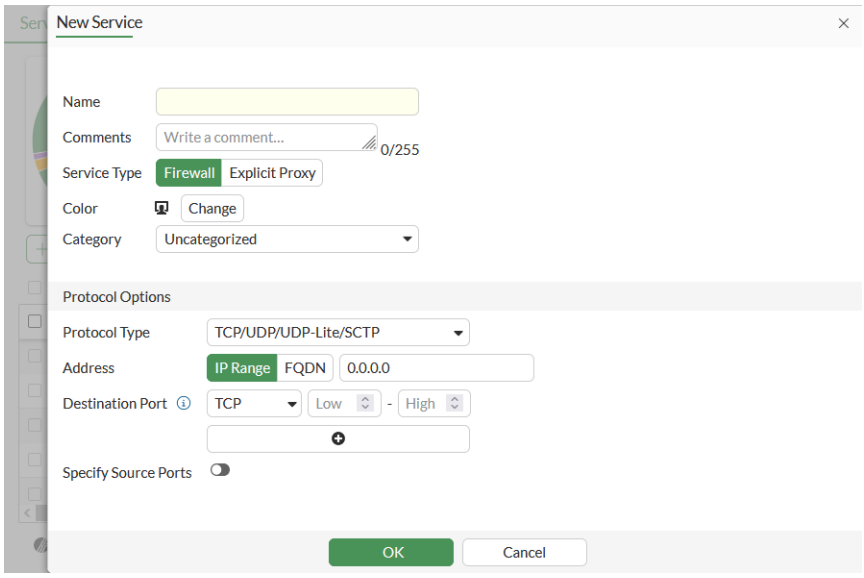
Custom services

You can create new, customized services in the *Policy & Objects > Services* page and the CLI. When creating a custom service, the ports, IP addresses, and protocols must be known for proper configuration. Once a service has been created, it must be applied to a firewall policy to take effect.

Custom services can also be created while configuring a new firewall policy.

To configure a custom service in the GUI:

1. Go to *Policy & Objects > Services*.
2. Go to the *Services* tab and click *Create new*.
3. Configure the service parameters as needed.



4. Click *OK*.

Custom services can be configured in the CLI for `TCP/UDP/UDP-Lite/SCTP`, `ICMP`, `ICMP6`, and `IP` protocols. Service parameters are dependent on the protocol type. See config firewall service custom in the CLI Reference guide for more information.

To configure a custom service in the CLI:

```
config firewall service custom
    edit <name>
        set protocol TCP/UDP/UDP-Lite/SCTP
        set tcp-portrange <destination port range>
        set udp-portrange <destination port range>
        set udplite-portrange <destination port range>
        set sctp-portrange <destination port range>
    next
end
```

Service groups

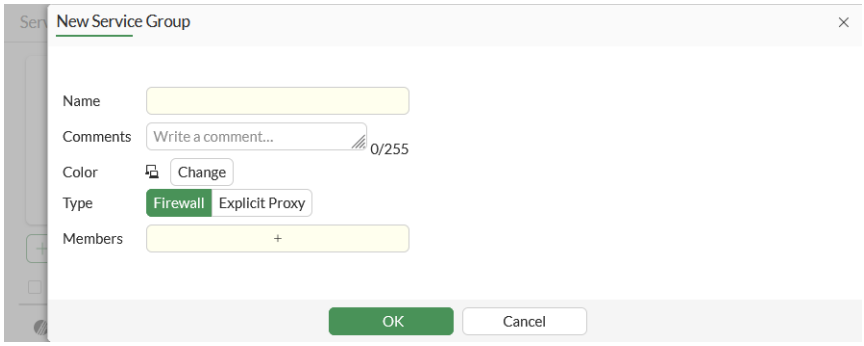
Service groups are a collection of services and other service groups, allowing multiple services to be applied in a firewall policy at once.

Service groups can be cloned and edited in the *Service Groups* tab using the same process as services. See Predefined services on page 1864.

You can enable the logging of specific service names instead of general service group names. When a firewall policy uses a service group, this setting ensures that logs display the exact service triggered (for example, `HTTPS`) rather than the group name (for example, `Web Access`).

To configure a service group in the GUI:

1. Go to *Policy & Objects* > *Services*.
2. Go to the *Service Groups* tab and click *Create new*. The *New Service Group* page is displayed.



3. Enter the *Name* of the group.
4. (Optional) Enter a comment and select a color for the service group.
5. Select the group type.
6. Click the *Members* field and select the services and service groups to include in the group.
7. Click *OK*.

To configure a service group in the CLI:

```
config firewall service group
    edit <name>
        set fabric-object {enable | disable}
        set member <service name1>, <service name2>
        set proxy {enable | disable}
    next
end
```

To log the detailed service name:

```
config log setting
    set detailed-svc-name {enable | disable}
end
```

NGFW policy

Profile-based next-generation firewall (NGFW) mode is the traditional mode where you create a profile (antivirus, web filter, and so on) and then apply the profile to a policy.

In policy-based NGFW mode, you allow applications and URL categories to be used directly in security policies, without requiring web filter or application control profiles. However, it is possible to select and apply web filter URL categories and groups.

In policy-based mode:

- Central NAT is always enabled. If no Central SNAT policy exists, you must create one. See Central SNAT on page 1924 for more information.
- Pre-match rules are defined separately from security policies, and define broader rules, such as SSL inspection and user authentication.
- The IPsec wizard is not supported.
- Only flow mode security profiles are supported; no proxy-based inspection.
- These profiles can be configured in the GUI:
  - AntiVirus
  - Web Filter
  - DNS Filter
  - Intrusion Prevention
  - File Filter
  - Email Filter
  - Virtual Patching
- These profiles can only be configured in the CLI:
  - DLP

If your FortiGate operates in NAT mode, rather than enabling source NAT in individual NGFW policies, go to *Policy & Objects* > *Central SNAT* and add source NAT policies that apply to all matching traffic. In many cases, you may only need one SNAT policy for each interface pair.

The NGFW mode is set per VDOM, and it is only available when the VDOM inspection mode is flow-based. You can operate your entire FortiGate or individual VDOMs in NGFW policy mode. The application default port can be set as a service port in the NGFW mode using the `default-app-port-as-service` option.

Inspection mode is configured on a per-policy basis in NGFW mode. This gives you more flexibility when setting up different policies.

In NGFW mode, administrators can configure a security policy in learn mode to monitor traffic. See Learn mode in security policies in NGFW mode on page 1881 for more information.

Other limitations

- When a change is made to a policy, existing sessions are not re-evaluated; only new sessions are re-evaluated.
- In the *Security Policy* view, there is no *Policy match* option.
- No *Application signature* page.
- Some application signatures requiring large scan-range are not supported.
- Webfilter FortiGuard Category based configurations are configured directly in the Security Policy. There are no options to configure the actions *Warning* or *Allow user to override block categories*.
- No *Web Profile* overrides.
- No *Workflow* mode.
- No support for HA session pickup upon HA failover.
- No support for configuring a static URL filter directly within the security policy.

Enabling policy-based NGFW mode

To enable policy-based NGFW mode without VDOMs in the GUI:

1. Go to *System* > *Settings* > *General*.
2. In *NGFW Mode*, select *Policy-based*.
3. Click *Apply*.

To enable policy-based NGFW mode with VDOMs in the GUI:

1. Go to *System* > *VDOM* .
2. Double-click a VDOM to edit the settings.
3. In *NGFW Mode*, select *Policy-based*.
4. Click *OK*.

To enable policy-based NGFW mode without VDOMs in the CLI:

```
config system settings
    set ngfw-mode policy-based
end
```

To enable policy-based NGFW mode with VDOMs in the CLI:

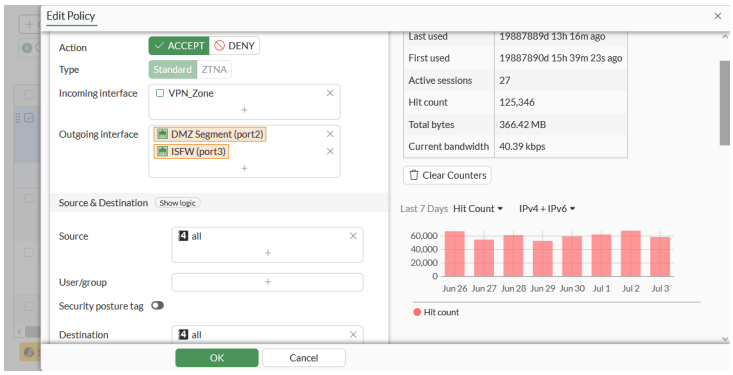
```
config vdom
    edit <vdom>
        config system settings
            set ngfw-mode policy-based
        end
    next
end
```

Hit count

When you edit a security policy, you can view the hit count for the last seven days. The hit count offers a rolling tally of how many times over the previous seven days a policy has been triggered, providing comprehensive, dynamic insight into policy usage patterns.

To view the hit count in the GUI:

1. Go to *Policy & Objects* > *Security Policy*.
2. Select a security policy, and click *Edit*. The *Last 7 Days* chart displays the hit count for the policy.



To view the hit count in the CLI:

1. Check the hit counter for a policy. The following example checks the hit count for the security policy with ID 1:

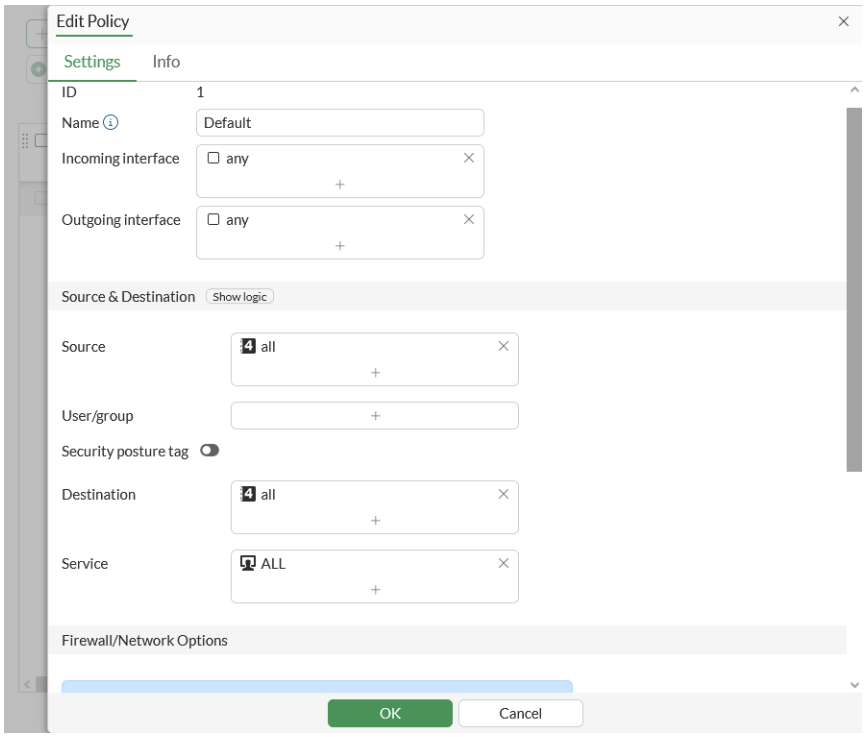
```
# diagnose ips pme policy stats
[  357] vdom:2 policy:1 ""
  first hit: 2024-06-26 10:22:56
  last hit:  2024-07-03 10:28:38
  hit count: 6 (1 0 0 1 1 1 1 1)

...
```

Security and SSL Inspection & Authentication policies

Security policies work with SSL Inspection & Authentication policies to inspect traffic. To allow traffic from a specific user or user group, both Security and SSL Inspection & Authentication policies must be configured. A default SSL Inspection & Authentication policy with the certificate-inspection SSL Inspection profile is preconfigured. Traffic will match the SSL Inspection & Authentication policy first. If the traffic is allowed, packets are sent to the IPS engine for application, URL category, user, and user group match, and then, if enabled, UTM inspection (antivirus, IPS, DLP, and email filter) is performed.

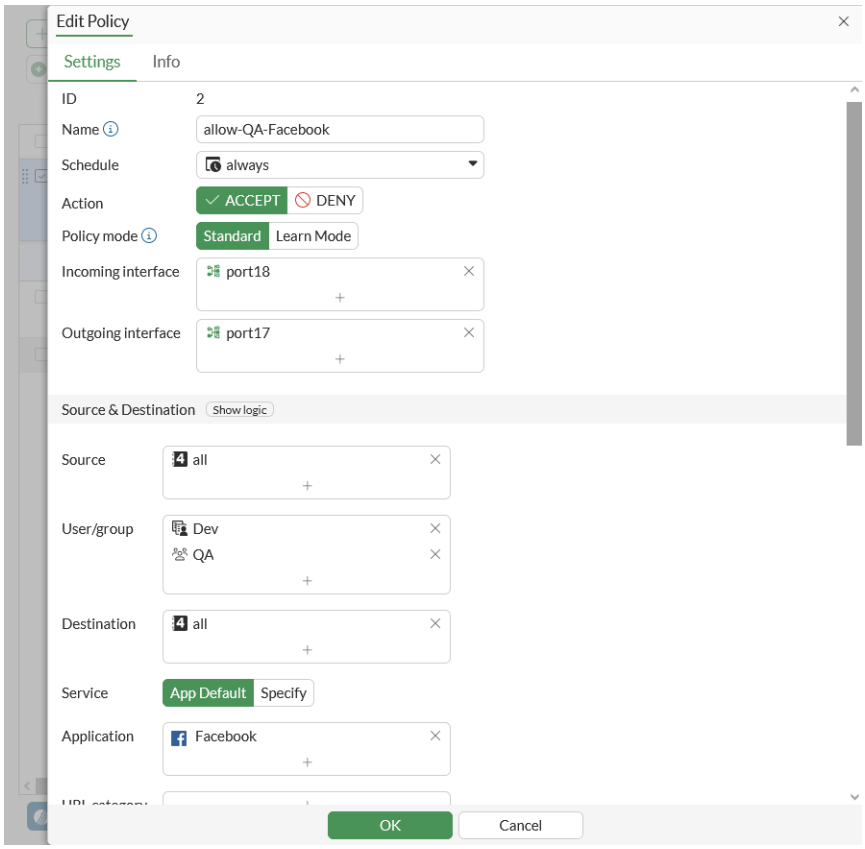
SSL Inspection & Authentication policies are used to pre-match traffic before sending the packets to the IPS engine:

- There are no schedule or action options; traffic matching the policy is always redirected to the IPS engine.
- SSL inspection, formerly configured in the VDOM settings, is configured in an SSL Inspection & Authentication policy.
- Users and user groups that require authentication must be configured in an SSL Inspection & Authentication policy.



Security policies work with SSL Inspection & Authentication policies to inspect traffic:

- Applications and URL categories can be configured directly in the policy.
- Users and user groups that require authentication must also be configured in a security policy.
- The available actions are *Accept* or *Deny*.
- The *Service* option can be used to enforce the standard port for the selected applications.
- UTM inspection is configured in a security policy.



To configure policies for Facebook and Gmail access in the CLI:

1. Configure an SSL Inspection & Authentication policy:

```
config firewall policy
    edit 1
        set name "Policy-1"
        set srcintf "port18"
        set dstintf "port17"
        set srcaddr "all"
        set dstaddr "all"
        set service "ALL"
        set ssl-ssh-profile "new-deep-inspection"
        set groups "Dev" "HR" "QA" "SYS"
    next
end
```

2. Configure security policies:

```
config firewall security-policy
    edit 2
        set name "allow-QA-Facebook"
        set srcintf "port18"
        set dstintf "port17"
        set srcaddr "all"
```

```
        set dstaddr "all"
        set action accept
        set schedule "always"
        set application 15832
        set groups "Dev" "QA"
    next
    edit 4
        set name "allow-QA-Email"
        set srcintf "port18"
        set dstintf "port17"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set url-category 23
        set groups "QA"
    next
end
```

Logs

In the application control and web filter logs, `securityid` maps to the security policy ID.

Application control log:

```
date=2024-06-17 time=16:35:47 logid="1059028704" type="utm" subtype="app-ctrl"
eventtype="signature" level="information" vd="vd1" eventtime=1560814547702405829 tz="-0700"
appid=15832 user="Jack" group="QA" srcip=10.1.100.102 dstip=157.240.3.29 srcport=56572 dstport=443
srcintf="port18" srcintfrole="undefined" dstintf="port17" dstintfrole="undefined" proto=6
service="P2P" direction="incoming" policyid=1 sessionid=42445 appcat="Social.Media" app="Facebook"
action="pass" hostname="external-sea1-1.xx.fbcdn.net" incidentserialno=1419629662 url="/"
securityid=2 msg="Social.Media: Facebook," apprisk="medium" scertcname="*.facebook.com"
scertissuer="DigiCert SHA2 High Assurance Server CA"
```

Web filter log:

```
date=2024-06-17 time=16:42:41 logid="0317013312" type="utm" subtype="webfilter" eventtype="ftgd_
allow" level="notice" vd="vd1" eventtime=1560814961418114836 tz="-0700" policyid=4 sessionid=43201
user="Jack" group="QA" srcip=10.1.100.102 srcport=56668 srcintf="port18" srcintfrole="undefined"
dstip=172.217.3.165 dstport=443 dstintf="port17" dstintfrole="undefined" proto=6 service="HTTPS"
hostname="mail.google.com" action="passthrough" reqtype="direct" url="/" sentbyte=709 rcvdbyte=0
direction="outgoing" msg="URL belongs to an allowed category in policy" method="domain" cat=23
catdesc="Web-based Email" securityid=4
```

Traffic logs:

```
date=2024-06-17 time=16:35:53 logid="0000000013" type="traffic" subtype="forward" level="notice"
vd="vd1" eventtime=1560814553778525154 tz="-0700" srcip=10.1.100.102 srcport=56572
srcintf="port18" srcintfrole="undefined" dstip=157.240.3.29 dstport=443 dstintf="port17"
dstintfrole="undefined" poluuid="b740d418-8ed3-51e9-5a7b-114e99ab6370" sessionid=42445 proto=6
action="server-rst" user="Jack" group="QA" policyid=1 policytype="consolidated" centralnatid=1
service="HTTPS" dstcountry="United States" srccountry="Reserved" trandisp="snat"
transip=172.16.200.2 transport=56572 duration=6 sentbyte=276 rcvdbyte=745 sentpkt=5 rcvdpkt=11
```

```
appid=15832 app="Facebook" appcat="Social.Media" apprisk="medium" utmaction="allow" countapp=1
utmref=65531-294

2: date=2024-06-17 time=16:47:45 logid="0000000013" type="traffic" subtype="forward"
level="notice" vd="vd1" eventtime=1560815265058557636 tz="-0700" srcip=10.1.100.102 srcport=56668
srcintf="port18" srcintfrole="undefined" dstip=172.217.3.165 dstport=443 dstintf="port17"
dstintfrole="undefined" poluuid="b740d418-8ed3-51e9-5a7b-114e99ab6370" sessionid=43201 proto=6
action="timeout" user="Jack" group="QA" policyid=1 policytype="consolidated" centralnatid=1
service="HTTPS" dstcountry="United States" srccountry="Reserved" trandisp="snat"
transip=172.16.200.2 transport=56668 duration=303 sentbyte=406 rcvdbyte=384 sentpkt=4 rcvdpkt=4
appcat="unscanned" utmaction="allow" countweb=1 utmref=65531-3486
```

Other NGFW policy-based mode options

You can combine *Application Control* and *Web Filter* in the same NGFW mode policy.

The following security profiles can be used in NGFW policy-based mode:

- AntiVirus
- Web Filter
- DNS Filter
- Intrusion Prevention
- File Filter
- Email Filter
- Virtual Patching
- DLP (CLI only)

Logging can also be enabled in security policies.

NGFW policy mode application default service

In NGFW policy-based mode, the application default service enforces applications running only on their default service port. The applications specified in the policy are monitored, and if traffic is detected from a nonstandard port, it is blocked, and a log entry is recorded with a *port-violation* event type.

If you are not using the default ports, and need to pick specific services, select *Specify* to select the required services.

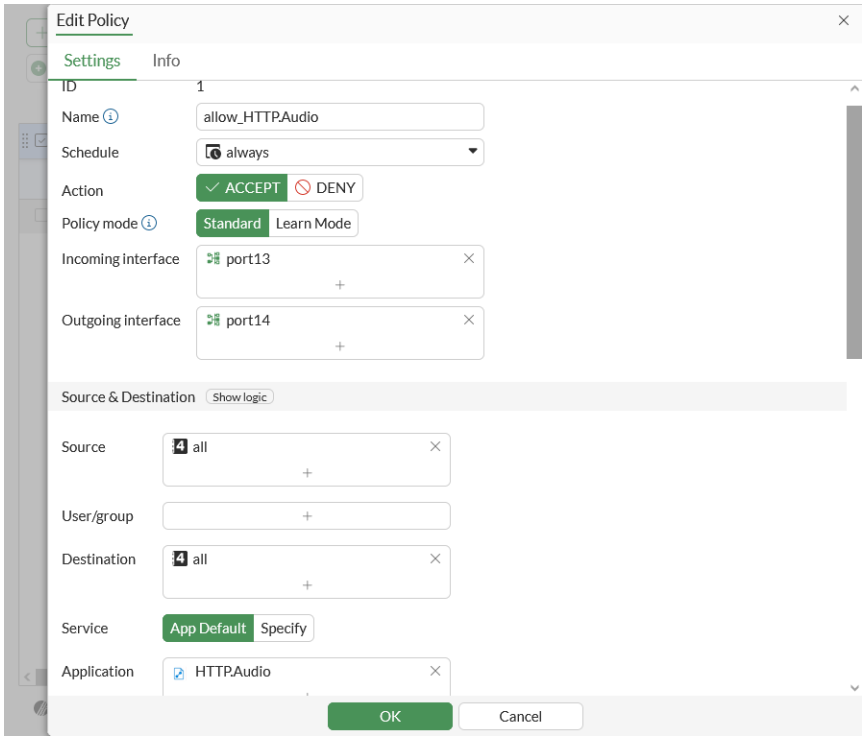
Example

In this example, the standard port is enforced for HTTPS traffic using the HTTP.Audio application.

First, an SSL Inspection & Authentication policy is created do to traffic pre-match, and then a security policy is created to allow the HTTP.Audio application when using the default port. Fetching an MP3 file from an HTTP server using port 443 is allowed, but is blocked when using a nonstandard port, such as 8443.

To enforce the HTTP.Audio application using the default port in the GUI:

1. Create a new SSL Inspection & Authentication policy, or use the default policy.
2. Go to *Policy & Objects* > *Security Policy*, and click *Create New*.
3. Enter a name for the policy, such as *allow_HTTP.Audio*.
4. Configure the ports as needed.
5. Set *Service* to *App Default*.
6. In the *Application* field, select *HTTP.Audio*.
7. Set the *Action* to *Accept*.



8. Click *OK*.

To enforce the HTTP.Audio application using the default port in the CLI:

1. Create a firewall policy:

```
config firewall policy
    edit 1
        set name "consolidated_all"
        set srcintf "port13"
        set dstintf "port14"
        set srcaddr "all"
        set dstaddr "all"
        set service "ALL"
        set ssl-ssh-profile "new-deep-inspection"
    next
end
```

2. Create a security policy:

```
config firewall security-policy
    edit 1
        set name "allow_HTTP.Audio"
        set srcintf "port13"
        set dstintf "port14"
        set srcaddr "all"
        set dstaddr "all"
        set enforce-default-app-port enable
        set action accept
        set schedule "always"
        set logtraffic all
        set application 15879
    next
end
```

Logs

The application logs show logs with an event type of `port-violation` for traffic on port 8443 that is blocked, and an event type of `signature` for traffic on port 443 that is allowed.

Blocked:

```
2: date=2024-06-18 time=16:15:40 logid="1060028736" type="utm" subtype="app-ctrl" eventtype="port-
violation" level="warning" vd="vd1" eventtime=1560899740218875746 tz="-0700" appid=15879
srcip=10.1.100.22 dstip=172.16.200.216 srcport=52680 dstport=8443 srcintf="port13"
srcintfrole="undefined" dstintf="port14" dstintfrole="undefined" proto=6 service="HTTPS"
direction="incoming" policyid=1 sessionid=5041 appcat="Video/Audio" app="HTTP.Audio"
action="block" hostname="172.16.200.216" incidentserialno=1906780850 url="/app_data/story.mp3"
securityid=2 msg="Video/Audio: HTTP.Audio," apprisk="elevated"
```

Allowed:

```
1: date=2024-06-18 time=16:15:49 logid="1059028704" type="utm" subtype="app-ctrl"
eventtype="signature" level="information" vd="vd1" eventtime=1560899749258579372 tz="-0700"
appid=15879 srcip=10.1.100.22 dstip=172.16.200.216 srcport=54527 dstport=443 srcintf="port13"
srcintfrole="undefined" dstintf="port14" dstintfrole="undefined" proto=6 service="HTTPS"
direction="incoming" policyid=1 sessionid=5064 appcat="Video/Audio" app="HTTP.Audio" action="pass"
hostname="172.16.200.216" incidentserialno=1139663486 url="/app_data/story.mp3" securityid=2
msg="Video/Audio: HTTP.Audio," apprisk="elevated"
```

Add option to set application default port as a service port

The `default-app-port-as-service` option can be used in NGFW mode to set the application default port as a service port. This allows applications to match the policy and be blocked immediately the first time that traffic hits the firewall. When this option is enabled, the NGFW policy aggregates the ports used by the applications in the policy and performs a pre-match on the traffic.

```
config system settings
    set default-app-port-as-service {enable | disable}
end
```

This option can be configured on a per-VDOM level.

This setting is enabled by default on new installations. When upgrading, the setting is disabled to retain the previous behavior.

To configure the application default port as service port:

1. Configure the VDOM settings:

```
config system settings
    set vdom-type traffic
    set opmode nat
    set ngfw-mode policy-based
    set block-land-attack disable
    set default-app-port-as-service enable
    set application-bandwidth-tracking disable
end
```

2. Configure the NGFW policy:

```
config firewall security-policy
    edit 1
        set name "test"
        set srcintf "port2"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set internet-service-src disable
        set enforce-default-app-port enable
        set action accept
    next
end
```

Sample logs

The following logging behavior occurs in NGFW mode with `default-app-port-as-service`:

- When `default-app-port-as-service` and `enforce-default-app-port` are enabled, traffic that does not match the default port is blocked immediately. Only a traffic log is generated.

  Log with SSH and FTP traffic:

```
1: date=2024-06-24 time=11:16:36 eventtime=1645730197145603994 tz="-0800" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="vd1" srcip=10.1.100.12 srcport=40402
srcintf="port2" srcintfrole="undefined" dstip=172.16.200.55 dstport=21 dstintf="port1"
dstintfrole="undefined" srccountry="Reserved" dstcountry="Reserved" sessionid=6811 proto=6
action="deny" policyid=0 policytype="security-policy" poluuid="7ed35582-95a2-51ec-0d21-
4093cb91e67b" policyname="Default" centralnatid=1 service="FTP" trandisp="snat"
transip=172.16.200.4 transport=40402 duration=10 sentbyte=0 rcvdbyte=0 sentpkt=0 rcvdpkt=0
appcat="unscanned"
```

Log with SSH and FTP traffic with port 2121:

```
1: date=2024-06-24 time=11:19:20 eventtime=1645730360685614031 tz="-0800" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="vd1" srcip=10.1.100.12 srcport=41362
srcintf="port2" srcintfrole="undefined" dstip=172.16.200.55 dstport=2121 dstintf="port1"
dstintfrole="undefined" srccountry="Reserved" dstcountry="Reserved" sessionid=7213 proto=6
action="deny" policyid=0 policytype="security-policy" poluuid="7ed35582-95a2-51ec-0d21-
4093cb91e67b" policyname="Default" centralnatid=1 service="tcp/2121" trandisp="snat"
transip=172.16.200.4 transport=41362 duration=9 sentbyte=60 rcvdbyte=0 sentpkt=1 rcvdpkt=0
appcat="unscanned"
```

- When `default-app-port-as-service` is disabled and `enforce-default-app-port` is enabled, traffic that does not match the default port is not blocked immediately. Application and traffic logs are generated.

Traffic log with SSH and FTP traffic:

```
1: date=2024-06-24 time=11:21:51 eventtime=1645730511325606916 tz="-0800" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="vd1" srcip=10.1.100.12 srcport=40408
srcintf="port2" srcintfrole="undefined" dstip=172.16.200.55 dstport=21 dstintf="port1"
dstintfrole="undefined" srccountry="Reserved" dstcountry="Reserved" sessionid=7522 proto=6
action="deny" policyid=0 policytype="security-policy" poluuid="7ed35582-95a2-51ec-0d21-
4093cb91e67b" policyname="Default" centralnatid=1 service="FTP" trandisp="snat"
transip=172.16.200.4 transport=40408 duration=14 sentbyte=164 rcvdbyte=171 sentpkt=3 rcvdpkt=2
appid=15896 app="FTP" appcat="Network.Service" apprisk="elevated" utmaction="block" countapp=1
utmref=65501-0
```

Application log with SSH and FTP traffic:

```
2: date=2024-02-24 time=11:21:39 eventtime=1645730499338228209 tz="-0800" logid="1059028705"
type="utm" subtype="app-ctrl" eventtype="signature" level="warning" vd="vd1" appid=15896
srcip=10.1.100.12 srccountry="Reserved" dstip=172.16.200.55 dstcountry="Reserved"
srcport=40408 dstport=21 srcintf="port2" srcintfrole="undefined" dstintf="port1"
dstintfrole="undefined" proto=6 service="FTP" direction="outgoing" policyid=0 sessionid=7522
action="block" appcat="Network.Service" app="FTP" incidentserialno=188744239
msg="Network.Service: FTP" apprisk="elevated"
```

Traffic log with SSH and FTP traffic with port 2121:

```
1: date=2024-06-24 time=11:24:25 eventtime=1645730665235613912 tz="-0800" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="vd1" srcip=10.1.100.12 srcport=41366
srcintf="port2" srcintfrole="undefined" dstip=172.16.200.55 dstport=2121 dstintf="port1"
dstintfrole="undefined" srccountry="Reserved" dstcountry="Reserved" sessionid=7876 proto=6
action="deny" policyid=0 policytype="security-policy" poluuid="7ed35582-95a2-51ec-0d21-
4093cb91e67b" policyname="Default" centralnatid=1 service="tcp/2121" trandisp="snat"
transip=172.16.200.4 transport=41366 duration=11 sentbyte=112 rcvdbyte=171 sentpkt=2 rcvdpkt=2
appid=15896 app="FTP" appcat="Network.Service" apprisk="elevated" utmaction="block" countapp=1
utmref=65500-0
```

Application log with SSH and FTP traffic with port 2121:

```
2: date=2024-06-24 time=11:24:16 eventtime=1645730656426052412 tz="-0800" logid="1060028736"
type="utm" subtype="app-ctrl" eventtype="port-violation" level="warning" vd="vd1" appid=15896
srcip=10.1.100.12 srccountry="Reserved" dstip=172.16.200.55 dstcountry="Reserved"
srcport=41366 dstport=2121 srcintf="port2" srcintfrole="undefined" dstintf="port1"
dstintfrole="undefined" proto=6 service="FTP" direction="outgoing" policyid=0 sessionid=7876
action="block" appcat="Network.Service" app="FTP" incidentserialno=188744241
msg="Network.Service: FTP, non-default port used: 2121" apprisk="elevated"
```

Application logging in NGFW policy mode

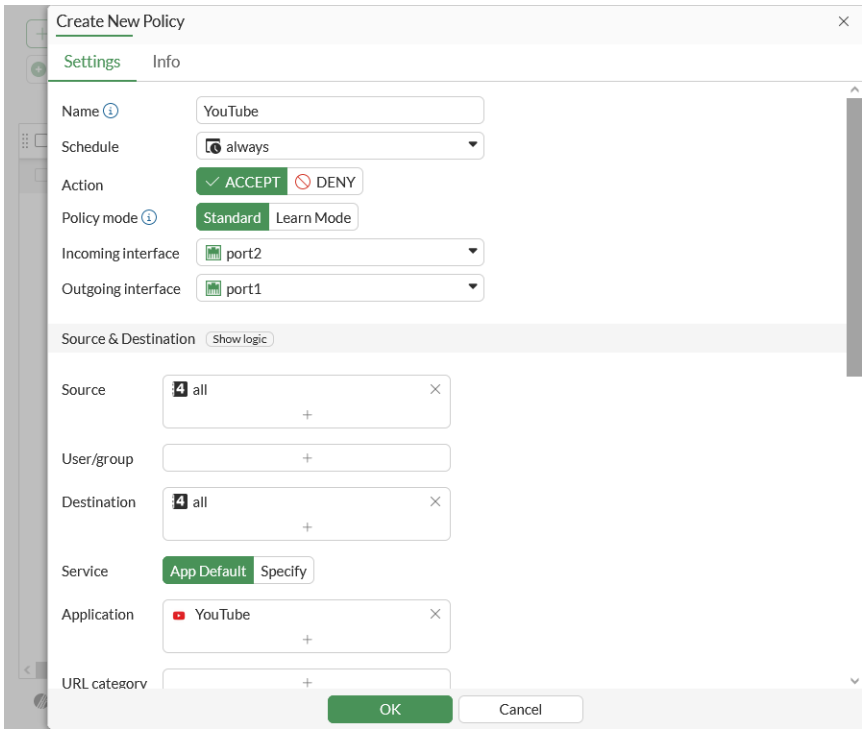
In NGFW policy mode, if an application, application category, or application group is selected on a security policy, and traffic logging is set to *UTM* or *All*, then application control logs will be generated. In addition, when a signature is set to the *ACCEPT* action under a security policy, all corresponding child signatures will be assessed and logged as well.

Under NGFW, with `default-app-port-as-service` enabled, enable APP Default. The traffic which doesn't match the default port will be blocked immediately, and there is only traffic log generated.

Under NGFW, with `default-app-port-as-service` disabled, enable APP Default. The traffic which doesn't match the default port will not be blocked immediately, and there is app and traffic logs generated.

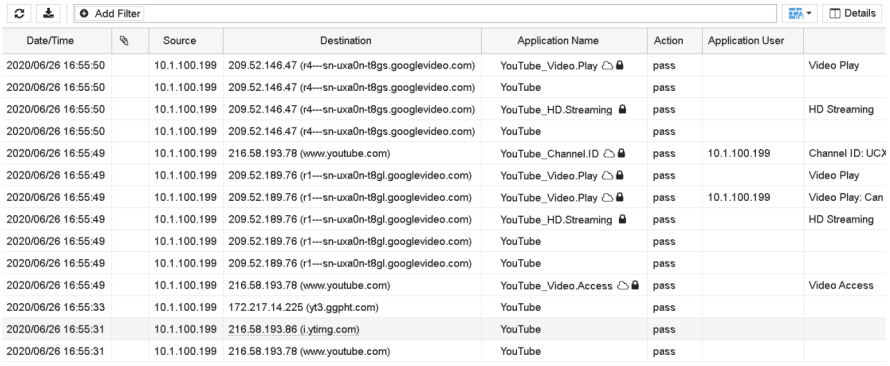
To verify application logging:

1. Go to *Policy & Objects* > *Security Policy* and configure a new policy for YouTube.
2. Set *Application* to *YouTube*.
3. Set *Action* to *ACCEPT* and *Log Allowed Traffic* to *Security Events*.



4. Configure the remaining settings as required, then click *OK*.
5. On a client system, play some YouTube videos.
6. On FortiOS, go to *Log & Report* > *Security Events* and view the *Application Control* logs.

   There are logs not only for *YouTube*, but also for *YouTube_Video.Play*, *YouTube_Video.Access*, and so on, as verified from the *Application Name* column.



Add Filter | Details

Date/Time	Source	Destination	Application Name	Action	Application User	
2020/06/26 16:55:50	10.1.100.199	209.52.146.47 (r4---sn-uxa0n-t8gs.googlevideo.com)	YouTube_Video.Play	pass		Video Play
2020/06/26 16:55:50	10.1.100.199	209.52.146.47 (r4---sn-uxa0n-t8gs.googlevideo.com)	YouTube	pass		
2020/06/26 16:55:50	10.1.100.199	209.52.146.47 (r4---sn-uxa0n-t8gs.googlevideo.com)	YouTube_HD.Streaming	pass		HD Streaming
2020/06/26 16:55:50	10.1.100.199	209.52.146.47 (r4---sn-uxa0n-t8gs.googlevideo.com)	YouTube	pass		
2020/06/26 16:55:49	10.1.100.199	216.58.193.78 (www.youtube.com)	YouTube_Channel.ID	pass	10.1.100.199	Channel ID: UCX
2020/06/26 16:55:49	10.1.100.199	209.52.189.76 (r1---sn-uxa0n-t8gl.googlevideo.com)	YouTube_Video.Play	pass		Video Play
2020/06/26 16:55:49	10.1.100.199	209.52.189.76 (r1---sn-uxa0n-t8gl.googlevideo.com)	YouTube_Video.Play	pass	10.1.100.199	Video Play: Can
2020/06/26 16:55:49	10.1.100.199	209.52.189.76 (r1---sn-uxa0n-t8gl.googlevideo.com)	YouTube_HD.Streaming	pass		HD Streaming
2020/06/26 16:55:49	10.1.100.199	209.52.189.76 (r1---sn-uxa0n-t8gl.googlevideo.com)	YouTube	pass		
2020/06/26 16:55:49	10.1.100.199	209.52.189.76 (r1---sn-uxa0n-t8gl.googlevideo.com)	YouTube	pass		
2020/06/26 16:55:49	10.1.100.199	216.58.193.78 (www.youtube.com)	YouTube_Video.Access	pass		Video Access
2020/06/26 16:55:33	10.1.100.199	172.217.14.225 (yt3.ggpht.com)	YouTube	pass		
2020/06/26 16:55:31	10.1.100.199	216.58.193.86 (i.ytimg.com)	YouTube	pass		
2020/06/26 16:55:31	10.1.100.199	216.58.193.78 (www.youtube.com)	YouTube	pass		

Learn mode in security policies in NGFW mode

In NGFW mode, administrators can configure a security policy in learn mode to monitor traffic that passes through the source and destination interfaces. The learn mode uses a special prefix in the `policymode` and `profile` fields in traffic and UTM logs for use by FortiAnalyzer and the Policy Analyzer Management Extension Application (MEA) that is available with FortiManager.

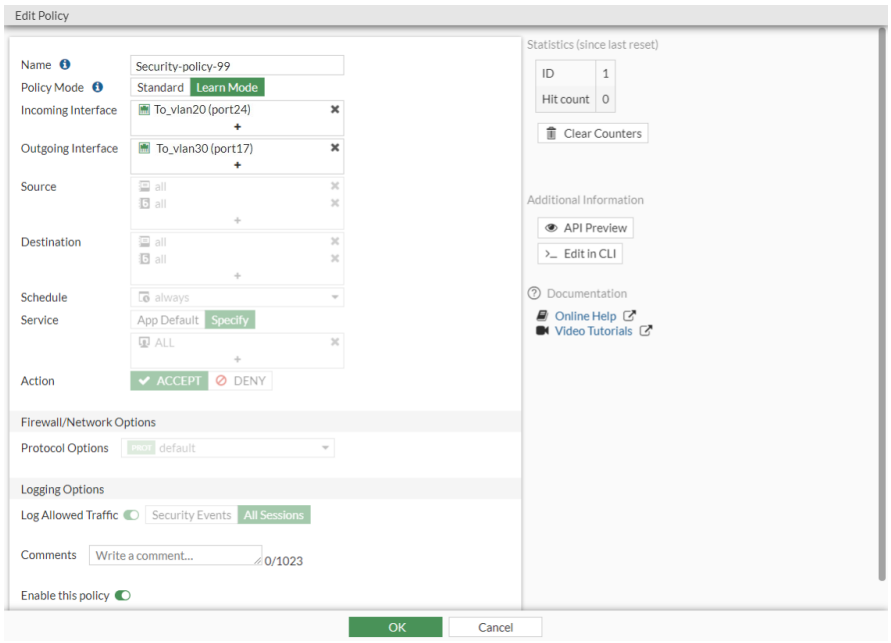
When enabled on FortiManager, Policy Analyzer MEA works with security policies in learning mode to analyze logs sent from a managed FortiGate to FortiAnalyzer. Based on the analyzed traffic, FortiManager administrators can choose to automatically create a policy in FortiManager for the managed FortiGate. For more information about Policy Analyzer MEA, see the Policy Analyzer Administration Guide.

The following limitations apply when learn mode is enabled in a security policy:

- Only interfaces with `device-identification enable` can be used as source interfaces in a security policy with learning mode enabled.
- Incoming and outgoing interfaces do not support `any`.
- Internet service is not supported.
- NAT46 and NAT64 are not supported.
- Users and groups are not supported.
- Some negate options are not supported.

To enable learn mode in the GUI:

1. Enable policy-based NGFW mode:
   a. Go to *System* > *Settings* > *General*.
   b. Set the *NGFW Mode* to *Policy-based* and click *Apply*.
2. Go to *Policy & Objects* > *Security Policy*, and open a security policy for editing.
3. Set the *Policy Mode* to *Learn Mode*.



4. Select an *Incoming Interface*.
5. Select an *Outgoing Interface*.
6. (Optional) Type a comment in the *Comments* box.
7. Toggle on *Enable this policy*.
8. Click *OK* to save the security policy.

To enable learn mode in the CLI:

1. Enable policy-based NGFW mode:

```
config system settings
    set ngfw-mode policy-based
end
```

2. Enable learn mode in a security policy:

```
config firewall security-policy
    edit <id>
        set learning-mode enable
    next
end
```

To view learn mode fields in logs in the CLI:

1. Filter and view fields in traffic logs:

```
# execute log filter category 0

# execute log display

1 logs found.
```

```
1 logs returned.

1: date=2022-03-21 time=10:21:11 eventtime=1647883271150012188 tz="-0700" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="root" srcip=10.1.100.41 srcport=43296
srcintf="port24" srcintfrole="undefined" dstip=172.16.200.55 dstport=80 dstintf="port17"
dstintfrole="wan" srccountry="Reserved" dstcountry="Reserved" sessionid=33934 proto=6
policymode="learn" action="accept" policyid=99 policytype="security-policy" poluuid="6e3f7f54-
a932-51ec-73ba-8282cfd0b73c" policyname="Security-policy-99" centralnatid=3 service="HTTP"
trandisp="snat" transip=172.16.200.9 transport=43296 duration=1 sentbyte=412 rcvdbyte=529
sentpkt=6 rcvdpkt=4 appid=15893 app="HTTP.BROWSER" appcat="Web.Client" apprisk="medium"
utmaction="allow" countweb=1 countav=1 countips=3 countapp=1 crscore=50 craction=2
srchwvendor="VMware" devtype="Computer" osname="Debian" mastersrcmac="00:0c:29:b5:92:8d"
srcmac="00:0c:29:b5:92:8d" srcserver=0 utmref=65534-0
```

2. Filter and view fields in UTM logs:

```
# execute log filter category 2

# execute log display

1 logs found.

1 logs returned.

1: date=2022-03-21 time=10:21:09 eventtime=1647883270101403283 tz="-0700" logid="0211008193"
type="utm" subtype="virus" eventtype="infected" level="notice" vd="root" policyid=99
poluuid="6e3f7f54-a932-51ec-73ba-8282cfd0b73c" policytype="security-policy" policymode="learn"
msg="File is infected." action="monitored" service="HTTP" sessionid=33934 srcip=10.1.100.41
dstip=172.16.200.55 srcport=43296 dstport=80 srccountry="Reserved" dstcountry="Reserved"
srcintf="port24" srcintfrole="undefined" dstintf="port17" dstintfrole="wan" proto=6
direction="incoming" filename="eicar.com" quarskip="Quarantine-disabled" virus="EICAR_TEST_
FILE" viruscat="Virus" dtype="av-engine" ref="http://www.fortinet.com/ve?vn=EICAR_TEST_FILE"
virusid=2172 url="http://172.16.200.55/virus/eicar.com" profile="learn-av" agent="curl/7.35.0"
httpmethod="GET"
analyticscksum="275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f"
analyticssubmit="false" crscore=50 craction=2 crlevel="critical" rawdata="Response-Content-
Type=application/x-msdos-program"
```

3. Filter and view fields in UTM-IPS logs:

```
# execute log filter category 4

# execute log display

3 logs found.

3 logs returned.

1: date=2022-03-21 time=10:21:09 eventtime=1647883270101485354 tz="-0700" logid="0419016384"
type="utm" subtype="ips" eventtype="signature" level="alert" vd="root" severity="info"
srcip=10.1.100.41 srccountry="Reserved" dstip=172.16.200.55 dstcountry="Reserved"
srcintf="port24" srcintfrole="undefined" dstintf="port17" dstintfrole="wan" sessionid=33934
```

```
action="detected" proto=6 service="HTTP" policyid=99 poluuid="6e3f7f54-a932-51ec-73ba-
8282cfd0b73c" policytype="security-policy" policymode="learn" attack="Eicar.Virus.Test.File"
srcport=43296 dstport=80 agent="curl/7.35.0" httpmethod="GET" direction="incoming"
attackid=29844 profile="learn-ips" ref="http://www.fortinet.com/ids/VID29844"
incidentserialno=158335134 attackcontextid="2/2"
attackcontext="YW0NCg0KWDVPIVAlQEFQWzRcUFpYNTQoUF4pN0NDKTd9JEVJQ0FSLVNUQU5EQVJELUFOVElWSVJVUy1
URVNULUZJTEUhJEgrSCo8L1BBQ0tFVD4="

2: date=2022-03-21 time=10:21:09 eventtime=1647883270101484791 tz="-0700" logid="0419016384"
type="utm" subtype="ips" eventtype="signature" level="alert" vd="root" severity="info"
srcip=10.1.100.41 srccountry="Reserved" dstip=172.16.200.55 dstcountry="Reserved"
srcintf="port24" srcintfrole="undefined" dstintf="port17" dstintfrole="wan" sessionid=33934
action="detected" proto=6 service="HTTP" policyid=99 poluuid="6e3f7f54-a932-51ec-73ba-
8282cfd0b73c" policytype="security-policy" policymode="learn" attack="Eicar.Virus.Test.File"
srcport=43296 dstport=80 agent="curl/7.35.0" httpmethod="GET" direction="incoming"
attackid=29844 profile="learn-ips" ref="http://www.fortinet.com/ids/VID29844"
incidentserialno=158335134 attackcontextid="1/2"
attackcontext="PFBBVFRFUk5TPiBYNU8hUCVAQVBbNFxQWlg1NChQXik3Q0MpN30kRUlDQVItU1RBTkRBUkQtQU5USVZ
JUlVTLVRFU1QtRklMRSEkSCtIKjtYNU8hUCVAQVBbNFxQWlg1NChQXik3Q0MpN30kRUlDQVItU1RBTkRBUkQtQU5USVZJU
lVTLVRFU1QtRklMRSEkSCtIKjwvUEFUVEVSTlM+CjxVUkk+IDwvVVJJPgo8SEVBREVSPiBIVFRQLzEuMSAyMDAgT0sNCkR
hdGU6IE1vbiwgMjEgTWFyIDIwMjIgMTc6MjE6MTAgR01UDQpTZXJ2ZXI6IEFwYWNoZS8yLjQuMTggKFVidW50dSkNCkxhc
3QtTW9kaWZpZWQ6IFRodSwgMDEgRGVjIDIwMTYgMDE6MjY6MzUgR01UDQpFVGFnOiAiNDQtNTQyOGViNjU4MDk3YSINCkF
jY2VwdC1SYW5nZXM6IGJ5dGVzDQpDb250ZW50LUxlbmd0aDogNjgNCkNvbnRlbnQtVHlwZTogYXBwbGljYXRpb24veC1tc
2Rvcy1wcm9ncmFtDQoNCjwvSEVBREVSPgo8Qk9EWT4gWDVPIVAlQEFQWzRcUFpYNTQoUF4pN0NDKTd9JEVJQ0FSLVNUQU5
EQVJELUFOVElWSVJVUy1URVNULUZJTEUhJEgrSCo8L0JPRFk+CjxQQUNLRVQ+IEhUVFAvMS4xIDIwMCBPSw0KRGF0ZTogT
W9uLCAyMSBNYXIgMjAyMiAxNzoyMToxMCBHTVQNClNlcnZlcjogQXBhY2hlLzIuNC4xOCAoVWJ1bnR1KQ0KTGFzdC1Nb2R
pZmllZDogVGh1LCAwMSBEZWMgMjAxNiAwMToyNjozNSBHTVQNCkVUYWc6ICI0NC01NDI4ZWI2NTgwOTdhIg0KQWNjZXB0L
VJhbmdlczogYnl0ZXMNCkNvbnRlbnQtTGVuZ3RoOiA2OA0KQ29udGVudC1UeXBlOiBhcHBsaWNhdGlvbi94LW1zZG9zLXB
yb2dy"

3: date=2022-03-21 time=10:21:09 eventtime=1647883270101483279 tz="-0700" logid="0419016384"
type="utm" subtype="ips" eventtype="signature" level="alert" vd="root" severity="info"
srcip=10.1.100.41 srccountry="Reserved" dstip=172.16.200.55 dstcountry="Reserved"
srcintf="port24" srcintfrole="undefined" dstintf="port17" dstintfrole="wan" sessionid=33934
action="detected" proto=6 service="HTTP" policyid=99 poluuid="6e3f7f54-a932-51ec-73ba-
8282cfd0b73c" policytype="security-policy" policymode="learn" attack="Eicar.Virus.Test.File"
srcport=43296 dstport=80 hostname="172.16.200.55" url="/virus/eicar.com" agent="curl/7.35.0"
httpmethod="GET" direction="incoming" attackid=29844 profile="learn-ips"
ref="http://www.fortinet.com/ids/VID29844" incidentserialno=158335134 msg="file_transfer:
Eicar.Virus.Test.File" attackcontextid="0/2" rawdataid="1/1" rawdata="Response-Content-
Type=application/x-msdos-program"
```

Filter and view fields in UTM-webfilter logs:

```
# execute log filter category 3

# execute log display

2 logs found.

2 logs returned.
```

```
2: date=2022-03-21 time=10:21:09 eventtime=1647883270100329681 tz="-0700" logid="0319013317"
type="utm" subtype="webfilter" eventtype="urlmonitor" level="notice" vd="root" policyid=99
poluuid="6e3f7f54-a932-51ec-73ba-8282cfd0b73c" policytype="security-policy" policymode="learn"
sessionid=33934 srcip=10.1.100.41 srcport=43296 srccountry="Reserved" srcintf="port24"
srcintfrole="undefined" dstip=172.16.200.55 dstport=80 dstcountry="Reserved" dstintf="port17"
dstintfrole="wan" proto=6 httpmethod="GET" service="HTTP" hostname="172.16.200.55"
agent="curl/7.35.0" profile="learn-webf" action="passthrough" reqtype="direct"
url="http://172.16.200.55/virus/eicar.com" sentbyte=92 rcvdbyte=0 direction="outgoing" msg="URL
has been visited" ratemethod="domain" cat=255 catdesc="Unknown"
```

Dynamic address tags

Tags for dynamic addresses, including EMS (normal and local EMS tags), FortiVoice, and FortiNAC can be used as the source or destination address in security policies. Once these tags are used in security policies, run `diagnose ips pme dynamic-address list` to show the addresses that are used in the policy. The following example uses an EMS tag.

To apply an EMS tag object to a security policy in the GUI:

1. Go to *Policy & Objects* > *Security Policy*.
2. Click *Create new* or edit an existing policy.
3. In the *Source* field, click the *+* and select *EMS1_ZTNA_ZT_OS_WIN*.
4. Configure the other settings as needed.
5. Click *OK*.

To apply an EMS tag object to a security policy in the CLI:

1. Configure the policy:

```
config firewall security-policy
    edit 1
        set name "ddd"
        set srcintf "port8"
        set dstintf "port7"
        set srcaddr "EMS1_ZTNA_ZT_OS_WIN"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set logtraffic all
    next
end
```

2. Verify which IP addresses are used in the policy:

```
# diagnose ips pme dynamic-address list
EMS1_ZTNA_ZT_OS_WIN [vdom=0 type=IP]:
```

Local-in policy

While security profiles control traffic flowing through the FortiGate, local-in policies control inbound traffic that is going to a FortiGate interface.

Administrative access traffic (HTTPS, PING, SSH, and others) can be controlled by allowing or denying the service in the interface settings. Trusted hosts can be configured under an administrator to restrict the hosts that can access the administrative service.

Local-in policies allow administrators to granularly define the source and destination addresses, interface, and services. Traffic destined for the FortiGate interface specified in the policy that meets the other criteria is subject to the policies action.

Local-in policies can be used to restrict administrative access or other services, such as VPN, that can be specified as services. You can define source addresses or address groups to restrict access from. For example, by using a geographic type address you can restrict a certain geographic set of IP addresses from accessing the FortiGate. An IP Address external feed can also be used as either a source or destination address; see Applying an IP address external feed in a local-in policy on page 4715 for more information.

Local-in policies can also use virtual patching to mitigate known vulnerabilities targeted at the FortiGate. Vulnerability rules are scanned on local-in traffic on the specified interface, and all matched local-in traffic is dropped accordingly. See Virtual patching on the local-in management interface on page 2009 for more information.

This topic includes the following sections:

- Default local-in policy on page 1886
- Configuring the local-in policy on page 1887
- Implicit deny rule on page 1892
- Additional options on page 1893
- Logging local traffic per local-in policy on page 1895

Default local-in policy

The default local-in policy has internet service source enabled for Malicious-Malicious.Server, Tor-Exit.Node, and Tor-Relay.Node ISDB sources. This policy is designed to utilize these three sources to identify known malicious threat actors and prevent them from accessing any interface on the FortiGate on any service and port.

The default local-in policy is automatically added when a FortiGate is in factory default setting, or a new VDOM is created. Resetting your device to factory default settings is not recommended, so you can manually add the policy on FortiOS versions that support ISDB as a local-in policy source (7.4.4 and higher).

To manually add the policy:

```
config firewall local-in-policy
    edit 1
        set intf "any"
        set dstaddr "all"
        set internet-service-src enable
        set internet-service-src-name "Malicious-Malicious.Server" "Tor-Exit.Node" "Tor-
Relay.Node"
```

```
        set service "ALL"
        set schedule "always"
    next
end
```

To view the malicious sources that are blocked:

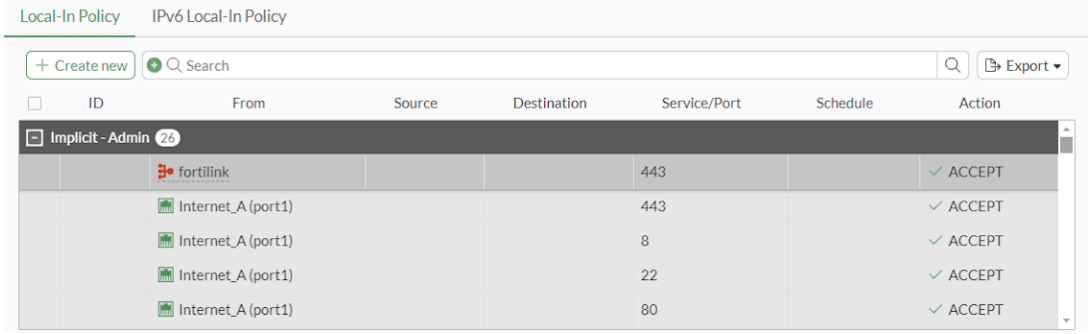
1. Go to *Policy & Objects* > *Internet Service Database* and select the *Internet Service* tab.
2. Search for *Malicious-Malicious.Server*, *Tor-Exit.Node*, or *Tor-Relay.Node*.
3. Hover over the entry and, in the pop-up, click *View/Edit Entries*.
   The listed addresses are the sources that will be blocked.

Configuring the local-in policy

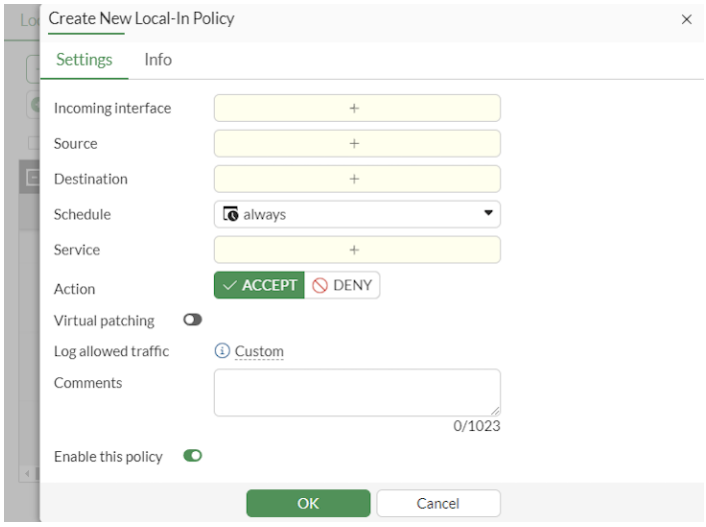
Local-in policies can be configured in the GUI and CLI.

To create an IPv4 local-in policy in the GUI:

1. Go to *Policy & Objects* > *Local-In Policy*.
2. Go to the *Local-In Policy* tab.



3. Click *Create new*. The *Create New Local-In Policy* pane is displayed.

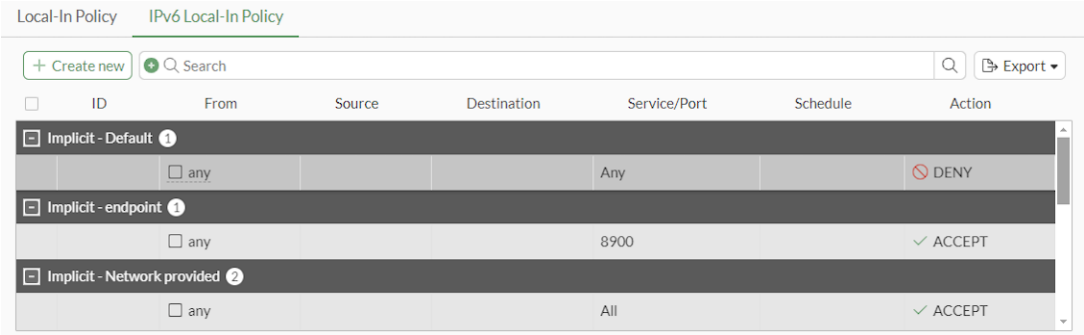


4. Configure the policy parameters.

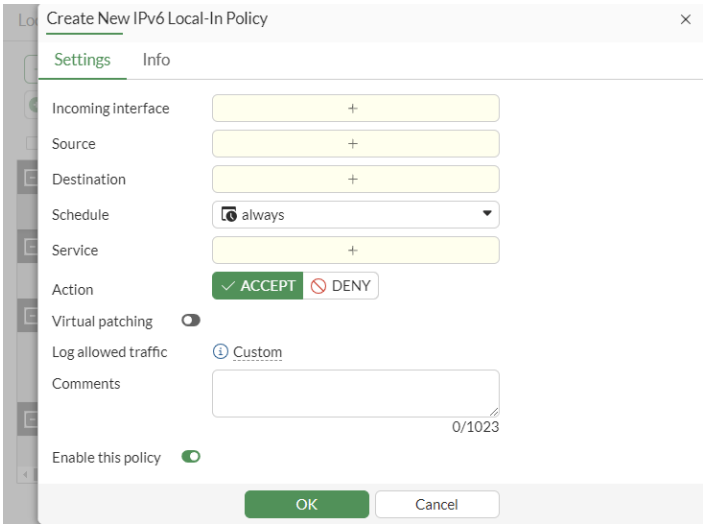
5. Click *OK*.

To create an IPv6 local-in policy in the GUI:

1. Go to *Policy & Objects* > *Local-In Policy*.
2. Go to the *IPv6 Local-In Policy* tab.



3. Click *Create new*. The *Create New IPv6 Local-In Policy* pane is displayed.



4. Configure the policy parameters.
5. Click *OK*.

To configure a local-in policy using the CLI:

```
config firewall {local-in-policy | local-in-policy6}
    edit <policy_number>
        set intf <interface>
        set srcaddr <source_address> [source_address] ...
        set dstaddr <destination_address> [destination_address] ...
        set action {accept | deny}
        set service <service_name> [service_name] ...
        set schedule <schedule_name>
        set virtual-patch {enable | disable}
        set comments <string>
```

```
    next
end
```

Example 1

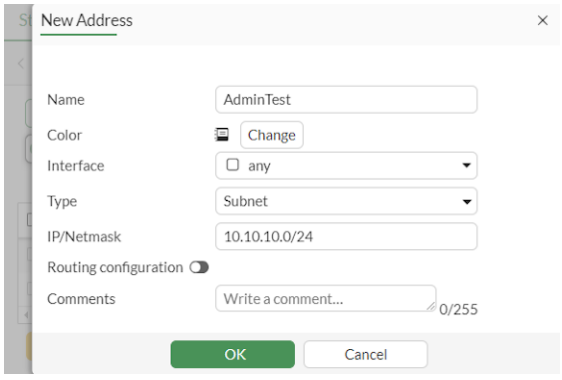
In this example, a local-in policy will be configured to prevent the source subnet 10.10.10.0/24 from pinging port1, but allow administrative access for PING on port1.

To configure the local-in policy for admin access in the CLI:

```
config firewall address
    edit "10.10.10.0"
        set subnet 10.10.10.0 255.255.255.0
    next
end
config firewall local-in-policy
    edit 1
        set intf "port1"
        set srcaddr "10.10.10.0"
        set dstaddr "all"
        set service "PING"
        set schedule "always"
    next
end
```

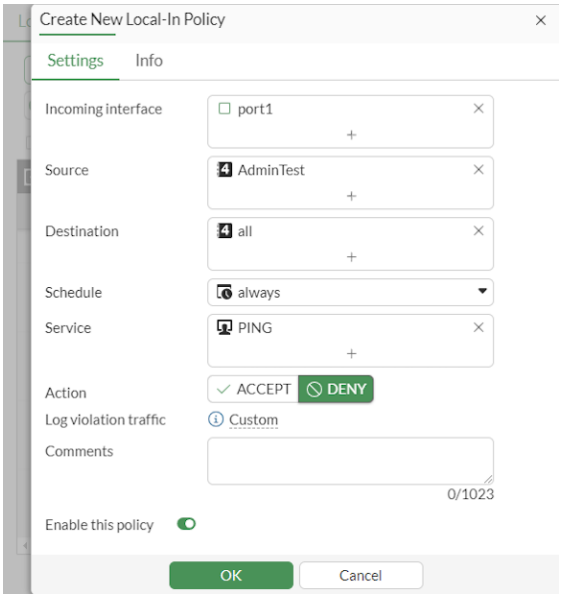
To configure the local-in policy for admin access:

1. Configure the firewall address:
   a. Go to *Policy & Objects* > *Addresses*.
   b. Click *Create new*.
   c. Enter a name for the address.
   d. Set *Type* to *Subnet*.
   e. Set *IP/Netmask* to *10.10.10.0/24*.
   f. Click *OK*.



2. Configure the local-in policy:

a. Go to *Policy & Objects* > *Local-In Policy*.
b. Click *Create new*.
c. Configure the following:
   i. Set *Interface* to *port1*.
   ii. Set *Source* to the address created previously.
   iii. Set *Destination* to *all*.
   iv. Set *Schedule* to *always*.
   v. Set *Service* to *PING*.
   vi. Set *Action* to *DENY*.



d. Click *OK*.

To test the configuration:

1. From the PC at 10.10.10.12, start a continuous ping to port1:

```
ping 192.168.2.5 –t
```

2. On the FortiGate, enable debug flow:

```
# diagnose debug flow filter addr 10.10.10.12
# diagnose debug flow filter proto 1
# diagnose debug enable
# diagnose debug flow trace start 10
```

3. The output of the debug flow shows that traffic is dropped by local-in policy 1:

```
# id=20085 trace_id=1 func=print_pkt_detail line=5746 msg="vd-root:0 received a packet
(proto=1, 10.10.10.12:1->192.168.2.5:2048) from port1. type=8, code=0, id=1, seq=128."
id=20085 trace_id=1 func=init_ip_session_common line=5918 msg="allocate a new session-
0017c5ad"
id=20085 trace_id=1 func=vf_ip_route_input_common line=2615 msg="find a route: flag=80000000
```

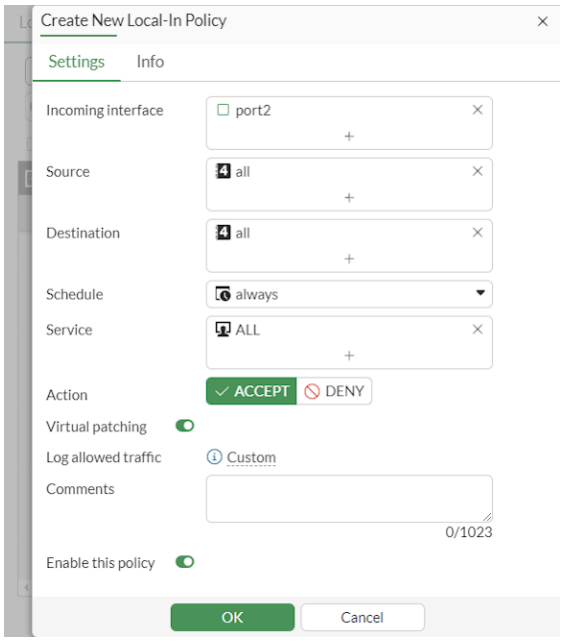
```
gw-192.168.2.5 via root"
id=20085 trace_id=1 func=fw_local_in_handler line=474 msg="iprope_in_check() check failed on
policy 1, drop"
```

Example 2

The following example demonstrates how to enable virtual patching on the port2 interface using a local-in policy. For information on enabling virtual patching in the CLI and observing the outcomes, see Virtual patching on the local-in management interface on page 2009.

To enable virtual patching:

1. Go to *Policy & Objects* > *Local-In Policy*.
2. Click *Create new*.
3. Configure the following:
   a. Set *Interface* to *port2*.
   b. Set *Source* to *all*.
   c. Set *Destination* to *all*.
   d. Set *Schedule* to *always*.
   e. Set *Service* to *ALL*.
   f. Set *Action* to *ACCEPT*.
   g. Enable *Virtual patching*.



4. Click *OK*.

Implicit deny rule

If a local-in-policy is not functioning correctly and traffic that should be blocked is being allowed through, the issue may be that the implicit deny local-in-policy has not been created. Unlike IPv4 policies, there is no default implicit deny policy. The implicit deny policy should be placed at the bottom of the list of local-in-policies. Local-in-policies are created for each interface, but if you want to create a general implicit deny rule for all interfaces for a specific service, source, address, or destination address, use the any interface.

When you define a local-in policy, if no action is set manually, then the action will default to deny.

For example, to allow only the source subnet 172.16.200.0/24 to ping port1:

```
config firewall address
    edit "172.16.200.0"
        set subnet 172.16.200.0 255.255.255.0
    next
end
config firewall local-in-policy
    edit 2
        set intf "port1"
        set srcaddr "172.16.200.0"
        set dstaddr "all"
        set action accept
        set service "PING"
        set schedule "always"
    next
    edit 3
        set intf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set service "PING"
        set schedule "always"
    next
end
```

To test the configuration:

1. From the PC at 172.16.200.2, start a continuous ping to port1:

   ```
   ping 172.16.200.1 –t
   ```

2. On the FortiGate, enable debug flow:

   ```
   # diagnose debug flow filter proto 1
   # diagnose debug enable
   # diagnose debug flow trace start 10
   ```

3. The output of the debug flow shows that ping traffic coming from the 172.16.200.0 subnet is allowed:

```
# id=65308 trace_id=25 func=print_pkt_detail line=5939 msg="vd-root:0 received a packet
(proto=1, 172.16.200.2:5->172.16.200.1:2048) tun_id=0.0.0.0 from port1. type=8, code=0, id=5,
seq=0."
id=65308 trace_id=25 func=init_ip_session_common line=6121 msg="allocate a new session-
00029409, tun_id=0.0.0.0"
id=65308 trace_id=25 func=__vf_ip_route_input_rcu line=2012 msg="find a route: flag=80000000
gw-0.0.0.0 via root"
id=65308 trace_id=25 func=ip_session_confirm_final line=3189 msg="npu_state=0x0, hook=1"
id=65308 trace_id=26 func=print_pkt_detail line=5939 msg="vd-root:0 received a packet(proto=1,
172.16.200.1:5->172.16.200.2:0) tun_id=0.0.0.0 from local. type=0, code=0, id=5, seq=0."
id=65308 trace_id=26 func=resolve_ip_tuple_fast line=6027 msg="Find an existing session, id-
00029409, reply direction"
id=65308 trace_id=27 func=print_pkt_detail line=5939 msg="vd-root:0 received a packet(proto=1,
172.16.200.2:5->172.16.200.1:2048) tun_id=0.0.0.0 from port1. type=8, code=0, id=5, seq=1."
id=65308 trace_id=27 func=resolve_ip_tuple_fast line=6027 msg="Find an existing session, id-
00029409, original direction"
id=65308 trace_id=28 func=print_pkt_detail line=5939 msg="vd-root:0 received a packet(proto=1,
172.16.200.1:5->172.16.200.2:0) tun_id=0.0.0.0 from local. type=0, code=0, id=5, seq=1."
```

4. From the PC at 172.20.120.13, start a continuous ping to port1:

```
ping 172.16.200.1 -t
```

5. The output of the debug flow shows that ping traffic coming from subnets other than 172.16.200.0 is dropped by local-in policy 3:

```
# id=65308 trace_id=21 func=print_pkt_detail line=5939 msg="vd-root:0 received a packet
(proto=1, 172.20.120.13:1->172.16.200.1:2048) tun_id=0.0.0.0 from port2. type=8, code=0, id=1,
seq=8."
id=65308 trace_id=21 func=init_ip_session_common line=6121 msg="allocate a new session-
0002929d, tun_id=0.0.0.0"
id=65308 trace_id=21 func=__vf_ip_route_input_rcu line=2012 msg="find a route: flag=80000000
gw-0.0.0.0 via root"
id=65308 trace_id=21 func=__iprope_tree_check line=520 msg="gnum-100004, use int hash,
slot=51, len=2"
id=65308 trace_id=21 func=fw_local_in_handler line=545 msg="iprope_in_check() check failed on
policy 3, drop"
```

Additional options

To disable or re-enable the local-in policy, use the `set status {enable | disable}` command.

To dedicate the interface as an HA management interface, use the `set ha-mgmt-intf-only enable` command.

Example:

```
config firewall local-in-policy
    edit 1
        set ha-mgmt-intf-only enable
        set intf port4
```

```
        set srcaddr all
        set dstaddr all
        set service ALL
        set schedule always
        set action accept
        set status enable
    next
end
```

If a user tries to set the HA reserved management interface during the local-in policy an error is generated. Use the `set ha-mgmt-intf-only enable` command to avoid the error.

TTL policies

You can configure a time-to-live (TTL) policy to block attack traffic with high TTLs. This feature only applies to local-in traffic and does not apply to traffic passing through the FortiGate. You can use `srcintf` to set the interface that the local-in traffic hits. See config firewall ttl-policy.

To configure a TTL policy using the CLI:

```
config firewall ttl-policy
    edit <id>
        set status {enable | disable}
        set action {accept | deny}
        set srcintf <interface>
        set srcaddr <source_address> [source_address] ...
        set service <service_name> [service_name] ...
        set schedule <schedule_name>
        set ttl <value/range>
    next
end
```

Internet service as source addresses

An internet service can be used as the source address in a local-in policy. This allows for more flexibility and control when managing local traffic, enhancing network security and efficiency.

```
config firewall local-in-policy
    edit <id>
        set internet-service-src {enable | disable}
        set internet-service-src-name <string>
        set internet-service-src-group <string>
        set internet-service-src-custom <string>
        set internet-service-src-custom-group <string>
        set internet-service-src-negate {enable | disable}
    next
end
```

internet-service-src {enable | disable}	Enable/disable use of Internet Services in source for this local-in policy. If enabled, the source address is not used.
internet-service-src-name <string>	Internet Service source name.
internet-service-src-group <string>	Internet Service source group name.
internet-service-src-custom <string>	Custom Internet Service source name.
internet-service-src-custom-group <string>	Custom Internet Service source group name.
internet-service-src-negate {enable | disable}	When enabled, `internet-service-src` specifies what the service must NOT be.

Logging local traffic per local-in policy

Local traffic logging can be configured for each local-in policy. This enables more precise and targeted logging by focusing on specific local-in policies that are most relevant to your needs.

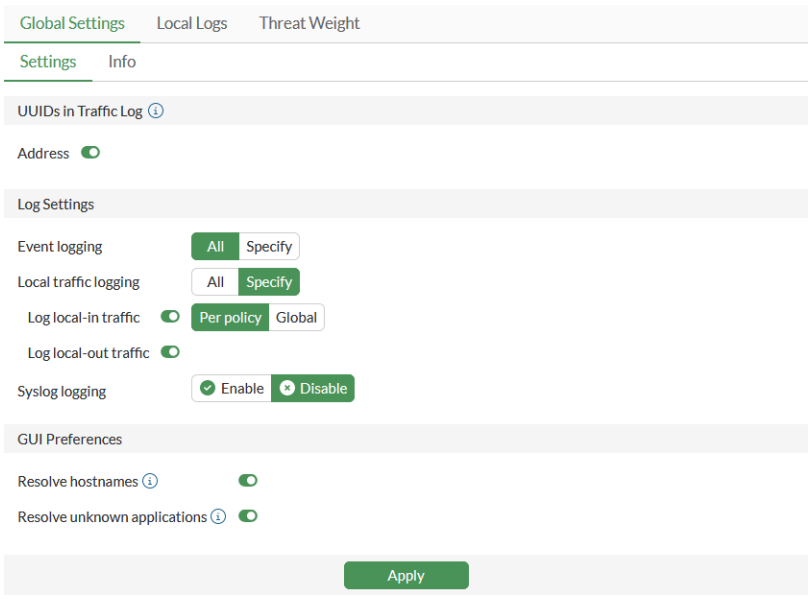
Logging can be configured per local-in policy in the *Log & Report > Log Settings* page or by using the following commands:

```
config log setting
    set local-in-policy-log {enable | disable}
end
config firewall local-in-policy
    edit <id>
        set logtraffic {enable | disable}
    next
end
config firewall local-in-policy6
    edit <id>
        set logtraffic {enable | disable}
    next
end
```

If per policy local-in traffic logging is enabled, the allowed traffic, denied unicast traffic, and denied broadcast traffic logging does not need to be configured for the log settings. When traffic logging is enabled for the local-in policy, the denied unicast traffic and denied broadcast traffic logs will be included.

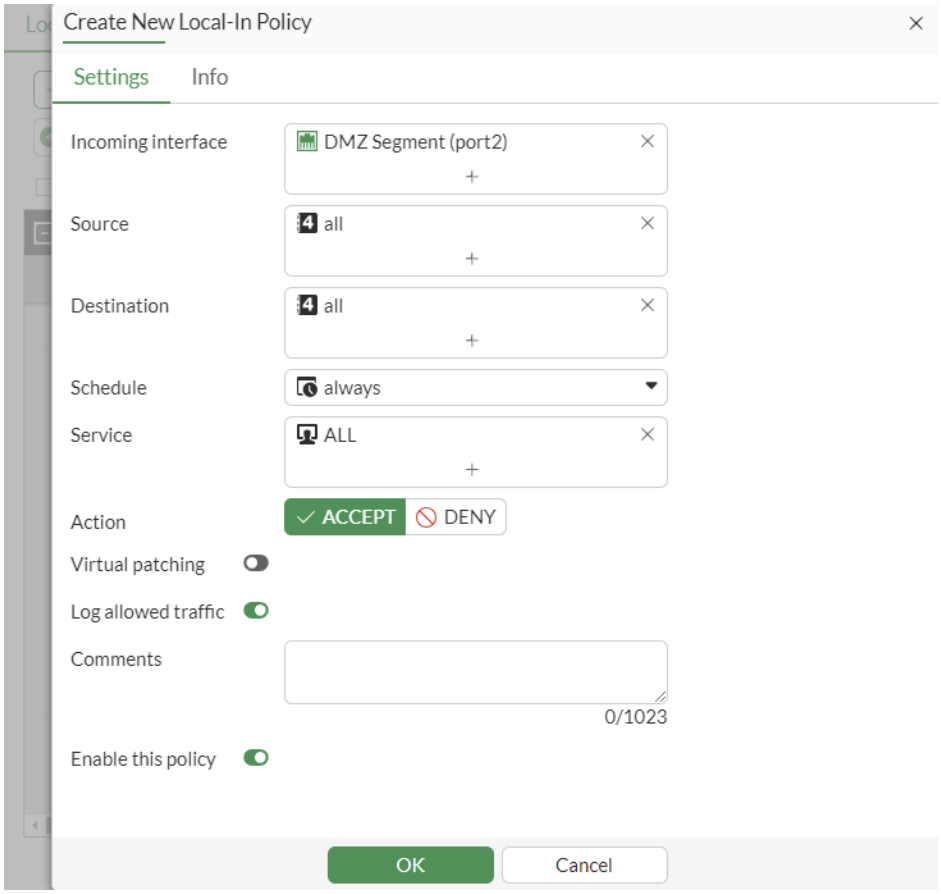
To log local traffic per local-in policy in the GUI:

1. Enable local-in traffic logging per policy:
   a. Go to *Log & Report > Log Settings*.
   b. Go to the *Global Settings* tab.
   c. Set *Local traffic logging* to *Specify*.
   d. Enable *Log local-in traffic* and set it to *Per policy*.

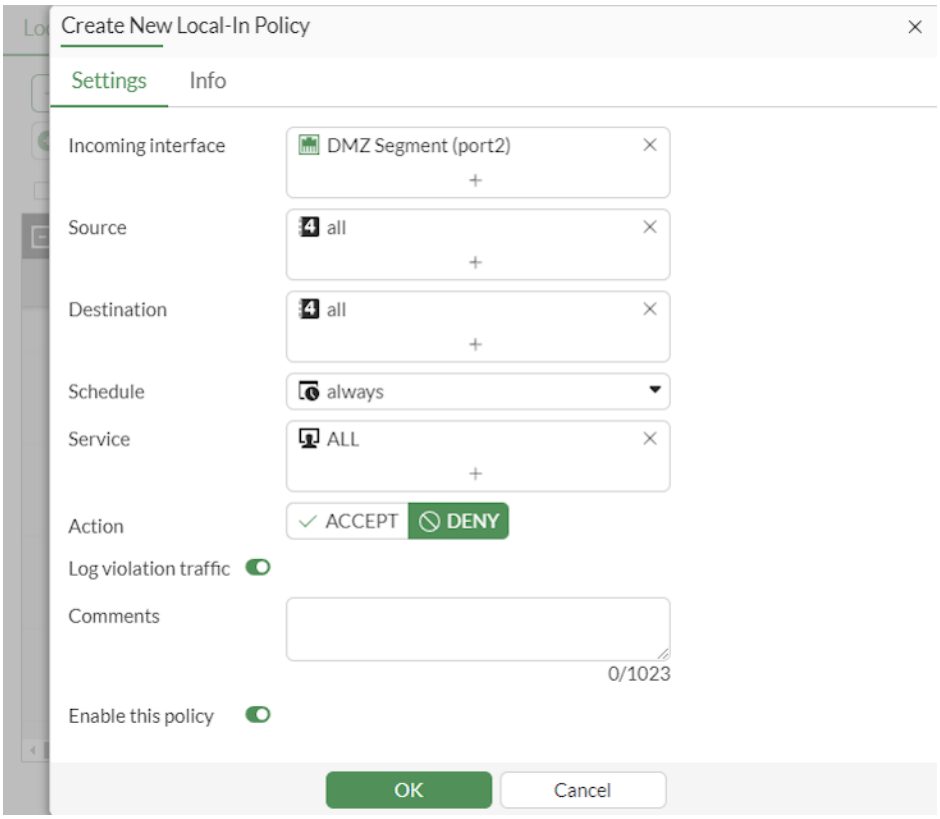


2. Log traffic in a local-in policy:
   a. Go to *Policy & Objects* > *Local-In Policy*.
   b. Select whether you want to configure a *Local-In Policy* or *IPv6 Local-In Policy*.
   c. Create a new policy or edit an existing policy.
   d. Enable traffic logging:

- For policies with the *Action* set to *ACCEPT*, enable *Log allowed traffic*.



- For policies with the *Action* set to *DENY*, enable *Log violation traffic*.



If traffic logging is enabled in the local-in policy, log denied unicast traffic and log denied broadcast traffic logs will display in *Log & Report > Local Traffic*.

To log local traffic per local-in policy in the CLI:

1. Enable logging local-in traffic per policy:

```
config log setting
    set local-in-policy-log enable
end
```

2. Enable local traffic logging in the policy:

```
config firewall local-in-policy
    edit 1
        set logtraffic enable
    next
end
config firewall local-in-policy6
    edit 2
        set logtraffic enable
    next
end
```

3. Review local logs. Traffic is logged for the local-in policies and denied unicast traffic and denied broadcast traffic logs are included.

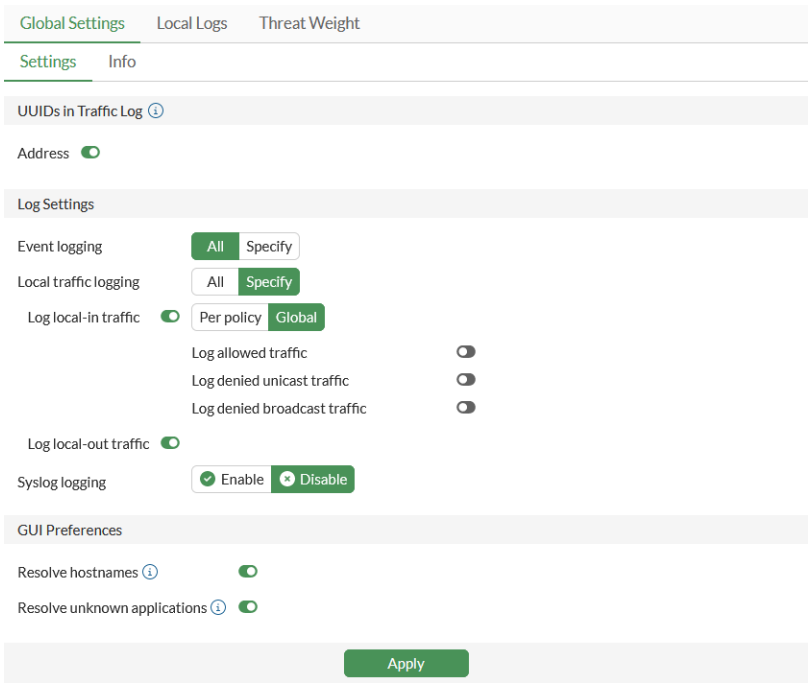
```
date=2024-04-17 time=12:32:02 eventtime=1713382322595245756 tz="-0700" logid="0001000014"
type="traffic" subtype="local" level="notice" vd="vdom1" srcip=2000:10:1:100::11
identifier=28808 srcintf="port1" srcintfrole="undefined" dstip=2000:172:16:200::5
dstintf="vdom1" dstintfrole="undefined" sessionid=317 proto=58 action="deny" policyid=1
policytype="local-in-policy6" poluuid="a07ee45a-fcf0-51ee-7740-806252a3d358"
srccountry="Reserved" dstcountry="Reserved" service="PING6" trandisp="noop" app="PING6"
duration=0 sentbyte=0 rcvdbyte=0 sentpkt=0 rcvdpkt=0 appcat="unscanned" crscore=5
craction=262144 crlevel="low" msg="Connection Failed"

date=2024-04-17 time=14:32:15 eventtime=1713389535257923782 tz="-0700" logid="0001000014"
type="traffic" subtype="local" level="notice" vd="vdom1" srcip=10.1.100.11 identifier=5747
srcintf="port1" srcintfrole="undefined" dstip=172.16.200.5 dstintf="vdom1"
dstintfrole="undefined" srccountry="Reserved" dstcountry="Reserved" sessionid=11134 proto=1
action="deny" policyid=1 policytype="local-in-policy" poluuid="c9b22eda-fc51-51ee-c116-
1f07e0898ea3" service="TIMESTAMP" trandisp="noop" app="TIMESTAMP" duration=0 sentbyte=0
rcvdbyte=0 sentpkt=0 rcvdpkt=0 appcat="unscanned" crscore=5 craction=262144 crlevel="low"
msg="Connection Failed"

date=2024-04-17 time=14:35:29 eventtime=1713389729517452423 tz="-0700" logid="0001000014"
type="traffic" subtype="local" level="notice" vd="vdom1" srcip=10.1.100.11 identifier=29479
srcintf="port1" srcintfrole="undefined" dstip=10.1.100.255 dstintf="unknown-0"
dstintfrole="undefined" replysrcintf="vdom1" srccountry="Reserved" dstcountry="Reserved"
sessionid=11180 proto=1 action="deny" policyid=1 policytype="local-in-policy"
poluuid="c9b22eda-fc51-51ee-c116-1f07e0898ea3" service="PING" trandisp="noop" app="PING"
duration=0 sentbyte=0 rcvdbyte=0 sentpkt=0 rcvdpkt=0 appcat="unscanned" msg="Connection
Failed"
```

Global logging

Local-in traffic can also continue to be logged globally instead of per policy. To configure global local-in traffic logging in the CLI, disable `local-in-policy-log`.

To configure global local traffic logging in the GUI:

1. Enable local-in traffic logging per policy:
   a. Go to *Log & Report > Log Settings*.
   b. Go to the *Global Settings* tab.
   c. Set *Local traffic logging* to *Specify*.
   d. Enable *Log local-in traffic* and set it to *Global*.



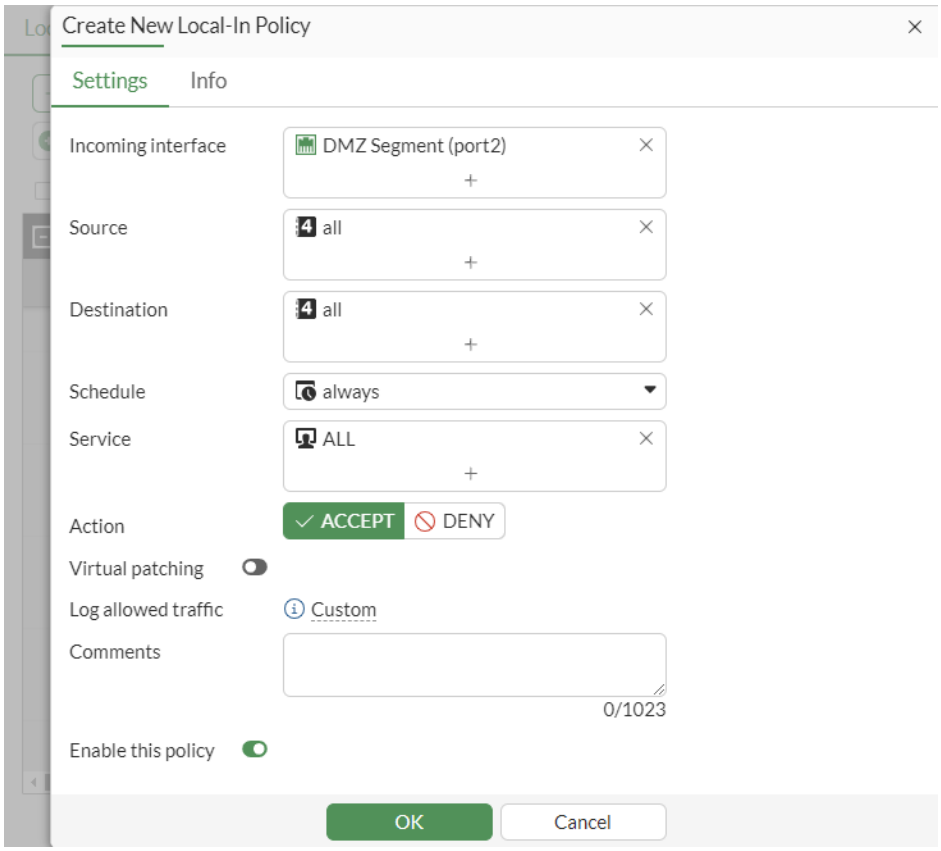
e. Enable the log events that should be globally logged.

2. Review logging in a local-in policy:

a. Go to *Policy & Objects* > *Local-In Policy*.

b. Select whether you want to configure a *Local-In Policy* or *IPv6 Local-In Policy*.

c. Create a new policy or edit an existing policy.



Log allowed traffic and *Log violation traffic* will be set to *Custom* and cannot be changed in the policy.

DoS policy

A Denial of Service (DoS) policy examines network traffic arriving at a FortiGate interface for anomalous patterns, which usually indicates an attack.

A denial of service occurs when an attacking system starts an abnormally large number of sessions with a target system. The large number of sessions slows down or disables the target system, preventing legitimate users from using it.

DoS policies are checked before security policies, preventing attacks from triggering more resource intensive security protection and slowing down the FortiGate.

DoS anomalies

Predefined sensors are setup for specific anomalous traffic patterns. New DoS anomalies cannot be added by the user.

The predefined anomalies that can be used in DoS policies are:

Anomaly	Description	Recommended Threshold
tcp_syn_flood	If the SYN packet rate of new TCP connections, including retransmission, to one destination IP address exceeds the configured threshold value, the action is executed. *	2000 packets per second.
tcp_port_scan	If the SYN packet rate of new TCP connections, including retransmission, from one source IP address exceeds the configured threshold value, the action is executed.	1000 packets per second.
tcp_src_session	If the number of concurrent TCP connections from one source IP address exceeds the configured threshold value, the action is executed.	5000 concurrent sessions.
tcp_dst_session	If the number of concurrent TCP connections to one destination IP address exceeds the configured threshold value, the action is executed.	5000 concurrent sessions.
udp_flood	If the UDP traffic to one destination IP address exceeds the configured threshold value, the action is executed.	2000 packets per second.
udp_scan	If the UDP sessions setup rate originating from one source IP address exceeds the configured threshold value, the action is executed.	2000 sessions per second.
udp_src_session	If the number of concurrent UDP connections from one source IP address exceeds the configured threshold value, the action is executed.	5000 concurrent sessions.
udp_dst_session	If the number of concurrent UDP connections to one destination IP address exceeds the configured threshold value, the action is executed.	5000 concurrent sessions.
icmp_flood	If the number of ICMP packets sent to one destination IP address exceeds the configured threshold value, the action is executed.	250 packets per second.
icmp_sweep	If the ICMP sessions setup rate originating from one source IP address exceeds the configured threshold value, the action is executed.	100 sessions per second.
icmp_src_session	If the number of concurrent ICMP connections from one source IP address exceeds the configured threshold value, the action is executed.	300 concurrent sessions
icmp_dst_session	If the number of concurrent ICMP connections to one destination IP address exceeds the configured threshold value, the action is executed.	1000 concurrent sessions
ip_src_session	If the number of concurrent IP connections from one source IP address exceeds the configured threshold value, the action is executed.	5000 concurrent sessions.

Anomaly	Description	Recommended Threshold
ip_dst_session	If the number of concurrent IP connections to one destination IP address exceeds the configured threshold value, the action is executed.	5000 concurrent sessions.
sctp_flood	If the number of SCTP packets sent to one destination IP address exceeds the configured threshold value, the action is executed.	2000 packets per second
sctp_scan	If the number of SCTP sessions originating from one source IP address exceeds the configured threshold value, the action is executed.	1000 packets per second
sctp_src_session	If the number of concurrent SCTP connections from one source IP address exceeds the configured threshold value, the action is executed.	5000 concurrent sessions
sctp_dst_session	If the number of concurrent SCTP connections to one destination IP address exceeds the configured threshold value, the action is executed.	5000 concurrent sessions

* The NP7 and NP6XLite have hardware modules that can provide tcp_syn-flood protection using the SYN cookies mechanism. The modules activate the mechanism by sending a reply in the hardware only after the tcp_syn_flood threshold is exceeded. DoS-offloading is required; see DoS policy hardware acceleration for more information.

For thresholds based on the number of concurrent sessions, blocking the anomaly will not allow more than the number of concurrent sessions to be set as the threshold.

For example, if the period for a particular anomaly is 60 seconds, such as those where the threshold is measured in concurrent sessions, after the 60 second timer has expired the number of allowed sessions that match the anomaly criteria is reset to zero. This means that, if you allow 10 sessions through before blocking, after the 60 seconds has elapsed, another 10 sessions will be allowed. The attrition of sessions from expiration should keep the allowed sessions from reaching the maximum.

For rate based thresholds, where the threshold is measured in packets per second, the *Block* action prevents anomalous traffic from overwhelming the firewall in two ways:

- continuous: Block packets once an anomaly is detected, and continue to block packets while the rate is above the threshold. This is the default setting.
- periodical: After an anomaly is detected, allow the configured number of packets per second.

For example, if a DoS policy is configured to block icmp_flood with a threshold of 10pps, and a continuous ping is started at a rate of 20pps for 1000 packets:

- In continuous mode, the first 10 packets are passed before the DoS sensor if triggered, and then the remaining 990 packets are blocked.
- In periodical mode, 10 packets are allowed to pass per second, so 500 packets are blocked in the 50 seconds during which the ping is occuring.

The actual numbers of passed and blocked packets may not be exact, as fluctuations in the rates can occur, but the numbers should be close to the defined threshold.

To configure the block action for rate based anomaly sensors:

```
config ips global
    set anomaly-mode {continuous | periodical}
end
```

DoS policies

A DoS policy can be configured to use one or more anomalies.

To configure a DoS policy in the GUI:

1. Go to *Policy & Objects* > *IPv4 DoS Policy* or *Policy & Objects* > *IPv6 DoS Policy* and click *Create New*.
   If the option is not visible, enable *DoS Policy* in *Feature Visibility*. See Feature visibility on page 3952 for details.
2. Configure the following:

Name	Enter a name for the policy.
Incoming Interface	Enter the interface that the policy applies to.
Source Address	Enter the source address.
Destination Address	Enter the destination address. This is the address that the traffic is addressed to. In this case, it must be an address that is associated with the firewall interface. For example, it could be an interface address, a secondary IP address, or the address assigned to a VIP address.
Service	Select the services or service groups. The ALL service can used or, to optimize the firewall resources, only the services that will be answered on an interface can be used.

L3 Anomalies
L4 Anomalies
	Configure the anomalies:
	• *Logging*: Enable/disable logging for specific anomalies or all of them. Anomalous traffic will be logged when the action is *Block* or *Monitor*.
	• *Action*: Select the action to take when the threshold is reached:
	  • *Disable*: Do not scan for the anomaly.
	  • *Block*: Block the anomalous traffic.
	  • *Monitor*: Allow the anomalous traffic, but record a log message if logging is enabled.
	• *Threshold*: The number of detected instances that triggers the anomaly action.

Comments	Optionally, enter a comment.

3. Enable the policy, then click *OK*.

The quarantine option is only available in the CLI. See Quarantine on page 1906 for information.

To configure a DoS policy in the GUI:

```
config firewall DoS-policy
    edit 1
        set name "Flood"
        set interface "port1"
        set srcaddr "all"
        set dstaddr "all"
        set service "ALL"
        config anomaly
            edit "icmp_flood"
                set status enable
                set log enable
                set action block
                set quarantine attacker
                set quarantine-expiry 1d1h1m
                set quarantine-log enable
                set threshold 100
            next
        end
    next
end
```

`name <string>`	Enter a name for the policy.
`interface <string>`	Enter the interface that the policy applies to.
`srcaddr <string>`	Enter the source address.
`dstaddr <string>`	Enter the destination address. This is the address that the traffic is addressed to. In this case, it must be an address that is associated with the firewall interface. For example, it could be an interface address, a secondary IP address, or the address assigned to a VIP address.
`service <string>`	Enter the services or service groups. The `ALL` service can used or, to optimize the firewall resources, only the services that will be answered on an interface can be used.
`status {enable | disable}`	Enable/disable this anomaly.
`log {enable | disable}`	Enable/disable anomaly logging. When enabled, a log is generated whenever the anomaly action is triggered, regardless of which action is configured.
`action {pass | block}`	Set the action to take when the threshold is reached: • `pass`: Allow traffic, but record a log message if logging is enabled. • `block`: Block traffic if this anomaly is found.

`quarantine {none | attacker}`	Set the quarantine method (see Quarantine on page 1906): • `none`: Disable quarantine. • `attacker`: Block all traffic from the attacker's IP address, and add the attacker's IP address to the banned user list.
`quarantine-expiry <###d##h##m>`	Set the duration of the quarantine, in days, hours, and minutes (###d##h##m) (1m - 364d23h59m, default = 5m). This option is available if `quarantine` is set `attacker`.
`quarantine-log {enable | disable}`	Enable/disable quarantine logging (default = disable). This option is available if `quarantine` is set `attacker`.
`threshold <integer>`	The number of detected instances - packets per second or concurrent session number - that triggers the anomaly action.

Quarantine

Quarantine is used to block any further traffic from a source IP address that is considered a malicious actor or a source of traffic that is dangerous to the network. Traffic from the source IP address is blocked for the duration of the quarantine, and the source IP address is added to the banned user list.

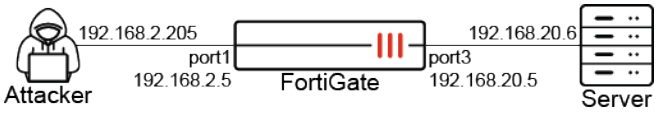
The banned user list is kept in the kernel, and used by Antivirus, Data Loss Prevention (DLP), DoS, and Intrusion Prevention System (IPS). Any policies that use any of these features will block traffic from the attacker's IP address.

To view the quarantined user list:

```
# diagnose user banned-ip list
src-ip-addr       created                  expires                  cause
192.168.2.205     Wed Nov 25 12:47:54 2020 Wed Nov 25 12:57:54 2020 DOS
```

Troubleshooting DoS attacks

The best way to troubleshoot DoS attacks is with Anomaly logs and IPS anomaly debug messages.



To test an icmp_flood attack:

1. From the Attacker, launch an icmp_flood with 50pps lasting for 3000 packets.
2. On the FortiGate, configure continuous mode and create a DoS policy with an icmp_flood threshold of 30pps:

```
config firewall DoS-policy
    edit 1
        set name icmpFlood
        set interface "port1"
```

```
        set srcaddr "all"
        set dstaddr "all"
        set service "ALL"
        config anomaly
            edit "icmp_flood"
                set status enable
                set log enable
                set action block
                set threshold 30
            next
        end
    next
end
```

3. Configure the debugging filter:

```
# diagnose ips anomaly config
DoS sensors in kernel vd 0:
DoS id 1 proxy 0
  0 tcp_syn_flood status 0 log 0 nac 0 action 0 threshold 2000
  ...
  7 udp_dst_session status 0 log 0 nac 0 action 0 threshold 5000
  8 icmp_flood status 1 log 1 nac 0 action 7 threshold 30
  9 icmp_sweep status 0 log 0 nac 0 action 0 threshold 100
  ...
total # DoS sensors: 1.
```

```
# diagnose ips anomaly filter id 8
```

4. Launch the icmp_flood from a Linux machine. This example uses Nmap:

```
$ sudo nping --icmp --rate 50 -c 3000 192.168.2.50
SENT (0.0522s) ICMP [192.168.2.205 > 192.168.2.50 Echo request (type=8/code=0) id=8597 seq=1]
IP [ttl=64 id=47459 iplen=28 ]
...
Max rtt: 11.096ms | Min rtt: 0.028ms | Avg rtt: 1.665ms
Raw packets sent: 3000 (84.000KB) | Rcvd: 30 (840B) | Lost: 2970 (99.00%)
Nping done: 1 IP address pinged in 60.35 seconds
```

5. During the attack, check the anomaly list on the FortiGate:

```
# diagnose ips anomaly list
list nids meter:
id=icmp_flood         ip=192.168.2.50 dos_id=1 exp=998 pps=46 freq=50

total # of nids meters: 1.
```

id=icmp_flood	The anomaly name.
ip=192.168.2.50	The IP address of the host that triggered the anomaly. It can be either the client or the server.

	For icmp_flood, the IP address is the destination IP address. For icmp_sweep, it would be the source IP address.
dos_id=1	The DoS policy ID.
exp=998	The time to be expired, in jiffies (one jiffy = 0.01 seconds).
pps=46	The number of packets that had been received when the diagnose command was executed.
freq=50	For session based anomalies, freq is the number of sessions. For packet rate based anomalies (flood, scan): • In continuous mode: freq is the greater of pps, or the number of packets received in the last second. • In periodic mode: freq is the pps.

6. Go to *Log & Report > Security Events* and download the *Anomaly* logs:

```
date=2020-11-20 time=14:38:39 eventtime=1605911919824184594 tz="-0800" logid="0720018433"
type="utm" subtype="anomaly" eventtype="anomaly" level="alert" vd="root" severity="critical"
srcip=192.168.2.205 srccountry="Reserved" dstip=192.168.2.50 srcintf="port1"
srcintfrole="undefined" sessionid=0 action="clear_session" proto=1 service="PING" count=1307
attack="icmp_flood" icmpid="0x2195" icmptype="0x08" icmpcode="0x00" attackid=16777316
policyid=1 policytype="DoS-policy" ref="http://www.fortinet.com/ids/VID16777316" msg="anomaly:
icmp_flood, 31 > threshold 30, repeats 28 times" crscore=50 craction=4096 crlevel="critical"
```

```
date=2020-11-20 time=14:39:09 eventtime=1605911949826224056 tz="-0800" logid="0720018433"
type="utm" subtype="anomaly" eventtype="anomaly" level="alert" vd="root" severity="critical"
srcip=192.168.2.205 srccountry="Reserved" dstip=192.168.2.50 srcintf="port1"
srcintfrole="undefined" sessionid=0 action="clear_session" proto=1 service="PING" count=1497
attack="icmp_flood" icmpid="0x2195" icmptype="0x08" icmpcode="0x00" attackid=16777316
policyid=1 policytype="DoS-policy" ref="http://www.fortinet.com/ids/VID16777316" msg="anomaly:
icmp_flood, 50 > threshold 30, repeats 1497 times" crscore=50 craction=4096 crlevel="critical"
```

Analysis

In the first log message:

msg="anomaly: icmp_flood, 31 > threshold 30	At the beginning of the attack, a log is recorded when the threshold of 30pps is broken.
repeats 28 times	The number of packets that has exceeded the threshold since the last time a log was recorded.
srcip=192.168.2.205 **dstip=192.168.2.50**	The source and destination IP addresses of the attack.
action="clear_session"	Equivalent to block. If `action` was set to `monitor` and logging was enabled, this would be `action="detected"`.

In the second log message:

- Because it is an ongoing attack, the FortiGate generates one log message for multiple packets every 30 seconds..
- It will not generate a log message if:
  - The same attack ID happened more than once in a five second period, or
  - The same attack ID happened more than once in a 30 second period and the actions are the same and have the same source and destination IP addresses.

msg="anomaly: icmp_flood, 50 > threshold 30	In the second before the log was recorded, 50 packets were detected, exceeding the configured threshold.
repeats 1497 times	The number of packets that has exceeded the threshold since the last time a log was recorded

Telemetry policy

This topic applies only to a FortiGate acting as a FortiTelemetry Controller with *FortiTelemetry* enabled on the *System > Feature Visibility* page. For a summary of FortiTelemetry-related tasks that you can do in FortiOS, see FortiTelemetry on page 139. See also *FortiTelemetry Administration Guide*.

A telemetry type of firewall policy must be created on the FortiTelemetry Controller to send monitoring tasks to the agents.

The telemetry firewall policy has type set to *Telemetry*, source set to telemetry addresses or telemetry address groups, and a telemetry profile selected. FortiTelemetry Controller uses the following firewall policy configuration elements to automatically create a monitor task for agents:

- FortiTelemetry agent from the policy source
- Applications to monitor from the telemetry profile

FortiOS automatically places the telemetry policy at the top of the policy table for you. This position allows the policy to function correctly.

You must also allow the monitor traffic from FortiGate.

To configure a telemetry policy in the GUI:

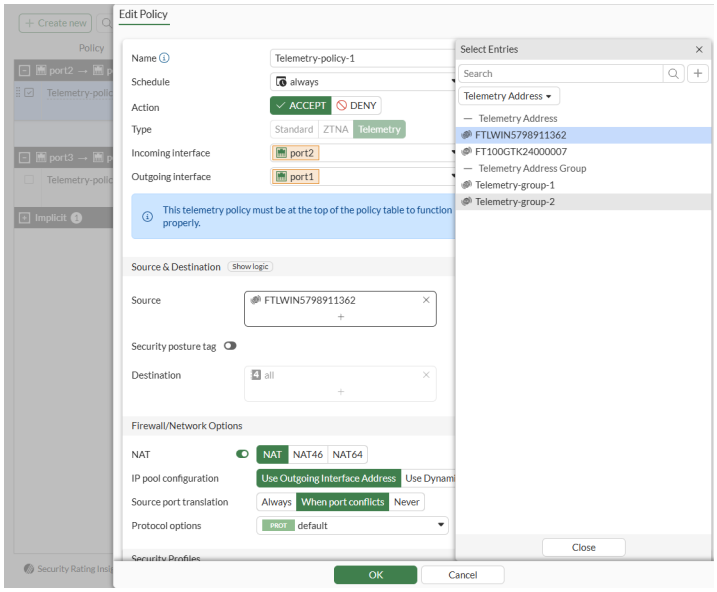
1. Go to *Policy & Objects > Firewall Policy*.
2. Click *Create New* to create a new firewall policy.
3. Configure the following settings:

Name	Enter a name for the policy.
Action	Set the action to *Accept* to send monitoring tasks to the agents.
Type	Select *Telemetry* as the policy type.
Incoming Interface	Choose the FortiGate port that is used to connect to the FortiTelemetry agent(s).
Outgoing Interface	Select the outgoing interface.

4. Set *Source* to one or more telemetry addresses or address groups.

See Telemetry addresses and address groups on page 2083 for more information.

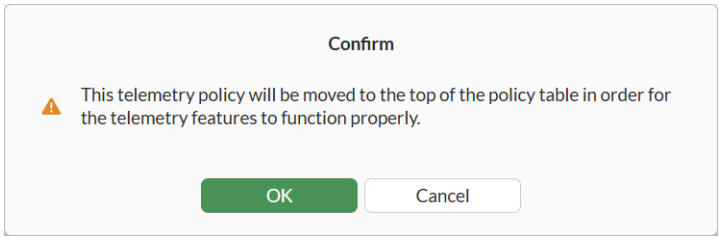
a. Click *Source*, and select *Telemetry Address* to display a list of telemetry addresses and telemetry address groups.



b. Select one or more addresses or address groups.

c. Click *Close* when done.

5. Set *Telemetry* to a telemetry security profile configured on the FortiGate.
   See Telemetry profile on page 2424 for more information.
6. Click *OK* to save the policy. A message is displayed:



7. Click *OK* to continue. The policy is saved and displayed at the top of the policy table.
   The *Type* column displays *Telemetry*.

To configure a FortiTelemetry policy in the CLI:

1. Use the following commands to configure a telemetry policy in the CLI:

```
config firewall policy
  edit "3"
    show
      config firewall policy
        edit 3
          set name "FTLPolicyWin136"
          set uuid 2a86e78c-d9c9-51ef-33d5-6ddd81cfcc63
          set srcintf "port3"
          set dstintf "port1"
```

```
            set action accept
            set srcaddr "FTLWIN4665500001"
            set dstaddr "all"
            set schedule "always"
            set service "ALL"
            set telemetry-profile "default"
            set logtraffic all
            set nat enable
            set comments "FTLPolicyWin136"
        next
    end
```

Access control lists

An access control list (ACL) is a granular, targeted blocklist that is used to block IPv4 and IPv6 packets on a specified interface based on the criteria configured in the ACL policy.

On FortiGate models with ports that are connected through an internal switch fabric with TCAM capabilities, ACL processing is offloaded to the switch fabric and does not use CPU resources. VLAN interfaces that are based on physical switch fabric interfaces are also supported. Interfaces that are connected through an internal switch fabric usually have names prefixed with *port* or *lan*, such as *port1* or *lan2*; other interfaces are not supported.

The packets will be processed by the CPU when offloading is disabled or not possible, such as when a port on a supported model does not connect to the internal fabric switch.

For a list of models that support ACL see config firewall acl.

Example

To block all IPv4 and IPv6 telnet traffic from port2 to Company_Servers:

```
config firewall acl
    edit 1
       set interface "port2"
       set srcaddr "all"
       set dstaddr "Company_Servers"
       set service "TELNET"
    next
end
config firewall acl6
    edit 1
        set interface "port2"
        set srcaddr "all"
        set dstaddr "Company_Servers_v6"
        set service "TELNET"
    next
end
```

Diagnose commands

To check the number of packets dropped by an ACL:

```
# diagnose firewall acl counter
ACL id 1 dropped 0 packets
```

```
# diagnose firewall acl counter6
ACL id 2 dropped 0 packets
```

To clear the packet drop counters:

```
# diagnose firewall acl clearcounter
```

```
# diagnose firewall acl clearcounter6
```

Interface policies

Interface policies are implemented before the security policies and are only flow-based. They are configured in the CLI.

This feature allows you to attach a set of IPS policies with the interface instead of the forwarding path, so packets can be delivered to IPS before entering the firewall. This feature is used for following IPS deployments:

- One-Arm: By defining interface policies with IPS and DoS anomaly checks and enabling sniff-mode on the interface, the interface can be used for one-arm IDS.
- IPv6 IPS: IPS inspection can be enabled through interface IPv6 policy.
- Scan traffic that is destined to the FortiGate.
- Scan and log traffic that are silently dropped or flooded by Firewall or Multicast traffic.

IPS sensors can be assigned to an interface policy. Both incoming and outgoing packets are inspected by IPS sensor (signature).

To configure an interface policy:

```
config firewall interface-policy
    edit 1
        set status enable
        set comments 'test interface policy #1'
        set logtraffic utm
        set interface "port2"
        set srcaddr all
        set dstaddr all
        set service "ALL"
        set application-list-status disable
        set ips-sensor-status disable
        set dsri disable
        set av-profile-status enable
```

```
        set av-profile default
        set webfilter-profile-status disable
    next
end
```

Source NAT

Network Address Translation (NAT) is the process that enables a single device, such as a router or firewall, to act as an agent between the internet or public network and a local or private network. This agent acts in real-time to translate the source or destination IP address of a client or server on the network interface. Source IP translation enables a single, public address to represent a significantly larger number of private addresses. Destination IP translation enables the firewall to translate a public, destination address to a private address. So we don't have to configure a real public IP address for the server deployed in a private network.

NAT can be subdivided into two types:

- Source NAT (SNAT)
- Destination NAT (DNAT)

This section is about SNAT. Three NAT working modes are supported: static SNAT, dynamic SNAT, and central SNAT. For information about DNAT, see Destination NAT on page 1939.

The following topics provide instructions on configuring policies with source NAT:

- Static SNAT on page 1913
- Dynamic SNAT on page 1914
- Central SNAT on page 1924
- Configuring an IPv6 SNAT policy on page 1935
- SNAT policies with virtual wire pairs on page 1937
- Configuring PCP port mapping with SNAT and DNAT on page 2014

Static SNAT

In static SNAT all internal IP addresses are always mapped to the same public IP address. This is a port address translation, Since we have 60416 available port numbers, this one public IP address can handle the conversion of 60,416 internal IP addresses to the same service, where a service is defined by a specified protocol, destination IP address, and destination port.

Internal Source IP	Source Port	Translated Source IP	Translated Source Port
10.1.100.1	11110	172.16.200.1	5117
10.1.100.1	11111	172.16.200.1	5118
10.1.100.2	11112	172.16.200.1	5119
.....	.....	172.16.200.1	.....
.....	.....	172.16.200.1	65533

FortiGate firewall configurations commonly use the Outgoing Interface address.

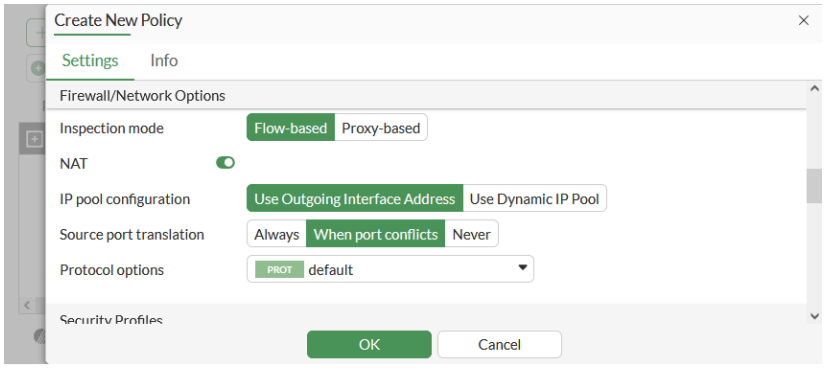
Sample configuration

The following example of static SNAT uses an internal network with subnet 10.1.100.0/24 (vlan20) and an external/ISP network with subnet 172.16.200.0/24 (vlan30).

When the clients in internal network need to access the servers in external network, We need to translate IP addresses from 10.1.100.0/24 to an IP address 172.16.200.0/24, In this example, we implement static SNAT by creating a firewall policy.

To configure static NAT:

1. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
2. Configure the required policy parameters.
3. Enable *NAT* and select *Use Outgoing Interface Address*. For packets that match this policy, its source IP address is translated to the IP address of the outgoing interface.
4. The default setting for *Source Port Translation* is *When Port Conflicts*, which uses the original source port unless there is a conflict. If required, set it to *Always* to always change the source port for Source NAT, or *Never* to prevent Source NAT from changing a session's source port.



5. Click *OK*.

Dynamic SNAT

Dynamic SNAT maps the private IP addresses to the first available public address from a pool of addresses. In the FortiGate firewall, this can be done by using IP pools. IP pools is a mechanism that allows sessions leaving the FortiGate firewall to use NAT. An IP pool defines a single IP address or a range of IP addresses to be used as the source address for the duration of the session. These assigned addresses are used instead of the IP address assigned to that FortiGate interface.

IP pool types

FortiGate uses four types of IPv4 IP pools. This topic focuses on some of the differences between them.

Overload

This type of IP pool is similar to static SNAT mode. We need to define an external IP range that contains one or more IP addresses. When there is only one IP address it is almost the same as static SNAT, the outgoing

interface address is used. When it contains multiple IP addresses, it is equivalent to an extended mode of static SNAT.

For instance, if we define an overload type IP pool with two external IP addresses (172.16.200.1—172.16.200.2), since there are 60,416 available port numbers per IP, this IP pool can handle 60,416*2 internal IP addresses to the same service, where a service is defined by a specific protocol, destination IP address, and destination port.

Original Source IP	Original Source Port	Translated Source IP	Translated Source Port
10.1.100.1	11110	172.16.200.1	5117
10.1.100.2	11111	172.16.200.1	5118
.....	.....	172.16.200.1	.....
.....	.....	172.16.200.1	65533
.....	.....	172.16.200.2	5117
.....	.....	.....	.....
.....	.....	172.16.200.2	65533

The mapped IP address can be calculated from the source IP address. The index number of the address in the pool is the remainder of the source IP address, in decimal, divided by the number addresses in the pool.

To calculate the decimal value of the source IP address, either use an online calculator, or use the following equation:

$$a.b.c.d = a * (256)^3 + b * (256)^2 + c * (256) + d$$

For example:

$$192.168.0.1 = 192 * (256)^3 + 168 * (256)^2 + 0 * (256) + 1 = 3232235521$$

If there is one IP pool, where:

- P_1 = the first address in the IP pool
- R_1 = the number of IP addresses in the IP pool
- X = the source IP address as a decimal number
- Y = the mapped IP address

Then the equation to determine the mapped address is:

$$Y = P_1 + X \bmod R_1$$

For example:

IP pool	Source IP address
172.26.73.20 to 172.26.73.90	192.168.1.200

1. Convert the source IP address to a decimal number:
$$192 * (256)^3 + 168 * (256)^2 + 1 * (256) + 200 = 3232235976$$
2. Determine the number of IP addresses in the pool:
$$172.26.73.90 - 172.26.73.20 = 71$$
3. Find the remainder of the source IP address divided by the number of addresses in the pool:
$$3232235976 \bmod 71 = 26$$
4. Add the remainder to the first IP address in the pool:
$$172.26.73.20 + 26 = 172.26.73.46$$

So, the mapped IP address is **172.26.73.46**.

If there are multiple IP pools, the calculation is similar to when there is only one pool.

If there are two IP pools, where:

- P_1 = the first address in the first IP pool
- P_2 = the first address in the second IP pool
- R_1 = the number of IP addresses in the first IP pool
- R_2 = the number of IP addresses in the second IP pool
- X = the source IP address as a decimal number
- Y = the mapped IP address

Then the equations to determine the mapped address are:

$$\text{If } X \bmod (R_1 + R_2) >= R_1, \text{ then } Y = P_2 + X \bmod R_2$$

$$\text{If } X \bmod (R_1 + R_2) < R_1, \text{ then } Y = P_1 + X \bmod R_1$$

For example:

IP pools	Source IP address
pool01: 172.26.73.20 to 172.26.73.90	192.168.1.200
pool02: 172.26.75.50 to 172.26.75.150	

1. Convert the source IP address to a decimal number:
$$192 * (256)^3 + 168 * (256)^2 + 1 * (256) + 200 = 3232235976$$
2. Determine the total number of IP addresses in the pools:
$$(172.26.73.90 - 172.26.73.20) + (172.26.75.50 - 172.26.75.150) = 71 + 101 = 172$$
3. Find the remainder of the source IP address divided by the number of addresses in the pools:
$$3232235976 \bmod 172 = 108$$
4. The remainder is greater than the number of addresses in pool01, so the address is selected from pool02 and the remainder is recalculated based only on pool02:
$$3232235976 \bmod 101 = 40$$
5. Add the new remainder to the first IP address in pool02:
$$172.26.75.50 + 40 = 172.26.75.90$$
So, the mapped IP address is **172.26.75.90**.

One-to-one

This type of IP pool means that the internal IP address and the external (translated) IP address match one-to-one. The port address translation (PAT) is disabled when using this type of IP pool. For example, if we define a one-to-one type IP pool with two external IP addresses (172.16.200.1 - 172.16.200.2), this IP pool only can handle two internal IP addresses.

Fixed port range

For the overload and one-to-one IP pool types, we do not need to define the internal IP range. For the fixed port range type of IP pool, we can define both internal IP range and external IP range. Since each external IP address and the number of available port numbers is a specific number, if the number of internal IP addresses is also

determined, we can calculate the port range for each address translation combination. So we call this type fixed port range. This type of IP pool is a type of port address translation (PAT).

For instance, if we define one external IP address (172.16.200.1) and ten internal IP addresses (10.1.100.1-10.1.100.10), we have translation IP+Port combination like following table:

Original Source IP	Original Source Port	Translated Source IP	Translated Source Port Range
10.1.100.1	.....	172.16.200.1	5117~11157
10.1.100.2	.....	172.16.200.1	11158~17198
10.1.100.3	.....	172.16.200.1	.....
10.1.100.4	.....	172.16.200.1	.....
10.1.100.5	.....	172.16.200.1	.....
10.1.100.6	.....	172.16.200.1	.....
10.1.100.7	.....	172.16.200.1	.....
10.1.100.8	.....	172.16.200.1	.....
10.1.100.9	.....	172.16.200.1	53445~59485
10.1.100.10	.....	172.16.200.1	59486~65526

Custom port ranges, NAT64, and full cone NAT are supported. See Full cone NAT for fixed port range IP pools on page 2028 for more information about full cone NAT.

Port block allocation

This type of IP pool is also a type of port address translation (PAT). It gives users a more flexible way to control the way external IPs and ports are allocated. Users need to define *Block Size/Block Per User* and external IP range. *Block Size* means how many ports each Block contains. *Block per User* means how many blocks each user (internal IP) can use.

The following is a simple example:

- **External IP Range**: 172.16.200.1—172.16.200.1
- **Block Size**: 128
- **Block Per User**: 8

Result:

- **Total-PBAs**: 472 (60416/128)
- **Maximum ports can be used per User (Internal IP Address)**: 1024 (128*8)
- **How many Internal IP can be handled**: 59 (60416/1024 or 472/8)

Custom port ranges are supported.

A randomized port selection algorithm is also supported. The algorithm makes the port allocation process less predictable, which enhances security. The `port-random` option is available with:

- The `config firewall central-snat-map` command:

```
config firewall central-snat-map
    edit 1
        set port-preserve disable
        set port-random {enable | disable}
    next
end
```

- The `config firewall policy` command:

```
config firewall policy
    edit 1
        set port-preserve disable
        set port-random {enable | disable}
    next
end
```

`port-random {enable | disable}`	Enable/disable random source port selection for source NAT (default = disable). Available when `port-preserve` is disabled.

Interim logs can be configured for port block allocation (PBA) NAT logging. This enables continuous access to PBA event logs during an ongoing session, and provides comprehensive logging throughout a session's lifespan.

PBA event logs are generated periodically based on the configured time interval:

```
config firewall ippool
    edit pba-ippool
        set type port-block-allocation
        set pba-interim-log <integer>
    next
end
```

For example, when the PBA interim log interval is set to 600 seconds, event logs are obtained every ten minutes:

- Configure the PBA IP pool with a time interval:

```
config firewall ippool
    edit "pba-ippool"
        set type port-block-allocation
        set startip 172.16.200.151
        set endip 172.16.200.151
        set block-size 64
        set num-blocks-per-user 1
        set pba-interim-log 600
    next
end
```

- Check the event logs:

```
# execute log display

2 logs found.

2 logs returned.

1: date=2024-02-04 time=13:34:04 eventtime=1707082444264865326 tz="-0800" logid="0100022024"
type="event" subtype="system" level="notice" vd="vdom1" logdesc="IP pool PBA interim log"
action="ippool-interim" saddr="10.1.100.42" nat=172.16.200.151 portbegin=5117 portend=5180
poolname="pba-ippool" duration=1200 msg="IPpool interim"

2: date=2024-02-04 time=13:24:03 eventtime=1707081844204865060 tz="-0800" logid="0100022024"
type="event" subtype="system" level="notice" vd="vdom1" logdesc="IP pool PBA interim log"
```

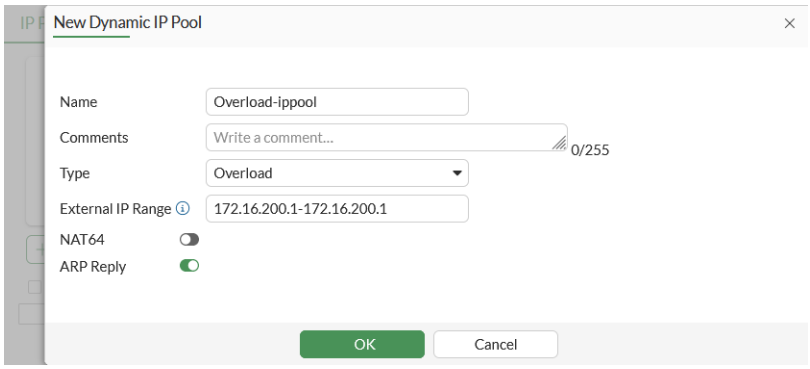
```
action="ippool-interim" saddr="10.1.100.42" nat=172.16.200.151 portbegin=5117 portend=5180
poolname="pba-ippool" duration=600 msg="IPpool interim"
```

Sample configuration

When an IP pool object is created with *ARP Reply* enabled, the object does not need to be referenced in any policies before a FortiGate interface starts responding to ARP requests for the addresses in the IP pool.

To configure overload IP pool in the GUI:

1. In *Policy & Objects* > *IP Pools*, click *IP Pool*.
2. Click *Create new*.
3. Set *Type* to *Overload*.
4. Enter the external IP range separated by a hyphen (*172.16.200.1-172.16.200.1*).



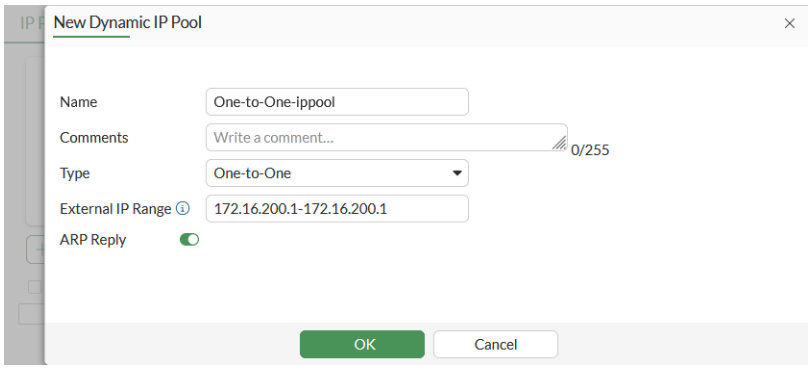
5. Click *OK*.

To configure overload IP pool in the CLI:

```
config firewall ippool
    edit "Overload-ippool"
        set startip 172.16.200.1
        set endip 172.16.200.1
    next
end
```

To configure one-to-one IP pool using the GUI:

1. In *Policy & Objects* > *IP Pools*, click *IP Pool*.
2. Click *Create new*.
3. Set *Type* to *One-to-One*.
4. Enter the external IP range separated by a hyphen (*172.16.200.1-172.16.200.2*).



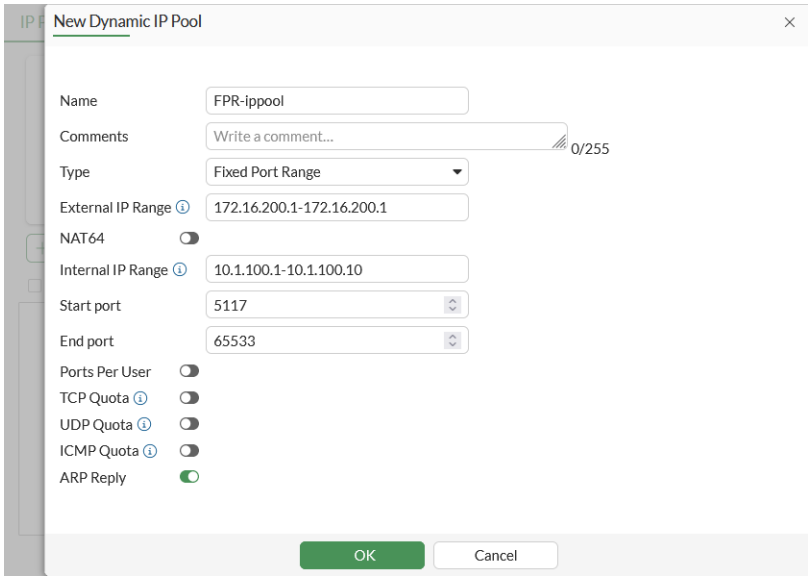
5. Click *OK*.

To configure one-to-one IP pool in the CLI:

```
config firewall ippool
    edit "One-to-One-ippool"
        set type one-to-one
        set startip 172.16.200.1
        set endip 172.16.200.2
    next
end
```

To configure fixed port range IP pool in the GUI:

1. In *Policy & Objects* > *IP Pools*, click *IP Pool*.
2. Click *Create new*.
3. Set *Type* to *Fixed Port Range*.
4. Enter the external IP range separated by a hyphen *172.16.200.1-172.16.200.1*).
5. Enter the internal IP range separated by a hyphen *10.1.100.1-10.1.100.10*).
6. (Optional) Enter the start port and end port numbers.



7. Click *OK*.

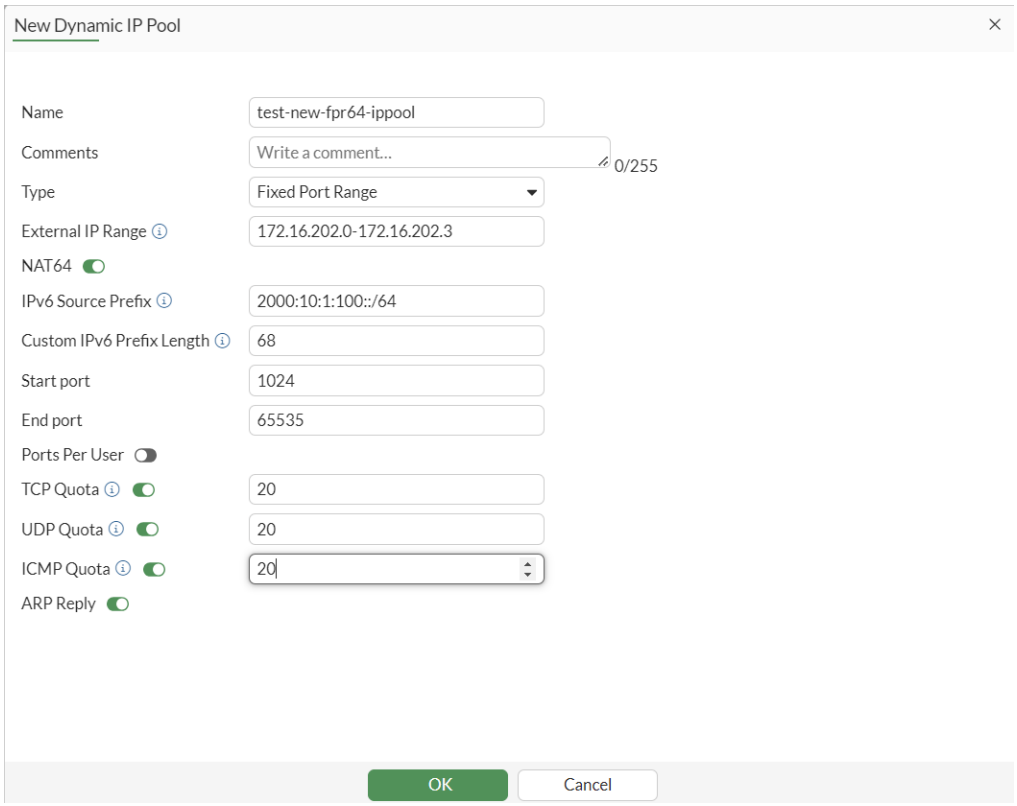
To configure fixed port range IP pool in the CLI:

```
config firewall ippool
    edit "FPR-ippool"
        set type fixed-port-range
        set startip 172.16.200.1
        set endip 172.16.200.1
        set startport 5117
        set endport 65533
        set source-startip 10.1.100.1
        set source-endip 10.1.100.10
   next
end
```

To configure NAT64 FPR IP pools in the GUI:

1. Go to *Policy & Objects > IP Pools*.
2. Select the *IP Pool* tab and click *Create New*.
3. Set the following options:

Name	Enter a name.
Type	Select *Fixed Port Range*.
External IP Range	Enter the IP address range.
NAT64	Enable.
IPv6 Source Prefix	Enter the source IPv6 network to be translated.
Custom IPv6 Prefix Length	Enter the subnet length of a single deterministic NAT64 client.



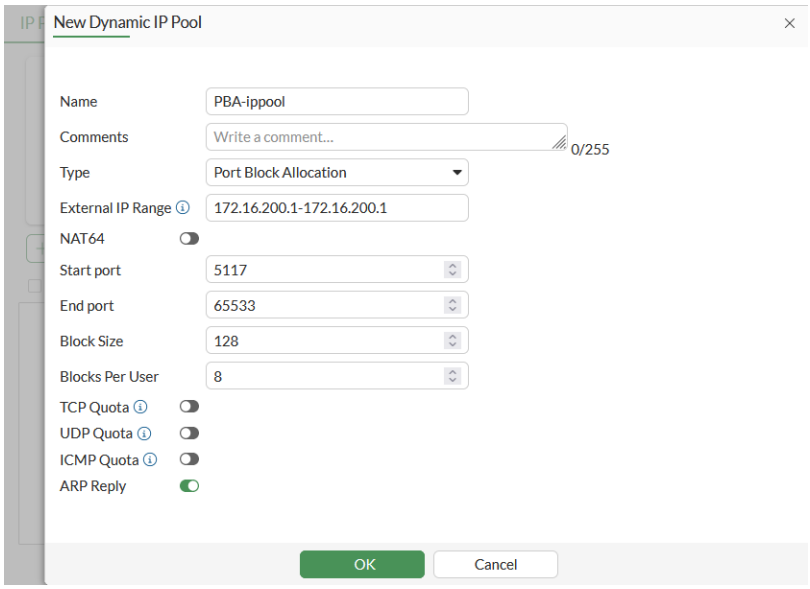
4. Set the remaining options as needed and click *OK*.

To configure NAT64 FPR IP pools in the CLI:

```
config firewall ippool
    edit "test-new-fpr64-ippool"
        set type fixed-port-range
        set startip 172.16.202.0
        set endip 172.16.202.3
        set startport 1024
        set endport 65535
        set nat64 enable
        set source-prefix6 2000:10:1:100::/64
        set client-prefix-length 68
        set tcp-session-quota 20
        set udp-session-quota 20
        set icmp-session-quota 10
    next
end
```

To configure port block allocation IP pool in the GUI:

1. In *Policy & Objects > IP Pools*, click *IP Pool*.
2. Click *Create new*.
3. Set *Type* to *Port Block Allocation*.
4. Enter the external IP range separated by a hyphen *172.16.200.1-172.16.200.1*).



5. Click *OK*.

To configure port block allocation IP pool in the CLI:

```
config firewall ippool
    edit PBA-ippool
        set type port-block-allocation
        set startip 172.16.200.1
        set endip 172.16.200.1
        set startport 5117
        set endport 65533
        set block-size 128
        set num-blocks-per-user 8
    next
end
```

IP pools and VIPs as local IP addresses

IP pools and VIPs are considered local IP addresses if responding to ARP requests on these external IP addresses is enabled (`set arp-reply enable`, by default). In this case, the FortiGate is considered a destination for those IP addresses and can receive reply traffic at the application layer successfully.

However, as a side-effect, once an IP pool or VIP has been configured, even if it is never used in a firewall policy, the FortiGate considers it as a local address and will not forward traffic based on the routing table. Therefore, any unused IP pools or VIPs should be deleted to prevent any unexpected behavior.

> ⚠ For a history of behavior changes related to IP pools and VIPs, see Technical Tip: IP pool and virtual IP behavior changes in FortiOS 6.4, 7.0, 7.2, and 7.4.

Central SNAT

The central SNAT table enables you to define and control (with more granularity) the address translation performed by FortiGate. With the NAT table, you can define the rules for the source address or address group, and which IP pool the destination address uses.

FortiGate reads the NAT rules from the top down until it hits a matching rule for the incoming address. This enables you to create multiple NAT policies that dictate which IP pool is used based on source address, destination address, and source port. NAT policies can be rearranged within the policy list. NAT policies are applied to network traffic after a security policy.

The central SNAT table allows you to create, edit, delete, and clone central SNAT entries.

Central SNAT notes

- The central NAT feature is not enabled by default.
- If central NAT is enabled, the NAT option under IPv4 policies is skipped and SNAT must be done via `central-snat-map`. The firewall policy list and dialog boxes have messages and redirection links to show this information.
- If NGFW mode is policy-based, then it is assumed that central NAT (specifically SNAT) is enabled implicitly.

Sample configuration

To enable central SNAT from the GUI:

1. In *System > Settings > General*, under *System Operations Settings*, enable *Central SNAT*.
2. Click *Apply*.

To enable or disable central SNAT using the CLI:

```
config system settings
   set central-nat {enable | disable}
end
```

When central NAT is enabled, *Policy & Objects* displays the Central SNAT section.

The Central SNAT policy has many options:

Field	Description
Type	Specify whether you are performing SNAT on IPv4 or IPv6. This option only appears when IPv6 is enabled under *Feature Visibility*.
Incoming Interface	Specify one or more interfaces for the ingress traffic.
Outgoing Interface	Specify one or more interfaces for the egress traffic.
Source Address	Specify the address or address group of the source.
Destination Address	Specify the address or address group of the destination.

Field	Description
NAT	Enable or disable to perform NAT. When disabled, no source address translation will occur.
IP Pool Configuration	Use outgoing interface address: • Use the address of the outgoing interfaces as source address. Use Dynamic IP Pool: • Choose an IP Pool to perform source NAT.
Protocol	Choose from any, TCP, UDP, SCTP, or specify the protocol number to match. For example, for ICMP, click *specify* with the protocol number 1.
Explicit port mapping	Enable in order to match this NAT policy only when the following ports are a match: • Choose an original source port from one to 65535. NAT'd port will be chosen by the FortiGate based on the IP Pool configuration. Explicit port mapping cannot apply to some protocols which do not use ports, such as ICMP. When enabling a NAT policy which uses Explicit port mapping, always consider that ICMP traffic will not match this policy. When using IP Pools, only the Overload type IP Pool allows Explicit port mapping. When Explicit port mapping is applied, you must define an original source port range and a translated sort port range. The source port will map one to one with the translated port. Refer to Dynamic SNAT to understand how each IP Pool type works.
Comments	Enter comments for this NAT policy.
Enable this policy	Enable or disable this policy.

To configure central SNAT using the CLI:

```
config firewall central-snat-map
    edit <policyID number>
        set status {enable | disable}
        set orig-addr <valid address object preconfigured on the FortiGate>
        set srcintf <name of interface on the FortiGate>
        set dst-addr <valid address object preconfigured on the FortiGate>
        set dstintf <name of interface on the FortiGate>
        set protocol <integer for protocol number>
        set dst-port <integer for destination port or port range>
        set orig-port <integer for original port number>
        set nat-port <integer for translated port number>
        set comments <string>
    next
end
```

Setting the destination port for traffic matching is available when the protocols are TCP, UDP, or SCTP.

The following examples demonstrate configuring central SNAT:

- Example one: Apply SNAT to all traffic on page 1926
- Example two: Apply an IP pool to all TCP traffic on page 1926
- Example three: Apply an IP pool to all traffic with a specific original port range on page 1928
- Example four: Create two central SNAT rules on page 1929
- Example five: Fine-tuning source port behavior on page 1931

IP address external feed objects can also be used as the source address of central SNAT policies. See Apply external feed connectors as source addresses in central SNAT on page 4732.

Example one: Apply SNAT to all traffic

Apply SNAT to all traffic from port2 to port3.

To configure from the CLI:

```
config firewall central-snat-map
    edit 1
        set srcintf "port3"
        set dstintf "port2"
        set orig-addr "all"
        set dst-addr "all"
    next
end
```

Example two: Apply an IP pool to all TCP traffic

Apply an IP pool to all traffic from port3 to port2 that are TCP. NAT all other traffic using the outgoing interface IP.

To configure from the CLI:

```
config firewall ippool
    edit "Overload-IPPOOL"
        set startip 192.168.2.201
        set endip 192.168.2.202
    next
end
config firewall central-snat-map
    edit 1
        set srcintf "port3"
        set dstintf "port2"
        set orig-addr "all"
        set dst-addr "all"
        set protocol 6
        set nat-ippool "Overload-IPPOOL"
    next
    edit 2
        set srcintf "port3"
```

```
        set dstintf "port2"
        set orig-addr "all"
        set dst-addr "all"
    next
end
```

To collect session table output from the CLI:

```
diagnose sys session list
```

The TCP session (protocol 6) is NAT'd with Overload-IPPOOL to 192.168.2.201:

```
session info: proto=6 proto_state=05 duration=14 expire=0 timeout=3600 flags=00000000 socktype=0
sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty
statistic(bytes/packets/allow_err): org=860/7/1 reply=555/8/1 tuples=2
tx speed(Bps/kbps): 60/0 rx speed(Bps/kbps): 38/0
orgin->sink: org pre->post, reply pre->post dev=9->6/6->9 gwy=192.168.2.1/192.168.0.10
hook=post dir=org act=snat 192.168.0.10:49531->23.57.57.114:443(192.168.2.201:61776)
hook=pre dir=reply act=dnat 23.57.57.114:443->192.168.2.201:61776(192.168.0.10:49531)
pos/(before,after) 0/(0,0), 0/(0,0)
dst_mac=04:d5:90:5f:a2:2a
misc=0 policy_id=2 auth_info=0 chk_client_info=0 vd=0
serial=00011065 tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id=00000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x040000
```

A UDP session (protocol 17) is NAT'd to the outgoing interface IP address 192.168.2.86:

```
session info: proto=17 proto_state=01 duration=16 expire=163 timeout=0 flags=00000000 socktype=0
sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ helper=dns-udp vlan_cos=0/255
state=may_dirty
statistic(bytes/packets/allow_err): org=59/1/1 reply=187/1/1 tuples=2
tx speed(Bps/kbps): 3/0 rx speed(Bps/kbps): 11/0
orgin->sink: org pre->post, reply pre->post dev=9->6/6->9 gwy=192.168.2.1/192.168.0.10
hook=post dir=org act=snat 192.168.0.10:52177->4.2.2.1:53(192.168.2.86:61770)
hook=pre dir=reply act=dnat 4.2.2.1:53->192.168.2.86:61770(192.168.0.10:52177)
dst_mac=04:d5:90:5f:a2:2a
misc=0 policy_id=2 auth_info=0 chk_client_info=0 vd=0
serial=00011061 tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id=00000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x040000
```

Example three: Apply an IP pool to all traffic with a specific original port range

Apply an IP Pool to all traffic from port3 to port2 that have a specific original port range, mapping the ports to the same NAT'd port range. Nat all other traffic using the outgoing interface IP.

To configure from the CLI:

```
config firewall central-snat-map
    edit 1
        set srcintf "port3"
        set dstintf "port2"
        set orig-addr "all"
        set dst-addr "all"
        set orig-port 50000-65535
        set nat-ippool "Overload-IPPOOL"
        set nat-port 50000-65535
    next
    edit 2
        set srcintf "port3"
        set dstintf "port2"
        set orig-addr "all"
        set dst-addr "all"
    next
end
```

To collect session table output from the CLI:

```
diagnose sys session list
```

Traffic with original port in the range between 50000-65535 will be NAT'd with the Overload type IP Pool. The mapped port is in the same port range:

```
session info: proto=17 proto_state=01 duration=3 expire=176 timeout=0 flags=00000000 socktype=0
sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ helper=dns-udp vlan_cos=0/255
state=may_dirty
statistic(bytes/packets/allow_err): org=71/1/1 reply=123/1/1 tuples=2
tx speed(Bps/kbps): 23/0 rx speed(Bps/kbps): 40/0
orgin->sink: org pre->post, reply pre->post dev=9->6/6->9 gwy=192.168.2.1/192.168.0.10
hook=post dir=org act=snat 192.168.0.10:52540->4.2.2.1:53(192.168.2.201:52540)
hook=pre dir=reply act=dnat 4.2.2.1:53->192.168.2.201:52540(192.168.0.10:52540)
dst_mac=04:d5:90:5f:a2:2a
misc=0 policy_id=2 auth_info=0 chk_client_info=0 vd=0
serial=00011399 tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id=00000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x040000
```

Traffic with original port outside the range of 50000-65535 will be NAT'd to the outgoing interface IP:

```
session info: proto=6 proto_state=01 duration=3 expire=3597 timeout=3600 flags=00000000 socktype=0
sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty
statistic(bytes/packets/allow_err): org=2262/10/1 reply=2526/11/1 tuples=2
tx speed(Bps/kbps): 741/5 rx speed(Bps/kbps): 828/6
orgin->sink: org pre->post, reply pre->post dev=9->6/6->9 gwy=192.168.2.1/192.168.0.10
hook=post dir=org act=snat 192.168.0.10:49805->142.250.68.66:443(192.168.2.86:62214)
hook=pre dir=reply act=dnat 142.250.68.66:443->192.168.2.86:62214(192.168.0.10:49805)
pos/(before,after) 0/(0,0), 0/(0,0)
dst_mac=04:d5:90:5f:a2:2a
misc=0 policy_id=2 auth_info=0 chk_client_info=0 vd=0
serial=0001139a tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id=00000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x040000
```

Protocols which do not use ports, such as ICMP, will be NAT'd to the outgoing interface IP:

```
session info: proto=1 proto_state=00 duration=7 expire=59 timeout=0 flags=00000000 socktype=0
sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty
statistic(bytes/packets/allow_err): org=480/8/1 reply=480/8/1 tuples=2
tx speed(Bps/kbps): 66/0 rx speed(Bps/kbps): 66/0
orgin->sink: org pre->post, reply pre->post dev=9->6/6->9 gwy=192.168.2.1/192.168.0.10
hook=post dir=org act=snat 192.168.0.10:1->4.2.2.1:8(192.168.2.86:62209)
hook=pre dir=reply act=dnat 4.2.2.1:62209->192.168.2.86:0(192.168.0.10:1)
dst_mac=04:d5:90:5f:a2:2a
misc=0 policy_id=2 auth_info=0 chk_client_info=0 vd=0
serial=0001138b tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id=00000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x040000
```

Example four: Create two central SNAT rules

In the following example, two central SNAT rules will be created:

- Rule 3 will have a destination port set and IP pool `test-ippool4-3` applied.
- Rule 5 will have IP pool `test-ippool4-1` applied but will not set the destination port.

Example traffic will then be passed to see how the rule is matched.

To test central SNAT rule destination port support:

1. Configure central SNAT rule 3 with the destination port range specified:

```
config firewall ippool
    edit "test-ippool4-3"
        set startip 172.16.200.150
        set endip 172.16.200.150
    next
end
config firewall central-snat-map
    edit 3
        set srcintf "port24"
        set dstintf "port17"
        set orig-addr "all"
        set dst-addr "all"
        set protocol 6
        set nat-ippool "test-ippool4-3"
        set dst-port 80-443
    next
end
```

2. Configure central SNAT rule 5:

```
config firewall ippool
    edit "test-ippool4-1"
        set startip 172.16.200.151
        set endip 172.16.200.151
    next
end
config firewall central-snat-map
    edit 5
        set srcintf "port24"
        set dstintf "port17"
        set orig-addr "all"
        set dst-addr "all"
        set nat-ippool "test-ippool4-1"
    next
end
```

3. Send HTTP traffic to pass through the FortiGate that is expected to match central SNAT rule 3. IP pool `test-ippool4-3` will perform source NAT.
4. Check the session to review for expected behavior:

```
# diagnose sys session list
session info: proto=6 proto_state=01 duration=2 expire=3599 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty f00
statistic(bytes/packets/allow_err): org=1800/31/1 reply=77304/60/1 tuples=2
```

```
tx speed(Bps/kbps): 602/4 rx speed(Bps/kbps): 25854/206
orgin->sink: org pre->post, reply pre->post dev=24->17/17->24 gwy=172.16.200.55/10.1.100.42
hook=post dir=org act=snat 10.1.100.42:46731->172.16.200.55:80(172.16.200.150:46731)
hook=pre dir=reply act=dnat 172.16.200.55:80->172.16.200.150:46731(10.1.100.42:46731)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=99 pol_uuid_idx=15864 auth_info=0 chk_client_info=0 vd=0
serial=00003c37 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000001 no_offload
no_ofld_reason:  disabled-by-policy
total session 1
```

5. Send PING traffic to pass through the FortiGate that is expected to match central SNAT rule 5. IP pool `test-ippool4-1` will perform source NAT.
6. Check the session to review for expected behavior:

```
# diagnose sys session list
session info: proto=1 proto_state=00 duration=2 expire=59 timeout=0 flags=00000000 socktype=0
sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty f00
statistic(bytes/packets/allow_err): org=252/3/1 reply=252/3/1 tuples=2
tx speed(Bps/kbps): 99/0 rx speed(Bps/kbps): 99/0
orgin->sink: org pre->post, reply pre->post dev=24->17/17->24 gwy=172.16.200.55/10.1.100.42
hook=post dir=org act=snat 10.1.100.42:36732->172.16.200.55:8(172.16.200.151:36732)
hook=pre dir=reply act=dnat 172.16.200.55:36732->172.16.200.151:0(10.1.100.42:36732)
misc=0 policy_id=99 pol_uuid_idx=15864 auth_info=0 chk_client_info=0 vd=0
serial=00003f62 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000001 no_offload
no_ofld_reason:  disabled-by-policy
total session 1
```

Example five: Fine-tuning source port behavior

FortiOS supports maintaining or altering the original source port in SNAT using the `port-preserve` command:

- When `port-preserve` is enabled, SNAT will use the original source port if it is not already in use. This is the default.
- When `port-preserve` is disabled, SNAT will always change the source port to use the next higher, available port in the range. When the highest available port is reached, the counter will roll back to the first available port in the range. This allows ports to remain free until the counter rolls back to them.

The `port-preserve` command is available for the central SNAT or for firewall policies when NAT is enabled.

Only ports within the source port range of 5117 to 65533 will be preserved. Anything below 5117 will be translated to a port higher than 5117 based on the internal SNAT source port algorithm. If your source port is less than 5117 and you want to preserve it, explicit port mapping must be used. For more information, see Explicit port mapping on page 1925.

To configure source port behavior for central SNAT:

```
config firewall central-snat-map
    edit 1
        set port-preserve {enable | disable}
    next
end
```

To preserve the original source port in a firewall policy:

1. Enable original source port preservation in the policy:

```
config firewall policy
    edit 2
        set srcintf "port7"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
        set port-preserve enable
    next
end
```

2. Check the session after the first traffic passes through the FortiGate:

```
# diagnose sys session list
session info: proto=6 proto_state=01 duration=7 expire=3594 timeout=3600 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty src-vis
statistic(bytes/packets/allow_err): org=165/3/1 reply=112/2/1 tuples=2
tx speed(Bps/kbps): 21/0 rx speed(Bps/kbps): 14/0
orgin->sink: org pre->post, reply pre->post dev=15->9/9->15 gwy=0.0.0.0/10.2.2.1
hook=post dir=org act=snat 10.1.100.42:20042->172.16.200.155:2156(172.16.200.199:5162)
hook=pre dir=reply act=dnat 172.16.200.155:2156->172.16.200.199:5162(10.1.100.42:20042)
po/(before,after) 0/(0,0), 0/(0,0)
src_mac=94:ff:3c:6e:d2:90  dst_mac=00:0c:29:3d:83:02
misc=0 policy_id=2 pol_uuid_idx=16000 auth_info=0 chk_client_info=0 vd=1
serial=0001cf04 tos=ff/ff app_list=0 app=0 url_cat=0
```

```
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  mac-host-check disabled-by-policy
total session: 1
```

SNAT uses source port 5162.

3. Clear the old session.
4. Send traffic again with the same source port from the client.
5. Check the new session:

```
# diagnose sys session list
session info: proto=6 proto_state=01 duration=4 expire=3598 timeout=3600 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty src-vis
statistic(bytes/packets/allow_err): org=165/3/1 reply=112/2/1 tuples=2
tx speed(Bps/kbps): 41/0 rx speed(Bps/kbps): 28/0
orgin->sink: org pre->post, reply pre->post dev=15->9/9->15 gwy=0.0.0.0/10.2.2.1
hook=post dir=org act=snat 10.1.100.42:20042->172.16.200.155:2156(172.16.200.199:5162)
hook=pre dir=reply act=dnat 172.16.200.155:2156->172.16.200.199:5162(10.1.100.42:20042)
pos/(before,after) 0/(0,0), 0/(0,0)
src_mac=94:ff:3c:6e:d2:90  dst_mac=00:0c:29:3d:83:02
misc=0 policy_id=2 pol_uuid_idx=16000 auth_info=0 chk_client_info=0 vd=1
serial=0001d0bf tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  mac-host-check disabled-by-policy
total session: 1
```

The same source port has been used.

To alter the original source port in a firewall policy:

1. Disable original source port preservation in the policy:

```
config firewall policy
    edit 2
        set srcintf "port7"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
        set port-preserve disable
    next
end
```

2. Check the session after the first traffic passes through the FortiGate:

```
# diagnose sys session list
session info: proto=6 proto_state=05 duration=34 expire=113 timeout=3600 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty
statistic(bytes/packets/allow_err): org=269/5/1 reply=164/3/1 tuples=2
tx speed(Bps/kbps): 4/0 rx speed(Bps/kbps): 2/0
orgin->sink: org pre->post, reply pre->post dev=15->9/9->15 gwy=0.0.0.0/10.2.2.1
hook=post dir=org act=snat 10.1.100.42:20042->172.16.200.155:2156(172.16.200.199:5149)
hook=pre dir=reply act=dnat 172.16.200.155:2156->172.16.200.199:5149(10.1.100.42:20042)
pos/(before,after) 0/(0,0), 0/(0,0)
src_mac=94:ff:3c:6e:d2:90  dst_mac=00:0c:29:3d:83:02
misc=0 policy_id=2 pol_uuid_idx=16000 auth_info=0 chk_client_info=0 vd=1
serial=0004a004 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
total session: 1
```

   SNAT uses source port 5149.
3. Clear the old session.
4. Send traffic again with the same source port from the client.
5. Check the new session:

```
# diagnose sys session list
session info: proto=6 proto_state=01 duration=3 expire=3597 timeout=3600 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty
statistic(bytes/packets/allow_err): org=165/3/1 reply=112/2/1 tuples=2
tx speed(Bps/kbps): 49/0 rx speed(Bps/kbps): 33/0
orgin->sink: org pre->post, reply pre->post dev=15->9/9->15 gwy=0.0.0.0/10.2.2.1
hook=post dir=org act=snat 10.1.100.42:20042->172.16.200.155:2156(172.16.200.199:5151)
hook=pre dir=reply act=dnat 172.16.200.155:2156->172.16.200.199:5151(10.1.100.42:20042)
pos/(before,after) 0/(0,0), 0/(0,0)
src_mac=94:ff:3c:6e:d2:90  dst_mac=00:0c:29:3d:83:02
misc=0 policy_id=2 pol_uuid_idx=16000 auth_info=0 chk_client_info=0 vd=1
serial=0004a1a5 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
total session: 1
```

   A new source port has been used.
6. Clear the old session again.

7. Send traffic again with the same source port from the client.
8. Check the new session:

```
# diagnose sys session list
session info: proto=6 proto_state=01 duration=20 expire=3581 timeout=3600 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty
statistic(bytes/packets/allow_err): org=165/3/1 reply=112/2/1 tuples=2
tx speed(Bps/kbps): 8/0 rx speed(Bps/kbps): 5/0
orgin->sink: org pre->post, reply pre->post dev=15->9/9->15 gwy=0.0.0.0/10.2.2.1
hook=post dir=org act=snat 10.1.100.42:20042->172.16.200.155:2156(172.16.200.199:5153)
hook=pre dir=reply act=dnat 172.16.200.155:2156->172.16.200.199:5153(10.1.100.42:20042)
pos/(before,after) 0/(0,0), 0/(0,0)
src_mac=94:ff:3c:6e:d2:90  dst_mac=00:0c:29:3d:83:02
misc=0 policy_id=2 pol_uuid_idx=16000 auth_info=0 chk_client_info=0 vd=1
serial=0004a519 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
total session: 1
```

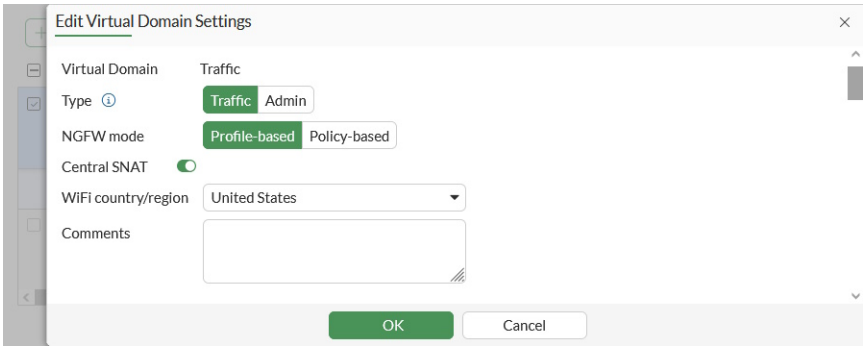
Another new source port has been used.

Configuring an IPv6 SNAT policy

IPv4 and IPv6 central SNAT maps are displayed in the same table.

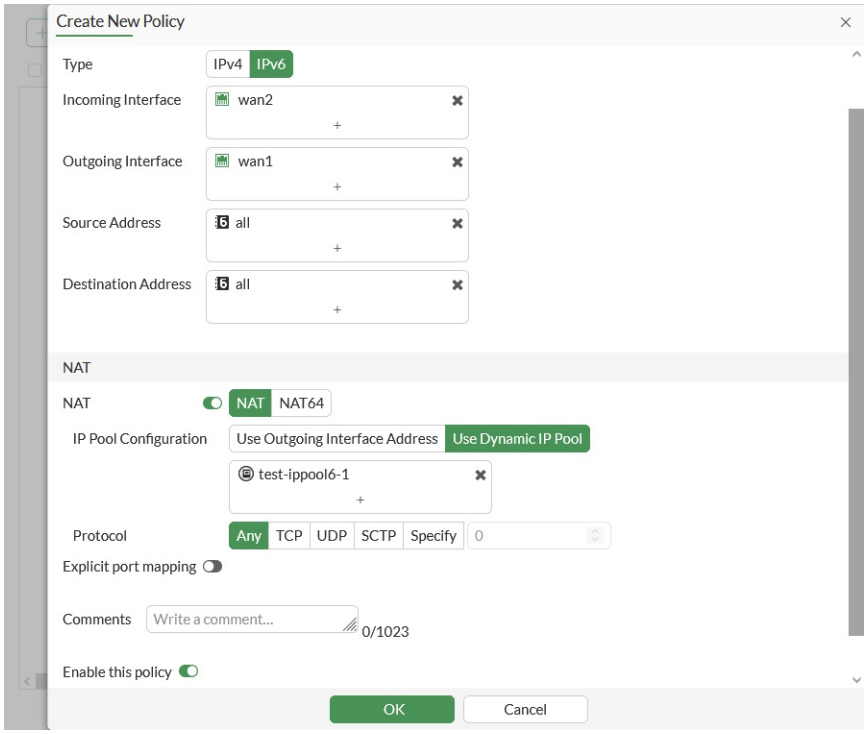
To configure an IPv6 policy with central SNAT in the GUI:

1. Enable central SNAT:
   a. In the Global VDOM, go to *System* > *VDOM*.
   b. Select a VDOM and click *Edit*. The *Edit Virtual Domain Settings* pane opens.
   c. Enable *Central SNAT*.



   d. Click *OK*.

2. In the VDOM with central SNAT enabled (FG-traffic in this example), go to *Policy & Objects* > *Central SNAT* and click *Create New*.
3. Configure the policy settings:
   a. For *Type*, select *IPv6*.
   b. Enter the interface, address, and IP pool information.
   c. Configure the remaining settings as needed.



   d. Click *OK*.
   The matching SNAT traffic will be handled by the IPv6 central SNAT map.

To configure an IPv6 policy with central SNAT in the CLI:

1. Enable central SNAT:

```
config vdom
    edit FG-traffic
        config system settings
            set central-nat enable
        end
    next
end
```

2. Create an IPv6 central SNAT policy:

```
config vdom
    edit FG-traffic
        config firewall central-snat-map
            edit 2
                set type ipv6
```

```
                set srcintf "wan2"
                set dstintf "wan1"
                set orig-addr6 "all"
                set dst-addr6 "all"
                set nat-ippool6 "test-ippool6-1"
            next
        end
    next
end
```

3. Verify the SNAT traffic:

```
(FG-traffic) # diagnose sniffer packet any icmp6 4
interfaces=[any]
filters=[icmp6]
3.602891 wan2 in 2000:10:1:100::41 -> 2000:172:16:200::55: icmp6: echo request seq 0
3.602942 wan1 out 2000:172:16:200::199 -> 2000:172:16:200::55: icmp6: echo request seq 0
3.603236 wan1 in 2000:172:16:200::55 -> 2000:172:16:200::199: icmp6: echo reply seq 0
3.603249 wan2 out 2000:172:16:200::55 -> 2000:10:1:100::41: icmp6: echo reply seq 0
4.602559 wan2 in 2000:10:1:100::41 -> 2000:172:16:200::55: icmp6: echo request seq 1
4.602575 wan1 out 2000:172:16:200::199 -> 2000:172:16:200::55: icmp6: echo request seq 1
4.602956 wan1 in 2000:172:16:200::55 -> 2000:172:16:200::199: icmp6: echo reply seq 1
4.602964 wan2 out 2000:172:16:200::55 -> 2000:10:1:100::41: icmp6: echo reply seq 1
^C
8 packets received by filter
0 packets dropped by kernel
```

SNAT policies with virtual wire pairs

Source NAT (SNAT) can be configured in IPv4 and IPv6 policies with virtual wire pair (VWP) interfaces, and between VWP interfaces when central NAT is enabled.

To configure a policy using SNAT and a VWP interface when central NAT is disabled:

1. Create the VWP interface:

```
config system virtual-wire-pair
    edit "test-vw-1"
        set member "port1" "port4"
    next
end
```

2. Create the IP pool. The IP pool must have a different subnet than the VWP peers.

```
config firewall ippool
    edit "vwp-pool-1"
        set startip 172.16.222.99
        set endip 172.16.222.100
    next
end
```

3. Configure the firewall policy:

```
config firewall policy
    edit 88
        set srcintf "port4"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set nat enable
        set ippool enable
        set poolname "vwp-pool-1"
    next
end
```

4. Verify the IP pool functions as expected and traffic passes through:

```
# diagnose sniffer packet any icmp 4
interfaces=[any]
filters=[icmp]
23.438095 port4 in 172.16.200.11 -> 172.16.200.156: icmp: echo request
23.438126 port1 out 172.16.222.100 -> 172.16.200.156: icmp: echo request
23.438492 port1 in 172.16.200.156 -> 172.16.222.100: icmp: echo reply
23.438501 port4 out 172.16.200.156 -> 172.16.200.11: icmp: echo reply
24.439305 port4 in 172.16.200.11 -> 172.16.200.156: icmp: echo request
24.439319 port1 out 172.16.222.100 -> 172.16.200.156: icmp: echo request
24.439684 port1 in 172.16.200.156 -> 172.16.222.100: icmp: echo reply
24.439692 port4 out 172.16.200.156 -> 172.16.200.11: icmp: echo reply

8 packets received by filter
0 packets dropped by kernel
```

To configure a SNAT between VWP interfaces when central NAT is enabled:

1. Enable central NAT:

```
config system settings
    set central-nat enable
end
```

2. Create the VWP interface:

```
config system virtual-wire-pair
    edit "test-vw-1"
        set member "port1" "port4"
    next
end
```

3. Create the IP pool. The IP pool must have a different subnet than the VWP peers.

```
config firewall ippool
    edit "vwp-pool-1"
        set startip 172.16.222.99
        set endip 172.16.222.100
    next
end
```

4. Configure the SNAT policy:

```
config firewall central-snat-map
    edit 2
        set srcintf "port4"
        set dstintf "port1"
        set orig-addr "all"
        set dst-addr "all"
        set nat-ippool "vwp-pool-1"
    next
end
```

5. Configure the firewall policy:

```
config firewall policy
    edit 90
        set srcintf "port4"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic all
    next
end
```

Destination NAT

Network Address Translation (NAT) is the process that enables a single device, such as a router or firewall, to act as an agent between the internet or public network and a local or private network. This agent acts in real-time to translate the source or destination IP address of a client or server on the network interface. NAT can be subdivided into two types:

- Source NAT (SNAT)
- Destination NAT (DNAT)

This section is about DNAT. For information about SNAT, see Source NAT on page 1913.

A virtual IP (VIP) maps external IP addresses to internal IP addresses for DNAT. See Configuring VIPs on page 1940 and Configuring VIP groups on page 1943.

The following types of VIPs can be created:

Static VIP	A virtual IP that maps an IP address or range to another IP address or range. Custom settings can allow the VIP to be filtered by Source Address and/or services, so that the VIP only applies to the filtered traffic. See Static virtual IPs on page 1944.
Static VIP with services	A virtual IP that defines services for a single port number mapping. See Virtual IP with services on page 1945.
Static VIP with port forwarding	A virtual IP that hides the port number for an internal server or maps several internal servers to the same public IP address. See Virtual IPs with port forwarding on page 1947. See also .
FQDN-based VIP	A virtual IP mapped to an FQDN. See Configure FQDN-based VIPs on page 1961.
Virtual server load balancing	A special type of virtual IP used to implement server load balancing. See Virtual server load balance on page 1949. Virtual IPs can also be used for server load balance multiplexing. See Virtual server load balance multiplexing on page 1959.
Central DNAT	Where DNAT is configured by creating virtual IPs and selecting the VIPs in firewall policies, central NAT is not configured in the firewall policy. Central NAT is enabled in *System Settings*. When enabled, the *Policy & Objects* tree displays the *Central SNAT* policy option. Use the *Central SNAT* policy to configure VIPs as separate objects. During use, FortiGate reads the enabled NAT rules from the top down, until it locates a matching rule. See Central DNAT on page 1962.

See also Configuring PCP port mapping with SNAT and DNAT on page 2014.

Configuring VIPs

Virtual IPs can be configured for IPv4 and IPv6. After creating the VIP, add it to a firewall policy.

FortiOS does not check whether VIPs overlap. As a result, you can configure multiple VIPs with the same external interface and IP. However, you can view overlapping VIPs in the security rating report. See Viewing VIP overlap in security rating reports on page 1942.

To configure a VIP in the GUI:

1. Go to *Policy & Objects* > *Virtual IPs*.
2. Select the *Virtual IP* tab, and click *Create New*.
3. Configure the following settings:

Name	Enter a name for the VIP.
Comments	Enter a description of the VIP.
Color	Click *Change* to select a color for the VIP.
Network	

	Interface (`extintf`)	The external interface that the firewall policy source interface must match. For example, if the external interface is port1, then the VIP can be used in a policy from port1 to port3, but not in a policy from port2 to port3. If the external interface is *any*, then the VIP can be used in any firewall policy.
	Type (`type`)	• Static NAT - Use an external IP address or address range. • FQDN - Use an external IP or FQDN address. • load-balance (CLI only) - Load balance traffic. • server-load-balance - Load balance traffic across multiple servers. SSL processing can be offloaded to the FortiGate. This type of VIP is configure from *Policy & Objects > Virtual Servers*. • dns-translation (CLI only) - DNS translation. • access-proxy - Used for ZTNA. See ZTNA web proxy example on page 1677 for details.
	External IP address/range (`extip`)	In a static NAT VIP, the external IP address is the IP address that the FortiGate listens for traffic on. When the external interface is not *any*, 0.0.0.0 can be used to make the external IP address equivalent to the external interface's IP address. The external IP address is also used to perform SNAT for the mapped server when the server outbound traffic with a destination interface that matches the external interface. The firewall policy must also have NAT enabled.
Map to		
	IPv4 address/range (`mappedip`)	The IPv4 address or range that the internal resource is being mapped to.
	IPv6 address/range (`ipv6-mappedip`)	The IPv6 address or range that the internal resource is being mapped to.
Optional Filters		Enable to access additional options.
	Source address (`src-filter`)	Restrict the source IP address, address range, or subnet that is allowed to access the VIP.
	Services (`service`)	Set the services that are allowed to be mapped.
Port Forwarding (`portforward`)		Enable port forwarding and display additional options. Enable port forwarding to specify the port (`mappedport`) to map to.
	Protocol (`protocol`)	Select the protocol to use when forwarding packets to the port.
	Port Mapping Type	• One to one - Each external service port is mapped to one port. A range is allowed, but the number of ports should be the same. • Many to Many - The port mapping can be one to one, one to many,

		or many to one. There are no restrictions on how many external ports must map to internal ports.
	External service port (`extport`)	Enter the external service port range to be mapped to a port range on the destination network.
	Map to IPv4 port (`mappedport`)	Enter the mapped IPv4 port range on the destination network.
	Map to IPv6 port (`ipv6-mappedport`)	Enter the mapped IPv6 port range on the destination network.

4. Click *OK* to save the VIP.

Viewing VIP overlap in security rating reports

There is no overlap check for VIPs, so there are no constraints when configuring multiple VIPs with the same external interface and IP. A new security rating report alerts users of any VIP overlaps.

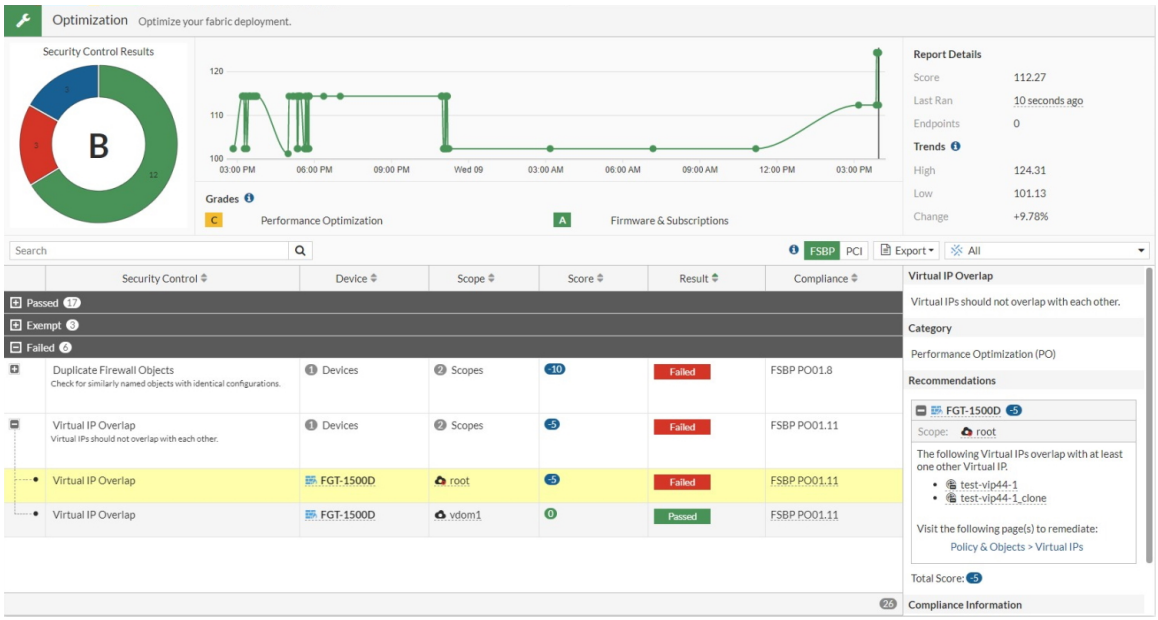
To configure two VIPs with the same external interface and IP:

```
config firewall vip
    edit "test-vip44-1"
        set extip 10.1.100.154
        set mappedip "172.16.200.156"
        set extintf "port24"
    next
    edit "test-vip44-1_clone"
        set extip 10.1.100.154
        set mappedip "172.16.200.156"
        set extintf "port24"
        set src-filter 10.1.100.11
    next
end
```

No error message appears regarding the overlapping VIPs.

To view the security rating report:

1. Go to *Security Fabric* > *Security Rating* and click the *Optimization* scorecard.
2. Expand the *Failed* section. The *Virtual IP Overlap* results show an overlap (*test-vip44-1* and *test-vip44-1_clone*) on the root FortiGate.



Configuring VIP groups

Virtual IP addresses (VIPs) can be organized into groups. After creating the VIP group, add it to a firewall policy.

VIP groups are useful when multiple VIPs are used together in firewall policies. If the VIP group members change, or a group member's settings change (such as the IP address, port, or port mapping type), then those changes are automatically updated in the corresponding firewall policies.

The following table summarizes which VIP types are allowed and not allowed to be members of a VIP group:

Group type	VIP types allowed as members	VIP types not allowed as members
IPv4	• Static NAT • Load balance • DNS translation • FQDN	• Access proxy • Server load balance
IPv6	• Static NAT	• Access proxy • Server load balance

Different VIP types can be added to the same group.

To configure a VIP group in the GUI:

1. Go to *Policy & Objects* > *Virtual IPs*.
2. Navigate to the *Virtual IP Group* or *IPv6 Virtual IP Group* tab.
3. Click *Create new*.
4. Enter a name.
5. Optionally, enter additional information in the *Comments* field.

6. For IPv4 groups, select the *Interface*. Select a specific interface if all of the VIPs are on the same interface; otherwise, select *any*.
7. Click the *+* in the *Members* field and select the members to add to the group.
8. Click *OK*.

To configure an IPv4 VIP group in the CLI:

```
config firewall vipgrp
    edit <name>
        set interface <name>
        set member <vip1> <vip2> ...
    next
end
```

To configure an IPv6 VIP group in the CLI:

```
config firewall vipgrp6
    edit <name>
        set member <vip1> <vip2> ...
    next
end
```

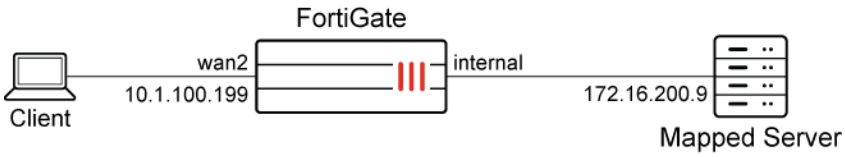
Static virtual IPs

Static Virtual IPs (VIP) are used to map external IP addresses to internal IP addresses. This is also called destination NAT, where a packet's destination is being NAT'd, or mapped, to a different address.

Static VIPs are commonly used to map public IP addresses to resources behind the FortiGate that use private IP addresses. A static one-to-one VIP is when the entire port range is mapped. A port forwarding VIP is when the mapping is configured on a specific port or port range.

When Central NAT is enabled, DNAT is no longer configured on the *Policy & Objects > Virtual IPs* page and is instead configured on the *Policy & Objects > DNAT & Virtual IPs* page. See Central DNAT on page 1962 for more information.

Sample configuration



To create a virtual IP in the GUI:

1. Go to *Policy & Objects > Virtual IPs* or, if Central NAT is enabled, *Policy & Objects > DNAT & Virtual IPs*.
2. Select the *Virtual IP* or *IPv6 Virtual IP* tab based on the IP versions used.
3. Click *Create new*.
4. Enter a unique name for the virtual IP.

5. Enter values for the external IP address/range and map to IPv4/IPv6 address/range fields.
6. Click *OK*.

To create a virtual IP in the CLI:

```
config firewall vip
    edit "Internal_WebServer"
        set extip 10.1.100.199
        set extintf "any"
        set mappedip "172.16.200.55"
    next
end
```

To apply a virtual IP to policy in the CLI:

```
config firewall policy
    edit 8
        set name "Example_Virtual_IP_in_Policy"
        set srcintf "wan2"
        set dstintf "internal"
        set srcaddr "all"
        set dstaddr "Internal_WebServer"
        set action accept
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

IP pools and VIPs as local IP addresses

IP pools and VIPs are considered local IP addresses if responding to ARP requests on these external IP addresses is enabled (`set arp-reply enable`, by default). In this case, the FortiGate is considered a destination for those IP addresses and can receive reply traffic at the application layer successfully.

However, as a side-effect, once an IP pool or VIP has been configured, even if it is never used in a firewall policy, the FortiGate considers it as a local address and will not forward traffic based on the routing table. Therefore, any unused IP pools or VIPs should be deleted to prevent any unexpected behavior.

> ⚠ For a history of behaviour changes related to IP pools and VIPs, see Technical Tip: IP pool and virtual IP behaviour changes in FortiOS 6.4, 7.0, 7.2, and 7.4.

Virtual IP with services

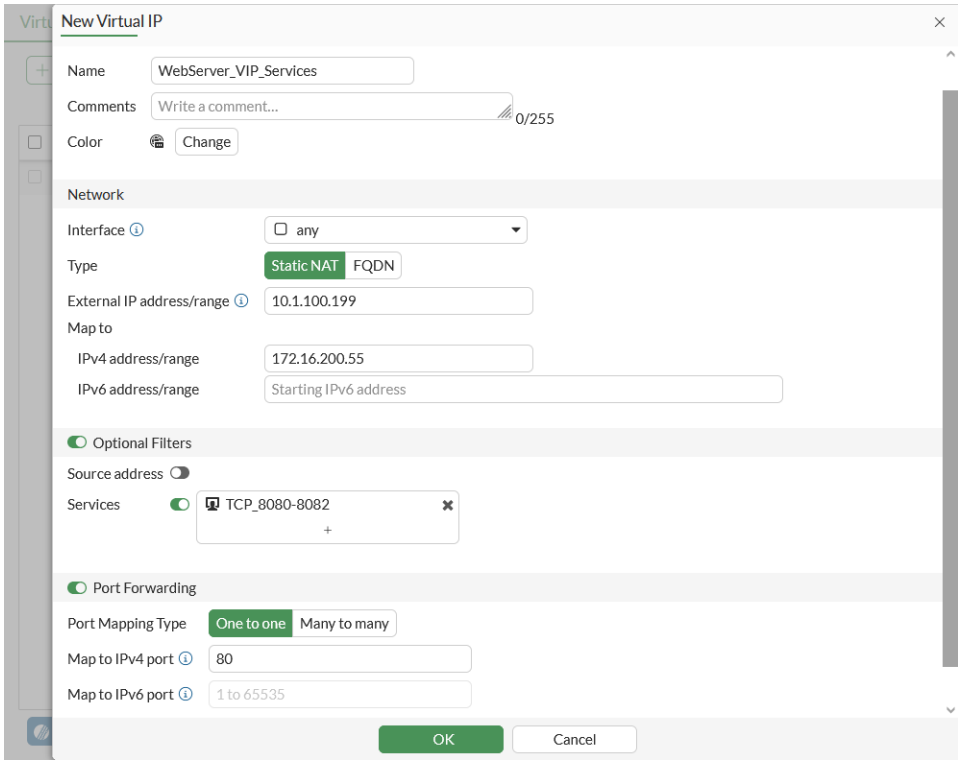
Virtual IP with services is a more flexible virtual IP mode. This mode allows users to define services to a single port number mapping.

This topic shows how to use virtual IP with services enabled. This example has one public external IP address. We map TCP ports 8080, 8081, and 8082 to an internal WebServer TCP port 80. This allows remote connections to communicate with a server behind the firewall.

Sample configuration

To create a virtual IP with services in the GUI:

1. Go to *Policy & Objects* > *Virtual IPs* and select the *Virtual IP* tab.
2. Click *Create new*.
3. Enter a unique name for the virtual IP.
4. Configure the fields in the *Network* section. For example:
   - Set *Interface* to *any*.
   - Set *External IP Address/Range* to *10.1.100.199*.
   - Set *Mapped IP Address/Range* to *172.16.200.55*.
5. Enable *Optional Filters* and then enable *Services*.
6. In the *Services* field, add TCP ports 8080, 8081, 8082. See Internet service customization on page 2200 for information about creating a custom port range service.
7. Enable *Port Forwarding* and set *Map to IPv4 port* to *80*.



8. Click *OK*.

To see the results:

1. Apply the above virtual IP to the firewall policy.
2. The results are:

- Access 10.1.100.199:8080 from external network and FortiGate maps to 172.16.200.55:80 in internal network.
- Access 10.1.100.199:8081 from external network and FortiGate maps to 172.16.200.55:80 in internal network.
- Access 10.1.100.199:8082 from external network and FortiGate maps to 172.16.200.55:80 in internal network.

To create a virtual IP with services in the CLI:

```
config firewall vip
    edit "WebServer_VIP_Services"
        set service "TCP_8080-8082"
        set extip 10.1.100.199
        set extintf "any"
        set portforward enable
        set mappedip "172.16.200.55"
        set mappedport 80
    next
end
```

Virtual IPs with port forwarding

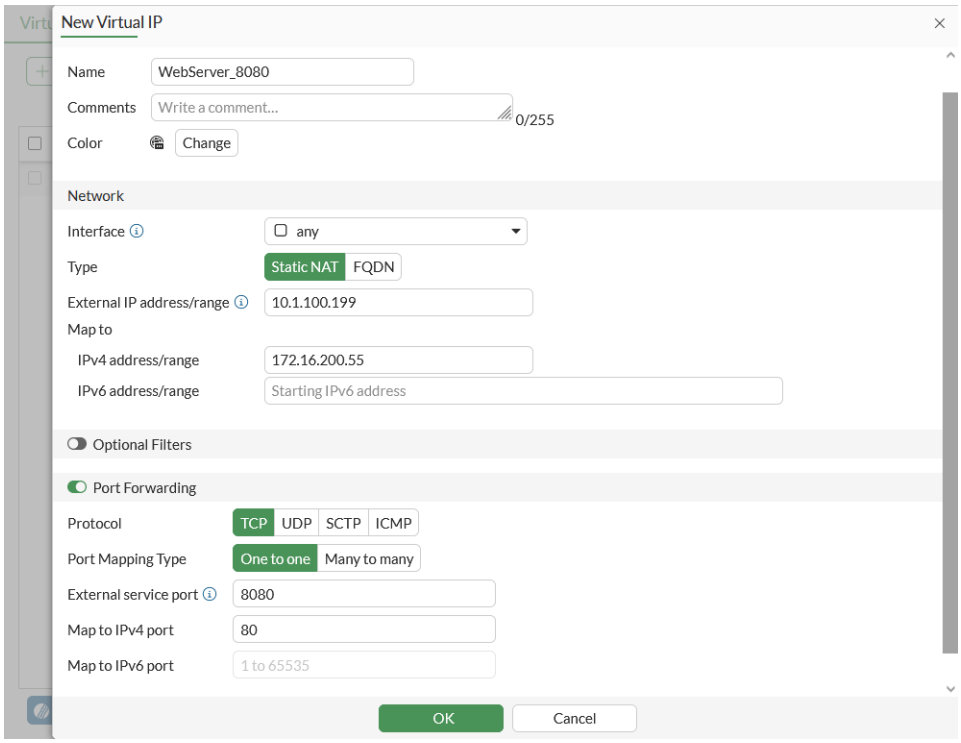
If you need to hide the internal server port number or need to map several internal servers to the same public IP address, enable port-forwarding for Virtual IP.

This topic shows how to use virtual IPs to configure port forwarding on a FortiGate unit. This example has one public external IP address. We map TCP ports 8080, 8081, and 8082 to different internal WebServers' TCP port 80. This allows remote connections to communicate with a server behind the firewall.

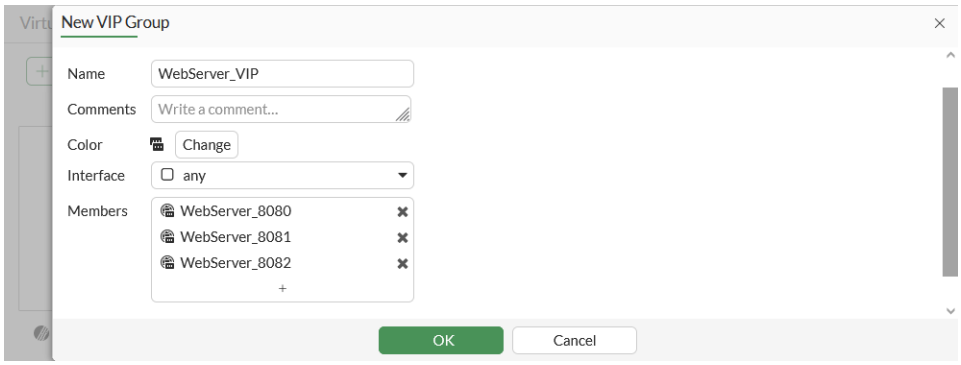
Sample configuration

To create a virtual IP with port forwarding in the GUI:

1. Go to *Policy & Objects* > *Virtual IPs* and select the *Virtual IP* tab.
2. Click *Create new*.
3. Enter a unique name for the virtual IP.
4. Configure the fields in the *Network* section. For example:
   - Set *Interface* to *any*.
   - Set *External IP Address/Range* to *10.1.100.199*.
   - Set *Mapped IP Address/Range* to *172.16.200.55*.
5. Leave *Optional Filters* disabled.
6. Enable *Port Forwarding* and configure the fields. For example:
   - Set *Protocol* to *TCP*.
   - Set *External Service Port* to *8080*.
   - Set *Map to IPv4 port* to *80*.



7. Click *OK*.
8. Follow the above steps to create two additional virtual IPs.
   a. For one virtual IP:
      - Use a different *Mapped IP Address/Range*, for example *172.16.200.56*.
      - Set *External Service Port* to *8081*.
      - Use the same *Map to IPv4 port* number: *80*.
   b. For the other virtual IP:
      - Use a different *Mapped IP Address/Range*, for example *172.16.200.57*.
      - Set *External Service Port* to *8082*.
      - Use the same *Map to IPv4 port* number: *80*.
9. Create a *Virtual IP Group* and put the above three virtual IPs into that group:
   a. Go to *Policy & Objects* > *Virtual IPs* and select the *Virtual IP Group* tab.
   b. Click *Create new*.
   c. Enter a name for the group.
   d. Add the three previously created virtual IPs as members.



e. Click *OK*.

To see the results:

1. Apply the above virtual IP to the Firewall policy.
2. The results are:
   - Access 10.1.100.199:8080 from external network and FortiGate maps to 172.16.200.55:80 in internal network.
   - Access 10.1.100.199:8081 from external network and FortiGate maps to 172.16.200.56:80 in internal network.
   - Access 10.1.100.199:8082 from external network and FortiGate maps to 172.16.200.57:80 in internal network

Virtual server load balance

This topic shows a special virtual IP type: virtual server. Use this type of VIP to implement server load balancing.

The FortiOS server load balancing contains all the features of a server load balancing solution. You can balance traffic across multiple backend servers based on multiple load balancing schedules including:

- Static (failover)
- Round robin
- Weighted (to account for different sized servers or based on the health and performance of the server including round trip time and number of connections)

The load balancer supports HTTP, HTTPS, IMAPS, POP3S, SMTPS, SSL/TLS, and generic TCP/UDP and IP protocols. Session persistence is supported based on the SSL session ID based on an injected HTTP cookie, or based on the HTTP or HTTPS host. SSL/TLS load balancing includes protection from protocol downgrade attacks. Server load balancing is supported on most FortiGate devices and includes up to 10,000 virtual servers on high-end systems.

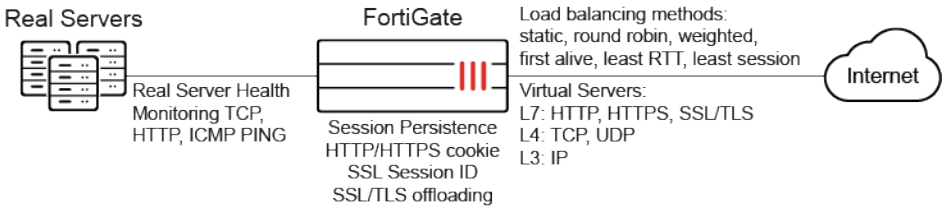
This feature is not supported on FortiGate models with 2 GB RAM or less. See Proxy-related features not supported on FortiGate 2 GB RAM models on page 126 for more information.

The following models do not support load balancing of Layer 7 protocols (HTTP, HTTPS, IMAPS, POP3S, SMTPS and SSL): FortiGate 40F, 60F and its variants.

FortiOS HTTP and HTTPS server load balancing does not support load balancing based on URL routing. You can use FortiWeb server pools or FortiADC server load balancing to load balance sessions to two or more URL based routes.

A policy with proxy-based inspection can accept all types of protocols configured in the Virtual Server object. However, a policy with flow-based inspection can only accept TCP/UDP/IP types in the Virtual Server object.

Sample topology



SSL/TLS offloading

FortiGate SSL/TLS offloading is designed for the proliferation of SSL/TLS applications. The key exchange and encryption/decryption tasks are offloaded to the FortiGate unit where they are accelerated using FortiASIC technology which provides significantly more performance than a standard server or load balancer. This frees up valuable resources on the server farm to give better response to business operations. Server load balancing offloads most SSL/TLS versions including SSL 3.0, TLS 1.0, TLS 1.2, and TLS 1.3, and supports full mode or half mode SSL offloading with DH key sizes up to 4096 bits.

FortiGate SSL offloading allows the application payload to be inspected before it reaches your servers. This prevents intrusion attempts, blocks viruses, stops unwanted applications, and prevents data loss. SSL/TLS content inspection supports TLS versions 1.0, 1.1, 1.2, and 1.3 and SSL versions 1.0, 1.1, 1.2, and 3.0.

The certificate used for SSL/TLS offloading by the virtual server configuration may contain multiple domain names, including wildcard domains. This certificate is returned to clients attempting to connect to the real server behind the FortiGate.

Virtual server requirements

When creating a new virtual server, you must configure the following options:

- Virtual Server Type.
- Load Balancing Methods.
- Health check monitoring (optional).
- Session persistence (optional).
- Virtual Server IP (External IP Address).
- Virtual Server Port (External Port).
- Real Servers (Mapped IP Address & Port).

Virtual server types

Select the protocol to be load balanced by the virtual server. If you select a general protocol such as IP, TCP, or UDP, the virtual server load balances all IP, TCP, or UDP sessions. If you select specific protocols such as HTTP, HTTPS, or SSL, you can apply additional server load balancing features such as *Persistence* and *HTTP Multiplexing*.

Protocol	Description
HTTP	Select *HTTP* to load balance only HTTP sessions with the destination port number that matches the *Virtual Server Port* setting. Change *Virtual Server Port* to match the destination port of the sessions to be load balanced (usually port 80 for HTTP sessions). You can enable *HTTP Multiplexing*. You can also set *Persistence* to *HTTP Cookie* to enable cookie-based persistence.
HTTPS	Select *HTTPS* to load balance only HTTPS sessions with the destination port number that matches the *Virtual Server Port* setting. Change *Virtual Server Port* to match the destination port of the sessions to be load balanced (usually port 443 for HTTPS sessions). You can enable *HTTP Multiplexing*. You can also set *Persistence* to *HTTP Cookie* to enable cookie-based persistence, or you can set *Persistence* to *SSL Session ID*.
IMAPS	Select *IMAPS* to load balance only IMAPS sessions with the destination port number that matches the *Virtual Server Port* setting. Change *Virtual Server Port* to match the destination port of the sessions to be load balanced (usually port 993 for IMAPS sessions). You can also set *Persistence* to *SSL Session ID*.
POP3S	Select *POP3S* to load balance only POP3S sessions with the destination port number that matches the *Virtual Server Port* setting. Change *Virtual Server Port* to match the destination port of the sessions to be load balanced (usually port 995 for POP3S sessions). You can also set *Persistence* to *SSL Session ID*.
SMTPS	Select *SMTPS* to load balance only SMTPS sessions with the destination port number that matches the *Virtual Server Port* setting. Change *Virtual Server Port* to match the destination port of the sessions to be load balanced (usually port 465 for SMTPS sessions). You can also set *Persistence* to *SSL Session ID*.
SSL	Select *SSL* to load balance only SSL sessions with the destination port number that matches the *Virtual Server Port* setting. Change *Virtual Server Port* to match the destination port of the sessions to be load balanced. You can also set *Persistence* to *SSL Session ID*.
TCP	Select *TCP* to load balance only TCP sessions with the destination port number that matches the *Virtual Server Port* setting. Change *Virtual Server Port* to match the destination port of the sessions to be load balanced.
UDP	Select *UDP* to load balance only UDP sessions with the destination port number that matches the *Virtual Server Port* setting. Change *Virtual Server Port* to match the destination port of the sessions to be load balanced.
IP	Select *IP* to load balance all sessions accepted by the security policy that contains this virtual server.

Load balancing methods

The load balancing method defines how sessions are load balanced to real servers.

All load balancing methods do not send traffic to real servers that are down or not responding. FortiGate can only determine if a real server is not responding by using a health check monitor. You should always add at least one health check monitor to a virtual server or to real servers; otherwise load balancing might try to distribute sessions to real servers that are not functioning.

Method	Description
Static	The traffic load is statically spread evenly across all real servers. Sessions are not assigned according to how busy individual real servers are. This load balancing method provides some persistence because all sessions from the same source address always go to the same real server. Because the distribution is stateless, so if a real server is added, removed, or goes up or down, the distribution is changed and persistence might be lost.
Round Robin	Directs new requests to the next real server. This method treats all real servers as equals regardless of response time or the number of connections. This method does not direct requests to real servers that down or non responsive.
Weighted	Real servers with a higher weight value receive a larger percentage of connections. Set the real server weight when adding a real server.
Least Session	Directs requests to the real server that has the least number of current connections. This method works best in environments where the real servers or other equipment you are load balancing all have similar capabilities. This load balancing method uses the FortiGate session table to track the number of sessions being processed by each real server. The FortiGate unit cannot detect the number of sessions actually being processed by a real server.
Least RTT	Directs sessions to the real server with the lowest round trip time. The round trip time is determined by a ping health check monitor. The default is 0 if no ping health check monitors are added to the virtual server.
First Alive	Directs sessions to the first live real server. This load balancing schedule provides real server failover protection by sending all sessions to the first live real server. If a real server fails, all sessions are sent to the next live real server. Sessions are not distributed to all real servers so all sessions are processed by the first real server only.
HTTP Host	Load balances HTTP host connections across multiple real servers using the host's HTTP header to guide the connection to the correct real server.

Health check monitoring

In the FortiGate GUI, you can configure health check monitoring so that the FortiGate unit can verify that real servers are able respond to network connection attempts. If a real server responds to connection attempts, the load balancer continues to send sessions to it. If a real server stops responding to connection attempts, the load balancer assumes that the server is down and does not send sessions to it. The health check monitor configuration determines how the load balancer tests real servers. You can use a single health check monitor for multiple load balancing configurations. You can configure TCP, HTTP, DNS, and ping health check monitors. You usually set the health check monitor to use the same protocol as the traffic being load balanced to it. For example, for an HTTP load balancing configuration, you would normally use an HTTP health check monitor.

Session persistence

Use persistence to ensure a user is connected to the same real server every time the user makes an HTTP, HTTPS, or SSL request that is part of the same user session. For example, if you are load balancing HTTP and

HTTPS sessions to a collection of eCommerce web servers, when users make a purchase, they will be starting multiple sessions as they navigate the eCommerce site. In most cases, all the sessions started by this user during one eCommerce session should be processed by the same real server. Typically, the HTTP protocol keeps track of these related sessions using cookies. HTTP cookie persistence ensure all sessions that are part of the same user session are processed by the same real server.

When you configure persistence, the FortiGate unit load balances a new session to a real server according to the load balance method. If the session has an HTTP cookie or an SSL session ID, the FortiGate unit sends all subsequent sessions with the same HTTP cookie or SSL session ID to the same real server.

Real servers

Starting in FortiOS 7.6.4, certificate verification of the real server is enabled by default. See Certificate verification for real servers on page 1629 for more details. You can manually adjust the setting using the following CLI command:

```
config realservers
    edit <id>
        set verify-cert {enable | disable}
    next
end
```

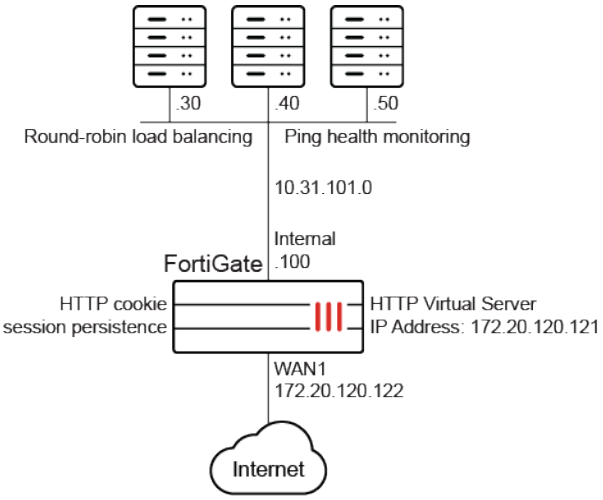
Add real servers to a load balancing virtual server to provide information the virtual server requires to send sessions to the server. A real server configuration includes the IP address of the real server and port number the real server receives sessions on. The FortiGate unit sends sessions to the real server's IP address using the destination port number in the real server configuration.

When configuring a real server, you can also specify the weight (if the load balance method is set to *Weighted*) and you can limit the maximum number of open connections between the FortiGate unit and the real server. If the maximum number of connections is reached for the real server, the FortiGate unit automatically switches all further connection requests to other real servers until the connection number drops below the limit. Setting *Maximum Connections* to 0 means that the FortiGate unit does not limit the number of connections to the real server.

Sample of HTTP load balancing to three real web servers

This example describes the steps to configure the load balancing configuration below. In this configuration, a FortiGate unit is load balancing HTTP traffic from the Internet to three HTTP servers on the internal network. HTTP sessions are accepted at the wan1 interface with destination IP address 172.20.120.121 on TCP port 8080, and forwarded from the internal interface to the web servers. When forwarded, the destination address of the session is translated to the IP address of one of the web servers.

This load balancing configuration also includes session persistence using HTTP cookies, round-robin load balancing, and TCP health monitoring for the real servers. Ping health monitoring consists of the FortiGate unit using ICMP ping to ensure the web servers can respond to network traffic.



General steps:

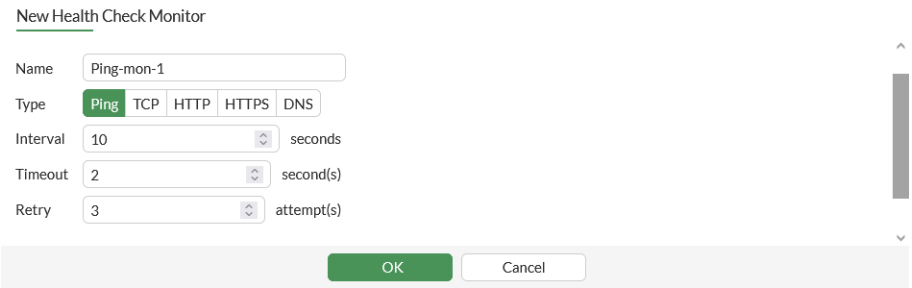
1. Create a health check monitor.
   A ping health check monitor causes the FortiGate to ping the real servers every 10 seconds. If one of the servers does not respond within 2 seconds, the FortiGate unit will retry the ping 3 times before assuming that the HTTP server is not responding.
2. Create a load balance virtual server with three real servers.
3. Add the load balancing virtual server to a policy as the destination address.

To see the virtual servers and health check monitors options in the GUI, *Load Balance* must be selected in *Feature Visibility > Additional Features*. See Feature visibility on page 3952 on page 1 for details.

Configure a load balancing virtual server in the GUI

To create a health check monitor:

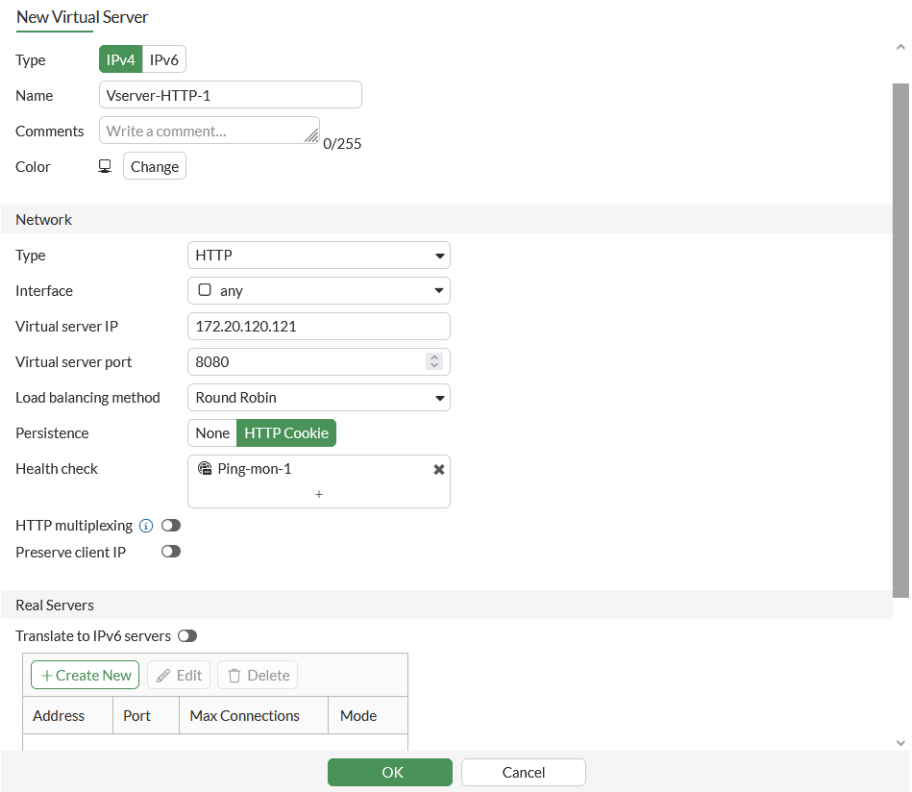
1. Go to *Policy & Objects > Health Check*.
2. Click *Create New*.
3. Set the following:
   - *Name* to *Ping-mon-1*
   - *Type* to *Ping*
   - *Interval* to *10* seconds
   - *Timeout* to *2* seconds
   - *Retry* to *3* attempt(s)



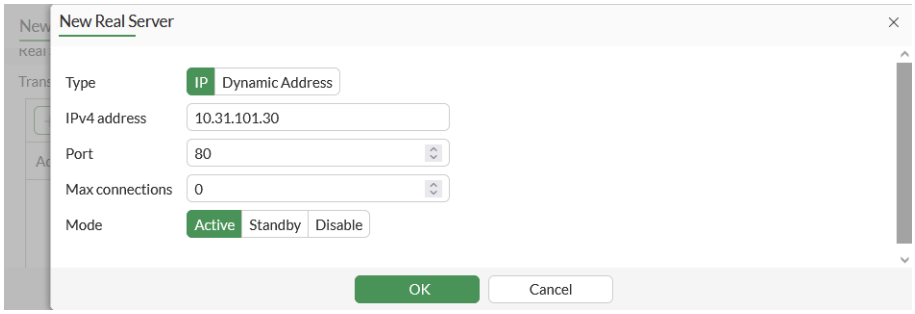
4. Click *OK*.

To create a virtual server:

1. Go to *Policy & Objects* > *Virtual Servers*.
2. Click *Create New*.
3. Set the following:
   - *Name* to *Vserver-HTTP-1*
   - *Type* to *HTTP*
   - *Interface* to *wan1*
   - *Virtual Server IP* to *172.20.120.121*
   - *Virtual Server Port* to *8080*
   - *Load Balance Method* to *Round Robin*
   - *Persistence* to *HTTP Cookie*
   - *Health Check* to *Ping-mon-1*



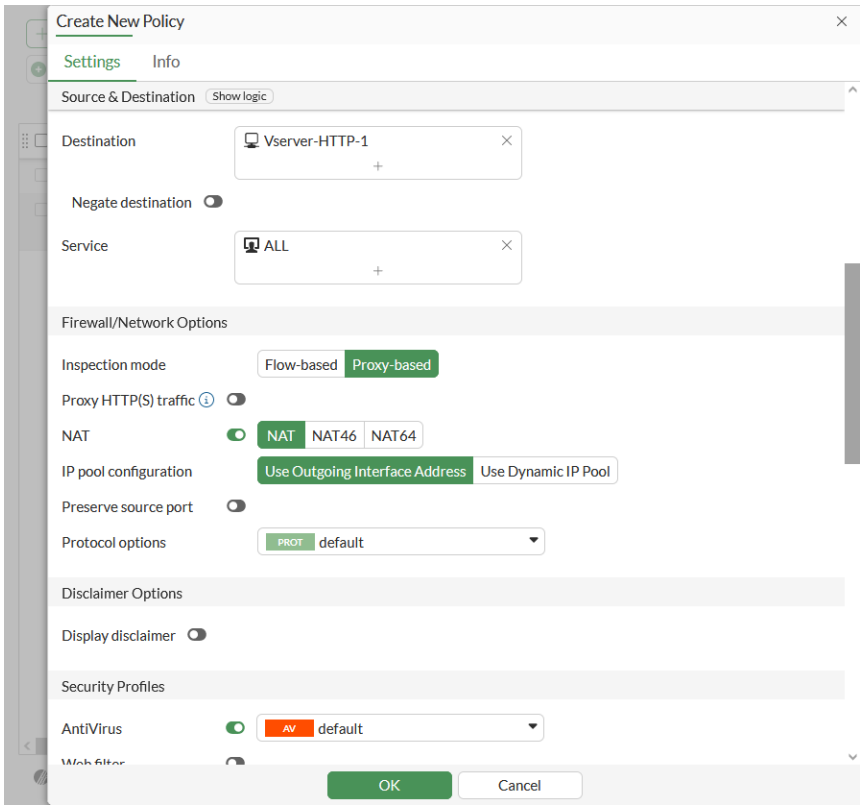
4. In the *Real Servers* table, click *Create New*.
5. Set the following for the first real server:
   - *Type* to *IP*
   - *IP Address* to *10.31.101.30*
   - *Port* to *80*
   - *Max Connections* to *0*
   - *Mode* to *Active*



6. Click *OK*. Configure two more real servers with IP addresses 10.31.101.40 and 10.31.101.50, and the same settings as the first real server.
7. Click *OK*.

To create a security policy that includes the load balance virtual server as the destination address:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Click *Create New*.
3. Set the *Inspection Mode* to *Proxy-based*. A flow-based firewall policy can only accept TCP/UDP/IP type virtual servers.
4. Set the following:
   - *Name* to *LB-policy*
   - *Incoming Interface* to *wan1*
   - *Outgoing Interface* to *internal*
   - *Source* to *all*
   - *Destination* to *Vserver-HTTP-1*
   - *Schedule* to *always*
   - *Service* to *ALL*
   - *Action* to *ACCEPT*
5. Enable *NAT* and set *IP Pool Configuration* to *Use Outgoing Interface Address*.
6. Enable *AntiVirus* and select an antivirus profile.



7. Click *OK*.

Configure a load balancing virtual server in the CLI

To configure HTTP load balancing to three real web servers in the CLI:

1. Create a health check monitor:

```
config firewall ldb-monitor
    edit "Ping-mon-1"
        set type ping
        set interval 10
        set timeout 2
        set retry 3
    next
end
```

2. Create a virtual server:

```
config firewall vip
    edit "Vserver-HTTP-1"
        set type server-load-balance
        set extip 172.20.120.121
        set extintf "any"
        set server-type http
        set monitor "Ping-mon-1"
        set ldb-method round-robin
```

```
        set persistence http-cookie
        set extport 8080
        config realservers
            edit 1
                set type ip
                set ip 10.31.101.30
                set port 80
            next
            edit 2
                set type ip
                set ip 10.31.101.40
                set port 80
            next
            edit 3
                set type ip
                set ip 10.31.101.50
                set port 80
            next
        end
    next
end
```

3. Add the load balancing virtual server to a policy as the destination address:

```
config firewall policy
    edit 2
        set name "LB-policy"
        set inspection-mode proxy
        set srcintf "wan1"
        set dstintf "internal"
        set srcaddr "all"
        set dstaddr "Vserver-HTTP-1"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "certificate-inspection"
        set av-profile "default"
        set fsso disable
        set nat enable
    next
end
```

Results

Traffic accessing 172.20.120.121:8080 is forwarded in turn to the three real servers.

If the access request has an http-cookie, FortiGate forwards the access to the corresponding real server according to the cookie.

Virtual server load balance multiplexing

HTTP2 connection coalescing and concurrent multiplexing allows multiple HTTP2 requests to share the same TLS connection when the destination IP is the same.

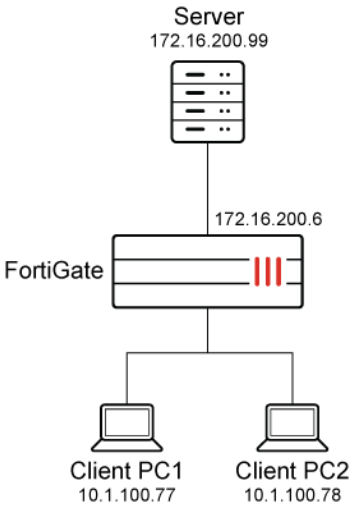
To configure the load balanced virtual server:

```
config firewall vip
    edit <name>
        set type server-load-balance
        set server-type {http | https}
        set http-multiplex {enable | disable}
        set http-multiplex-ttl <integer>
        set http-multiplex-max-request <integer>
        set h2-support {enable | disable}
    next
end
```

`http-multiplex {enable | disable}`	Enable/disable HTTP multiplexing.
`http-multiplex-ttl <integer>`	Set the time-to-live for idle connections to servers (in seconds, 0 - 2147483647, default = 15).
`http-multiplex-max-request <integer>`	Set the maximum number of requests that the multiplex server can handle before disconnecting (0 - 2147483647, default = 0).
`h2-support {enable | disable}`	Enable/disable HTTP2 support (default = enable).

Example

In this example, multiple clients submit requests in HTTP2. The requests hit the VIP address, and then FortiGate opens a session between itself (172.16.200.6) and the server (172.16.200.99). The coalescing occurs in this session as the multiple streams share the same TLS session to connect to the same destination server.



To configure connection coalescing and concurrent multiplexing with virtual server load balancing:

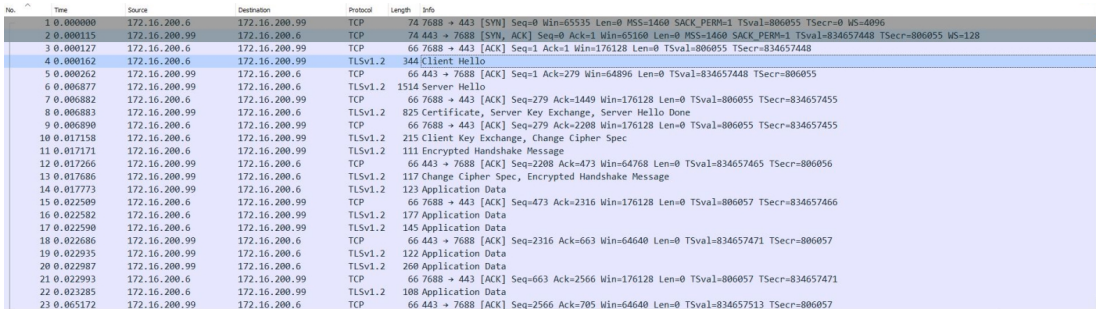
1. Configure the virtual server:

```
config firewall vip
    edit "vip-test"
        set type server-load-balance
        set extip 10.1.100.222
        set extintf "port2"
        set server-type https
        set extport 443
        config realservers
            edit 1
                set ip 172.16.200.99
                set port 443
            next
        end
        set http-multiplex enable
        set ssl-mode full
        set ssl-certificate "Fortinet_SSL"
    next
end
```

2. Configure the firewall policy:

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "port3"
        set action accept
        set srcaddr "all"
        set dstaddr "vip-test"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "deep-inspection-clone"
        set av-profile "av"
        set logtraffic all
        set nat enable
    next
end
```

3. Get the clients to access the VIP address (10.1.100.222). The FortiGate shares the first TLS connection with second TLS connection.
4. Verify the sniffer packet capture on the FortiGate server side. There is one client hello.

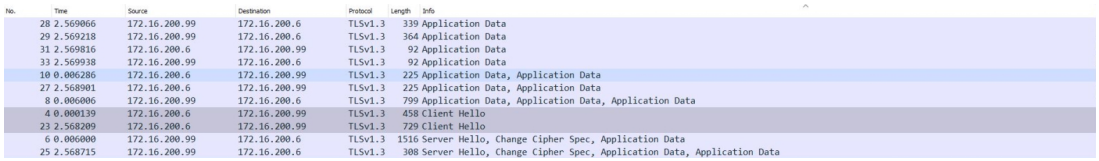


No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	172.16.200.6	172.16.200.99	TCP	74	7688 → 443 [SYN] Seq=0 Win=65535 Len=0 MSS=1460 SACK_PERM=1 TSval=806055 TSecr=0 WS=4096
2	0.000115	172.16.200.99	172.16.200.6	TCP	74	443 → 7688 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=834657448 TSecr=806055 WS=128
3	0.000127	172.16.200.6	172.16.200.99	TCP	66	7688 → 443 [ACK] Seq=1 Ack=1 Win=176128 Len=0 TSval=806055 TSecr=834657448
4	0.000162	172.16.200.6	172.16.200.99	TLSv1.2	344	Client Hello
5	0.000262	172.16.200.99	172.16.200.6	TCP	66	443 → 7688 [ACK] Seq=1 Ack=279 Win=64896 Len=0 TSval=834657448 TSecr=806055
6	0.006877	172.16.200.99	172.16.200.6	TLSv1.2	1514	Server Hello
7	0.006882	172.16.200.6	172.16.200.99	TCP	66	7688 → 443 [ACK] Seq=279 Ack=1449 Win=176128 Len=0 TSval=806055 TSecr=834657455
8	0.006883	172.16.200.99	172.16.200.6	TLSv1.2	825	Certificate, Server Key Exchange, Server Hello Done
9	0.006890	172.16.200.6	172.16.200.99	TCP	66	7688 → 443 [ACK] Seq=279 Ack=2208 Win=176128 Len=0 TSval=806055 TSecr=834657455
10	0.017158	172.16.200.6	172.16.200.99	TLSv1.2	215	Client Key Exchange, Change Cipher Spec
11	0.017171	172.16.200.6	172.16.200.99	TLSv1.2	111	Encrypted Handshake Message
12	0.017266	172.16.200.99	172.16.200.6	TCP	66	443 → 7688 [ACK] Seq=2208 Ack=473 Win=64768 Len=0 TSval=834657465 TSecr=806056
13	0.017686	172.16.200.99	172.16.200.6	TLSv1.2	117	Change Cipher Spec, Encrypted Handshake Message
14	0.017773	172.16.200.99	172.16.200.6	TLSv1.2	123	Application Data
15	0.022509	172.16.200.6	172.16.200.99	TCP	66	7688 → 443 [ACK] Seq=473 Ack=2316 Win=176128 Len=0 TSval=806057 TSecr=834657466
16	0.022582	172.16.200.6	172.16.200.99	TLSv1.2	177	Application Data
17	0.022590	172.16.200.6	172.16.200.99	TLSv1.2	145	Application Data
18	0.022686	172.16.200.99	172.16.200.6	TCP	66	443 → 7688 [ACK] Seq=2316 Ack=663 Win=64640 Len=0 TSval=834657471 TSecr=806057
19	0.022935	172.16.200.99	172.16.200.6	TLSv1.2	122	Application Data
20	0.022987	172.16.200.99	172.16.200.6	TLSv1.2	260	Application Data
21	0.022993	172.16.200.6	172.16.200.99	TCP	66	7688 → 443 [ACK] Seq=663 Ack=2566 Win=176128 Len=0 TSval=806057 TSecr=834657471
22	0.023285	172.16.200.6	172.16.200.99	TLSv1.2	108	Application Data
23	0.065172	172.16.200.99	172.16.200.6	TCP	66	443 → 7688 [ACK] Seq=2566 Ack=705 Win=64640 Len=0 TSval=834657513 TSecr=806057

5. Disable HTTP multiplexing:

```
config firewall vip
    edit "vip-test"
        config realservers
            edit 1
                set type ip
                set ip 172.16.200.99
                set port 443
            next
        end
        set http-multiplex disable
    next
end
```

6. Verify the sniffer packet capture. This time, the FortiGate does reuse the TLS connection, so there are two client hellos sent to the real server.



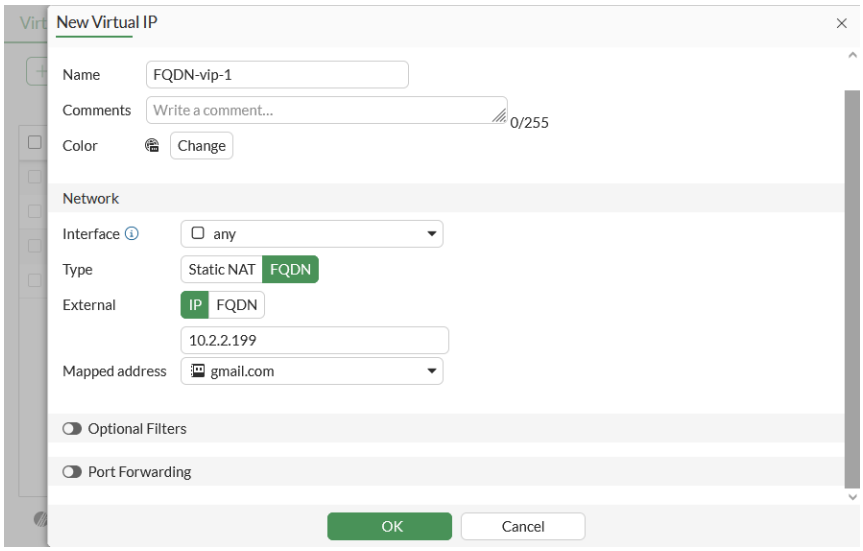
No.	Time	Source	Destination	Protocol	Length	Info
28	2.569066	172.16.200.99	172.16.200.6	TLSv1.3	339	Application Data
29	2.569218	172.16.200.99	172.16.200.6	TLSv1.3	364	Application Data
31	2.569816	172.16.200.6	172.16.200.99	TLSv1.3	92	Application Data
33	2.569938	172.16.200.99	172.16.200.6	TLSv1.3	92	Application Data
10	0.006286	172.16.200.6	172.16.200.99	TLSv1.3	225	Application Data, Application Data
27	2.568901	172.16.200.6	172.16.200.99	TLSv1.3	225	Application Data, Application Data
8	0.006006	172.16.200.99	172.16.200.6	TLSv1.3	799	Application Data, Application Data, Application Data
4	0.000139	172.16.200.6	172.16.200.99	TLSv1.3	458	Client Hello
23	2.568209	172.16.200.6	172.16.200.99	TLSv1.3	729	Client Hello
6	0.006000	172.16.200.99	172.16.200.6	TLSv1.3	1516	Server Hello, Change Cipher Spec, Application Data
25	2.568715	172.16.200.99	172.16.200.6	TLSv1.3	308	Server Hello, Change Cipher Spec, Application Data, Application Data

Configure FQDN-based VIPs

In public cloud environments, sometimes it is necessary to map a VIP to an FQDN address.

To configure an FQDN-based VIP in the GUI:

1. Go to *Policy & Objects* > *Virtual IPs* and select the *Virtual IP* tab.
2. Click *Create new*.
3. Enter a name for the VIP.
4. Select an interface.
5. For *Type*, select *FQDN*.
6. For *External*, select *IP* and enter the external IP address.
7. For *Mapped address*, select an FQDN address.



8. Click *OK*.

To configure an FQDN-based VIP in the CLI:

```
config firewall vip
    edit "FQDN-vip-1"
        set type fqdn
        set extip 10.2.2.199
        set extintf "any"
        set mapped-addr "destination"
    next
end
```

Central DNAT

Central NAT allows for the central configuration of SNAT (source NAT) and DNAT (destination NAT).

To enable central NAT in the GUI:

1. Go to *System* > *Settings* > *General*.
2. In the *System Operation Settings*, enable *Central SNAT*.
3. Click *Apply*.

To enable central NAT in the CLI:

```
config system settings
    set central-nat {enable | disable}
end
```

When central NAT is enabled, virtual IPs (VIPs) are not configured in the firewall policy. The VIPs are configured as separate objects where their status must be enabled.

This option is only available for IPv4 VIP and VIP46 objects.

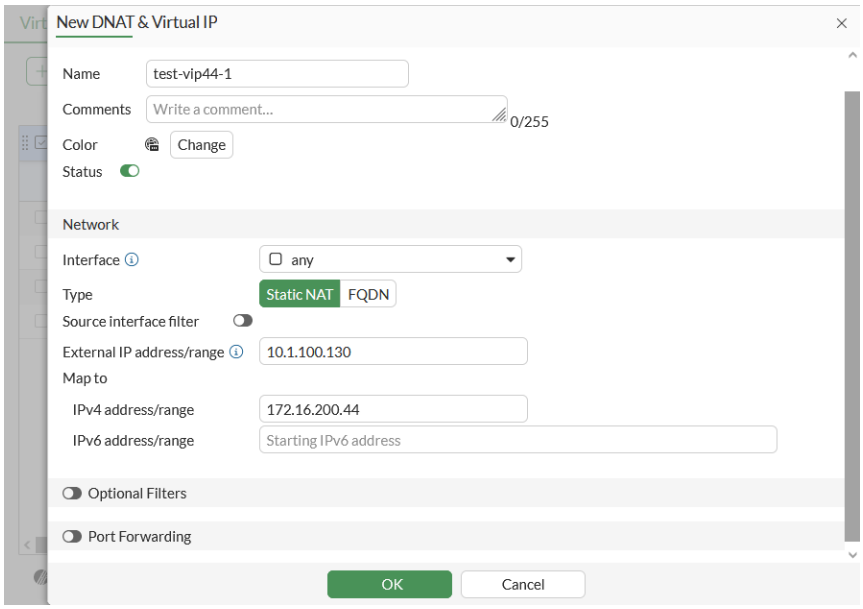
Configuring a DNAT and VIP object in central NAT mode is similar to configuring a VIP when central NAT is disabled. See Static virtual IPs on page 1944 for more information on each setting.

VIP objects can carry over when switching from non-central NAT mode to central NAT mode or vice-versa. However, if a VIP is assigned to a firewall policy in non-central NAT mode, it must be unassigned before switching to central NAT mode.

In this example, a DNAT and VIP are configured to forward traffic from 10.1.100.130 to 172.16.200.44. This example assumes that the firewall address, Addr_172.16.200.44/32, has already been configured.

To configure DNAT and a VIP in the GUI:

1. Configure the VIP:
   a. Go to *Policy & Objects* > *DNAT & Virtual IPs* and on the *Virtual IP* tab, click *Create New*.
   b. Enter a name (*test-vip44-1*).
   c. Set the *External IP address/range* to *10.1.100.130*.
   d. Set the *Map to IPv4 address/range* to *172.16.200.44*.

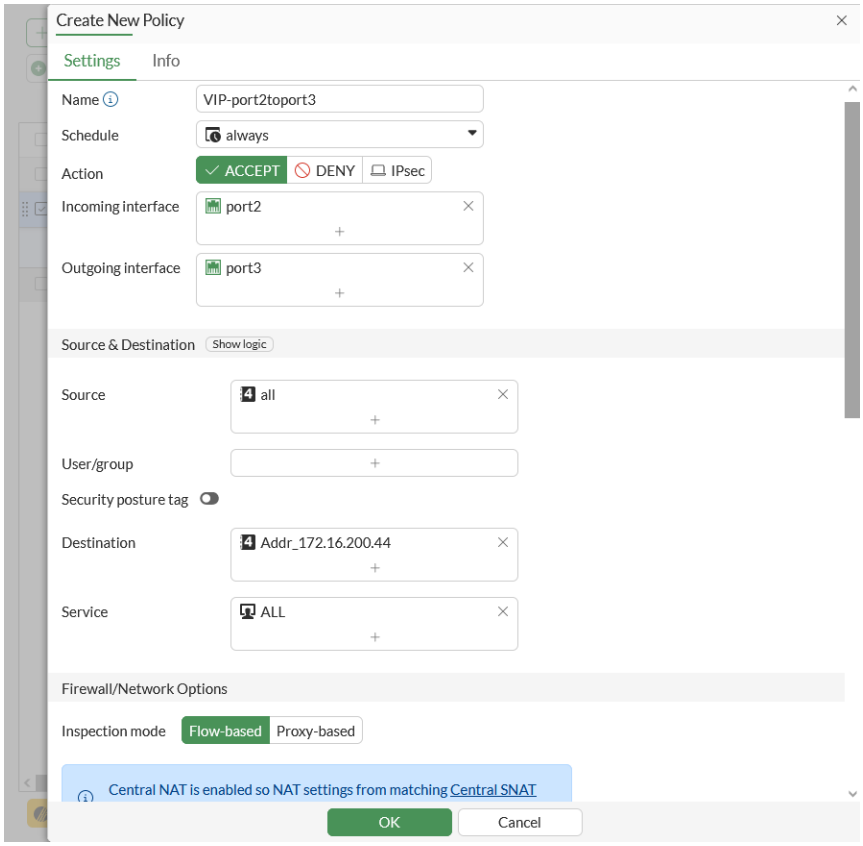


   e. Click *OK*.
2. Configure a firewall policy that allows traffic in the direction of the VIP:
   a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
   b. Configure the following settings:

Name	*VIP-port2toport3*
Source	*all*
Destination	*Addr_172.16.200.44*
Schedule	*always*

Service	*ALL*
Action	*ACCEPT*

c. Configure the other settings as needed. There is no SNAT configuration section, so central SNAT policies will be applied.



d. Click *OK*.

To configure DNAT and a VIP in the CLI:

1. Configure the VIP:

```
config firewall vip
    edit "test-vip44-1"
        set extip 10.1.100.130
        set mappedip "172.16.200.44"
        set extintf "any"
        set status enable
    next
end
```

2. Configure a firewall policy that allows traffic in the direction of the VIP:

```
config firewall policy
    edit 3
        set name "VIP-port2toport3"
```

```
        set srcintf "port2"
        set dstintf "port3"
        set action accept
        set srcaddr "all"
        set dstaddr "Addr_172.16.200.44"
        set schedule "always"
        set service "ALL"
    next
end
```

To verify the DNAT and VIP:

If the VIP status is enabled, it will appear in the VIP table:

```
# diagnose firewall iprope list 100000
policy index=7 uuid_idx=625 action=accept
flag (8000104): f_p nat pol_stats
cos_fwd=0  cos_rev=0
group=00100000 av=00000000 au=00000000 split=00000000
host=0 chk_client_info=0x0 app_list=0 ips_view=0
misc=0
zone(1): 0 -> zone(1): 0
source(1): 0.0.0.0-255.255.255.255, uuid_idx=0,
dest(1): 10.1.100.130-10.1.100.130, uuid_idx=625,
service(1):
        [0:0x0:0/(0,0)->(0,0)] helper:auto
nat(1): flag=0 base=10.1.100.130:0 172.16.200.44-172.16.200.44(0:0)
```

If the VIP status is disabled, it will not appear in the VIP table.

In this example, a one-to-one static NAT is enabled. Send a ping to 10.1.100.130, and the traffic will be forwarded to the destination 172.16.200.44.

Examples and policy actions

The following topics provide examples and instructions on policy actions:

- NAT46 and NAT64 policy and routing configurations on page 1966
- Hairpin NAT on page 1977
- Mirroring SSL traffic in policies on page 1981
- Recognize anycast addresses in geo-IP blocking on page 1983
- Matching GeoIP by registered and physical location on page 1984
- HTTP to HTTPS redirect for load balancing on page 1986
- Use Active Directory objects directly in policies on page 1988
- No session timeout on page 1992
- MAP-E support on page 1993
- DHCP-PD support for MAP-E on page 1998
- Seven-day rolling counter for policy hit counters on page 2004

- Cisco Security Group Tag as policy matching criteria on page 2006
- Virtual patching on the local-in management interface on page 2009
- Configuring PCP port mapping with SNAT and DNAT on page 2014
- Refreshing active sessions for specific protocols and port ranges per VDOM in a specified direction on page 2023
- Per-policy disclaimer messages on page 2025
- Full cone NAT for fixed port range IP pools on page 2028

NAT46 and NAT64 policy and routing configurations

Multiple NAT46 and NAT64 related objects are consolidated into regular objects. A per-VDOM virtual interface, naf.<vdom>, is automatically added to process NAT46/NAT64 traffic. The features include:

- `vip46` and `vip64` settings are consolidated in `vip` and `vip6` configurations.
- `policy46` and `policy64` settings are consolidated in `firewall policy` settings.
- `nat46`/`nat64` are included in `firewall policy` settings.
- `ippool` and `ippool6` support NAT46 and NAT64 (when enabled, the IP pool should match a subnet).
- Central SNAT supports NAT46 and NAT64.
- `add-nat46-route` in `ippool6` and `add-nat64-route` in `ippool` are enabled by default. The FortiGate generates a static route that matches the IP range in `ippool6` or `ippool` for the naf tunnel interface.

Automatic processing of the naf tunnel interface is not supported in security policies.

To configure NAT46/NAT64 translation, use the standard `vip`/`vip6` setting, apply it in a firewall policy, enable NAT46/NAT64, and enter the IP pool to complete the configuration.

The external IP address cannot be the same as the external interface IP address.

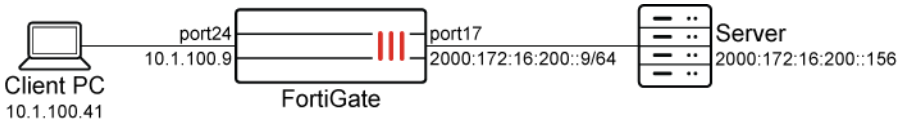
Examples

IPv6 must be enabled to configure these examples. In the GUI, so go to *System > Feature Visibility* and enable *IPv6*. In the CLI, enter the following:

```
config system global
    set gui-ipv6 enable
end
```

NAT46 policy

In this example, a client PC is using IPv4 and an IPv4 VIP to access a server that is using IPv6. The FortiGate uses NAT46 to translate the request from IPv4 to IPv6 using the virtual interface naf.root. An `ippool6` is applied so that the request is SNATed to the `ippool6` address (2000:172:16:101::1 - 2000:172:16:101::1).

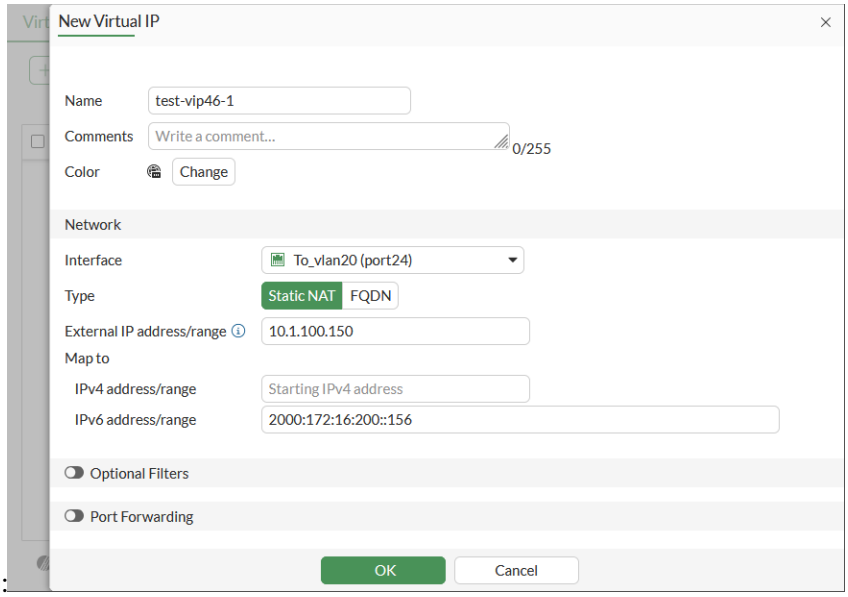


To create a NAT46 policy in the GUI:

1. Configure the VIP:
   a. Go to *Policy & Objects* > *Virtual IPs* and select the *Virtual IP* tab.
   b. Click *Create new*.

Name	test-vip46-1
Interface	To_vlan20
Type	Static NAT
External IP address/range	10.1.100.150
Map to IPv6 address/range	2000:172:16:200::156

   Enter the following:



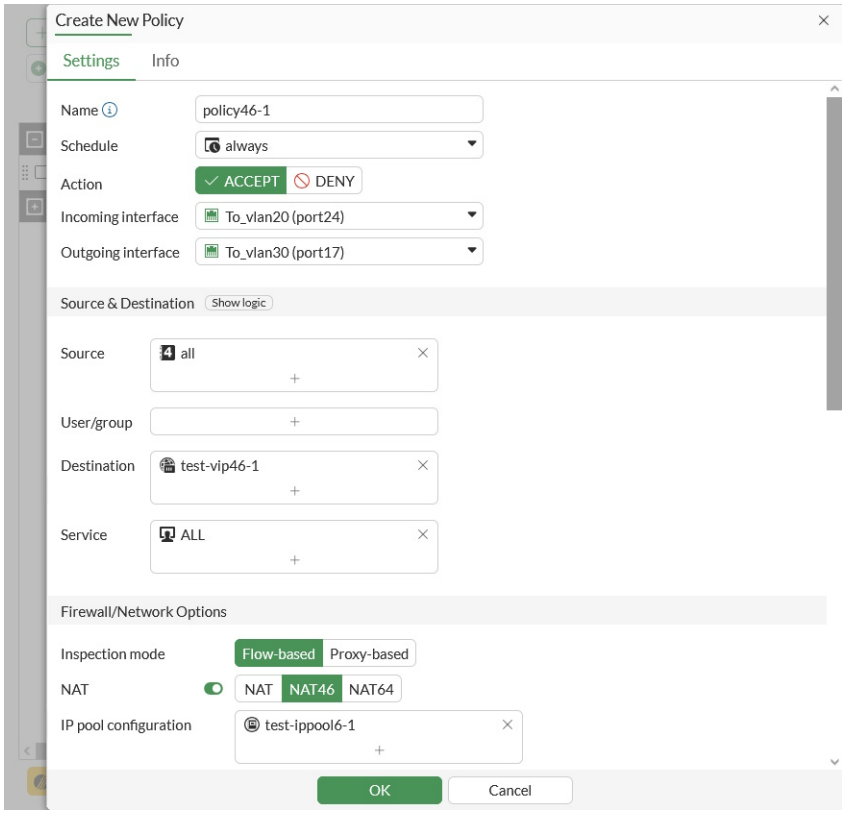
   c. Click *OK*.
2. Configure the IPv6 pool:
   a. Go to *Policy & Objects* > *IP Pools* and select the *IPv6 IP Pool* tab.
   b. Click *Create new*.
   c. Enter the following:

Name	test-ippool6-1
External IP address/range	2000:172:16:101::1-2000:172:16:101::1
NAT46	Enable

   d. Click *OK*.
3. Configure the firewall policy:
   a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New* or edit an existing policy.
   b. Enter the following:

Name	policy46-1
Schedule	always
Action	ACCEPT
Incoming Interface	To_vlan20
Outgoing Interface	To_vlan30
Source	all
Destination	test-vip46-1
Service	ALL
NAT	NAT46
IP Pool Configuration	test-ippool6-1

c. Configure the other settings as needed.



d. Click *OK*.

To create a NAT46 policy in the CLI:

1. Configure the VIP:

```
config firewall vip
    edit "test-vip46-1"
```

```
        set extip 10.1.100.150
        set nat44 disable
        set nat46 enable
        set extintf "port24"
        set arp-reply enable
        set ipv6-mappedip 2000:172:16:200::156
    next
end
```

2. Configure the IPv6 pool:

```
config firewall ippool6
    edit "test-ippool6-1"
        set startip 2000:172:16:101::1
        set endip 2000:172:16:101::1
        set nat46 enable
        set add-nat46-route enable
    next
end
```

3. Configure the firewall policy:

```
config firewall policy
    edit 2
        set name "policy46-1"
        set srcintf "port24"
        set dstintf "port17"
        set action accept
        set nat46 enable
        set srcaddr "all"
        set dstaddr "test-vip46-1"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set auto-asic-offload disable
        set ippool enable
        set poolname6 "test-ippool6-1"
    next
end
```

To verify the traffic and session tables:

1. Verify the traffic by the sniffer packets:

```
(root) # diagnose sniffer packet any 'icmp or icmp6' 4
interfaces=[any]
filters=[icmp or icmp6]
2.593302 port24 in 10.1.100.41 -> 10.1.100.150: icmp: echo request
2.593344 naf.root out 10.1.100.41 -> 10.1.100.150: icmp: echo request
2.593347 naf.root in 2000:172:16:101::1 -> 2000:172:16:200::156: icmp6: echo request seq 1
```

```
2.593383 port17 out 2000:172:16:101::1 -> 2000:172:16:200::156: icmp6: echo request seq 1
2.593772 port17 in 2000:172:16:200::156 -> 2000:172:16:101::1: icmp6: echo reply seq 1
2.593788 naf.root out 2000:172:16:200::156 -> 2000:172:16:101::1: icmp6: echo reply seq 1
2.593790 naf.root in 10.1.100.150 -> 10.1.100.41: icmp: echo reply
2.593804 port24 out 10.1.100.150 -> 10.1.100.41: icmp: echo reply
11 packets received by filter
0 packets dropped by kernel
```

2. Verify the session tables for IPv4 and IPv6:

```
(root) # diagnose sys session list
session info: proto=1 proto_state=00 duration=2 expire=59 timeout=0 flags=00000000 socktype=0
sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty f00 netflow-origin netflow-reply
statistic(bytes/packets/allow_err): org=252/3/1 reply=252/3/1 tuples=2
tx speed(Bps/kbps): 106/0 rx speed(Bps/kbps): 106/0
orgin->sink: org pre->post, reply pre->post dev=24->53/53->24 gwy=10.1.100.150/10.1.100.41
hook=pre dir=org act=noop 10.1.100.41:29388->10.1.100.150:8(0.0.0.0:0)
hook=post dir=reply act=noop 10.1.100.150:29388->10.1.100.41:0(0.0.0.0:0)
peer=2000:172:16:101::1:29388->2000:172:16:200::156:128 naf=1
hook=pre dir=org act=noop 2000:172:16:101::1:29388->2000:172:16:200::156:128(:::0)
hook=post dir=reply act=noop 2000:172:16:200::156:29388->2000:172:16:101::1:129(:::0)
misc=0 policy_id=2 auth_info=0 chk_client_info=0 vd=0
serial=00012b77 tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id=00000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x040001 no_offload
no_ofld_reason:  disabled-by-policy non-npu-intf
total session 1
```

```
(root) # diagnose sys session6 list
session6 info: proto=58 proto_state=00 duration=5 expire=56 timeout=0 flags=00000000
sockport=0 socktype=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=log may_dirty
statistic(bytes/packets/allow_err): org=312/3/0 reply=312/3/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=53->17/17->53
hook=pre dir=org act=noop 2000:172:16:101::1:29388->2000:172:16:200::156:128(:::0)
hook=post dir=reply act=noop 2000:172:16:200::156:29388->2000:172:16:101::1:129(:::0)
peer=10.1.100.150:29388->10.1.100.41:0 naf=2
hook=pre dir=org act=noop 10.1.100.41:29388->10.1.100.150:8(0.0.0.0:0)
hook=post dir=reply act=noop 10.1.100.150:29388->10.1.100.41:0(0.0.0.0:0)
misc=0 policy_id=2 auth_info=0 chk_client_info=0 vd=0
serial=00001bbc tos=ff/ff ips_view=1024 app_list=0 app=0 url_cat=0
```

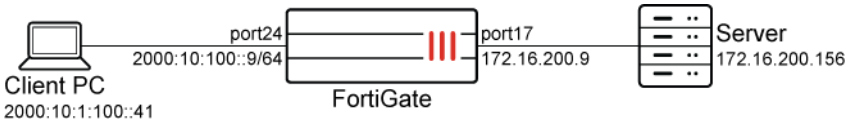
```
rpdb_link_id = 00000000 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
total session 1
```

The IPv4 session is between the incoming physical interface port24 and naf.root. The IPv6 session is between the naf.root and the outgoing physical interface port17.

NAT64 policy

In this example, a client PC is using IPv6 and an IPv6 VIP to access a server that is using IPv4. The FortiGate uses NAT64 to translate the request from IPv6 to IPv4 using the virtual interface naf.root. An `ippool` is applied so that the request is SNATed to the `ippool` address (172.16.101.2 - 172.16.101.3).

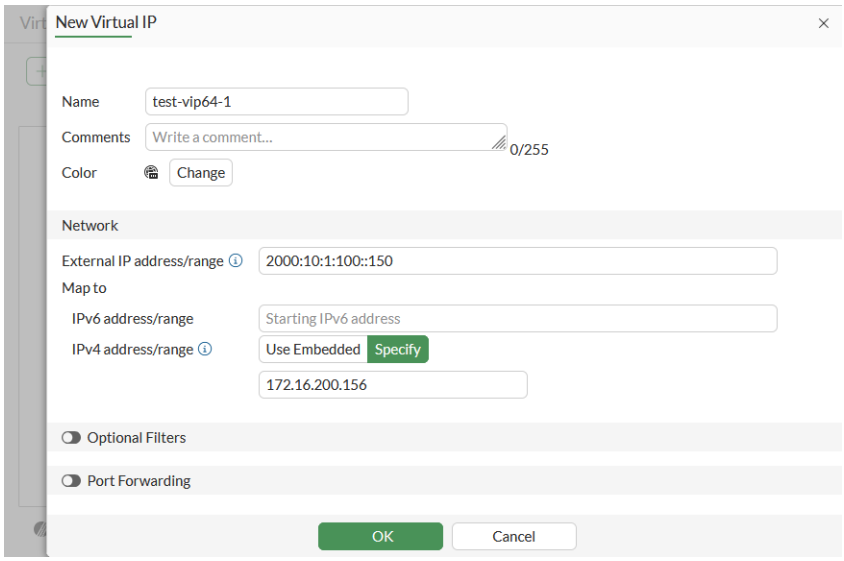
An embedded VIP64 object is used in this configuration so a specific IPv4 mapped IP does not need to be set. The lower 32 bits of the external IPv6 address are used to map to the IPv4 address. Only an IPv6 prefix is defined. In this example, the IPv6 prefix is 2001:10:1:100::, so the IPv6 address 2001:10:1:100::ac10:c89c will be translated to 172.16.200.156.



To create a NAT64 policy in the GUI:

1. Configure the VIP:
   a. Go to *Policy & Objects* > *Virtual IPs* and select the *IPv6 Virtual IP* tab.
   b. Click *Create new*.
   c. Enter the following:

Name	test-vip64-1
External IP address/range	2000:10:1:100::150
Map to IPv4 address/range	**Specify**: 172.16.200.156



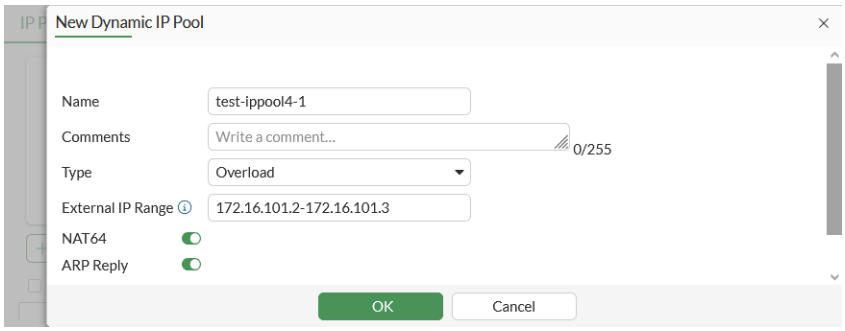
d. Click *OK*.

2. Configure the VIP with the embedded IPv4 address enabled:
   a. Go to *Policy & Objects* > *Virtual IPs* and select the *IPv6 Virtual IP* tab.
   b. Click *Create new*.
   c. Enter the following:

Name	test-vip64-2
External IP address/range	2001:10:1:100::-2001:10:1:100::ffff:ffff
Map to IPv4 address/range	Use Embedded

   d. Click *OK*.
3. Configure the IP pool:
   a. Go to *Policy & Objects* > *IP Pools* and select the *IP Pool* tab.
   b. Click *Create new*.
   c. Enter the following:

Name	test-ippool4-1
Type	Overload
External IP address/range	172.16.101.2-172.16.101.3
NAT64	Enable



d. Click *OK*.

4. Configure the firewall policy:

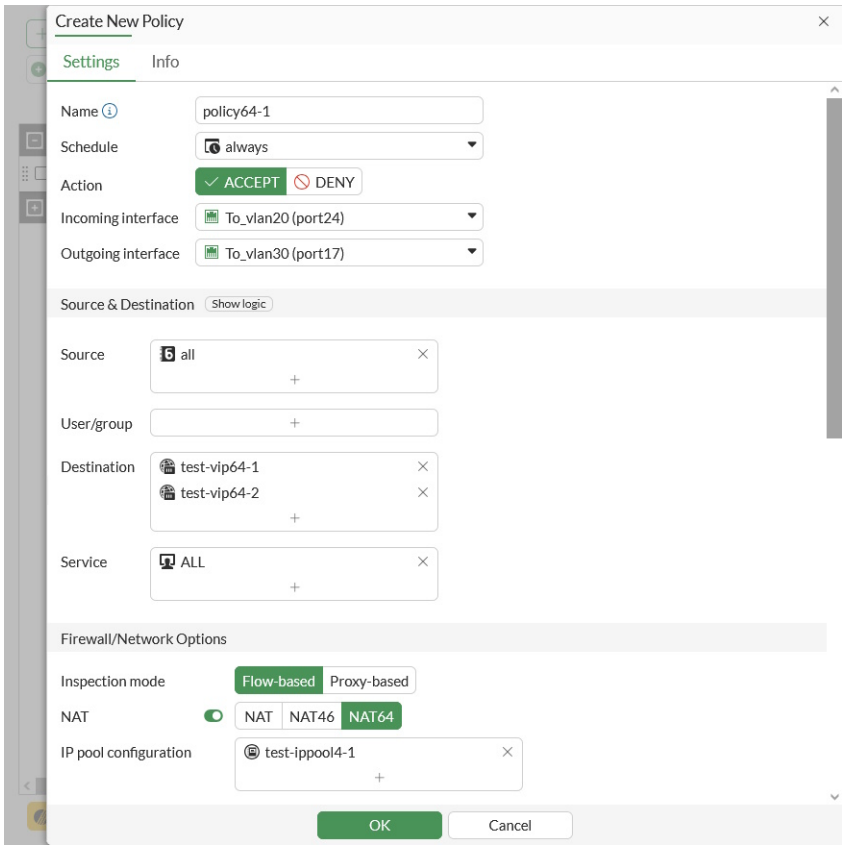
a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New* or edit an existing policy.

b. Enter the following:

c.

Name	policy64-1
Incoming Interface	To_vlan20
Outgoing Interface	To_vlan30
Source	all
Destination	test-vip64-1 test-vip64-2
Schedule	always
Service	ALL
Action	ACCEPT
NAT	NAT64
IP Pool Configuration	test-ippool4-1

d. Configure the other settings as needed.



e. Click *OK*.

To create a NAT64 policy in the CLI:

1. Configure the VIP:

```
config firewall vip6
    edit "test-vip64-1"
        set extip 2000:10:1:100::150
        set nat66 disable
        set nat64 enable
        set ipv4-mappedip 172.16.200.156
    next
end
```

2. Configure the VIP with the embedded IPv4 address enabled:

```
config firewall vip6
    edit "test-vip64-2"
        set extip 2001:10:1:100::-2001:10:1:100::ffff:ffff
        set nat66 disable
        set nat64 enable
        set embedded-ipv4-address enable
    next
end
```

3. Configure the IP pool:

```
config firewall ippool
    edit "test-ippool4-1"
        set startip 172.16.101.2
        set endip 172.16.101.3
        set nat64 enable
        set add-nat64-route enable
    next
end
```

4. Configure the firewall policy:

```
config firewall policy
    edit 1
        set name "policy64-1"
        set srcintf "port24"
        set dstintf "port17"
        set action accept
        set nat64 enable
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "test-vip64-1" "test-vip64-2"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set auto-asic-offload disable
        set ippool enable
        set poolname "test-ippool4-1"
    next
end
```

To verify the traffic and session tables:

1. Verify the VIP64 traffic by the sniffer packets:

```
(root) # diagnose sniffer packet any 'icmp or icmp6' 4
interfaces=[any]
filters=[icmp or icmp6]
20.578417 port24 in 2000:10:1:100::41 -> 2000:10:1:100::150: icmp6: echo request seq 1
20.578495 naf.root out 2000:10:1:100::41 -> 2000:10:1:100::150: icmp6: echo request seq 1
20.578497 naf.root in 172.16.101.2 -> 172.16.200.156: icmp: echo request
20.578854 port17 out 172.16.101.2 -> 172.16.200.156: icmp: echo request
20.579083 port17 in 172.16.200.156 -> 172.16.101.2: icmp: echo reply
20.579093 naf.root out 172.16.200.156 -> 172.16.101.2: icmp: echo reply
20.579095 naf.root in 2000:10:1:100::150 -> 2000:10:1:100::41: icmp6: echo reply seq 1
20.579377 port24 out 2000:10:1:100::150 -> 2000:10:1:100::41: icmp6: echo reply seq 1
11 packets received by filter
0 packets dropped by kernel
```

2. Verify the session tables for IPv6 and IPv4:

```
(root) # diagnose sys session6 list
session6 info: proto=58 proto_state=00 duration=5 expire=56 timeout=0 flags=00000000
sockport=0 socktype=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=log may_dirty
statistic(bytes/packets/allow_err): org=312/3/0 reply=312/3/0 tuples=2
tx speed(Bps/kbps): 55/0 rx speed(Bps/kbps): 55/0
orgin->sink: org pre->post, reply pre->post dev=24->53/53->24
hook=pre dir=org act=noop 2000:10:1:100::41:29949->2000:10:1:100::150:128(:::0)
hook=post dir=reply act=noop 2000:10:1:100::150:29949->2000:10:1:100::41:129(:::0)
peer=172.16.101.2:45392->172.16.200.156:8 naf=1
hook=pre dir=org act=noop 172.16.101.2:45392->172.16.200.156:8(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.200.156:45392->172.16.101.2:0(0.0.0.0:0)
misc=0 policy_id=1 auth_info=0 chk_client_info=0 vd=0
serial=000021ec tos=ff/ff ips_view=1024 app_list=0 app=0 url_cat=0
rpdb_link_id = 00000000 ngfwid=n/a
npu_state=0x040001 no_offload
no_ofld_reason:  disabled-by-policy non-npu-intf
total session 1
```

```
(root) # diagnose sys session list
session info: proto=1 proto_state=00 duration=7 expire=54 timeout=0 flags=00000000 socktype=0
sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty f00
statistic(bytes/packets/allow_err): org=252/3/1 reply=252/3/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=53->17/17->53 gwy=172.16.200.156/172.16.101.2
hook=pre dir=org act=noop 172.16.101.2:45392->172.16.200.156:8(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.200.156:45392->172.16.101.2:0(0.0.0.0:0)
peer=2000:10:1:100::150:29949->2000:10:1:100::41:129 naf=2
hook=pre dir=org act=noop 2000:10:1:100::41:29949->2000:10:1:100::150:128(:::0)
hook=post dir=reply act=noop 2000:10:1:100::150:29949->2000:10:1:100::41:129(:::0)
misc=0 policy_id=1 auth_info=0 chk_client_info=0 vd=0
serial=0001347f tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id=00000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
total session 1
```

The IPv6 session is between the incoming physical interface port24 and naf.root. The IPv4 session is between the naf.root and the outgoing physical interface port17.

3. Verify the embedded VIP64 traffic by the sniffer packets:

```
(root) # diagnose sniffer packet any 'icmp or icmp6' 4
interfaces=[any]
filters=[icmp or icmp6]
7.696010 port24 in 2000:10:1:100::41 -> 2001:10:1:100::ac10:c89c: icmp6: echo request seq 1
7.696057 naf.root out 2000:10:1:100::41 -> 2001:10:1:100::ac10:c89c: icmp6: echo request seq 1
7.696060 naf.root in 172.16.101.2 -> 172.16.200.156: icmp: echo request
7.696544 port17 out 172.16.101.2 -> 172.16.200.156: icmp: echo request
7.696821 port17 in 172.16.200.156 -> 172.16.101.2: icmp: echo reply
7.696839 naf.root out 172.16.200.156 -> 172.16.101.2: icmp: echo reply
7.696841 naf.root in 2001:10:1:100::ac10:c89c -> 2000:10:1:100::41: icmp6: echo reply seq 1
7.697167 port24 out 2001:10:1:100::ac10:c89c -> 2000:10:1:100::41: icmp6: echo reply seq 1
11 packets received by filter
0 packets dropped by kernel
```

Hairpin NAT

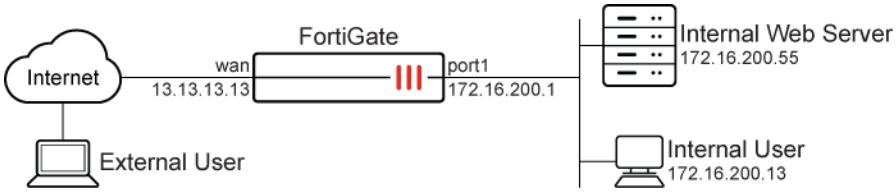
Hairpin NAT, also known as NAT loopback or U-turn NAT, allows devices on the same internal network to communicate using their external IP addresses. This simplifies network configurations and enhances connectivity.

How hairpin NAT works:

1. Request initiation: An employee within the office network attempts to access an internal resource using its external IP address.
2. NAT processing: The request is sent to the FortiGate, which recognizes that the destination IP address is its own external IP on the WAN interface.
3. Hairpin NAT: The FortiGate redirects the request back into the internal network, forwarding it to the web server's internal IP address and allowing the employee to access the internal web server using the external address.

Example

This configuration allows both internal and remote employees to use the same domain name to access the server. Hairpin NAT ensures consistency, avoiding confusion and complications in network configurations. This setup is particularly useful for companies that save bookmarks to internal web servers using public domain names in users' profiles, ensuring access whether connected to the internal LAN or the internet.



In this scenario, the user can access saved bookmarks containing public domain names from both their office computer on the internal network and their personal computer over the internet.

Before continuing, make sure that the port1 and wan interfaces have been configured with valid IP addresses and some publicly accessible domain names are saved as bookmarks in the user profile.

It is assumed that you have administrative access, the FortiGate is incorporated into your network, and the external IP address is the same as the FortiGate wan interface.

The configuration has four steps:

1. Create a VIP: Map the external IP address to the internal IP address.
2. Create a firewall policy for DNAT: Attach the VIP to a firewall policy, allowing external users to access the internal server using the external IP address. See Static virtual IPs on page 1944 for more information.
3. Create a firewall policy for SNAT: Allow internal users to access the external servers using the external IP address. See Static SNAT on page 1913 for more information.
4. Verify the result: Confirm that the employee can access the internal web server using the external address.

To create a VIP in the GUI:

1. Go to *Policy & Objects* > *Virtual IPs* or, if Central NAT is enabled, *Policy & Objects* > *DNAT & Virtual IPs*.
2. Select the *Virtual IP* and click *Create new*.
3. Configure the following:

Field	Value
Name	Internal_WebServer
Interface	any
Type	Static NAT
External IP address/range	13.13.13.13
IPv4 address/range	172.16.200.55

4. Click *OK*.

To create a firewall policy for DNAT in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy*, and click *Create New*.
2. Configure the following:

Field	Value
Name	wan-lan
Schedule	always
Action	ACCEPT
Type	Standard
Incoming interface	wan
Outgoing interface	port1
Source	all
Destination	Internal_WebServer
Service	ALL

Field	Value
NAT	Enabled
IP pool configuration	Use Outgoing Interface Address
Log allowed traffic	All sessions

3. Click *OK*.

To create a firewall policy for SNAT in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy*, and click *Create New*.
2. Configure the following:

Field	Value
Name	lan-wan
Schedule	always
Action	ACCEPT
Type	Standard
Incoming interface	port1
Outgoing interface	wan
Source	all
Destination	all
Service	ALL
NAT	Enabled
IP pool configuration	Use Outgoing Interface Address
Log allowed traffic	All sessions

3. Click *OK*.

To configure the VIP and policies in the CLI:

1. Create the VIP:

```
config firewall vip
    edit "Internal_WebServer "
        set extip 13.13.13.13
        set mappedip "172.16.200.55"
        set extintf "any"
    next
end
```

2. Create firewall policies for DNAT (`wan-lan`) and SNAT (`lan-wan`):

```
config firewall policy
    edit 2
        set name "wan-lan"
        set srcintf "wan"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "Internal_WebServer"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set nat enable
    next
    edit 3
        set name "lan-wan"
        set srcintf "port1"
        set dstintf "wan"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set nat enable
    next
end
```

To verify the results:

1. Access the internal machine and use the saved bookmark or the external IP address of the internal web server to access it. This should be successful, and the web server should load correctly. On the FortiGate, this can be verified using the logs and session list
2. On the FortiGate, go to *Log & Report > Forward Traffic* and double click the desired log entry to view its details, or use the CLI to view the logs:

```
#execute log filter category 0
#execute log display
1: date=2024-12-05 time=11:43:58 eventtime=1733427838055576829 tz="-0800" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="root" srcip=172.16.200.13 srcport=54134
srcintf="port1" srcintfrole="undefined" dstip=13.13.13.13 dstport=443 dstintf="wan"
dstintfrole="undefined" srccountry="Reserved" dstcountry="Reserved" sessionid=5797 proto=6
action="close" policyid=2 policytype="policy" poluuid="1dbc281e-b32c-51ef-f783-7b95542c1baf"
policyname="wan-lan" service="HTTPS" trandisp="snat+dnat" tranip=172.16.200.55 tranport=443
transip=172.16.200.1 transport=54134 appcat="unscanned" duration=2 sentbyte=2063 rcvdbyte=1748
sentpkt=8 rcvdpkt=5
```

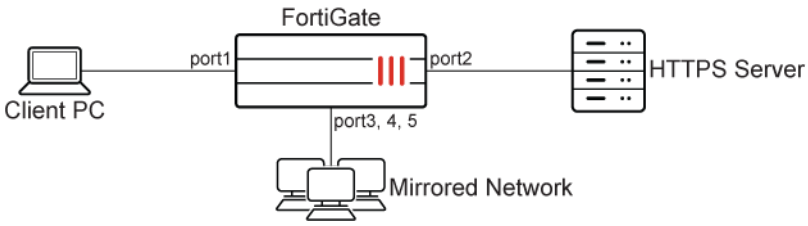
> If the logs are missing, go to *Log & Report > Forward Traffic* and increase the time frame to ensure that all pertinent log entries are available.

3. Check session list:

```
#get system session list | grep 13.13.13.13
PROTO   EXPIRE SOURCE           SOURCE-NAT       DESTINATION      DESTINATION-NAT
tcp     59     172.16.200.13:3  172.16.200.1:3   13.13.13.13:8    172.16.200.55:3
```

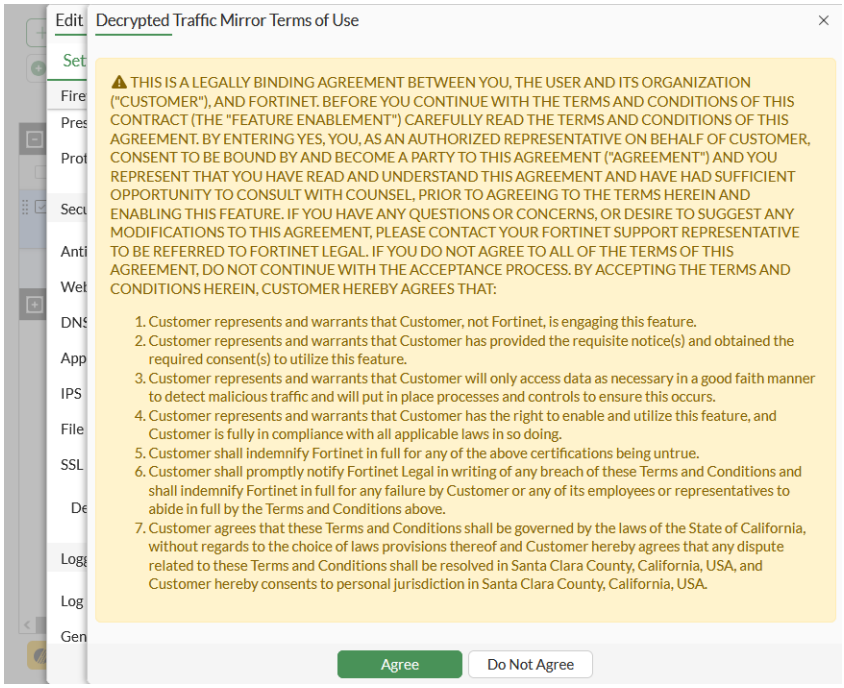
Mirroring SSL traffic in policies

SSL mirroring allows the FortiGate to decrypt and mirror traffic to a designated port. A new decrypted traffic mirror profile can be applied to IPv4, IPv6, and explicit proxy firewall policies in both flow and proxy mode. Full SSL inspection and in flow mode, a security profile, must be used in the policy for the traffic mirroring to occur. Proxy mode must be used if a security profile cannot be applied.

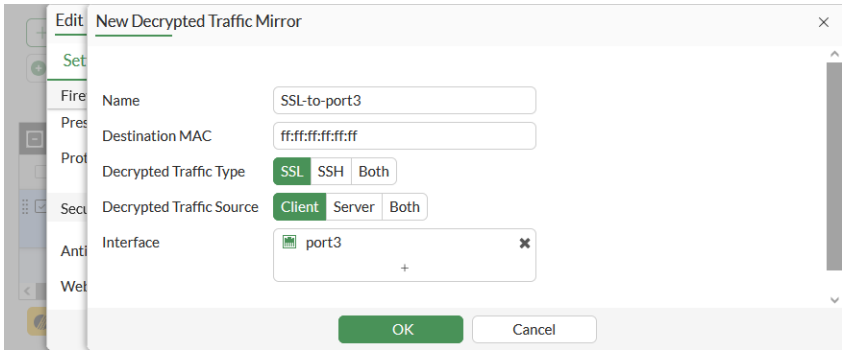


To configure SSL mirroring in a policy in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Create a new policy, or edit an existing one.
3. Configure the interfaces, sources, and other required information.
4. In the *Security Profiles* section, for *SSL Inspection*, select *deep-inspection*, or another profile that uses *Full SSL Inspection*.
5. If using flow mode, apply a security profile.
6. Enable *Decrypted Traffic Mirror* . The terms of use agreement opens.



7. Click *Agree* to accept the terms.
8. In the drop-down list, select a decrypted traffic mirror, or click *Create* to create a new one.
   In this example, a new decrypted traffic mirror is created using the port3 interface.



9. Click *OK* to save the policy.

To configure SSL mirroring in proxy mode in the CLI:

1. Create the decrypted traffic mirror profile:

```
config firewall decrypted-traffic-mirror
    edit SSL-to-port3
        set dstmac ff:ff:ff:ff:ff:ff
        set traffic-type ssl
        set traffic-source client
        set interface port3
    next
end
```

2. Configure the policy to enable SSL traffic mirroring. If using flow mode, a security profile must also be applied.

```
config firewall policy
    edit 1
        set name "mirror-policy"
        set srcintf "port1"
        set dstintf "port2"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set nat enable
        set ssl-ssh-profile "deep-inspection"
        set decrypted-traffic-mirror "SSL-to-port3"

THIS IS A LEGALLY BINDING AGREEMENT BETWEEN YOU, THE USER AND ITS ORGANIZATION ("CUSTOMER"),
AND FORTINET. BEFORE YOU CONTINUE WITH THE TERMS AND CONDITIONS OF THIS CONTRACT (THE "FEATURE
ENABLEMENT") CAREFULLY READ THE TERMS AND CONDITIONS OF THIS AGREEMENT. BY ENTERING YES, YOU,
AS AN AUTHORIZED REPRESENTATIVE ON BEHALF OF CUSTOMER, CONSENT TO BE BOUND BY AND BECOME A
PARTY TO THIS AGREEMENT ("AGREEMENT") AND YOU REPRESENT THAT YOU HAVE READ AND UNDERSTAND THIS
```

```
AGREEMENT AND HAVE HAD SUFFICIENT OPPORTUNITY TO CONSULT WITH COUNSEL, PRIOR TO AGREEING TO
THE TERMS HEREIN AND ENABLING THIS FEATURE. IF YOU HAVE ANY QUESTIONS OR CONCERNS, OR DESIRE
TO SUGGEST ANY MODIFICATIONS TO THIS AGREEMENT, PLEASE CONTACT YOUR FORTINET SUPPORT
REPRESENTATIVE TO BE REFERRED TO FORTINET LEGAL. IF YOU DO NOT AGREE TO ALL OF THE TERMS OF
THIS AGREEMENT, DO NOT CONTINUE WITH THE ACCEPTANCE PROCESS. BY ACCEPTING THE TERMS AND
CONDITIONS HEREIN, CUSTOMER HEREBY AGREES THAT:

1. Customer represents and warrants that Customer, not Fortinet, is engaging this feature.

2. Customer represents and warrants that Customer has provided the requisite notice(s) and
obtained the required consent(s) to utilize this feature.

3. Customer represents and warrants that Customer will only access data as necessary in a good
faith manner to detect malicious traffic and will put in place processes and controls to
ensure this occurs.

4. Customer represents and warrants that Customer has the right to enable and utilize this
feature, and Customer is fully in compliance with all applicable laws in so doing.

5. Customer shall indemnify Fortinet in full for any of the above certifications being untrue.

6. Customer shall promptly notify Fortinet Legal in writing of any breach of these Terms and
Conditions and shall indemnify Fortinet in full for any failure by Customer or any of its
employees or representatives to abide in full by the Terms and Conditions above.

7. Customer agrees that these Terms and Conditions shall be governed by the laws of the State
of California, without regards to the choice of laws provisions thereof and Customer hereby
agrees that any dispute related to these Terms and Conditions shall be resolved in Santa Clara
County, California, USA, and Customer hereby consents to personal jurisdiction in Santa Clara
County, California, USA.

Do you want to continue? (y/n) y
    next
end
```

Recognize anycast addresses in geo-IP blocking

An anycast IP can be advertised from multiple locations and the router selects a path based on latency, distance, cost, number of hops, and so on. This technique is widely used by providers to route users to the closest server. Since the IP is hosted in multiple geographic locations, there is no way to specify one single location to that IP.

Anycast IP address ranges can be bypassed in geo-IP blocking. The ISDB contains a list of confirmed anycast IP ranges that can be used for this purpose.

When the source or destination is set to `geoip`, you can enable the `geoip-anycast` option. Once enabled, IPs where the anycast option is set to `1` in `geoip_db` are bypassed in country matching and blocking.

You can only use the CLI to configure this feature.

To enable the geoip-anycast option using the CLI:

```
config firewall policy
    edit 1
        set name "policyid-1"
        set srcintf "wan2"
        set dstintf "wan1"
        set srcaddr "all"
        set dstaddr "test-geoip-CA_1"
        set action accept
        set schedule "always"
        set service "ALL"
        set geoip-anycast enable
        set logtraffic all
        set nat enable
    next
end
```

To check the geoip-anycast option for an IP address using the CLI:

```
# diagnose geoip ip2country 1.0.0.1
    1.0.0.1 - Australia, is anycast ip
```

The anycast IP is `1.0.0.1`.

Matching GeoIP by registered and physical location

IP addresses have both a physical and registered location in the geography IP database. Sometimes these two locations are different. The `geoip-match` command allows users to match an IPv4 address in an firewall policy to its physical or registered location when a GeoIP is used as a source or destination address. IPv6 policies currently support geography address objects but do not support `geoip-match`.

In the following example, the physical location of 220.243.219.10 is CA (Canada), the registered location is CN (China), and it is not an anycast IP.

To configure GeoIP matching based on registered location:

1. Create a firewall policy to match the IP:

```
config firewall policy
    edit 1
        set name "policy_id_1"
        set srcintf "wan2"
        set dstintf "wan1"
        set srcaddr "all"
        set dstaddr "test-geoip-CA"
        set action accept
        set schedule "always"
        set service "ALL"
        set geoip-match registered-location
```

```
        set logtraffic all
        set auto-asic-offload disable
        set nat enable
    next
end
```

Since CA is applied as a destination address and registered location IP matching is enabled, if the destination IP of the traffic is 220.243.219.10, then the traffic will be blocked because the registered location is CN.

2. Verify that the policy is blocking traffic from the IP address:

```
# diagnose sniffer packet any icmp 4
interfaces=[any]
filters=[icmp]
5.383798 wan2 in 10.1.100.41 -> 220.243.219.10: icmp: echo request
6.381982 wan2 in 10.1.100.41 -> 220.243.219.10: icmp: echo request
7.382608 wan2 in 10.1.100.41 -> 220.243.219.10: icmp: echo request
^C
3 packets received by filter
0 packets dropped by kernel
```

To configure GeoIP matching based on physical location:

1. Create a firewall policy to match the IP:

```
config firewall policy
    edit 1
        set name "policy_id_1"
        set srcintf "wan2"
        set dstintf "wan1"
        set srcaddr "all"
        set dstaddr "test-geoip-CA"
        set action accept
        set schedule "always"
        set service "ALL"
        set geoip-match physical-location
        set logtraffic all
        set auto-asic-offload disable
        set nat enable
    next
end
```

Since CA is applied as a destination address and physical location IP matching is enabled, if the destination IP of the traffic is 220.243.219.10, then the traffic will pass through.

2. Verify that the policy is allowing traffic from the IP address:

```
# diagnose sniffer packet any icmp 4
interfaces=[any]
filters=[icmp]
5.273985 wan2 in 10.1.100.41 -> 220.243.219.10: icmp: echo request
5.274176 wan1 out 172.16.200.10 -> 220.243.219.10: icmp: echo request
```

```
6.274426 wan2 in 10.1.100.41 -> 220.243.219.10: icmp: echo request
6.274438 wan1 out 172.16.200.10 -> 220.243.219.10: icmp: echo request
7.273978 wan2 in 10.1.100.41 -> 220.243.219.10: icmp: echo request
7.273987 wan1 out 172.16.200.10 -> 220.243.219.10: icmp: echo request
^C
6 packets received by filter
0 packets dropped by kernel
```

HTTP to HTTPS redirect for load balancing

You can configure a virtual server with HTTP to HTTPS redirect enabled. When enabled, a virtual server can convert a client's HTTP requests to HTTPS requests. Through this mandatory conversion, HTTP traffic is converted to HTTPS traffic. This conversion improves the security of the user network.

You can only enable this feature by using the CLI. After you enable this feature, traffic flows as follows:

- When FortiGate receives an HTTP request for an external IP, such as 10.1.100.201 in the following example, FortiGate sends an HTTP 303 response back to the original client and redirects HTTP to HTTPS, instead of forwarding the HTTP request to the real backend servers.
- The client browser restarts the TCP session to HTTPS.
- The HTTPS session comes to the FortiGate where a matching firewall policy allows the HTTPS traffic and establishes a secure SSL connection, and then forwards the request to the real backend servers.

To configure virtual server with HTTPS redirect enabled:

1. Create a virtual server with `server-type` set to `http`:

```
config firewall vip
    edit "virtual-server-http"
        set type server-load-balance
        set extip 10.1.100.201
        set extintf "wan2"
        set server-type http
        set ldb-method round-robin
        set extport 80
        config realservers
            edit 1
                set ip 172.16.200.44
                set port 80
            next
            edit 2
                set ip 172.16.200.55
                set port 80
            next
        end
    next
end
```

2. Create a virtual server with `server-type` set to `https` and with the same external IP address:

```
config firewall vip
    edit "virtual-server-https"
        set type server-load-balance
        set extip 10.1.100.201
        set extintf "wan2"
        set server-type https
        set ldb-method round-robin
        set extport 443
        config realservers
            edit 1
                set ip 172.16.200.44
                set port 443
            next
            edit 2
                set ip 172.16.200.55
                set port 443
            next
        end
        set ssl-certificate "Fortinet_SSL"
    next
end
```

3. Enable the `http-redirect` option for the virtual server with `server-type` set to `http`:

```
config firewall vip
    edit "virtual-server-http"
        <b>set http-redirect enable</b>
    next
end
```

4. Add the two virtual servers to a policy:

```
config firewall policy
    edit 9
        set srcintf "wan2"
        set dstintf "wan1"
        set srcaddr "all"
        set dstaddr "virtual-server-http" "virtual-server-https"
        set action accept
        set schedule "always"
        set service "ALL"
        set inspection-mode proxy
        set logtraffic all
        set auto-asic-offload disable
        set nat enable
    next
end
```

Use Active Directory objects directly in policies

Active Directory (AD) groups can be used directly in identity-based firewall policies. You do not need to add remote AD groups to local FSSO groups before using them in policies.

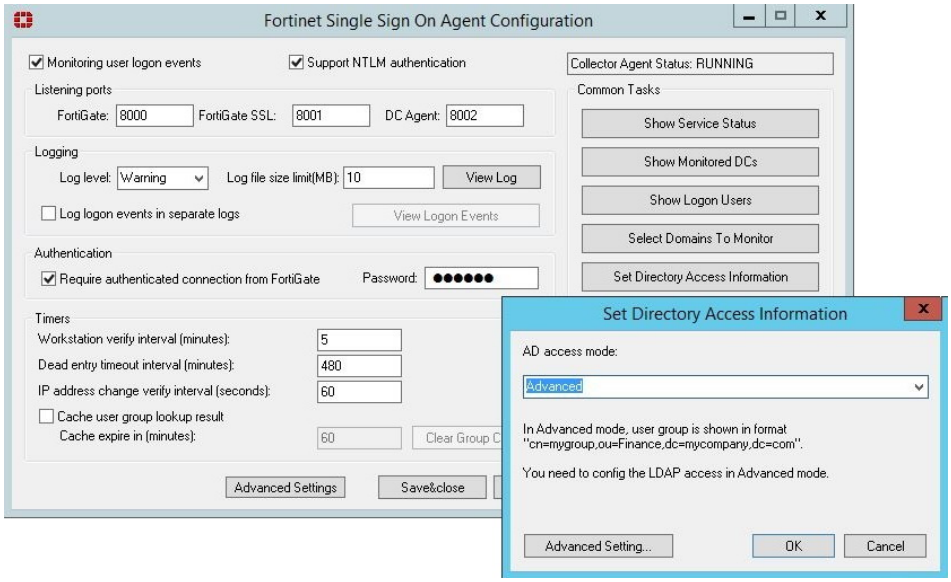
FortiGate administrators can define how often group information is updated from AD LDAP servers.

To retrieve and use AD user groups in policies:

1. Set the FSSO Collector Agent AD access mode on page 1988
2. Add an LDAP server on page 1989
3. Create the FSSO collector that updates the AD user groups list on page 1990
4. Use the AD user groups in a policy on page 1991

Set the FSSO Collector Agent AD access mode

To use this feature, you must set FSSO Collector Agent to *Advanced* AD access mode. If the FSSO Collector Agent is running in the default mode, FortiGate cannot correctly match user group memberships.



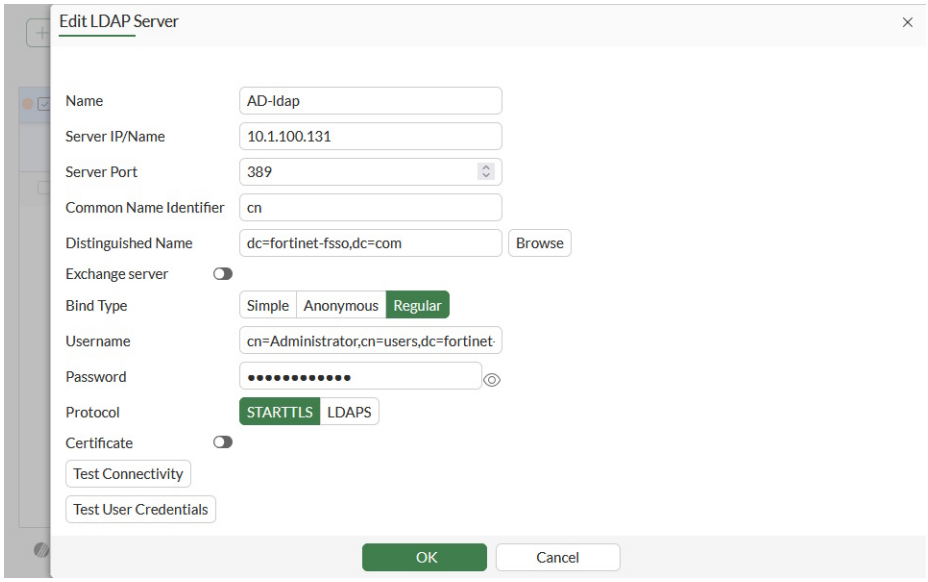
Add an LDAP server

When configuring an LDAP connection to an Active Directory server, an administrator must provide Active Directory user credentials.

- To secure this connection, use LDAPS on both the Active Directory server and FortiGate. See Configuring an LDAP server on page 3509 and Configuring client certificate authentication on the LDAP server on page 3524.
- Apply the principle of least privilege. For the LDAP regular bind operation, do not use credentials that provide full administrative access to the Windows server when using credentials. See Configuring least privileges for LDAP admin account authentication in Active Directory on page 3517.

To add an LDAP server in the GUI:

1. Go to *User & Authentication* > *LDAP Servers*.
2. Click *Create New*.
3. Configure the settings as needed.



4. If secure communication over TLS is not supported by the remote AD LDAP server, disable the secure connection in the CLI:

```
config user ldap
    edit "AD-ldap"
        set secure disable
    next
end
```

5. Click *OK*.

To add an LDAP server in the CLI:

```
config user ldap
    edit "AD-ldap"
        set server "10.1.100.131"
        set cnid "cn"
        set dn "dc=fortinet-fsso,dc=com"
        set type regular
        set username "cn=Administrator,cn=users,dc=fortinet-fsso,dc=com"
        set password XXXXXXXXXXXXXXXXXXXXXXXX
    next
end
```

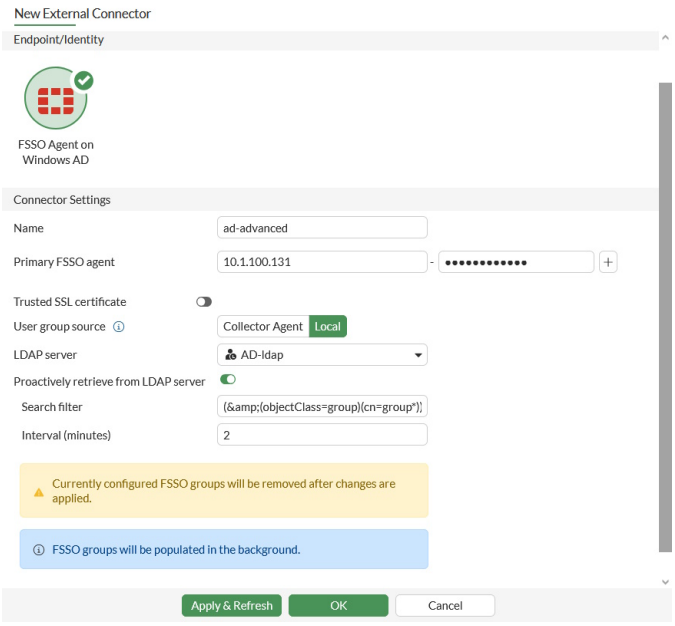
Create the FSSO collector that updates the AD user groups list

To create an FSSO agent connector in the GUI:

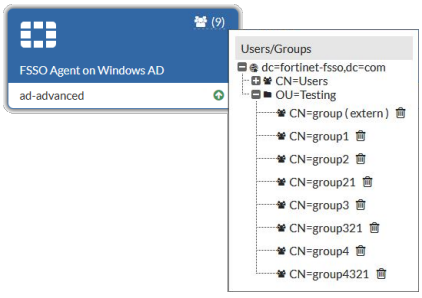
1. Go to *Security Fabric > External Connectors*.
2. Click *Create New*.
3. In the *Endpoint/Identity* section, click *FSSO Agent on Windows AD*.
4. Fill in the *Name*
5. Set the *Primary FSSO Agent* to the IP address of the FSSO Collector Agent, and enter its password.
6. Set the *User group source* to *Local*.
7. Set the *LDAP server* to the just created *AD-ldap* server.
8. Enable *Proactively retrieve from LDAP server*.
9. Set the *Search filter* to *(&(objectClass=group)(cn=group*))*.

   The default search filter retrieves all groups, including Microsoft system groups. In this example, the filter is configured to retrieve *group1*, *group2*, etc, and not groups like *grp199*.

   The filter syntax is not automatically checked; if it is incorrect, the FortiGate might not retrieve any groups.
10. Set the *Interval (minutes)* to configure how often the FortiGate contacts the remote AD LDAP server to update the group information.



11. Click *OK*.
12. To view the AD user groups that are retrieved by the FSSO agent, hover the cursor over the group icon on the fabric connector listing.



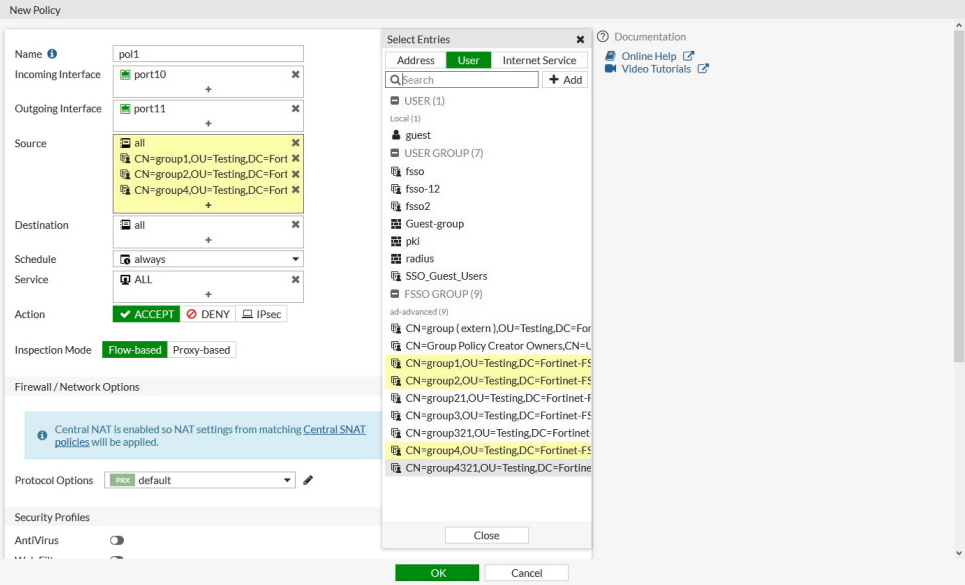
To create an FSSO agent connector in the CLI:

```
config user fsso
    edit "ad-advanced"
        set server "10.1.100.131"
        set password XXXXXXXXXXXXXX
        set ldap-server "AD-ldap"
        set ldap-poll enable
        set ldap-poll-interval 2
        set ldap-poll-filter "(&amp;(objectClass=group)(cn=group*))"
    next
end
```

You can view the retrieved AD user groups with the `show user adgrp` command.

Use the AD user groups in a policy

The AD user groups retrieved by the FortiGate can be used directly in firewall policies.



No session timeout

To allow clients to permanently connect with legacy medical applications and systems that do not have keepalive or auto-reconnect features, the session timeout can be set to never for firewall services, policies, and VDOMs.

The options to disable session timeout are hidden in the CLI.

To set the session TTL value of a custom service to never:

```
config firewall service custom
    edit "tcp_23"
        set tcp-portrange 23
        set session-ttl never
    next
end
```

To set the session TTL value of a policy to never:

```
config firewall policy
    edit 201
        set srcintf "wan1"
        set dstintf "wan2"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "TCP_8080"
        set logtraffic disable
        set session-ttl never
        set nat enable
```

```
    next
end
```

To set the session TTL value of a VDOM to never:

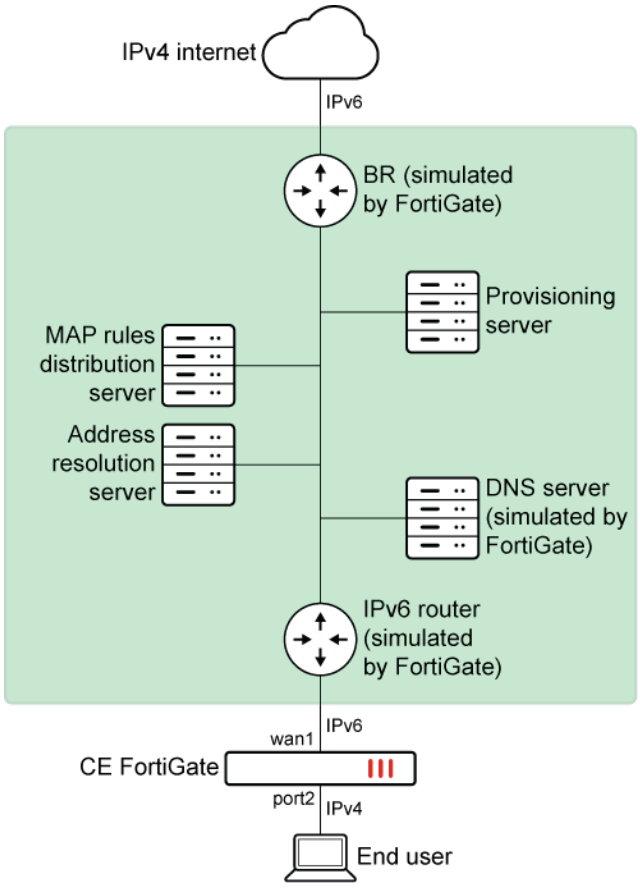
```
config system session-ttl
    set default never
    config port
        edit 1
            set protocol 6
            set timeout never
            set start-port 8080
            set end-port 8080
        next
    end
end
```

To view a session list with the timeout set to never:

```
# diagnose sys session list

session info: proto=6 proto_state=01 duration=9 expire=never timeout=never flags=00000000
sockflag=00000000 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty f00
statistic(bytes/packets/allow_err): org=2290/42/1 reply=2895/34/1 tuples=2
tx speed(Bps/kbps): 238/1 rx speed(Bps/kbps): 301/2
orgin->sink: org pre->post, reply pre->post dev=18->17/17->18 gwy=172.16.200.55/10.1.100.41
hook=post dir=org act=snat 10.1.100.41:34256->172.16.200.55:23(172.16.200.10:34256)
hook=pre dir=reply act=dnat 172.16.200.55:23->172.16.200.10:34256(10.1.100.41:34256)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=9 auth_info=0 chk_client_info=0 vd=1
serial=00000b27 tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id = 00000000 ngfwid=n/a
dd_type=0 dd_mode=0
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
total session 1
```

MAP-E support

On a customer edge (CE) FortiGate, an IPv4-over-IPv6 (MAP-E) tunnel can be created between the FortiGate and the border relay (BR) operating in an IPv6 network. A tunnel interface is created between the FortiGate and BR, which can be applied to firewall policies and IPsec VPN.



To configure a MAP-E tunnel between the FortiGate and the BR:

1. Configure fixed IP mode.
   a. Configure IPv6 on the interface:

```
config system interface
    edit "wan1"
        config ipv6
            set autoconf enable
            set unique-autoconf-addr enable
            set interface-identifier ::6f:6c1f:3400:0
        end
    next
end
```

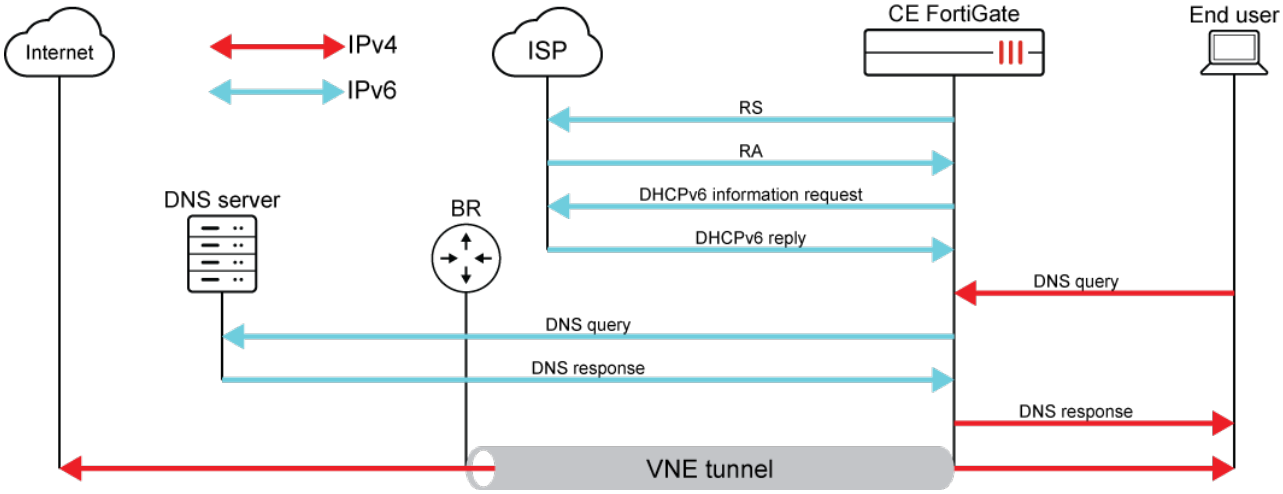
   The `interface-identifier` is an IPv6 address. Its last 64-bit will be kept and the rest will be cleared automatically. It will combine with the IPv6 prefix it gets from the IPv6 router to generate the IPv6 address of the interface.

   By default, `unique-autoconf-addr` is disabled. It must be enabled so it can handle IPv6 prefix changing.

   b. Configure the VNE tunnel:

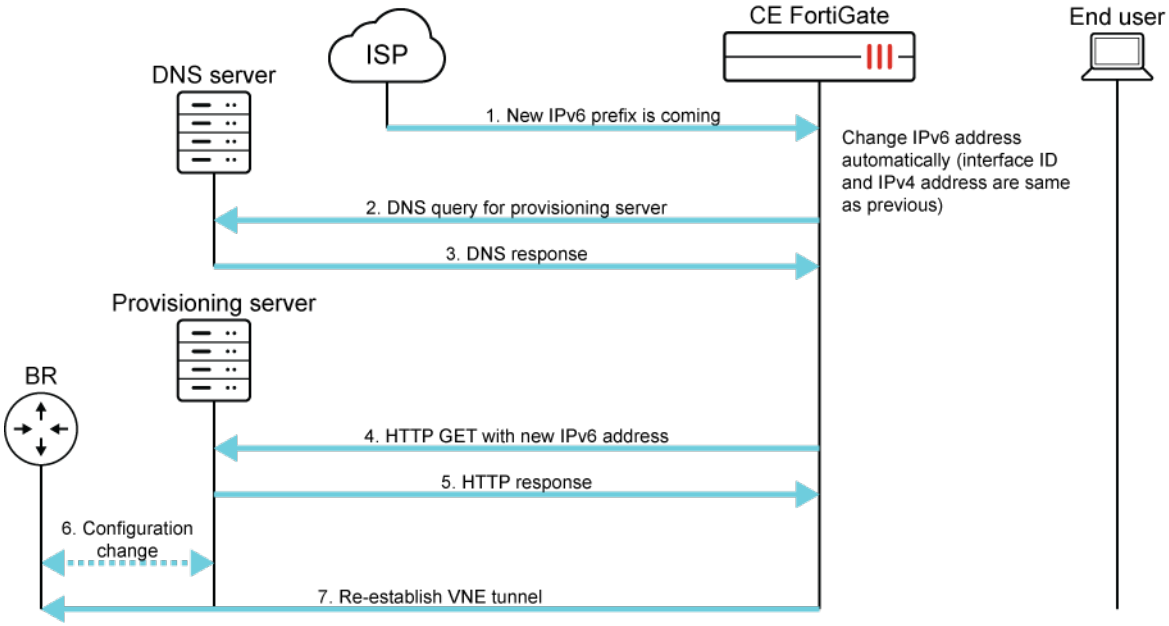
```
config system vne-interface
    edit vne1
        set interface "wan1"
        set mode fixed-ip
        set ipv4-address 10.10.81.81 255.255.255.255
        set br 2001:160::82
        set update-url http://qa.forosqa.com/update?user=xxxx&pass=yyyy
    next
end
```

Initial sequence overview of VNE tunnel under fixed IP mode:



Once the IPv6 address of the FortiGate changes, the tunnel will be down because the BR does not know the FortiGate's new IPv6 address. The FortiGate uses `update-url` to update the new IPv6 address to the provisioning server. The provisioning server updates the FortiGate's IPv6 address to the BR so the VNE tunnel can be re-established.

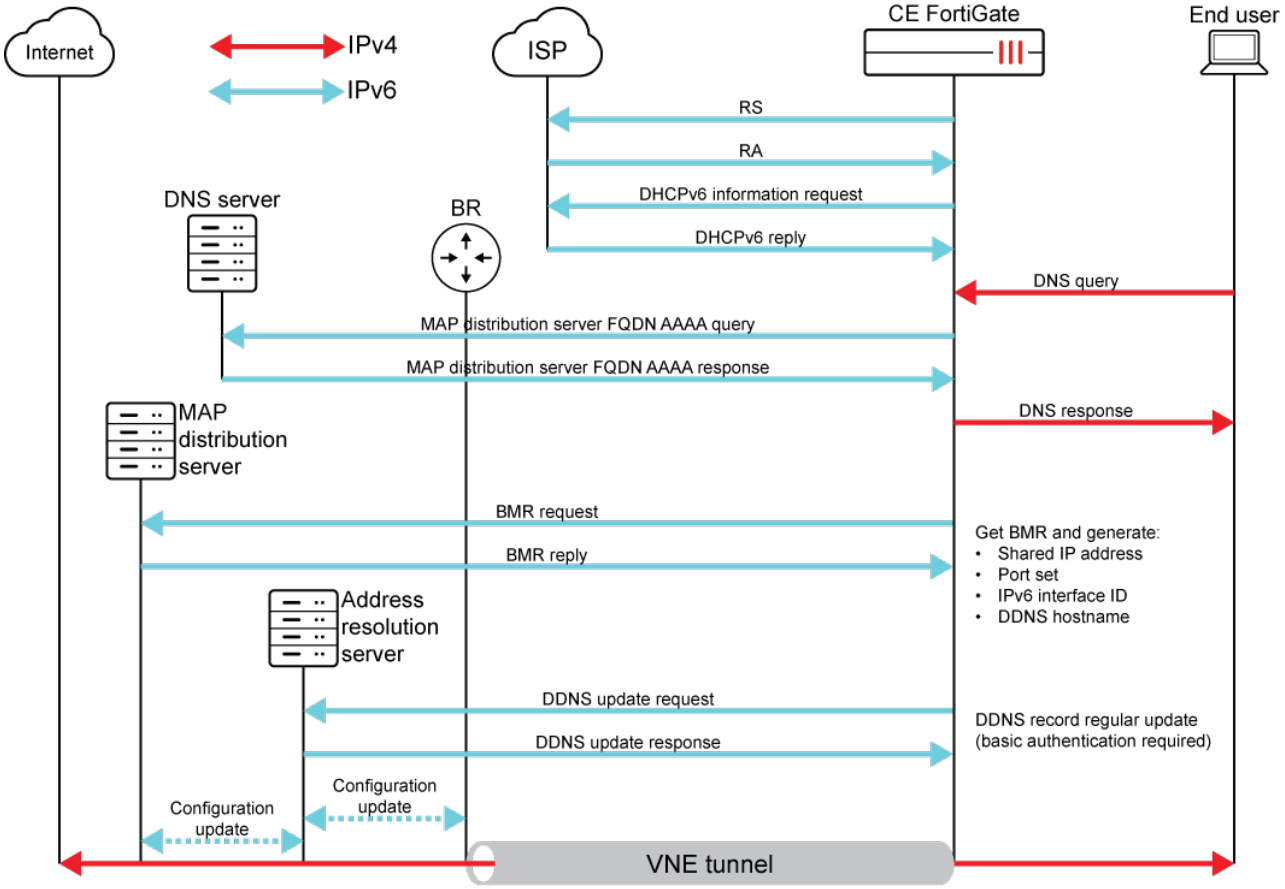
Communication sequence overview of re-establishing VNE tunnel:



2. Configure the VNE tunnel to use MAP-E mode:

```
config system vne-interface
    edit vne1
        set interface 'wan1'
        set ssl-certificate "Fortinet_Factory"
        set bmr-hostname ********
        set auto-asic-offload enable
        set mode map-e
    next
end
```

Initial sequence overview of VNE tunnel under MAP-E mode:



The FortiGate sends a MAP rule request to the MAP distribution server once the IPv6 address is configured on the FortiGate by RS/RA. Next, the FortiGate will send an AAAA query to get the IPv6 address of the MAP distribution server. After sending the BMR request to the MAP distribution server, the FortiGate will get the IPv4 address, port set, BR IPv6 address, and hostname of the address resolution server from the BMR reply. The VNE tunnel between the FortiGate and BR is now established.

The address resolution server is actually a dynamic DNS. The hostname is used for the FortiGate to maintain an IPv6 address when it changes.

The FortiGate updates the DDNS server with its IPv6 address whenever it updates, which in turn provides the update to the MAP distribution server and BR so they know how to resolve the FortiGate by hostname.

Once the VNE tunnel is established, an IPv4-over-IPv6 tunnel is set up between the FortiGate and BR. The route, firewall policy, and DNS server can now be configured to let the traffic go through the VNE tunnel and

protect the end-user. The VNE tunnel can also be used in IPsec phase 1.

3. Configure the route:

```
config router static
    edit 1
        set device "vne1"
    next
end
```

4. Configure the firewall policy:

```
config firewall policy
    edit 111
        set name "ff"
        set srcintf "port2"
        set dstintf "vne1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "certificate-inspection"
        set av-profile "default"
        set nat enable
    next
end
```

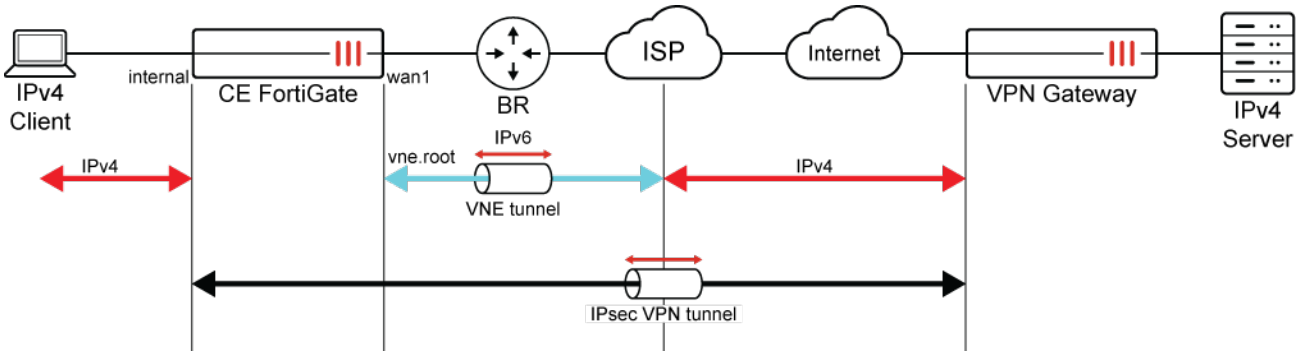
5. Configure the DNS server:

```
config system dns-server
    edit "port2"
    next
end
```

MAP-E can also operate in DHCPv6-PD environments; see DHCP-PD support for MAP-E on page 1998 for more information.

IPsec over VNE and offloading

In some scenarios, after the VNE tunnel is established, users might want to build IPsec tunnels over VNE for end-to-end encryption. This allows IPsec traffic to be encapsulated within the VNE tunnel to pass through the IPv6 ISP infrastructure and stay encrypted as it reaches the VPN Gateway and final destination.



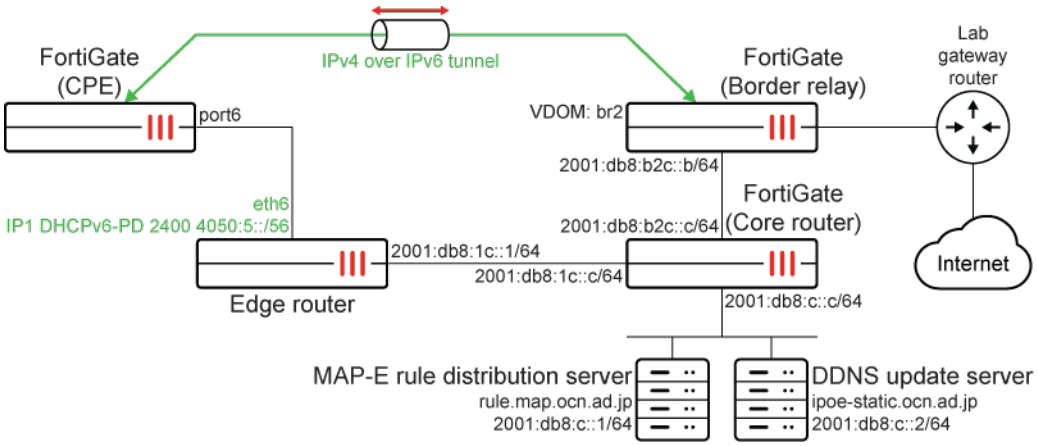
In these scenarios, NP-offloading (NP7 and NP7Lite) is supported over the IPsec tunnel when the following criteria are met:

1. The interface settings of the IPsec phase1-interface must be either a physical, trunk, or npu-vlink interface, or a VLAN on one of these interfaces.
2. The interface setting of the IPsec phase1-interface cannot be set to the VNE tunnel interface.
3. If the npu-vlink is used, the VNE and IPsec tunnels must be in different VDOMs.

DHCP-PD support for MAP-E

Mapping of address and port with encapsulation (MAP-E) can operate in DHCPv6 prefix delegation (DHCPv6-PD) environments, providing greater flexibility, improved automation, and scalability in network configurations. Previously, MAP-E utilized the RA IPv6 prefix for deployment; see MAP-E support on page 1993.

MAP-E mode



To configure MAP-E mode:

1. Configure the border relay FortiGate:

```
config system interface
    edit "core-br21"
        set vdom "br2"
        set type vdom-link
```

```
        set snmp-index 27
        config ipv6
            set ip6-address 2001:db8:b2c::b/64
            set ip6-allowaccess ping
        end
    next
end
config system ipv6-tunnel
    edit "6"
        set source 2001:db8:b2c::b
        set destination 2400:4050:6:0:c0:2:2200:0
    next
end
```

2. Configure the CPE FortiGate:
   a. Configure port6 to get an IPv6 address through the delegate:

```
config system interface
    edit "port6"
        set vdom "root"
        set type physical
        set snmp-index 6
        config ipv6
            set ip6-mode delegated
            set ip6-allowaccess ping
            set dhcp6-prefix-delegation enable
            set ip6-delegated-prefix-iaid 1
            set ip6-upstream-interface "port6"
            set ip6-subnet ::1/64
            config dhcp6-iapd-list
                edit 1
                    set prefix-hint ::/56
                next
            end
        end
    next
end
```

```
# dianose ipv ad list | grep port6
dev=7 devname=port6 flag= scope=0 prefix=64 addr=2400:4050:6::1 preferred=22 valid=45
cstamp=9913 tstamp=6047755
dev=7 devname=port6 flag=P scope=253 prefix=64 addr=fe80::20c:29ff:fe7b:d83b
preferred=4294967295 valid=4294967295 cstamp=6263 tstamp=6263
```

   b. Configure the virtual network enabler (VNE) tunnel:

```
config system vne-interface
    edit "vne.port6"
        set interface "port6"
    next
end
```

```
# diagnose test application vned 1
------------------------------------------------------------------------------
vdom: root/0, is master, devname=port6 link=1 tun=vne.port6 mode=map-e ssl_cert=Fortinet_
Factory
end user ipv6 perfix: 2400:4050:6::/56
interface ipv6 addr: 2400:4050:6::1
bmr rule ipv6 perfix: 2400:4050:6::/56
bmr rule ipv4 perfix: 192.0.2.34/32
bmr rule br: 2001:db8:b2c::b
bmr rule hostname: Testbed6_dhcpv6-pd-IP1
saved hostname: Testbed6_dhcpv6-pd-IP1
tunnel br: 2001:db8:b2c::b
tunnel ipv6 addr: 2400:4050:6:0:c0:2:2200:0
tunnel ipv4 addr: 192.0.2.34/255.255.255.255
update result: good

Map-e rule client: state=succeed retries=0 interval=60 expiry=23634 reply_code=200
fqdn=rule.map.ocn.ad.jp num=1 cur=0 ttl=86400 expiry=0
2001:db8:c::1
Map-e DDNS client: state=succeed retries=0 interval=120 expiry=61372 reply_code=200
fqdn=ipoe-static.ocn.ad.jp num=1 cur=0 ttl=86400 expiry=0
2001:db8:c::2
```

vne.port6 is automatically configured after the VNE tunnel is set up:

```
config system interface
    edit "vne.port6"
        set vdom "root"
        set ip 192.0.2.34 255.255.255.255
        set type tunnel
        set snmp-index 16
        set interface "port6"
    next
end
```

```
# diagnose ipv ad li | grep port6
dev=7 devname=port6 flag=Pn scope=0 prefix=128 addr=2400:4050:6:0:c0:2:2200:0
preferred=4294967295 valid=4294967295 cstamp=10620 tstamp=10620
dev=7 devname=port6 flag= scope=0 prefix=64 addr=2400:4050:6::1 preferred=37 valid=60
cstamp=9913 tstamp=6108593
dev=7 devname=port6 flag=P scope=253 prefix=64 addr=fe80::20c:29ff:fe7b:d83b
preferred=4294967295 valid=4294967295 cstamp=6263 tstamp=6263
```

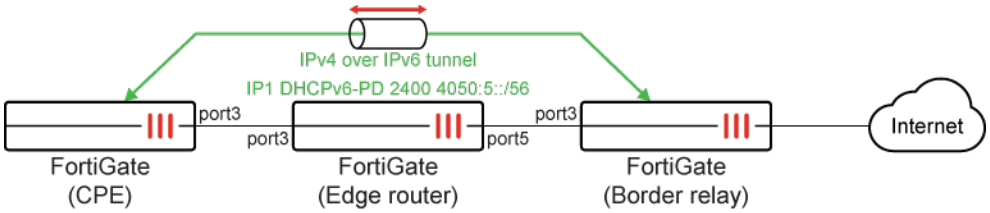
c. Traffic can now pass through the VNE tunnel:

```
config router static
    edit 1
        set device "vne.port6"
    next
end
```

```
# execute ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: icmp_seq=0 ttl=116 time=5.3 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=116 time=4.1 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=116 time=3.7 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=116 time=4.2 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=116 time=4.0 ms

--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 3.7/4.2/5.3 ms
```

```
# diagnose sniffer packet port6 'ip6'
Using Original Sniffing Mode
interfaces=[port6]
filters=[ip6]
pcap_lookupnet: port6: no IPv4 address assigned
3.091513 2400:4050:6:0:c0:2:2200:0 -> 2001:db8:b2c::b: 192.0.2.34 -> 8.8.8.8: icmp: echo
request
3.096651 2001:db8:b2c::b -> 2400:4050:6:0:c0:2:2200:0: 8.8.8.8 -> 192.0.2.34: icmp: echo
reply
4.091842 2400:4050:6:0:c0:2:2200:0 -> 2001:db8:b2c::b: 192.0.2.34 -> 8.8.8.8: icmp: echo
request
4.095918 2001:db8:b2c::b -> 2400:4050:6:0:c0:2:2200:0: 8.8.8.8 -> 192.0.2.34: icmp: echo
reply
5.092029 2400:4050:6:0:c0:2:2200:0 -> 2001:db8:b2c::b: 192.0.2.34 -> 8.8.8.8: icmp: echo
request
5.095576 2001:db8:b2c::b -> 2400:4050:6:0:c0:2:2200:0: 8.8.8.8 -> 192.0.2.34: icmp: echo
reply
6.092221 2400:4050:6:0:c0:2:2200:0 -> 2001:db8:b2c::b: 192.0.2.34 -> 8.8.8.8: icmp: echo
request
6.096358 2001:db8:b2c::b -> 2400:4050:6:0:c0:2:2200:0: 8.8.8.8 -> 192.0.2.34: icmp: echo
reply
7.092372 2400:4050:6:0:c0:2:2200:0 -> 2001:db8:b2c::b: 192.0.2.34 -> 8.8.8.8: icmp: echo
request
7.096299 2001:db8:b2c::b -> 2400:4050:6:0:c0:2:2200:0: 8.8.8.8 -> 192.0.2.34: icmp: echo
reply
```

Fixed IP mode



To configure fixed IP mode:

1. Configure the border relay FortiGate:

```
config system ipv6-tunnel
    edit "VNE-IPV6-2"
        set source 2606:f900:8102:302::2
        set destination 2606:f900:8102:301:8000::
        set interface "port3"
    next
end
config system interface
    edit "port3"
        set vdom "root"
        set type physical
        set snmp-index 9
        config ipv6
            set ip6-address 2606:f900:8102:302::2/64
            set ip6-allowaccess ping
        end
    next
end
```

2. Configure the CPE FortiGate:
   a. Configure port3 to get an IPv6 address through the delegate:

```
config system interface
    edit "port3"
        set vdom "root"
        set type physical
        set snmp-index 9
        config ipv6
            set ip6-mode delegated
            set ip6-allowaccess ping
            set dhcp6-prefix-delegation enable
            set ip6-delegated-prefix-iaid 1
            set ip6-upstream-interface "port3"
            set ip6-subnet ::8000:0:0:0/64
            config dhcp6-iapd-list
                edit 1
                    set prefix-hint 2606:f900:8102:301::/64
                next
                edit 2
                    set prefix-hint 2606:f900:8102:303::/64
                next
            end
        end
    next
end
```

```
# diagnose ipv ad list | grep port3
dev=13 devname=port3 flag=P scope=0 prefix=64 addr=2606:f900:8102:301:8000::
```

```
preferred=4294967295 valid=4294967295 cstamp=61106 tstamp=61106
dev=13 devname=port3 flag=P scope=253 prefix=64 addr=fe80::e223:ffff:fe67:8e6e
preferred=4294967295 valid=4294967295 cstamp=56882 tstamp=56882
```

b. Configure the VNE tunnel:

```
config system vne-interface
    edit "VNE"
        set interface "port3"
        set ipv4-address 10.100.100.1 255.255.255.255
        set br "2606:f900:8102:302::2"
        set update-url "http://test.com"
        set mode fixed-ip
    next
end
```

```
# diagnose test application vned 1
------------------------------------------------------------------------------
vdom: root/0, is master, devname=port3 link=1 tun=VNE mode=fixed-ip ssl_cert=Fortinet_
Factory
end user ipv6 perfix: 2606:f900:8102:301::/64
interface ipv6 addr: 2606:f900:8102:301:8000::
config ipv4 perfix: 10.100.100.1/255.255.255.0
config br: 2606:f900:8102:302::2
HTTP username:
update url: http://test.com
host: test.com path: / port:80 ssl: 0
tunnel br: 2606:f900:8102:302::2
tunnel ipv6 addr: 2606:f900:8102:301:8000::
tunnel ipv4 addr: 10.100.100.1/255.255.255.0
update result:
Fixed IP rule client: state=succeed retries=0 interval=0 expiry=0 reply_code=0
fqdn=2606:f900:8102:302::2 num=1 cur=0 ttl=4294967295 expiry=0
2606:f900:8102:302::2
Fixed IP DDNS client: state=init retries=0 interval=10 expiry=4 reply_code=0
fqdn=test.com num=0 cur=0 ttl=0 expiry=0
```

c. Traffic can now pass through the VNE tunnel:

```
# execute ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8): 56 data bytes
64 bytes from 8.8.8.8: icmp_seq=0 ttl=52 time=5.0 ms
64 bytes from 8.8.8.8: icmp_seq=1 ttl=52 time=6.4 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=52 time=4.7 ms
64 bytes from 8.8.8.8: icmp_seq=3 ttl=52 time=4.6 ms
64 bytes from 8.8.8.8: icmp_seq=4 ttl=52 time=4.6 ms

--- 8.8.8.8 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 4.6/5.0/6.4 mss
```

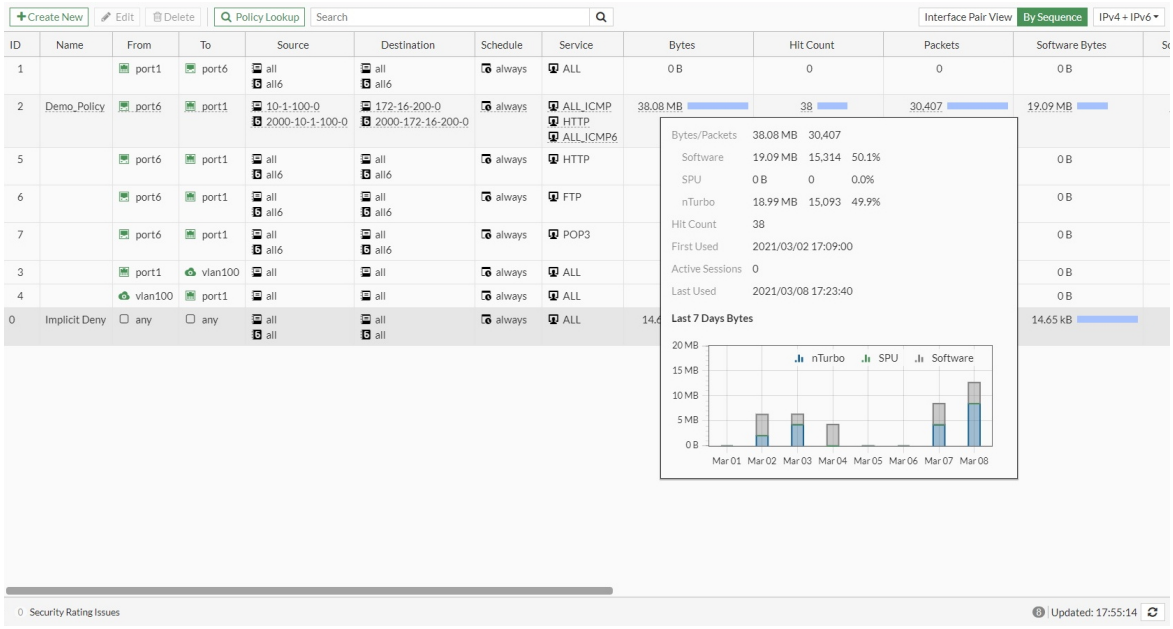
```
# diagnose sniffer packet port3 'ip6'
interfaces=[port3]
filters=[ip6]
4.319283 2606:f900:8102:301:8000:: -> 2606:f900:8102:302::2: 10.100.100.1 -> 8.8.8.8:
icmp: echo request
4.324193 2606:f900:8102:302::2 -> 2606:f900:8102:301:8000::: 8.8.8.8 -> 10.100.100.1:
icmp: echo reply
5.331276 2606:f900:8102:301:8000:: -> 2606:f900:8102:302::2: 10.100.100.1 -> 8.8.8.8:
icmp: echo request
5.337550 2606:f900:8102:302::2 -> 2606:f900:8102:301:8000::: 8.8.8.8 -> 10.100.100.1:
icmp: echo reply
6.341289 2606:f900:8102:301:8000:: -> 2606:f900:8102:302::2: 10.100.100.1 -> 8.8.8.8:
icmp: echo request
6.345900 2606:f900:8102:302::2 -> 2606:f900:8102:301:8000::: 8.8.8.8 -> 10.100.100.1:
icmp: echo reply
7.351258 2606:f900:8102:301:8000:: -> 2606:f900:8102:302::2: 10.100.100.1 -> 8.8.8.8:
icmp: echo request
7.355743 2606:f900:8102:302::2 -> 2606:f900:8102:301:8000::: 8.8.8.8 -> 10.100.100.1:
icmp: echo reply
8.361257 2606:f900:8102:301:8000:: -> 2606:f900:8102:302::2: 10.100.100.1 -> 8.8.8.8:
icmp: echo request
8.365788 2606:f900:8102:302::2 -> 2606:f900:8102:301:8000::: 8.8.8.8 -> 10.100.100.1:
icmp: echo reply
```

Seven-day rolling counter for policy hit counters

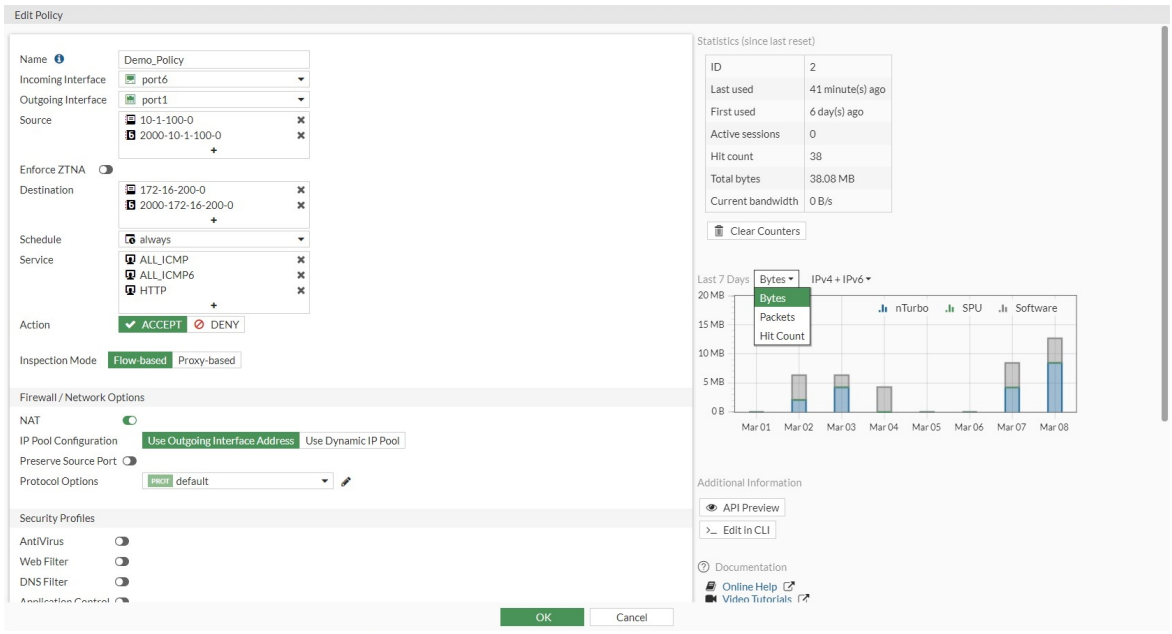
Instead of storing a single number for the hit count and byte count collected since the inception of each policy, seven numbers for the last seven days and an active counter for the current day are stored. The past seven-day hit count is displayed in the policy list and policy pages. A seven-day bar chart shows statistics on each policy page. This feature is currently supported in firewall and multicast policies, but not security policies.

To view the rolling counter information in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy* or *Policy & Objects* > *Multicast Policy*.
2. Select a policy and hover over the *Bytes*, *Packets*, or *Hit Count* values to view the tooltip with the corresponding traffic statistics and bar graph (this example uses firewall policies).



3. Click *Edit*. The policy traffic statistics appear in the right-hand side of the page.
4. Use the dropdowns to filter the bar graph data by counter (*Bytes*, *Packets*, or *Hit Count*) and policy type (*IPv4*, *IPv6*, or *IPv4 + IPv6*).



5. Optionally, click *Clear Counters* to delete the traffic statistics for the policy.
6. Click *OK*.

To view the rolling counter information in the CLI:

```
# diagnose firewall iprope show 100004 2
idx=2 pkts/bytes=14709/18777329 asic_pkts/asic_bytes=8087/10413737 nturbo_pkts/nturbo_
bytes=8087/10413737 flag=0x0 hit count:19 (4 7 0 1 1 3 3 0)
```

```
    first:2021-03-02 17:09:00 last:2021-03-08 17:23:40
 established session count:0
    first est:2021-03-02 17:11:20 last est:2021-03-08 17:23:40
```

```
# diagnose firewall iprope6 show 100004 2
idx=2 pkts/bytes=15698/19307164 asic_pkts/asic_bytes=7006/8578911 nturbo_pkts/nturbo_
bytes=7006/8578911 flag=0x0 hit count:19 (4 7 0 1 3 2 2 0)
    first:2021-03-02 17:10:32 last:2021-03-08 17:23:33
 established session count:0
    first est:2021-03-02 17:11:43 last est:2021-03-08 17:23:33
```

Cisco Security Group Tag as policy matching criteria

The FortiGate can read the Cisco Security Group Tag (SGT) in Ethernet frames, and use them as matching criteria in firewall policies. A policy can match based on the presence of an SGT, or the detection of a specific ID or IDs.

When a packet with a SGT passes through and a session is established, the `ext_header_type=0xc5:0xc5` flag is included in the session table.

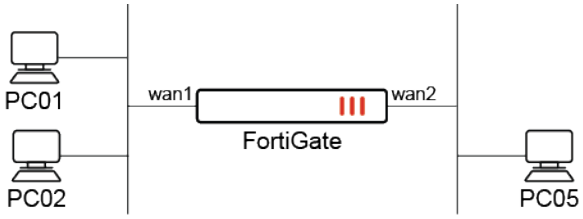
This feature is available in flow mode policies for virtual wire pair policies only.

To configure a firewall policy to detect SGTs in Ethernet frames:

```
config firewall policy
    edit 1
        set sgt-check {enable | disable}
        set sgt <ID numbers>
    next
end
```

Examples

In these examples, wan1 and wan2 are in a virtual wire pair. Firewall policies are created that pass traffic with SGTs with a specific ID number, any ID number, or either of two specific ID numbers.



To configure the virtual wire pair:

```
config system virtual-wire-pair
    edit "test-vwp-1"
        set member "wan1" "wan2"
        set wildcard-vlan enable
```

```
    next
end
```

To configure a firewall policy to match frames that have an SGT with ID 20 and allow them through:

```
config firewall policy
    edit 1
        set srcintf "wan1"
        set dstintf "wan2"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set sgt-check enable
        set sgt 20
    next
end
```

To configure a firewall policy to match frames that have an SGT with any ID:

```
config firewall policy
    edit 1
        set srcintf "wan1"
        set dstintf "wan2"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set sgt-check enable
    next
end
```

To configure a firewall policy to match frames that have the SGT with IDs 20 or 21:

```
config firewall policy
    edit 1
        set srcintf "wan1"
        set dstintf "wan2"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set sgt-check enable
        set sgt 20 21
    next
end
```

Processing only Ethernet frames with a Cisco Security Group Tag

In this example, an Ethernet frame is sent from PC01 with an SGT tag (ID 20), which can pass through to PC05 based on any of the firewall policies in the previous examples.

To verify the configuration:

```
# diagnose sys session list

session info: proto=6 proto_state=01 duration=10 expire=3593 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=log may_dirty br dst-vis f00
statistic(bytes/packets/allow_err): org=112/2/1 reply=60/1/1 tuples=2
tx speed(Bps/kbps): 10/0 rx speed(Bps/kbps): 5/0
orgin->sink: org pre->post, reply pre->post dev=13->10/10->13 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 10.1.1.11:36970->10.1.2.11:80(0.0.0.0:0)
hook=post dir=reply act=noop 10.1.2.11:80->10.1.1.11:36970(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
dst_mac=00:b0:e1:22:cf:e4
misc=0 policy_id=1 auth_info=0 chk_client_info=0 vd=1
serial=0000183c tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id=00000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
ext_header_type=0xc5:0xc5
total session 1
```

Processing Ethernet frames with a Cisco Security Group Tag and VLAN tag

The FortiGate has the ability to process Ethernet frames with both the Cisco Security Group Tag and VLAN tag.

In this example, PC02 is connected to a switch port configured for VLAN 2. An Ethernet frame is sent from PC02 with an SGT tag (ID 20) and VLAN ID (2), which can pass through to PC05 based on any of the firewall policies in the previous examples.

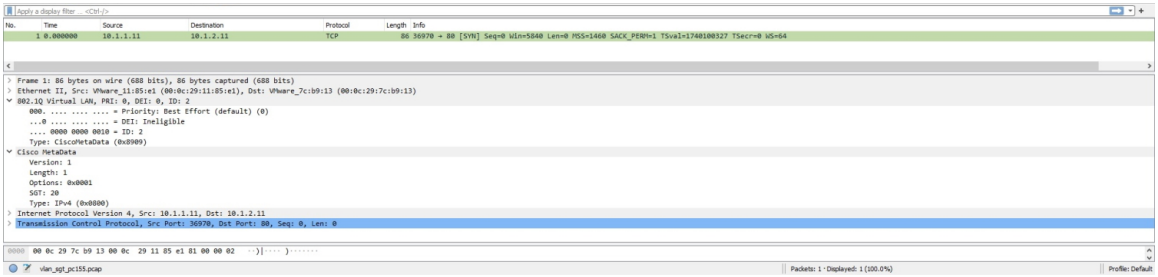
To verify the configuration:

1. Check the session list:

```
# diagnose sys session list

session info: proto=6 proto_state=01 duration=2007 expire=3482 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
```

```
state=may_dirty br
statistic(bytes/packets/allow_err): org=164/3/1 reply=120/2/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=8->7/7->8 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 10.1.1.11:36970->10.1.2.11:80(0.0.0.0:0)
hook=post dir=reply act=noop 10.1.2.11:80->10.1.1.11:36970(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=1 pol_uuid_idx=572 auth_info=0 chk_client_info=0 vd=0
serial=0432fb8f tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
vlanid=2
npu_state=0x4000001 no_offload
no_ofld_reason:  disabled-by-policy
           ext_header_type=0xc5:0xc5
```

2. Perform a packet capture on PC05 (Wireshark is used in this example) and check that the packet includes the VLAN ID and Cisco SGT fields.



Virtual patching on the local-in management interface

Virtual patching is a method of mitigating vulnerability exploits by using the FortiGate's IPS engine to block known vulnerabilities. Virtual patching can be applied to traffic destined to the FortiGate by applying the FMWP (Firmware Virtual Patch) database to the local-in interface using local-in policies. Attacks geared towards GUI and SSH management access, for example, can be mitigated using the FMWP database pushed from FortiGuard, thereby virtually patching these vulnerabilities.

When the `virtual-patch` option is enabled in a local-in policy, the IPS engine queries the FortiGuard API server using the WAD process to obtain a list of vulnerabilities targeting the FortiGate on a particular version. IPS enables vulnerability rules to scan local-in traffic on the specified interface. All matched local-in traffic is dropped accordingly. This will be applied to any new sessions after virtual patching is enabled. Reset your sessions accordingly if this needs to apply to existing long-lived sessions.

If a tagged session lacks vulnerability signatures for the FortiOS version, then the IPS engine bypasses the session. This optimizes performance by only scanning and dropping sessions that are exploiting a vulnerability.

To configure virtual patching:

```
config firewall local-in-policy
    edit <id>
        set action accept
        set virtual-patch {enable | disable}
```

```
    next
end
```

The FortiGate must have a valid FMWR (Firmware) license to install the FMWP database. The FMWP database can be viewed by running the `diagnose autoupdate versions` command.

```
# diagnose autoupdate versions
FMWP Definitions
---------
Version: 23.00084 signed
Contract Expiry Date: Wed Jan  1 2031
Last Updated using manual update on Wed Sep  6 15:19:11 2023
Last Update Attempt: Wed Sep  6 15:40:08 2023
Result: No Updates
```

Once `virtual-patch` is enabled, the WAD process will periodically query vulnerability items from the FortiGuard API server at "productapi.corp.fortinet.com" and forward it to IPS.

> ⚠ For Agentless VPN and ZTNA connections that terminate on the FortiGate from a client that is using a client certificate, enabling `virtual-patch` will cause the connection to fail. Do not enable `virtual-patch` if you are using either of these configurations.

Sample vulnerability item found on the FortiGuard API server

```
{"ID":918630,"product":"fortios","vendor":"fortinet","max_version":"7.2.5","min_
version":"7.2.0","severity":"high","vuln_type":"Format String","refs":
["https://www.fortiguard.com/psirt/FG-IR-23-137"],"description":"This indicates detection of a
Zero-Day vulnerability protected by a signature from Fortinet's FortiGuard Labs. This signature
should help mitigate the threat proactively both prior to, and after an official statement is
available from the vendor. Once an official advisory or statement is available from the vendor,
the signature name and its description will be updated to provide more details regarding this
vulnerability. Further details may also be made available in an advisory on FortiGuard Center
(http://www.fortiguard.com).","patch_sig_id":10004065,"patch_sig_ids":[],"detection_sig_
ids":null,"date_added":"2023-08-22T13:09:11","date_updated":"2023-08-22T13:09:11"}
```

FortiGuard can be queried from the FortiOS CLI for a list of vulnerability rules while specifying parameters for the vendor, version, product, and model by running the `diagnose wad dev-vuln query` command. For example, to query Fortinet Inc.'s FortiOS 7.2.5:

```
# diagnose debug enable
# diagnose wad dev-vuln query vendor=fortinet&version=7.2.5&product=fortios
Dev-Vuln Lookup result: success, cache: found, fgd: unknown, item: 0x7fb474e0b4a0
Vulnerability details:
info entry (1):
        'vendor' = fortinet
       'product' = fortios
         'model' = N/A
   'version.min' = 7.2.0
   'version.max' = 7.2.5
      'firmware' = N/A
         'build' = N/A
```

```
    'date_added' = 2023-08-22T13:09:11
  'date_updated' = 2023-08-22T13:09:11
       'sig_id' = 10004065
      'vuln_id' = 918630
     'severity' = 3
...
# diagnose debug disable
```

After receiving the vulnerability rules from the WAD process, the IPS engine marks them as virtual patch rules mapped to each CVE vulnerability signature. For example:

FortiOS.NodeJS.Proxy.Authentication.Bypass(CVE-2022-40684)

FortiOS.SSL.VPN.Web.Portal.Password.Improper.Authentication(CVE-2018-13382)

FortiOS.SSL.VPN.Web.Protoal.Pathname.Information.Disclosure(CVE-2018-13379)

To show the list of available FMWP signatures from the FMWP database:

```
# get rule fmwp status
rule-name: "FortiOS.Fclicense.Daemon.Format.String."
rule-id: 10004067
rev: 23.082
date: 1697644800
action: block
status: enable
log: disable
log-packet: disable
severity: 3.high
service: TCP, HTTP
location: server
os: Linux
application: Other
rate-count: 0
rate-duration: 0
rate-track: none
rate-mode: continuous
vuln_type: Format String
cve: 202329181
fos_comp: Web-GUI
....
```

The following are the diagnose commands:

```
# diagnose ips vpatch {fmwp-status | fmwp-enable-all  | fmwp-reset}
```

`fmwp-status`	Shows the current status of enabled FMWP signatures.
`fmwp-enable-all`	Enable all FMWP signatures in FMWP database.
`fmwp-reset`	Revert the results of `fmwp-enable-all`.

Example

In this example, virtual patching is enabled for the local-in policy and the following scenarios are described:

- FortiGate with an Agentless VPN vulnerability
- FortiGate with a web GUI vulnerability
- FortiGate with both an Agentless VPN and web GUI vulnerability

To enable virtual patching:

1. Enable virtual patching in the local-in policy:

```
config firewall local-in-policy
    edit 1
        set intf "port2"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set service "ALL"
        set schedule "always"
        set virtual-patch enable
    next
end
```

For enabling virtual patching in a local-in policy in the GUI, see Local-in policy on page 1886.

Because the IPS engine can currently only tag services related to Agentless VPN and web GUI signatures, all other protocols are scanned when `service` is set to `ALL`. However, you can bypass scanning of other protocols, such as SSH and FTP, by setting `service` to only `HTTPS`.

2. Observe the outcome of the following scenarios:
   - In this example, FortiOS has an Agentless VPN vulnerability. The IPS engine drops Agentless VPN traffic to the local-in interface on the FortiGate and bypasses web GUI traffic. Traffic for other services is scanned and passed to the interface.

     Following is a log of the Agentless VPN traffic that was dropped because of the vulnerability. Bypassed web GUI traffic did not generate any logs.

```
# diagnose ips vpatch fmwp-status
Enabled FMWP signatures: 3

  10002887 FortiOS.SSL-VPN.Heap.Buffer.Overflow.

1: date=2023-11-07 time=14:53:44 eventtime=1699325624346021995 tz="+1200"
logid="0419016384" type="utm" subtype="ips" eventtype="signature" level="alert" vd="root"
severity="critical" srcip=10.1.100.22 srccountry="Reserved" dstip=10.1.100.1
dstcountry="Reserved" srcintf="port2" srcintfrole="undefined" dstintf="root"
dstintfrole="undefined" sessionid=284 action="dropped" proto=6 service="HTTPS" policyid=1
attack="FortiOS.SSL-VPN.Heap.Buffer.Overflow." srcport=53250 dstport=11443
hostname="myfortigate.example" url="/error" httpmethod="POST" direction="outgoing"
```

```
attackid=10002887 ref="http://www.fortinet.com/ids/VID10002887" incidentserialno=99614721
msg="vPatch: FortiOS.SSL-VPN.Heap.Buffer.Overflow." crscore=50 craction=4096
crlevel="critical"
```

- In this example, FortiOS has a web GUI vulnerability. The IPS engine drops web GUI traffic to the local-in interface on the FortiGate and bypasses Agentless VPN traffic. Traffic for other services is scanned and passed to the interface.

  Following is a log of the web GUI traffic that was dropped because of the vulnerability. Bypassed Agentless VPN traffic did not generate any logs.

```
# diagnose ips vpatch fmwp-status
Enabled FMWP signatures: 2

  10002156 FortiOS.NodeJS.Proxy.Authentication.Bypass.
  10002890 FortiOS.HTTPD.Content-Length.Memory.Corruption.

1: date=2023-11-07 time=14:55:15 eventtime=1699325715311370215 tz="+1200"
logid="0419016384" type="utm" subtype="ips" eventtype="signature" level="alert" vd="root"
severity="critical" srcip=10.1.100.22 srccountry="Reserved" dstip=10.1.100.1
dstcountry="Reserved" srcintf="port2" srcintfrole="undefined" dstintf="root"
dstintfrole="undefined" sessionid=304 action="dropped" proto=6 service="HTTPS" policyid=1
attack="FortiOS.NodeJS.Proxy.Authentication.Bypass." srcport=53622 dstport=443
hostname="127.0.0.1:9980" url="/api/v2/cmdb/system/admin" agent="Node.js" httpmethod="GET"
direction="outgoing" attackid=10002156 ref="http://www.fortinet.com/ids/VID10002156"
incidentserialno=99614722 msg="vPatch: FortiOS.NodeJS.Proxy.Authentication.Bypass."
crscore=50 craction=4096 crlevel="critical"
```

- In this example, FortiOS has an Agentless VPN and a web GUI vulnerability. The IPS engine drops both Agentless VPN and web GUI traffic to the local-in interface on the FortiGate. Traffic for other services is scanned and passed to the interface.

  Following is a log of the Agentless VPN and web GUI traffic that was dropped because of the vulnerability.

```
# diagnose ips vpatch fmwp-status
Enabled FMWP signatures: 3

  10002156 FortiOS.NodeJS.Proxy.Authentication.Bypass.
  10002887 FortiOS.SSL-VPN.Heap.Buffer.Overflow.
  10002890 FortiOS.HTTPD.Content-Length.Memory.Corruption.

1: date=2023-11-07 time=06:42:44 eventtime=1699296164649894963 tz="+1200"
logid="0419016384" type="utm" subtype="ips" eventtype="signature" level="alert" vd="root"
severity="critical" srcip=10.1.100.22 srccountry="Reserved" dstip=10.1.100.1
dstcountry="Reserved" srcintf="port2" srcintfrole="undefined" dstintf="root"
dstintfrole="undefined" sessionid=1094 action="dropped" proto=6 service="HTTPS" policyid=1
attack="FortiOS.SSL-VPN.Heap.Buffer.Overflow." srcport=44164 dstport=10443
hostname="myfortigate.example" url="/error" httpmethod="POST" direction="outgoing"
attackid=10002887 ref="http://www.fortinet.com/ids/VID10002887" incidentserialno=116392250
msg="vPatch: FortiOS.SSL-VPN.Heap.Buffer.Overflow." crscore=50 craction=4096
crlevel="critical"

2: date=2023-11-07 time=06:42:09 eventtime=1699296129458704870 tz="+1200"
```

```
logid="0419016384" type="utm" subtype="ips" eventtype="signature" level="alert" vd="root"
severity="critical" srcip=10.1.100.22 srccountry="Reserved" dstip=10.1.100.1
dstcountry="Reserved" srcintf="port2" srcintfrole="undefined" dstintf="root"
dstintfrole="undefined" sessionid=1066 action="dropped" proto=6 service="HTTPS" policyid=1
attack="FortiOS.NodeJS.Proxy.Authentication.Bypass." srcport=42352 dstport=443
hostname="127.0.0.1:9980" url="/api/v2/cmdb/system/admin" agent="Node.js" httpmethod="GET"
direction="outgoing" attackid=10002156 ref="http://www.fortinet.com/ids/VID10002156"
incidentserialno=116392236 msg="vPatch: FortiOS.NodeJS.Proxy.Authentication.Bypass."
crscore=50 craction=4096 crlevel="critical"
```

Air-gapped and closed networks

In air-gapped and closed networks, it is possible to bypass direct queries to the FortiGuard server for vulnerability items by using a FortiManager as an override server. This defers the FortiGuard queries to the FortiManager, and FortiGate only talks to the FortiManager inside the closed network.

For information about how this is configured, see Using FortiManager as an override server for IoT query services on page 3914.

Configuring PCP port mapping with SNAT and DNAT

FortiOS supports the Port Control Protocol (PCP) by allowing the FortiGate to act as a PCP server, and dynamically manage network addresses and port translations for PCP clients.

IPv6 PCP for DNAT46 is also supported, enabling inbound PCP MAP requests in NAT64 deployments and allowing FortiOS to dynamically manage IPv6-to-IPv4 mappings through PCP, improving compatibility with CLAT-based CPEs.

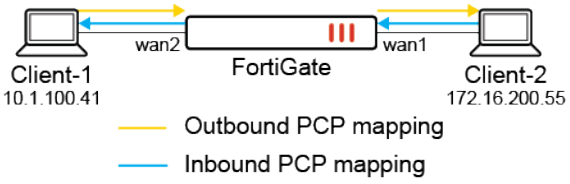
The PCP server must be enabled with a pool (`config system pcp-server`). In the firewall policy, enable either `pcp-outbound` or `pcp-inbound` mode and assign the pool.

```
config system pcp-server
    set status {enable | disable}
    config pools
        edit <name>
            set client-subnet <ip_address/subnet>
            set client6-prefix <IPv6 client prefixes>
            set ext-intf <string>
            set extip ip>[-<ip>]
            set extport <port>[-<port>]
            set minimal-lifetime <integer>
            set maximal-lifetime <integer>
            set client-mapping-limit <integer>
            set mapping-filter-limit <integer>
            set allow-opcode {map peer announce}
            set third-party {allow | disallow}
            set multicast-announcement {enable | disable}
            set announcement-count <integer>
            set intl-intf <string>
            set recycle-delay <integer>
            set nat46 {enable | disable}
```

```
        next
    end
end
```

Command	Description
`client-subnet <ip_ address/subnet>`	Enter the IP address with subnet from which PCP requests are accepted.
`client6-prefix <IPv6 client prefixes>`	Prefixes from which PCP requests are accepted.
`ext-intf <string>`	Enter the external interface name.
`extip <ip>[-<ip>]`	Enter the IP address or address range on the external interface to map to an address on the internal network.
`extport <port>[-<port>]`	Enter the incoming port number or port range to map to a port number on the internal network.
`minimal-lifetime <integer>`	Set the minimal lifetime of a PCP mapping, in seconds (60 - 300, default = 120).
`maximal-lifetime <integer>`	Set the maximal lifetime of a PCP mapping, in seconds (3600 - 604800, default = 86400).
`client-mapping-limit <integer>`	Mapping limit per client (0 - 65535, default = 0, 0 = unlimited).
`mapping-filter-limit <integer>`	Filter limit per mapping (0 - 5, default = 1).
`allow-opcode {map peer announce}`	Set the allowed PCP OpCode: • `map`: allow MAP OpCode • `peer`: allow PEER OpCode • `announce`: allow ANNOUNCE OpCode
`third-party {allow | disallow}`	Allow/disallow the third-party option.
`multicast-announcement {enable | disable}`	Enable/disable multicast announcements.
`announcement-count <integer>`	Set the number of multicast announcements (3 - 10, default = 3).
`intl-intf <string>`	Enter the internal interface name.
`recycle-delay <integer>`	Set the minimum delay the PCP server will wait before recycling mappings that have expired, in seconds (0 - 3600, default = 0).
`nat46 {enable | disable}`	Enable/disable NAT46.

The following topology is used in examples one and two to demonstrate PCP mapping with SNAT and DNAT.



- Example 1: PCP mapping with SNAT on page 2016
- Example 2: PCP mapping with DNAT on page 2018
- Example 3: PCP IPv6-to-IPv4 mapping with NAT64/DNAT46 New on page 2020

Example 1: PCP mapping with SNAT

This example demonstrates how PCP mapping works with SNAT. In the FortiGate PCP server settings, the pcp-pool1 pool is applied in the firewall policy with `pcp-outbound` mode. A PCP request is sent from Client-1 to the FortiGate to create PCP outbound mapping. When traffic is sent from Client-1 to Client-2, SNAT is performed by the PCP outbound mapping.

To configure the FortiGate as a PCP server:

1. Configure the PCP server settings:

```
config system pcp-server
    set status enable
    config pools
        edit "pcp-pool1"
            set client-subnet "10.1.100.41/32"
            set ext-intf "wan1"
            set extip 172.16.200.231
            set extport 50000-51000
            set intl-intf "wan2"
        next
    end
end
```

2. Configure the firewall policy:

```
config firewall policy
    edit 999
        set name "Outbound-pcp-policy999"
        set srcintf "wan2"
        set dstintf "wan1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set auto-asic-offload disable
        set nat enable
        set pcp-outbound enable
        set pcp-poolname "pcp-pool1"
    next
end
```

To verify the configuration:

1. Generate a PCP peer request from Client-1 (10.1.100.41) to the FortiGate.
2. Verify the client's PCP request to the PCP server. In this example, an PCP client was installed on Ubuntu:

```
root@pc41:~# pcp -i 10.1.100.41:41111 -p 172.16.200.55:80 -s 10.1.100.8
```

3. On the FortiGate, verify the PCP outbound mappings list:

```
# diagnose firewall pcp-mapping list outbound
PCP outbound mappings (vdom=root):
pool:1 nonce:04307eb4037e0448317dc8b7 protocol:6 duration:8 lifetime:900 expiry:893
intl:10.1.100.41:41111 ext:172.16.200.231:50000 remote:172.16.200.55:80
```

4. Send HTTP traffic that passes through the FortiGate and access Client-2 (172.16.200.55:80) from Client-1.
5. On the FortiGate, verify the session list. The source IP address of Client-1 is translated to 172.16.200.231:50000, which follows the PCP outbound mapping:

```
# diagnose sys session list
session info: proto=6 proto_state=01 duration=8 expire=3599 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty f00 pcp_outbound
statistic(bytes/packets/allow_err): org=1812/33/1 reply=124168/92/1 tuples=2
tx speed(Bps/kbps): 204/1 rx speed(Bps/kbps): 13998/111
orgin->sink: org pre->post, reply pre->post dev=8->7/7->8 gwy=172.16.200.55/10.1.100.41
hook=post dir=org act=snat 10.1.100.41:41111->172.16.200.55:80(172.16.200.231:50000)
hook=pre dir=reply act=dnat 172.16.200.55:80->172.16.200.231:50000(10.1.100.41:41111)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=999 pol_uuid_idx=677 auth_info=0 chk_client_info=0 vd=0
serial=0000b4f8 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000001 no_offload
no_ofld_reason:  disabled-by-policy
total session 1
```

6. Send HTTP traffic that passes through the FortiGate and access another server from Client-1.
7. On the FortiGate, verify the session list. This time, the source IP address of Client-1 is not translated to 172.16.200.231:50000, since the traffic does not match the existing PCP outbound mapping:

```
# diagnose sys session list
session info: proto=6 proto_state=01 duration=6 expire=3596 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty f00
statistic(bytes/packets/allow_err): org=1449/26/1 reply=98808/72/1 tuples=2
```

```
tx speed(Bps/kbps): 215/1 rx speed(Bps/kbps): 14703/117
orgin->sink: org pre->post, reply pre->post dev=8->7/7->8 gwy=172.16.200.155/10.1.100.41
hook=post dir=org act=snat 10.1.100.41:41111->172.16.200.155:80(172.16.200.8:41111)
hook=pre dir=reply act=dnat 172.16.200.155:80->172.16.200.8:41111(10.1.100.41:41111)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=999 pol_uuid_idx=677 auth_info=0 chk_client_info=0 vd=0
serial=0000b596 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000001 no_offload
no_ofld_reason:  disabled-by-policy
total session 1
```

Example 2: PCP mapping with DNAT

This example demonstrates how PCP mapping works with DNAT. In the FortiGate PCP server settings, the pcp-pool1 pool is applied in the firewall policy with `pcp-inbound` mode. A PCP request is sent from Client-1 to the FortiGate to create PCP inbound mapping. When traffic is sent from Client-2 to access the external IP of Client-1 (172.16.200.231:50000), traffic passes by due to the PCP inbound mapping.

To configure the FortiGate as a PCP server:

1. Configure the PCP server settings:

```
config system pcp-server
    set status enable
    config pools
        edit "pcp-pool1"
            set client-subnet "10.1.100.41/32"
            set ext-intf "wan1"
            set extip 172.16.200.231
            set extport 50000-51000
            set intl-intf "wan2"
        next
    end
end
```

2. Configure the firewall policy:

```
config firewall policy
    edit 998
        set name "Inbound-pcp-policy998"
        set srcintf "wan1"
        set dstintf "wan2"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
```

```
        set auto-asic-offload disable
        set nat enable
        set pcp-inbound enable
        set pcp-poolname "pcp-pool1"
    next
end
```

To verify the configuration:

1. Generate a PCP peer request from Client-1 (10.1.100.41) to the FortiGate.
2. Verify the client's PCP request to the PCP server. In this example, an PCP client was installed on Ubuntu:

```
root@pc41:~# pcp -i 10.1.100.41:80 -s 10.1.100.8
```

3. On the FortiGate, verify the PCP inbound mappings list:

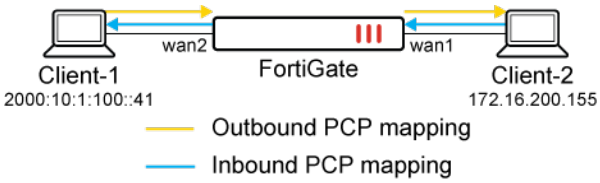
```
# diagnose firewall pcp-mapping list inbound
PCP inbound mappings (vdom=root):
pool:1 nonce:35e2ff035b959f7a4e669791 protocol:6 duration:3 lifetime:900 expiry:900
intl:10.1.100.41:80 ext:172.16.200.231:50000
```

4. From Client-2 (172.16.200.55:80), send traffic that passes through the FortiGate and access the external IP of Client-1 (172.16.200.231:50000).
5. On the FortiGate, run a sniffer trace. The traffic is allowed through policy 998, and the destination IP:port is translated from 172.16.200.231:50000 to 10.1.100.41:80, which follows the PCP inbound mapping:

```
# diagnose sniffer packet any 'tcp and port 50000 or port 80' 4
interfaces=[any]
filters=[tcp and port 50000 or port 80]
2.959915 wan1 in 172.16.200.55.43284 -> 172.16.200.231.50000: syn 3480016601
2.960051 wan2 out 10.1.100.8.43284 -> 10.1.100.41.80: syn 3480016601
2.960390 wan2 in 10.1.100.41.80 -> 10.1.100.8.43284: syn 2813145613 ack 3480016602
2.960447 wan1 out 172.16.200.231.50000 -> 172.16.200.55.43284: syn 2813145613 ack 3480016602
2.960644 wan1 in 172.16.200.55.43284 -> 172.16.200.231.50000: ack 2813145614
2.960664 wan2 out 10.1.100.8.43284 -> 10.1.100.41.80: ack 2813145614
2.961194 wan1 in 172.16.200.55.43284 -> 172.16.200.231.50000: psh 3480016602 ack 2813145614
2.961209 wan2 out 10.1.100.8.43284 -> 10.1.100.41.80: psh 3480016602 ack 2813145614
2.961516 wan2 in 10.1.100.41.80 -> 10.1.100.8.43284: ack 3480016686
2.961533 wan1 out 172.16.200.231.50000 -> 172.16.200.55.43284: ack 3480016686
2.993623 wan2 in 10.1.100.41.80 -> 10.1.100.8.43284: psh 2813145614 ack 3480016686
2.993637 wan1 out 172.16.200.231.50000 -> 172.16.200.55.43284: psh 2813145614 ack 3480016686
2.993947 wan1 in 172.16.200.55.43284 -> 172.16.200.231.50000: ack 2813145875
2.993962 wan2 out 10.1.100.8.43284 -> 10.1.100.41.80: ack 2813145875
2.995677 wan1 in 172.16.200.55.43284 -> 172.16.200.231.50000: fin 3480016686 ack 2813145875
2.995691 wan2 out 10.1.100.8.43284 -> 10.1.100.41.80: fin 3480016686 ack 2813145875
2.996059 wan2 in 10.1.100.41.80 -> 10.1.100.8.43284: fin 2813145875 ack 3480016687
2.996075 wan1 out 172.16.200.231.50000 -> 172.16.200.55.43284: fin 2813145875 ack 3480016687
2.996230 wan1 in 172.16.200.55.43284 -> 172.16.200.231.50000: ack 2813145876
2.996245 wan2 out 10.1.100.8.43284 -> 10.1.100.41.80: ack 2813145876
```

Only traffic matching the PCP inbound mapping will be forwarded by policy 998. Any other traffic is dropped.

Example 3: PCP IPv6-to-IPv4 mapping with NAT64/DNAT46 NEW



This example demonstrates how PCP IPv6-to-IPv4 mappings work with NAT64/DNAT46.

Client-1 has an IPv6 IP address, and Client-2 has an IPv4 address. On the FortiGate, a PCP server is configured with a PCP pool that has NAT46 enabled. The PCP pool is applied to a NAT46 firewall policy with pcp-inbound enabled and the pool specified. Client-1 sends an IPv6 PCP request to FortiGate to request a PCP inbound mapping. FortiGate uses the inbound PCP mapping to DNAT translate the IPv4 IP address from Client-2 to the IPv6 IP address for Client-1.

To configure the FortiGate as a PCP server:

1. Configure the PCP server settings:

   The pool is named `35366_pool46_inbound`. NAT46 is enabled and an IPv6 prefix is defined. One IP/host subnet is used for both client and external IP.

```
config system pcp-server
   set status enable
   config pool
        edit "35366_pool46_inbound"
            set client-subnet "10.1.100.41/32"
            set client6-prefix "2000:10:1:100::41/128"
            set ext-intf "port1"
            set extip 172.16.200.121
            set extport 50000-51000
            set intl-intf "port9"
            set nat46 enable
        next
   end
end
```

2. Configure the firewall policy:

   In the firewall policy, `nat46` is enabled, `pcp-inbound` is enabled, and the PCP pool name (`35366_pool46_inbound`) is specified.

   The external interface (`ext-intf`) is port1, and the internal interface (`intl-intf`) is port9.

```
config firewall policy
    edit 46
        set name "pcp-46-inbound"
        set uuid de6c3632-95a6-51f0-0124-aa2705db9569
        set srcintf "port1"
        set dstintf "port9"
        set action accept
        set nat46 enable
        set custom-tags "test_tag_pcp_test"
```

```
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set auto-asic-offload disable
        set pcp-inbound enable
        set pcp-poolname "35366_pool46_inbound"
        set ippool enable
        set poolname6 "pcp_ippool6_1"
    next
end
```

To verify the configuration:

1. Generate a PCP map request from Client-1 (IPv6 2000:10:1:100::41) to the FortiGate:

```
root@pc41:~# pcp -i [2000:10:1:100::41]:80 -s [2000:10:1:100::5]
```

2. On the FortiGate, verify the PCP inbound mappings list:

```
# diagnose firewall pcp-mapping list inbound
PCP inbound mappings (vdom=root):
pool:2 nonce:19c410963f68158746b735ba protocol:6 duration:809 lifetime:900 expiry:90
intl:2000:10:1:100::41:80 ext:172.16.200.121:50000
```

3. From Client-2 (IPv4 172.16.200.155), send traffic to pass through FortiGate and access the external interface (`extip`) of Client-1 (IPv4 172.16.200.121:50000).

   The traffic passes by the policy (policy id `46`), and the destination IP:port is DNAT translated from `172.16.200.231:50000` to `2000:10:1:100::41:80`, which follows the inbound pcp-mapping.

   Policy 46 only forwards traffic that matches the inbound PCP mapping (`pcp-mapping`). Other traffic is dropped.

```
# diagnose sniffer packet any 'tcp and port 80 or port 50000' 4
interfaces=[any]
filters=[tcp and port 80 or port 50000]
4.645467 port1 in 172.16.200.155.56926 -> 172.16.200.121.50000: syn 3388939279
4.645517 naf.root out 172.16.200.155.56926 -> 172.16.200.121.50000: syn 3388939279
4.645521 naf.root in 2000:10:1:100::121.56926 -> 2000:10:1:100::41.80: syn 3388939279
4.645557 port9 out 2000:10:1:100::121.56926 -> 2000:10:1:100::41.80: syn 3388939279
4.645866 port9 in 2000:10:1:100::41.80 -> 2000:10:1:100::121.56926: syn 3670124269 ack
3388939280  [flowlabel 0x60fc6]
4.645882 naf.root out 2000:10:1:100::41.80 -> 2000:10:1:100::121.56926: syn 3670124269 ack
3388939280  [flowlabel 0x60fc6]
4.645884 naf.root in 172.16.200.121.50000 -> 172.16.200.155.56926: syn 3670124269 ack
3388939280
4.645906 port1 out 172.16.200.121.50000 -> 172.16.200.155.56926: syn 3670124269 ack 3388939280
4.646006 port1 in 172.16.200.155.56926 -> 172.16.200.121.50000: ack 3670124270
4.646013 naf.root out 172.16.200.155.56926 -> 172.16.200.121.50000: ack 3670124270
```

```
4.646014 naf.root in 2000:10:1:100::121.56926 -> 2000:10:1:100::41.80: ack 3670124270
4.646020 port9 out 2000:10:1:100::121.56926 -> 2000:10:1:100::41.80: ack 3670124270
4.646138 port1 in 172.16.200.155.56926 -> 172.16.200.121.50000: psh 3388939280 ack 3670124270
4.646141 naf.root out 172.16.200.155.56926 -> 172.16.200.121.50000: psh 3388939280 ack
3670124270
4.646142 naf.root in 2000:10:1:100::121.56926 -> 2000:10:1:100::41.80: psh 3388939280 ack
3670124270
4.646144 port9 out 2000:10:1:100::121.56926 -> 2000:10:1:100::41.80: psh 3388939280 ack
3670124270
4.646227 port9 in 2000:10:1:100::41.80 -> 2000:10:1:100::121.56926: ack 3388939364  [flowlabel
0x60fc6]
4.646232 naf.root out 2000:10:1:100::41.80 -> 2000:10:1:100::121.56926: ack 3388939364
[flowlabel 0x60fc6]
4.646233 naf.root in 172.16.200.121.50000 -> 172.16.200.155.56926: ack 3388939364
4.646236 port1 out 172.16.200.121.50000 -> 172.16.200.155.56926: ack 3388939364
4.646587 port9 in 2000:10:1:100::41.80 -> 2000:10:1:100::121.56926: psh 3670124270 ack
3388939364  [flowlabel 0x60fc6]
4.646590 naf.root out 2000:10:1:100::41.80 -> 2000:10:1:100::121.56926: psh 3670124270 ack
3388939364  [flowlabel 0x60fc6]
4.646591 naf.root in 172.16.200.121.50000 -> 172.16.200.155.56926: psh 3670124270 ack
3388939364
4.646593 port1 out 172.16.200.121.50000 -> 172.16.200.155.56926: psh 3670124270 ack 3388939364
4.646668 port1 in 172.16.200.155.56926 -> 172.16.200.121.50000: ack 3670124531
4.646671 naf.root out 172.16.200.155.56926 -> 172.16.200.121.50000: ack 3670124531
4.646672 naf.root in 2000:10:1:100::121.56926 -> 2000:10:1:100::41.80: ack 3670124531
4.646675 port9 out 2000:10:1:100::121.56926 -> 2000:10:1:100::41.80: ack 3670124531
4.648833 port1 in 172.16.200.155.56926 -> 172.16.200.121.50000: fin 3388939364 ack 3670124531
4.648836 naf.root out 172.16.200.155.56926 -> 172.16.200.121.50000: fin 3388939364 ack
3670124531
4.648837 naf.root in 2000:10:1:100::121.56926 -> 2000:10:1:100::41.80: fin 3388939364 ack
3670124531
4.648839 port9 out 2000:10:1:100::121.56926 -> 2000:10:1:100::41.80: fin 3388939364 ack
3670124531
4.649053 port9 in 2000:10:1:100::41.80 -> 2000:10:1:100::121.56926: fin 3670124531 ack
3388939365  [flowlabel 0x60fc6]
4.649057 naf.root out 2000:10:1:100::41.80 -> 2000:10:1:100::121.56926: fin 3670124531 ack
3388939365  [flowlabel 0x60fc6]
4.649058 naf.root in 172.16.200.121.50000 -> 172.16.200.155.56926: fin 3670124531 ack
3388939365
4.649061 port1 out 172.16.200.121.50000 -> 172.16.200.155.56926: fin 3670124531 ack 3388939365
4.649138 port1 in 172.16.200.155.56926 -> 172.16.200.121.50000: ack 3670124532
4.649142 naf.root out 172.16.200.155.56926 -> 172.16.200.121.50000: ack 3670124532
4.649143 naf.root in 2000:10:1:100::121.56926 -> 2000:10:1:100::41.80: ack 3670124532
4.649146 port9 out 2000:10:1:100::121.56926 -> 2000:10:1:100::41.80: ack 3670124532
^C

40 packets received by filter
0 packets dropped by kernel
```

Refreshing active sessions for specific protocols and port ranges per VDOM in a specified direction

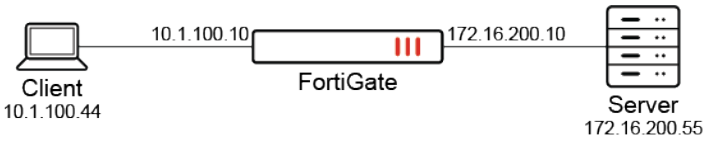
Active sessions can be refreshed for specific protocols and port ranges per VDOM in a specified direction. This option can help prevent potential denial of service (DoS) attacks by controlling the direction of traffic that refreshes existing sessions.

```
config system session-ttl
    config port
        edit <id>
            set protocol <integer>
            set timeout <timeout_value>
            set refresh-direction {both | outgoing | incoming}
        next
    end
end
```

Setting the `refresh-direction` to `outgoing` will use the original direction, while `incoming` will use the reply direction. To refresh in both directions, select `both`.

Example

In this example, active sessions for UDP port 5001 will be refreshed in the incoming direction.



To refresh active sessions for UDP port 5001 in the incoming direction:

1. Configure the global session TTL timer:

```
config system session-ttl
    set default 3600
    config port
        edit 5001
            set protocol 17
            set timeout 5001
            set refresh-direction incoming
            set start-port 5001
            set end-port 5001
        next
    end
end
```

2. Send UDP 5001 traffic from the client to the server.
3. Verify the session table:

```
# diagnose sys session list
session info: proto=17 proto_state=00 duration=77 expire=4923 timeout=5001 refresh_dir=reply
flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=log may_dirty f00
statistic(bytes/packets/allow_err): org=58/2/1 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=18->17/17->18 gwy=172.16.200.55/0.0.0.0
hook=post dir=org act=snat 10.1.100.41:2041->172.16.200.55:5001(172.16.200.10:62458)
hook=pre dir=reply act=dnat 172.16.200.55:5001->172.16.200.10:62458(10.1.100.41:2041)
src_mac=00:0c:29:b6:e8:be  dst_mac=00:0c:29:92:89:96
misc=0 policy_id=99 pol_uuid_idx=1501 auth_info=0 chk_client_info=0 vd=0
serial=00005071 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
total session: 1
```

The timeout and refresh for the reply direction are attached to the session.

4. Send UDP 5001 traffic again from the client to the server.
5. Verify the diagnostics.
   a. Run the sniffer trace:

```
# diagnose sniffer packet any 'udp and port 5001' 4
interfaces=[any]
filters=[udp and port 5001]
3.387747 wan2 in 10.1.100.41.2041 -> 172.16.200.55.5001: udp 1
3.387757 wan1 out 172.16.200.10.62458 -> 172.16.200.55.5001: udp 1
^C
2 packets received by filter
0 packets dropped by kernel
```

   b. Verify the session table:

```
# diagnose sys session list
session info: proto=17 proto_state=00 duration=119 expire=4881 timeout=5001 refresh_
dir=reply flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=log may_dirty f00
statistic(bytes/packets/allow_err): org=116/4/1 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 1/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=18->17/17->18 gwy=172.16.200.55/0.0.0.0
hook=post dir=org act=snat 10.1.100.41:2041->172.16.200.55:5001(172.16.200.10:62458)
hook=pre dir=reply act=dnat 172.16.200.55:5001->172.16.200.10:62458(10.1.100.41:2041)
src_mac=00:0c:29:b6:e8:be  dst_mac=00:0c:29:92:89:96
```

```
misc=0 policy_id=99 pol_uuid_idx=1501 auth_info=0 chk_client_info=0 vd=0
serial=00005071 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
total session: 1
```

As the traffic flows from the client to the server (outgoing), the expiration timer continues to count down and is not refreshed.

6. Send reverse UDP 5001 traffic from the server to the client.
7. Verify the diagnostics again.
   a. Run the sniffer trace:

```
# diagnose sniffer packet any 'udp and port 62458 or port 2041' 4
interfaces=[any]
filters=[udp and port 62458 or port 2041]
3.237328 wan1 in 172.16.200.55.5001 -> 172.16.200.10.62458: udp 1
3.237339 wan2 out 172.16.200.55.5001 -> 10.1.100.41.2041: udp 1
^C
2 packets received by filter
0 packets dropped by kernel
```

   b. Verify the session table:

```
# diagnose sys session list
session info: proto=17 proto_state=01 duration=1710 expire=4995 timeout=5001 refresh_
dir=reply flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=log may_dirty f00
statistic(bytes/packets/allow_err): org=116/4/1 reply=116/4/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=18->17/17->18
gwy=172.16.200.55/10.1.100.41
hook=post dir=org act=snat 10.1.100.41:2041->172.16.200.55:5001(172.16.200.10:62458)
hook=pre dir=reply act=dnat 172.16.200.55:5001->172.16.200.10:62458(10.1.100.41:2041)
src_mac=00:0c:29:b6:e8:be  dst_mac=00:0c:29:92:89:96
misc=0 policy_id=99 pol_uuid_idx=1501 auth_info=0 chk_client_info=0 vd=0
serial=00005071 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
total session: 1
```

As the traffic flows from the server to the client (incoming), the expiration timer is refreshed.

Per-policy disclaimer messages

FortiOS supports a customizable captive portal to direct users to install or enable required software.

Per-policy custom disclaimers in each VDOM are supported. For example, you may want to configure three firewall policies, each of which matches traffic from endpoints with different FortiClient statuses:

Endpoint status	FortiOS behavior
Endpoint does not have FortiClient installed.	Traffic matches a firewall policy that displays an in-browser warning to install FortiClient from the provided link.
Endpoint has FortiClient installed, registered to EMS, and connected to the FortiGate.	Traffic matches a dynamic firewall policy which allows the endpoint to reach its destination via this policy.
Endpoint is deregistered from EMS and disconnected from the FortiGate.	Traffic matches another dynamic firewall policy that displays warning to register FortiClient to EMS.

The replacement message groups and policy disclaimer settings must be enabled.

To enable per-policy disclaimer messages in the GUI:

1. Go to *System > Feature Visibility*.
2. Enable *Replacement Message Groups* and *Policy Disclaimer*.
3. Click *Apply*.

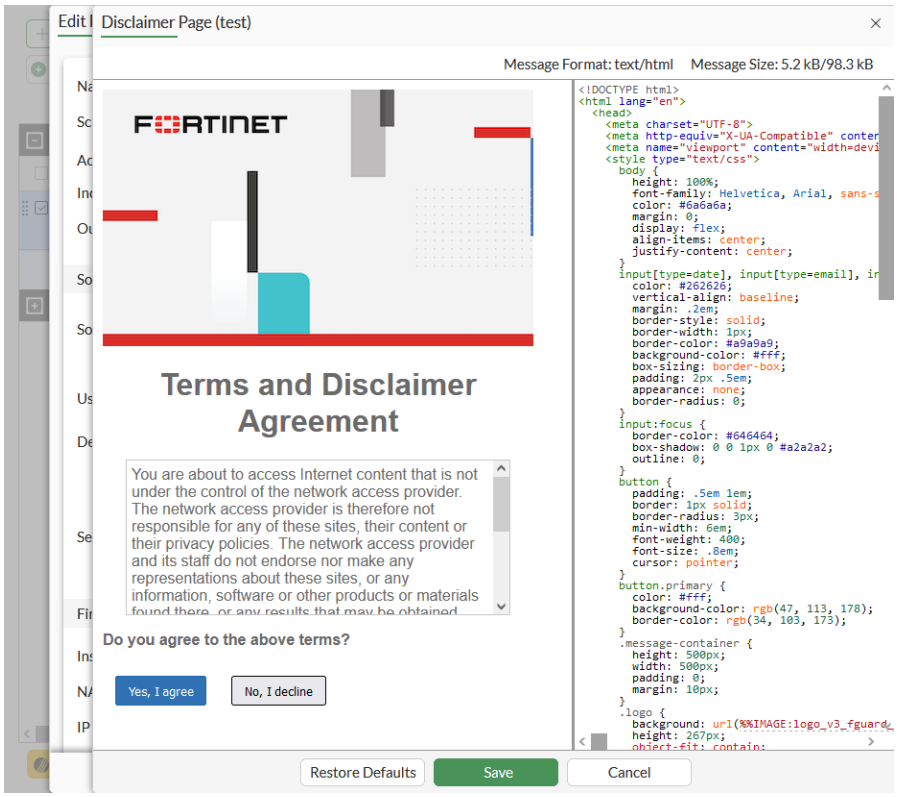
To enable per-policy disclaimer messages in the CLI:

```
config system global
    set gui-replacement-message-groups enable
end
```

```
config system settings
    set gui-policy-disclaimer enable
end
```

To configure per-policy disclaimers in the GUI:

1. Ensure the per-policy disclaimer messages option is enabled.
2. Go to *Policy & Objects > Firewall Policy*.
3. Edit the policy that applies when an endpoint does not have FortiClient installed.
4. Under *Disclaimer Options*, enable *Display Disclaimer* and *Customize Messages*.
5. Add a replacement message group:
   a. Select an existing replacement message group from the dropdown and click *Edit Disclaimer Message*.
   b. Click *Create*, enter a name, and click *OK*. Select the replacement message group and click *Edit Disclaimer Message*.



6. Edit the message to warn users to install FortiClient, and provide the FortiClient download link.
7. Click *Save*.
8. Repeat the above steps for each policy that requires a custom disclaimer message.

To configure per-policy disclaimers in the CLI:

```
config firewall policy
    edit 1
        set name "111"
        set srcintf "port12"
        set dstintf "port11"
        set srcaddr "all"
        set dstaddr "pc155_address"
        set action accept
        set schedule "always"
        set service "ALL"
        set wsso disable
        set groups "ems_03_group"
        set disclaimer enable
        set replacemsg-override-group "test"
        set nat enable
    next
    edit 4
        set name "44"
        set srcintf "port12"
        set dstintf "port11"
```

```
        set srcaddr "all"
        set dstaddr "pc5-address"
        set action accept
        set schedule "always"
        set service "ALL"
        set wsso disable
        set groups "ems_03_group"
        set disclaimer enable
        set replacemsg-override-group "test2"
        set nat enable
    next
    edit 6
        set name "66"
        set srcintf "port12"
        set dstintf "port11"
        set srcaddr "all"
        set dstaddr "all"
        set status disable
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set fsso disable
        set block-notification enable
        set replacemsg-override-group "endpoint-override"
    next
end
```

Full cone NAT for fixed port range IP pools

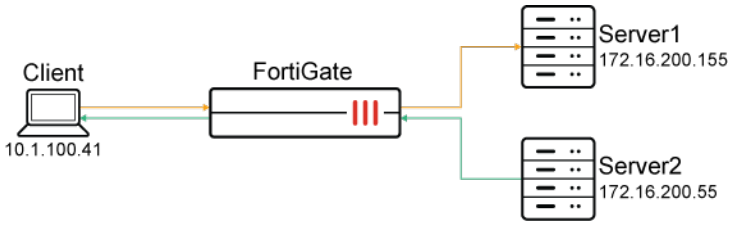
Full cone Network Address Translation (NAT) support is added for Fixed Port Range (FPR) IP pools. It allows all external hosts to send packets to internal hosts through a mapped external IP address and port, enhancing connectivity and communication efficiency. Full cone NAT is also known as Endpoint Independent Filtering (EIF).

To enable full cone NAT, enable the `permit-any-host` command when configuring the FPR IP pool.

```
config firewall ippool
    edit <name>
        set type fixed-port-range
        set permit-any-host {enable | disable}
    next
end
```

Only UDP is supported for full cone NAT.

To support HA session synchronization for full cone NAT, see FGCP and FGSP synchronizes full-cone expectation sessions when session-pickup-expectation enabled on page 4125.



In this example, a NAT44 FPR IP pool with `permit-any-host` enabled is created and applied to a policy. The packet is sent from Client1 to Server1 that hits the policy. The session list is checked for the NAT IP address and port, 172.16.201.3 and 1033. The expectation session list is checked to see that the session will be used to allow access to the NAT IP address and port from any other external host, such as Server2. The packet sent from Server2 to the NAT IP address and port is forwarded to Client1.

To configure and check the full cone IP pool:

1. Create an IP pool with full cone NAT:

```
config firewall ippool
    edit "test-new-fpr-ippool-1"
        set type fixed-port-range
        set startip 172.16.201.3
        set endip 172.16.201.4
        set startport 1024
        set endport 1087
        set source-startip 10.1.100.41
        set source-endip 10.1.100.42
        set port-per-user 64
        set permit-any-host enable
        set comments "test"
    next
end
```

2. Use the IP pool with full cone NAT in a policy:

```
config firewall policy
    edit 7
        set name "policy-fpr-ippool"
        set srcintf "port2"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set auto-asic-offload disable
        set nat enable
        set ippool enable
        set poolname "test-new-fpr-ippool-1"
    next
end
```

3. Check the session list:

```
# diagnose sys session list
session info: proto=17 proto_state=00 duration=65 expire=175 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=5
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty f00
statistic(bytes/packets/allow_err): org=58/2/1 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=8->7/7->8 gwy=0.0.0.0/0.0.0.0
hook=post dir=org act=snat 10.1.100.41:20041->172.16.200.155:2156(172.16.201.3:1033)
hook=pre dir=reply act=dnat 172.16.200.155:2156->172.16.201.3:1033(10.1.100.41:20041)
misc=0 policy_id=7 pol_uuid_idx=8173 auth_info=0 chk_client_info=0 vd=0
serial=0001fb57 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
total session: 1
```

4. Check the expectation session list:

```
# diagnose sys session list expectation
session info: proto=17 proto_state=00 duration=74 expire=-44 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=2
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=255/255
state=new log f31
statistic(bytes/packets/allow_err): org=0/0/0 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=8->7/7->8 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=dnat 10.1.100.41:0->172.16.201.3:1033(10.1.100.41:0)
hook=pre dir=org act=noop 0.0.0.0:0->0.0.0.0:0(0.0.0.0:0)
misc=0 policy_id=7 pol_uuid_idx=8173 auth_info=0 chk_client_info=0 vd=0
serial=0001fb57 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  new disabled-by-policy

session info: proto=17 proto_state=00 duration=74 expire=-44 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=2
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=1 tunnel=/ vlan_cos=255/255
state=new log f31
statistic(bytes/packets/allow_err): org=0/0/0 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=0->8/8->0 gwy=0.0.0.0/0.0.0.0
```

```
hook=pre dir=org act=dnat 0.0.0.0:0->172.16.201.3:1033(10.1.100.41:20041)
hook=pre dir=org act=noop 0.0.0.0:0->0.0.0.0:0(0.0.0.0:0)
misc=0 policy_id=7 pol_uuid_idx=8173 auth_info=0 chk_client_info=0 vd=0
serial=0001fb57 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  new disabled-by-policy
```

5. Check that the packet sent from Server2 to the NAT IP address and port number is forwarded to Client1:

```
# diagnose sniffer packet any 'udp and port 1033 or 20041' 4
interfaces=[any]
filters=[udp and port 1033 or 20041]
12.001145 port1 in 172.16.200.55.4155 -> 172.16.201.3.1033: udp 4
12.001180 port2 out 172.16.200.55.4155 -> 10.1.100.41.20041: udp 4

2 packets received by filter
0 packets dropped by kernel
```

```
# diagnose sys session list
session info: proto=17 proto_state=00 duration=234 expire=1734 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=4
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=1 tunnel=/ vlan_cos=255/255
state=log intree
statistic(bytes/packets/allow_err): org=32/1/0 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=7->8/8->7 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=dnat 172.16.200.55:4155->172.16.201.3:1033(10.1.100.41:20041)
hook=post dir=reply act=snat 10.1.100.41:20041->172.16.200.55:4155(172.16.201.3:1033)
misc=0 policy_id=7 pol_uuid_idx=8173 auth_info=0 chk_client_info=0 vd=0
serial=0001fb57 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
```

Address objects

Addresses define sources and destinations of network traffic and can be used in many functions such as firewall policies, ZTNA, etc.

To view the possible uses list of address object usage:

1. Go to *Policy & Objects* > *Addresses*.
2. Click the number under *Ref*. The *Usage of Address:<Predefined address>* pane opens, where *<Predefined address>* is one of the predefined addresses, such as SSLVPN_TUNNEL_ADDR1.

3. In the *Usage of Address:<Predefined address>* pane, click *Possible Uses* to view the list.

When properly set up, these address objects can be used with great flexibility to make the configuration of different functions simpler and more intuitive. When used in a firewall policy, the FortiGate compares the IP addresses contained in packet headers with a policy's source and destination addresses to determine if the policy matches the traffic. The matching of IP addresses in packet headers is also performed for other FortiGate functions configured with address objects.

Address types

When creating an IPv4 address, there are several different types of addresses that can be specified. Which one is chosen will depend on which method most easily yet accurately describes the addresses that you are trying to include with as few entries as possible based on the information that you have. For instance, if you are trying to describe the addresses of a specific company's web server but do not know how extensive their web server farm is, you would be more likely to use a Fully Qualified Domain Name (FQDN) rather than a specific IP address. On the other hand, some computers do not have FQDNs and a specific IP address must be used.

The following table provides a short description of the different types of addresses:

Address type	Description
Subnet	The subnet type of address is expressed using a host address and a subnet mask. This is the most flexible of the address types because the address can refer to as little as one individual address (x.x.x.x/32) or as many as all of the available addresses (0.0.0.0/0). See Subnet on page 2034 and Dynamic policy — Fabric devices on page 2035 for more information.
IP range	The IP range type can be used to define a continuous set of IP addresses between one specific IP address and another (inclusive). It is a flexible way to describe a continuous set of addresses while being specific and granular, without needing to fall within the boundaries of standard subnets. See IP range on page 2038 for more information.
FQDN	The Fully Qualified Domain Name (FQDN) address type accepts an address string and resolves it to one or more IP addresses. It relies on DNS to keep up with address changes without having to manually change the IP addresses on the FortiGate. See FQDN addresses on page 2038 for more information. FQDN can also be specified as wildcard addresses such as *.example.com. See Using wildcard FQDN addresses in firewall policies on page 2040 for more information.
Geography	Geography addresses are those determined by the country/region of origin. The IPs for the country/region is automatically determined from the Geography IP database. See Geography based addresses on page 2042 and IPv6 geography-based addresses on page 2046 for more information.

Address type	Description
Wildcard	Wildcard addresses are addresses that identify ranges of IP addresses, reducing the amount of firewall addresses and security policies required to match some of the traffic on your network. See Wildcard addressing on page 2048 for more information.
Dynamic	Dynamic address objects are collections of addresses that are integrated from different external sources or other modules within the FortiGate. They can be used in policies that support the dynamic address type and come in different subtypes. • ClearPass: IP addresses gathered from the ClearPass Policy Manager. See ClearPass integration for dynamic address objects on page 2065 for more information. • Device & OS Identification: MAC addresses gathered from device detection that can be filtered by hardware vendor, model, OS, and OS version. See MAC address-based policies on page 2072 for more information. • Fabric Connector Address: IP addresses retrieved from SDN connectors, such as public and private cloud connectors. See Getting started with public and private SDN connectors on page 4599 for more information. • FortiNAC Tag: IP addresses collected from FortiNAC. See FortiNAC tag dynamic address on page 2068 for more information. • FortiVoice Tag: IP and MAC addresses collected from FortiVoice. See FortiVoice tag dynamic address on page 2068 for more information. • Fortinet Single Sign-On (FSSO): IP addresses of authenticated users from a FSSO collector agent, CPPM by FortiManager, or FortiNAC. See FSSO dynamic address subtype on page 2061 for more information. • RADIUS Single Sign-On (RSSO): IP address of the RADIUS single sign-on user matching the group value. See RSSO dynamic address subtype on page 2078 for more information. • Switch Controller NAC Policy Tag: MAC addresses collected from NAC policies.
IPAM	IPAM type firewall addresses can be configured on VPN gateways for automatic IP range allocation to VPN clients, allowing centralized IP management and traffic control across distributed subnets. See IPAM firewall address New on page 2051 for more information.
Device (Mac address)	A MAC address is a link layer-based address type and it cannot be forwarded across different IP segments. In FortiOS, you can configure a firewall address object with a singular MAC, wildcard MAC, multiple MACs, or a MAC range. See MAC address-based policies on page 2072, Adding MAC-based addresses to devices on page 153, ISDB well-known MAC address list on page 2074, and IPv6 MAC addresses and usage in firewall policies on page 2076 for more information.

Address type	Description
Interface subnet (CLI only)	For all interfaces set to a LAN or DMZ role, an option is available and enabled by default to automatically create an address object for the connected network. If the interface's subnet changes, the address object subnet changes too. See Interface subnet on page 2050 for more information.

Address group

Address groups are designed for ease of use in the administration of the device. If you have several addresses or address ranges that will commonly be treated the same or require the same security policies, you can put them into address groups, rather than entering multiple individual addresses in each policy that refers to them.

There are two different types of address groups and the following table provides a short description of each type:

Address group type	Description
Group	Members of an address group type group can belong to multiple address groups. See Address group on page 2057, Allow empty address groups on page 2059, and Address group exclusions on page 2060 for more information.
Folder	Members or an address group type folder can only belong to a single address folder. See Address folders on page 2057 for more information.

When an address group with no members is configured in a firewall policy, the policy will not match any traffic and will just match the implicit deny policy. See Allow empty address groups on page 2059 for more information.

Subnet

A subnet address object is usually used to refer internal networks or addresses which are defined by the network administrator.

A subnet address usually consists of a network address and a netmask, for example, 192.168.1.0 255.255.255.0. In this example, the network address is 192.168.1.0 and the netmask is 255.255.255.0. The network address defines the network to match and the netmask specify the IP address to match on the network.

In the above example, the subnet address 192.168.1.0 255.255.255.0 would match the following IP addresses:

```
192.168.1.1
192.168.1.2
192.168.1.3
```

```
...
192.168.1.255
```

For defining a subnet address object the valid format of IP address and netmask could be either:

`x.x.x.x/x.x.x.x`, such as 192.168.1.0/255.255.255.0

or

`x.x.x.x/x`, such as 192.168.1.0/24

To define a single address using subnet, use the netmask 255.255.255.255 or /32. A warning message will be shown if any other netmask is used and will not let the user save the address object.

To create a subnet address:

1. Go to *Policy & Objects* > *Addresses* and select *Address*.
2. Select *Create new*.
3. Enter a *Name* for the address object.
4. In the *Type* field, select *Subnet* from the dropdown menu.
5. In the *IP/Netmask* field, enter the address and subnet mask according to the format x.x.x.x/x.x.x.x or the short hand format of x.x.x.x/x
6. In the *Interface* field, leave as the default *any* or select a specific interface from the dropdown menu.
7. Enable/disable *Static route configuration*.
8. Enter any additional information in the *Comments* field.
9. Click *OK*.

Dynamic policy — Fabric devices

The dynamic address group represents the configured IP addresses of all Fortinet devices connected to the Security Fabric. It currently includes FortiManager, FortiAnalyzer, FortiClient EMS, FortiMail, FortiAP(s), and FortiSwitch(es). Like other dynamic address groups for fabric connectors, it can be used as an IPv4 address in firewall policies and objects.

The list of firewall addresses includes a default address object called `FABRIC_DEVICE`. You can apply the `FABRIC_DEVICE` object to the following types of policies:

- Firewall policy, including virtual wire pairs, NAT 46, and NAT 64 (IPv4 only)
- IPv4 shaping policy
- IPv4 ACL policy
- Security policy (NGFW mode)

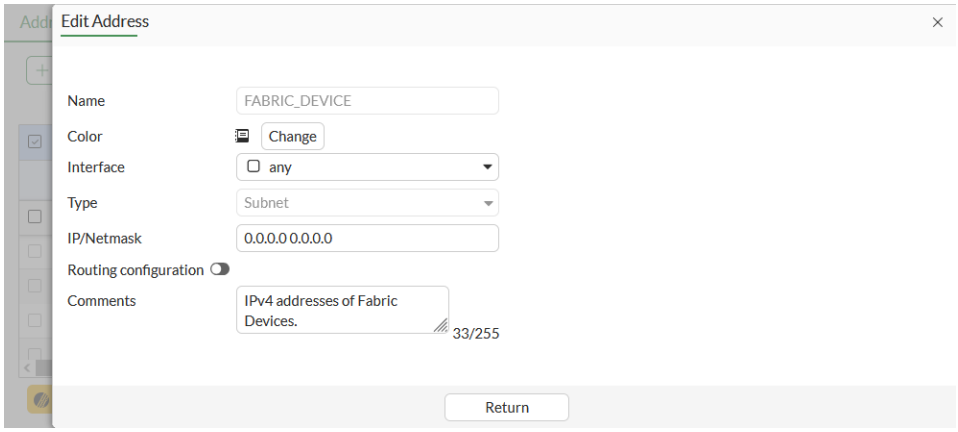
You cannot apply the `FABRIC_DEVICE` object to the following types of policies:

- IPv4 explicit proxy policy

You also cannot use the `FABRIC_DEVICE` object with the following settings:

- Custom extension on `internet-service`
- Exclusion of `addrgrp`

Initially the FABRIC_DEVICE object does not have an address value. The address value is populated dynamically as things change. As a result, you cannot edit the FABRIC_DEVICE object, add any addresses to the object, or remove any addresses from the object. The *Edit Address* pane in the GUI only has a *Return* button because the object is read-only:



The FABRIC_DEVICE object address values are populated based on:

- FortiAnalyzer IP (from the *Fabric Settings* pane)
- FortiManager IP (from the *Fabric Settings* pane)
- FortiMail IP (from the *Fabric Settings* pane)
- FortiClient EMS IP (from the *Fabric Settings* pane)
- FortiAP IPs (from the *FortiAP Setup* pane or DHCP)
- FortiSwitch IPs (from the *FortiSwitch Setup* page or DHCP)

To apply the FABRIC_DEVICE object to a firewall policy using the GUI:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Create a new policy or edit an existing policy.
3. For the *Destination* field, select *FABRIC_DEVICE* from the list of address entries.
4. Configure the rest of the policy as needed.
5. Click *OK*.

To apply the FABRIC_DEVICE object to a firewall policy using the CLI:

```
config firewall address
    edit "FABRIC_DEVICE"
        set type ipmask
        set comment "IPv4 addresses of Fabric Devices."
        set visibility enable
        set associated-interface ''
        set color 0
        set allow-routing disable
        set subnet 0.0.0.0 0.0.0.0
    next
end
```

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "FABRIC_DEVICE"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set fsso disable
        set nat enable
    next
end
```

Diagnose commands

You can run diagnose commands to list IP addresses of Fortinet devices that are configured in the Security Fabric or used in a security policy.

To view the IP addresses of Fabric devices:

```
(root) # diagnose firewall sf-addresses list

FabricDevices: 172.18.64.48
FortiAnalyzer: 172.18.60.25
FortiSandbox: 172.18.52.154
FortiManager: 172.18.28.31
FortiClientEMS: 172.18.62.6
FortiAP:
FortiSwitch:
FortiAP/SW-DHCP:
```

To view which IP addresses are used in a security policy:

```
(root) # diagnose ips pme fabric-address list
VDOM 0:
- builtin [mask=0x1e]:
  - type=4: 172.18.62.213
  - type=4: 172.18.62.219
  - type=2: 172.18.70.82
- query:
  - 168.254.1.2
  - 0.0.0.0
  - 168.254.1.2
```

IP range

The IP range type of address can describe a group of addresses while being specific and granular. It does this by specifying a continuous set of IP addresses between one specific IP address and another.

The format would be:

x.x.x.x-x.x.x.x, such as 192.168.110.100-192.168.110.120

To create an IP range address:

1. Go to *Policy & Objects* > *Addresses* and select *Address*.
2. Select *Create new*.
3. Enter a *Name* for the address object.
4. In the *Type* field, select *IP Range* from the dropdown menu.
5. In the *IP Range* field, enter the range of addresses in the following format: x.x.x.x-x.x.x.x (no spaces)
6. In the *Interface* field, leave as the default any or select a specific interface from the drop down menu.
7. Enter any additional information in the *Comments* field.
8. Click *OK*.

FQDN addresses

By using Fully Qualified Domain Name (FQDN) addressing you can take advantage of the dynamic ability of DNS to keep up with address changes without having to manually change the addresses on the FortiGate. FQDN addresses are most often used with external web sites but they can be used for internal web sites as well if there is a trusted DNS server that can be accessed. FQDN addressing also comes in handy for large web sites that may use multiple addresses and load balancers for their web sites. The FortiGate firewall automatically maintains a cached record of all the addresses resolved by the DNS for the FQDN addresses used.

For example, if you were doing this manually and you wanted to have a security policy that involved Google, you could track down all of the IP addresses that they use across multiple countries. Using the FQDN address is simpler and more convenient.

When representing hosts by an FQDN, the domain name can also be a subdomain, such as mail.example.com.

Valid FQDN formats include:

- <host_name>.<top_level_domain_name>, such as example.com
- <host_name>.<second_level_domain_name>.<top_level_domain_name>, such as mail.example.com.

The FortiGate firewall keeps track of the DNS TTLs so as the entries change on the DNS servers the IP address will effectively be updated for the FortiGate. As long as the FQDN address is used in a security policy, it stores the address in the DNS cache.

To create a Fully Qualified Domain Name address:

1. Go to *Policy & Objects* > *Addresses* and select *Address*.
2. Select *Create new*.
3. Enter a *Name* for the address object.

4. In the *Type* field, select *FQDN* from the dropdown menu.
5. Enter the domain name in the *FQDN* field.
6. In the *Interface* field, leave as the default *any* or select a specific interface from the dropdown menu.
7. Enable/disable *Static route configuration*.
8. Enter any additional information in the *Comments* field.
9. Click *OK*.

Address learning

FQDN addresses are learned in one of two ways:

1. FortiGate periodically queries its DNS server to resolve the FQDNs.
2. FortiGate passively learns the IP addresses by sniffing DNS traffic from the clients.

Using the second method, there is a possible security downside to using FQDN addresses. Using a fully qualified domain name in a security policy means that your policies are relying on the DNS server of the client to be accurate and correct. Should the DNS server be compromised, security policies requiring domain name resolution may no longer function properly.

You can manually override and disable passive learning of FQDN addresses by disabling the following command on the firewall address object:

```
config firewall address
    edit <address>
        set passive-fqdn-learning {disable | enable}
    next
end
```

Option	Description
`disable`	Disable passive learning of FQDNs.
`enable`	Enable passive learning of FQDNs (default).

Troubleshooting and diagnosis

To view the address objects and IPs resolved:

```
# diagnose firewall fqdn list-ip
```

```
# diagnose firewall fqdn get-ip <fqdn address>
```

To dump the DNS FQDN and DNS cache:

```
# diagnose test application dnsproxy 6
```

```
# diagnose test application dnsproxy 7
```

Using wildcard FQDN addresses in firewall policies

You can use wildcard FQDN addresses in firewall policies. IPv4, IPv6, ACL, local, shaping, NAT64, NAT46, and NGFW policy types support wildcard FQDN addresses. Wildcard FQDN addresses can also be used in Web and FTP proxy policies for FortiGate firewalls.

For wildcard FQDN addresses to work, the FortiGate should allow DNS traffic to pass through.

Initially, the wildcard FQDN object is empty and contains no addresses. When the client tries to resolve a FQDN address, the FortiGate will analyze the DNS response. The IP address(es) contained in the answer section of the DNS response will be added to the corresponding wildcard FQDN object. It is therefore necessary to have the DNS session-helpers defined in the `config system session-helper` setting.

Since FortiGate must analyze the DNS response, it does not work with DNS over HTTPS.

In FortiOS 7.0 and later, FortiGate supports DNS over TLS. It is possible to analyze DNS responses sent over DoT, as long as there is a firewall policy that allows the DNS traffic from the client and is configured with a DNS filter that supports DoT. For information on configuring this, see DNS inspection with DoT and DoH on page 2412.

When the wildcard FQDN gets the resolved IP addresses, FortiOS loads the addresses into the firewall policy for traffic matching.

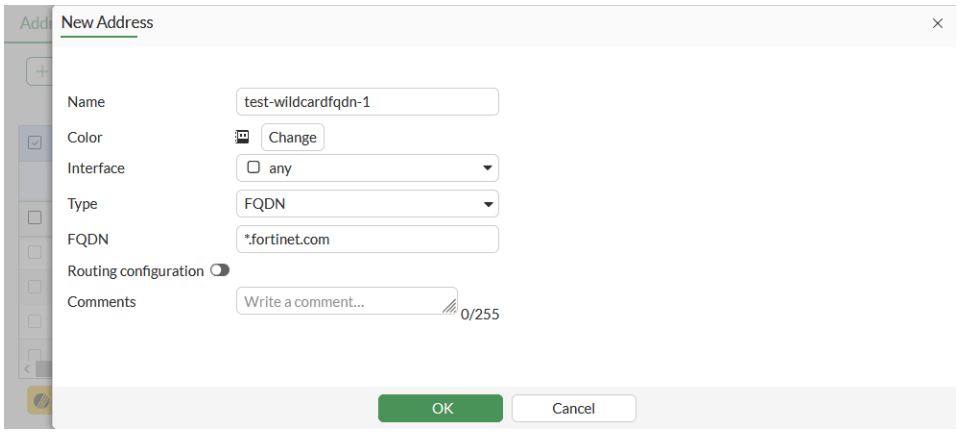
The FortiGate will keep the IP addresses in the FQDN object table as long as the DNS entry itself has not expired. Once it expires, the IP address is removed from the wildcard FQDN object until another query is made. At any given time, a single wildcard FQDN object may have up to 1000 IP addresses.

The DNS expiry TTL value is set by the authoritative name server for that DNS record. If the TTL for a specific DNS record is very short and you would like to cache the IP address longer, then you can extend it with the CLI. See To extend the TTL for a DNS record in the CLI: on page 2042

Wildcard FQDN IPs are synchronized to other autoscale members whenever a peer learns of a wildcard FQDN address.

To create a wildcard FQDN using the GUI:

1. Go to *Policy & Objects* > *Addresses* and select *Address*.
2. Click *Create new*.
3. Specify a *Name*.
4. For *Type*, select *FQDN*.
5. For *FQDN*, enter a wildcard FQDN address, for example, `*.fortinet.com`.



6. Click *OK*.

To use a wildcard FQDN in a firewall policy using the GUI:

1. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
2. For *Destination*, select the wildcard FQDN.
3. Configure the rest of the policy as needed.
4. Click *OK*.

To create a wildcard FQDN using the CLI:

```
config firewall address
    edit "test-wildcardfqdn-1"
        set type fqdn
        set fqdn "*.fortinet.com"
    next
end
```

To use wildcard FQDN in a firewall policy using the CLI:

```
config firewall policy
    edit 2
        set srcintf "port3"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "test-wildcardfqdn-1"
        set action accept
        set schedule "always"
        set service "ALL"
        set auto-asic-offload disable
        set nat enable
    next
end
```

To use the diagnose command to list resolved IP addresses of wildcard FQDN objects:

```
# diagnose firewall fqdn list
List all FQDN:
*.fortinet.com: ID(48) ADDR(96.45.36.159) ADDR(192.168.100.161) ADDR(65.39.139.161)
```

Alternatively:

```
# diagnose test application dnsproxy 6
worker idx: 0
vfid=0 name=*.fortinet.com ver=IPv4  min_ttl=3266:0, cache_ttl=0 , slot=-1, num=3, wildcard=1
      96.45.36.159 (ttl=68862:68311:68311) 192.168.100.161 (ttl=3600:3146:3146) 65.39.139.161
(ttl=3600:3481:3481)
```

To use the diagnose command for firewall policies which use wildcard FQDN:

```
# diagnose firewall iprope list 100004
...
destination fqdn or dynamic address (1):*.fortinet.com ID(48) uuid_idx=57 ADDR(208.91.114.104)
ADDR(208.91.114.142) ADDR(173.243.137.143) ADDR(65.104.9.196) ADDR(96.45.36.210)
...
```

To extend the TTL for a DNS record in the CLI:

The TTL for DNS records can be configured globally, or for a specific FQDN address. If it is configured for an FQDN address, that setting will supersede the global setting for that address. See Important DNS CLI commands on page 358 for information about configuring a global TTL.

In this the example the `set cache-ttl` value has been extended to 3600 seconds.

```
config firewall address
    edit "fortinet.com"
        set type fqdn
        set fqdn "www.fortinet.com"
        set cache-ttl 3600
    next
end
```

Geography based addresses

Geography addresses are those determined by country of origin. The IP for the country or region is automatically determined from the Geography IP database. The `geoip-full-db` option controls whether the full geographic database is loaded into the kernel.

To view IP Geography database:

```
# diagnose autoupdate versions | grep -A 6 "IP Geography DB"
IP Geography DB
---------
```

```
Version: 3.00152
Contract Expiry Date: n/a
Last Updated using manual update on Thu Nov 17 17:52:00 2022
Last Update Attempt: Wed Nov 23 10:56:46 2022
Result: No Updates
```

Without a valid license, local IP geography database will continue to work. However the FortiGate will stop receiving geography IP updates from the FortiGuard servers and the geography IP database will no longer be updated. IP geolocation service is part of base services included with all FortiCare support contracts. See FortiGuard Security Services for more information.

To control if the full database is loaded:

```
config sys global
    set geoip-full-db {enable | disable}
end
```

Option	Description
`enable`	The full Geography IP database is downloaded, and geographic information is available in traffic logs. The FortiView Countries/Regions widget and monitor are available.
`disable`	The full Geography IP database is not downloaded, conserving memory. The FortiView Countries/Regions widget and monitor cannot be used.

By default, after upgrading to FortiOS 7.6.5:

- On devices with 4GB RAM or more, `geoip-full-db` is enabled.
- On devices with 2GB RAM, `geoip-full-db` is disabled. If required, it must be enabled manually after upgrading, but note that doing so will increase memory usage and may affect device performance.

To create a geography address:

1. Go to *Policy & Objects* > *Addresses* and select *Address*.
2. Select *Create new*.
3. Enter a *Name* for the address object.
4. In the *Type* field, select *Geography* from the dropdown menu.
5. In the *Country/Region* field, select a single country from the dropdown menu.
6. In the *Interface* field, leave as the default *any* or select a specific interface from the dropdown menu.
7. Enter any additional information in the *Comments* field.
8. Click *OK*.

Overrides

It is possible to assign a specific IP address range to a customized country ID. Generally, geographic addressing is done at the VDOM level; it could be considered global if you are using the root VDOM, but the `geoip-override`

`setting` is a global setting.

To configure a geography IP override:

1. Assign a specific IP address range to a customized country ID:

```
config system geoip-override
    edit "MyCustomCountry"
        config ip-range
            edit 1
                set start-ip 1.1.1.1
                set end-ip 1.1.1.2
            next
        end
    next
end
```

2. Use get sys geoip-country XX to determine the name corresponding to the custom 2-digit country code A0:

```
# get sys geoip-country A0
id                  : A0
name                : MyCustomCountry
```

3. Show the full configuration of the geography IP override just created to show that it corresponds to country code A0:

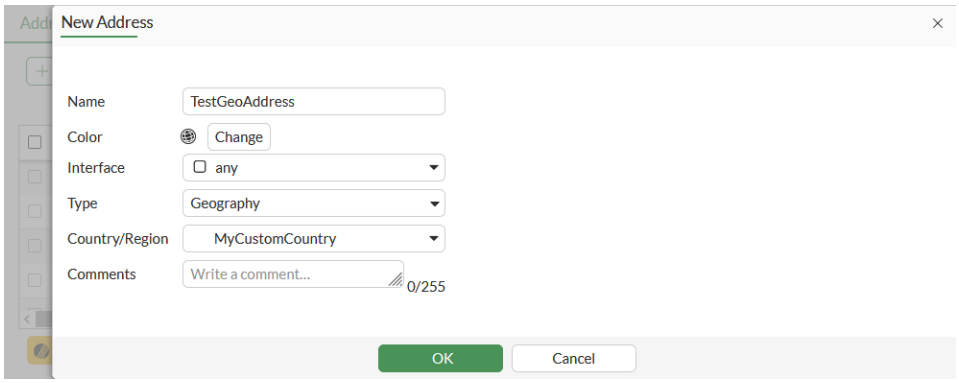
```
# show full sys geoip-override
config system geoip-override
    edit "MyCustomCountry"
        set description ''
        set country-id "A0"
        config ip-range
            edit 1
                set start-ip 1.1.1.1
                set end-ip 1.1.1.2
            next
        end
    next
end
```

To configure a geography address:

1. Enable debug to display the CLI commands running on the backend in response to certain GUI configuration:

```
# diagnose debug enable
# diagnose debug cli 7
Debug messages will be on for 30 minutes.
```

2. Go to *Policy & Objects > Addresses* and create a geography address using the previously created custom country code:



3. Observe the corresponding CLI commands run on the backend:

```
FGT # 0: config firewall address
0: edit "TestGeoAddress"
0: set type geography
0: set country "A0"
0: end
```

Diagnose commands

There are a few diagnose commands used with geographic addresses:

```
diagnose firewall ipgeo [country-list | ip-list | ip2country | override | copyright-notice]
```

Diagnose command	Description
country-list	List of all countries.
ip-list	List of the IP addresses associated with the country.
ip2country	Used to determine the physical and registered locations of the IP address as well and if the type is anycast.
override	List of user defined geography data; items configured with the `config system geoip-override` command.
copyright-notice	Shows the copyright notice.

```
diagnose geoip [geoip-query | ip2country | iprange]
```

Diagnose command	Description
geoip-query	Used to determine the complete geolocation of a specific IP address from the FortiGuard IP Geography DB.
ip2country	Used to determine which country a specific IP address is assigned to.
Iprange	List the IP addresses or IP ranges associated with the country.

For more details and examples using these diagnose commands, see the Fortinet Community article Technical Tip: Commands to verify GeoIP information and troubleshoot GeoIP database.

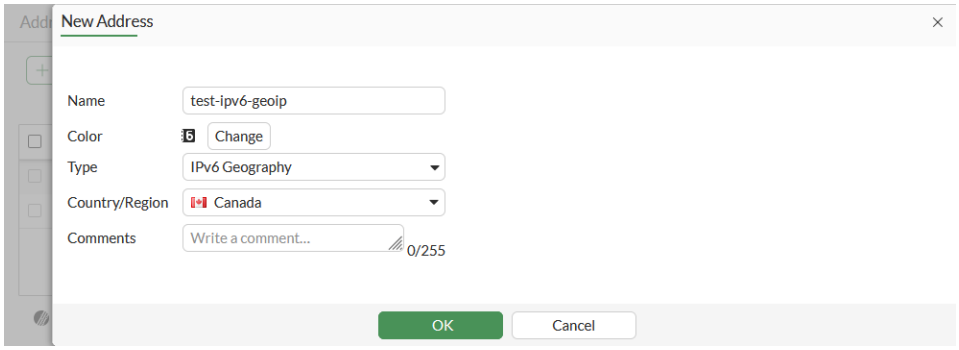
IPv6 geography-based addresses

Geography-based IPv6 addresses can be created and applied to IPv6 firewall policies.

IPv6 geography-based addresses do not support `geoip-override` or `geoip-anycast`.

To create an IPv6 geography-based address in the GUI:

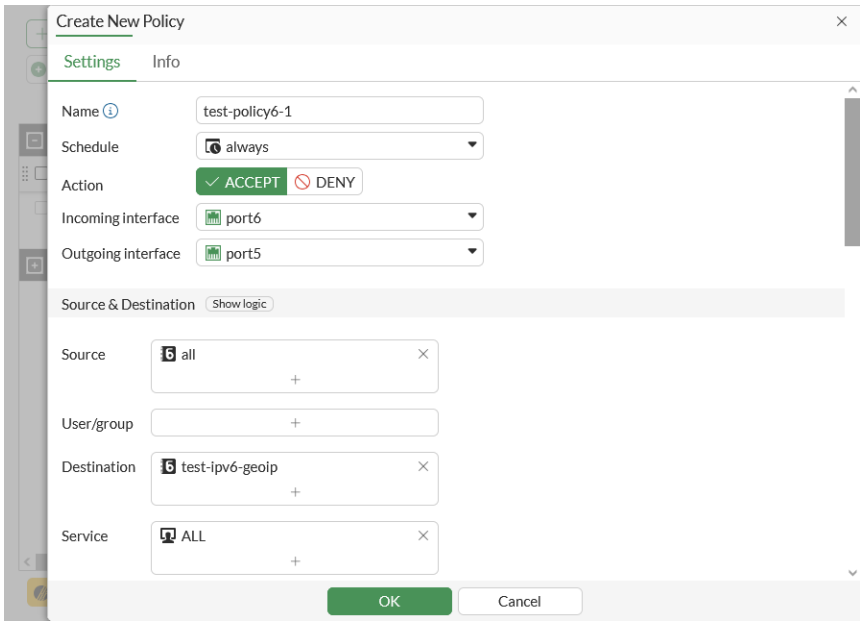
1. Go to *Policy and Objects* > *Addresses* and select *IPv6 Address*.
2. Click *Create new*.
3. Enter a name for the address.
4. Set *Type* to *IPv6 Geography*.
5. Select the *Country/Region* from the list.
6. Optionally, enter comments.



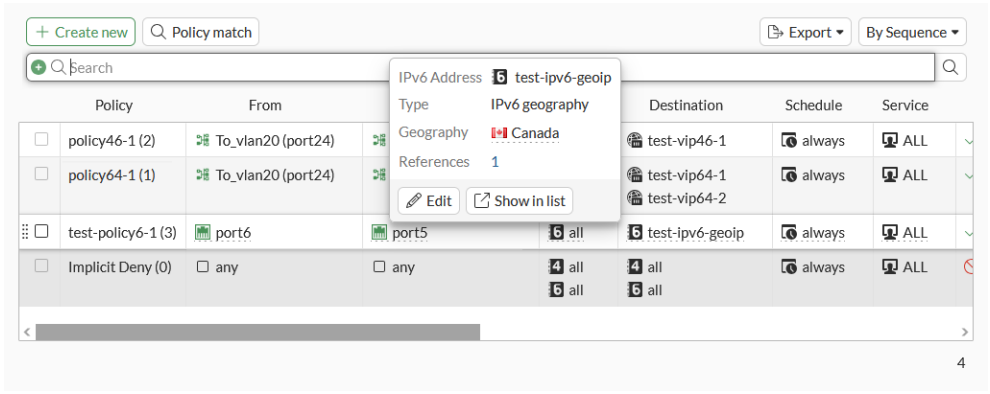
7. Click *OK*.

To use the IPv6 geography address in a policy:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Edit an existing policy, or create a new one, using the IPv6 geography address as the *Source* or *Destination Address*.



3. In the policy list, hover over the address to view details.



To configure an IPv6 geography-based address in the CLI:

1. Create an IPv6 geography-based address:

```
config firewall address6
    edit "test-ipv6-geoip"
        set type geography
        set color 6
        set comment "IPv6 Geography address"
        set country "CA"
    next
end
```

2. Use the IPv6 geography-based address in a policy:

```
config firewall policy
    edit 1
        set name "test-policy6-1"
```

```
        set srcintf "port6"
        set dstintf "port5"
        set srcaddr6 "all"
        set dstaddr6 "test-ipv6-geoip"
        set action accept
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

Wildcard addressing

Wildcard addresses are addresses that identify ranges of IP addresses, reducing the amount of firewall addresses and security policies required to match some of the traffic on your network. Wildcard addresses are an advanced feature, usually required only for complex networks with complex firewall filtering requirements. By using these wildcard addresses in the firewall configuration, administrators can eliminate creating multiple, separate IP-based address objects and then grouping them to then apply to multiple security policies.

A wildcard address consists of an IP address and a wildcard netmask, for example, 192.168.0.56 255.255.0.255 or 2000:172:16:200::55 ffff:ffff:ffff:ffff::ffff. The IP address defines the networks to match and the wildcard netmask defines the specific addresses to match on these networks.

In a wildcard netmask, zero denotes ignoring the value of the octet in the IP address. This means that the wildcard firewall address matches any number in this address octet. This also means that the number included in this octet of IP address is ignored and can be any number. Usually, if the octet in the wildcard netmask is zero, the corresponding octet in the IP address is also zero.

In a wildcard netmask, a number denotes matching addresses according to how the numbers translate into binary addresses. For example, the wildcard netmask is 255; the wildcard address will only match addresses with the value for this octet that is in the IP address part of the wildcard address. So, if the first octet of the IP address is 192 and the first octet of the wildcard netmask is 255, the wildcard address will only match addresses with 192 in the first octet.

For example, if the IP address is 192.168.0.56 and the wildcard netmask is 255.255.0.255, then the wildcard address would match the following IP addresses:

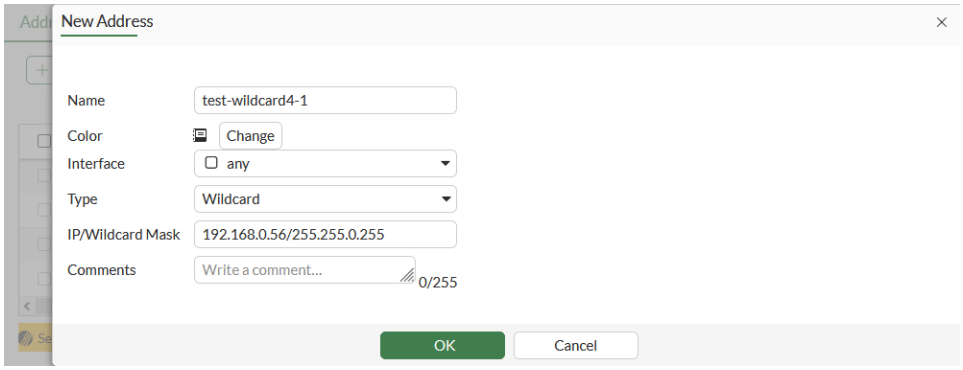
```
192.168.0.56
192.168.1.56
192.168.2.56
...
192.168.255.56
```

The wildcard addresses 192.168.0.56 255.255.0.255 and 192.168.1.56 255.255.0.255 define the same thing since the 0 in the wildcard mask means to match any address in the third octet.

To configure an IPv4 wildcard address in the GUI:

1. Go to *Policy & Objects* > *Addresses* and, on the *Address* tab, click *Create New*.
2. Enter a name for the address.
3. Set *Type* to *Wildcard*.

4. Set *IP/Wildcard Mask* to the IPv4 address and wildcard netmask.



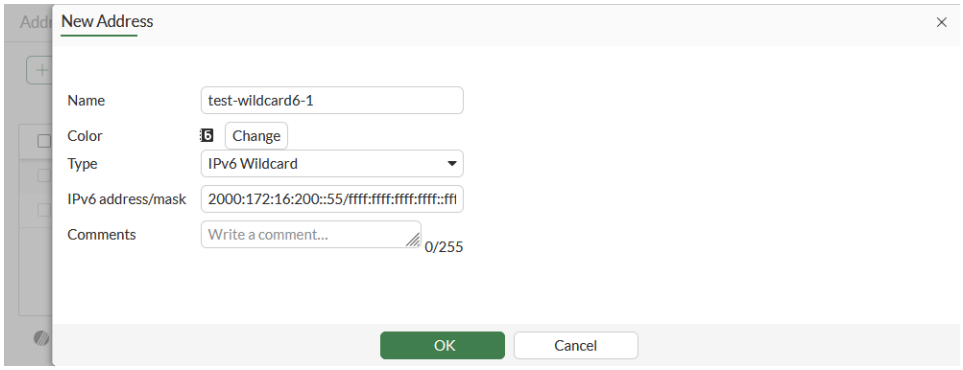
5. Click *OK*.

To configure an IPv4 wildcard address in the CLI:

```
config firewall address
    edit "test-wildcard4-1"
        set type wildcard
        set wildcard 192.168.0.56 255.255.0.255
    next
end
```

To configure an IPv6 wildcard address in the GUI:

1. Go to *Policy & Objects* > *Addresses* and, on the *IPv6 Address* tab, click *Create New*.
2. Enter a name for the address.
3. Set *Type* to *IPv6 Wildcard*.
4. Set *IPv6 address/mask* to the IPv6 address and wildcard netmask.



5. Click *OK*.

To configure an IPv6 wildcard address in the CLI:

```
config firewall address6
    edit "test-wildcard6-1"
        set type wildcard
        set wildcard 2000:172:16:200::55 ffff:ffff:ffff:ffff::ffff
```

```
    next
end
```

Interface subnet

Interface subnet address type enables an address object to be created automatically for the interface with which it is associated. Once created, the address object is updated when the interface IP/netmask changes on the associated interface.

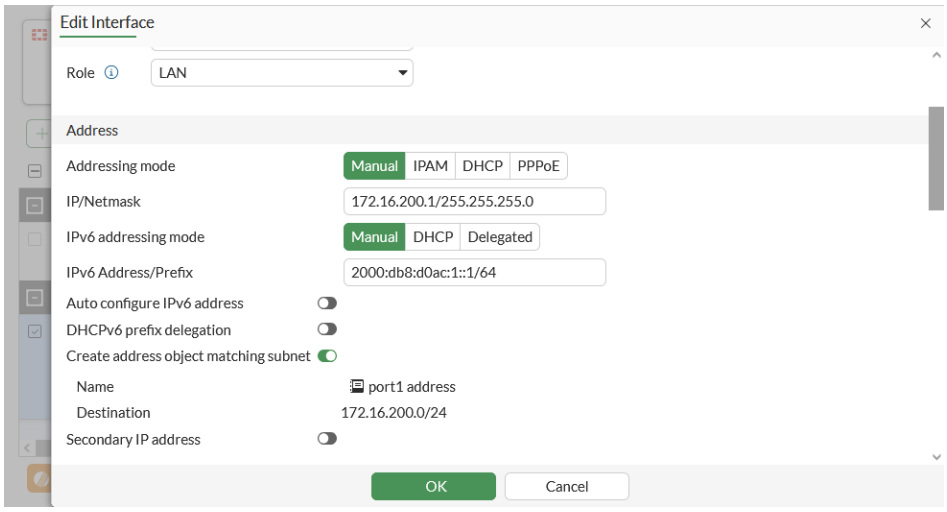
To create the interface subnet address type object, create or edit an interface under *Network* > *Interfaces*, and enable the *Create address object matching subnet* option.

The *Create address object matching subnet* option is automatically enabled and displayed in the GUI when *Role* is set to *LAN* or *DMZ*.

When you disable the *Create address object matching subnet* option, the feature is disabled, and the associated firewall address is deleted.

To create an interface subnet:

1. Go to *Network* > *Interfaces*.
2. Select *Create New* > *Interface* or select existing interface and *Edit*.
3. Set *Role* to either *LAN* or *DMZ*.
4. Verify that *Create address object matching subnet* is available and automatically enabled.



5. Click *OK*.

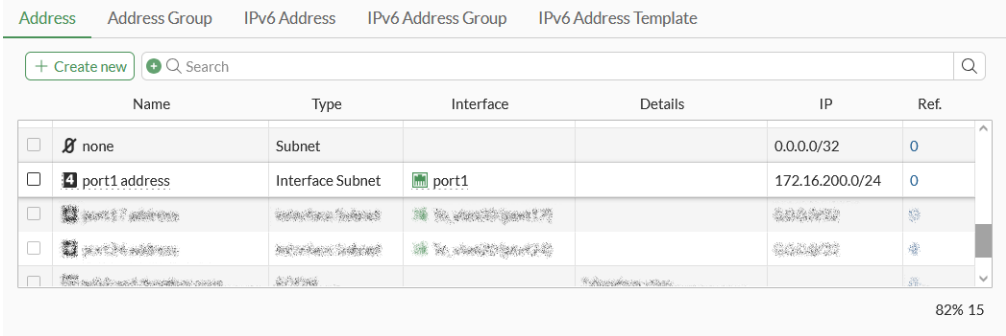
The following is an example of how to configure an interface subnet firewall address on the CLI:

```
config firewall address
    edit "port1 address"
        set type interface-subnet
        set interface "port1"
```

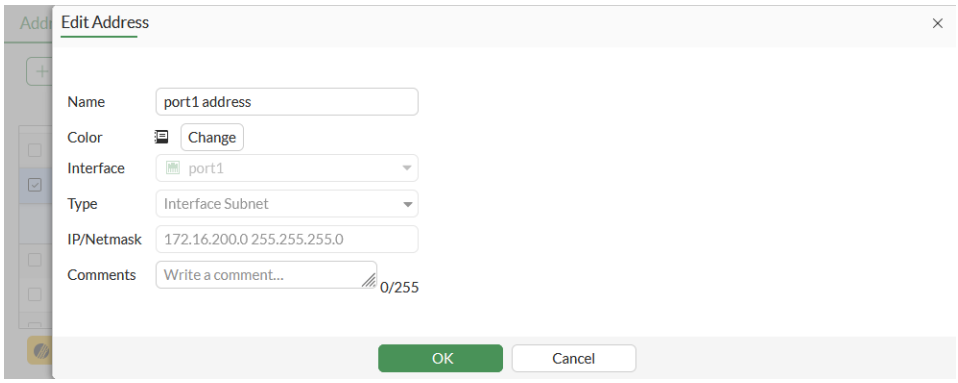
```
    next
end
```

Interface subnet addresses are automatically created when *Role* is set to *LAN* or *DMZ* in the *Interface* page, or you can manually configure interface subnet addresses in the CLI. You cannot choose *Interface Subnet* in the GUI when creating the address, but after the address is created, *Interface Subnet* displays in the GUI. However, all the settings are grayed out, except *Name* and *Comments*, which can be edited.

When *Role* is set to *LAN* or *DMZ* in the *Interface* page, the new address object displays on the *Policy & Objects* > *Address* page.



After the address is created, the subnet is dynamically assigned to the address object, which can be seen in both GUI and CLI. If the interface address changes, the subnet will update dynamically.




```
config firewall address
    edit "port1 address"
        set type interface-subnet
        set subnet 172.16.200.0 255.255.255.0
        set interface "port1"
    next
end
```

IPAM firewall address NEW

IPAM traditionally manages IP assignments directly on FortiGate interfaces by automatically configuring its IP, subnet, and DHCP based on network sizes. With this feature, IPAM can manage IP pools for a Firewall Address object. IPAM type firewall addresses in turn can be configured on VPN gateways for automatic IP range

allocation to VPN clients. This allows centralized IP management and traffic control across distributed subnets, improving scalability and operational efficiency.

FortiGate also supports per-device subnet allocation for Fabric object firewall addresses that use IPAM. The `ipam-allocate-unique` option can be enabled for firewall address objects configured with type `ipam` and `fabric-object enable`. When enabled, each FortiGate in the Security Fabric receives a unique subnet allocation from IPAM instead of inheriting the same subnet through CSF synchronization.

This feature allows greater flexibility in multi-device deployments where shared IPAM addresses across all FortiGates are not desirable.

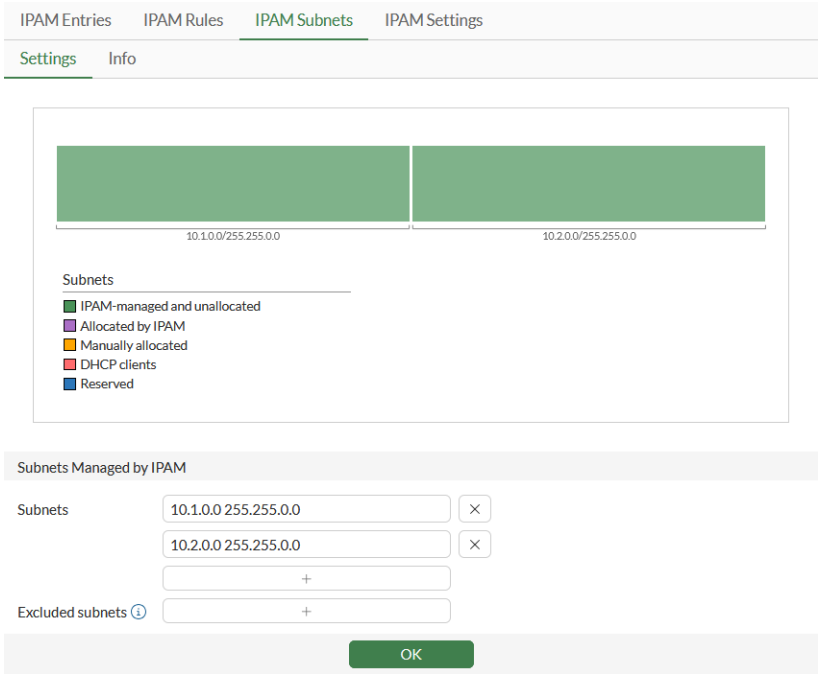
```
config firewall address
    edit "ipam"
        set type ipam
        set fabric-object enable
        set subnet 192.168.0.0 255.255.255.0
        set ipam-allocate-unique enable
    next
end
```

Example

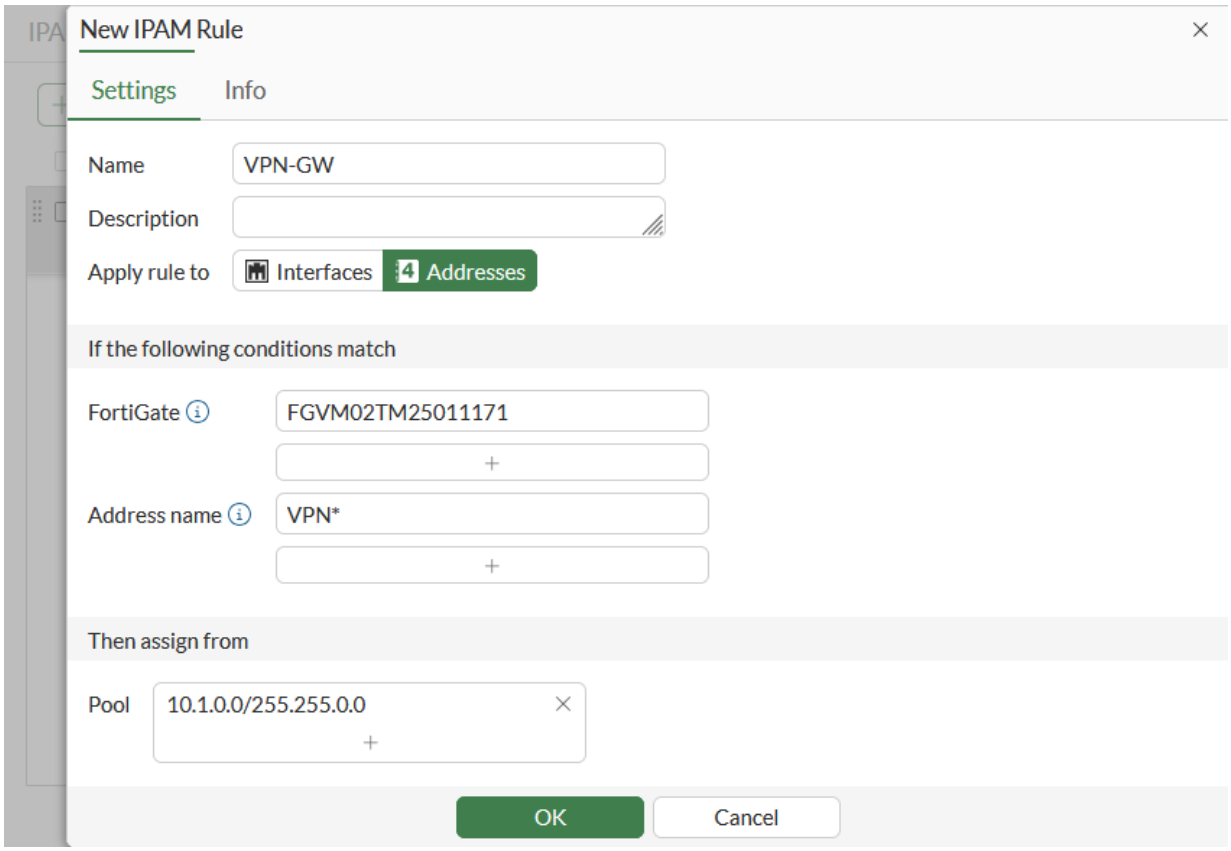
In this example, we have 2 IP range allocation on IPAM. One range will be dedicated to address objects with a name starting with VPN. The other range will be assigned to other address objects.

To configure an IPAM firewall address in the GUI:

1. Enable IPAM and create 2 IPAM subnets:
   a. Go to *Network > IPAM > IPAM Settings*.
   b. Set *Status* to *Enable* then click *OK*.
   c. Go to *Network > IPAM > IPAM Subnets*.
   d. Create 2 subnets *10.1.0.0 255.255.0.0* and *10.2.0.0 255.255.0.0*.
   e. Click *OK*.

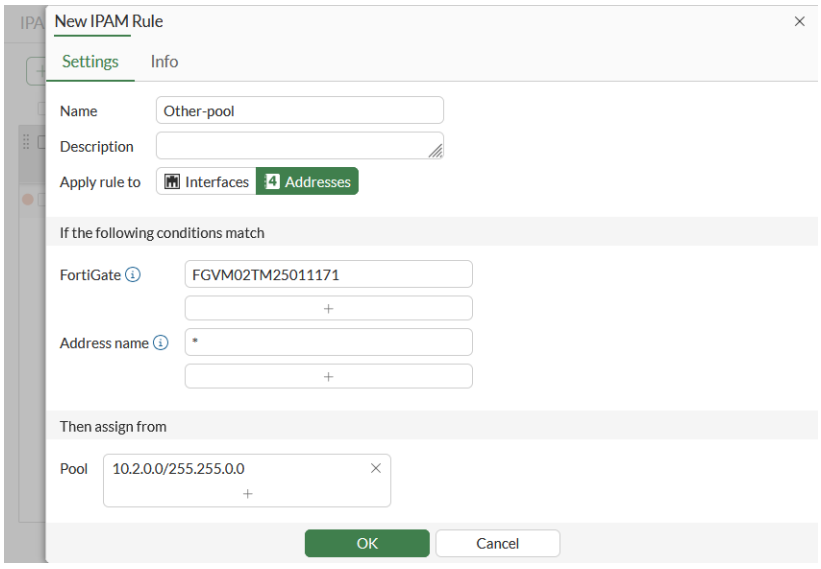


2. Create an IPAM rule for VPN firewall addresses:
   a. Go to *Network* > *IPAM* > *IPAM Rules* and click *Create new*.
   b. Enter a name for the rule.
   c. Set *Apply rule to* to *Addresses*.
   d. Set *Address Name* to the wildcard VPN*.
   e. For the *Pool*, select the first subnet pool.



f. Click *OK*.

3. Create another IPAM rule for other firewall addresses, when *Address name* matches *, and assign the second address pool, *10.2.0.0/255.255.0.0*.



4. Create a firewall address with the IPAM type that starts with *VPN*:

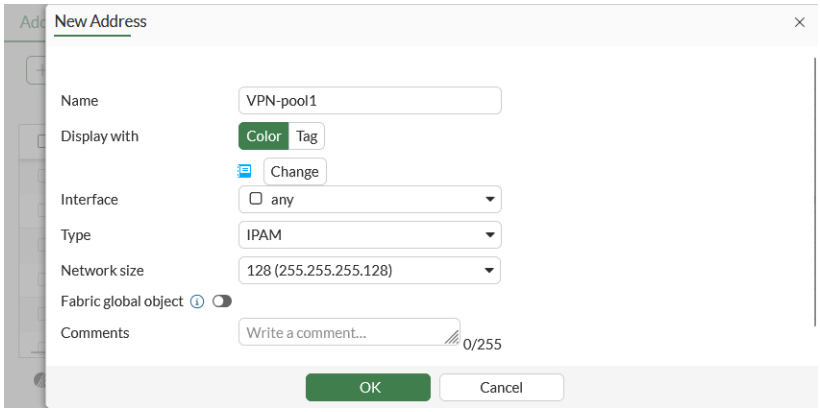
a. Go to *Policy & Objects* > *Addresses* and click *Create new*.

b. Enter a name for the address, such as *VPN-pool1*.

c. Set *Type* to *IPAM*.

d. Choose a *Network size*.

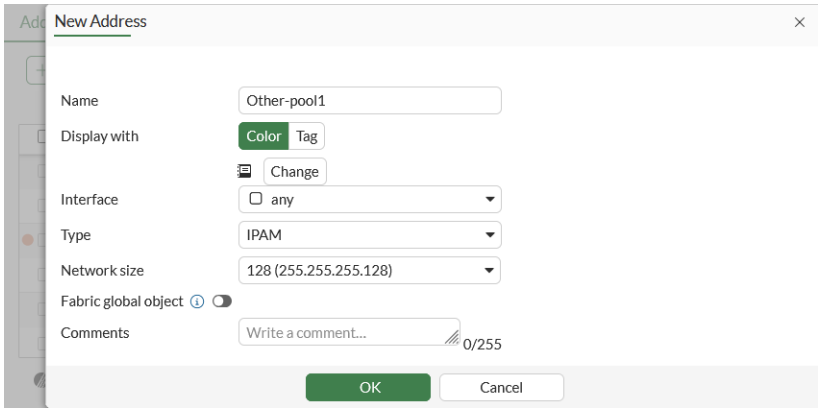
e. Configure the remaining settings as needed.



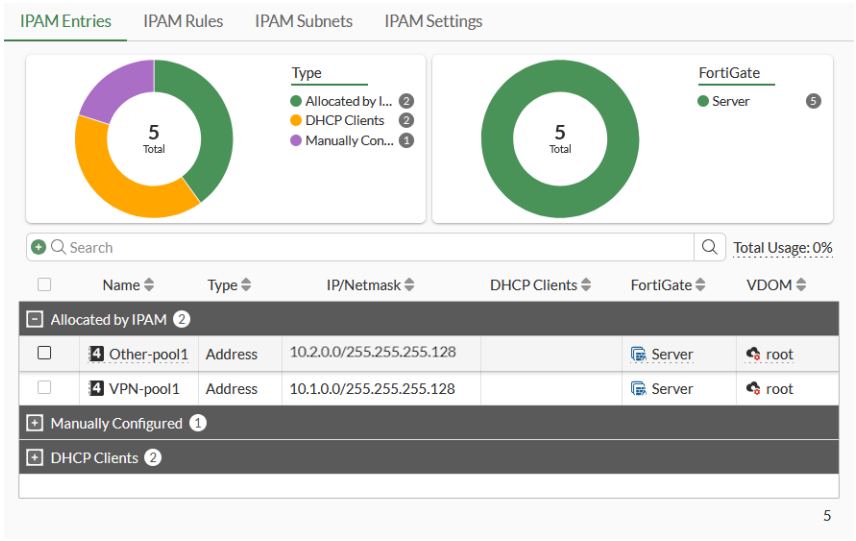
f. Click *OK*.

IPAM will start to allocate an address in the range 10.1.0.0/16.

5. Create a firewall address with the IPAM type that starts with another prefix.



6. Go to *Network* > *IPAM* > *IPAM Entries*. As the address is managed by IPAM, it is listed on the *IPAM Entries* tab.



7. Check the IPAM assignments from the CLI as well.

To configure an IPAM firewall address in the CLI:

1. Create an IPAM rule for firewall addresses:

```
config system ipam
    set status enable
    set manage-lan-addresses disable
    set manage-lan-extension-addresses disable
    set manage-ssid-addresses disable
    config pools
        edit "1"
            set subnet 10.1.0.0 255.255.0.0
        next
        edit "2"
            set subnet 10.2.0.0 255.255.0.0
        next
    end
    config rules
        edit "VPN-GW"
            set item-type address
            set item-name "VPN*"
            set device "FGVM02TM25XXXXXX"
            set vdom "*"
            set pool "1"
        next
        edit "Other-pool"
            set item-type address
            set item-name "*"
            set device "FGVM02TM25XXXXXX"
            set vdom "*"
            set pool "2"
        next
    end
end
```

2. Create a firewall address with the IPAM type:

```
config firewall address
    edit "VPN-pool1"
        set type ipam
        set color 17
        set managed-subnetwork-size 128
    next
    edit "Other-pool1"
        set type ipam
        set managed-subnetwork-size 128
    next
end
```

3. Check that the IPAM type firewall address is managed by the IPAM sever:

```
# diagnose sys ipam list entries
Entries: (sn, vdom, item-type, item, subnet/mask, conflict)
```

```
IPAM Entries:
FGVM02TM25011171 root firewall.address Other-pool1 10.2.0.0/25
FGVM02TM25011171 root firewall.address VPN-pool1 10.1.0.0/25
```

Address group

The use of groups is not mandatory. However, adding individual addresses to a policy sometimes becomes tedious. If you use several different addresses with a given policy, these address objects can be grouped into an address group as it is much easier to add or subtract addresses from the group.

Security policies require addresses with homogenous network interfaces. Therefore, address groups should contain only addresses bound to the same network interface or Any.

For example, if address 1.1.1.1 is associated with port1, and address 2.2.2.2 is associated with port2, they cannot be in the same group. However, if 1.1.1.1 and 2.2.2.2 are configured with an interface of Any, they can be grouped, even if the addresses involve different networks.

To create an address group:

1. Go to *Policy & Objects* > *Addresses* and select *Address Group*.
2. Go to *Create new*.
3. Enter a *Name* for the address object.
4. In the *Type* field, select *Group*.
5. Select the *+* in the *Members* field. The *Select Entries* pane opens.
6. Select members of the group. It is possible to select more than one entry. Select the *x* icon in the field to remove an entry.
7. Enable/disable *Static route configuration*.
8. Enter any additional information in the *Comments* field.
9. Click *OK*.

Address folders

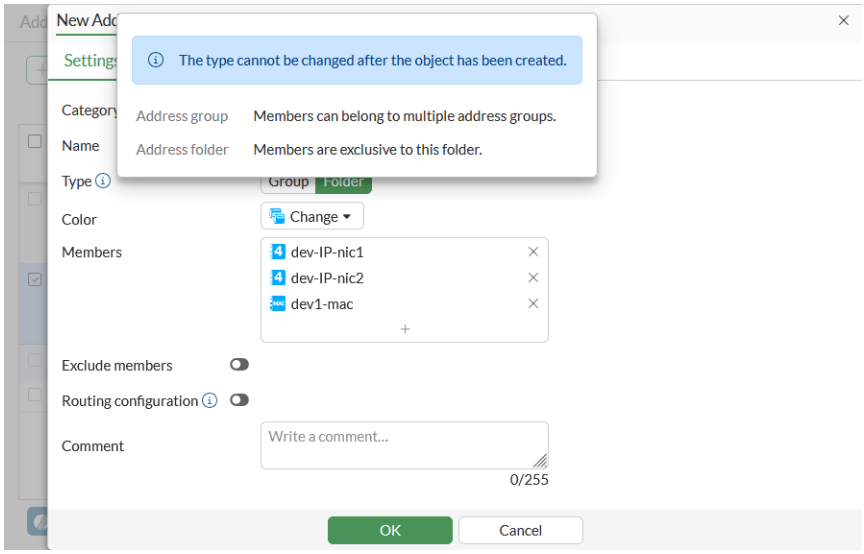
Some address objects logically belong to the same device, such as two IPs from the same computer. These address objects can be grouped into an address folder, which is an exclusive list of address objects that do not appear in other address groups or folders.

In the CLI, the folder type can be set after the member list is already populated. If the member list contains an incompatible entry, then the setting will be discarded when the `next`/`end` command is issued. If the folder type is set before the member list is populated, then the possible member entry list will be filtered according to the selected type.

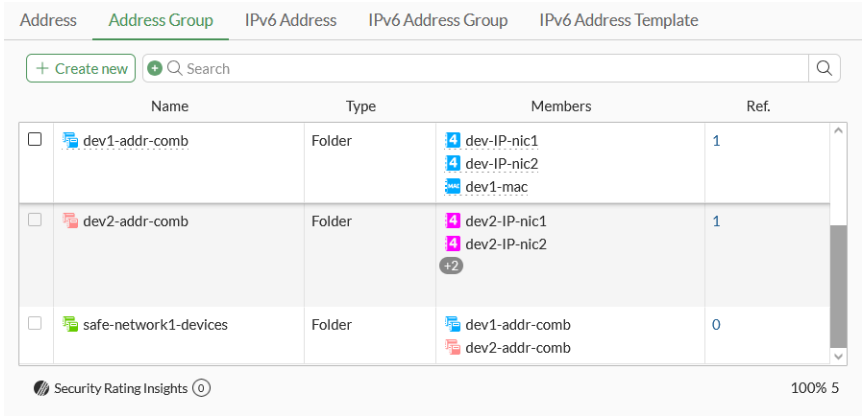
To create an address folder in the GUI:

1. Go to *Policy & Objects* > *Addresses* and select *Address Group*.
2. Click *Create new* and enter a name.

3. For *Type*, select *Folder*.
4. For *Members*, click the *+* to add the addresses. Address folders and groups are exclusive, so the *Select Entries* window filters out address objects that are a member of an existing group or folder.



5. Click *OK*.
6. In the address group table, the folders' members are shown in the *Members* column:



To configure an address folder in the CLI:

```
config firewall addrgrp
    edit "dev1-addr-comb"
        set type folder
        set member "dev1-IP-nic1" "dev1-IP-nic2" "dev1-mac"
        set comment ''
        set exclude disable
        set color 18
    next
end
```

```
config firewall addrgrp
    edit "dev2-addr-comb"
```

```
        set type folder
        set member "dev2-IP-nic1" "dev2-IP-nic2" "dev2-IP-nic3" "dev2-mac"
        set comment ''
        set exclude disable
        set color 5
    next
end
```

```
config firewall addrgrp
    edit "safe-network1-devices"
        set type folder
        set member "dev1-addr-comb" "dev2-addr-comb"
        set comment ''
        set exclude disable
        set color 13
    next
end
```

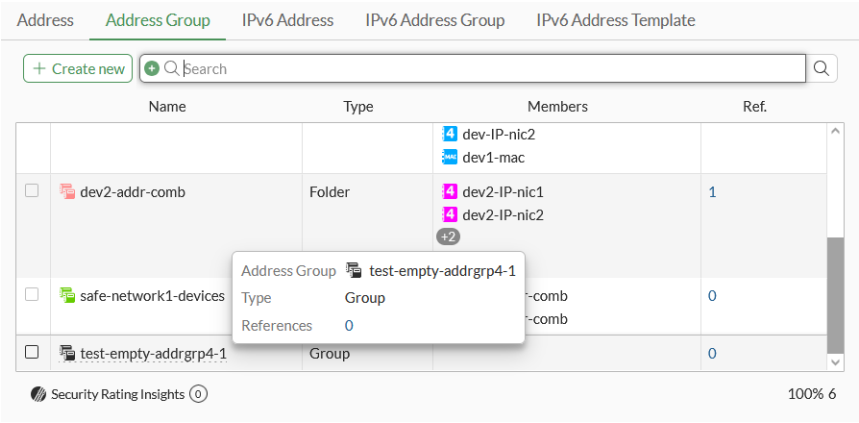
Allow empty address groups

Address groups with no members can be configured in the GUI, CLI, and through the API. In previous versions of FortiOS, error messages appear for empty address groups and they cannot be configured.

When an address group with no members is configured in a firewall policy, the policy will not match any traffic. In this case, policy matching logic will proceed down the list of firewall policies until matching the implicit deny policy.

To create an empty address group in the GUI:

1. Go to *Policy & Objects* > *Addresses* and select *Address Group*.
2. Click *Create new* .
3. Enter a name.
4. Click *OK*. The *This field is required* error is not displayed under the *Members* field, and the group is listed in the *Address Group* list with no members:



To create an empty address group in the CLI:

```
config firewall addrgrp
    edit "test-empty-addrgrp4-1"
    next
end
```

No error message is returned in the console.

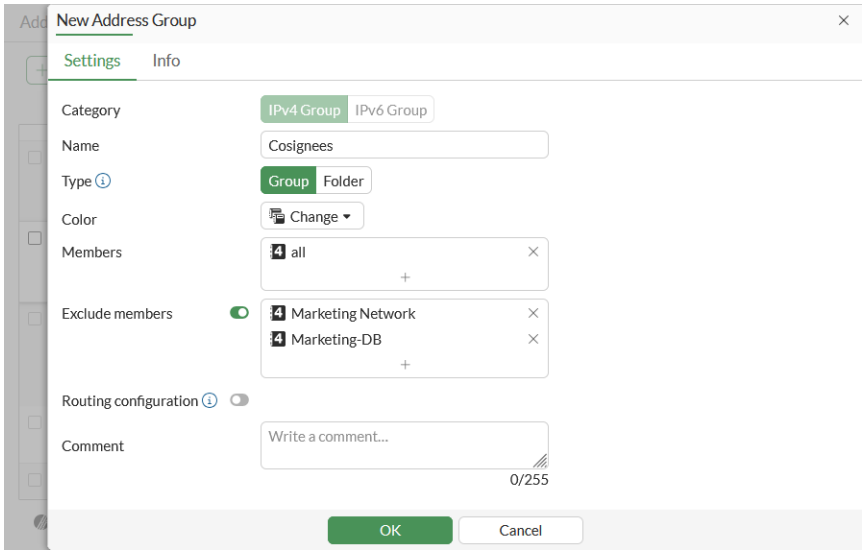
Address group exclusions

Specific IP addresses or ranges can be subtracted from the address group with the *Exclude Members* setting in IPv4 address groups.

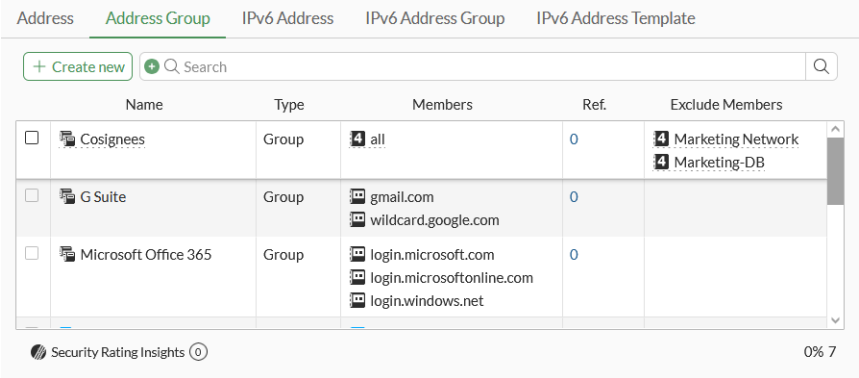
> This feature is only supported for IPv4 address groups, and only for addresses with a *Type* of *IP Range* or *Subnet*.

To exclude addresses from an address group using the GUI:

1. Go to *Policy & Objects* > *Addresses* and select *Address Group*.
2. Create a new address group, or edit an existing address group.
3. Enable *Exclude Members* and click the + to add entries.
4. Configure the other settings as needed.
5. Click *OK*.



The excluded members are listed in the *Exclude Members* column. Note that the *Exclude Members* column is not shown by default, see Column settings on page 58.



To exclude addresses from an address group using the CLI:

```
config firewall addrgrp
    edit <address group>
        set exclude enable
        set exclude-member <address> <address> ... <address>
    next
end
```

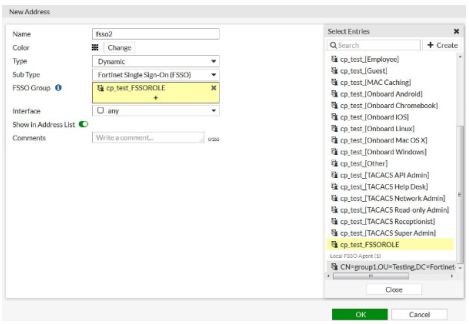
FSSO dynamic address subtype

The Fortinet Single Sign-ON (FSSO) dynamic firewall address subtype can be used in policies that support dynamic address types. The FortiGate will update the dynamic address used in firewall policies based on the source IP information for the authenticated FSSO users.

It can also be used with FSSO group information that is forwarded by ClearPass Policy Manager (CPPM) via FortiManager, and other FSSO groups provided by the FSSO collector agent or FortiNAC. Up to 3000 dynamic FSSO IP addresses are supported per dynamic FSSO group.

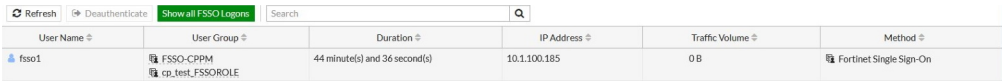
To configure FSSO dynamic addresses with CPPM and FortiManager in the GUI:

1. Create the dynamic address object:
   a. Go to *Policy & Objects* > *Addresses* and select *Address*.
   b. Click *Create new*.
   c. For *Type*, select *Dynamic*.
   d. For *Sub Type*, select *Fortinet Single Sign-On (FSSO)*.
   e. Select one or more groups.
   f. Click *OK* to save the configuration.

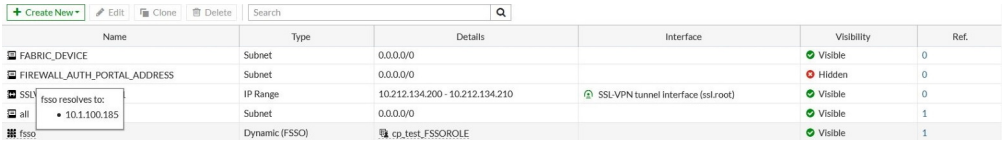


In the address table, there will be an error message for the address you just created (*Unresolved dynamic address: fsso*). This is expected because there are currently no authenticated FSSO users (based on source IP) in the local FSSO user list.

2. Add the dynamic address object to a firewall policy:
   a. Go to *Policy & Objects* > *Firewall Policy*.
   b. Create a new policy or edit an existing policy.
   c. For *Source*, select Address from the dropdown list and add the dynamic FSSO address object you just created.
   d. Configure the rest of the policy as needed.
   e. Click *OK* to save your changes.
3. Test the authentication to add a source IP address to the FSSO user list:
   a. Log in as user and use CPPM for user authentication to connect to an external web server. After successful authentication, CPPM forwards the user name, source IP address, and group membership to the FortiGate via FortiManager.
   b. Go to *Monitor* > *Firewall User Monitor* to view the user name (*fsso1*) and IP address.

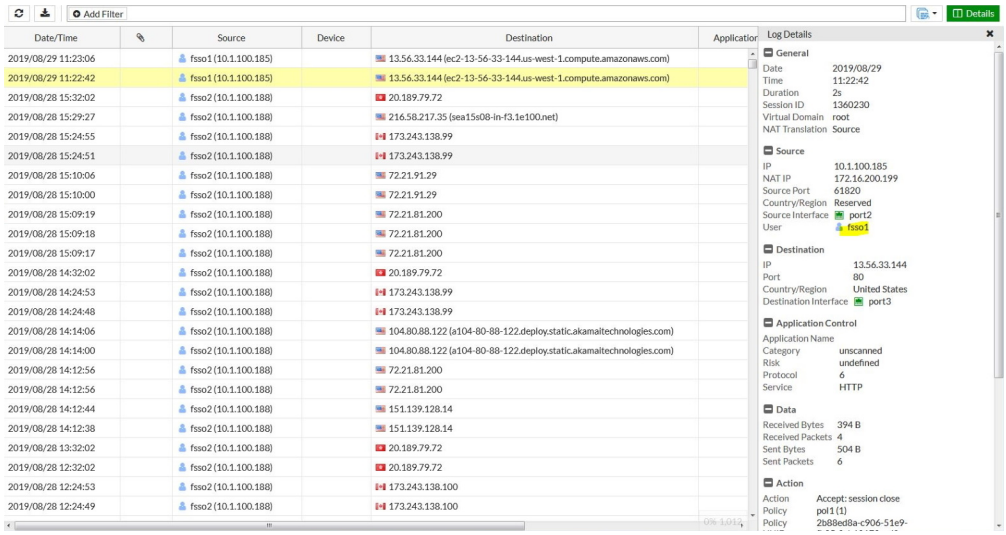


   c. Go to *Policy & Objects* > *Addresses* to view the updated address table. The error message no longer appears.
   d. Hover over the dynamic FSSO address to view the IP address (*fsso resolves to: 10.1.100.185*).

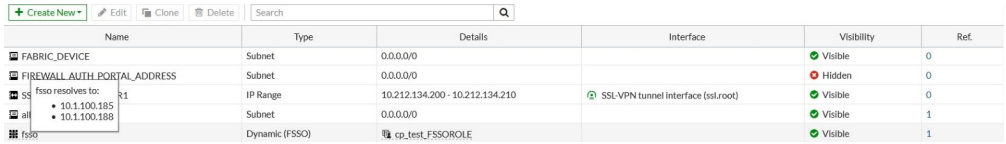


To verify user traffic in the GUI:

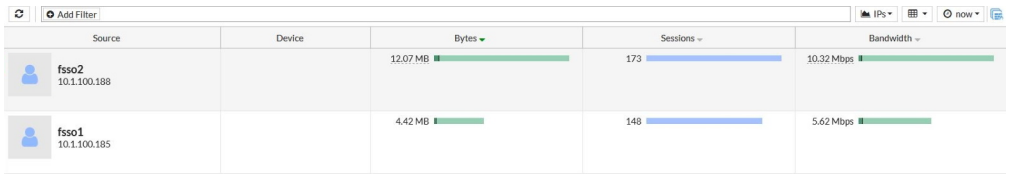
1. Go to *Log & Report* > *Forward Traffic*.
   Details for the user *fsso1* are visible in the traffic log:



- If another user is authenticated by CPPM, then the dynamic address *fsso* entry in the address table will be updated. The IP address for user *fsso2* (*10.1.100.188*) is now visible:



2. Go to *FortiView* > *Sources* to verify that the users were able to successfully pass the firewall policy.



If a user logs off and CPPM receives log off confirmation, then CPPS updates the FortiGate FSSO user list via FortiManager. The user IP address is deleted from the dynamic FSSO address, and the user is no longer be able to pass the firewall policy.

To configure FSSO dynamic addresses with CPPM and FortiManager in the CLI:

1. Create the dynamic address object:

```
config firewall address
    edit "fsso"
        set type dynamic
        set sub-type fsso
        set fsso-group "cp_test_FSSOROLE"
    next
end
```

2. Add the dynamic address object to a policy:

```
config firewall policy
    edit 1
        set name "pol1"
        set srcintf "port2"
        set dstintf "port3"
        set srcaddr "fsso"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set fsso disable
        set nat enable
    next
end
```

To verify user traffic in the CLI:

1. Check the FSSO user list:

```
diagnose debug authd fsso list
----FSSO logons----
IP: 10.1.100.185  User: fsso1  Groups: cp_test_FSSOROLE  Workstation:  MemberOf: FSSO-CPPM cp_
test_FSSOROLE
Total number of logons listed: 1, filtered: 0
----end of FSSO logons----
```

2. Check the authenticated firewall users list:

```
diagnose firewall auth list
10.1.100.185, fsso1
type: fsso, id: 0, duration: 2928, idled: 2928
server: FortiManager
packets: in 0 out 0, bytes: in 0 out 0
group_id: 2 33554433
group_name: FSSO-CPPM cp_test_FSSOROLE
----- 1 listed, 0 filtered ------
```

After user traffic passes through the firewall, the nu

```
diagnose firewall auth list
10.1.100.185, fsso1
type: fsso, id: 0, duration: 3802, idled: 143
server: FortiManager
packets: in 1629 out 1817, bytes: in 2203319 out 133312
group_id: 2 33554433
group_name: FSSO-CPPM cp_test_FSSOROLE
----- 1 listed, 0 filtered ------
```

ClearPass integration for dynamic address objects

ClearPass Policy Manager (CPPM) can gather information about the statuses of network hosts, for example, the latest patches or virus infections. Based on this information, CPPM send the IP addresses and current states, such as Healthy or Infected, to the FortiGate.

On the FortiGate, the IP addresses received from CPPM are added to a dynamic firewall address with the *clearpass-spt* subtype. This address can be used in any policy that supports dynamic addresses, such as Firewall.

In this example, you create two dynamic IP addresses that are used in two firewall policies (deny and allow). One policy allows traffic (host state = Healthy), and the other denies traffic (host state = Infected). When CPPM sends the information, the IP addresses are assigned according to their host state: Healthy or Infected.

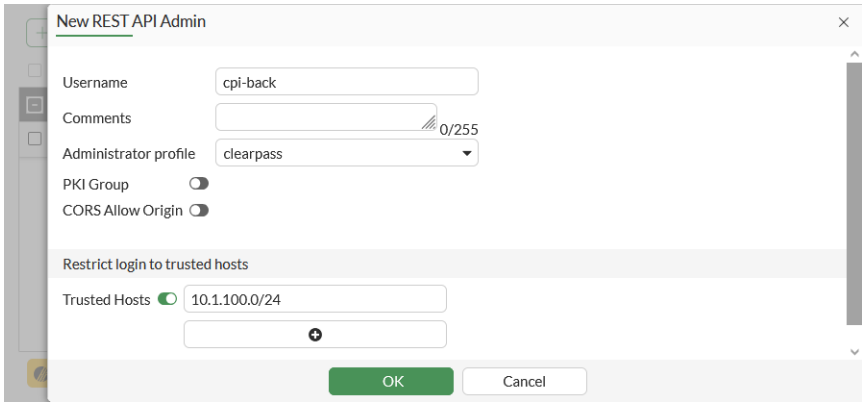
You can then verify that traffic from the Infected host is denied access by the deny policy, and traffic from the Healthy host is allowed access by the allow policy.

Create a REST API administrator

A REST API administrator is required to generate an authorization token for REST API messages, and to limit hosts that can send REST API messages to the FortiGate.

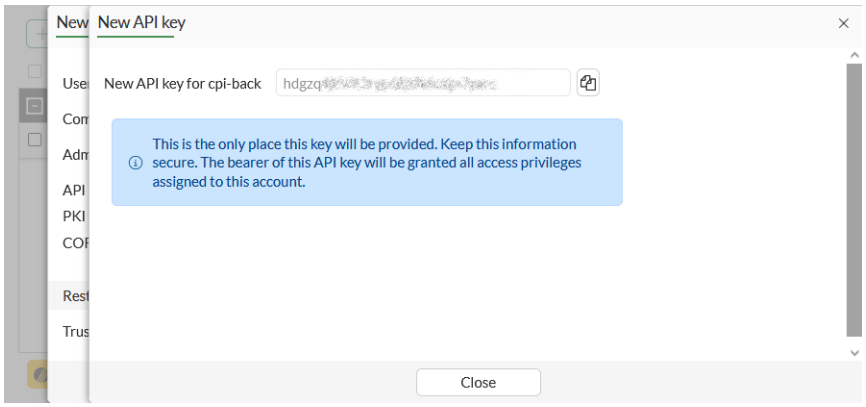
To create a REST API administrator in the GUI:

1. Go to *System* > *Administrators*.
2. Click *Create New* > *REST API Admin*.
3. Configure the *Username* and other information as needed.
4. Disable *PKI Group*.
5. In the *Trusted Hosts* field, enter *10.1.100.0/24*.



For this example, an administrator profile called *clearpass* was created with full read/write access. See Administrator profiles on page 3736 for details.

6. Click *OK*.
The *New API key* pane opens.



The API key is the REST API authorization token that is used in REST API messages sent by CPPM to the FortiGate.

7. Copy the API key to a secure location. A new key can be generated if this one is lost or compromised.
8. Click *Close*.

To create a REST API administrator in the CLI:

```
config system api-user
    edit "cpi-back"
        set accprofile "clearpass"
        config trusthost
            edit 1
                set ipv4-trusthost 10.1.100.0 255.255.255.0
            next
        end
    next
end
```

```
execute api-user generate-key cp-api
    New API key: 0f1HxGHh9r9p74k7qgfHNH40p51bjs
    NOTE: The bearer of this API key will be granted all access privileges assigned to the api-
user cp-api.
```

Create dynamic IP addresses with the clearpass subtype

Two dynamic IP addresses are required, one for the allow policy, and the other for the deny policy.

To create the dynamic IP addresses:

```
config firewall address
    edit "cppm"
        set type dynamic
        set sub-type clearpass-spt
        set clearpass-spt healthy
        set comment ''
        set visibility enable
```

```
        set associated-interface ''
        set color 0
    next
    edit "cppm-deny"
        set type dynamic
        set sub-type clearpass-spt
        set clearpass-spt infected
        set comment ''
        set visibility enable
        set associated-interface ''
        set color 0
    next
end
```

Create firewall policies

Two firewall policies are required, one to accept traffic (*cppm-allow*), and the other to deny traffic (*cppm-deny*).

To create the firewall policies in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Configure the allow policy:
   a. Click *Create New*.
   b. Enter a name for the policy.
   c. Set *Source* set to *cppm*.
   d. Set *Action* to *ACCEPT*.
   e. Configure the remaining settings as needed.
   f. Click *OK*.
3. Configure the deny policy:
   a. Click *Create New*.
   b. Enter a name for the policy.
   c. Set *Source* set to *cppm-deny*.
   d. Set *Action* to *DENY*.
   e. Configure the remaining settings as needed.
   f. Click *OK*.

To create the firewall policies in the CLI:

```
config firewall address
    edit "cppm"
        set type dynamic
        set sub-type clearpass-spt
        set clearpass-spt healthy
        set comment ''
        set visibility enable
        set associated-interface ''
```

```
        set color 0
    next
    edit "cppm-deny"
        set type dynamic
        set sub-type clearpass-spt
        set clearpass-spt infected
        set comment ''
        set visibility enable
        set associated-interface ''
        set color 0
    next
end
```

Verification

Go to *Log & Report* > *Forward Traffic* to review traffic logs and ensure that traffic is allowed or denied as expected.

To verify that FortiGate addresses are assigned correctly, enter the following:

```
# diagnose firewall dynamic list
List all dynamic addresses:
cppm-deny: ID(141)
        ADDR(10.1.100.188)

cppm: ID(176)
        ADDR(10.1.100.185)
        ADDR(10.1.100.186)
```

FortiNAC tag dynamic address

The FortiNAC tag dynamic firewall address type is used to store the device IP, FortiNAC firewall tags, and FortiNAC group information sent from FortiNAC by the REST API when user logon and logoff events are registered.

For more information see Dynamic Address Tag in the FortiNAC-F Security Fabric SSO guide.

FortiVoice tag dynamic address

When a FortiVoice-supplied MAC or IP address is used in a firewall policy, a FortiVoice tag (MAC/IP) dynamic address is automatically created on the FortiGate that contains all the provisioned FortiFones registered with FortiVoice. The dynamic address can be used in firewall policies to restrict rules to authorized FortiFones only. This is useful for large voice deployments that require security and efficiency. See Example of a firewall policy on page 2069.

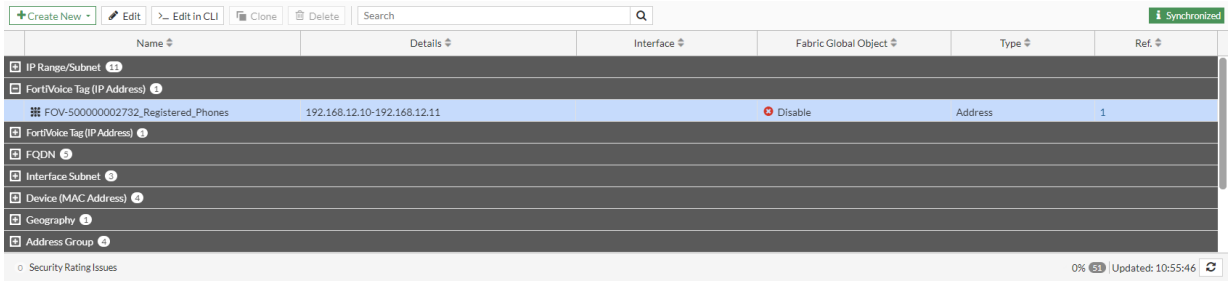
FortiVoice tag dynamic addresses can also be applied to a NAC policy. See Example of a NAC policy on page 2070.

Example of a firewall policy

In this example, two FortiFones are registered to FortiVoice and are assigned names and extension numbers. A FortiVoice Fabric connector has been authorized to join the Security Fabric. The dynamic FortiVoice tags are applied to a firewall policy.

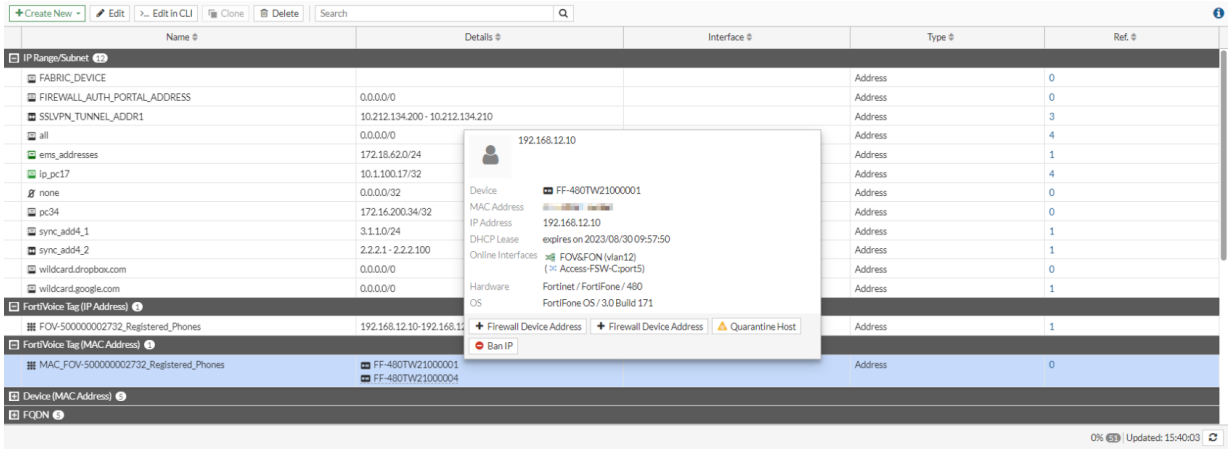
To use a FortiVoice tag dynamic firewall address in a policy:

1. Configure and authorize the FortiVoice Fabric connector (see Configuring FortiVoice on page 4389 for more information).
2. Go to *Policy & Objects* > *Addresses* to view the newly created dynamic firewall address objects:
   a. Expand the *FortiVoice Tag (IP Address)* section.



   There is one entry, *FOV-500000002732_Registered_Phones*, which matches 192.168.12.10 to 192.168.12.11.

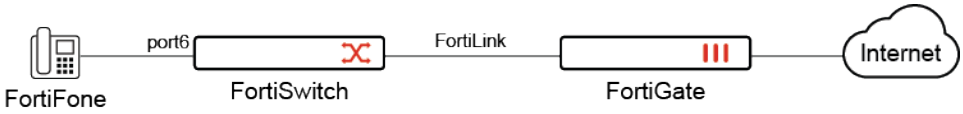
   b. Expand the *FortiVoice Tag (MAC Address)* section. There is one entry, *MAC_FOV-500000002732_Registered_Phones*, which matches two devices. Hover over the device serial number to view the tooltip that contains the MAC address and additional information.



3. Go to *Policy & Objects* > *Firewall Policy* and click *Create new* or edit an existing policy.
4. In the *Source* field, click the *+* and add the *FOV-500000002732_Registered_Phones* and *MAC_FOV-500000002732_Registered_Phones* addresses.
5. In the *Destination* field, click the *+* and add the *FOV-500000002732_Registered_Phones* address.
6. Configure the other settings as needed.
7. Click *OK*.

Example of a NAC policy

In this example, a dynamic FortiVoice tag MAC address (MAC_FOV-500000003139_Registered_Phones) is applied to a NAC policy on the FortiGate. Subsequently, the connected FortiSwitch port is moved to vlan12, where traffic can be controlled for registered FortiFones. For more information about NAC policies, see Defining a FortiSwitch NAC policy in the FortiLink Administration Guide. This example assumes that the FortiVoice Fabric connector is authorized to join the Security Fabric and vlan12 is already configured. See Configuring FortiVoice on page 4389 for more information.



To configure FortiVoice Tag MAC address on NAC policies:

1. Configure the NAC policy:
   a. Go to *WiFi & Switch Controller* > *NAC Policies* and click *Create New*, or edit an existing policy.
   b. In the *Device Patterns* section:
      - Set *Category* to *FortiVoice tag*.
      - Set *FortiVoice tag* to *MAC_FOV-500000003139_Registered_Phones*.
   c. In the *Switch Controller Action* section, enable *Assign VLAN* and select *vlan12*.
   d. Configure the other settings as needed.
   e. Click *OK*.
2. Enable NAC mode on the desired FortiSwitch ports (port6 in this example):
   a. Go to *WiFi & Switch Controller* > *FortiSwitch Ports*.
   b. Select *port6*, then right-click and set the *Mode* to *NAC*.
3. Configure firewall policy that is used to control outbound internet access for FortiFones (vlan12 to wan1):
   a. Go to *Policy & Objects* > *Firewall Policy*.
   b. Click *Create New*.
   c. Name the policy and configure the following parameters:

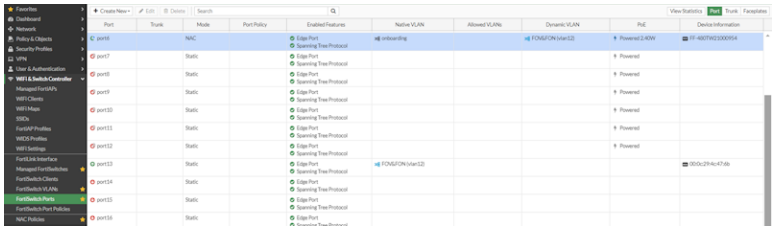
Incoming Interface	vlan12
Outgoing Interface	wan1
Source	all
Destination	all
Schedule	always
Service	ALL
Action	ACCEPT

   d. Configure the other settings as needed.
   e. Click *OK*.
4. Generate traffic from the FortiFone.

5. Once the NAC policy is matched, go to *WiFi & Switch Controller > NAC Policies* to view the device matched to the policy.

FortiFone is also shown on *Dashboards > Assets & Identities* in the *Matched NAC Devices* widget.

6. Go to *WiFi & Switch Controller > FortiSwitch Ports* and locate the port that the FortiFone is connected to. The port has been dynamically assigned vlan12.



To configure FortiVoice Tag MAC address on NAC policies in the CLI:

1. Configure the NAC policy:

```
config user nac-policy
    edit "nac-policy-1"
        set category fortivoice-tag
        set fortivoice-tag "MAC_FOV-500000003139_Registered_Phones"
        set switch-fortilink "fortilink"
        set switch-mac-policy "mac-policy-
    next
end
```

2. Configure the VLAN in the MAC policy:

```
config switch-controller mac-policy
    edit "mac-policy-1"
        set fortilink "fortilink"
        set vlan "vlan12"
    next
end
```

3. Enable NAC mode on the desired FortiSwitch ports:

```
config switch-controller managed-switch
    edit "Access-FSW-C”
        config ports
            edit "port6"
                set access-mode nac
            next
```

```
        end
    next
end
```

4. Configure the firewall policy:

```
config firewall policy
    edit 1
        set name "c_fov_fon"
        set srcintf "vlan12"
        set dstintf "wan1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set nat enable
    next
end
```

MAC address-based policies

MAC addresses can be added to the following IPv4 policies:

- Firewall
- Virtual wire pair
- ACL
- Central SNAT
- DoS

A MAC address is a link layer-based address type and it cannot be forwarded across different IP segments. In FortiOS, you can configure a firewall address object with a singular MAC, wildcard MAC, multiple MACs, or a MAC range.

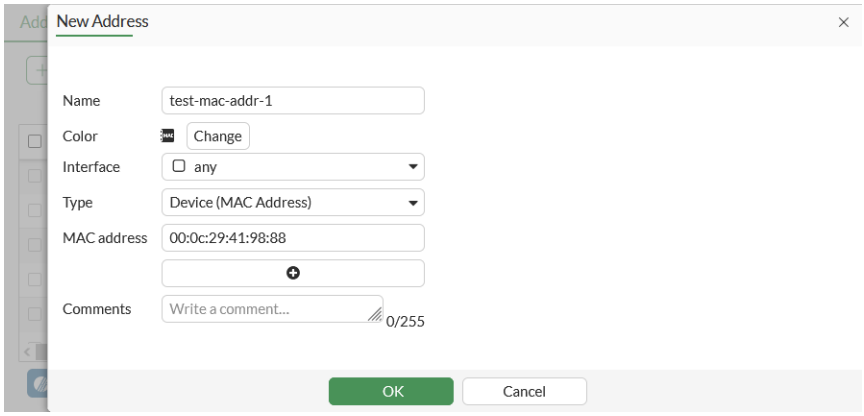
FortiOS only supports the MAC address type as source address for policies in NAT mode VDOM. When you use the MAC address type in a policy as source address in NAT mode VDOM, IP address translation (NAT) is still performed according to the rules defined in the policy. The MAC address type only works for source address matching. It does not have any association with NAT actions.

For policies in transparent mode or the virtual wire pair interface, you can use the MAC address type as source or destination address.

To configure a MAC address using the GUI:

1. Go to *Policy & Objects* > *Addresses* and select *Address*.
2. Click *Create new*.
3. Enter a name.
4. For *Category*, select *Address*.
5. For *Type*, select *Device (MAC Address)*.

6. Enter the MAC address.



7. Click *OK*.
8. Go to *Policy & Objects* > *Firewall Policy* to apply the address type to a policy in NAT mode VDOM:
   a. For *Source*, select the MAC address you just configured.
   b. For *Destination*, select an address.

   In NAT mode VDOM, this address type cannot be used as destination address.

   c. Configure the other settings as needed.
   d. Click *OK*.

To configure a MAC address using the CLI:

1. Create a new MAC address:

```
config firewall address
    edit "test-mac-addr-1"
        set type mac
        set macaddr 00:0c:29:41:98:88
    next
end
```

2. Apply the address type to a policy. In transparent mode or the virtual wire pair interface, this address type can be mixed with other address types in the policy:

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "port1"
        set srcaddr "test-mac-addr-1" "10-1-100-42"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set nat enable
```

```
    next
end
```

Device & OS Identification dynamic address subtype

Another type of MAC address object is the dynamic *Device & OS identification* subtype. This firewall address subtype is an advanced feature that can be used in policies that support dynamic address subtypes, and it relies on device detection configured on the interface connected to user devices to determine device information.

The FortiGate will update the dynamic address used in firewall policies based on the MAC address and other device and OS information for devices matching configured criteria. The criteria could be hardware vendor, hardware model, software OS, software version, or a combination of these parameters.

> Only existing devices whose device information has already been detected by the FortiGate and is known can be added to this dynamic address subtype.

Similar to MAC address-based objects, the dynamic address subtype can be used as a source address for firewall policies, proxy policies, and ZTNA rules. The dynamic address subtype can be used as a source or destination address for transparent mode policies or a virtual wire pair policy.

To use the dynamic *Device & OS Identification* subtype, go to *System > Feature Visibility* and enable *Dynamic Device & OS Identification*. Once enabled, the dynamic address subtype can be configured on the *Policy & Objects > Addresses* page.

ISDB well-known MAC address list

The Internet Service Database (ISDB) includes well-known vendor MAC address range lists. The lists can only be used for source MAC addresses in IPv4 policies, and include the vendor name and the MAC address ranges that the vendor belongs to.

To view the vendor list:

```
# diagnose vendor-mac id
Please input Vendor MAC ID.
ID: 1 name: "Asus"
ID: 2 name: "Acer"
ID: 3 name: "Amazon"
ID: 4 name: "Apple"
ID: 5 name: "Xiaomi"
ID: 6 name: "BlackBerry"
ID: 7 name: "Canon"
ID: 8 name: "Cisco"
ID: 9 name: "Linksys"
ID: 10 name: "D-Link"
ID: 11 name: "Dell"
ID: 12 name: "Ericsson"
```

```
ID: 13 name: "LG"
ID: 14 name: "Fujitsu"
ID: 15 name: "Fitbit"
ID: 16 name: "Fortinet"
ID: 17 name: "OPPO"
ID: 18 name: "Hitachi"
ID: 19 name: "HTC"
ID: 20 name: "Huawei"
ID: 21 name: "HP"
ID: 22 name: "IBM"
ID: 23 name: "Juniper"
ID: 24 name: "Lenovo"
ID: 25 name: "Microsoft"
ID: 26 name: "Motorola"
ID: 27 name: "Netgear"
ID: 28 name: "Nokia"
ID: 29 name: "Nintendo"
ID: 30 name: "PaloAltoNetworks"
ID: 31 name: "Polycom"
ID: 32 name: "Samsung"
ID: 33 name: "Sharp"
ID: 34 name: "Sony"
ID: 35 name: "Toshiba"
ID: 36 name: "VMware"
ID: 37 name: "Vivo"
ID: 38 name: "Zyxel"
ID: 39 name: "ZTE"
```

To view the MAC address ranges for a vendor:

```
# diagnose vendor-mac id 16
Vendor MAC: 16(Fortinet)
Version: 0000700021
Timestamp: 201908081432
Number of MAC ranges: 6
00:09:0f:00:00:00 - 00:09:0f:ff:ff:ff
04:d5:90:00:00:00 - 04:d5:90:ff:ff:ff
08:5b:0e:00:00:00 - 08:5b:0e:ff:ff:ff
70:4c:a5:00:00:00 - 70:4c:a5:ff:ff:ff
90:6c:ac:00:00:00 - 90:6c:ac:ff:ff:ff
e8:1c:ba:00:00:00 - e8:1c:ba:ff:ff:ff
```

To query the vendor of a specific MAC address or range:

```
# diagnose vendor-mac match 00:09:0f:ff:ff:ff 48
Vendor MAC: 16(Fortinet), matched num: 1
```

To use the vendor ID in a firewall policy:

```
config firewall policy
    edit 9
        set name "policy_id_9"
        set uuid 6150cf30-308d-51e9-a7a3-bcbd05d61f93
        set srcintf "wan2"
        set dstintf "wan1"
        set srcaddr "all"
        set dstaddr "all"
        set vendor-mac 36 16
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set auto-asic-offload disable
        set nat enable
    next
end
```

Only packets whose source MAC address belong to Fortinet or VMware are passed by the policy.

IPv6 MAC addresses and usage in firewall policies

Users can define IPv6 MAC addresses that can be applied to the following policies:

- Firewall
- Virtual wire pair
- ACL/DoS
- Central NAT
- NAT64
- Local-in

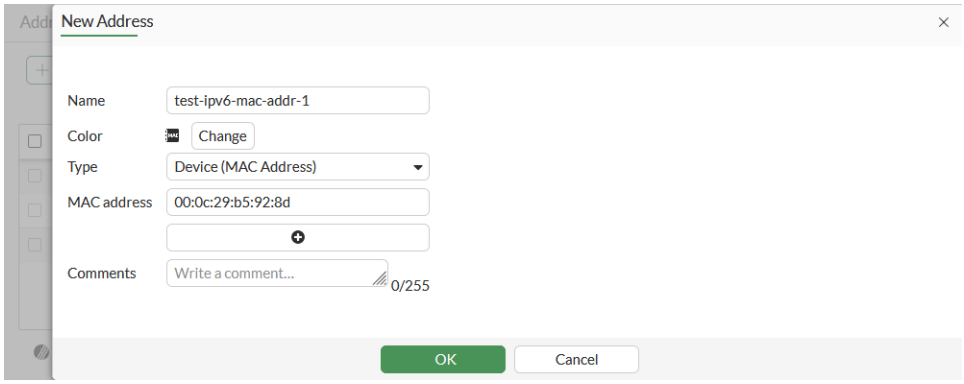
In FortiOS, you can configure a firewall address object with a singular MAC, wildcard MAC, multiple MACs, or a MAC range. In this example, a firewall policy is configured in a NAT mode VDOM with the IPv6 MAC address as a source address.

IPv6 MAC addresses cannot be used as destination addresses in VDOMs when in NAT operation mode.

To configure IPv6 MAC addresses in a policy in the GUI:

1. Create the MAC address:
   a. Go to *Policy & Objects* > *Addresses* and select *IPv6 Address*.
   b. Click *Create New*.
   c. Enter an address name.
   d. For *Type*, select *Device (MAC Address)*.

e. Enter the the MAC address.



f. Click *OK*.

2. Configure the policy:
   a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
   b. For *Source*, select the IPv6 MAC address object.
   c. Configure the other settings as needed.
   d. Click *OK*.

To configure IPv6 MAC addresses in a policy in the CLI:

1. Create the MAC address:

```
config firewall address6
    edit "test-ipv6-mac-addr-1"
        set type mac
        set macaddr 00:0c:29:b5:92:8d
    next
end
```

2. Configure the policy:

```
config firewall policy
    edit 2
        set srcintf "wan2"
        set dstintf "wan1"
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "test-ipv6-mac-addr-1" "2000-10-1-100-0"
        set dstaddr6 "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set auto-asic-offload disable
        set nat enable
    next
end
```

RSSO dynamic address subtype

The RSSO dynamic address object subtype can be used in a firewall policy's source and destination fields. It allows for more granular and precise policies based on RSSO group membership, enhancing security and flexibility when managing network traffic and enforcing policies.

When the sub-type is `rsso`, the `sso-attribute-value` must be set. The IP address of the RADIUS single sign-on user matching the group value will be loaded to the address object.

To configure the RSSO dynamic address subtype:

```
config firewall address
    edit "test-rsso-addr-1"
        set type dynamic
        set sub-type rsso
        set sso-attribute-value <name(s)>
    next
end
```

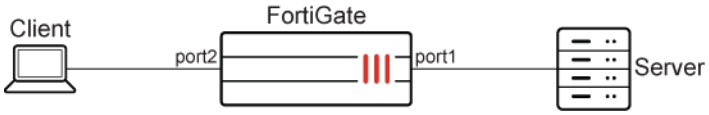
Variable	Description
`sub-type rsso`	RSSO address sub-type.
`sso-attribute-value <name (s)>`	RADIUS attributes value.

To check the RADIUS dynamic address database information :

```
# diagnose test application radiusd {6 | 66}
```

Test level	Description
6	Show RADIUS dynamic address database summary information.
66	Show RADIUS dynamic address database information.

Example



To configure and use an RSSO dynamic address object:

1. Enable RADIUS account access on port 1.
   When the RADIUS server sends an RSSO message to the FortiGate on port 1, which includes an IP address, the FortiGate will add it to the RSSO dynamic address list.

```
config system interface
    edit port1
        append allowaccess radius-acct
    next
end
```

2. Configure the RADIUS user and user group for the RSSO address:

```
config user radius
    edit "rsso_server"
        set rsso enable
        set rsso-radius-response enable
        set rsso-secret **************
        set rsso-flush-ip-session enable
    next
end
config user group
    edit "rsso_g1"
        set group-type rsso
        set sso-attribute-value "rsso_group_1"
    next
end
```

3. Configure a dynamic address with RSSO subtype:

```
config firewall address
    edit "test-rsso-addr-1"
        set type dynamic
        set sub-type rsso
        set sso-attribute-value "rsso_group_1"
    next
end
```

4. Apply the RSSO dynamic address as a destination in the firewall policy:

```
config firewall policy
    edit 44
        set name "nat44_policy"
        set srcintf "port2"
        set dstintf "port1"
        set action accept
        set srcaddr "10-1-100-0"
        set dstaddr "test-rsso-addr-1"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set auto-asic-offload disable
        set nat enable
    next
end
```

5. Check the RSSO dynamic address. In this case, 172.16.200.155 is loaded into the RSSO dynamic address:

```
# diagnose test application radiusd 6
dynamic addresses total[vd:root]:0.
dynamic addresses total[vd:vdom1]:1.
  name, ip_db-total
  test-rsso-addr-1, 1
```

```
# diagnose test application radiusd 66
dynamic addresses total[vd:root]:0.
dynamic addresses total[vd:vdom1]:1.
  name:test-rsso-addr-1, ip_db total:1.
        ip,     installed
        172.16.200.155, 1.
```

```
# diagnose firewall dynamic list test-rsso-addr-1
CMDB name: test-rsso-addr-1
test-rsso-addr-1: ID(90)
        ADDR(172.16.200.155)
Total IP dynamic range blocks: 0.
Total IP dynamic addresses: 1.
```

6. Send a packet that hits the policy, then check the session to see that the RSSO dynamic address works as a destination in the firewall policy:

```
# diagnose sys session list
session info: proto=6 proto_state=07 duration=6 expire=115 timeout=3600 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
dst_user=dynamic_user_1 dst_authsvr=vdom1 state=log may_dirty f00
statistic(bytes/packets/allow_err): org=269/5/1 reply=715/4/1 tuples=2
tx speed(Bps/kbps): 42/0 rx speed(Bps/kbps): 112/0
orgin->sink: org pre->post, reply pre->post dev=10->9/9->10 gwy=172.16.200.155/10.1.100.42
hook=post dir=org act=snat 10.1.100.42:20042->172.16.200.155:80(172.16.200.6:20042)
hook=pre dir=reply act=dnat 172.16.200.155:80->172.16.200.6:20042(10.1.100.42:20042)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=44 pol_uuid_idx=2522 auth_info=0 chk_client_info=0 vd=1
serial=0000254c tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
total session: 1
```

ISDB record for SOCaaS

The *Fortinet-FortiGuard.SOCaaS* Internet service database (ISDB) entry for Fortinet SOCaaS enables policies to be configured for devices to forward data to SOCaaS collectors without relying on DNS. Eliminating the dependency on DNS reduces the risk of DNS mapping failures and helps ensure a more reliable and seamless data forwarding processing.

To check that the new entry is added to the ISDB:

1. Update the object versions:

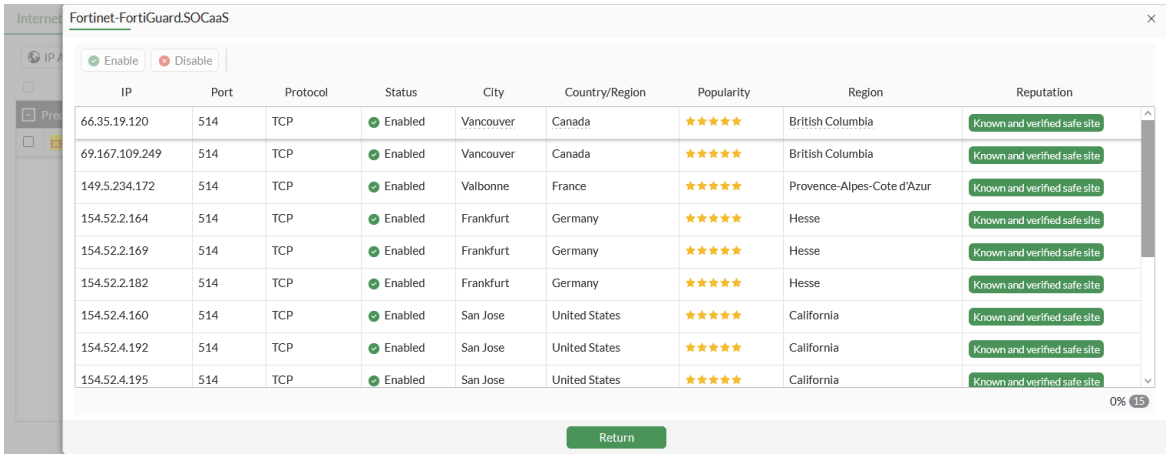
```
# diagnose autoupdate versions

Internet-service Full Database
---------
Version: 7.03917 signed
Contract Expiry Date: n/a
Last Updated using manual update on Wed Oct 30 10:33:22 2024
Last Update Attempt: Wed Oct 30 10:33:22 2024
Result: Updates Installed
```

2. Check the Internet service database for the entry:

```
# diagnose internet-service id 1245514

Internet Service: 1245514(Fortinet-FortiGuard.SOCaaS)
Version: 00007.03917
Timestamp: 202410291605
Number of Entries: 14
66.35.19.120-66.35.19.120 country(840) region(283) city(23352) blocklist(0x0) reputation(5),
popularity(5) domain(5225) botnet(0) proto(6) port(514)
149.5.234.172-149.5.234.172 country(250) region(1509) city(24955) blocklist(0x0) reputation
(5), popularity(5) domain(0) botnet(0) proto(6) port(514)
154.52.2.164-154.52.2.164 country(276) region(697) city(7844) blocklist(0x0) reputation(5),
popularity(5) domain(0) botnet(0) proto(6) port(514)
154.52.2.169-154.52.2.169 country(276) region(697) city(7844) blocklist(0x0) reputation(5),
popularity(5) domain(0) botnet(0) proto(6) port(514)
154.52.2.182-154.52.2.182 country(276) region(697) city(7844) blocklist(0x0) reputation(5),
popularity(5) domain(0) botnet(0) proto(6) port(514)
154.52.4.160-154.52.4.160 country(840) region(283) city(21092) blocklist(0x0) reputation(5),
popularity(5) domain(0) botnet(0) proto(6) port(514)
154.52.4.192-154.52.4.192 country(840) region(283) city(21092) blocklist(0x0) reputation(5),
popularity(5) domain(0) botnet(0) proto(6) port(514)
154.52.4.195-154.52.4.195 country(840) region(283) city(21092) blocklist(0x0) reputation(5),
popularity(5) domain(0) botnet(0) proto(6) port(514)
154.52.4.197-154.52.4.197 country(840) region(283) city(21092) blocklist(0x0) reputation(5),
popularity(5) domain(0) botnet(0) proto(6) port(514)
154.52.4.204-154.52.4.204 country(840) region(283) city(21092) blocklist(0x0) reputation(5),
popularity(5) domain(0) botnet(0) proto(6) port(514)
154.52.4.224-154.52.4.224 country(840) region(283) city(21092) blocklist(0x0) reputation(5),
popularity(5) domain(5225) botnet(0) proto(6) port(514)
154.52.6.181-154.52.6.181 country(840) region(2039) city(1106) blocklist(0x0) reputation(5),
popularity(5) domain(0) botnet(0) proto(6) port(514)
154.52.22.161-154.52.22.161 country(36) region(1287) city(23487) blocklist(0x0) reputation(5),
popularity(5) domain(0) botnet(0) proto(6) port(514)
154.52.29.66-154.52.29.66 country(840) region(482) city(25915) blocklist(0x0) reputation(5),
popularity(5) domain(0) botnet(0) proto(6) port(514)
```

3. Check the entry in the GUI:

a. Go to *Policy & Objects* > *Internet Service Database* and select the *Internet Service* tab.
b. Search for *Fortinet-FortiGuard.SOCaaS*.
c. Hover over the result, and click *View/Edit Entries*.



To use the new entry in a firewall policy:

1. Configure a firewall policy:

```
config firewall policy
    edit 99
        set name "ISDB_Policy"
        set srcintf "wan2"
        set dstintf "wan1"
        set action accept
        set srcaddr "all"
        set internet-service enable
        set internet-service-name "Fortinet-FortiGuard.SOCaaS"
        set schedule "always"
        set logtraffic all
        set auto-asic-offload disable
        set nat enable
    next
end
```

2. Generate and then check a log generated by traffic hitting the policy:

```
1: date=2024-10-29 time=17:52:49 eventtime=1730249569310005321 tz="-0700" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="root" srcip=10.1.100.42 srcport=38380
srcintf="wan2" srcintfrole="lan" dstip=66.35.19.120 dstport=514 dstintf="wan1"
dstintfrole="undefined" srcuuid="70a3388c-dfec-51ea-f8dd-88267a721f36" srccountry="Reserved"
dstinetsvc="Fortinet-FortiGuard.SOCaaS" dstcountry="United States" dstregion="California"
dstcity="Sunnyvale" dstreputation=5 sessionid=474401 proto=6 action="close" policyid=99
policytype="policy" poluuid="ac8e35a2-dffd-51ea-9df6-e3860c663d3b" policyname="ISDB_Policy"
service="Fortinet-FortiGuard.SOCaaS" trandisp="snat" transip=172.16.200.10 transport=38380
appcat="unscanned" duration=2 sentbyte=300 rcvdbyte=172 sentpkt=5 rcvdpkt=4
mastersrcmac="00:0c:29:6e:3a:47" srcmac="00:0c:29:6e:3a:47" srcserver=0
```

Telemetry addresses and address groups

This topic applies only to a FortiGate acting as a FortiTelemetry Controller with *FortiTelemetry* enabled on the *System > Feature Visibility* page. For a summary of FortiTelemetry-related tasks that you can do in FortiOS, see FortiTelemetry on page 139. See also *FortiTelemetry Administration Guide*.

After a FortiTelemetry agent is authorized in FortiOS, a dynamic telemetry address is automatically created. The name of the telemetry address is the serial number of the FortiTelemetry agent. See Connecting to FortiTelemetry agents on page 4364 for more information.

A static telemetry address group named TELEMETRY is also available. The telemetry address from each approved FortiTelemetry agent is automatically added to the TELEMETRY group.

You can view telemetry addresses and address groups in the GUI or CLI.

In the GUI, go to the *Policy & Objects > Addresses* pane, and click the *Telemetry* tab to view telemetry addresses and telemetry address groups.

In the CLI, use the `config firewall address` command for telemetry addresses:

```
config firewall address
    edit <FortiTelemetry agent serial number>
        set type dynamic
        set sub-type {telemetry}
        set agent-id <FortiTelemetry agent serial number>
    next
end
```

Option	Description
`sub-type {telemetry}`	Dynamic telemetry address.
`agent-id <FortiTelemetry agent serial number>`	Serial number of the FortiTelemetry agent, for example, FT100GTK24000001.

Use the `config firewall addrgrp` command for telemetry address groups:

```
config firewall addrgrp
    edit "telemetry-group"
        set category telemetry
        set member FT100GTK24000001
    next
end
```

Option	Description
`category {telemetry}`	Members must be telemetry groups or telemetry addresses. Can be used to determine telemetry policy.

Use the `config telemetry-controller global` command to control whether telemetry addresses are automatically added to the TELEMETRY address group:

```
config telemetry-controller global
    set auto-group-telemetry-addr {enable | disable}
end
```

Option	Description
`auto-group-telemetry-addr {enable | disable}`	Enable/disable automatically adding the telemetry address to the default address group named TELEMETRY (default = enable). • `enable`: Automatically add telemetry address to the default addrgrp TELEMETRY. • `disable`: Do not automatically add telemetry address to the default addrgrp TELEMETRY.

In FortiOS 7.6.3, FortiTelemetry agents were represented as firewall addresses of type `ipmask`. The firewall address name started with the reserved prefixes "FT100G" or "FTLWIN" and included the agent's serial number. The telemetry address could not be renamed, and the firewall addresses were dynamically updated by telemetryd.

When upgrading from FortiOS 7.6.3 to 7.6.4 or later:

- The `ipmask` firewall address is automatically migrated to the new address type. Starting in FortiOS 7.6.4, telemetry firewall objects no longer require the "FT100G" or "FTLWIN" prefixes. Although the telemetry firewall objects can be renamed in the CLI, it is not recommended to rename the objects.
- All existing telemetry addresses are added to the TELEMETRY address group.

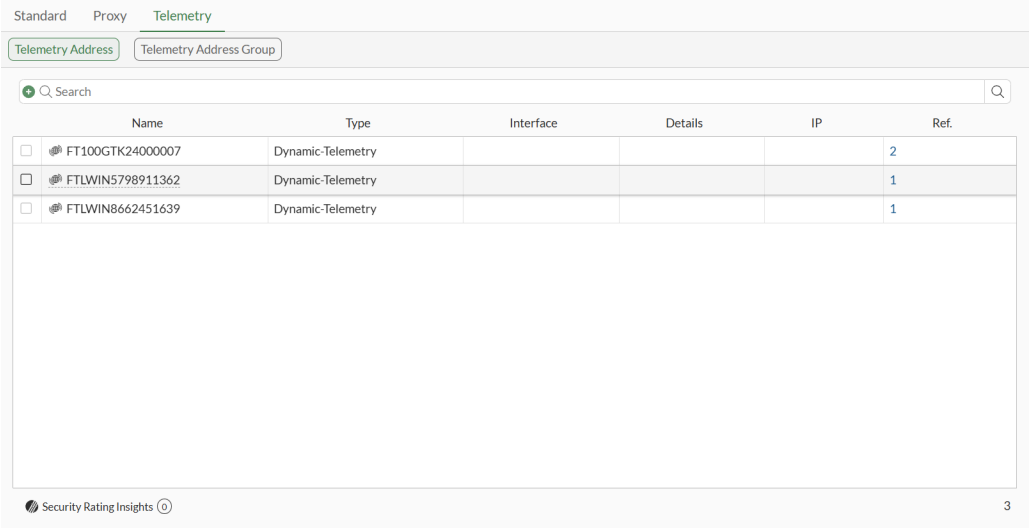
This topic includes the following examples:

- Viewing telemetry addresses on page 2084
- Adding telemetry addresses to TELEMETRY address group on page 2086
- Creating telemetry address groups on page 2087
- Viewing telemetry addresses before and after upgrade on page 2088

Viewing telemetry addresses

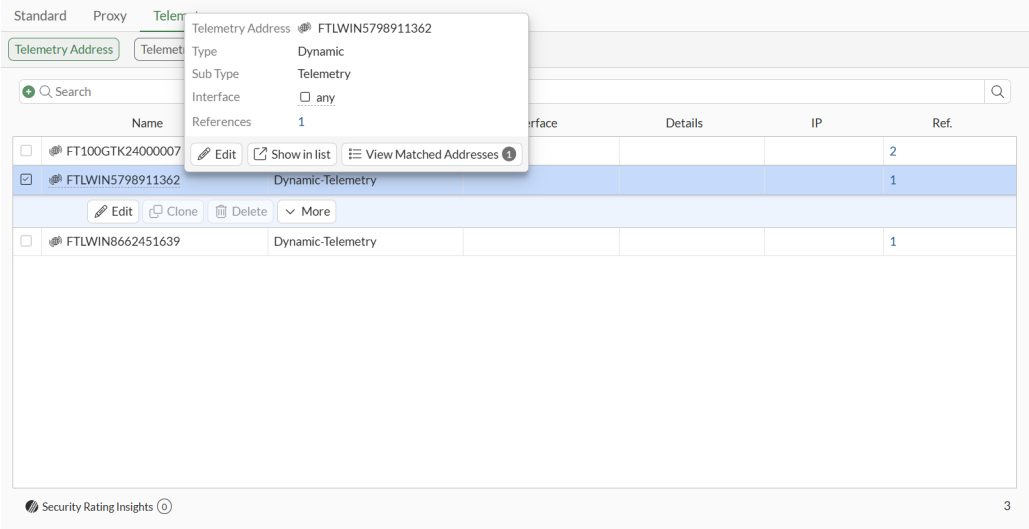
To view telemetry addresses in the GUI:

1. Go to *Policy & Objects* > *Addresses* > *Telemetry*. The *Telemetry Address* pane is displayed.
   This example lists three telemetry addresses.



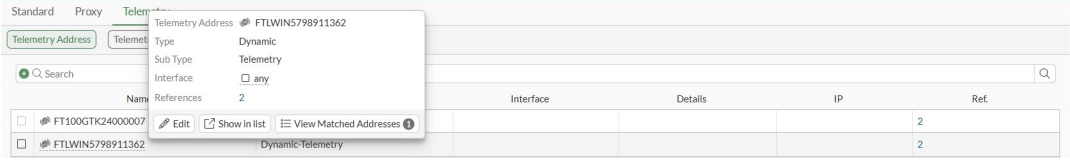
Name	Name of the address is the serial number of the FortiTelemetry agent. Hover over the name to display a tooltip of additional information and buttons, such as *Edit*, *Show in list*, and *View Matched Addresses*.
Type	Displays *Dynamic-Telemetry* to indicate a dynamic type of address with sub-type telemetry.
Ref.	Displays the number of references to the address. Click the number to view current usage details and possible uses.

2. Click a telemetry address to display a tooltip of additional information and more buttons.



Edit	Click to display the Edit Address pane and edit the address object.
More	Click to select *>_Edit in the CLI*.
View Matched Addresses	Click to view the resolved IP address for the FortiTelemetry agent.

3. Click the *View Matched Addresses* button to view the resolved IP address.



The *Resolved Addresses* pane is displayed and shows the IP address (*10.1.100.106*) of the FortiTelemetry agent.

To view a telemetry address in the CLI:

1. View the telemetry address.

   The `type` is `dynamic`, the `sub-type` is `telemetry`, and the `agent-id` is the serial number of the FortiTelemetry agent:

```
show full firewall address FTLWIN5798911362
config firewall address
    edit "FTLWIN5798911362"
        set uuid 8e6e691a-4ba6-51f0-5a93-9e03e58c5a19
        set type dynamic
        set sub-type telemetry
        set comment ''
        set associated-interface ''
        set color 0
        set fabric-object disable
        set agent-id "FTLWIN5798911362"
    next
end
```

2. View the IP address (`10.1.100.106`) of the FortiTelemetry agent.

```
# diagnose firewall dynamic address FTLWIN5798911362
CMDB name: FTLWIN5798911362
FTLWIN5798911362: ID(250)
        ADDR(10.1.100.106)
Total IP dynamic range blocks: 0.
Total IP dynamic addresses: 1.
```

Adding telemetry addresses to TELEMETRY address group

By default, FortiOS automatically adds the telemetry address from each approved FortiTelemetry agent to the default, static TELEMETRY group. Then you can select the TELEMETRY address group in a firewall policy to use all approved agents.

To automatically add telemetry addresses to the TELEMETRY address group:

```
config telemetry-controller global
    set auto-group-telemetry-addr enable
end
```

Creating telemetry address groups

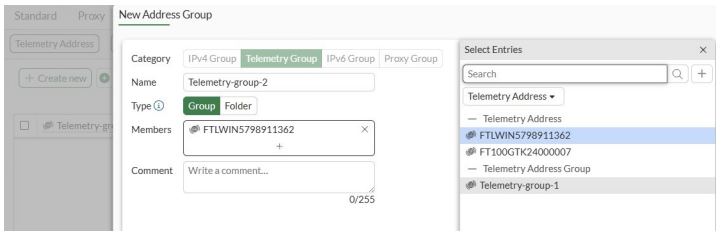
You can create telemetry address groups in the GUI or CLI.

To create a telemetry address group in the GUI:

1. Go to *Policy & Objects* > *Addresses* > *Telemetry* > *Telemetry Address Group*.

Create new	Click to create a new telemetry address group.
Name	Name of the telemetry address group. A default TELEMETRY address group is provided. Hover over the name to display a tooltip of additional information and buttons, such as *Edit* and *Show in list*. By default, FortiOS adds the telemetry address for each approved FortiTelemetry agent to the TELEMETRY address group.
Type	Displays *Group* to indicate a group of telemetry addresses.
Members	Lists the members of the group. Members must be telemetry addresses or telemetry address groups.
Ref.	Displays the number of references to the address group. Click the number to view current usage details and possible uses.

2. Click the *Create new* button. The *New Address Group* pane is displayed.
3. Set the following options and click *OK*.
   - Set *Name* to a unique name.
   - Set *Type* to *Group*.
   - Click *Members*, select *Telemetry Address*, and select one or more telemetry addresses or telemetry address groups. Click *Close* when done.



The telemetry group contains the selected telemetry addresses.

4. Click *OK* to save the new address group.

To create a telemetry address group in the CLI:

1. Create a telemetry address group.
   The example `Telemetry-group-1` is for a hardware agent. The `type` can be `default` or `folder`, and the `category` is `telemetry`. The member is the `FT100GTK24000007` telemetry address:

```
config firewall addrgrp
    edit "Telemetry-group-1"
        set type default
        set category telemetry
        set member "FT100GTK24000007"
        set comment ''
        set uuid 4f197cee-4d65-51f0-2942-8214af708a17
        set exclude disable
        set color 0
        set fabric-object disable
    next
end
```

2. Create a new telemetry address group.
   The example `Telemetry-group-2` is for a software agent. The `category` is `telemetry`, and the member is the `FTLWIN5798911362` telemetry address:

```
config firewall addrgrp
    edit Telemetry-group-2
        set category telemetry
        set member FTLWIN5798911362
    next
end
```

Viewing telemetry addresses before and after upgrade

During upgrade from FortiOS 7.6.3 to 7.6.4 or later, the `ipmask` telemetry address is migrated to the new telemetry type.

To view the telemetry address:

1. In FortiOS 7.6.3, view the configuration before upgrading to 7.6.4:

```
show firewall address
config firewall address
    edit "FTLWIN5798911362"
        set uuid adfe7cd2-4d6f-51f0-5f30-08450239a4e7
        set subnet 10.1.100.106 255.255.255.255
    next
    edit "FTLWIN8662451639"
        set uuid ae9083b6-4d6f-51f0-b13d-a61ffcd634e9
        set subnet 10.1.100.108 255.255.255.255
```

```
    next
    edit "FT100GTK24000007"
        set uuid cd9c80f2-4d6f-51f0-4d5a-b160e44881d1
        set subnet 192.168.14.8 255.255.255.255
    next
end
```

2. Upgrade FortiOS to 7.6.4 or later.
3. In FortiOS 7.6.4, view the configuration.

   For each address, the `type` is `dynamic`, the `sub-type` is `telemetry`, and the `agent-id` is the serial number for the FortiTelemetry agent.

```
show firewall address
config firewall address
    edit "FTLWIN5798911362"
        set uuid 8489a542-4d70-51f0-327e-e91bd439cf25
        set type dynamic
        set sub-type telemetry
        set agent-id "FTLWIN5798911362"
    next
    edit "FTLWIN8662451639"
        set uuid 8489a9d4-4d70-51f0-153f-e36cde5329d0
        set type dynamic
        set sub-type telemetry
        set agent-id "FTLWIN8662451639"
    next
    edit "FT100GTK24000007"
        set uuid 8489ae5c-4d70-51f0-1617-716c6bd9df2a
        set type dynamic
        set sub-type telemetry
        set agent-id "FT100GTK24000007"
    next
end
```

Custom tags NEW

Custom tags can be created and applied to various address types, address groups, and policy types. This helps administrators organize their addresses and policies, quickly visualize the category, and easily filter based on the assigned tags when many addresses and policies are configured.

The following objects can have custom tags assigned:

- Firewall address and address groups (IPv4 and IPv6)
- Proxy address and address groups
- Multicast addresses (IPv4 and IPv6)
- Firewall policy
- NGFW security policy
- Multicast policy

- DoS policy
- Local-in policy
- Central SNAT mapping
- Proxy policy

When address groups use custom tags as group members, they are called dynamic tag address groups. You can define more address objects for the tag associated with the address group for the firewall policy to automatically use the new address objects. No changes to the firewall policy are required.

Custom tags have tablesize limits. Refer to the Max Values table for exact size limits.

CLI syntax

Options are available to create custom tags:

```
config firewall custom-tag
    edit <tag name>
        set abbreviation <string>
        set comment <string>
        set color <integer>
    next
end
```

Command	Descriptions
`abbreviation <string>`	Custom tag name abbreviation. Maximum of four uppercase letters and numbers.
`comment <string>`	Comment.
`color <integer>`	Color of the custom tag on the GUI (0 to 32, default = 1).

To apply the tag to an address or policy:

```
config firewall address
    edit "taggedAddress"
        set custom-tags <tag> [tag]
    next
end
```

```
config firewall policy
    edit <ID>
        set custom-tags <tag> [tags]
    next
end
```

Options are available to create dynamic tag address groups with tags as group members:

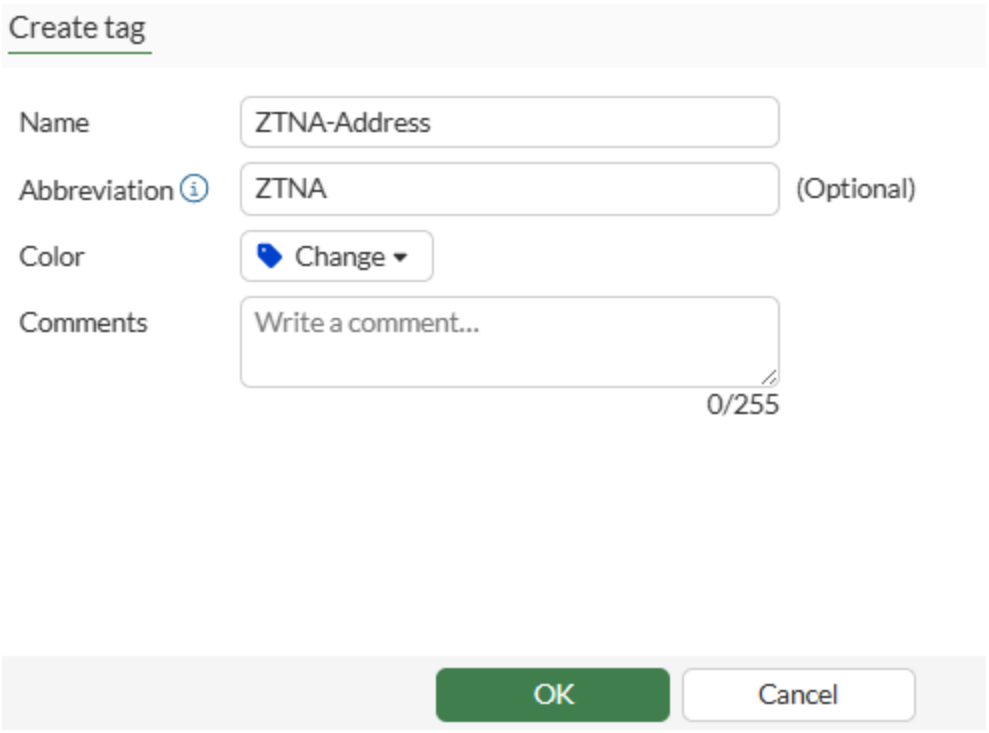
```
config firewall addrgrp
    edit <name>
```

```
        set type {default | folder | dynamic-tag}
        set custom-tags <string>
    next
end
```

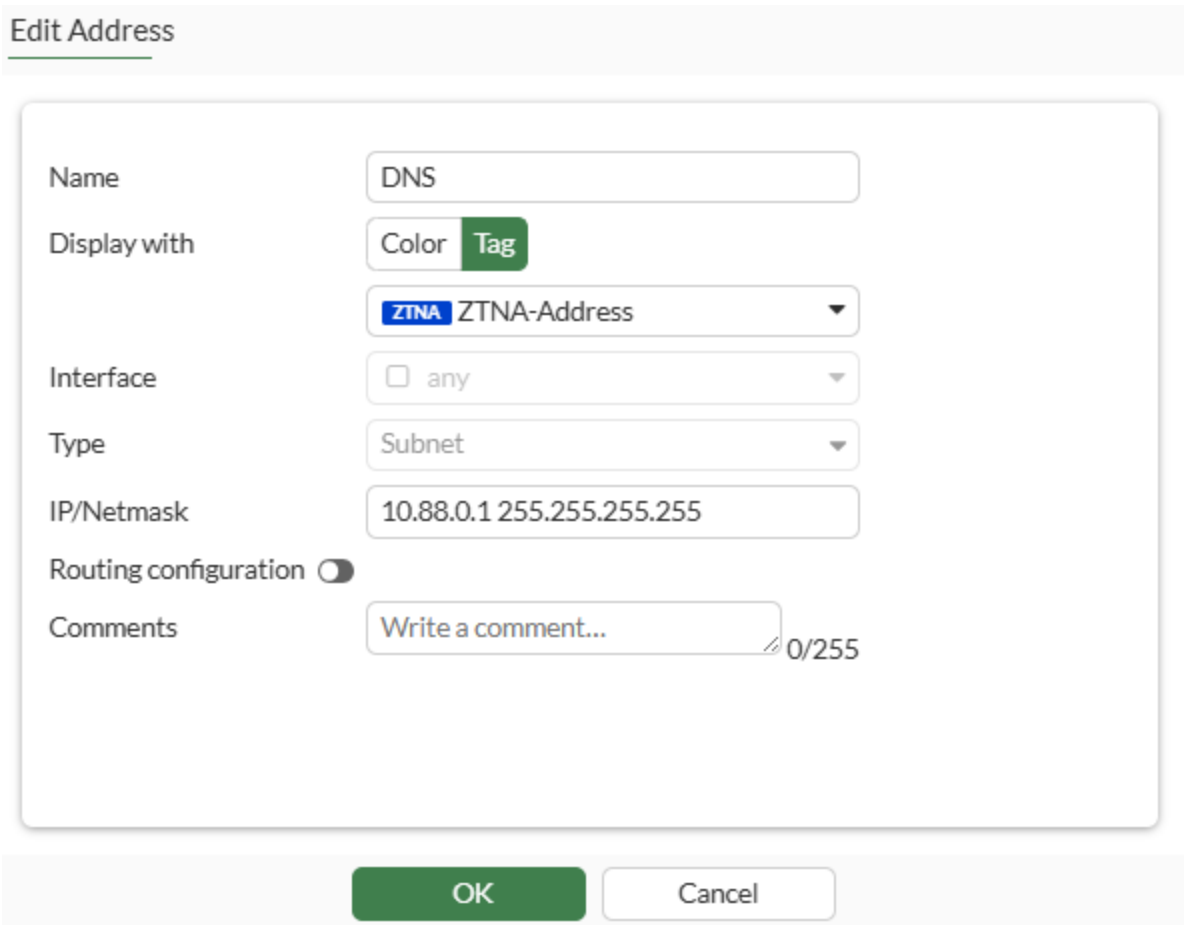
Command	Descriptions
`type {default | folder | dynamic-tag}`	Type of address group: • `default`: Default address group type (address may belong to multiple groups). • `folder`: Address folder group (members may not belong to any other group). • `dynamic-tag`: Dynamic tag group (members are addresses only if they have any one of the configured tags).
`custom-tags <string>`	Name of the custom tag. Available when `type` is set to `dynamic-tag`.

GUI

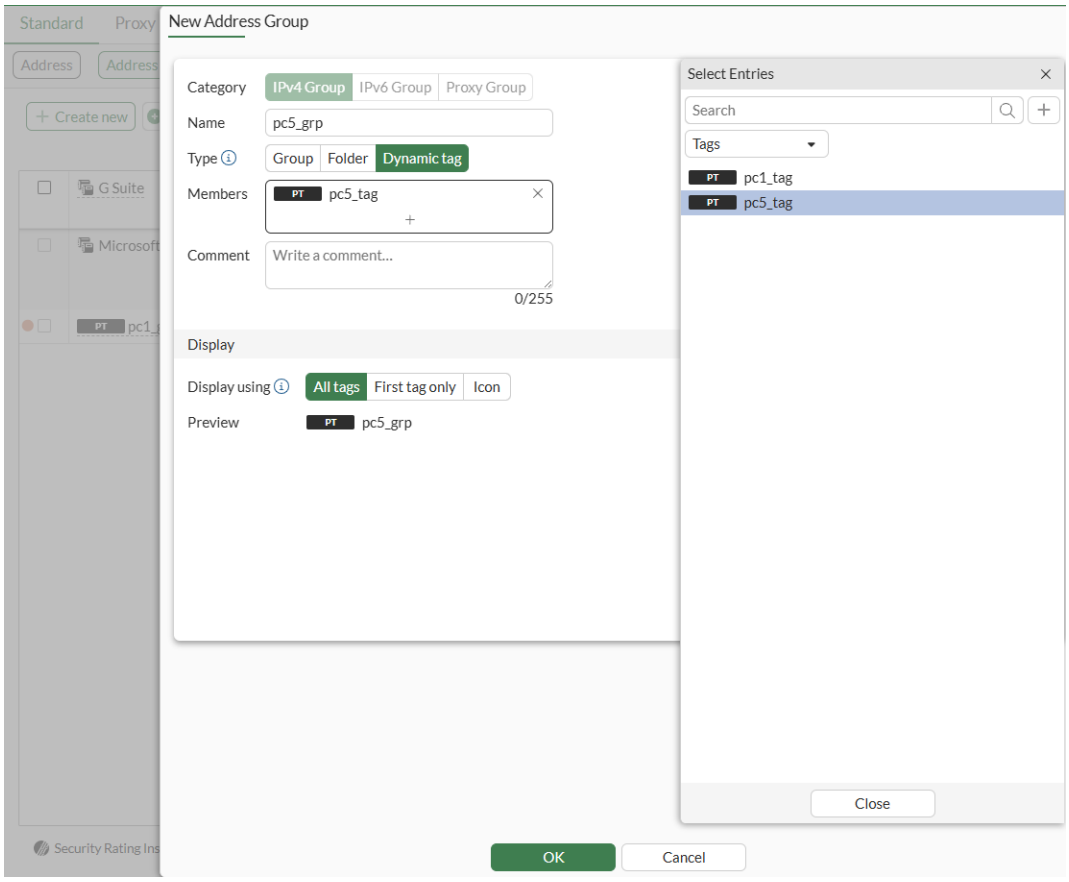
Custom tags can be configured from *Policy & Objects* > *Tags*.



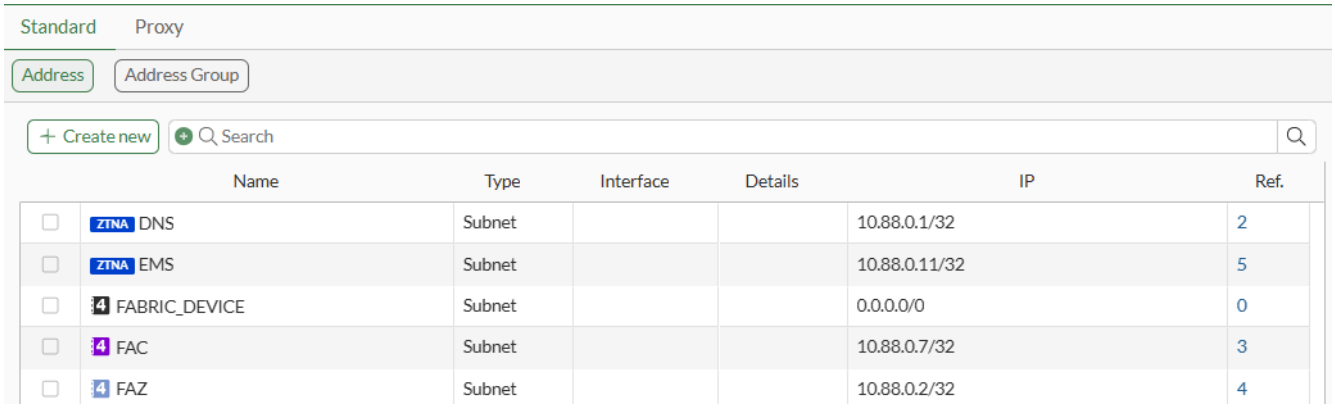
The abbreviation can be up to three uppercase letters and numbers.



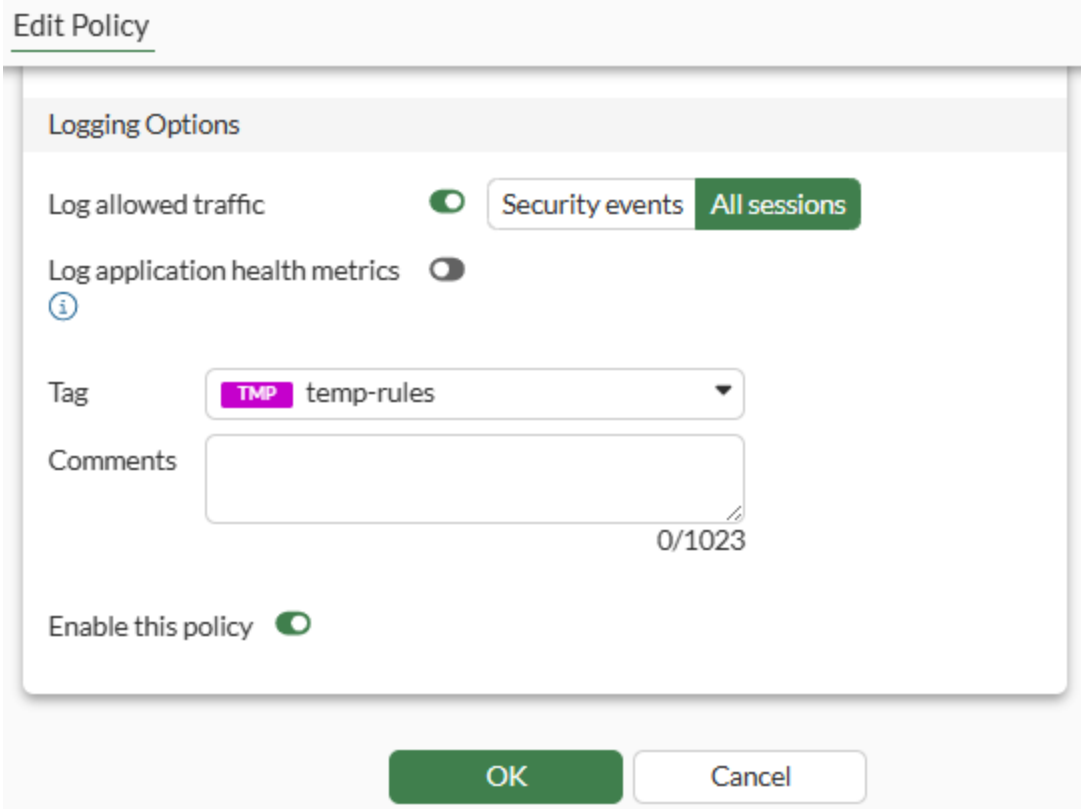
In address groups, set *Type* to *Dynamic* tag to select one or more tags:



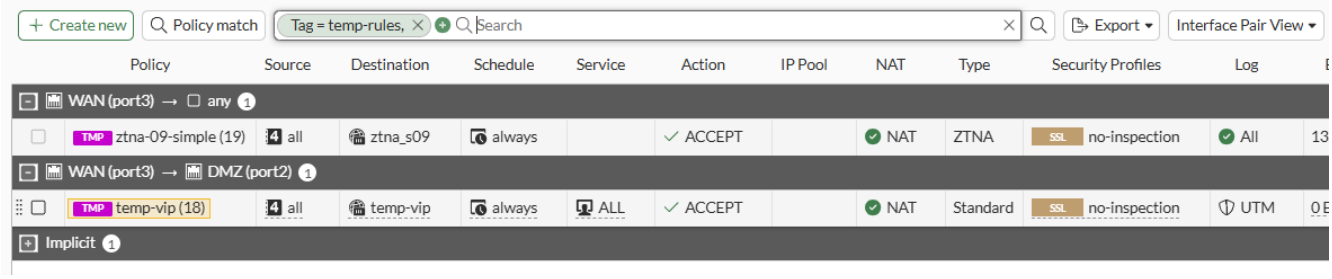
When *All tags* or *First tag only* is chosen as the display, the tag is shown as opposed to the default colored icon in the Address objects page for that particular object.



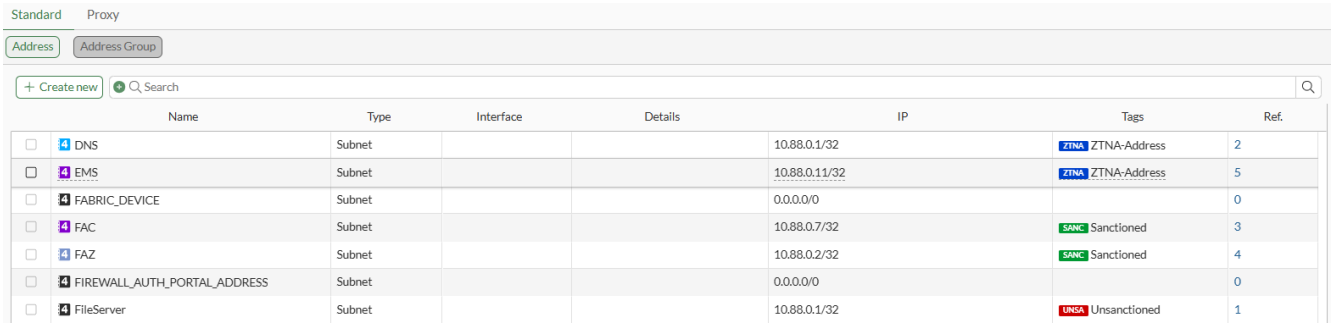
Policy Tagging is not enabled by default and must be enabled in *Feature Visibility*. Once enabled, a tag can be applied to the policy.



Tags are visible in the Policy list page and can be filtered in the Search bar.



Display changes slightly when multiple object tags are used. Use the *Tags* column to view the assigned tags.



Example

This example describes how to create two custom tags, associate the custom tags with address objects, and configure dynamic tag address groups. Dynamic tag address groups use tag names as group members instead of object names. The firewall policy is configured to use the dynamic tag address groups to allow traffic from PC1 to PC5.

The following items are created:

- Two custom tags:
  - pc1_tag
  - pc5_tag
- Two address objects:
  - pc1 assigned the pc1_tag
  - pc5 assigned the pc5_tag
- Two dynamic tag address groups:
  - pc1_grp assigned the pc1_tag
  - pc5_grp assigned the pc5_tag

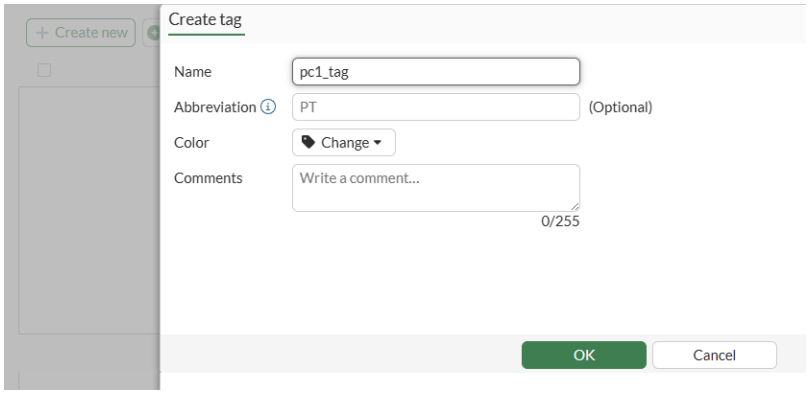
The firewall policy is configured to use the dynamic tag address groups (pc1_grp and pc5_grp) to allow traffic from PC1 to PC5.

When you add more address objects with tags, the tags are used to automatically update the dynamic tag address group used by the firewall policy. No changes are required on the associated firewall policy.

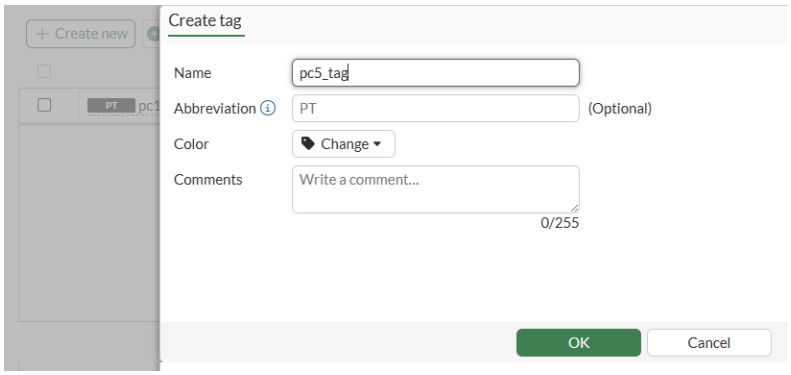
For example, you could create more address objects assigned the pc1_tag for the firewall policy to automatically use the address objects for the pc1_grp, which is configured to use address objects assigned the pc1_tag.

To create dynamic tag address groups in the GUI:

1. Create two custom tags:
   a. Go to *Policy & Objects* > *Tags*.
   b. Click *Create new*.
   c. Enter a *Name*, such as pc1_tag.

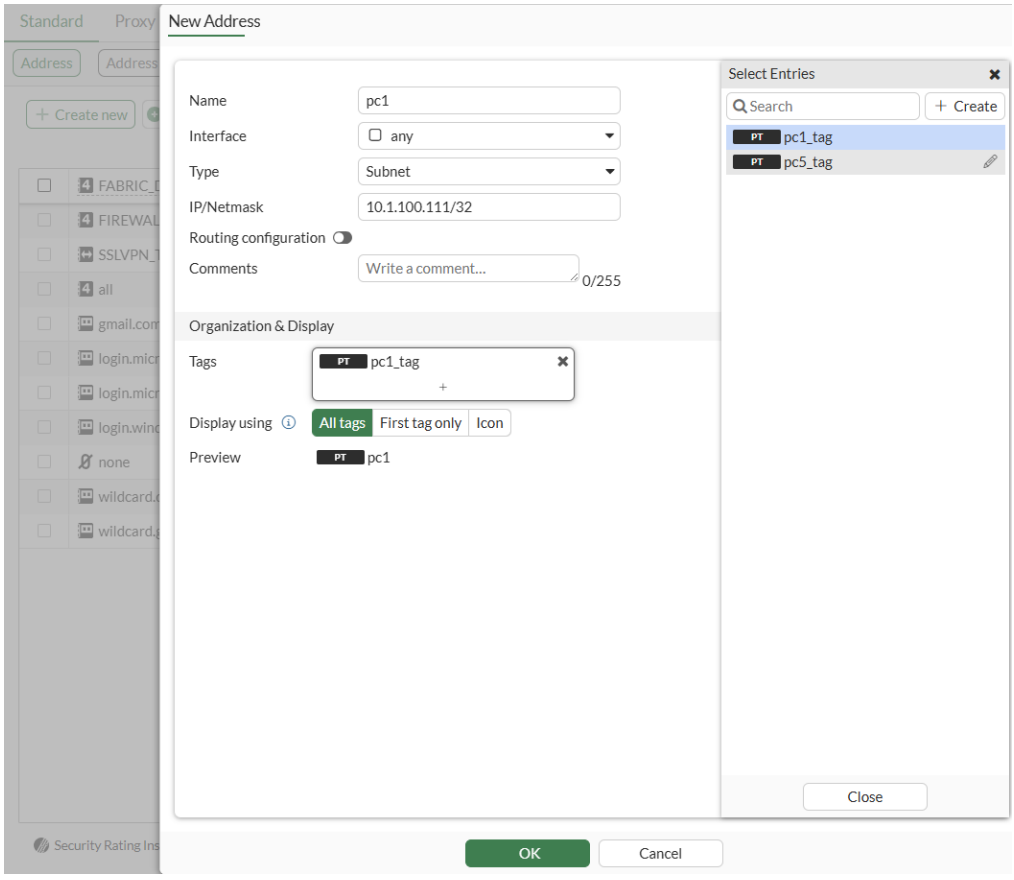


   d. Click *OK*.
   e. Repeat this procedure to create another custom tag named pc5_tag.

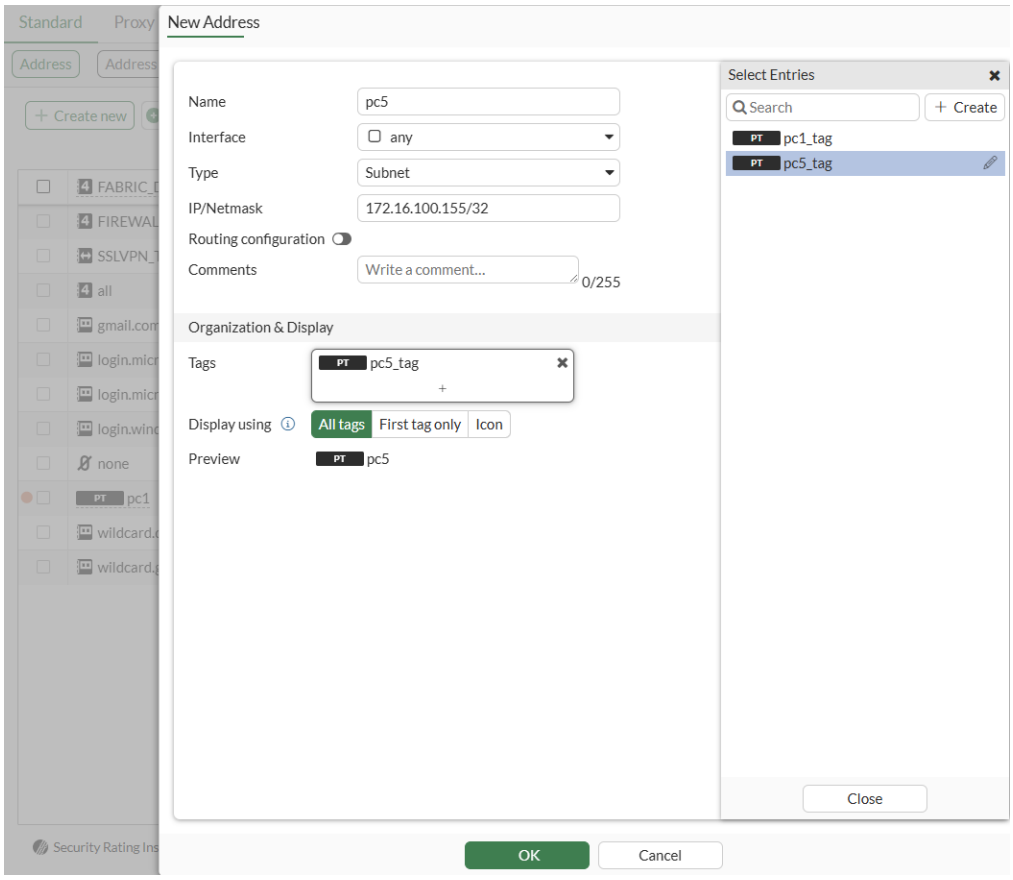


Two custom tags are available: pc1_tag and pc5.

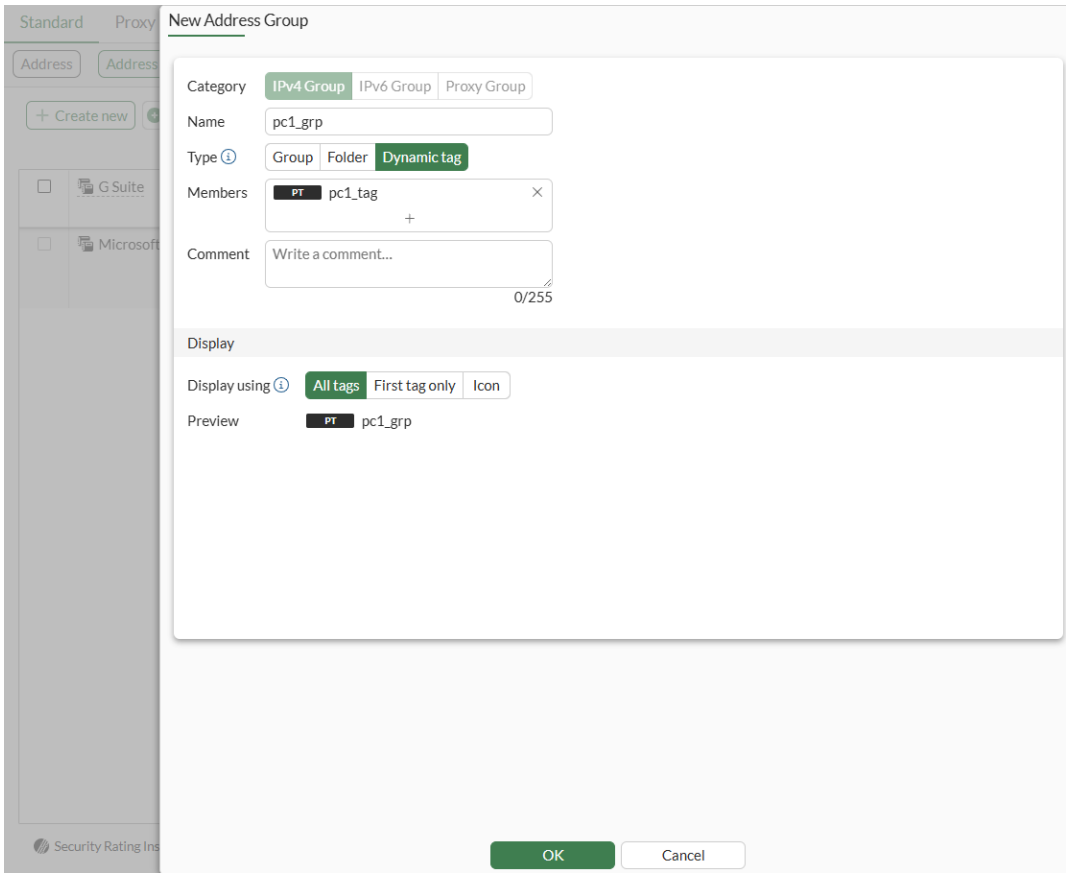
2. Create two address objects for PC1 and PC5 and assign to them the tags you created:
   a. Go to *Policy & Objects* > *Addresses*.
   b. On the *Address* tab, click *Create new*.
   c. Enter a *Name*, such as pc1.
   d. Beside *Tags*, click *+*, and select the pc1_tag you created.



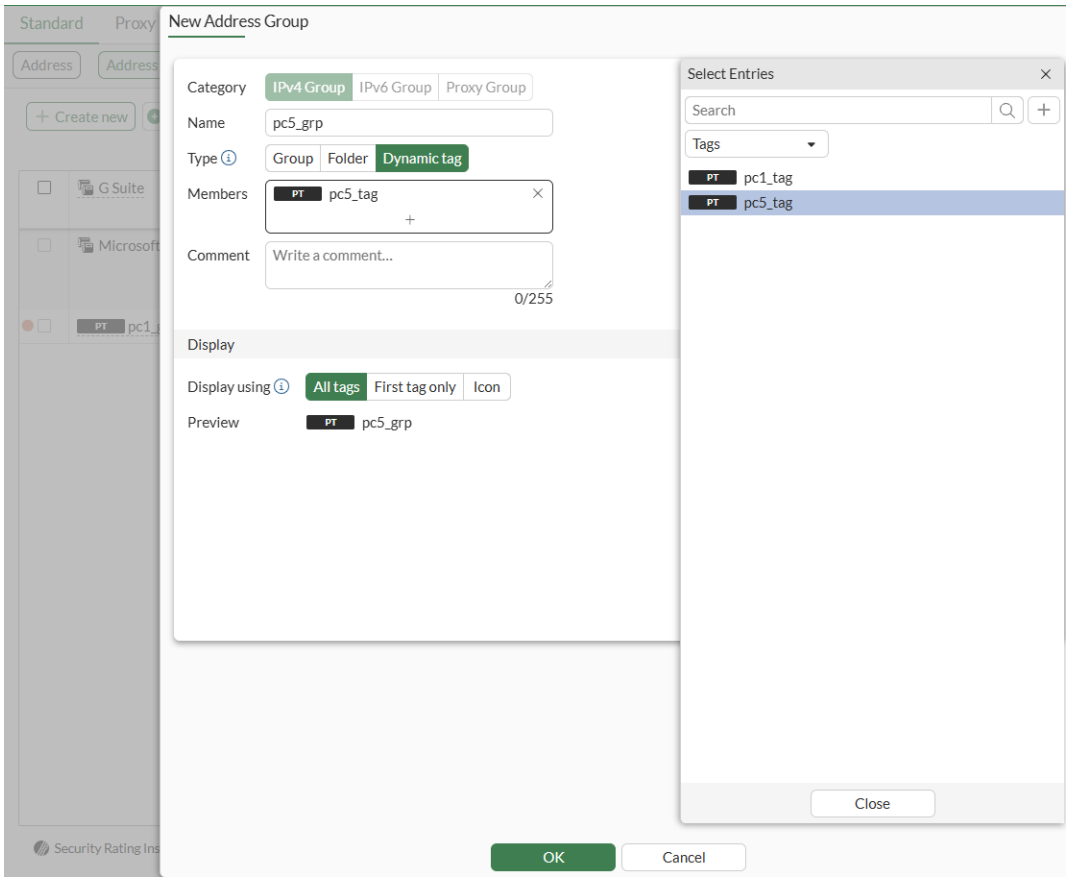
   e. Click *OK*.
   f. Repeat this procedure to create an address for pc5 and select the pc5_tag.



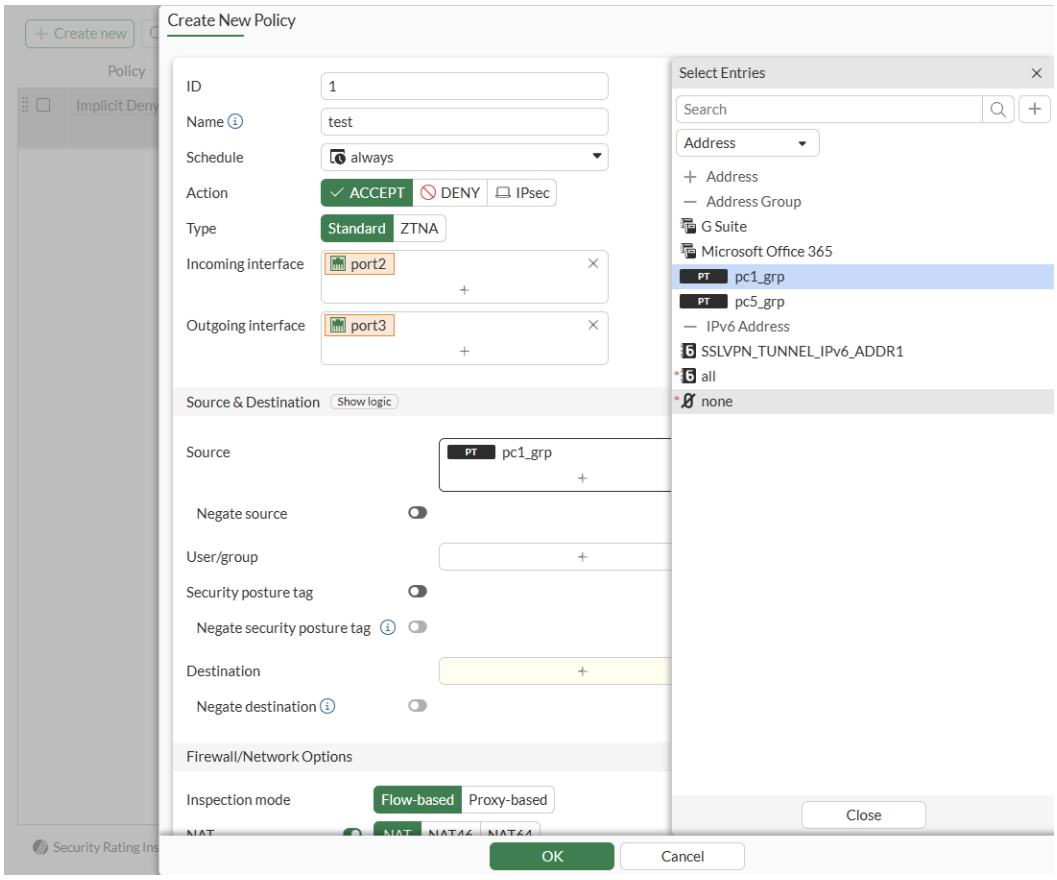
3. Create two dynamic tag address groups, with the tag names (pc1_tag and pc5_tag) as group members instead of the object names.
   a. In *Policy & Objects* > *Addresses*, click the *Address Group* tab.
   b. Click *Create new*.
   c. Enter a *Name*, such as pc1_grp.
   d. Set *Type* to *Dynamic tag*.
   e. Beside *Members*, click *+*, select the tag name (pc1_tag), and click *Close*.



f. Repeat this procedure to create an address group named pc5_grp and add custom tag pc5_tag as group member.



4. Enable policy tagging:
   a. Go to *System* > *Feature Visibility*.
   b. Under *Additional Features*, enable *Policy Tagging*.
   c. Click *Apply*.
5. Configure a firewall policy to allow traffic from PC1 to PC5.
   a. Go to *Policy & Objects* > *Firewall Policy*.
   b. Click *Create new*.
   c. Enter a *Name*.
   d. Beside *Source*, click *+*, select the address group named pc1_grp, and click *Close*.



e. Beside *Destination*, click *+*, select the address group named pc5_grp, and click *Close*

f. Set the remaining options as needed.

g. Click *OK*.

To create dynamic tag address groups in the CLI:

1. Create two custom tags:

```
config firewall custom-tag
    edit "pc1_tag"
    next
    edit "pc5_tag"
    next
end
```

2. Create two address objects named pc1 and pc5 with the pc1_tag and pc5_tag:

```
config firewall address
    edit "pc1"
        set custom-tags "pc1_tag"
        set subnet 10.1.100.111 255.255.255.255
    next

    edit "pc5"
```

```
        set custom-tags "pc5_tag"
        set subnet 172.16.200.155 255.255.255.255
    next
end
```

3. Create two dynamic tag address groups with the following tags as group members: pc1_tag and pc5_tag:

```
config firewall addrgrp
    edit "pc1_grp"
        set type dynamic-tag
        set custom-tags "pc1_tag"
    next
    edit "pc5_grp"
        set type dynamic-tag
        set custom-tags "pc5_tag"
    next
end
```

4. Configure a firewall policy to allow traffic from PC1 to PC5 using the address groups:

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "port3"
        set action accept
        set srcaddr "pc1_grp"
        set dstaddr "pc5_grp"
        set schedule "always"
        set service "ALL"
        set auto-asic-offload disable
    next
end
```

Protocol options

Firewall policies contain a *Protocol Options* field that defines the parameters for handling protocol-specific traffic. Multiple protocol options profiles can be configured in FortiOS since the requirements may differ between policies. A single protocol options profile is applied per policy, but the profile can be used in multiple policies.

To create a protocol options profile, go to *Policy & Objects* > *Protocol Options*. The following settings can be configured.

Log oversized files

Enable this option to log the occurrence of oversized files being processed. This does not change how they are processed. It only allows the FortiGate to log that they were either blocked or allowed through.

It is common practice to allow larger files through without antivirus processing. Monitor the logs for the frequency of oversized file processing to determine whether or not to alter the settings for treating oversized files. The threshold setting for oversized files and emails is located in the *Common Options* section.

RPC over HTTP

This protocol is used by Microsoft Exchange Servers to perform virus scanning on emails that use RPC over HTTP.

Protocol port mapping

To optimize the FortiGate's resources, the mapping and inspection of the following protocols can be enabled or disabled:

- HTTP
- SMTP
- POP3
- IMAP
- FTP
- NNTP
- MAPI
- DNS
- CIFS

Each protocol has a default TCP port. The ports can be modified to inspect any port with flowing traffic. The packet headers indicate which protocol generated the packet.

Protocol port mapping only works with proxy-based inspection. Flow-based inspection inspects all ports regardless of the protocol port mapping configuration.

Common options

The *Comfort Clients* and *Block Oversized File/Email* options apply to multiple protocols.

Comfort clients

When proxy-based antivirus scanning is enabled, the FortiGate buffers files as they are downloaded. Once the entire file is captured, the FortiGate begins scanning the file. The user must wait during the buffering and scanning procedure. After the scan is completed and if no infection is found, the file is sent to the next step in the process flow. If the file is large, this part of the process can take some time. In some cases, enough time that some users may get impatient and cancel the download.

The *Comfort Clients* option mitigates this potential issue by feeding a trickle of data while waiting for the scan to complete. The user is aware that processing is taking place, and that there has not been a failure in the transmission. The slow transfer rate continues until the antivirus scan is complete. The transfer will proceed at full speed once the file is scanned successfully and does not contain any viruses.

If there is evidence of an infection, the FortiGate caches the URL and drops the connection. The client does not receive any notification of what happened because the download to the client has already started. Instead, the download stops and the user is left with a partially downloaded file. If the user tries to download the same file

again within a short period of time, the cached URL is matched and the download is blocked. A notification is displayed that the download was blocked. The number of URLs in the cache is limited by the size of the cache.

Client comforting is available for HTTP and FTP traffic. If the FortiGate supports SSL content scanning and inspection, client comforting can be configured for HTTPS and FTPS traffic.

> ⚠ Buffering the entire file allows the FortiGate to eliminate the danger of missing an infection due to fragmentation because the file is reassembled before examination. This buffering is performed whenever the *Comfort Clients* option is disabled.
>
> Client comforting can send unscanned and potentially infected content to the client, so only enable this option if you are prepared to accept this risk. Keeping the client comforting interval high and the amount low will reduce the amount of potentially infected data that is downloaded.

Block oversized files and emails

This option is related to antivirus scanning. The FortiGate has a finite amount of resources to buffer and scan a file. If a large file (such as an ISO image or video file) is downloaded, this could overwhelm or exceed the FortiGate's memory, especially if other large files are being downloaded at the same time.

A threshold is assigned to identify an oversize file or email. The default is 10 MB. The range varies per model, and the minimum is 1 MB. Any file or email over this threshold will not be processed by policies applying the antivirus security profile.

> 💡 If the FortiGate enters conserve mode on a regular basis, lowering the threshold can lessen the impact of processing the files on memory. This can increase risk, even though malware is more likely to be in smaller files.

Web options

The *Chunked Bypass* option applies to traffic containing web protocols.

Chunked bypass

Chunked bypass is a mechanism in HTTP 1.1 that allows a web server to start sending chunks of dynamically generated output in response to a request before actually knowing the actual size of the content. For dynamically generated content, enabling chunked bypass speeds up the initial response to HTTP requests, but the content is not held in the proxy as an entire file before proceeding.

Email options

The *Allow Fragmented Messages* and *Append Signature (SMTP)* options apply to email protocols.

Allow fragmented messages

The specifications of RFC 2046 allow for the breaking up of emails and sending the fragments in parallel to be rebuilt and read at the other end by the mail server. It was originally designed to increase the performance over slower connections where larger email messages were involved. Feasibility of using this function depends on the mail configuration. Outside of Microsoft Outlook, not many email clients are set up to break up messages like this. The drawback of this feature is that if malware is broken up between multiple fragments of the message, there is a risk that it will not be detected by some antivirus configurations because all the code may not be present at the same time to identify the malware.

Append signature

This option adds a plain text email signature to SMTP email messages as they pass through the FortiGate. The message maximum is 1023 characters.

This feature works best in an environment where there is some standardization of what goes into the senders' personal signatures so that there is no duplication or contradiction of information. For example:

- *This email should not be forwarded without prior approval.*
- *Please consider the environment before printing this email.*
- *For questions regarding purchasing our products, please call ...*

Configuring handling of unknown content types

FortiOS allows you to configure its behavior when handling unknown content types for the HTTP protocol.

Unknown content types for the HTTP protocol can be blocked (default), inspected, or bypassed:

```
config firewall profile-protocol-options
    edit <name>
        config http
            set unknown-content-encoding {block | inspect | bypass}
        end
    next
end
```

Option	Description
`Block`	Block the session when detecting an unknown content type (default)
`inspect`	Inspect the session as plain text when detecting an unknown content type
`bypass`	Bypass the session when detecting an unknown content type

For cases where legitimate websites with specific unknown content types are being accessed, or legitimate files with specific unknown content types are being downloaded, you can configure this setting as `inspect` or `bypass`.

Stripping the X-Forwarded-For value in the HTTP header

The X-Forwarded-For value in the HTTP header can be stripped when the `strip-x-forwarded-for` option is enabled under `firewall profile-protocol-options`. This feature sets the value to empty using the IPS engine.

The following types of traffic support X-Forwarded-For stripping:

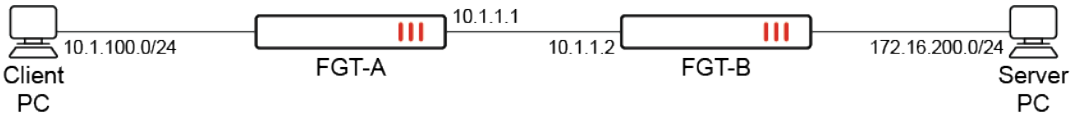
- HTTP/1.1, HTTP/2, and HTTP/3 traffic that matches an NGFW mode security policy with flow-based inspection.
- Plain HTTP/1.1 traffic that matches a firewall policy with proxy-based inspection.

The following types of traffic do not support X-Forwarded-For stripping:

- HTTPS traffic that matches a firewall policy with proxy-based inspection.
- HTTP and HTTPS traffic that matches an explicit web proxy policy.

Example

In this example, FGT-A is configured with `strip-x-forwarded-for` enabled for HTTP. On FGT-B, the IPS sensor is configured to monitor the Eicar.Virus.Test.File signature. The IPS logs on FGT-B are used to verify the traffic sent from FGT-A to FGT-B, namely the `forwardedfor` value in the `rawdata` field.



To configure X-Forwarded-For stripping:

1. Configure FGT-A:
   a. Configure the protocol options for HTTP:

```
config firewall profile-protocol-options
    edit "protocol-xff"
        config http
            set ports 80
            unset options
            set strip-x-forwarded-for enable
            unset post-lang
        end
    next
end
```

   b. Configure the firewall policy (ensure that an IPS sensor is applied):

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "port5"
        set action accept
```

```
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set profile-protocol-options "protocol-xff"
        set ssl-ssh-profile "ssl-deep"
        set ips-sensor "default"
        set nat enable
    next
end
```

2. Configure FGT-B:
   a. Configure the IPS sensor with extended logging:

```
config ips sensor
    edit "monitor-eicar"
        set extended-log enable
        config entries
            edit 1
                set rule 29844
                set status enable
                set action pass
            next
        end
    next
end
```

   b. Configure the firewall policy (ensure that an IPS sensor is applied):

```
config firewall policy
    edit 3
        set srcintf "port5"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "ssl-deep"
        set ips-sensor "monitor-eicar"
        set nat enable
    next
end
```

To verify the configuration:

1. Use a cURL request to send HTTPS traffic with HTTP header X-Forwarded-For from the Client PC to the Server PC:

```
curl -vk -H "X-Forwarded-For: 10.22.22.22" https://172.16.200.52/eicar.com
```

2. On FGT-B, verify the corresponding IPS logs.
   a. For HTTP/1.1, the `X-Forwarded-For` value is removed from the `rawdata` field, and the `forwardedfor` value is not included:

```
1: date=2023-09-21 time=14:05:34 eventtime=1695330334919589600 logid="0419016384"
type="utm" subtype="ips" eventtype="signature" level="alert" vd="root" severity="info"
srcip=10.1.1.1 srccountry="Reserved" dstip=172.16.200.42 dstcountry="Reserved"
srcintf="port5" srcintfrole="undefined" dstintf="port1" dstintfrole="undefined"
sessionid=2471 action="detected" proto=6 service="HTTPS" policyid=3 poluuid="782b9e86-
58a3-51ee-8e0f-79c7682223dd" policytype="policy" attack="Eicar.Virus.Test.File"
srcport=36018 dstport=443 hostname="172.16.200.42" url="/eicar.com" agent="curl/7.61.1"
httpmethod="GET" direction="incoming" attackid=29844 profile="monitor-eicar"
ref="http://www.fortinet.com/ids/VID29844" incidentserialno=75497475 msg="file_transfer:
Eicar.Virus.Test.File" rawdataid="1/1" rawdata="Response-Content-Type=application/x-msdos-
program" crscore=5 craction=65536 crlevel="low"
```

   b. For HTTP/2 and HTTP/3, the `X-Forwarded-For` value is removed from the `rawdata` field, and `forwardedfor` is included:

```
1: date=2023-09-21 time=14:05:56 eventtime=1695330356543624871 logid="0419016384"
type="utm" subtype="ips" eventtype="signature" level="alert" vd="root" severity="info"
srcip=10.1.1.1 srccountry="Reserved" dstip=172.16.200.52 dstcountry="Reserved"
srcintf="port5" srcintfrole="undefined" dstintf="port1" dstintfrole="undefined"
sessionid=2474 action="detected" proto=6 service="HTTPS" policyid=3 poluuid="782b9e86-
58a3-51ee-8e0f-79c7682223dd" policytype="policy" attack="Eicar.Virus.Test.File"
srcport=37786 dstport=443 hostname="172.16.200.52" url="/eicar.com" agent="curl/7.61.1"
httpmethod="GET" direction="incoming" attackid=29844 profile="monitor-eicar"
ref="http://www.fortinet.com/ids/VID29844" incidentserialno=75497476 msg="file_transfer:
Eicar.Virus.Test.File" rawdataid="1/1" forwardedfor="\r\n" rawdata="Response-Content-
Type=application/x-msdos-program" crscore=5 craction=65536 crlevel="low"
```

3. On FGT-A, disable `strip-x-forwarded-for` for HTTP:

```
config firewall profile-protocol-options
    edit "protocol-xff"
        config http
            set strip-x-forwarded-for disable
        end
    next
end
```

4. Send the same HTTPS traffic with HTTP header X-Forwarded-For from the Client PC to the Server PC.
5. On FGT-B, verify the corresponding IPS log, which includes `forwardedfor` and `X-Forwarded-For` values in the `rawdata` field:

```
1: date=2023-09-21 time=16:33:06 eventtime=1695339187144132034 logid="0419016384" type="utm"
subtype="ips" eventtype="signature" level="alert" vd="root" severity="info" srcip=10.1.1.1
srccountry="Reserved" dstip=172.16.200.52 dstcountry="Reserved" srcintf="port5"
srcintfrole="undefined" dstintf="port1" dstintfrole="undefined" sessionid=3776
action="detected" proto=6 service="HTTPS" policyid=3 poluuid="782b9e86-58a3-51ee-8e0f-
79c7682223dd" policytype="policy" attack="Eicar.Virus.Test.File" srcport=37788 dstport=443
hostname="172.16.200.52" url="/eicar.com" agent="curl/7.61.1" httpmethod="GET"
```

```
direction="incoming" attackid=29844 profile="monitor-eicar"
ref="http://www.fortinet.com/ids/VID29844" incidentserialno=75497478 msg="file_transfer:
Eicar.Virus.Test.File" rawdataid="1/1" forwardedfor="10.22.22.22" rawdata="Response-Content-
Type=application/x-msdos-program|X-Forwarded-For=10.22.22.22" crscore=5 craction=65536
crlevel="low"
```

Traffic shaping

A FortiGate provides quality of service (QoS) by applying bandwidth limits and prioritization to network traffic. Traffic shaping is one technique used by the FortiGate to provide QoS. A basic approach to traffic shaping is to prioritize higher priority traffic over lower priority traffic during periods of traffic congestion. This provides a stabilizing effect for important traffic while throttling less important traffic.

The FortiGate can be configured to deliver traffic shaping with policing or traffic shaping with queuing. The general difference between the two is as follows:

Technique	Description
Traffic shaping with policing	When traffic exceeds the configured bandwidth limits, traffic is dropped.
Traffic shaping with queuing	When traffic exceeds the configured bandwidth limits, traffic is delayed for transport until bandwidth frees up. Traffic may be dropped if the queues are full.

Policing and queuing can both prioritize traffic and deliver guaranteed bandwidth and maximum bandwidth by setting bandwidth limits. The implementation differs though, since queuing uses queues, and policing does not. In queuing, before a packet egresses an interface, it is first enqueued to a queue using an algorithm such as RED or FIFO. The kernel dequeues the packet based on the HTB algorithm before sending it out. In policing, traffic simply drops if it is over the allocated bandwidth.

The following topics provide information about configuring traffic shaping:

- Traffic shaping policies on page 2110
- Traffic shaping profiles on page 2120
- Traffic shapers on page 2131
- Global traffic prioritization on page 2148
- DSCP matching and DSCP marking on page 2152
- Examples on page 2159

Configuration methods

There are different methods to configure traffic shaping on the FortiGate. The following table lists the methods and their capabilities in order of preference. If all three methods are configured, the first will be preferred over the second, which is preferred over the third.

Method	Policing		Queuing
	Traffic prioritization	Guaranteed and maximum bandwidth limits	Traffic queuing
Traffic shaping profile*	Yes	Yes, based on percentage of outbandwidth	Yes
Traffic shaper	Yes	Yes, based on rate	No
Global traffic prioritization	Yes	No	No

* Traffic shaping profiles are configured as either policing or queuing types. Queuing allows for additional options when configuring a shaping class entry.

The features of each method's implementation are slightly different. The following is a brief summary of the traffic policing features and the approach each method takes.

Traffic prioritization

The FortiGate can place packets into different priority levels in order to prioritize certain traffic over others.

Method	Description
Traffic shaping profile	Traffic is placed into classes. A total of 30 classes are available. For each class, traffic can be configured into five priority levels.
Traffic shaper	Traffic can be prioritized into the high (2), medium (3), or low (4) levels. When traffic is below the guaranteed bandwidth of the shaper, the traffic is automatically applied the critical level (1).
Global traffic prioritization	Traffic is prioritized into high (2), medium (3), or low (4) based on ToS (type of service) or DSCP.

Guaranteed and maximum bandwidth limits

The general purpose for configuring guaranteed bandwidth is to allocate a certain proportion of the total outbandwidth to guarantee transport for a certain type of traffic. This is configured and handled differently in each method.

A traffic shaping profile, when applied to an interface's egress shaping profile, can be configured to use up to 100% of the interface's configured bandwidth between all the classes. It does not matter what priority is configured in each class. The guaranteed bandwidth is always honored.

Traffic shapers, however, do not have a hard limit on the guaranteed bandwidth. Administrators need to be aware how much guaranteed bandwidth has been allocated to all their traffic shapers, so that they do not exceed the total outbandwidth of an interface. Traffic under the guaranteed bandwidth of a traffic shaper is given a priority of one. If the total traffic with priority one exceeds the total outbandwidth, traffic can be dropped.

The maximum bandwidth limit caps the maximum bandwidth that can be used. This is configured as a percentage of the outbandwidth in a traffic shaping profile. It is configured as a rate for traffic shapers.

Configuring outbandwidth

Traffic shaping is generally configured for egress traffic leaving the FortiGate. Therefore, it is necessary for the interface outbandwidth to be defined for traffic prioritization to take place in all of the traffic shaping configuration methods. Interface outbandwidth is also needed when defining the guaranteed and maximum bandwidth in a traffic shaping profile.

For traffic shapers, configuring outbandwidth is not necessary to apply maximum bandwidth limits; however, outbandwidth is necessary for guaranteed bandwidth. Traffic under the guaranteed bandwidth limit on a traffic shaper is given priority 1. If outbandwidth is not configured, traffic prioritization does not take place and the priority is meaningless.

Traffic shaping policy

Traffic shaping profiles and traffic shapers are methods of policing traffic. Traffic shaping policies are used to map traffic to a traffic shaper or assign them to a class.

A traffic shaping policy is a rule that matches traffic based on certain IP header fields and/or upper layer criteria. For example, it can match traffic based on source and destination IP, service, application, and URL category. One common use case is to match traffic based on the ToS or DS (differentiated services) field in the IP header. This allows Type of Service or Differentiated Services (DiffServ) tags to be read from traffic from a downstream device and prioritized accordingly on the FortiGate.

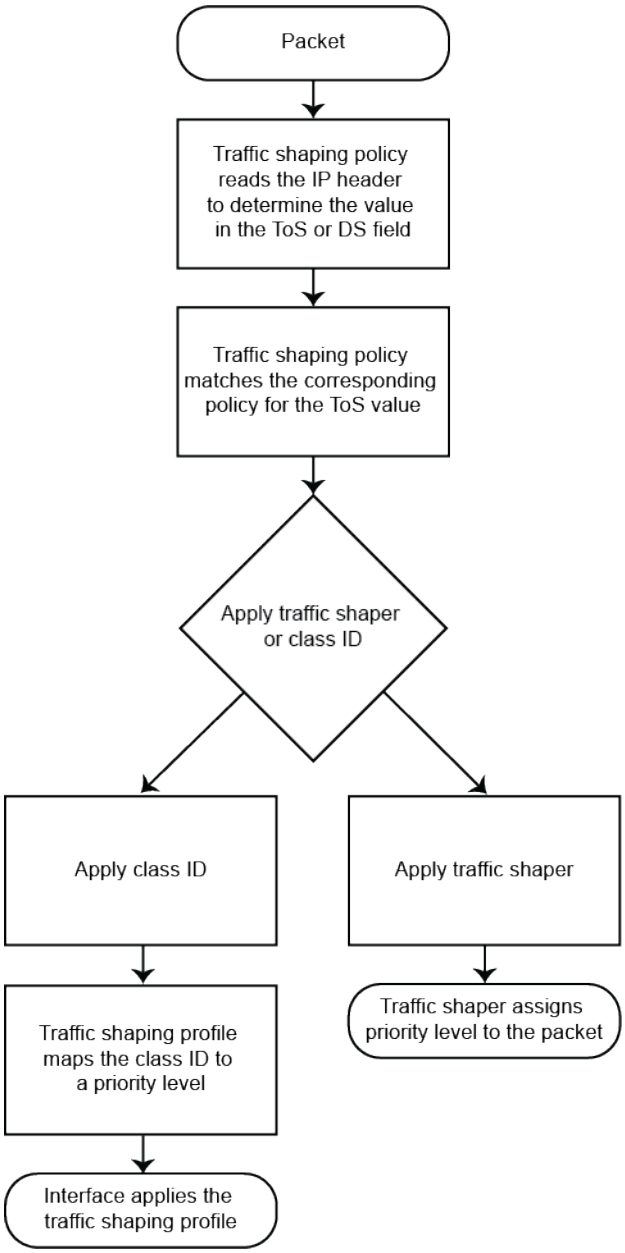
DSCP matching and DSCP marking

DSCP matching and DSCP marking can be performed on a firewall shaping policy and a regular firewall policy. DSCP matching is used to match DSCP tags from ingress traffic, and DSCP marking is used to change the DSCP tag on egress traffic.

In a firewall shaping policy and regular firewall policy, use the `tos` and `tos-mask` fields to perform DSCP matching. Use the `diffserv-forward` and `diffserv-reverse` fields to perform DSCP marking.

Traffic shaping policies

As mentioned in Traffic shaping on page 2108, traffic shaping starts with the traffic shaping policy. Traffic shaping policies are used to map traffic to a traffic shaper or assign them to a class. Traffic is then shaped by the shaper or the shaping profile that is applied on an interface.



Traffic can also be shaped by applying traffic shapers directly on a firewall policy. However, this legacy approach can only be configured from the CLI, and is not a preferred method for applying traffic shaping. As the number of firewall policies increases, managing shaping on each individual policy becomes increasingly difficult. For the same reason, it is also not recommended to mix the legacy approach with traffic shaping policies to avoid the added complexity.

Overview

A traffic shaping policy is a rule that matches traffic based on certain IP header fields and/or upper layer criteria. When traffic hits the firewall, the FortiGate will first look up a firewall policy, and then match a shaping policy. The matching traffic will apply a traffic shaper, class ID, or assign a DSCP DiffServ tag to the outgoing traffic.

The traffic shaping policies must be placed in the correct order in the traffic shaping policy list page to obtain the desired results. Policies are matched from top-down, so the traffic shaping policies should be arranged in a sequence that places the more granular policies above general policies.

The policy can be configured by going to *Policy & Objects* > *Traffic Shaping* and selecting the *Traffic Shaping Policy* tab. If the menu does not display the traffic shaping settings, go to *System* > *Feature Visibility* and enable *Traffic Shaping*.

Configuring traffic shaping policies

A traffic shaping policy can be split into two parts:

- Options used to match the traffic
- Options used to apply actions to the matched traffic

In the GUI, the options are configured in the *If Traffic Matches* and *Then* sections. In the CLI, all options are configured under `config firewall shaping-policy`. Some options can only be configured from the CLI.

The following options can be configured for traffic matching criteria:

GUI option	CLI option	Description
Source		
Address	`set srcaddr <address_object>`	Select the address object to match the source IP.
User	`set users <user_object>`	Select the user object to match the user authenticated for the session.

Internet Service	`set internet-service-src enable`
	`set internet-service-src-name <name>`
	`set internet-service-src-group <group>`
	`set internet-service-src-custom <custom>`
	`set internet-service-src-custom-group <custom_group>`
		Select the internet service to match the source of the incoming traffic. Internet service currently cannot be used with source address.

Destination		
Address	`set dstaddr <address_object>`	Select the address object to match the destination IP.

Internet Service	`set internet-service enable`
	`set internet-service-name <name>`
	`set internet-service-group <group>`
	`set internet-service-custom <custom>`
	`set internet-service-custom-group <custom_group>`
		Select the internet service to match the destination of the incoming traffic. Internet service currently cannot be used with destination address and service.

GUI option	CLI option	Description
Schedule	`set schedule <schedule>`	Enable to select a schedule (one-time, recurring, or group).
Service	`set service <service>`	Select the service or service group for the traffic.
Application		Application control must be enabled in the related firewall policy to learn the application of the traffic.
Application	`set application <application>`	Select the application to match the application of the traffic.
Category	`set app-category <category>`	Select the application category to match the application of the traffic.
Group	`set app-group <groups>`	Select the application group to match the application of the traffic.
URL Category	`set url-category <category>`	Select the URL category to match the URL of the traffic. A web filter profile must be enabled in the related firewall policy to know the URL of the traffic (see Web filter on page 2311).

n/a	`set tos-mask <hexadecimal_mask>`
	`set tos <value>`
	`set tos-negate {enable | disable}`
		Specify the type of service (ToS) and mask to match.
		These options can only be configured in the CLI.

The following options can be configured for actions to apply to the matched traffic:

GUI option	CLI option	Description
Outgoing interface	`set dstintf <interface>`	Select the destination interface that the traffic shaping applies to (required).
Apply shaper		
Shared shaper	`set traffic-shaper <shaper>`	Select the shared shaper to be applied to traffic in the ingress-to-egress direction. For example, on traffic that egresses on the wan interface, the shaper is applied to upload or outbound traffic.
Reverse shaper	`set traffic-shaper-reverse <shaper>`	Select the reverse shaper to be applied to traffic in the egress-to-ingress direction. For example, on traffic that egresses on the wan interface, the shaper is applied to download or inbound traffic.

GUI option	CLI option	Description
Per-IP shaper	`set per-ip-shaper <shaper>`	Select the per-IP shaper. Per-IP shapers affect downloads and uploads. The allotted bandwidth applies to each individual IP. In a shared shaper, the allotted bandwidth applies to all IPs.
Assign shaping class ID		
Traffic shaping class ID	`set class-id <class>`	Set the class ID to apply the matching traffic. Class IDs are further prioritized within a traffic shaping profile and applied to an interface.

n/a	`set diffserv-forward {enable | disable}`
	`set diffservcode-forward <code>`
	`set diffserv-reverse {enable | disable}`
	`set diffservcode-reverse <code>`
		Specify the settings to apply a DSCP tag to the forward or reverse traffic. The DiffServ code is in 6-bit binary format.
		These options can only be configured in the CLI.

Traffic shapers and class IDs can be applied at the same time when configuring traffic shaping policies. However, to reduce the complexity, it is recommended to use one method over the other.

The following topics include examples with traffic shaping policies:

- Local-in and local-out traffic matching on page 2114
- VLAN CoS matching on a traffic shaping policy on page 2117
- Interface-based traffic shaping profile on page 2159
- Shared traffic shaper on page 2131
- Per-IP traffic shaper on page 2136

Local-in and local-out traffic matching

A FortiGate can apply shaping policies to local traffic entering or leaving the firewall interface based on source and destination IP addresses, ports, protocols, and applications.

```
config firewall shaping-policy
    edit <id>
        set traffic-type {forwarding | local-in | local-out}
    next
end
```

This topic contains the following examples:

- Example 1: local-in traffic shaping
- Example 2: local-out traffic shaping

Example 1: local-in traffic shaping

In this example, the traffic shaping policy applies to local-in traffic. The local-in traffic originates from the Linux client and is destined to port1 on the FortiGate.

To configure the traffic shaping policy:

```
config firewall shaping-policy
    edit 2
        set traffic-type local-in
        set service "ALL"
        set schedule "always"
        set class-id 3
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

To verify the configuration:

1. Check the shaping policy information for local-in traffic to verify that the correct class ID (3) is applied:

```
# diagnose firewall iprope list 100018
policy index=2 uuid_idx=1300 action=accept
flag (0):
schedule(always)
cos_fwd=0  cos_rev=0
group=00100018 av=00000000 au=00000000 split=00000000
host=1 chk_client_info=0x0 app_list=0 ips_view=0
misc=0
zone(1): 0 -> zone(1): 0
source(1): 0.0.0.0-255.255.255.255, uuid_idx=1106,
dest(1): 0.0.0.0-255.255.255.255, uuid_idx=1106,
service(1):
        [0:0x0:0/(0,65535)->(0,65535)] flags:0 helper:auto
class_id: 3
```

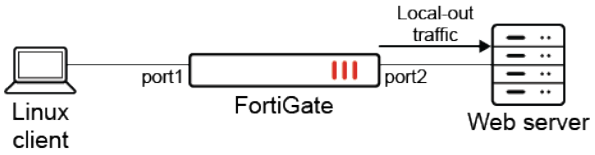
2. Check the session list to verify that the class ID (3) matches the shaping policy ID (2):

```
# diagnose sys session list
session info: proto=6 proto_state=01 duration=1195 expire=3574 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=3 shaping_policy_id=2 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
```

```
state=log local may_dirty
statistic(bytes/packets/allow_err): org=18274/350/1 reply=826037/603/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 1/0
orgin->sink: org pre->in, reply out->post dev=17->34/34->17 gwy=172.16.200.2/0.0.0.0
hook=pre dir=org act=noop 172.16.200.254:55432->172.16.200.2:443(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.200.2:443->172.16.200.254:55432(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
src_mac=08:5b:0e:7d:42:db
misc=0 policy_id=4294967295 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=1
serial=0000009d tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=00000000
no_ofld_reason:  local
```

Example 2: local-out traffic shaping

In this example, the traffic shaping policy applies to local-out traffic. The local-out traffic originates from port2 on the FortiGate and is destined to an external web server.



To configure the traffic shaping policy:

```
 config firewall shaping-policy
    edit 3
        set traffic-type local-out
        set service "ALL"
        set schedule "always"
        set class-id 2
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

To verify the configuration:

1. Check the shaping policy information for local-out traffic to verify that the correct class ID (2) is applied:

```
# diagnose firewall iprope list 100019
policy index=3 uuid_idx=1301 action=accept
flag (0):
schedule(always)
cos_fwd=0  cos_rev=0
group=00100019 av=00000000 au=00000000 split=00000000
host=1 chk_client_info=0x0 app_list=0 ips_view=0
misc=0
zone(1): 0 -> zone(1): 0
```

```
source(1): 0.0.0.0-255.255.255.255, uuid_idx=1106,
dest(1): 0.0.0.0-255.255.255.255, uuid_idx=1106,
service(1):
        [0:0x0:0/(0,65535)->(0,65535)] flags:0 helper:auto
class_id: 2
```

2. Check the session list to verify that the class ID (2) matches the shaping policy ID (3):

```
# diagnose sys session list
session info: proto=6 proto_state=05 duration=40 expire=110 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=2 shaping_policy_id=3 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=255/255
state=log local
statistic(bytes/packets/allow_err): org=3676/14/1 reply=3848/11/1 tuples=2
tx speed(Bps/kbps): 90/0 rx speed(Bps/kbps): 94/0
orgin->sink: org out->post, reply pre->in dev=34->17/17->34 gwy=0.0.0.0/172.16.200.2
hook=out dir=org act=noop 172.16.200.2:19178->140.174.22.68:443(0.0.0.0:0)
hook=in dir=reply act=noop 140.174.22.68:443->172.16.200.2:19178(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
dst_mac=08:5b:0e:7d:42:db
misc=0 policy_id=0 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=1
serial=00000f1b tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=00000000
no_ofld_reason:  local
```

VLAN CoS matching on a traffic shaping policy

A FortiGate can use the class of service (CoS) value of VLAN packets as a matching criterion for shaping policies. This enables the FortiGate to prioritize traffic based on the CoS value assigned by the switch or router.

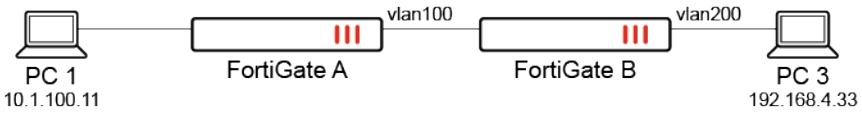
```
config firewall shaping-policy
    edit <id>
        set traffic-type {forwarding | local-in | local-out}
        set cos-mask <3-bit_binary>
        set cos <3-bit_binary>
    next
end
```

`traffic-type {forwarding | local-in | local-out}`	Set the traffic type. • `forwarding`: use forwarding traffic (default) • `local-in`: local-in traffic • `local-out`: local-out traffic
`cos-mask <3-bit_binary>`	Set the VLAN CoS evaluated bits, 3-bit binary (000 - 111). This setting is only available for forwarding traffic.

`cos <3-bit_binary>`	Set the VLAN CoS bit pattern, 3-bit binary (000 - 111). This setting is available once `cos-mask` is configured.

Example

In this example, FortiGate A forwards traffic to FortiGate B with VLAN CoS 3, which matches firewall policy 6. When FortiGate B receives traffic, it applies the traffic shaping policy and will prioritize based on the CoS value.



The VLAN CoS range is 000 to 111 (0 - 7), which includes the following values: 000, 001, 010, 011, 100, 101, 110, and 111. The `cos` and `cos-mask` settings can be used to match multiple `vlan_cos` values with a single shaping policy. The following matching logic is used: (`vlan_cos` AND `cos-mask`) == (`cos` AND `cos-mask`).

To match all possible `vlan_cos` values, set the `cos-mask` to `000`.

To configure VLAN CoS marking with traffic shaping:

1. Configure the firewall policy on FortiGate A with VLAN CoS forwarding:

```
config firewall policy
    edit 6
        set srcintf "port1"
        set dstintf "vlan100"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set vlan-cos-fwd 3
    next
end
```

   Traffic marked with CoS 3 will be forwarded to FortiGate B.

2. On FortiGate A, check the session list to verify that CoS 3 is marked:

```
# diagnose sys session list
session info: proto=1 proto_state=00 duration=1 expire=59 timeout=0 flags=00000000 socktype=0
sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=3/255
state=log may_dirty npu f00
statistic(bytes/packets/allow_err): org=168/2/1 reply=168/2/1 tuples=2
```

```
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=19->47/47->19 gwy=20.20.20.2/10.1.100.11
hook=pre dir=org act=noop 10.1.100.11:28489->192.168.4.33:8(0.0.0.0:0)
hook=post dir=reply act=noop 192.168.4.33:28489->10.1.100.11:0(0.0.0.0:0)
src_mac=00:0c:29:57:2a:01  dst_mac=70:4c:a5:7d:d4:95
misc=0 policy_id=6 pol_uuid_idx=1128 auth_info=0 chk_client_info=0 vd=2
serial=000717ca tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000c00 ofld-O ofld-R
npu info: flag=0x81/0x81, offload=8/8, ips_offload=0/0, epid=79/78, ipid=78/79,
vlan=0x0000/0x0064
vlifid=78/79, vtag_in=0x0000/0x0064 in_npu=2/2, out_npu=2/2, fwd_en=0/0, qid=0/1
```

3. Configure the traffic shaping policy to match VLAN CoS 3:

```
config firewall shaping-policy
    edit 1
        set traffic-type forwarding
        set name "vlan-cos-matching"
        set service "ALL"
        set srcintf "vlan100"
        set dstintf "vlan200"
        set class-id 2
        set cos-mask 111
        set cos 011
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

Based on this shaping policy:

- `vlan_cos` = 3, which corresponds to `011`
  `cos-mask` = `111`
  AND both get `011`
- `cos-mask` = `111`
  `cos` = `011`
  AND both get `011`
- (`vlan_cos` AND `cos-mask`) == (`cos` AND `cos-mask`), so traffic will pass

The shaping policy will match `vlan_cos3`.

4. Configure the firewall policy on FortiGate B:

```
config firewall policy
    edit 3
        set srcintf "vlan100"
        set dstintf "vlan200"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
```

```
        set schedule "always"
        set service "ALL"
        set logtraffic all
    next
end
```

5. On FortiGate B, check the session list to verify that the class ID (2) matches the shaping policy ID (1):

```
# diagnose sys session list
session info: proto=1 proto_state=00 duration=672 expire=59 timeout=0 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=2 shaping_policy_id=1 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=log may_dirty f00
statistic(bytes/packets/allow_err): org=56532/673/1 reply=56532/673/1 tuples=2
tx speed(Bps/kbps): 82/0 rx speed(Bps/kbps): 82/0
orgin->sink: org pre->post, reply pre->post dev=59->61/61->59 gwy=20.20.200.3/20.20.20.1
hook=pre dir=org act=noop 10.1.100.11:28735->192.168.4.33:8(0.0.0.0:0)
hook=post dir=reply act=noop 192.168.4.33:28735->10.1.100.11:0(0.0.0.0:0)
src_mac=90:6c:ac:fb:bb:97  dst_mac=04:d5:90:36:73:3f
misc=0 policy_id=3 pol_uuid_idx=1245 auth_info=0 chk_client_info=0 vd=1
serial=0000160b tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x040000
no_ofld_reason:  non-npu-intf
```

If a particular session matches both the `firewall policy` and `firewall shaping-policy`, then anything configured in the `firewall shaping-policy` overrides whatever was configured in the `firewall policy`.

Traffic shaping profiles

As mentioned in Traffic shaping on page 2108, the three main methods of configuring traffic shaping are:

- Traffic shaping profiles
- Traffic shapers
- Global traffic prioritization

A traffic shaping profile allows traffic shaping to be configured with policing or queuing. Up to 30 classes can be defined, with prioritization and bandwidth limits configured for each class. When queuing is enabled, metrics can be configured for traffic queuing in each class.

Traffic shaping with policing

At the most basic level, policing involves traffic prioritization and bandwidth limits. Traffic prioritization helps categorize traffic into different priority levels: low, medium, high, critical, and top. When bandwidth is limited,

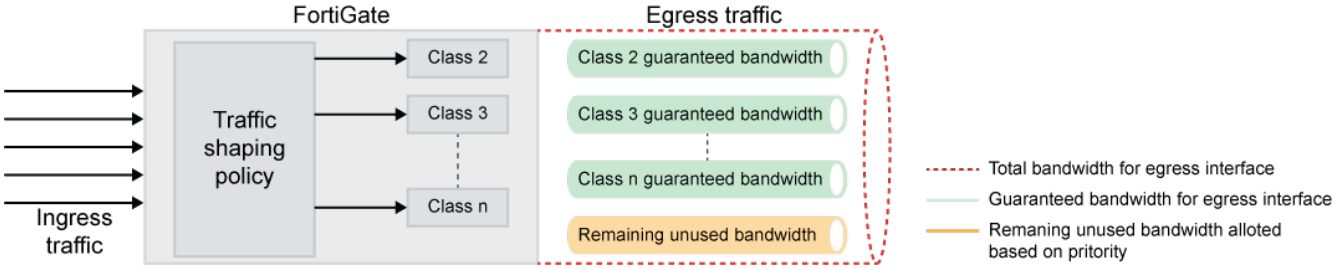
traffic with higher priority levels will take precedence over lower priority traffic. Traffic with lower priority levels that exceeds available bandwidth will be dropped. These levels are only applicable in the context of traffic shaping profiles and should not be confused with global traffic prioritization levels.

Bandwidth limits define the guaranteed and maximum bandwidth allotted to each traffic class. These limits are configured as a percentage of the outbandwidth, which is the outbound bandwidth configured on an interface.

Guaranteed bandwidth limits guarantee the minimum bandwidth that is allotted to a given class of traffic. The sum of all guaranteed bandwidth of all classes within a traffic shaping profile cannot exceed 100%. However, the sum of all guaranteed bandwidth does not need to add up to 100%. The guaranteed bandwidth is always respected, even if one class has lower priority than another.

Maximum bandwidth limits define the maximum percentage of the outbandwidth that a traffic class can use up. This value often will be 100%, given that when there is no other traffic going through other classes, you would want to fully utilize the bandwidth of the outbound link. Traffic throughput exceeding the maximum bandwidth will be dropped.

The following diagram illustrates ingress traffic and how the FortiGate assigns classes and bandwidth to each class.



When comparing traffic shaping profiles and traffic shapers, it is important to remember that guaranteed and maximum bandwidth in a traffic shaping profile is a percentage of the outbandwidth, while guaranteed and maximum bandwidth in a traffic shaper is a rate (Kbps, Mbps, and so on). As long as the outbandwidth is true to its measurement, the bandwidth usage should not exceed the available bandwidth of a link when using a traffic shaping profile.

Congestion occurs when actual traffic surpasses the outbandwidth limit. At this point, traffic prioritization helps determine which traffic will be prioritized over others. First, the guaranteed bandwidth limit is allocated for each class. The left over bandwidth is allocated to traffic classes based on priority. The traffic classes with the highest priority can use as much of the remaining bandwidth as needed. Then, the remaining bandwidth can be allocated to classes at the next priority level, and so forth.

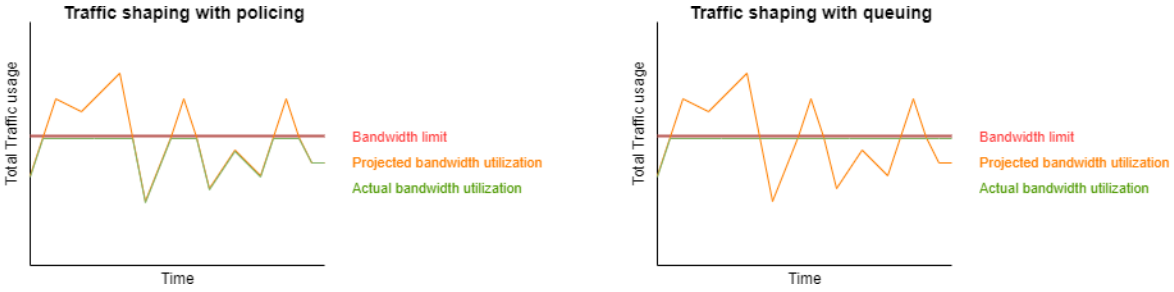
To see examples of applied traffic prioritization and bandwidth limits, see the debugs in Verifying that the traffic is being shaped on page 2125.

Traffic shaping with queuing

When traffic congestion occurs and if there is no queuing, then the excess packets are dropped. With queuing, when traffic exceeds the configured bandwidth limits, the traffic is delayed for transport until bandwidth frees up. Traffic may still be dropped if the queues are full.

In queuing, before a packet egresses an interface, it is first enqueued using an algorithm, such as random early detection (RED) or first in, first out (FIFO). The kernel then dequeues the packet based on the HTB algorithm before sending it out. Queuing can be configured per shaping profile, and it can be customized per class.

The following diagram shows how traffic policing differs from traffic queuing by comparing the bandwidth limit, projected bandwidth utilization, and actual bandwidth utilization.



For more information about traffic shaping with queuing, see Traffic shaping with queuing using a traffic shaping profile on page 2127.

Configuring traffic shaping profiles

The main steps to configure traffic shaping are:

1. Configure the traffic shaping policy, and assign matched traffic to a class (see Traffic shaping policies on page 2110).
2. Configure the traffic shaping profile and apply traffic bandwidth, prioritization and/or queuing per class.
3. Configure the interface outbandwidth and apply an egress shaping profile to the interface.

Configuring the traffic shaping profile

A traffic shaping profile consists of the class ID and the settings per class ID. It also defines the type of traffic shaping to apply (policing or queuing) and the default class ID for traffic that does not match any traffic shaping policies.

A class can be configured in the GUI as part of a traffic shaping profile or policy. In the CLI, a traffic class must be defined before it can be assigned within a traffic shaping profile. Class IDs range from 2 - 31, and they can be reused between different traffic shaping profiles.

- When NPU offloading is enabled on FortiGates with NP6, NP6Lite (SoC3), or NP6XLite (SoC4) processors, the class ID limit for egress traffic is 2 - 15.
- When NPU offloading is enabled on FortiGates with NP7 or NP7Lite (SoC5) processors, the class ID limit for egress traffic is 2–11.
- When NPU offloading is disabled, the class ID limit for egress traffic is 2–31.

Setting the egress traffic class ID outside of these limits can result in unexpected behavior.

When configuring a traffic shaping profile, the settings can be defined per class.

The following options can be configured for traffic shaping classes:

GUI option	CLI option	Description
Default	`set default-class-id <class-id>`	Set the default class ID.

GUI option	CLI option	Description
		Each profile must have one default class ID. The default class ID can be changed at any time.
Traffic shaping class ID	`set class-id <integer>`	Set the class ID (2 - 31).
Guaranteed bandwidth	`set guaranteed-bandwidth-percentage <integer>`	Set the percentage of the outbandwidth that will be guaranteed for the class ID.
Maximum bandwidth	`set maximum-bandwidth-percentage <integer>`	Set the percentage of the outbandwidth that will be the maximum bandwidth for the class ID.
Priority	`set priority {top | critical | high | medium | low}`	Select the priority level for the class ID.

To configure a traffic shaping profile in the GUI:

1. Go to *Policy & Objects* > *Traffic Shaping*, select the *Traffic Shaping Profile* tab, and click *Create New*.
2. Enter the profile name, and optionally enter a comment.
3. In the *Traffic Shaping Classes* section, click *Create New*.
4. Configure the traffic shaping class ID settings (*Traffic shaping class ID*, *Guaranteed bandwidth*, *Maximum bandwidth*, and *Priority*).
5. Click *OK*.
6. Create more shaping classes as needed (the total guaranteed bandwidth of all classes cannot exceed 100%).
7. Click *OK*.

To configure a traffic shaping profile in the CLI:

1. Configure the shaping class:

```
config firewall traffic-class
    edit <integer>
        set class-name <string>
    next
end
```

2. Configure the shaping profile:

```
config firewall shaping-profile
    edit <name>
        set type {policing | queuing}
        set default-class-id <class-id>
        config shaping-entries
            edit <id>
                set class-id <integer>
                set priority {top | critical | high | medium | low}
                set guaranteed-bandwidth-percentage <integer>
                set maximum-bandwidth-percentage <integer>
```

```
            next
        end
    next
end
```

Configuring the interface outbandwidth

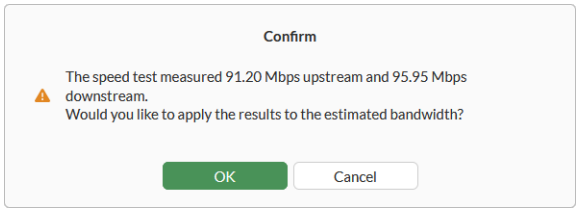
There are two settings that must be configured on an interface that has traffic shaping applied to egressing traffic: a traffic shaping profile must be assigned, and the outbound bandwidth must be configured.

Since traffic shaping is often configured on the WAN interface for egressing traffic, the outbound bandwidth is effectively the upstream bandwidth allowed by your ISP. On the FortiGate, it is possible to perform a speed test on interfaces are assigned a WAN role assigned (see GUI speed test on page 1239). The speed test performs measurements against public cloud servers, and provides an accurate measurement of the upstream bandwidth. After the test is complete, the results can be used to populate the *Outbound bandwidth* field.

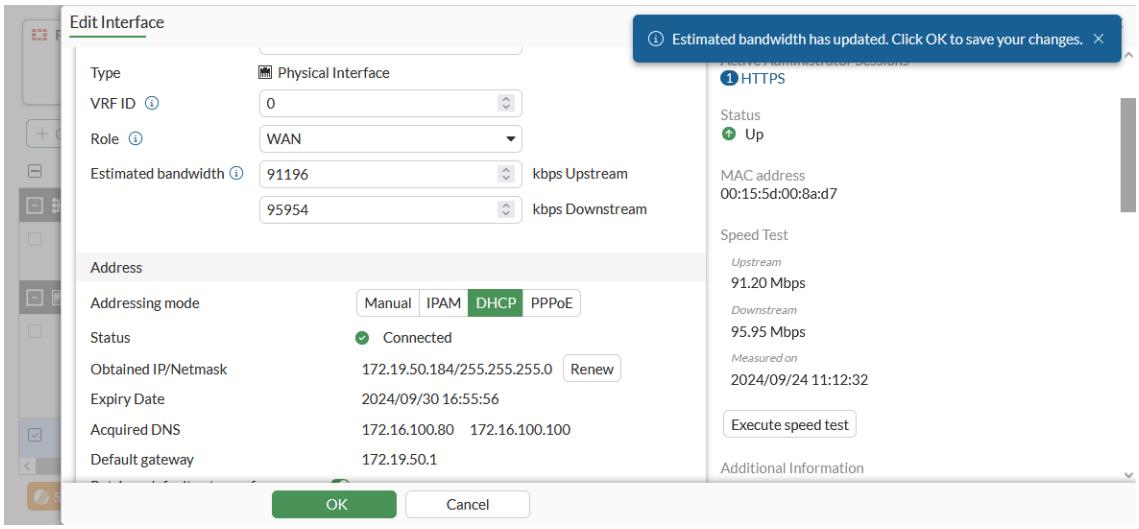
Changes to outbandwidth or egress shaping profiles on a physical or VLAN interface do not take effect for IPsec tunnels or sessions that are already established and offloaded by NP7 or NP7Lite (SoC5) processors. To apply the updated egress shaping settings, you must manually flush or reinstall the affected IPsec SAs and clear any offloaded sessions. Doing this rebuilds the IPsec tunnel and associated sessions using the new interface shaping configuration. You can configure FortiGates with NP7Lite (SoC5) processors to automatically flush or reinstall the affected IPsec SAs and clear any offloaded sessions after changing the configuration of an outbandwidth or egress shaping profile, see Automatic NP7Lite (SOC5) egress shaping profile refresh.

To configure traffic shaping on an interface:

1. Go to *Network* > *Interfaces* and double-click an interface to edit it.
2. For interfaces assigned a *WAN* role, in the right-side of the screen, click *Execute speed test*.
3. When the test completes, click *OK* in the *Confirm* pane to apply the results to the estimated bandwidth.



The speed test results are populated in the *Estimated bandwidth* fields for *kbps Upstream* and *kbps Downstream*.



4. In the *Traffic Shaping* section, enable *Outbound shaping profile* and select a profile.
5. Enable *Outbound bandwidth* and copy the *kbps Upstream* value from the speed test, or enter a custom value.
6. Click *OK*.

Verifying that the traffic is being shaped

In this example, three traffic classes are defined in the traffic shaping profile assigned to port1. The outbandwidth configured on port1 is 1000 Kbps. Each class has an `allocated-bandwidth`, `guaranteed-bandwidth`, `max-bandwidth`, and `current-bandwidth` value.

- The `guaranteed-bandwidth` and `max-bandwidth` are rates that are converted from the percentage of outbandwidth configured for each class. For example, `class-id` 2 has 10% `guaranteed-bandwidth`, equivalent to 100 Kbps, and 100% `max-bandwidth` equivalent to 1000 Kbps.
- The `allocated-bandwidth` displays the real-time bandwidth allocation for the traffic class based on all available factors. This value changes as traffic demand changes.
- The `current-bandwidth` displays the real-time bandwidth usage detected for the traffic class.

To verify that traffic is being shaped by the traffic shaping profile:

1. Enable debug flow to view the live traffic as it matches a traffic shaping policy:

```
# diagnose debug flow show function-name enable
# diagnose debug flow show iprope enable
# diagnose debug flow filter <filters>
# diagnose debug flow trace start <repeat_number>
# diagnose debug enable
```

The `iprope_shaping_check` function outputs the shaping policy matched for any given traffic:

```
...
id=20085 trace_id=21 func=iprope_shaping_check line=934 msg="in-[port3], out-[port1], skb_
flags-02000000, vid-0"
id=20085 trace_id=21 func=__iprope_check line=2277 msg="gnum-100015, check-ffffffffa002a8fe"
```

```
id=20085 trace_id=21 func=__iprope_check_one_policy line=2029 msg="checked gnum-100015 policy-
3, ret-matched, act-accept"
id=20085 trace_id=21 func=__iprope_check_one_policy line=2247 msg="policy-3 is matched, act-
accept"
id=20085 trace_id=21 func=__iprope_check line=2294 msg="gnum-100015 check result: ret-matched,
act-accept, flag-00000000, flag2-00000000"
```

2. Display the session list:

```
# diagnose sys session filter <filters>
# diagnose sys session list
```

Sessions that match a shaping policy will display `class_id` and `shaping_policy_id` fields:

```
...
session info: proto=6 proto_state=05 duration=32 expire=0 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=4
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=4 shaping_policy_id=3 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
```

3. Display the interface statistics:

```
# diagnose netlink interface list port1
if=port1 family=00 type=1 index=3 mtu=1500 link=0 master=0
ref=95 state=start present fw_flags=2001b800 flags=up broadcast run allmulti multicast
Qdisc=pfifo_fast hw_addr=52:54:00:7e:af:a6 broadcast_addr=ff:ff:ff:ff:ff:ff
inbandwidth=10000(kbps)         total_bytes=2098887K    drop_bytes=7854K
egress traffic control:
        bandwidth=1000(kbps) lock_hit=241 default_class=3 n_active_class=3
        class-id=2      allocated-bandwidth=140(kbps)   guaranteed-bandwidth=100(kbps)
                        max-bandwidth=1000(kbps)        current-bandwidth=147(kbps)
                        priority=low    forwarded_bytes=8161K
                        dropped_packets=2032    dropped_bytes=3074K
        class-id=3      allocated-bandwidth=30(kbps)    guaranteed-bandwidth=300(kbps)
                        max-bandwidth=1000(kbps)        current-bandwidth=10(kbps)
                        priority=medium         forwarded_bytes=501K
                        dropped_packets=1       dropped_bytes=1195
        class-id=4      allocated-bandwidth=830(kbps)   guaranteed-bandwidth=500(kbps)
                        max-bandwidth=1000(kbps)        current-bandwidth=810(kbps)
                        priority=high   forwarded_bytes=1393K
                        dropped_packets=379     dropped_bytes=572K
stat: rxp=8349728 txp=11101735 rxb=2216101183 txb=1394077978 rxe=0 txe=0 rxd=0 txd=0 mc=0
collision=0 @ time=1654202868
re: rxl=0 rxo=0 rxc=0 rxf=0 rxfi=0 rxm=0
te: txa=0 txc=0 txfi=0 txh=0 txw=0
misc rxc=0 txc=0
input_type=0 state=3 arp_entry=0 refcnt=95
```

If the debug output does not display egress traffic control by class and displays them by priority, it is likely that global traffic prioritization is configured. The global traffic prioritization settings must be disabled to view the preceding debug output (see Global traffic prioritization on page 2148).

Traffic shaping with queuing using a traffic shaping profile

You can use the weighted random early detection (WRED) queuing function within traffic shaping.

This topic includes two parts:

- Traffic shaping with queuing on page 2127
- Burst control in queuing mode on page 2128

You cannot configure or view WRED in the GUI; you must use the CLI.

WRED is not supported when traffic is offloaded to an NPU.

Traffic shaping with queuing

Traffic shaping has a queuing option. Use this option to fine-tune the queue by setting the profile queue size or performing random early drop (RED) according to queue usage.

This example shows setting the profile queue size limit to 5 so that the queue can contain a maximum of five packets and more packets are dropped.

To set the profile queue size limit:

```
config firewall shaping-profile
    edit "profile"
        set type queuing
        set default-class-id 31
        config shaping-entries
            edit 31
                set class-id 31
                set guaranteed-bandwidth-percentage 5
                set maximum-bandwidth-percentage 10
                set limit 5  <range from 5 to 10000; default: 1000>
            next
        end
    next
end
```

This example shows performing RED according to queue usage by setting `red-probability`, `min`, and `max`. Setting `red-probability` to `10` means start to drop packets when queue usage reaches the `min` setting. When queue usage reaches the `max` setting, drop 10% of the packets.

- Level 1: when queue is less than `min` packets, drop 0% of packets.
- Level 2: when queue reaches `min` packets, start to drop packets.
- Level 3: when queue usage is between `min` and `max` packets, drop 0–10% of packets by proportion.

- Level 4: when queue (average queue size) is more than `max` packets, drop 100% of packets.

To set RED according to queue usage:

```
config firewall shaping-profile
    edit "profile"
        set type queuing
        set default-class-id 31
        config shaping-entries
            edit 31
                set class-id 31
                set guaranteed-bandwidth-percentage 5
                set maximum-bandwidth-percentage 10
                set red-probability 10 <range from 0 to 20; default: 0 no drop>
                set min 100 <range from 3 to 3000>
                set max 300 <range from 3 to 3000>
            next
        end
    next
end
```

To troubleshoot this function, use the following diagnose commands:

```
diagnose netlink intf-class list <intf>
diagnose netlink intf-qdisc list <intf>
```

Burst control in queuing mode

In a hierarchical token bucket (HTB) algorithm, each traffic class has buckets to allow a burst of traffic. The maximum burst is determined by the bucket size `burst` (for guaranteed bandwidth) and `cburst` (for maximum bandwidth). The shaping profile has `burst-in-msec` and `cburst-in-msec` parameters for each shaping entry (`class id`) to control the bucket size.

This example uses the outbandwidth of the interface as 1 Mbps and the maximum bandwidth of class is 50%.

burst = burst-in-msec * guaranteed bandwidth = 100 ms × 1 Mbps x 50% = 50000 b = 6250 B

cburst = cburst-in-msec * maximum bandwidth = 200 ms × 1 Mbps x 50% = 100000 b = 12500 B

The following example sets `burst-in-msec` to 100 and `cburst-in-msec` to 200.

To set burst control in queuing mode:

```
config firewall shaping-profile
    edit "profile"
        set type queuing
        set default-class-id 31
        config shaping-entries
            edit 31
                set class-id 31
                set guaranteed-bandwidth-percentage 5
                set maximum-bandwidth-percentage 50
```

```
                set burst-in-msec 100 <range from 0 to 2000>
                set cburst-in-msec 200 <range from 0 to 2000>
            next
        end
    next
end
```

Example

Enabling RED for FTP traffic from QA

This example shows how to enable RED for FTP traffic from QA. This example sets a maximum of 10% of the packets to be dropped when queue usage reaches the maximum value.

To configure the firewall address:

```
config firewall address
    edit QA_team
        set subnet 10.1.100.0/24
    next
end
```

To set the shaping policy to classify traffic into different class IDs:

```
config firewall shaping-policy
    edit 1
        set service HTTPS HTTP
        set dstintf port1
        set srcaddr QA_team
        set dstaddr all
        set class-id 10
    next
    edit 2
        set service FTP
        set dstintf port1
        set srcaddr QA_team
        set dstaddr all
        set class-id 20
    next
end
```

To set the shaping policy to define the speed of each class ID:

```
config firewall shaping-profile
    edit QA_team_profile
       set type queuing
       set default-class-id 30
            config shaping-entries
                edit 1
```

```
                    set class-id 10
                    set guaranteed-bandwidth-percentage 50
                    set maximum-bandwidth-percentage 100
                next
                edit 2
                    set class-id 20
                    set guaranteed-bandwidth-percentage 30
                    set maximum-bandwidth-percentage 60
                    set red-probability 10
                next
                edit 3
                    set class-id 30
                    set guaranteed-bandwidth-percentage 20
                    set maximum-bandwidth-percentage 50
                next
            end
    next
end
```

To apply the shaping policy to the interface:

```
config sys interface
    edit port1
        set outbandwidth 10000
        set egress-shaping-profile QA_team_profile
    next
end
```

To use diagnose commands to troubleshoot:

```
# diagnose netlink intf-class list port1
class htb 1:1 root rate 1250000Bps ceil 1250000Bps burst 1600B/8 mpu 0B overhead 0B cburst 1600B/8
mpu 0B overhead 0B level 7 buffer [00004e20] cbuffer [00004e20]
 Sent 11709 bytes 69 pkt (dropped 0, overlimits 0 requeues 0)
 rate 226Bps 2pps backlog 0B 0p
 lended: 3 borrowed: 0 giants: 0
 tokens: 18500 ctokens: 18500
class htb 1:10 parent 1:1 leaf 10: prio 1 quantum 62500 rate 625000Bps ceil 1250000Bps burst
1600B/8 mpu 0B overhead 0B cburst 1600B/8 mpu 0B overhead 0B level 0 buffer [00009c40] cbuffer
[00004e20]
 Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
 rate 0Bps 0pps backlog 0B 0p
 lended: 0 borrowed: 0 giants: 0
 tokens: 40000 ctokens: 20000
class htb 1:20 parent 1:1 leaf 20: prio 1 quantum 37500 rate 375000Bps ceil 750000Bps burst
1599B/8 mpu 0B overhead 0B cburst 1599B/8 mpu 0B overhead 0B level 0 buffer [0001046a] cbuffer
[00008235]
 Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
 rate 0Bps 0pps backlog 0B 0p
 lended: 0 borrowed: 0 giants: 0
 tokens: 66666 ctokens: 33333
```

```
class htb 1:30 parent 1:1 leaf 30: prio 1 quantum 25000 rate 250000Bps ceil 625000Bps burst
1600B/8 mpu 0B overhead 0B cburst 1600B/8 mpu 0B overhead 0B level 0 buffer [000186a0] cbuffer
[00009c40]
 Sent 11709 bytes 69 pkt (dropped 0, overlimits 0 requeues 0)
 rate 226Bps 2pps backlog 0B 0p
 lended: 66 borrowed: 3 giants: 0
 tokens: 92500 ctokens: 37000
class red 20:1 parent 20:0
```

```
 # diagnose netlink intf-qdisc list port1
qdisc htb 1: root refcnt 5 r2q 10 default 30 direct_packets_stat 0 ver 3.17
 Sent 18874 bytes 109 pkt (dropped 0, overlimits 5 requeues 0)
 backlog 0B 0p
qdisc pfifo 10: parent 1:10 refcnt 1 limit 1000p
 Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
 backlog 0B 0p
qdisc red 20: parent 1:20 refcnt 1 limit 4000000B min 300000B max 1000000B ewma 9 Plog 23 Scell_
log 20 flags 0
 Sent 0 bytes 0 pkt (dropped 0, overlimits 0 requeues 0)
 backlog 0B 0p
  marked 0 early 0 pdrop 0 other 0
qdisc pfifo 30: parent 1:30 refcnt 1 limit 1000p
 Sent 18874 bytes 109 pkt (dropped 0, overlimits 0 requeues 0)
 backlog 0B 0p
```

Traffic shapers

The following topics provide more information about traffic shapers:

- Shared traffic shaper on page 2131
- Per-IP traffic shaper on page 2136
- Changing traffic shaper bandwidth unit of measurement on page 2140
- Multi-stage DSCP marking and class ID in traffic shapers on page 2141
- Multi-stage VLAN CoS marking on page 2143
- Adding traffic shapers to multicast policies on page 2147

Shared traffic shaper

Shared traffic shaper is used in a firewall shaping policy to indicate the priority and guaranteed and maximum bandwidth for a specified type of traffic use.

The maximum bandwidth indicates the largest amount of traffic allowed when using the policy. You can set the maximum bandwidth to a value between 1 and 16776000 Kbps. The GUI displays an error if any value outside this range is used. If you want to allow unlimited bandwidth, use the CLI to enter a value of 0.

The guaranteed bandwidth ensures that there is a consistent reserved bandwidth available. When setting the guaranteed bandwidth, ensure that the value is significantly less than the interface's bandwidth capacity.

Otherwise, the interface will allow very little or no other traffic to pass through, potentially causing unwanted latency.

In a shared traffic shaper, the administrator can prioritize certain traffic as high, medium, or low. FortiOS provides bandwidth to low priority connections only when high priority connections do not need the bandwidth. For example, you should assign a high traffic priority to a policy for connecting a secure web server that needs to support e-commerce traffic. You should assign less important services a low priority.

When you configure a shared traffic shaper, you can apply bandwidth shaping per policy or for all policies. By default, a shared traffic shaper applies traffic shaping evenly to all policies that use the shared traffic shaper.

When configuring a per-policy traffic shaper, FortiOS applies the traffic shaping rules defined for each security policy individually. For example, if a per-policy traffic shaper is configured with a maximum bandwidth of 1000 Kbps, any security policies that have that traffic shaper enabled get 1000 Kbps of bandwidth each.

If a traffic shaper for all policies is configured with a maximum bandwidth of 1000 Kbps, all policies share the 1000 Kbps on a first-come, first-served basis.

The configuration is as follows:

```
config firewall shaper traffic-shaper
    edit "traffic_shaper_name"
        set per-policy enable
    next
end
```

The shared traffic shaper selected in the traffic shaping policy affects traffic in the direction defined in the policy. For example, if the source port is LAN and the destination is WAN1, the traffic shaping affects the flow in this direction only, affecting the outbound traffic's upload speed. You can define the traffic shaper for the policy in the opposite direction (reverse shaper) to affect the inbound traffic's download speed. In this example, that would be from WAN1 to LAN.

Only traffic through forward traffic shapers will be included in FortiView; reverse and per-IP shapers are not included.

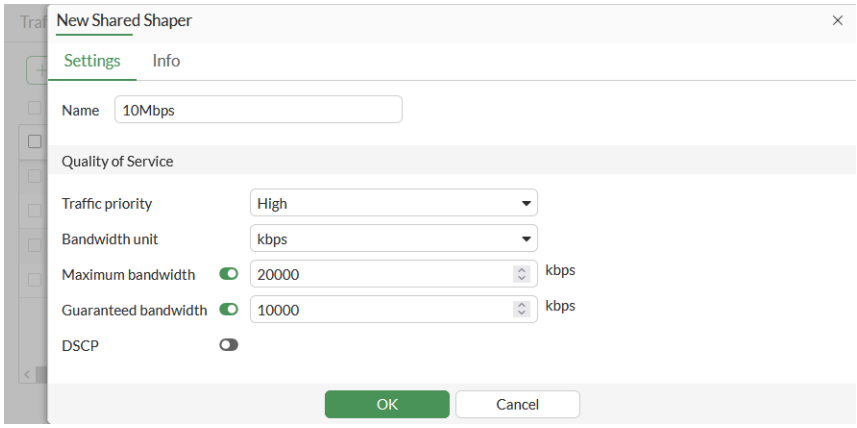
Traffic shapers can be added to a multicast policy when multicast routing is enabled.

The following example shows how to apply different speeds to different types of service. The example configures two shared traffic shapers to use in two firewall shaping policies. One policy guarantees a speed of 10 Mbps for VoIP traffic. The other policy guarantees a speed of 1 Mbps for other traffic. In the example, FortiOS communicates with a PC using port10 and the Internet using port9.

To configure shared traffic shapers in the GUI:

1. Create a firewall policy:
   a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
   b. Set the *Name* to *Internet Access*.
   c. Set the *Incoming Interface* to *port10*.
   d. Set the *Outgoing Interface* to *port9*.
   e. Set the *Source* and *Destination* to *all*.
   f. Set the *Schedule* to *always*.
   g. Set the *Service* to *ALL*.
   h. Click *OK*.

2. Create the shared traffic shapers:
   a. Go to *Policy & Objects* > *Traffic Shaping*, select the *Traffic Shaper* tab, then click *Create New*.
   b. Set the *Name* to *10Mbps*. This shaper is for VoIP traffic.
   c. Set the *Traffic Priority* to *High*.
   d. Enable *Max Bandwidth* and enter *20000*.
   e. Enable *Guaranteed Bandwidth* and enter *10000*.



   f. Click *OK*.
   g. Repeat the above steps to create another traffic shaper named *1Mbps* with the *Traffic Priority* set to *Low*, the *Max Bandwidth* set to *10000*, and the *Guaranteed Bandwidth* set to *1000*.
3. Create a firewall shaping policy:
   a. Go to *Policy & Objects* > *Traffic Shaping*, select the *Traffic Shaping Policy* tab, and click *Create New*.
   b. Set the *Name* to *VoIP_10Mbps_High*. This policy is for VoIP traffic.
   c. Set the *Source* and *Destination* to *all*.
   d. Set the *Service* to all VoIP services.
   e. Set the *Outgoing Interface* to *port9*.
   f. Enable *Shared shaper* and select *10Mbps*.
   g. Enable *Reverse shaper* and select *10Mbps*.
   h. Click *OK*.
   i. Repeat the above steps to create another firewall shaping policy named *Other_1Mbps_Low* for other traffic, with the *Source* and *Destination* set to *all*, *Service* set to *ALL*, *Outgoing Interface* set to *port9*, and *Shared shaper* and *Reverse shaper* set to *1Mbps*.

To configure shared traffic shapers in the CLI:

1. Create a firewall policy:

```
config firewall policy
    edit 1
        set name "Internet Access"
        set srcintf "port10"
        set dstintf "port9"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
```

```
        set schedule "always"
        set service "ALL"
        set fsso disable
        set nat enable
    next
end
```

2. Create the shared traffic shapers:

```
config firewall shaper traffic-shaper
    edit "10Mbps"
        set guaranteed-bandwidth 10000
        set maximum-bandwidth 20000
    next
    edit "1Mbps"
        set guaranteed-bandwidth 1000
        set maximum-bandwidth 10000
        set priority low
    next
end
```

3. Create a firewall shaping policy:

```
config firewall shaping-policy
    edit 1
        set name "VOIP_10Mbps_High"
        set service "H323" "IRC" "MS-SQL" "MYSQL" "RTSP" "SCCP" "SIP" "SIP-MSNmessenger"
        set dstintf "port9"
        set traffic-shaper "10Mbps"
        set traffic-shaper-reverse "10Mbps"
        set srcaddr "all"
        set dstaddr "all"
    next
    edit 2
        set name "Other_1Mbps_Low"
        set service "ALL"
        set dstintf "port9"
        set traffic-shaper "1Mbps"
        set traffic-shaper-reverse "1Mbps"
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

To troubleshoot shared traffic shapers:

1. Check if specific traffic is attached to the correct traffic shaper. The example output shows the traffic attached to the 10Mbps and 1Mbps shapers:

```
# diagnose firewall iprope list 100015
policy index=1 uuid_idx=0 action=accept
flag (0):
```

```
shapers: orig=10Mbps(2/1280000/2560000)
cos_fwd=0  cos_rev=0
group=00100015 av=00000000 au=00000000 split=00000000
host=4 chk_client_info=0x0 app_list=0 ips_view=0
misc=0 dd_type=0 dd_mode=0
zone(1): 0 -> zone(1): 38
source(1): 0.0.0.0-255.255.255.255, uuid_idx=0,
dest(1): 0.0.0.0-255.255.255.255, uuid_idx=0,
service(15):
    [6:0x0:0/(1,65535)->(1720,1720)] helper:auto
    [6:0x0:0/(1,65535)->(1503,1503)] helper:auto
    [17:0x0:0/(1,65535)->(1719,1719)] helper:auto
    [6:0x0:0/(1,65535)->(6660,6669)] helper:auto
    [6:0x0:0/(1,65535)->(1433,1433)] helper:auto
    [6:0x0:0/(1,65535)->(1434,1434)] helper:auto
    [6:0x0:0/(1,65535)->(3306,3306)] helper:auto
    [6:0x0:0/(1,65535)->(554,554)] helper:auto
    [6:0x0:0/(1,65535)->(7070,7070)] helper:auto
    [6:0x0:0/(1,65535)->(8554,8554)] helper:auto
    [17:0x0:0/(1,65535)->(554,554)] helper:auto
    [6:0x0:0/(1,65535)->(2000,2000)] helper:auto
    [6:0x0:0/(1,65535)->(5060,5060)] helper:auto
    [17:0x0:0/(1,65535)->(5060,5060)] helper:auto
    [6:0x0:0/(1,65535)->(1863,1863)] helper:auto

policy index=2 uuid_idx=0 action=accept
flag (0):
shapers: orig=1Mbps(4/128000/1280000)
cos_fwd=0  cos_rev=0
group=00100015 av=00000000 au=00000000 split=00000000
host=4 chk_client_info=0x0 app_list=0 ips_view=0
misc=0 dd_type=0 dd_mode=0
zone(1): 0 -> zone(1): 38
source(1): 0.0.0.0-255.255.255.255, uuid_idx=0,
dest(1): 0.0.0.0-255.255.255.255, uuid_idx=0,
service(1):
    [0:0x0:0/(0,0)->(0,0)] helper:auto
```

2. Check if the correct traffic shaper is applied to the session. The example output shows that the 1Mbps shaper is applied to the session:

```
# diagnose sys session list
session info: proto=6 proto_state=01 duration=11 expire=3599 timeout=3600 flags=00000000
sockflag=00000000 sockport=0 av_idx=0 use=5
origin-shaper=1Mbps prio=4 guarantee 128000Bps max 1280000Bps traffic 1050Bps drops 0B
reply-shaper=
per_ip_shaper=
class_id=0 shaping_policy_id=2 ha_id=0 policy_dir=0 tunnel=/ helper=ftp vlan_cos=0/255
state=may_dirty npu npd os mif route_preserve
statistic(bytes/packets/allow_err): org=868/15/1 reply=752/10/1 tuples=2
tx speed(Bps/kbps): 76/0 rx speed(Bps/kbps): 66/0
orgin->sink: org pre->post, reply pre->post dev=39->38/38->39 gwy=172.16.200.55/0.0.0.0
```

```
hook=post dir=org act=snat 10.1.100.11:58241->172.16.200.55:21(172.16.200.1:58241)
hook=pre dir=reply act=dnat 172.16.200.55:21->172.16.200.1:58241(10.1.100.11:58241)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=1 auth_info=0 chk_client_info=0 vd=4
serial=0003255f tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id = 00000000
dd_type=0 dd_mode=0
npu_state=0x100000
npu info: flag=0x00/0x00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0, vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
no_ofld_reason:  offload-denied helper
total session 1
```

3. Check the statuses of shared traffic shapers:

```
# diagnose firewall shaper traffic-shaper list
name 10Mbps
maximum-bandwidth 2500 KB/sec
guaranteed-bandwidth 1250 KB/sec
current-bandwidth 0 B/sec
priority 2
tos ff
packets dropped 0
bytes dropped 0

name 1Mbps
maximum-bandwidth 1250 KB/sec
guaranteed-bandwidth 125 KB/sec
current-bandwidth 0 B/sec
priority 4
tos ff
packets dropped 0
bytes dropped 0
```

Per-IP traffic shaper

With per-IP traffic shaping, you can limit each IP address's behavior to avoid a situation where one user uses all of the available bandwidth. In addition to controlling the maximum bandwidth used per IP address, you can also define the maximum number of concurrent sessions for an IP address. For example, if you apply a per-IP shaper of 1 Mbps to your entire network, FortiOS allocates each user/IP address 1 Mbps of bandwidth. Even if the network consists of a single user, FortiOS allocates them 1 Mbps. If there are ten users, each user gets 1 Mbps of bandwidth, totaling 10 Mbps of outgoing traffic.

For shared shapers, all users share the set guaranteed and maximum bandwidths. For example, if you set a shared shaper for all PCs using an FTP service to 10 Mbps, all users uploading to the FTP server share the 10 Mbps.

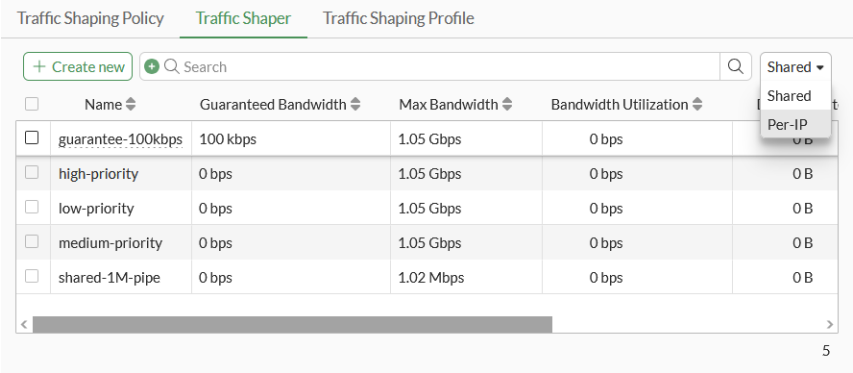
Shared shapers affect upload speed. If you want to limit the download speed from the FTP server in the example, you must configure the shared shaper as a reverse shaper. Per-IP shapers apply the speed limit on

both upload and download operations. Only traffic through forward traffic shapers will be included in FortiView; reverse and per-IP shapers are not included.

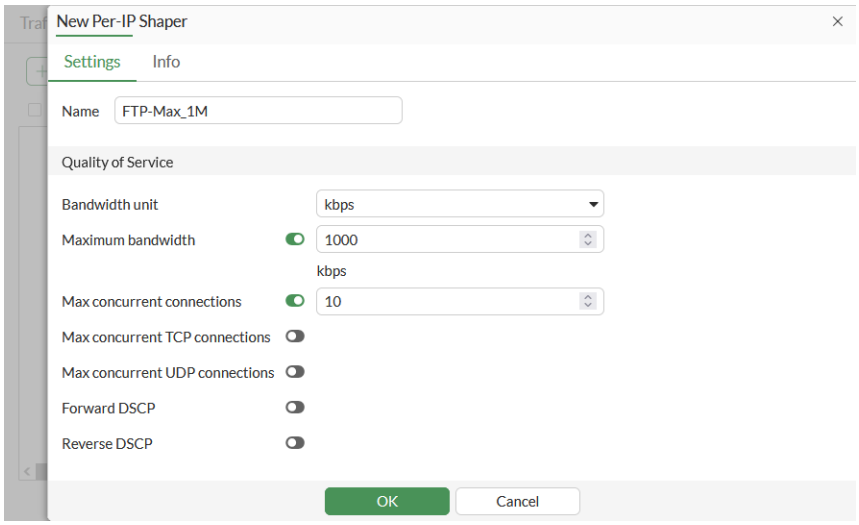
The following example shows how to apply a per-IP shaper to a traffic shaping policy. This shaper assigns each user a maximum bandwidth of 1 Mbps and allows each user to have a maximum of ten concurrent connections to the FTP server. In the example, FortiOS communicates with users using port10 and the FTP server using port9.

To configure a per-IP traffic shaper in the GUI:

1. Create a firewall policy:
   a. Go to *Policy & Objects* > *IPv4 Policy* and click *Create New*.
   b. Set the *Name* to *FTP Access*.
   c. Set the *Incoming Interface* to *port10*.
   d. Set the *Outgoing Interface* to *port9*.
   e. Set the *Source* to *all*.
   f. Set the *Destination* to *FTP_Server*.
   g. Set the *Schedule* to *always*.
   h. Set the *Service* to *ALL*.
   i. Click *OK*.
2. Create the per-IP traffic shaper:
   a. Go to *Policy & Objects* > *Traffic Shaping* and select the *Traffic Shaper* tab.
   b. Select *Per-IP* in the type drop-down, then click *Create new*.



   c. Enter the *Name* (*FTP_Max_1M*). This shaper is for VoIP traffic.
   d. Enable *Max Bandwidth* and enter *1000*.
   e. Enable *Max Concurrent Connections* and enter *10*. This means that each user can have up to ten concurrent connections to the FTP server.



f. Click *OK*.

3. Create a firewall shaping policy:
   a. Go to *Policy & Objects* > *Traffic Shaping*, select the *Traffic Shaping Policy* tab, and click *Create New*.
   b. Enter the *Name* (*FTP speed 1M*).
   c. Set the *Source* to the addresses and users that require access to the FTP server.
   d. Set the *Destination* to *FTP_Server*.
   e. Set the *Service* to *ALL*.
   f. Set the *Outgoing Interface* to *port9*.
   g. Enable *Per-IP shaper* and select *FTP_Max_1M*.
   h. Click *OK*.

To configure a per-IP traffic shaper in the CLI:

1. Create a firewall policy:

```
config firewall policy
    edit 1
        set name "FTP Access"
        set srcintf "port10"
        set dstintf "port9"
        set srcaddr "all"
        set dstaddr "FTP_Server"
        set action accept
        set schedule "always"
        set service "ALL"
        set fsso disable
        set nat enable
    next
end
```

2. Create the per-IP traffic shaper:

```
config firewall shaper per-ip-shaper
    edit "FTP_Max_1M"
```

```
        set max-bandwidth 1000
        set max-concurrent-session 10
    next
end
```

3. Create a firewall shaping policy:

```
config firewall shaping-policy
    edit 1
        set name "FTP speed 1M"
        set service "ALL"
        set dstintf "port9"
        set per-ip-shaper "FTP_Max_1M"
        set srcaddr "PC1" "WinPC" "PC2"
        set dstaddr "FTP_Server"
    next
end
```

To troubleshoot per-IP traffic shapers:

1. Check if specific traffic is attached to the correct traffic shaper. The example output shows the traffic attached to the FTP_Max_1M shaper:

```
# diagnose firewall iprope list 100015
policy index=3 uuid_idx=0 action=accept
flag (0):
shapers:  per-ip=FTP_Max_1M
cos_fwd=0  cos_rev=0
group=00100015 av=00000000 au=00000000 split=00000000
host=2 chk_client_info=0x0 app_list=0 ips_view=0
misc=0 dd_type=0 dd_mode=0
zone(1): 0 -&gt; zone(1): 38
source(3): 10.1.100.11-10.1.100.11, uuid_idx=30, 10.1.100.143-10.1.100.143, uuid_idx=32,
10.1.100.22-10.1.100.22, uuid_idx=31,
dest(1): 172.16.200.55-172.16.200.55, uuid_idx=89,
service(1):
    [0:0x0:0/(0,65535)-&gt;(0,65535)] helper:auto
```

2. Check if the correct traffic shaper is applied to the session. The example output shows that the FTP_Max_1M shaper is applied to the session:

```
# diagnose sys session list
session info: proto=6 proto_state=01 duration=36 expire=3567 timeout=3600 flags=00000000
sockflag=00000000 sockport=0 av_idx=0 use=4
origin-shaper=
reply-shaper=
per_ip_shaper=FTP_Max_1M
class_id=0 shaping_policy_id=3 ha_id=0 policy_dir=0 tunnel=/ helper=ftp vlan_cos=0/255
state=may_dirty per_ip npu npd mif route_preserve
statistic(bytes/packets/allow_err): org=506/9/1 reply=416/6/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin-&gt;sink: org pre-&gt;post, reply pre-&gt;post dev=39-&gt;38/38-&gt;39
```

```
gwy=172.16.200.55/0.0.0.0
hook=post dir=org act=snat 10.1.100.11:58275-&gt;172.16.200.55:21(172.16.200.1:58275)
hook=pre dir=reply act=dnat 172.16.200.55:21-&gt;172.16.200.1:58275(10.1.100.11:58275)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=1 auth_info=0 chk_client_info=0 vd=2
serial=0000211a tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id = 00000000
dd_type=0 dd_mode=0
npu_state=0x100000
npu info: flag=0x00/0x00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0, vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
no_ofld_reason:  offload-denied helper
```

3. Check the statuses of per-IP traffic shapers. The output should resemble the following:

```
# diagnose firewall shaper per-ip-shaper list
name FTP_Max_1M
maximum-bandwidth 125 KB/sec
maximum-concurrent-session 10
tos ff/ff
packets dropped 0
bytes dropped 0
addr=10.1.100.11 status: bps=0 ses=3
```

Changing traffic shaper bandwidth unit of measurement

Bandwidth speeds are measured in kilobits per second (Kbps), and bytes that are sent and received are measured in megabytes (MB). In some cases, this can cause confusion depending on whether your ISP uses kilobits per second (Kbps), kilobytes per second (KBps), megabits per second (Mbps), or gigabits per second (Gbps).

You can change the unit of measurement for traffic shapers in the CLI.

To change the bandwidth unit of measurement for a shared traffic shaper:

```
config firewall shaper traffic-shaper
    edit <traffic_shaper_name>
        set bandwidth-unit {kbps | mbps | gbps}
    next
end
```

To change the bandwidth unit of measurement for a per-IP traffic shaper:

```
config firewall shaper per-ip-shaper
    edit <traffic_shaper_name>
        set bandwidth-unit {kbps | mbps | gbps}
    next
end
```

Multi-stage DSCP marking and class ID in traffic shapers

Traffic shapers have a multi-stage method so that packets are marked with a different differentiated services code point (DSCP) and `class id` at different traffic speeds. Marking packets with a different DSCP code is for the next hop to classify the packets. The FortiGate benefits by marking packets with a different `class id`. Combined with the egress interface shaping profile, the FortiGate can handle the traffic differently according to its `class id`.

Rule	DSCP code	Class ID
speed < guarantee bandwidth	`diffservcode`	`class id` in shaping policy
guarantee bandwidth < speed < exceed bandwidth	`exceed-dscp`	`exceed-class-id`
exceed bandwidth < speed	`maximum-dscp`	`exceed-class-id`

This example sets the following parameters:

- When the current bandwidth is less than 50 Kbps, mark packets with `diffservcode` 100000 and set `class id` to 10.
- When the current bandwidth is between 50 Kbps and 100 Kbps, mark packets with `exceed-dscp` 111000 and set `exceed-class-id` to 20.
- When the current bandwidth is more than 100 Kbps, mark packets with `maximum-dscp` 111111 and set `exceed-class-id` to 20.

To set multi-stage DSCP marking and class ID in a traffic shaper:

```
config firewall shaper traffic-shaper
    edit "50k-100k-150k"
        set guaranteed-bandwidth 50
        set maximum-bandwidth 150
        set diffserv enable
        set dscp-marking-method multi-stage
        set exceed-bandwidth 100
        set exceed-dscp 111000
        set exceed-class-id 20
        set maximum-dscp 111111
        set diffservcode 100000
    next
end
```

```
config firewall shaping-policy
     edit 1
         set service "ALL"
         set dstintf PORT2
         set srcaddr "all"
         set dstaddr "all"
         set class-id 10
     next
end
```

Traffic shapers also have an overhead option that defines the per-packet size overhead used in rate computation.

To set the traffic shaper overhead option:

```
config firewall shaper traffic-shaper
    edit "testing"
        set guaranteed-bandwidth 50
        set maximum-bandwidth 150
        set overhead 14 <range from 0 to 100>
    next
end
```

Example

This example shows how to mark QA traffic with a different DSCP according to real-time traffic speed.

To configure the firewall address:

```
config firewall address
    edit QA_team
        set subnet 10.1.100.0/24
    next
end
```

To configure the firewall shaper traffic shaper:

```
config firewall shaper traffic-shaper
    edit "500k-1000k-1500k"
        set guaranteed-bandwidth 500
        set maximum-bandwidth 1500
        set diffserv enable
        set dscp-marking-method multi-stage
        set exceed-bandwidth 1000
        set exceed-dscp 111000
        set maximum-dscp 111111
        set diffservcode 100000
    next
end
```

```
config firewall shaping-policy
    edit QA_team
        set service "ALL"
        set dstintf port1
        set traffic-shaper "500k-1000k-1500k"
        set traffic-shaper-reverse "500k-1000k-1500k"
        set srcaddr "QA_team"
        set dstaddr "all"
    next
end
```

Multi-stage VLAN CoS marking

A FortiGate can configure the traffic shaper to dynamically change the CoS value of outgoing VLAN packets based on the shaper profile. This allows the FortiGate to mark traffic with different CoS values at different stages of the shaping process.

```
config firewall shaper traffic-shaper
    edit <name>
        set bandwidth-unit {kbps | mbps | gbps}
        set guaranteed-bandwidth <integer>
        set maximum-bandwidth <integer>
        set cos-marking {enable | disable}
        set cos-marking-method {static | multi-stage}
        set cos <3-bit_binary>
        set exceed-cos <3-bit_binary>
        set maximum-cos <3-bit_binary>
        set exceed-bandwidth <integer>
    next
end
```

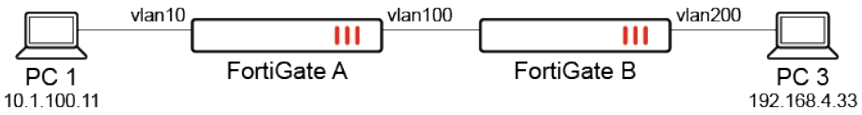
`cos-marking {enable | disable}`	Enable/disable VLAN CoS marking (default = `disable`).
`cos-marking-method {static | multi-stage}`	Set the VLAN CoS marking method. • `static`: use static VLAN CoS marking (default) • `multi-stage`: multi-stage VLAN CoS marking
`cos <3-bit_binary>`	Set the VLAN CoS mark, 3-bit binary (000 - 111).
`exceed-cos <3-bit_binary>`	Set the VLAN CoS mark for traffic in `guaranteed-bandwidth` and `exceed-bandwidth`, 3-bit binary (000 - 111).
`maximum-cos <3-bit_binary>`	Set the VLAN CoS mark for traffic in `exceed-bandwidth` and `maximum-bandwidth`, 3-bit binary (000 - 111).
`exceed-bandwidth <integer>`	Set the exceed bandwidth used for DSCP or VLAN CoS multi-stage marking. The integer value range depends on the `bandwidth-unit` setting. This setting is only available for CoS multi-stage marking.

Example

In this example, mutli-stage VLAN CoS marking is configured using traffic shapers on FortiGate A and FortiGate B. FortiGate A applies multi-stage CoS marking with the following traffic shaper settings:

- Traffic below the guaranteed bandwidth will apply CoS 6.
- Traffic greater than the guaranteed bandwidth will apply CoS 6 and 5.
- Traffic greater than the exceed bandwidth will apply CoS 6, 5, and 4.

A traffic shaper and shaping policy are configured on FortiGate B. When traffic comes from FortiGate A with CoS 6, the traffic shaping policy will be applied because the CoS matches.



Multi-stage VLAN CoS marking is not supported on NP models. Traffic is not offloaded when it is enabled.

To configure mutli-stage VLAN CoS marking on FortiGate A:

1. Configure the firewall policy:

```
config firewall policy
    edit 7
        set srcintf "port1"
        set dstintf "vlan100"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set traffic-shaper "multi-stage-cos-fgta"
        set traffic-shaper-reverse "multi-stage-cos-fgta"
    next
end
```

2. Configure the traffic shaper:

```
config firewall shaper traffic-shaper
    edit "multi-stage-cos-fgta"
        set guaranteed-bandwidth 1000
        set maximum-bandwidth 4000
        set per-policy enable
        set exceed-bandwidth 2000
        set cos-marking enable
        set cos-marking-method multi-stage
        set cos 110
        set exceed-cos 101
        set maximum-cos 100
    next
end
```

3. Check the session list to verify that CoS 6 is marked:

```
# diagnose sys session list
session info: proto=17 proto_state=00 duration=6 expire=180 timeout=0 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=multi-stage-cos-fgta prio=2 guarantee 125000Bps max 500000Bps traffic 504900Bps
drops 163905268B
```

```
reply-shaper=multi-stage-cos-fgta prio=2 guarantee 125000Bps max 500000Bps traffic 504900Bps
drops 0B
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=6/6
state=log may_dirty npu npd os rs f00
statistic(bytes/packets/allow_err): org=3804176/292/1 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 583462/4667 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=19->47/47->19 gwy=20.20.20.2/0.0.0.0
hook=pre dir=org act=noop 10.1.100.11:37586->192.168.4.33:5001(0.0.0.0:0)
hook=post dir=reply act=noop 192.168.4.33:5001->10.1.100.11:37586(0.0.0.0:0)
src_mac=00:0c:29:57:2a:01  dst_mac=70:4c:a5:7d:d4:95
misc=0 policy_id=7 pol_uuid_idx=1129 auth_info=0 chk_client_info=0 vd=2
serial=0006613c tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000000
npu info: flag=0x00/0x00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0, vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
no_ofld_reason:  offload-denied
```

To configure mutli-stage VLAN CoS marking on FortiGate B:

1. Configure the firewall policy:

```
config firewall policy
    edit 4
        set srcintf "vlan100"
        set dstintf "vlan200"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
    next
end
```

2. Configure the traffic shaper:

```
config firewall shaper traffic-shaper
    edit "multi-stage-cos-fgtb"
        set guaranteed-bandwidth 250
        set maximum-bandwidth 1000
        set per-policy enable
        set cos-marking enable
        set cos-marking-method multi-stage
        set cos 100
        set exceed-cos 101
        set maximum-cos 110
        set exceed-bandwidth 500
```

```
    next
end
```

Based on this traffic shaper, the following CoS marking rules will be applied:

- If all traffic is less than the guaranteed bandwidth, then the traffic will be marked with CoS 4.
- If all traffic is greater than the guaranteed bandwidth but less than the exceed bandwidth, then 50% of the traffic will be marked as CoS 4 and 50% as CoS 5.
- If traffic is greater than the guaranteed bandwidth but less than the maximum bandwidth, then 50% of the traffic will be marked as CoS 6; CoS 4 and 5 will have another 50%.
- If traffic is greater than the maximum bandwidth, then 50% of the traffic will be marked as CoS 6, 25% will be marked as CoS 4, and 25% will be marked as CoS 5. Packet drops will be visible in the debug output.

3. Configure the traffic shaping policy:

```
config firewall shaping-policy
    edit 1
        set service "ALL"
        set srcintf "vlan100"
        set dstintf "vlan200"
        set traffic-shaper "multi-stage-cos-fgtb"
        set traffic-shaper-reverse "multi-stage-cos-fgtb"
        set class-id 2
        set cos-mask 111
        set cos 110
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

4. Check the session list to verify that the shaping ID (1) applied and CoS 4 is marked:

```
# diagnose sys session list
session info: proto=1 proto_state=00 duration=1 expire=59 timeout=0 flags=00000000 socktype=0
sockport=0 av_idx=0 use=3
origin-shaper=multi-stage-cos-fgtb prio=2 guarantee 31250Bps max 125000Bps traffic 236Bps
drops 0B
reply-shaper=multi-stage-cos-fgtb prio=2 guarantee 31250Bps max 125000Bps traffic 236Bps drops
0B
per_ip_shaper=
class_id=2 shaping_policy_id=1 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=4/4
state=log may_dirty os rs f00
statistic(bytes/packets/allow_err): org=168/2/1 reply=168/2/1 tuples=2
tx speed(Bps/kbps): 120/0 rx speed(Bps/kbps): 120/0
orgin->sink: org pre->post, reply pre->post dev=59->61/61->59 gwy=20.20.200.3/20.20.20.1
hook=pre dir=org act=noop 10.1.100.11:29899->192.168.4.33:8(0.0.0.0:0)
hook=post dir=reply act=noop 192.168.4.33:29899->10.1.100.11:0(0.0.0.0:0)
src_mac=90:6c:ac:fb:bb:97  dst_mac=04:d5:90:36:73:3f
misc=0 policy_id=3 pol_uuid_idx=1377 auth_info=0 chk_client_info=0 vd=4
serial=00024329 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x040000
```

```
no_ofld_reason:  non-npu-intf
total session 1
```

Adding traffic shapers to multicast policies

When multicast routing is enabled, a traffic shaper can be added to a multicast policy.

Only a shared traffic shaper with the `per-policy` option disabled can be used. This is the default state of the `per-policy` option. The `auto-asic-offload` option must also be disabled on the multicast policy.

This feature is currently not supported on IPv6 multicast policies or on transparent mode VDOMs.

Example

In this example, a traffic shaper is applied to the multicast policy. A multicast flow sender sends the multicast data stream. The shaper attached to the multicast session is checked, and the shaping of the data stream is confirmed in the multicast session.

To apply traffic shaping to a multicast policy:

1. Enable multicast routing on the VDOM:

```
config router multicast
    set multicast-routing enable
    config pim-sm-global
        config rp-address
            edit 1
                set ip-address 10.1.100.10
            next
        end
    end
    config interface
        edit "wan2"
            set pim-mode sparse-mode
        next
        edit "wan1"
            set pim-mode sparse-mode
        next
    end
end
```

2. Create a traffic shaper:

```
config firewall shaper traffic-shaper
    edit "shaper128kbps-high"
```

```
        set guaranteed-bandwidth 128
        set maximum-bandwidth 128
        set per-policy disable
        set diffserv enable
        set diffservcode 010101
    next
end
```

3. Apply the traffic shaper to the multicast policy and disable NPU offloading:

```
config firewall multicast-policy
    edit 1
        set name "test_multicast-policy"
        set logtraffic enable
        set srcintf "wan2"
        set dstintf "wan1"
        set srcaddr "all"
        set dstaddr "all"
        set snat enable
        set auto-asic-offload disable
        set traffic-shaper "shaper128kbps-high"
    next
end
```

4. Check the shaper and DSCP in the multicast session:

```
# diagnose sys mcast-session list
    session info: id=26 vf=0 proto=17 10.1.100.41.35537->230.0.0.1.7878
    used=2 path=1 duration=118 expire=179 indev=18 pkts=119 bytes=64260
    state=00000000:
    session-npu-info: ipid/vlifid=0/0 vlanid/vtag_in=0/0 in_npuid=0 tae_index=0 qid=0 fwd_
map=0x00000000
    path: log snat npu-deny nsaddr=172.16.200.10 policy=1, outdev=17, tos=0x15
            origin-shaper=shaper128kbps-high prio=2 tos=0x15 guarantee 16000Bps max 16000Bps
traffic 620Bps drops 0pkt/0B
    Total 1 sessions
```

Global traffic prioritization

Global traffic prioritization allows your traffic to be prioritized as high (2), medium (3), or low (4) based on ToS (type of service) or DSCP. When using ToS-based priority, integers 0 to 15 can be used, which correspond to the definitions of the ToS field values in RFC 1349. When using DSCP, values 0 to 63 can be used, which correspond to the six bits in the DSCP value.

The outbandwidth must be defined in order for global prioritization to take effect. When the outbandwidth is defined on an interface without an applied `egress-shaping-profile`, the interface has a total of five priority levels:

Priority level	Description
0	Top
1	Critical
2	High
3	Medium
4	Low

Priority level 0 is reserved for administrative and local out traffic. Priority level 1 is used for traffic that is below guaranteed bandwidth when using a traffic shaper.

Traffic shaper and traffic shaping profile configurations take precedence over global traffic prioritization.

CLI commands

The following commands are used to configure the prioritization either by ToS or DSCP.

To configure the traffic prioritization type and level:

```
config system global
    set traffic-priority {tos | dscp}
    set traffic-priority-level {high | medium | low}
end
```

To configure the ToS-based priority table:

```
config system tos-based-priority
    edit <id>
        set tos <0-15>
        set priority (high | medium | low)
    next
end
```

To configure the DSCP-based priority table:

```
config system dscp-based-priority
    edit <id>
        set ds <0-63>
        set priority (high | medium | low)
    next
end
```

To configure the interface outbandwidth:

```
config system interface
    edit <name>
        set outbandwidth <bandwidth in kbps>
    next
end
```

Example

In the following configuration, packets with DSCP markings of 1 are prioritized as high, and packets with DSCP markings of 2 are prioritized as medium. All the other traffic is prioritized as low. The outbandwidth on interface port3 is set to 1000 kbps.

To configure DSCP-based traffic prioritization:

1. Configure DSCP-based prioritization in the global settings:

```
config system global
    set traffic-priority dscp
    set traffic-priority-level low
end
```

2. Configure the DSCP-based priority table:

```
config system dscp-based-priority
    edit 1
        set ds 1
        set priority high
    next
    edit 2
        set ds 2
        set priority medium
    next
end
```

3. Configure the outbandwidth on port3:

```
config system interface
    edit "port3"
        set outbandwidth 1000
    next
end
```

Verifying the traffic prioritization

When traffic exceeds the outbandwidth of 1000 kbps, traffic prioritization will take effect. Since the form of traffic shaping applied here is policing, excess packets above the outbandwidth are dropped.

In scenario 1, approximately 300 kbps of high priority traffic and 300 kbps of medium priority traffic passes through the FortiGate on port3.

To debug the bandwidth allocation:

```
# diagnose netlink interface list port3
if=port3 family=00 type=1 index=5 mtu=1500 link=0 master=0
ref=35 state=start present fw_flags=3800 flags=up broadcast run allmulti multicast
Qdisc=pfifo_fast hw_addr=52:54:00:fb:81:0c broadcast_addr=ff:ff:ff:ff:ff:ff
outbandwidth=1000(kbps)
        priority=0      allocated-bandwidth=0(kbps)     total_bytes=9311K       drop_bytes=197K
        priority=1      allocated-bandwidth=0(kbps)     total_bytes=0   drop_bytes=0
        priority=2      allocated-bandwidth=354(kbps)   total_bytes=20407K      drop_bytes=48K
        priority=3      allocated-bandwidth=354(kbps)   total_bytes=7093K       drop_bytes=1262K
        priority=4      allocated-bandwidth=290(kbps)   total_bytes=266018K     drop_bytes=7743K
stat: rxp=15450901 txp=25933756 rxb=5456860515 txb=17257309292 rxe=0 txe=0 rxd=0 txd=0 mc=0
collision=0 @ time=1629439926
re: rxl=0 rxo=0 rxc=0 rxf=0 rxfi=0 rxm=0
te: txa=0 txc=0 txfi=0 txh=0 txw=0
misc rxc=0 txc=0
input_type=0 state=3 arp_entry=0 refcnt=35
```

High priority (2) traffic is allocated 354 kbps of bandwidth. Medium priority (3) traffic is also allocated 354 kbps of bandwidth. The remaining bandwidth is allocated to low priority (4) traffic.

In scenario 2, approximately 400 kbps of high priority traffic and 800 kbps of medium priority traffic passes through the FortiGate on port3.

To debug the bandwidth allocation:

```
# diagnose netlink interface list port3
if=port3 family=00 type=1 index=5 mtu=1500 link=0 master=0
ref=36 state=start present fw_flags=3800 flags=up broadcast run allmulti multicast
Qdisc=pfifo_fast hw_addr=52:54:00:fb:81:0c broadcast_addr=ff:ff:ff:ff:ff:ff
outbandwidth=1000(kbps)
        priority=0      allocated-bandwidth=7(kbps)     total_bytes=9981K       drop_bytes=240K
        priority=1      allocated-bandwidth=0(kbps)     total_bytes=0   drop_bytes=0
        priority=2      allocated-bandwidth=425(kbps)   total_bytes=31478K      drop_bytes=101K
        priority=3      allocated-bandwidth=567(kbps)   total_bytes=12056K      drop_bytes=1984K
        priority=4      allocated-bandwidth=0(kbps)     total_bytes=266795K     drop_bytes=7771K
stat: rxp=15461740 txp=25950805 rxb=5459688950 txb=17273940560 rxe=0 txe=0 rxd=0 txd=0 mc=0
collision=0 @ time=1629440553
re: rxl=0 rxo=0 rxc=0 rxf=0 rxfi=0 rxm=0
te: txa=0 txc=0 txfi=0 txh=0 txw=0
misc rxc=0 txc=0
input_type=0 state=3 arp_entry=0 refcnt=36
```

High priority (2) traffic is allocated 425 kbps of bandwidth. Medium priority (3) traffic is allocated 567 kbps of bandwidth. Since the total bandwidth required exceeds 1000 kbps, the remaining medium priority (3) traffic is dropped. In comparing the successive debug outputs, the `drop_bytes` counter for medium priority (3) traffic gets bigger.

DSCP matching and DSCP marking

This section includes:

- DSCP matching in firewall policies
- DSCP matching in firewall shaping policies
- DSCP marking in firewall shaping policies
- DSCP marking for self-generated traffic on page 2156

DSCP matching in firewall policies

Traffic is allowed or blocked according to the Differentiated Services Code Point (DSCP) values in the incoming packets.

The following CLI variables are available in the `config firewall policy` command:

`tos-mask <mask_value>`	Non-zero bit positions are used for comparison. Zero bit positions are ignored (default = 0x00). This variable replaces the `dscp-match` variable.
`tos <tos_value>`	Type of Service (ToC) value that is used for comparison (default = 0x00). This variable is only available when `tos-mask` is not zero. This variable replaces the `dscp-value` variable.
`tos-negate {enable | disable}`	Enable/disable negated ToS match (default = disable). This variable is only available when `tos-mask` is not zero. This variable replaces the `dscp-negate` variable.

DSCP matching in firewall shaping policies

Shaping is applied to the session or not according to the DSCP values in the incoming packets. The same logic and commands as in firewall policies are used.

DSCP marking in firewall shaping policies

Traffic is allowed or blocked according to the DSCP values in the incoming packets. DSCP marking in firewall shaping policies uses the same logic and commands as in firewall policy and traffic-shaper.

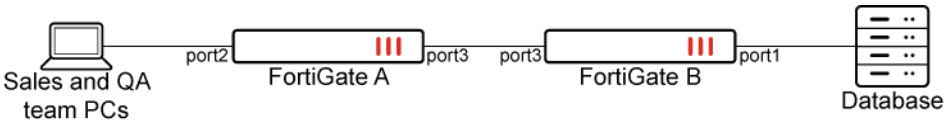
When DSCP marking on `firewall shaper traffic-shaper`, `firewall shaping-policy`, and `firewall policy` all apply to the same session, `shaping-policy` overrides `policy`, and `shaper traffic-shaper` overrides both `shaping-policy` and `policy`.

The following CLI variables in `config firewall policy` are used to mark the packets:

`diffserv-forward {enable | disable}`	Enable/disable changing a packet's DiffServ values to the value specified in `diffservcode-forward` (default = disable).

`diffservcode-forward <dscp_value>`	The value that packet's DiffServ is set to (default = 000000). This variable is only available when `diffserv-forward` is enabled.
`diffserv-reverse {enable | disable}`	Enable/disable changing a packet's reverse (reply) DiffServ values to the value specified in `diffservcode-rev` (default = disable).
`diffservcode-rev <dscp_value>`	The value that packet's reverse (reply) DiffServ is set to (default = 000000). This variable is only available when `diffserv-rev` is enabled.

The following topology is used in the examples:



Example 1

FortiGate A marks traffic from the sales and QA teams with different DSCP values. FortiGate B does DSCP matching, allowing only the sales team to access the database.

1. Configure FortiGate A:

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "port3"
        set srcaddr "QA"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set diffserv-forward enable
        set diffservcode-forward 110000
        set nat enable
    next
    edit 5
        set srcintf "port2"
        set dstintf "port3"
        set srcaddr "Sales"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set diffserv-forward enable
        set diffservcode-forward 111011
        set nat enable
    next
end
```

2. Configure FortiGate B:

```
config firewall policy
    edit 2
        set srcintf "port3"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "Database"
        set action accept
        set schedule "always"
        set service "ALL"
        set tos-mask 0xf0
        set tos 0xe0
        set fsso disable
        set nat enable
    next
end
```

Example 2

FortiGate A marks traffic from the sales and QA teams with different DSCP values. FortiGate B uses a firewall shaping policy to do the DSCP matching, limiting the connection speed of the sales team to the database to 10MB/s.

1. Configure FortiGate A:

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "port3"
        set srcaddr "QA"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set diffserv-forward enable
        set diffservcode-forward 110000
        set nat enable
    next
    edit 5
        set srcintf "port2"
        set dstintf "port3"
        set srcaddr "Sales"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set diffserv-forward enable
        set diffservcode-forward 111011
        set nat enable
    next
end
```

2. Configure FortiGate B:

```
config firewall policy
    edit 2
        set srcintf "port3"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
config firewall shaper traffic-shaper
    edit "10MB/s"
        set guaranteed-bandwidth 60000
        set maximum-bandwidth 80000
    next
end
config firewall shaping-policy
    edit 1
        set service "ALL"
        set dstintf "port1"
        set tos-mask 0xf0
        set tos 0xe0
        set traffic-shaper "10MB/s"
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

Example 3

FortiGate A has a traffic shaping policy to mark traffic from the QA team with a DSCP value of 100000, while reverse traffic is marked with 000011.

1. Configure FortiGate A:

```
config firewall shaping-policy
    edit 1
        set name "QA Team 50MB"
        set service "ALL"
        set dstintf "port3"
        set traffic-shaper "50MB/s"
        set traffic-shaper-reverse "50MB/s"
        set diffserv-forward enable
        set diffserv-reverse enable
        set srcaddr "QA"
        set dstaddr "all"
        set diffservcode-forward 100000
        set diffservcode-rev 000011
    next
end
```

DSCP marking for self-generated traffic

FortiOS supports DSCP and VLAN CoS marking for both local-in and local-out traffic.

Most network providers often require that both application traffic and FortiGate self-generated traffic must be marked with specific DSCP values to ensure efficient traffic management and quality of service (QoS). FortiOS DSCP marking ensures that self-generated traffic complies with the network's standards. This enables the FortiGate to operate as a fully functional Customer Premises Equipment (CPE) that is capable of directly connecting to the provider's network without a CPE router.

This feature does not apply to self-generated traffic like DHCP requests that do not go through the kernel IP stack and therefore do not generate a kernel session to apply the traffic shaping to.

To configure DSCP and VLAN CoS for local-in traffic:

1. Configure the traffic shaper with bandwidth settings and the DSCP and VLAN CoS mark:

```
config firewall shaper traffic-shaper
    edit "test-shaper-300kbps"
        set guaranteed-bandwidth 30
        set maximum-bandwidth 300
        set per-policy enable
        set diffserv enable
        set cos-marking enable
        set cos 001
        set diffservcode 000001
    next
end
```

2. Configure the shaping policy for local-in traffic:

```
config firewall shaping-policy
    edit 2
        set traffic-type local-in
        set service "ALL"
        set traffic-shaper-reverse "test-shaper-300kbps"
        set class-id 2
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

3. Verify that the shaper was successfully applied to the shaping policy:

```
# diagnose firewall iprope list 100018
policy index=2 uuid_idx=926 action=accept
flag (0):
schedule(always)
shapers: reply=test-shaper-300kbps(2/3750/37500)
cos_fwd=255  cos_rev=255
group=00100018 av=00000000 au=00000000 split=00000000
```

```
host=2 chk_client_info=0x0 app_list=0 ips_view=0
misc=0
zone(1): 0 -> zone(1): 0
source(1): 0.0.0.0-255.255.255.255, uuid_idx=799,
dest(1): 0.0.0.0-255.255.255.255, uuid_idx=799,
service(1):
        [0:0x0:0/(0,65535)->(0,65535)] flags:0 helper:auto
class_id: 2
```

4. Test local-in traffic from the PC to the FortiGate.
   a. Check the session list:

```
# diagnose sys session list
session info: proto=17 proto_state=01 duration=9 expire=179 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=test-shaper-300kbps prio=2 guarantee 3750Bps max 37500Bps traffic 7881Bps
drops 651B
per_ip_shaper=
class_id=2 shaping_policy_id=2 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/1
state=local may_dirty rs
statistic(bytes/packets/allow_err): org=337599/4717/1 reply=342414/4708/1 tuples=2
tx speed(Bps/kbps): 34948/279 rx speed(Bps/kbps): 35446/283
orgin->sink: org pre->in, reply out->post dev=7->48/48->7 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 172.16.200.55:58382->172.16.200.2:161(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.200.2:161->172.16.200.55:58382(0.0.0.0:0)
src_mac=00:0c:29:d6:12:20
misc=0 policy_id=4294967295 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=2
serial=0000249b tos=ff/01 app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=00000000
no_ofld_reason:  local
```

   b. Check the shaper information to verify the DSCP mark and bandwidth limitation:

```
# diagnose firewall shaper traffic-shaper list | grep test- -A 10
name test-shaper-300kbps
maximum-bandwidth 37 KB/sec
guaranteed-bandwidth 3 KB/sec
current-bandwidth 37 KB/sec
priority 2
policy 2
overhead 0
tos 01
packets dropped 10
bytes dropped 725
```

To configure DSCP and VLAN CoS for local-out traffic:

1. Configure the traffic shaper with bandwidth settings and the DSCP and VLAN CoS mark:

```
config firewall shaper traffic-shaper
    edit "test-shaper-600kbps"
        set guaranteed-bandwidth 60
        set maximum-bandwidth 600
        set per-policy enable
        set diffserv enable
        set cos-marking enable
        set cos 110
        set diffservcode 110000
    next
end
```

2. Configure the shaping policy for local-out traffic:

```
config firewall shaping-policy
    edit 5
        set traffic-type local-out
        set service "ALL"
        set traffic-shaper "test-shaper-600kbps"
        set class-id 5
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

3. Verify that the shaper was successfully applied to the shaping policy:

```
# diagnose firewall iprope list 100019
policy index=5 uuid_idx=928 action=accept
flag (0):
schedule()
shapers: orig=test-shaper-600kbps(2/7500/75000)
cos_fwd=255  cos_rev=255
group=00100019 av=00000000 au=00000000 split=00000000
host=2 chk_client_info=0x0 app_list=0 ips_view=0
misc=0
zone(1): 0 -> zone(1): 0
source(1): 0.0.0.0-255.255.255.255, uuid_idx=799,
dest(1): 0.0.0.0-255.255.255.255, uuid_idx=799,
service(1):
        [0:0x0:0/(0,65535)->(0,65535)] flags:0 helper:auto
class_id: 5
```

4. Test local-in traffic from the FortiGate to the remote PC.
   a. Check the session list:

```
# diagnose sys session list
session info: proto=6 proto_state=01 duration=4 expire=3599 timeout=3600 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
```

```
origin-shaper=test-shaper-600kbps prio=2 guarantee 7500Bps max 75000Bps traffic 73557Bps
drops 70500B
reply-shaper=
per_ip_shaper=
class_id=5 shaping_policy_id=5 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=6/255
state=log local os
statistic(bytes/packets/allow_err): org=85701/60/1 reply=2140/41/1 tuples=2
tx speed(Bps/kbps): 19172/153 rx speed(Bps/kbps): 478/3
orgin->sink: org out->post, reply pre->in dev=48->7/7->48 gwy=0.0.0.0/0.0.0.0
hook=out dir=org act=noop 172.16.200.2:23964->209.52.38.114:5201(0.0.0.0:0)
hook=in dir=reply act=noop 209.52.38.114:5201->172.16.200.2:23964(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
dst_mac=04:d5:90:5d:ed:fe
misc=0 policy_id=0 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=2
serial=000152f5 tos=30/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=00000000
no_ofld_reason:  local
```

b. Check the shaper information to verify the DSCP mark and bandwidth limitation:

```
#  diagnose firewall shaper traffic-shaper list | grep test- -A 10
name test-shaper-600kbps
maximum-bandwidth 75 KB/sec
guaranteed-bandwidth 7 KB/sec
current-bandwidth 65 KB/sec
priority 2
policy 5
overhead 0
tos 30
packets dropped 5086
bytes dropped 1148949
```

Examples

This section includes the following traffic shaping configuration examples:

- Interface-based traffic shaping profile on page 2159
- Interface-based traffic shaping with NP acceleration on page 2169
- QoS assignment and rate limiting for FortiSwitch quarantined VLANs on page 2171
- Ingress traffic shaping profile on page 2172

Interface-based traffic shaping profile

A traffic shaping policy can be used for interface-based traffic shaping by organizing traffic into 30 class IDs. The shaping profile defines the percentage of the interface bandwidth that is allocated to each class. Each traffic class ID is shaped to the assigned speed according to the outgoing bandwidth limit configured to the interface.

Traffic classification

A shaping policy classifies traffic and organizes it into different class IDs, based on matching criteria. For traffic matching a criteria, you can choose to put it into 30 different shaping classes, identified by class ID 2 - 31.

- When NPU offloading is enabled on FortiGates with NP6, NP6Lite (SoC3), or NP6XLite (SoC4) processors, the class ID limit for egress traffic is 2 - 15.
- When NPU offloading is enabled on FortiGates with NP7 or NP7Lite (SoC5) processors, the class ID limit for egress traffic is 2–11.
- When NPU offloading is disabled, the class ID limit for egress traffic is 2–31.

Setting the egress traffic class ID outside of these limits can result in unexpected behavior.

You must select an outgoing interface for the traffic. The shaping policy is only applied when the traffic goes to one of the selected outgoing interfaces.

Criterion	Description
Source	• Address: match the source address of the traffic to the selected address or address group. • User: use the user credentials of the traffic to match the selected user or user group. At least one address, address group, or internet service must also be selected. • Internet service: match the traffic to the selected internet service. Internet services cannot be used if addresses or address or groups are used.
Destination	• Address: match the destination address of the traffic to the selected address or address group. • Internet service: match the traffic to the selected internet service. Internet services cannot be used if addresses or address or groups are used.
Schedule	Match the current date and time to the selected schedule. You can select a one-time schedule, recurring schedule, or schedule group. This setting is optional.
Service	Match the service of the traffic to the selected service or service group.
Application	Match the application of the traffic to the selected application, application category, or application group. Application control must be enabled in the related firewall policy to know the application of the traffic. See Application control on page 2428 for more information.
URL category	Match the URL of the traffic to the selected URL category. Web filter must be enabled in the related firewall policy to know the URL of the traffic. See Web filter on page 2311 for more information.

When multiple items are selected in one criterion, it is considered a match when traffic matches any one of them.

Traffic prioritization

Shaping profiles define how different shaping classes of traffic are prioritized. For each class, you can define three prioritization strategies: guaranteed bandwidth, maximum bandwidth, and priority.

For each shaping profile, a default shaping class must be defined. Traffic is prioritized based on the default shaping group in the following two circumstances:

- All traffic to the outgoing interface that does not match to any shaping policy
- Traffic with a shaping group that is not defined in a shaping profile

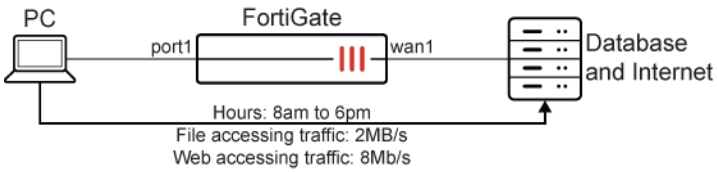
Prioritization strategy	Description
Guaranteed bandwidth	The percentage of the link speed that is reserved for the shaping group. The total guaranteed bandwidth for all shaping groups cannot exceed 100%.
Maximum bandwidth	The maximum percentage of the link speed that the shaping group can use.
Priority	The shaping class priority: top, critical, high, medium, or low. When groups are competing for bandwidth on the interface, the group with the higher priority wins.

Applying a shaping profile to an interface

Traffic shaping is accomplished by configuring the outgoing bandwidth and outgoing shaping profile on an interface. The shaping profile uses the outgoing bandwidth of the interface as the maximum link speed, and it only works when the outgoing bandwidth is configured.

This example shows how to apply interface-based traffic shaping to web and file accessing traffic according to a schedule:

- The link speed of the wan1 interface is 10 Mb/s.
- File access can use up to 2 Mb/s from 8:00 AM to 6:00 PM.
- Web access can use 8 Mb/s from 8:00 AM to 6:00 PM.



Putting the traffic into shaping classes

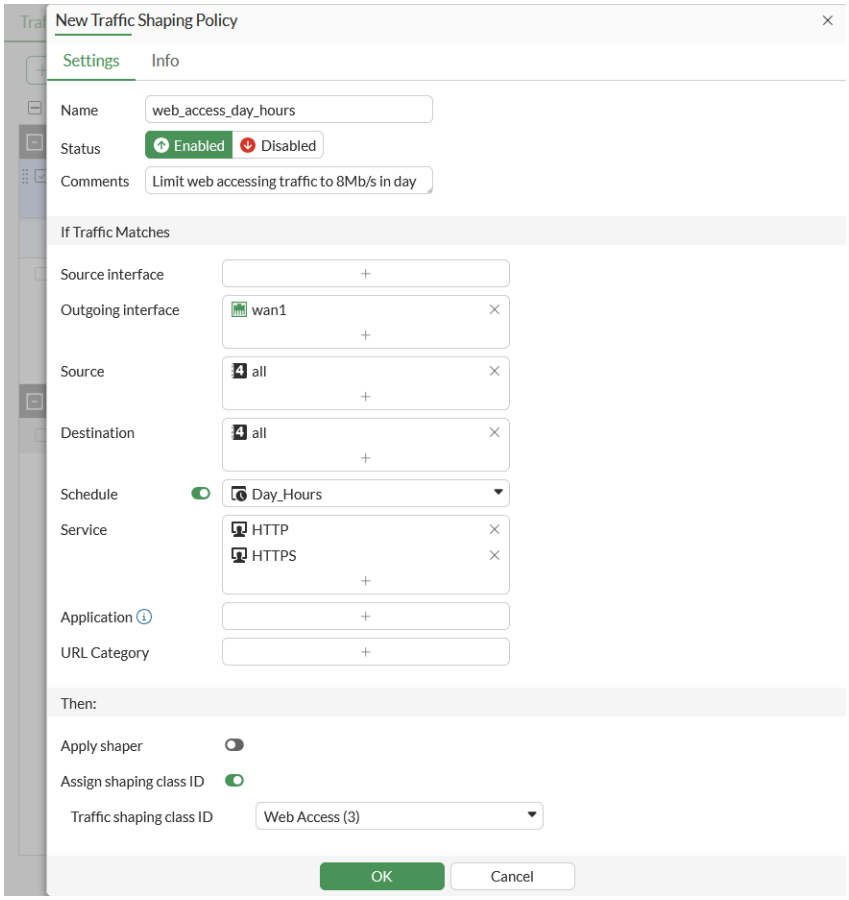
To create a recurring schedule in the GUI:

1. Go to *Policy & Objects* > *Schedules* and navigate to the *Recurring Schedule* tab.
2. Click *Create New*.

3. Configure a recurring schedule called *Day_Hours* for everyday from 8:00 AM to 6:00 PM.
4. Click *OK*.

To create a traffic shaping policy and class ID for the web accessing traffic in the GUI:

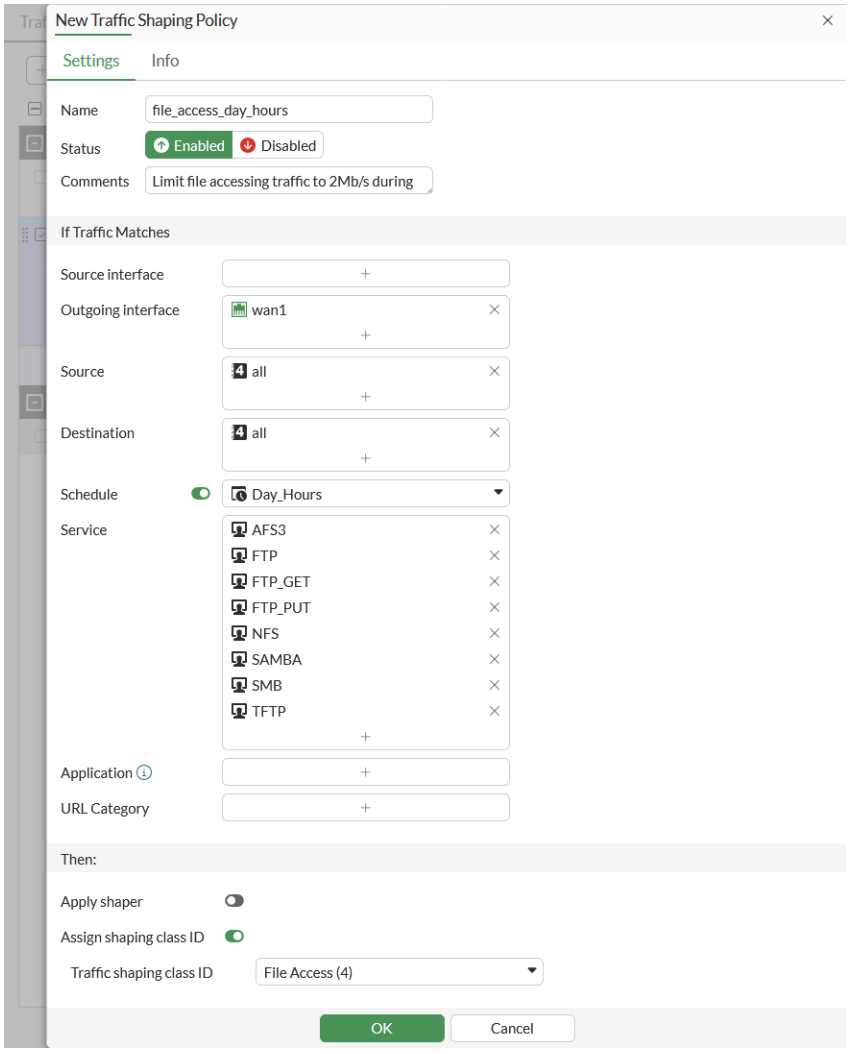
1. Go to *Policy & Objects* > *Traffic Shaping*, select the *Traffic Shaping Policy* tab, and click *Create New*.
2. Enter a name for the policy, such as *web_access_day_hours*.
3. Set *Outgoing interface* to *wan1*.
4. Enable *Schedule* and select the schedule you just created.
5. Set *Service* to web accessing services, such as *HTTP* and *HTTPS*.
6. Enable *Assign shaping class ID*.
7. Click the *Traffic shaping class ID* drop down then click *Create*.
8. Enter an integer value for the *ID* (*3*) and a description for the *Name*, such as *Web Access*.
9. Click *OK*.
10. Select the class ID you just created for *Traffic shaping class ID*.



11. Configure the remaining settings as required.
12. Click *OK*.

To create a traffic shaping policy and class ID for the file accessing traffic in the GUI:

1. Go to *Policy & Objects* > *Traffic Shaping*, select the *Traffic Shaping Policy* tab, and click *Create New*.
2. Enter a name for the policy, such as *file_access_day_hours*.
3. Set *Outgoing interface* to *wan1*.
4. Enable *Schedule* and select the schedule you just created.
5. Set *Service* to file accessing services, such as *ASF3*, *FTP* and *SMB*.
6. Enable *Assign shaping class ID*.
7. Click the *Traffic shaping class ID* drop down then click *Create*.
8. Enter an integer value for the *ID* (*4*) and a description for the *Name*, such as *File Access*.
9. Click *OK*.
10. Select the class ID you just created for *Traffic shaping class ID*.



11. Configure the remaining settings as required.
12. Click *OK*.

To put the traffic into shaping classes in the CLI:

1. Create a recurring schedule:

```
config firewall schedule recurring
    edit "Day_Hours"
        set start 08:00
        set end 18:00
        set day sunday monday tuesday wednesday thursday friday saturday
    next
end
```

2. Create the traffic class IDs:

```
config firewall traffic-class
    edit 3
        set class-name "Web Access"
    next
    edit 4
        set class-name "File Access"
    next
end
```

3. Create the web and file accessing traffic shaping policies:

```
config firewall shaping-policy
    edit 2
        set name "web_access_day_hours"
        set comment "Limit web accessing traffic to 8Mb/s in day time"
        set service "HTTP" "HTTPS"
        set schedule "Day_Hours"
        set dstintf "wan1"
        set class-id 3
        set srcaddr "all"
        set dstaddr "all"
    next
    edit 3
        set name "file_access_day_hours"
        set comment "Limit file accessing traffic to 2Mb/s during the day"
        set service "AFS3" "FTP" "FTP_GET" "FTP_PUT" "NFS" "SAMBA" "SMB" "TFTP"
        set schedule "Day_Hours"
        set dstintf "wan1"
        set class-id 4
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

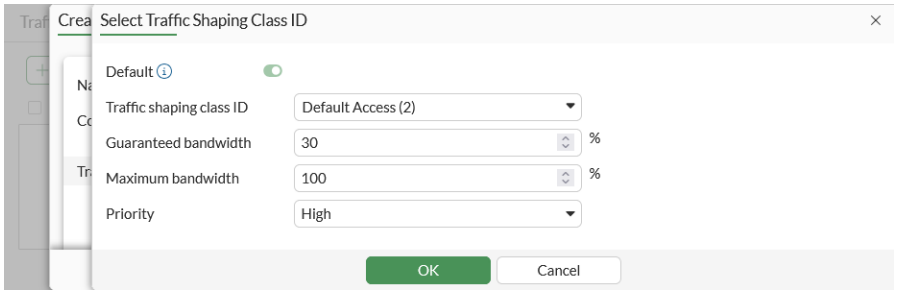
Allocating bandwidth to the shaping classes

A traffic shaping profile defines the guaranteed and maximum bandwidths each class receives. In this example, file access can use up to 2 Mb/s and web access can use 8 Mb/s from 8:00 AM to 6:00 PM.

To create a traffic shaping profile using the GUI:

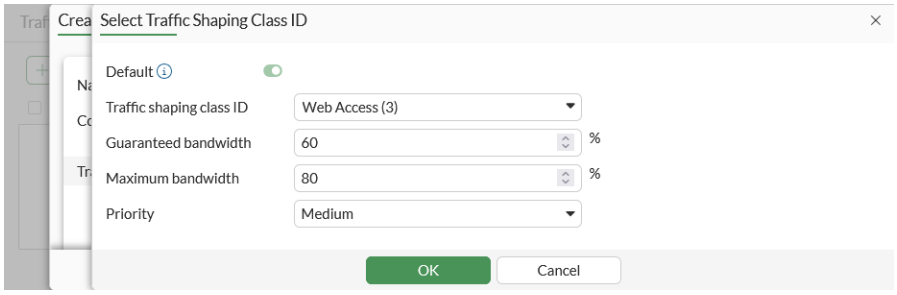
1. Go to *Policy & Objects* > *Traffic Shaping*, select the *Traffic Shaping Profile* tab, and click *Create New*.
2. Enter a name for the profile, such as *Day_Hours_Profile*.
3. Configure a default traffic shaping class:
   This class has a high priority, meaning that when the other classes have reached their guaranteed bandwidths, this default class will use the rest of the available bandwidth.
   a. In the *Traffic Shaping Classes* table click *Create New*.
   b. Click the *Traffic shaping class ID* drop down then click *Create*.
   c. Enter a name for the class, such as *Default Access*.
   d. Click *OK*.
   e. Select the class ID you just created for *Traffic shaping class ID*.
   f. Configure the following settings, then click *OK*:

Guaranteed bandwidth	30
Maximum bandwidth	100
Priority	High



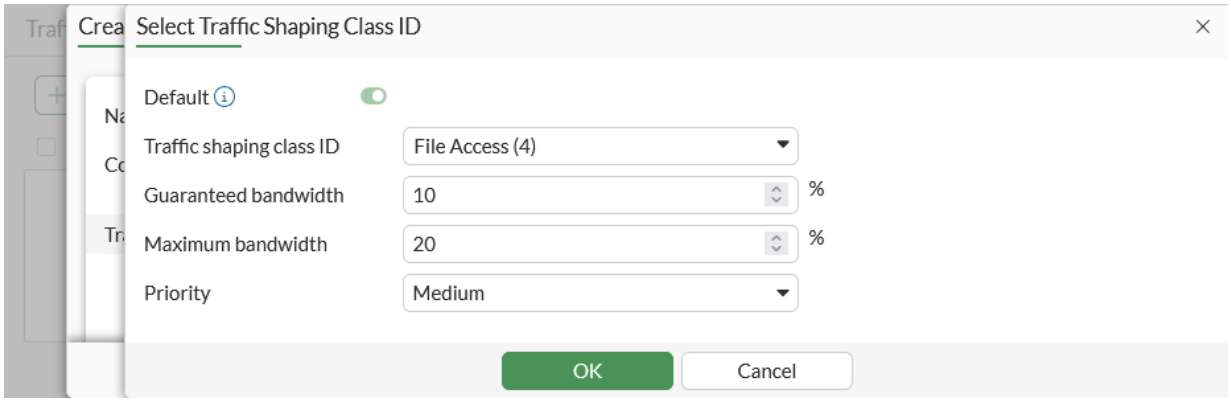
4. Configure a web accessing traffic shaping class:
   When other types of traffic are competing for bandwidth, this class is guaranteed to 6 Mb/s, or 60% of the bandwidth.
   a. In the *Traffic Shaping Classes* table click *Create New*.
   b. Configure the following settings, then click *OK*:

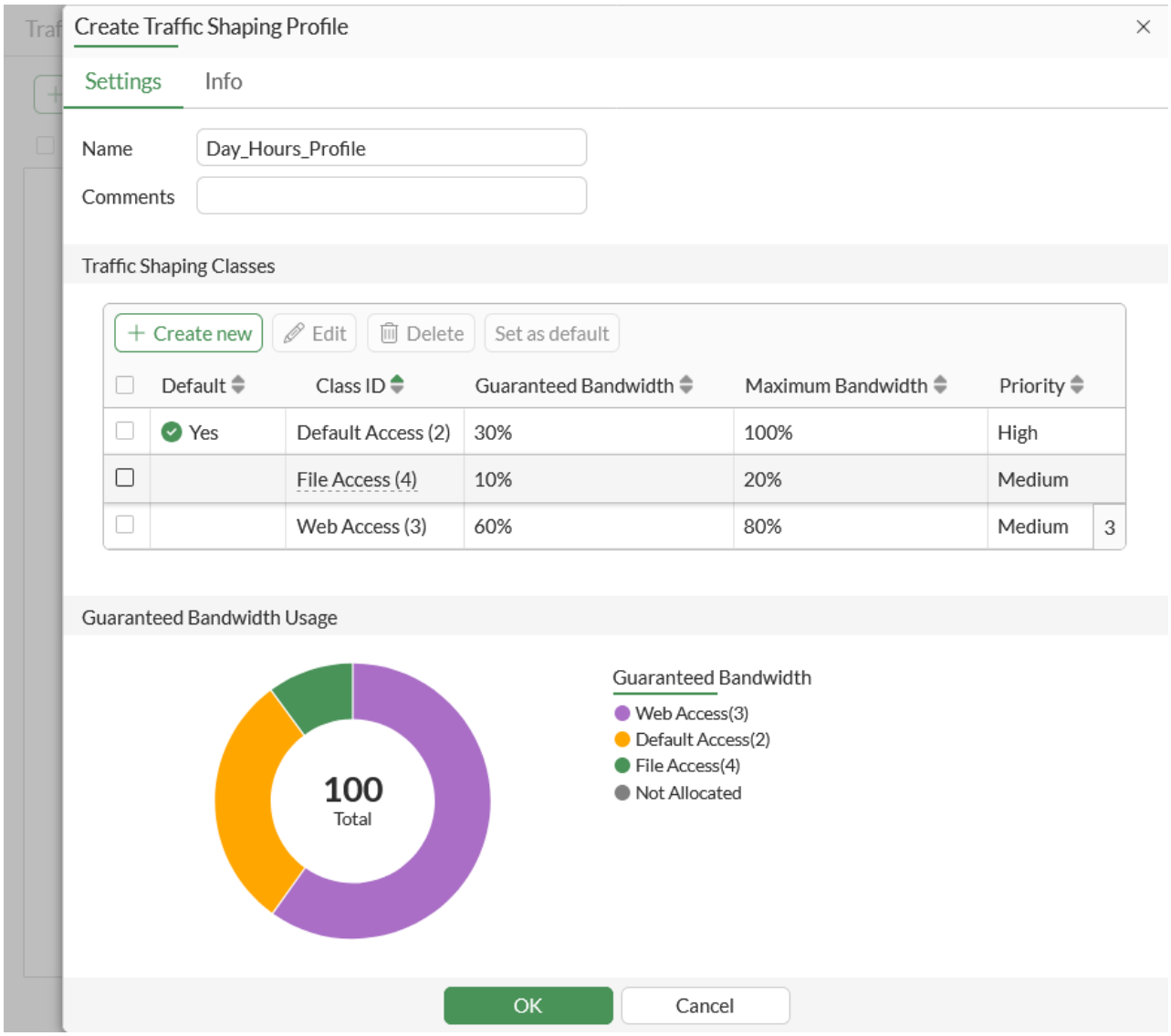
Traffic shaping class ID	Web Access
Guaranteed bandwidth	60
Maximum bandwidth	80
Priority	Medium



5. Configure a file accessing traffic shaping class:
   When other types of traffic are competing for bandwidth, this group is guaranteed to 1 Mb/s, or 10% of the bandwidth.
   a. In the *Traffic Shaping Classes* table click *Create New*.
   b. Configure the following settings, then click *OK*:

Traffic shaping class ID	File Access
Guaranteed bandwidth	10
Maximum bandwidth	20
Priority	Medium





6. Click *OK*.

To create a traffic shaping profile using the CLI:

```
config firewall shaping-profile
    edit "Day_Hours_Profile"
        set default-class-id 2
        config shaping-entries
            edit 1
                set class-id 2
                set guaranteed-bandwidth-percentage 30
                set maximum-bandwidth-percentage 100
            next
            edit 2
                set class-id 3
                set priority medium
```

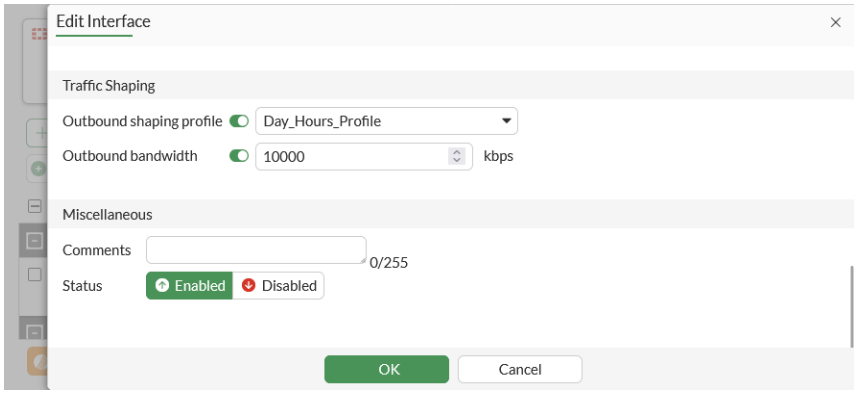
```
                set guaranteed-bandwidth-percentage 60
                set maximum-bandwidth-percentage 80
            next
            edit 3
                set class-id 4
                set priority medium
                set guaranteed-bandwidth-percentage 10
                set maximum-bandwidth-percentage 20
            next
        end
    next
end
```

Defining the available bandwidth on an interface

In this example, the link speed of the wan1 interface is 10 Mb/s.

To set the bandwidth of the wan1 interface in the GUI:

1. Go to *Network* > *Interfaces*.
2. Edit the wan1 interface.
3. Under Traffic Shaping, enable *Outbound shaping profile* and select the profile that you just created, *Day_Hours_Profile*.
4. Enable *Outbound Bandwidth* and set it to *10000* Kbps.



5. Click *OK*.

To set the bandwidth of the wan1 interface in the CLI:

```
config system interface
    edit "wan1"
        set egress-shaping-profile "Day_Hours_Profile"
        set outbandwidth 10000
    next
end
```

Diagnose commands

To check that the specific traffic is put into the correct shaping group or class ID:

```
# diagnose firewall iprope list 100015
```

To check the speed limit for each class ID on an interface:

```
# diagnose netlink interface list wan1
```

Interface-based traffic shaping with NP acceleration

Interface-based traffic shaping with NP acceleration is supported on some devices.

An administrator configures the WAN interface's maximum outbound bandwidth and, based on that, creates a traffic shaping profile with a percentage based shaper. This allows for proper QoS and traffic shaping. VLAN interfaces are not supported.

This feature is supported on FortiGate 600E, 500E, 400E, and 300E models.

To configure interface-based traffic shaping:

1. Enable NPU offloading when doing interface-based traffic shaping according to the egress-shaping-profile:

```
config system npu
    set intf-shaping-offload enable
end
```

2. Configure shaping profiles:

```
config firewall shaping-profile
    edit "sdwan"
        set default-class-id 4
        config shaping-entries
            edit 1
                set class-id 4
                set guaranteed-bandwidth-percentage 3
                set maximum-bandwidth-percentage 5
            next
            edit 2
                set class-id 3
                set priority medium
                set guaranteed-bandwidth-percentage 50
                set maximum-bandwidth-percentage 100
            next
            edit 3
                set class-id 2
                set priority low
```

```
                set guaranteed-bandwidth-percentage 1
                set maximum-bandwidth-percentage 5
            next
        end
    next
end
```

The class number is limited to 16.

3. Configure a traffic shaper and shaping policy:

```
config firewall shaper traffic-shaper
    edit "Transactional"
        set priority medium
    next
end
```

```
config firewall shaping-policy
    edit 1
        set service "ALL"
        set dstintf "any"
        set traffic-shaper "Transactional"
        set class-id 3
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

4. Apply the egress shaping profile on the interface:

```
config system interface
    edit "port2"
        set vdom "root"
        set ip 10.1.100.23 255.255.255.0
        set allowaccess ping
        set type physical
        set outbandwidth 500
        set egress-shaping-profile "sdwan"
        set snmp-index 4
    next
end
```

5. Configure a firewall policy:

```
config firewall policy
    edit 3
        set srcintf "port2"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
```

```
        set logtraffic all
        set nat enable
    next
end
```

QoS assignment and rate limiting for FortiSwitch quarantined VLANs

When devices are quarantined, they are isolated from the rest of the network. However, they can still impact the network if not controlled beyond isolation. A quarantined host, which offers heavy traffic, could congest the network and create a DOS-style reduction in service to authorized hosts.

Within the quarantined VLAN, two restrictions are available within the network:

- Traffic policing (also known as rate limiting)
- QoS (Quality of Service) assignment (also known as priority assignment)

Each quarantined host's traffic can be subject to rate limiting and priority adjustment. This reduces the impact that any quarantined host can have on authorized traffic on the network.

To configure QoS assignment and rate limiting for quarantined VLANs:

1. Configure a traffic policy, or use the default "quarantine" policy:

```
config switch-controller traffic-policy
    edit "quarantine"
        set description "Rate control for quarantined traffic"
        set guaranteed-bandwidth 163840
        set guaranteed-burst 8192
        set maximum-burst 163840
        set cos-queue 0
    next
end
```

2. Configure an interface:

```
config system interface
    edit "qtn.aggr1"
        set vdom "root"
        set ip 10.254.254.254 255.255.255.0
        set description "Quarantine VLAN"
        set security-mode captive-portal
        set replacemsg-override-group "auth-intf-qtn.aggr1"
        set device-identification enable
        set snmp-index 30
        set switch-controller-access-vlan enable
        set switch-controller-traffic-policy "quarantine"
        set color 6
        set interface "aggr1"
        set vlanid 4093
```

```
    next
end
```

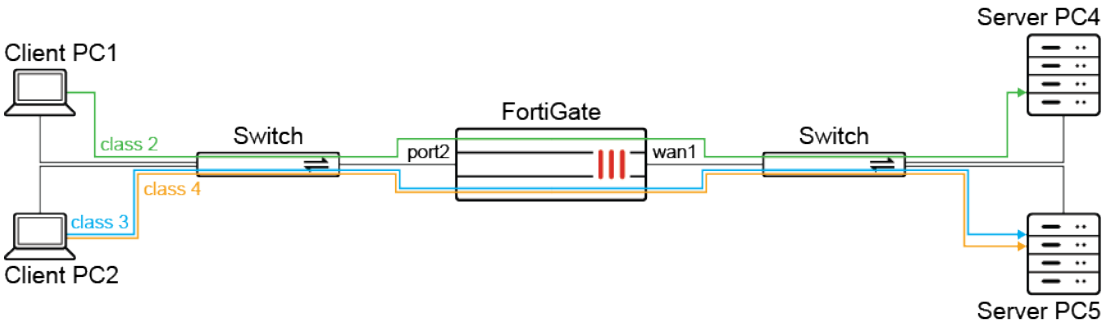
By default, `switch-controller-traffic-policy` is empty. You need to apply the necessary traffic policy (not only limited to "quarantine").

Ingress traffic shaping profile

A traffic shaping profile can be applied to an interface for traffic in the ingress direction. Similar to an egress traffic shaping profile, the guaranteed bandwidth and priority of the profile will be respected when an interface receives inbound traffic. When congestion occurs, any remaining bandwidth will be allotted to classes based on priority.

Example

In this example, the port2 interface has a total inbound bandwidth of 100 Mbps. Traffic from certain clients to certain servers are assigned different classes.



IPv6 traffic from any client PCs to server PCs is assigned class 5.

For each class, the priority, guaranteed bandwidth, and maximum bandwidth are as follows:

Class	Priority	Guaranteed bandwidth	Maximum bandwidth
2	Low	10%	60%
3	High	20%	100%
4	High	30%	100%
5	Medium	10%	50%

Bandwidth will first be allotted to each class according to its guaranteed bandwidth. Then remaining available bandwidth will be allotted to class 3 and 4 first based on their priority. The allocation will be proportional to their guaranteed bandwidth ratio.

To configure ingress traffic shaping:

1. Configure the client and server addresses:

```
config firewall address
    edit "pc1"
        set subnet 10.1.100.11 255.255.255.255
    next
    edit "pc2"
        set subnet 10.1.100.22 255.255.255.255
    next
    edit "pc4"
        set subnet 172.16.200.44 255.255.255.255
    next
    edit "pc5"
        set subnet 172.16.200.55 255.255.255.255
    next
end
```

2. Configure the class IDs:

```
config firewall traffic-class
    edit 2
        set class-name "class2"
    next
    edit 3
        set class-name "class3"
    next
    edit 4
        set class-name "class4"
    next
    edit 4
        set class-name "class5"
    next
end
```

3. Configure traffic shaping policies to assign classes to each group of traffic.
   a. Configure a policy to assign traffic from PC1 to PC4 in class 2:

```
config firewall shaping-policy
    edit 1
        set name "shaping policy 1"
        set service "ALL"
        set dstintf "wan1"
        set class-id 2
        set srcaddr "pc1"
        set dstaddr "pc4"
    next
end
```

   b. Configure a policy to assign traffic from PC2 to PC4 in class 3:

```
config firewall shaping-policy
    edit 2
        set name "shaping policy 2"
        set service "ALL"
        set dstintf "wan1"
        set class-id 3
        set srcaddr "pc2"
        set dstaddr "pc4"
    next
end
```

c. Configure a policy to assign traffic from PC2 to PC5 in class 4:

```
config firewall shaping-policy
    edit 3
        set name "shaping policy 3"
        set service "ALL"
        set dstintf "wan1"
        set class-id 4
        set srcaddr "pc2"
        set dstaddr "pc5"
    next
end
```

d. Configure a policy to assign all IPv6 traffic to class 5:

```
config firewall shaping-policy
    edit 4
        set name "shaping policy 4"
        set ip-version 6
        set service "ALL"
        set dstintf "wan1"
        set class-id 5
        set srcaddr6 "all"
        set dstaddr6 "all"
    next
end
```

4. Configure a shaping profile to set the priority, and the guaranteed and maximum bandwidth percentages for each class:

```
config firewall shaping-profile
    edit "ingShapeProfile"
        set default-class-id 2
        config shaping-entries
            edit 2
                set class-id 2
                set priority low
                set guaranteed-bandwidth-percentage 10
                set maximum-bandwidth-percentage 60
            next
            edit 3
```

```
                set class-id 3
                set guaranteed-bandwidth-percentage 20
                set maximum-bandwidth-percentage 100
            next
            edit 4
                set class-id 4
                set guaranteed-bandwidth-percentage 30
                set maximum-bandwidth-percentage 100
            next
            edit 5
                set class-id 5
                set priority medium
                set guaranteed-bandwidth-percentage 10
                set maximum-bandwidth-percentage 50
            next
        end
    next
end
```

5. Configure the inbandwidth and apply the ingress shaping profile on port2:

```
config system interface
    edit "port2"
        set ip 10.1.100.1 255.255.255.0
        set inbandwidth 100000
        set ingress-shaping-profile "ingShapeProfile"
        config ipv6
            set ip6-address 2000:10:1:100::1/64
        end
    next
end
```

Inbandwidth must be configured for traffic shaping to take effect.

6. Configure a firewall policy to allow traffic to go through. Since traffic shaping is for inbound traffic on port2, the policy is defined from port2 to wan1:

```
config firewall policy
    edit 2
        set srcintf "port2"
        set dstintf "wan1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set auto-asic-offload enable
        set nat enable
    next
end
```

Ingress traffic shaping supports NPU offloading and it is enabled by default. Set `auto-asic-offload` to `disable` to disable it.

Verifying that the traffic is being shaped

In each of the following cases, the server PCs (PC4 and PC5) are configured as iPerf servers. The client PCs (PC1 and PC2) are configured as iPerf clients. The client sends traffic to the server from the client to server direction, triggering inbound traffic shaping on the port2 interface. The inbound bandwidth on port2 is 100 Mbps.

Case 1: single stream, PC1 to PC4

Traffic is sent from PC1 to PC4. There is no other traffic. Traffic is marked with class ID 2 and allocated the maximum bandwidth 60 Mbps (60%).

```
# diagnose netlink interface list port2
if=port2 family=00 type=1 index=20 mtu=1500 link=0 master=0
ref=25 state=start present fw_flags=3800 flags=up broadcast run multicast
Qdisc=mq hw_addr=70:4c:a5:7d:d4:95 broadcast_addr=ff:ff:ff:ff:ff:ff
ingress traffic control:
        bandwidth=100000(kbps) lock_hit=50 default_class=2 n_active_class=4
        class-id=2      allocated-bandwidth=60000(kbps)         guaranteed-bandwidth=10000(kbps)
                        max-bandwidth=60000(kbps)       current-bandwidth=60002(kbps)
                        priority=low    forwarded_bytes=58157K
                        dropped_packets=94K     dropped_bytes=125385K
        class-id=5      allocated-bandwidth=1000(kbps)  guaranteed-bandwidth=10000(kbps)
                        max-bandwidth=50000(kbps)       current-bandwidth=0(kbps)
                        priority=medium         forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
        class-id=3      allocated-bandwidth=15000(kbps)         guaranteed-bandwidth=20000(kbps)
                        max-bandwidth=100000(kbps)      current-bandwidth=0(kbps)
                        priority=high   forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
        class-id=4      allocated-bandwidth=24000(kbps)         guaranteed-bandwidth=30000(kbps)
                        max-bandwidth=100000(kbps)      current-bandwidth=0(kbps)
                        priority=high   forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
stat: rxp=173465879 txp=2430534 rxb=194665548609 txb=2767375732 rxe=0 txe=0 rxd=0 txd=0 mc=0
collision=0 @ time=1628814469
re: rxl=0 rxo=0 rxc=0 rxf=0 rxfi=0 rxm=0
te: txa=0 txc=0 txfi=0 txh=0 txw=0
misc rxc=0 txc=0
input_type=0 state=3 arp_entry=0 refcnt=25
```

The `diagnose netlink interface list <interface>` command might not show the status of the ingress multi-class shaper on NP7 models due to hardware limitations.

Case 2: dual stream, PC1 to PC4, PC2 to PC4

Traffic is sent from both PC1 and PC2 to PC4. PC1 to PC4 traffic is marked with class ID 2 and low priority, and PC2 to PC4 traffic is marked with class ID 3 and high priority. Both class 2 and 3 will be allocated their guaranteed bandwidth first, using up 10% and 20% respectively. The remaining available bandwidth is used by class 3 since it has a higher priority. Class 2 uses around 10 Mbps, and class 3 uses around 90 Mbps.

```
# diagnose netlink interface list port2
if=port2 family=00 type=1 index=20 mtu=1500 link=0 master=0
ref=36 state=start present fw_flags=3800 flags=up broadcast run multicast
Qdisc=mq hw_addr=70:4c:a5:7d:d4:95 broadcast_addr=ff:ff:ff:ff:ff:ff
ingress traffic control:
        bandwidth=100000(kbps) lock_hit=181 default_class=2 n_active_class=4
        class-id=2      allocated-bandwidth=10000(kbps)         guaranteed-bandwidth=10000(kbps)
                        max-bandwidth=60000(kbps)       current-bandwidth=10001(kbps)
                        priority=low    forwarded_bytes=1799482K
                        dropped_packets=5998K   dropped_bytes=7965553K
        class-id=5      allocated-bandwidth=1000(kbps)  guaranteed-bandwidth=10000(kbps)
                        max-bandwidth=50000(kbps)       current-bandwidth=0(kbps)
                        priority=medium         forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
        class-id=3      allocated-bandwidth=88000(kbps)         guaranteed-bandwidth=20000(kbps)
                        max-bandwidth=100000(kbps)      current-bandwidth=88000(kbps)
                        priority=high   forwarded_bytes=345039K
                        dropped_packets=324K    dropped_bytes=430862K
        class-id=4      allocated-bandwidth=1000(kbps)  guaranteed-bandwidth=30000(kbps)
                        max-bandwidth=100000(kbps)      current-bandwidth=0(kbps)
                        priority=high   forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
stat: rxp=181269891 txp=2433428 rxb=205136511596 txb=2771214402 rxe=0 txe=0 rxd=0 txd=0 mc=0
collision=0 @ time=1628815849
re: rxl=0 rxo=0 rxc=0 rxf=0 rxfi=0 rxm=0
te: txa=0 txc=0 txfi=0 txh=0 txw=0
misc rxc=0 txc=0
input_type=0 state=3 arp_entry=0 refcnt=36
```

Case 3: multiple streams

Multiple streams of traffic are sent at the same time:

- PC1 to PC4 traffic is assigned class 2 with low priority, and a guaranteed bandwidth of 10 Mbps.
- PC2 to PC4 traffic is assigned class 3 with high priority, and a guaranteed bandwidth of 20 Mbps.
- PC2 to PC5 traffic is assigned class 4 with high priority, and a guaranteed bandwidth of 30 Mbps.

All classes will be allocated their guaranteed bandwidth first, using up 10 Mbps, 20 Mbps, and 30 Mbps respectively. The remaining available bandwidth (40 Mbps) is shared by class 3 and class 4 based on their guaranteed bandwidth ratio of 20:30.

- Class 3's share of the remaining 40 Mbps traffic = 40 × 20/(20 + 30) =16 Mpbs
- Class 4's share of the remaining 40 Mbps traffic = 40 × 30/(20 + 30) =24 Mpbs

Each class is allocated roughly the following bandwidth:

- Class 2: 10 Mbps
- Class 3: 20 Mbps + 16 Mbps = 36 Mbps
- Class 4: 30 Mbps + 24 Mbps = 54 Mbps

```
# diagnose netlink interface list port2
if=port2 family=00 type=1 index=20 mtu=1500 link=0 master=0
ref=27 state=start present fw_flags=3800 flags=up broadcast run multicast
Qdisc=mq hw_addr=70:4c:a5:7d:d4:95 broadcast_addr=ff:ff:ff:ff:ff:ff
ingress traffic control:
        bandwidth=100000(kbps) lock_hit=148731 default_class=2 n_active_class=4
        class-id=2      allocated-bandwidth=10000(kbps)         guaranteed-bandwidth=10000(kbps)
                        max-bandwidth=60000(kbps)       current-bandwidth=10004(kbps)
                        priority=low    forwarded_bytes=2267956K
                        dropped_packets=10389K  dropped_bytes=13796469K
        class-id=5      allocated-bandwidth=1000(kbps)  guaranteed-bandwidth=10000(kbps)
                        max-bandwidth=50000(kbps)       current-bandwidth=0(kbps)
                        priority=medium         forwarded_bytes=0
                        dropped_packets=0       dropped_bytes=0
        class-id=3      allocated-bandwidth=35000(kbps)         guaranteed-bandwidth=20000(kbps)
                        max-bandwidth=100000(kbps)      current-bandwidth=35729(kbps)
                        priority=high   forwarded_bytes=2119502K
                        dropped_packets=6020K   dropped_bytes=7994926K
        class-id=4      allocated-bandwidth=54000(kbps)         guaranteed-bandwidth=30000(kbps)
                        max-bandwidth=100000(kbps)      current-bandwidth=53907(kbps)
                        priority=high   forwarded_bytes=902415K
                        dropped_packets=4141K   dropped_bytes=5499248K
stat: rxp=197827723 txp=2433885 rxb=227356779526 txb=2771602657 rxe=0 txe=0 rxd=0 txd=0 mc=0
collision=0 @ time=1628816440
re: rxl=0 rxo=0 rxc=0 rxf=0 rxfi=0 rxm=0
te: txa=0 txc=0 txfi=0 txh=0 txw=0
misc rxc=0 txc=0
input_type=0 state=3 arp_entry=0 refcnt=27
```

Internet Services

The Internet Service Database (ISDB) data comes from the FortiGuard service system. Information is regularly added to this database, for example, geographic location, IP reputation, popularity & DNS, and so on. All this information helps users define Internet security more effectively. You can use the contents of the database as criteria for inclusion or exclusion in a policy.

Internet Service can be applied both as source and destination objects in a policy. You can also apply Internet Services to shaping policy.

The Internet Service Database can be configured with different numbers of addresses:

```
config system global
    set internet-service-database {mini | standard | full | on-demand}
end
```

Value	Description
mini	Small sized Internet Service database with very limited IP addresses.
standard	Medium sized Internet Service database with most IP addresses.
full	Full sized Internet Service database with all IP addresses.
on-demand	Internet Service database with customer selected IP addresses.

The default value varies between model. For FortiGate models that have 2GB of memory, it is recommended that the on-demand database be used.

The following topics provide instructions on configuring policies with Internet Service:

- Using Internet Service in a policy on page 2179
- Using custom Internet Service in policy on page 2183
- Using extension Internet Service in policy on page 2185
- Global IP address information database on page 2188
- IP reputation filtering on page 2190
- Internet service groups in policies on page 2191
- Allow creation of ISDB objects with regional information on page 2198
- Internet service customization on page 2200
- Look up IP address information from the Internet Service Database page on page 2201
- Internet Service Database on-demand mode on page 2202
- Enabling the ISDB cache in the FortiOS kernel on page 2205
- ISDB FQDN groups on page 2206

Using Internet Service in a policy

This topic shows how to apply a predefined Internet Service entry into a policy.

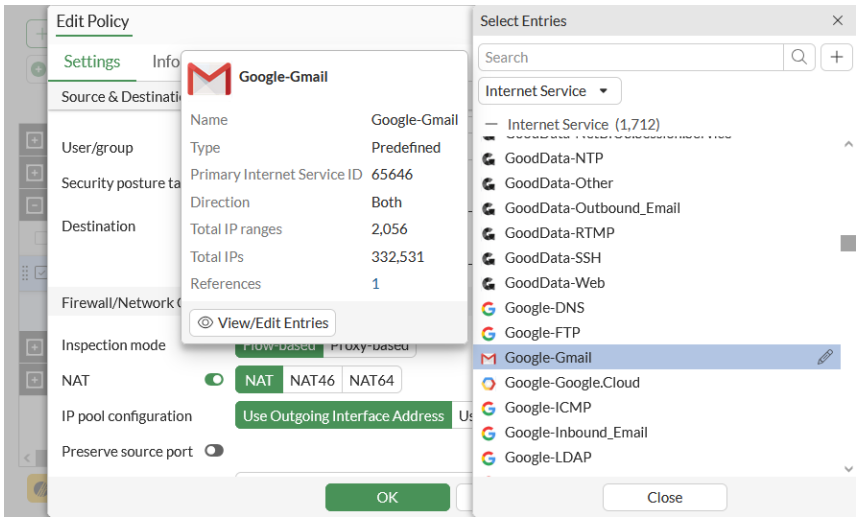
There are three types of Internet Services you can apply to a firewall policy:

- Predefined Internet Services
- Custom Internet Services
- Extension Internet Services

Sample IPv4 configuration

To apply a predefined Internet Service entry to a policy using the GUI:

1. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
2. Click in the *Destination* field.
3. In the *Select Entries* pane, select *Internet Service* from the dropdown list and select *Google-Gmail*.



4. Configure the remaining fields as needed.
5. Click *OK*.

To apply a predefined Internet Service entry to a policy in the CLI:

In the CLI, enable the `internet-service` first and then use its ID to apply the policy.

This example uses Google Gmail and its ID is 65646. Each Internet Service has a unique ID.

```
config firewall policy
    edit 9
        set name "Internet Service in Policy"
        set srcintf "wan2"
        set dstintf "wan1"
        set srcaddr "all"
        set internet-service enable
        set internet-service-id 65646
        set action accept
        set schedule "always"
        set utm-status enable
        set av-profile "g-default"
        set ssl-ssh-profile "certificate-inspection"
        set nat enable
    next
end
```

To diagnose an Internet Service entry in the CLI:

```
# diagnose internet-service id-summary 65646
Version: 0000600096
Timestamp: 201902111802
Total number of IP ranges: 444727
Number of Groups: 7
Group(0), Singularity(20), Number of IP ranges(142740)
Group(1), Singularity(19), Number of IP ranges(1210)
```

```
Group(2), Singularity(16), Number of IP ranges(241)
Group(3), Singularity(15), Number of IP ranges(38723)
Group(4), Singularity(10), Number of IP ranges(142586)
Group(5), Singularity(8), Number of IP ranges(5336)
Group(6), Singularity(6), Number of IP ranges(113891)
Internet Service: 65646(Google.Gmail)
Number of IP range: 60
Number of IP numbers: 322845
Singularity: 15
Reputation: 5(Known and verified safe sites such as Gmail, Amazon, eBay, etc.)
Icon Id: 510
Second Level Domain: 53(gmail.com)
Direction: dst
Data source: isdb
```

Result

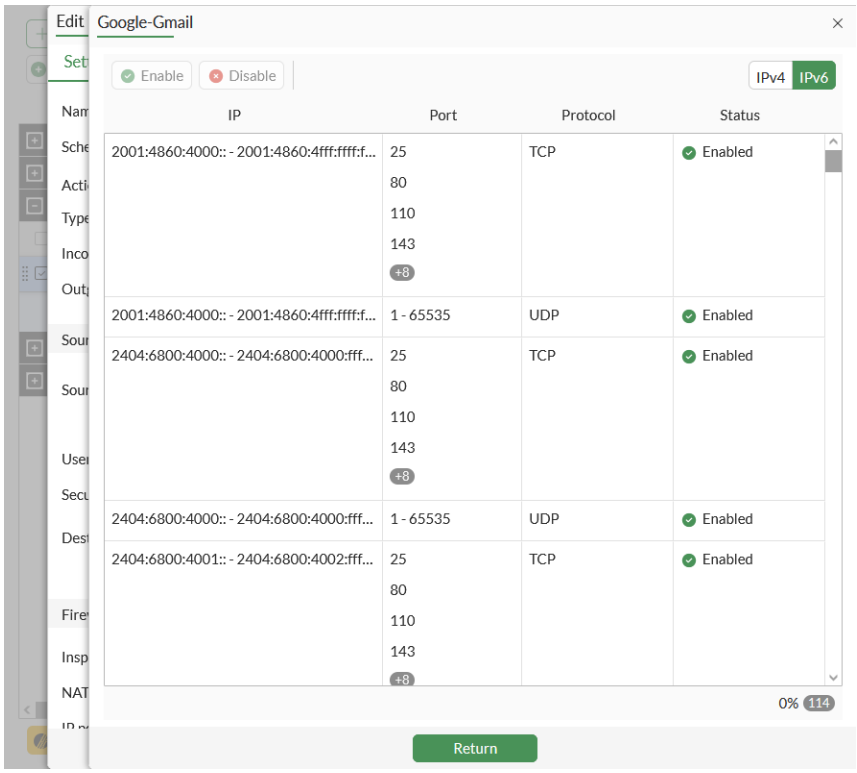
Because the IP and services related to Google Gmail on the Internet are included in this Internet Service (65646), all traffic to Google Gmail is forwarded by this policy.

Sample IPv6 configuration

In this example, the Google Gmail IPv6 ISDB address (ID 65646) is used as a destination in a firewall policy.

To apply a predefined IPv6 Internet Service entry to a policy using the GUI:

1. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
2. In the *Destination* field, select *Internet Service* from the dropdown list.
3. In the *IPv6 Internet Service* section, select *Google-Gmail*.
4. Optionally, hover over the *Google Gmail* and click *View/Edit Entries*. A pane appears that displays the IPv6 address ranges for this Internet Service.



5. Click *Return* to close the pane.
6. Configure the other settings as needed.
7. Click *OK*.

To apply a predefined IPv6 Internet Service entry to a policy using the CLI:

```
config firewall policy
    edit 4
        set name "Internet Service6 policy"
        set srcintf "vlan100"
        set dstintf "wan1"
        set action accept
        set srcaddr6 "all"
        set internet-service6 enable
        set internet-service6-name "Google-Gmail"
        set schedule "always"
        set nat enable
    next
end
```

To diagnose an IPv6 Internet Service entry in the CLI:

```
# diagnose internet-service6 id-summary 65646

Version: 00007.02907
Timestamp: 202212161345
```

```
Total number of IP ranges: 36878
Number of Groups: 12
Group(0), Singularity(20), Number of IP ranges(60)
Group(1), Singularity(18), Number of IP ranges(12)
Group(2), Singularity(17), Number of IP ranges(2728)
Group(3), Singularity(16), Number of IP ranges(2812)
Group(4), Singularity(15), Number of IP ranges(4011)
Group(5), Singularity(10), Number of IP ranges(2345)
Group(6), Singularity(9), Number of IP ranges(14)
Group(7), Singularity(8), Number of IP ranges(1555)
Group(8), Singularity(7), Number of IP ranges(2704)
Group(9), Singularity(6), Number of IP ranges(7300)
Group(10), Singularity(5), Number of IP ranges(3154)
Group(11), Singularity(4), Number of IP ranges(10183)
Internet Service: 65646(Google-Gmail)
Number of IP ranges: 482
Singularity: 15
Icon Id: 510
Direction: both
Data source: isdb
Country: 32 36 56 76 124 152 158 203 208 246 250 276 344 348 356 372 376 380 392 404 458 484
        528 616 634 643 682 702 710 724 752 756 784 826 840
Region: 65535
City: 65535
```

Result

Because the IP and services related to Google Gmail on the Internet are included in this Internet Service (65646), all traffic to Google Gmail is forwarded by this policy.

Using custom Internet Service in policy

Custom Internet Services can be created and used in firewall policies.

When creating a custom Internet Service, you must set following elements:

- IP or IP ranges
- Protocol number
- Port or port ranges
- Reputation

You must use CLI to create a custom Internet Service, except for geographic based services (see Allow creation of ISDB objects with regional information on page 2198).

CLI syntax

```
config firewall internet-service-custom
    edit <name>
```

```
        set comment <comment>
        set reputation {1 | 2 | 3 | 4 | 5}
        config entry
            edit <ID>
                set protocol <protocol #>
                set dst <object_name>
                config port-range
                    edit <ID>
                        set start-port <port #>
                        set end-port <port #>
                    next
                end
            next
        end
    end
end
```

Sample configuration

To configure a custom Internet Service:

```
config firewall internet-service-custom
   edit "test-isdb-1"
        set comment "Test Custom Internet Service"
        set reputation 4
        config entry
           edit 1
               set protocol 6
               config port-range
                   edit 1
                       set start-port 80
                       set end-port 443
                   next
               end
               set dst "10-1-100-0"
           next
           edit 2
               set protocol 6
               config port-range
                   edit 1
                       set start-port 80
                       set end-port 80
                   next
               end
               set dst "172-16-200-0"
           next
        end
   next
end
```

To apply a custom Internet Service into a policy:

```
config firewall policy
    edit 1
        set name "Internet Service in Policy"
        set srcintf "wan2"
        set dstintf "wan1"
        set srcaddr "all"
        set internet-service enable
        set internet-service-id 65646
        set internet-service-custom "test-isdb-1"
        set action accept
        set schedule "always"
        set utm-status enable
        set av-profile "g-default"
        set ssl-ssh-profile "certificate-inspection"
        set nat enable
    next
end
```

Result

In addition to the IP address, IP address ranges, and services allowed by Google.Gmail, this policy also allows the traffic which access to 10.1.100.0/24 and TCP/80-443 and 172.16.200.0/24 and TCP/80.

Using extension Internet Service in policy

Extension Internet Service lets you add custom or remove existing IP address and port ranges to an existing predefined Internet Service entries. Using an extension type Internet Service is actually editing a predefined type Internet Service entry and adding IP address and port ranges to it.

When creating an extension Internet Service and adding custom ranges, you must set following elements:

- IP or IP ranges
- Protocol number
- Port or port ranges

You must use CLI to add custom IP address and port entries into a predefined Internet Service.

You must use GUI to remove entries from a predefined Internet Service.

Custom extension Internet Service CLI syntax

```
config firewall internet-service-extension
    edit <ID #>
        set comment <comment>
        config entry
            edit <ID #>
                set protocol <number #>
```

```
                set dst <object_name>
                config port-range
                    edit <ID #>
                         set start-port <number #>
                         set end-port <number #>
                    next
                end
            next
        end
    end
end
```

Sample configuration

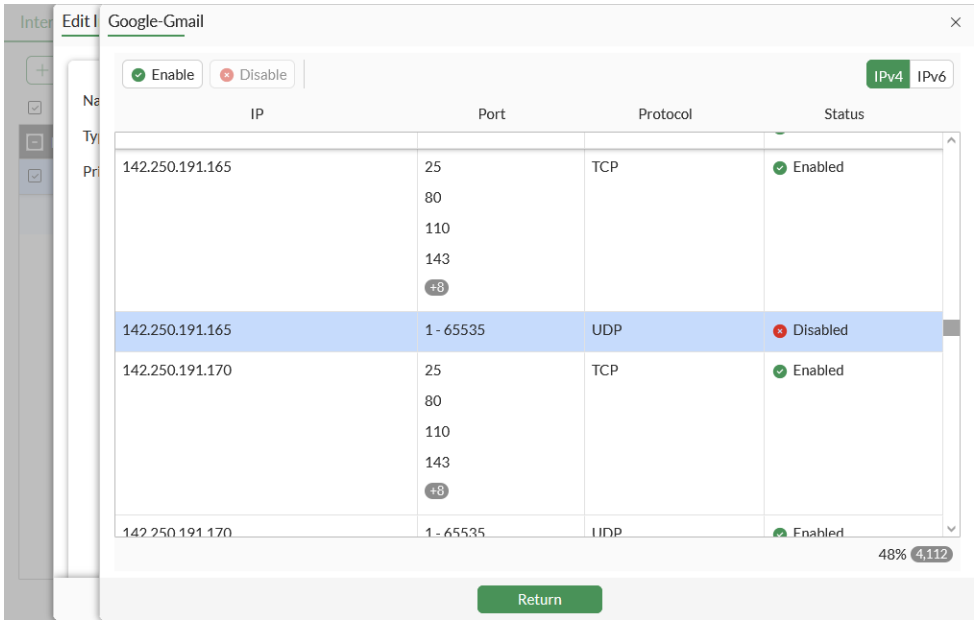
To configure an extension Internet Service in the CLI:

```
config firewall internet-service-extension
    edit 65646
        set comment "Test Extension Internet Service 65646"
        config entry
            edit 1
                set protocol 6
                config port-range
                    edit 1
                        set start-port 80
                        set end-port 443
                    next
                end
                set dst "172-16-200-0"
            next
            edit 2
                set protocol 17
                config port-range
                    edit 1
                        set start-port 53
                        set end-port 53
                    next
                end
                set dst "10-1-100-0"
            next
        end
    next
end
```

To remove IP address and port entries from an existing Internet Service in the GUI:

1. Go to *Policy & Objects > Internet Service Database*.
2. Search for *google-gmail*.
3. Select *Google-Gmail* and click *Edit*.
4. In the gutter, click *View/Edit Entries*.

5. Select the *IP* entry that you need to remove and click *Disable*.



6. Click *Return* twice.

To remove IP address and port entries from an existing Internet Service in the CLI:

```
config firewall internet-service-extension
    edit 65646
        config disable-entry
            edit 1
                set protocol 17
                config port-range
                    edit 1
                    next
                end
                config ip-range
                    edit 1
                        set start-ip 142.250.191.165
                        set end-ip 142.250.191.165
                    next
                end
            next
        end
    next
end
```

To apply an extension Internet Service into policy in the CLI:

```
config firewall policy
    edit 9
        set name "Internet Service in Policy"
        set srcintf "wan2"
```

```
        set dstintf "wan1"
        set srcaddr "all"
        set internet-service enable
        set internet-service-id 65646
        set action accept
        set schedule "always"
        set utm-status enable
        set av-profile "g-default"
        set ssl-ssh-profile "certificate-inspection"
        set nat enable
    next
end
```

Result

In addition to the IP addresses, IP address ranges, and services allowed by Google.Gmail, this policy also allows the traffic which accesses 10.1.100.0/24 and UDP/53 and 172.16.200.0/24 and TCP/80-443. At the same time, the traffic that accesses 2.20.183.160 is dropped because this IP address and port is disabled from Google.Gmail.

Global IP address information database

The Internet Service and IP Reputation databases download details about public IP address, including: ownership, known services, geographic location, blocklisting information, and more. The details are available in drilldown information, tooltips, and other mechanisms in the FortiView and other pages.

The global IP address database is an integrated database containing all public IP addresses, and is implemented in the Internet Service Database.

To view the owner of the IP address:

```
(global) # get firewall internet-service-owner ?
    id    Internet Service owner ID.
    1  Google
    2  Facebook
    3  Apple
    4  Yahoo
    5  Microsoft
    ......
    115  Cybozu
    116  VNC
```

To check for any known service running on an IP address:

```
(global) # diagnose internet-service info FG-traffic 6 80 8.8.8.8
    Internet Service: 65537(Google.Web)
```

To check GeoIP location and blocklist information:

```
(global) # diagnose internet-service id 65537 | grep 8.8.8.8
    8.8.8.8-8.8.8.8 geo_id(11337) block list(0x0) proto(6) port(80 443)
    8.8.8.8-8.8.8.8 geo_id(11337) block list(0x0) proto(17) port(443)
```

To check a known malicious server:

```
(global) # diagnose internet-service id-summary 3080383
    Version: 0000600096
    Timestamp: 201902111802
    Total number of IP ranges: 444727
    Number of Groups: 7
    Group(0), Singularity(20), Number of IP ranges(142740)
    Group(1), Singularity(19), Number of IP ranges(1210)
    Group(2), Singularity(16), Number of IP ranges(241)
    Group(3), Singularity(15), Number of IP ranges(38723)
    Group(4), Singularity(10), Number of IP ranges(142586)
    Group(5), Singularity(8), Number of IP ranges(5336)
    Group(6), Singularity(6), Number of IP ranges(113891)
    Internet Service: 3080383(Botnet.C&C.Server)
    Number of IP range: 111486
    Number of IP numbers: 111486
    Singularity: 20
    Reputation: 1(Known malicious sites related to botnet servers, phishing sites, etc.)
    Icon Id: 591
    Second Level Domain: 1(other)
    Direction: dst
    Data source: irdb
```

To check questionable usage:

```
(global) # diagnose internet-service id-summary 2818238
    Version: 0000600096
    Timestamp: 201902111802
    Total number of IP ranges: 444727
    Number of Groups: 7
    Group(0), Singularity(20), Number of IP ranges(142740)
    Group(1), Singularity(19), Number of IP ranges(1210)
    Group(2), Singularity(16), Number of IP ranges(241)
    Group(3), Singularity(15), Number of IP ranges(38723)
    Group(4), Singularity(10), Number of IP ranges(142586)
    Group(5), Singularity(8), Number of IP ranges(5336)
    Group(6), Singularity(6), Number of IP ranges(113891)
    Internet Service: 2818238(Tor.Relay.Node)
    Number of IP range: 13718
    Number of IP numbers: 13718
    Singularity: 20
    Reputation: 2(Sites providing high risk services such as TOR, proxy, P2P, etc.)
    Icon Id: 43
    Second Level Domain: 1(other)
```

```
    Direction: dst
    Data source: irdb

(global) # diagnose internet-service id-summary 2818243
    Version: 0000600096
    Timestamp: 201902111802
    Total number of IP ranges: 444727
    Number of Groups: 7
    Group(0), Singularity(20), Number of IP ranges(142740)
    Group(1), Singularity(19), Number of IP ranges(1210)
    Group(2), Singularity(16), Number of IP ranges(241)
    Group(3), Singularity(15), Number of IP ranges(38723)
    Group(4), Singularity(10), Number of IP ranges(142586)
    Group(5), Singularity(8), Number of IP ranges(5336)
    Group(6), Singularity(6), Number of IP ranges(113891)
    Internet Service: 2818243(Tor.Exit.Node)
    Number of IP range: 1210
    Number of IP numbers: 1210
    Singularity: 19
    Reputation: 2(Sites providing high risk services such as TOR, proxy, P2P, etc.)
    Icon Id: 43
    Second Level Domain: 1(other)
    Direction: src
    Data source: irdb
```

IP reputation filtering

There are currently five reputation levels in the Internet Service Database (ISDB), and custom reputation levels can be defined in a custom internet service. You can configure firewall policies to filter traffic according to the desired reputation level. If the reputation level of either the source or destination IP address is equal to or greater than the level set in the policy, then the packet is forwarded, otherwise, the packet is dropped.

The five default reputation levels are:

1	Known malicious sites, such as phishing sites or sites related to botnet servers
2	High risk services sites, such as TOR, proxy, and P2P
3	Unverified sites
4	Reputable social media sites, such as Facebook and Twitter
5	Known and verified safe sites, such as Gmail, Amazon, and eBay

The default minimum reputation level in a policy is zero, meaning that the reputation filter is disabled.

For IP addresses that are not included in the ISDB, the default reputation level is three.

The default reputation direction is `destination`.

Example 1

Packets from the source IP address with reputation levels three, four, or five will be forwarded by this policy.

To set the reputation level and direction in a policy using the CLI:

```
config firewall policy
    edit 1
        set srcintf "wan2"
        set dstintf "port1"
        set srcaddr “all”
        set dstaddr "all"
        set reputation-minimum 3
        set reputation-direction source
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set auto-asic-offload disable
        set nat enable
    next
end
```

Packets from the source IP address with reputation levels three, four, or five will be forwarded by this policy.

Example 2

This policy allows only outbound FTP traffic, if the destination server has a minimum reputation of 4.

To set the reputation level and direction in a policy using the CLI:

```
config firewall policy
    edit 1
        set srcintf "port1"
        set dstintf "wan2"
        set srcaddr “all”
        set dstaddr "all"
        set reputation-minimum 4
        set reputation-direction destination
        set action accept
        set schedule "always"
        set service "FTP"
        set logtraffic all
        set auto-asic-offload disable
        set nat enable
    next
end
```

Internet service groups in policies

Groups of internet services can be created using the GUI or CLI. Once created, you can use the group with traffic shaping and firewall policies as criteria to match and shape traffic. The shaper is applied when the traffic matches.

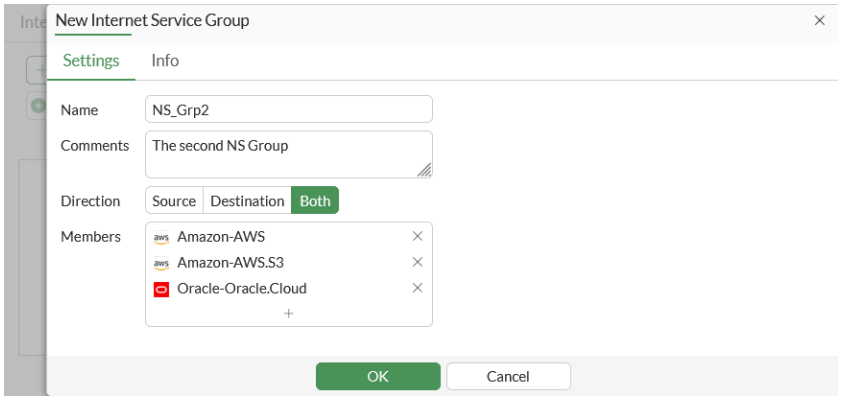
Internet service groups can be used as the source and destination of the policy. When using the CLI, `internet-service` must be enabled to use an internet service group as a destination, and `internet-service-src` must be enabled to use an internet service group as a source.

The following CLI variables are available in the `firewall policy` and `firewall shaping-policy` commands:

Variable	Description
`internet-service-group <string>`	Internet service group name.
`internet-service-custom-group <string>`	Custom internet service group name.
`internet-service-src-group <string>`	Source internet service group name.
`internet-service-src-custom-group <string>`	Custom source internet service group name.

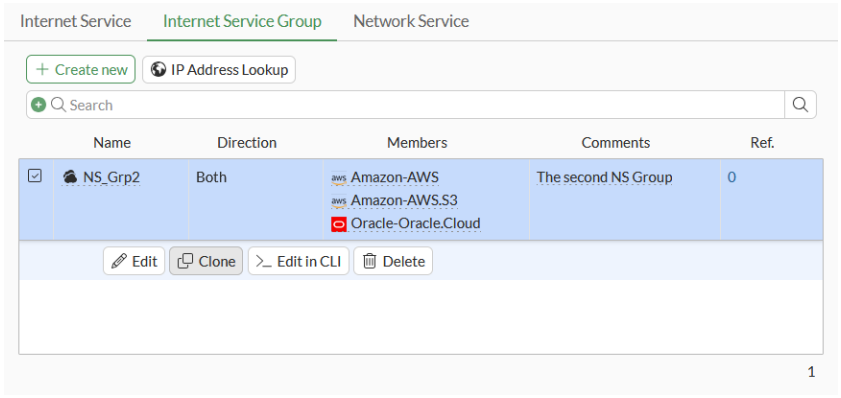
To create an internet service group in the GUI:

1. Go to *Policy & Objects* > *Internet Service Database*, select the *Internet Service Group* tab, and click *Create New*.
2. Configure the settings as needed, and click *OK*.



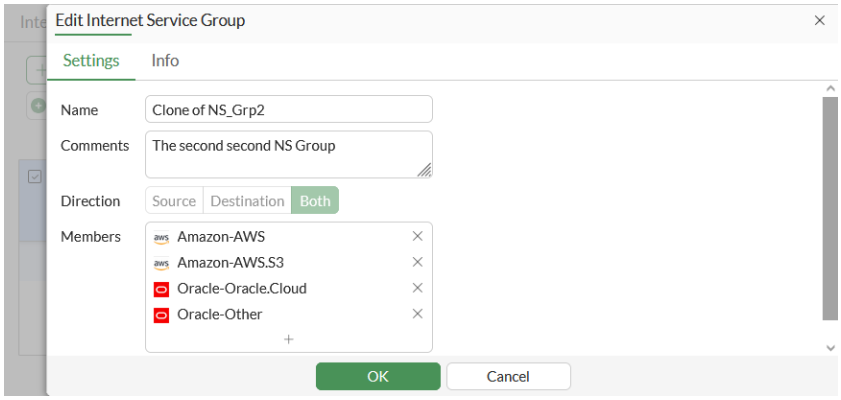
To clone an internet service group:

1. Go to *Policy & Objects* > *Internet Service Database*, and click the *Internet Service Group* tab.
2. Select an internet service group, and click *Clone*.



The *Edit Internet Service Group* pane opens.

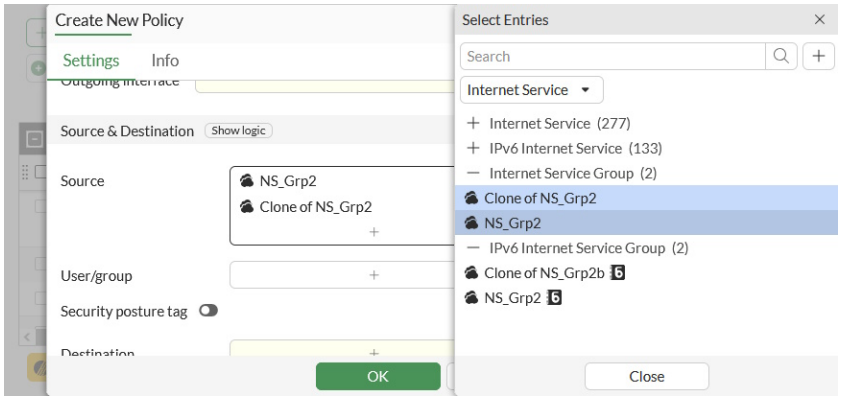
3. Edit the options.



4. Click *OK*. The clone is created.

To apply the internet service group to a firewall policy:

1. Go to *Policy & Objects* > *Firewall Policy*, and click *Create New*.
2. Select the internet service group:
   a. Click the *Source* or *Destination* field. The *Select Entries* pane opens.
   b. Select *Internet Service*, and select one or more internet service groups, such as *NS_Grp2*.
   c. Click *Close*.



3. Configure the remaining options as needed, then click *OK*.

Examples

The following examples use the below topology.

Example 1

In this example, the PC is allowed to access Google, so all Google services are put into an internet service group.

To configure access to Google services using an internet service group using the CLI:

1. Create an internet service group for Google services.
   In this example, the internet service group is named `Google_Group`.

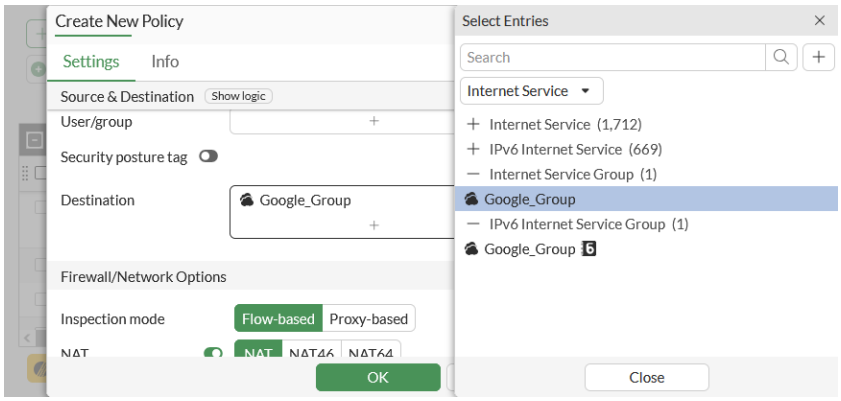
```
config firewall internet-service-group
    edit "Google_Group"
        set direction destination
        set member Google-Other Google-Web Google-ICMP Google-DNS Google-Outbound_Email
Google-SSH Google-FTP Google-NTP Google-Inbound_Email Google-LDAP Google-
NetBIOS.Session.Service Google-RTMP Google-NetBIOS.Name.Service Google-Google.Cloud Google-
Gmail
    next
end
```

2. Create a firewall policy and select the internet service group to allow access to all Google services from the PC.
   In this example, `internet-service` is `enabled`, and the internet service group named `Google_Group` is selected to allow access to all Google services from the PC.

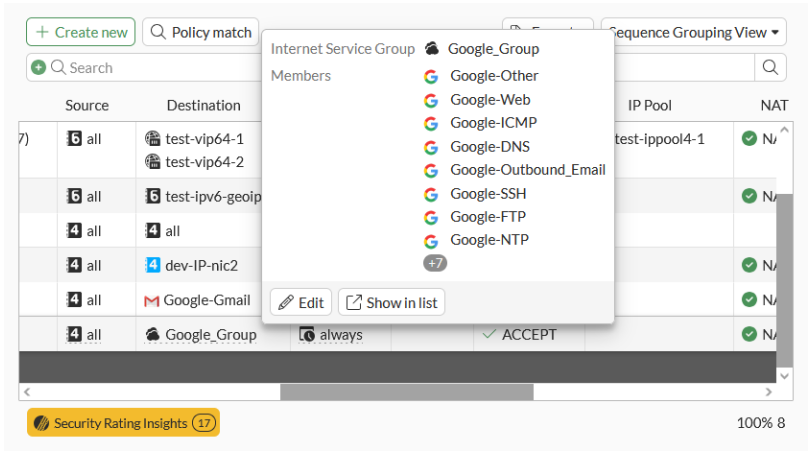
```
config firewall policy
    edit 1
        set name "PC to Google"
        set srcintf "port2"
        set dstintf "port1"
        set srcaddr "all"
        set internet-service enable
        set internet-service-group "Google_Group"
        set action accept
        set schedule "always"
        set fsso disable
        set nat enable
    next
end
```

To configure access to Google services using an internet service group in the GUI:

1. On the FortiGate, create an internet service group.
2. Go to *Policy & Objects > Firewall Policy*, and create a new policy.
3. Set the *Destination* to the internet service group.



4. Configure the remaining options, and click *OK*.
5. Go to *Policy & Objects* > *Firewall Policy* and hover over the group to view a list of its members.



Example 2

In this example, two office FTP servers are put into a custom internet service group, and the PC connection to the FTP servers is limited to 1Mbps.

To put two FTP servers into a custom internet service group and limit the PC connection speed to them in the CLI:

1. Create a custom internet service group for the internal FTP servers in the CLI:

```
config firewall internet-service-custom
    edit "FTP_PM"
        config entry
            edit 1
                config port-range
                    edit 1
                        set start-port 21
                        set end-port 21
                    next
                end
                set dst "PM_Server"
```

```
            next
        end
    next
    edit "FTP_QA"
        config entry
            edit 1
                config port-range
                    edit 1
                        set start-port 21
                        set end-port 21
                    next
                end
                set dst "QA_Server"
            next
        end
    next
end
```

2. Edit a firewall policy and add the custom internet service group to it:

```
config firewall internet-service-custom-group
    edit "Internal_FTP"
        set member "FTP_QA" "FTP_PM"
    next
end
```

3. Create a traffic shaper to limit the maximum bandwidth:

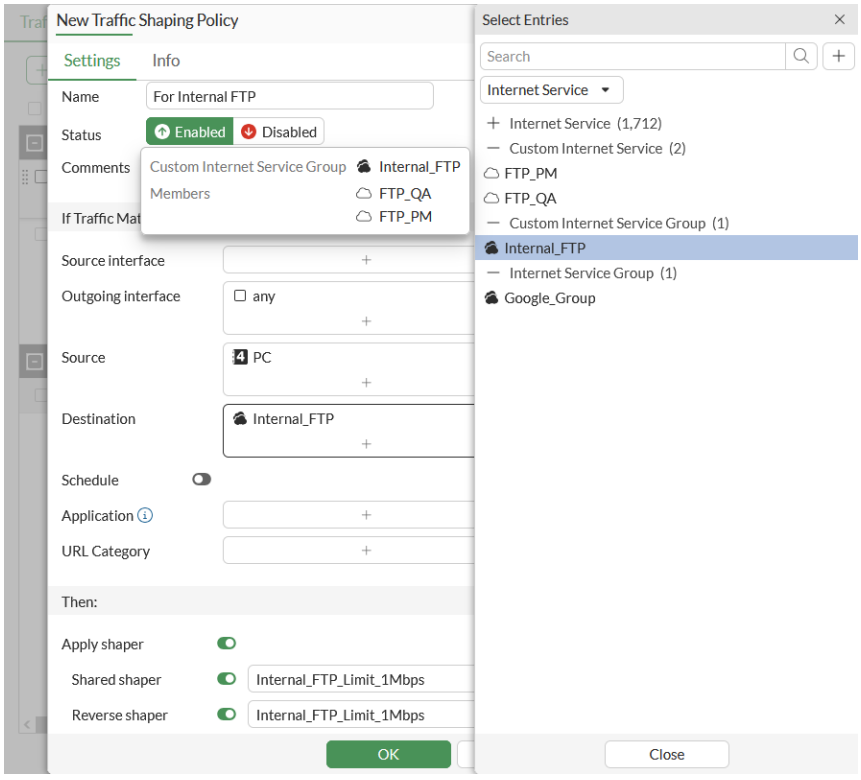
```
config firewall shaper traffic-shaper
    edit "Internal_FTP_Limit_1Mbps"
        set guaranteed-bandwidth 500
        set maximum-bandwidth 1000
        set priority medium
    next
end
```

4. Create a firewall shaping policy to limit the speed from the PC to the internal FTP servers:

```
config firewall shaping-policy
    edit 1
        set name "For Internal FTP"
        set internet-service enable
        set internet-service-custom-group "Internal_FTP"
        set dstintf "port1"
        set traffic-shaper "Internal_FTP_Limit_1Mbps"
        set traffic-shaper-reverse "Internal_FTP_Limit_1Mbps"
        set srcaddr "PC"
    next
end
```

To put two FTP servers into a custom service group and limit the PC connection speed to the in the GUI:

1. Create custom internet services for the internal FTP servers using the CLI.
2. Create a custom internet server group and add the just created custom internet services to it using the CLI.
3. Create a traffic shaper to limit the maximum bandwidth:
   a. Go to *Policy & Objects* > *Traffic Shaping*, select the *Traffic Shaper* tab, and click *Create New*.
   b. Enter a *Name* for the shaper, such as *Internal_FTP_Limit_1Mbps*.
   c. Set the *Traffic Priority* to *Medium*.
   d. Enable *Max Bandwidth* and set it to *1000*.
   e. Enable *Guaranteed Bandwidth* and set it to *500*.
   f. Click *OK*.
4. Create a firewall shaping policy to limit the speed from the PC to the internal FTP servers:
   a. Go to *Policy & Objects* > *Traffic Shaping*, select the *Traffic Shaping Policy* tab, and click *Create New*.
   b. Enter a name for the policy.
   c. Set *Source* to the address of the PC.
   d. Set the *Destination* to the just created custom internet service group.
   e. Enable *Apply shaper* and apply the just create traffic shaper to the *Shared shaper* and *Reverse shaper*.



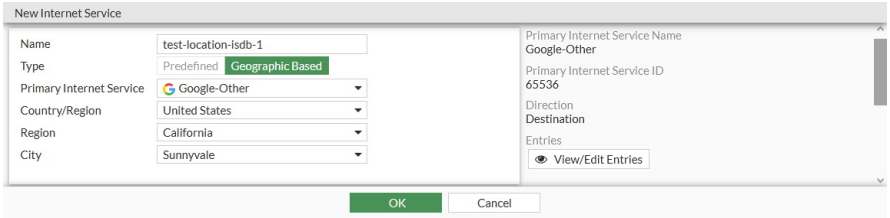
   f. Configure the remaining options as required, then click *OK*.

Allow creation of ISDB objects with regional information

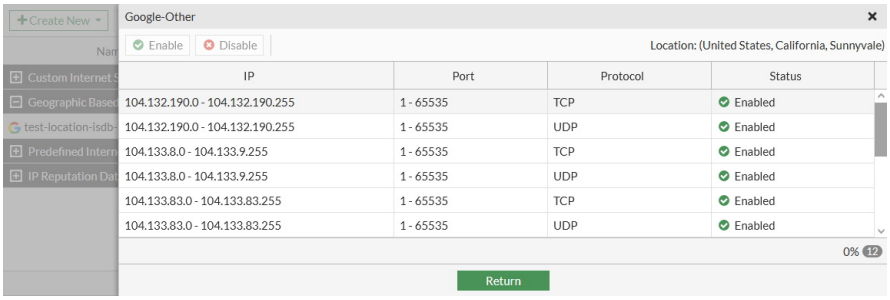
Geographic-based Internet Service Database (ISDB) objects allow users to define a country, region, and city. These objects can be used in firewall policies for more granular control over the location of the parent ISDB object. ISDB objects are now referenced in policies by name instead of ID.

To apply a location-based ISDB object to a policy in the GUI:

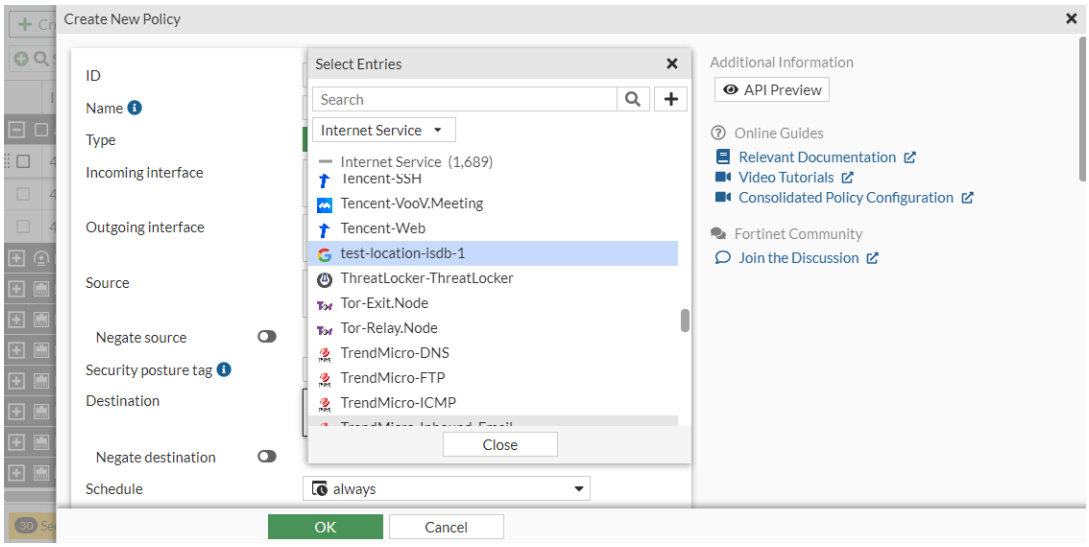
1. Create the ISDB object:
   a. Go to *Policy & Objects* > *Internet Service Database* and click *Create New* > *Geographic Based Internet Service*.
   b. Configure the settings as required.



   c. Click *OK*.
2. View the IP ranges in the location-based internet service:
   a. Go to *Policy & Objects* > *Internet Service Database* .
   b. In the table, hover over the object created in step 1 and click *View/Edit Entries*. The list of IPs is displayed:



   c. Click *Return*.
3. Add the ISDB object to a policy:
   a. Go to *Policy & Objects* > *Firewall Policy* and create a new policy or edit an existing one.
   b. For *Destination*, select *Internet Service* from the dropdown list and select the ISDB object created in step 1.
   c. Configure the other settings as needed.



d. Click *OK*.

To apply a location-based ISDB object to a policy in the CLI:

1. Create the ISDB object:

```
config firewall internet-service-name
    edit "test-location-isdb-1"
        set type location
        set internet-service-id 65536
        set country-id 840
        set region-id 283
        set city-id 23352
    next
end
```

2. View the IP ranges in the location-based internet service:

```
# diagnose internet-service id 65536 | grep "country(840) region(283) city(23352)"
96.45.33.73-96.45.33.73 country(840) region(283) city(23352) blocklist(0x0) reputation(4),
domain(5) popularity(0) botnet(0) proto(6) port(1-65535)
96.45.33.73-96.45.33.73 country(840) region(283) city(23352) blocklist(0x0) reputation(4),
domain(5) popularity(0) botnet(0) proto(17) port(1-65535)
198.94.221.56-198.94.221.56 country(840) region(283) city(23352) blocklist(0x0) reputation(4),
domain(5) popularity(4) botnet(0) proto(6) port(1-65535)
198.94.221.56-198.94.221.56 country(840) region(283) city(23352) blocklist(0x0) reputation(4),
domain(5) popularity(4) botnet(0) proto(17) port(1-65535)
```

3. Add the ISDB object to a policy:

```
config firewall policy
    edit 3
        set name "PC to Google"
        set srcintf "port2"
        set dstintf "port1"
```

```
        set srcaddr "PC"
        set internet-service enable
        set internet-service-name "test-location-isdb-1"
        set action accept
        set schedule "always"
        set logtraffic all
        set logtraffic-start enable
        set auto-asic-offload disable
        set nat enable
    next
end
```

Internet service customization

Internet Service Database (ISDB) entries can be tuned for their environments by adding custom ports and port ranges, as well as port mapping.

If you are in multi-VDOM mode, Internet service customization can only occur at the Global level and not in a VDOM. See VDOM overview on page 3835 for more information.

To add a custom port range:

```
config firewall internet-service-addition
    edit 65646
        set comment "Add custom port-range:tcp/8080-8090 into 65646"
        config entry
            edit 1
                set protocol 6
                config port-range
                    edit 1
                        set start-port 8080
                        set end-port 8090
                    next
                end
            next
        end
    next
end
Warning: Configuration will only be applied after rebooting or using the 'execute internet-service
refresh' command.
```

To verify that the change was applied:

```
 # diagnose internet-service info FG-traffic 6 8080 2.20.183.160
 Internet Service: 65646(Google.Gmail)
```

To configure additional port mapping:

```
config firewall internet-service-append
    set match-port 10
    set append-port 20
end
Warning: Configuration will only be applied after rebooting or using the 'execute internet-service
refresh' command.
```

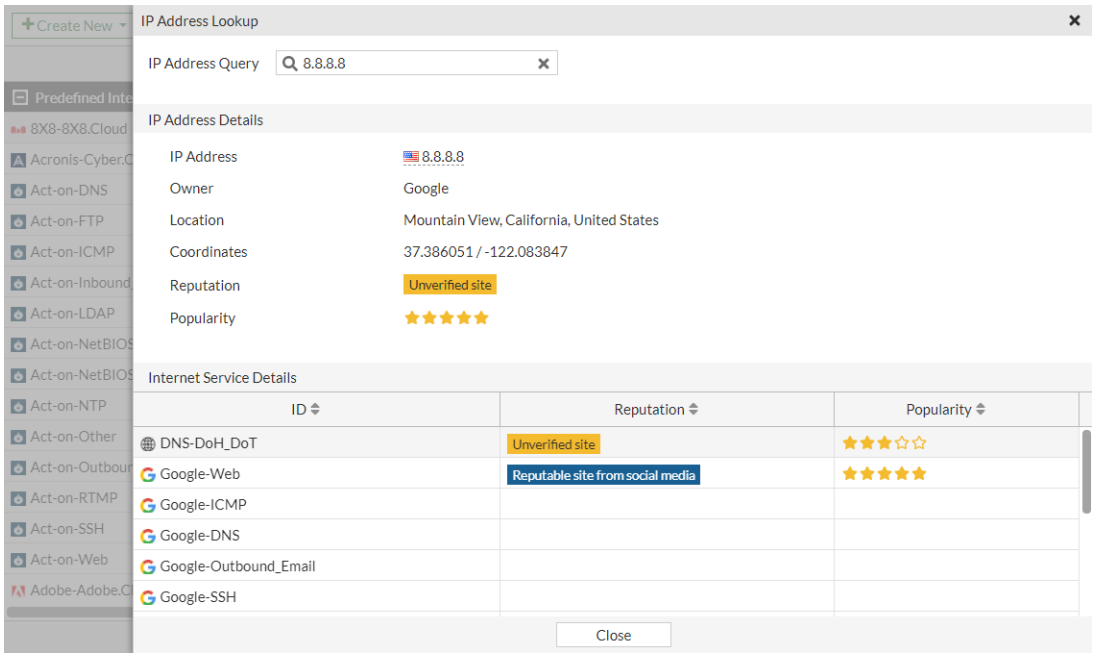
Look up IP address information from the Internet Service Database page

The *IP Address Lookup* button allows users to look up IP address information from the Internet Service Database and GeoIP Database. Returned IP address information includes the reverse IP address/domain lookup, location, reputation, and other internet service information.

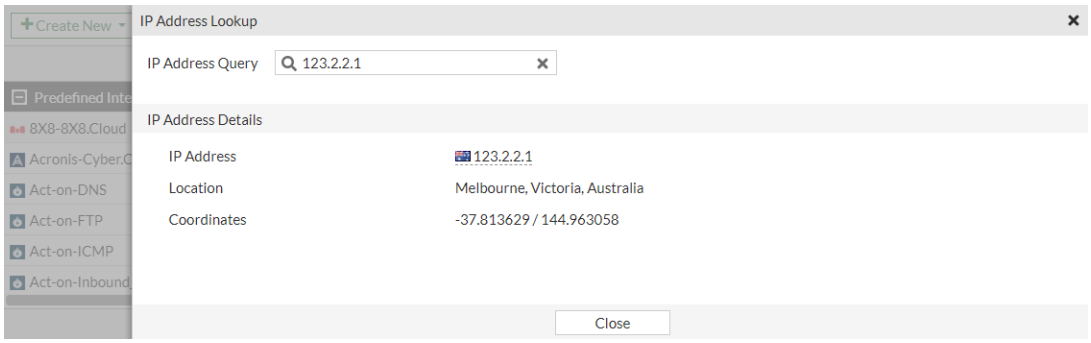
To look up IP address information:

1. Go to *Policy & Objects* > *Internet Service Database*.
2. Click *IP Address Lookup*. The *IP Address Lookup* pane opens.
3. In the *IP Address Query* field, enter the IP address and press `Enter`.

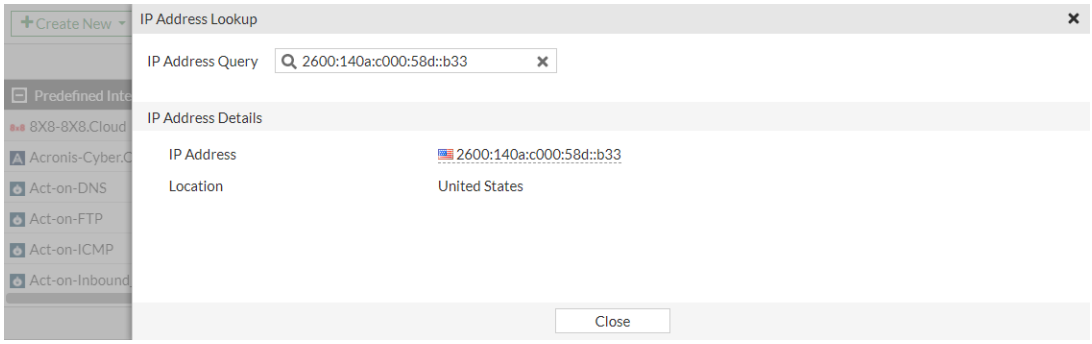
   Results of an IP address from the Internet Service Database:



   Results of an IP address from the GeoIP Database:



Results of an IPv6 address from the GeoIP Database:



4. Click *Close*.

Internet Service Database on-demand mode

Internet Service Database (ISDB) on-demand mode replaces the full-sized ISDB file with a much smaller file that is downloaded onto the flash drive. This file contains only the essential entries for Internet Services. When a service is used in a firewall policy, the FortiGate queries FortiGuard to download the IP addresses and stores them on the flash drive. The FortiGate also queries the local MAC Database (MADB) for corresponding MAC information. The content of the ISDB entries used in firewall policies persists through reboots.

For FortiGate models that have 2GB of memory, it is recommended that the on-demand database be used.

To enable ISDB (FFDB) on-demand mode:

1. Configure the global setting:

```
config system global
    set internet-service-database on-demand
end
```

All FFDB files are erased.

2. Verify that there are no ISDB (FFDB) files:

```
# diagnose autoupdate versions | grep Internet -A 6
Internet-service On-Demand Database
---------
Version: 0.00000
```

```
Contract Expiry Date: n/a
Last Updated using manual update on Mon Jan  1 00:00:00 2001
Last Update Attempt: n/a
Result: Updates Installed
```

Shortly after, the ISDB (FFDB) data structure is downloaded on the FortiGate. The following message appears in the debug messages:

```
do_ffsr_update[1567]-Starting  Update FFDB ondemand:(not final retry)
```

3. Run diagnostics again to verify that the ISDB (FFDB) files are saved on the FortiGate flash drive:

```
# diagnose autoupdate versions | grep Internet -A 6
Internet-service On-Demand Database
---------
Version: 7.02950
Contract Expiry Date: n/a
Last Updated using manual update on Fri Jan  6 06:45:00 2023
Last Update Attempt: n/a
Result: Updates Installed
```

4. Since no services have been applied to a policy, the IP range and IP address values are blank in the the summary details. For example, check the summary details for ID 1245187, Fortinet DNS:

```
# diagnose internet-service id-summary 1245187
Version: 00007.02950
Timestamp: 202301060645
Total number of IP ranges: 3085
Number of Groups: 1
Group(0), Singularity(90), Number of IP ranges(3085)
Internet Service: 1245187(Fortinet-DNS)
Number of IP ranges: 0
Number of IP addresses: 0
Singularity: 0
Icon Id: 19
Direction: dst
Data source: isdb
Country:
Region:
City:
```

5. Apply the Fortinet DNS service in a firewall policy:

```
config firewall policy
    edit 1
        set name "FDNS"
        set srcintf "port1"
        set dstintf "wan1"
        set action accept
        set srcaddr "all"
        set internet-service enable
        set internet-service-name "Fortinet-DNS"
```

```
        set schedule "always"
        set nat enable
    next
end
```

6. Verify the summary details again for ID 1245187 (Fortinet DNS). There is now data for the IP range and IP address values:

```
# diagnose internet-service id-summary 1245187
Version: 00007.02951
Timestamp: 202301061144
Total number of IP ranges: 3558
Number of Groups: 2
Group(0), Singularity(90), Number of IP ranges(3078)
Group(1), Singularity(10), Number of IP ranges(480)
Internet Service: 1245187(Fortinet-DNS)
Number of IP ranges: 480
Number of IP addresses: 55242
Singularity: 10
Icon Id: 19
Direction: dst
Data source: isdb
Country: 12 32 36 40 56 124 158 170 203 222 250 276 320 332 344 356 360 372 380 392 458 484
        528 591 600 604 642 643 702 764 784 807 826 840
Region: 55 132 159 169 251 261 283 444 501 509 529 565 596 634 697 709 721 742 744 758 776 860
        1002 1056 1073 1151 1180 1190 1195 1216 1264 1280 1283 1284 1287 1290 1315 1319 1348
1363 1373 1380 1387
        1437 1457 1509 1536 1539 1660 1699 1740 1752 1776 1777 1826 1833 1874 1906 1965 2014
2028 2039 2060 2063
        2147 2206 65535
City: 615 679 818 1001 1106 1117 1180 1207 1330 1668 1986 2139 2812 2868 3380 3438 3485 3670
4276 4588 4622 4904
        5334 5549 5654 5827 6322 6325 6330 6355 6652 7844 9055 10199 10333 11420 12930 13426
13685 13769 14107 14813 15121
        15220 15507 15670 16347 16561 16564 16567 16631 17646 17746 17885 17975 17995 18071
18476 19066 19285 20784 21065 21092 21136
        21146 21266 21337 21779 21993 22292 22414 22912 23352 23367 23487 23574 23635 23871
23963 24076 24203 24298 24611 24955 25050
        25332 26854 27192 27350 28825 28866 65535
```

To verify MAC vendor information:

```
# diagnose vendor-mac id 1
Vendor MAC: 1(ASUS)
Version: 0000100146
Timestamp: 202301031100
Number of MAC ranges: 85
00:04:0f:00:00:00 - 00:04:0f:ff:ff:ff
00:0c:6e:00:00:00 - 00:0c:6e:ff:ff:ff
00:0e:a6:00:00:00 - 00:0e:a6:ff:ff:ff
...
```

Troubleshooting

In on-demand mode, the ISDB will be updated automatically on schedule after a service is used in a firewall policy. However, to trigger an explicit download, use the following command:

```
# execute update-ffdb-on-demand
```

To ensure ISDB can be updated, the following setting must be enabled:

```
config system fortiguard
    set update-ffdb enable
end
```

To debug ISDB update issues in real-time, use the following real-time debugs:

```
# diagnose debug application update -1
# diagnose debug enable
```

Enabling the ISDB cache in the FortiOS kernel

A software ISDB cache can be enabled in the FortiOS kernel. This ISDB cache can be used to enhance lookup performance by circumventing the ISDB lookup penalty when revisiting the same resources.

The ISDB cache can be enabled using the following command:

```
config system settings
    set internet-service-database-cache {enable | disable}
end
```

Example

In the following example, after enabling the software ISDB cache, traffic will be generated twice to the same resource. Since the ISDB cache is enabled, no new query will occur in the ISDB. Instead, the ISDB lookup in performed in the cache table.

To enable the software ISDB cache:

1. Enable the ISDB cache:

   ```
   config system settings
       set internet-service-database-cache enable
   end
   ```

2. Create an ISDB firewall policy:

   ```
   config firewall policy
       edit 1
           set internet-service enable
   ```

```
        set internet-service-name "Google-DNS" "Google-Other" "Google-Web"
        set internet-service6 enable
        set internet-service6-name "Google-DNS" "Google-Other" "Google-Web"
    next
end
```

3. Generate traffic to access the resource which matches the ISDB ID in the firewall policy.
4. Check the Internet Service cache lists:

```
# diagnose firewall internet-service-cache list
List Internet Service (IPV4) Cache in Kernel:
MAX_ISDB_CACHE_ENTRY_SIZE=1024 num_isdb_cache_entry=2 isdb_cache_hit_count=0 isdb_query_
count=2
proto=6 port=443 IP=10.151.118.105 id=1245185 country_id=840 region_id=283 city_id=21065
reputation=5 insert_timestamp=4302579542 cache_hit_count=0
proto=6 port=443 IP=10.8.8.8 id=65537 country_id=840 region_id=283 city_id=15905 reputation=5
insert_timestamp=4302579760 cache_hit_count=0

# diagnose firewall internet-service6-cache list
List Internet Service (IPV6) Cache in Kernel:
MAX_ISDB_CACHE_ENTRY_SIZE=1024 num_isdb_cache_entry=1 isdb_cache_hit_count=0 isdb_query_
count=1
proto=6 port=443 IP=2600:140a:1000:196::b33 id=7929993 country_id=124 region_id=65535 city_
id=65535 reputation=4 insert_timestamp=4302580009 cache_hit_count=0
```

5. Generate traffic to access the same resource again.
6. Check the Internet Service cache lists:

```
# diagnose firewall internet-service-cache list
List Internet Service (IPV4) Cache in Kernel:
MAX_ISDB_CACHE_ENTRY_SIZE=1024 num_isdb_cache_entry=2 isdb_cache_hit_count=1 isdb_query_
count=2
proto=6 port=443 IP=10.151.118.105 id=1245185 country_id=840 region_id=283 city_id=21065
reputation=5 insert_timestamp=4302579542 cache_hit_count=0
proto=6 port=443 IP=10.8.8.8 id=65537 country_id=840 region_id=283 city_id=15905 reputation=5
insert_timestamp=4302579760 cache_hit_count=1

# diagnose firewall internet-service6-cache list
List Internet Service (IPV6) Cache in Kernel:
MAX_ISDB_CACHE_ENTRY_SIZE=1024 num_isdb_cache_entry=1 isdb_cache_hit_count=1 isdb_query_
count=1
proto=6 port=443 IP=2600:140a:1000:196::b33 id=7929993 country_id=124 region_id=65535 city_
id=65535 reputation=4 insert_timestamp=4302580009 cache_hit_count=1
```

The ISDB lookup is performed in the cache table so there is no new query in the full ISDB.

ISDB FQDN groups

ISDB Fully Qualified Domain Name (FQDN) groups are groups of well-known services like Amazon-AWS, Cisco-Webex, Google-Gmail, Microsoft-Update, and others that are comprised of various FDQNs that make up the service.

For example, Microsoft-Microsoft.Update includes several FQDNs used for their updating service:

- *.dsp.mp.microsoft.com
- *.windowsupdate.com
- windowsupdate.microsoft.com
- *.update.microsoft.com
- and others

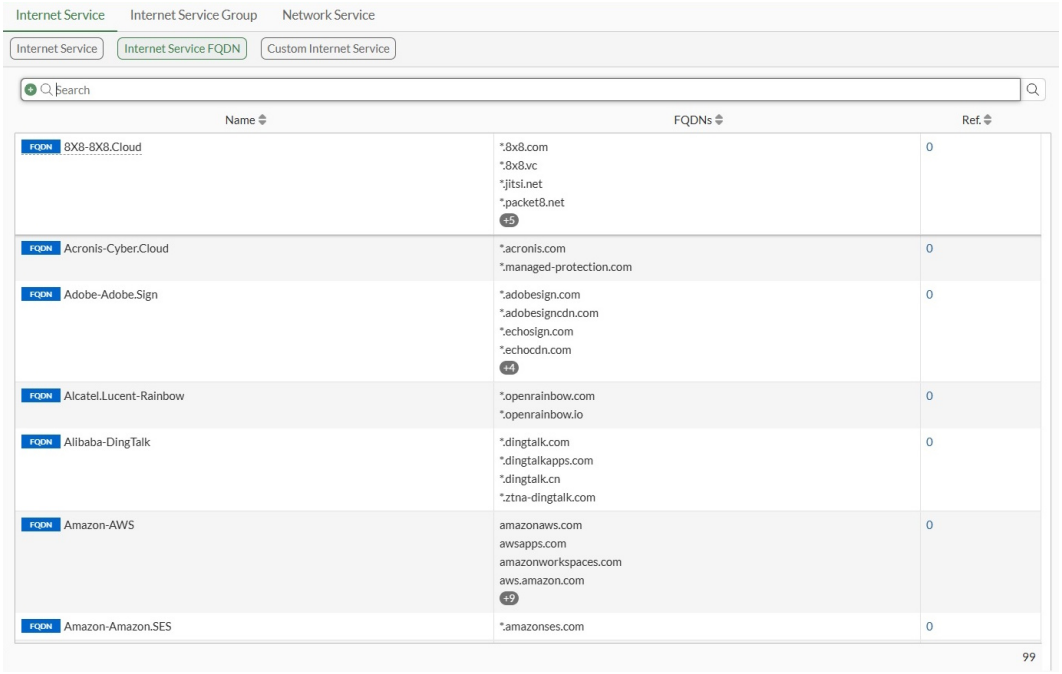
FortiGate can resolve these FDQN addresses to their respective IP addresses and groups.

ISDB FQDN groups allow users to select well-known services as an FQDN group, as opposed to individual ISDB addresses that map directly to IP address and port entries.

ISDB FQDN groups can be used in firewall policies and in security policies when FortiGate is in NGFW policy-based mode.

Applying FQDN address groups within the ISDB in the GUI

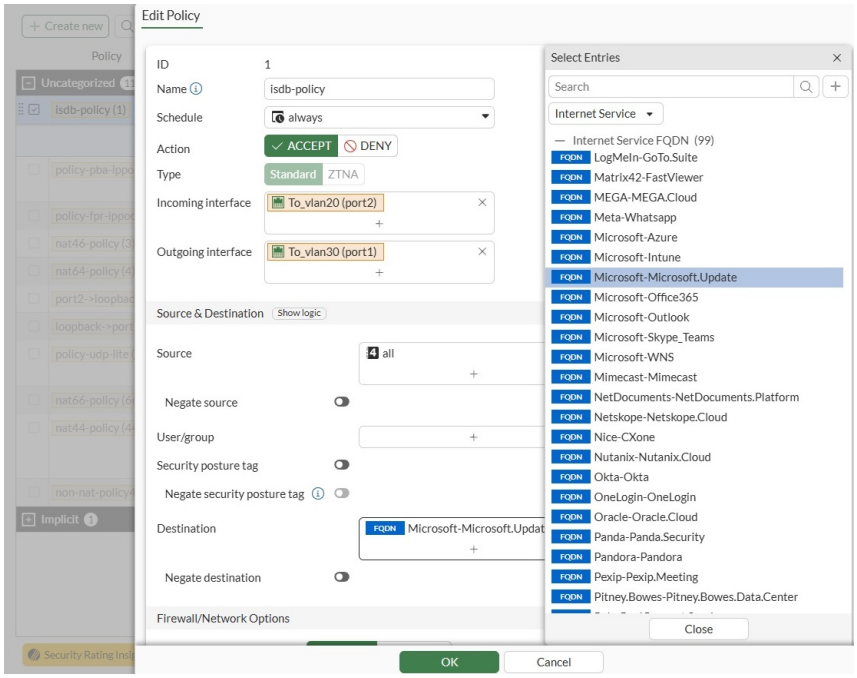
The option to filter ISDB entries by FQDN group is available in the *Policy & Objects > Internet Service Database* page. On the *Internet Service* tab, select *Internet Service FQDN* to view the list of FQDN entries. These address group entries can be applied to a policy in the GUI.



To apply an ISDB FQDN group to a firewall policy in the GUI:

1. Go to *Policy & Objects > Firewall Policy*.
2. Click *Create New* or edit an existing policy.
3. Assign the FQDN address group:
   a. Select the *Source* or *Destination* field. The *Select Entries* pane is displayed.
   b. Select *Internet Service* from the dropdown list.

c. In the *Internet Service FQDN* section, select the FQDN address groups you would like to assign.



d. Click *Close*.

4. Configure the other fields, as needed.
5. Click *OK*.

To apply an ISDB FQDN group to an NGFW policy in the GUI:

1. Ensure policy-based NGFW mode is enabled. See NGFW policy on page 1868.
2. Go to *Policy & Objects* > *Security Policy*.
3. Click *Create New* or edit an existing policy.
4. Assign the FQDN address group:
   a. Select the *Source* or *Destination* field. The *Select Entries* pane is displayed.
   b. Select *Internet Service* from the dropdown list.
   c. In the *Internet Service FQDN* section, select the FQDN address groups you would like to assign.
   d. Click *Close*.
5. Configure the other fields, as needed.
6. Click *OK*.

Applying FQDN address groups within the ISDB in the CLI

FQDN address groups can be applied in the CLI. Likewise, the following `diagnose` commands identify predefined FQDNs:

```
# diagnose ffdb-fqdn group-list [<Built-in group name>]
# diagnose ffdb-fqdn group-get <Built-in FQDN name>
# diagnose ffdb-fqdn icon-get <Built-in FQDN group name>
```

Command	Description
`ffdb-fqdn group-list`	List all of the built-in ISDB FQDN groups. If the specific built-in FQDN name is appended to the end of the command, only that entry's information is displayed. If the specific built-in FQDN name name is not defined, all entries in the current database will be listed.
`ffdb-fqdn group-get`	Determine which built-in FQDN group the given FQDN name belongs to.
`ffdb-fqdn icon-get`	Determine which icon ID the given FQDN group name belongs to.

Example: firewall policy

In the following example, a predefined FQDN Internet service, `g-BuiltIn-Microsoft-Microsoft.Update`, is applied to a firewall policy. We will then review how the customer Internet service is loaded in the kernel and try to send packets to the FortiGate from the client to hit the policy. Finally, we will review the Internet service related fields in the traffic logs.

To apply an ISDB FQDN group to a firewall policy in the CLI:

1. Apply the built-in FQDN entry to the firewall policy:

```
config firewall policy
    edit 1
        set name "isdb-policy"
        set srcintf "port2"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set internet-service enable
        set internet-service-fortiguard "FQDN-Microsoft-Microsoft.Update"
        set schedule "always"
        set logtraffic all
        set auto-asic-offload disable
        set nat enable
    next
end
```

2. Review how the built-in FQDN entry is loaded into the kernel:

```
# diagnose firewall iprope list 100004 | grep -A 16 index=1

policy index=1 uuid_idx=8207 action=accept
flag (8050109): log redir nat master use_src pol_stats
flag2 (6200): no_asic log_fail resolve_sso
flag3 (a0): link-local best-route
flag4 (200): port-preserve
schedule(always)
cos_fwd=255  cos_rev=255
group=00100004 av=00004e20 au=00000000 split=00000000
host=0 chk_client_info=0x0 app_list=0 ips_view=0
misc=0
```

```
zone(1): 8 -> zone(1): 7
source(1): 0.0.0.0-255.255.255.255, uuid_idx=8050,
dest(1): 0.0.0.0-255.255.255.255, uuid_idx=8050,
service(1):
        [0:0x0:0/(0,65535)->(0,65535)] flags:0 helper:auto
internet service(1): FQDN-Microsoft-Microsoft.Update(4278190087,0,0,0)
```

3. Review the FQDN domain name and corresponding IP address information contained in the built-in FQDN entry are listed. The resolution of these domain names comes from the DNS server set by the FortiGate.

```
# diagnose firewall internet-service-fortiguard list
List internet service in kernel(fortiguard):
name=FQDN-Microsoft-Microsoft.Update id=4278190087 reputation=3 Unverified site. singularity=0
flags=0x0 protocol=0
fqdn=(9):
        devicelistenerprod.microsoft.com ID(26) ADDR(4.255.252.37)
        *.download.microsoft.com ID(42)
        *.update.microsoft.com ID(79)
        *.dsp.mp.microsoft.com ID(83)
        ntservicepack.microsoft.com ID(147) ADDR(132.196.74.18)
        windowsupdate.microsoft.com ID(210) ADDR(20.72.235.82)
        wustat.windows.com ID(216)
        *.windowsupdate.com ID(241)
        *.delivery.mp.microsoft.com ID(339)
dynamic=(0):
```

4. Send packets from the client to the FQDN domain `windowsupdate.microsoft.com`, which is one of the FQDN domains included in `FQDN-Microsoft-Microsoft.Update`. The traffic will hit and be forwarded by the firewall policy.

```
1: date=2025-06-16 time=12:30:24 eventtime=1729107023938309279 tz="-0700" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="root" srcip=10.1.100.41
srcport=46900 srcintf="port2" srcintfrole="undefined" dstip=20.72.235.82 dstport=443
dstintf="port1" dstintfrole="undefined" srcuuid="5cfbeb4e-b05e-51ee-233c-5026f6bf6d00"
srccountry="Reserved" dstcountry="United States" dstreputation=3 sessionid=1243 proto=6
action="close" policyid=1 policytype="policy" poluuid="ea3d8560-ea62-51ee-fa88-
20dfc247a38c" policyname="isdb-policy" dstinetsvc="FQDN-Microsoft-Microsoft.Update"
service="FQDN-Microsoft-Microsoft.Update" trandisp="snat" transip=172.16.200.6 transport=46900
appcat="unscanned" duration=2 sentbyte=1623 rcvdbyte=5268
sentpkt=16 rcvdpkt=13
```

Example: NGFW policy

To apply an ISDB FQDN group to an NGFW policy:

1. Ensure policy-based NGFW mode is enabled. See NGFW policy on page 1868.
2. Apply the built-in FQDN entry to a security policy in NGFW policy-based mode:

```
config firewall security-policy
    edit 100
        set uuid 4473436c-9b01-51f0-eea3-44af202bf3a1
```

```
        set srcintf "port3"
        set dstintf "port1"
        set srcaddr "all"
        set internet-service enable
        set internet-service-fortiguard "FQDN-Microsoft-Microsoft.Update"
        set action accept
        set schedule "always"
        set logtraffic all
    next
end
```

3. Send packets from the client to the FQDN domain `windowsupdate.microsoft.com` (IP address `128.85.102.70`), which is one of the FQDN domains included in `FQDN-Microsoft-Microsoft.Update`. The traffic is affected, and FortiGate forwards the traffic according to the security policy.

```
1: date=2025-09-26 time=10:54:25 eventtime=1758909265590012214 tz="-0700" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="vdom1" srcip=10.1.100.11 identifier=987
srcintf="port3" srcintfrole="undefined" dstip=128.85.102.70 dstintf="port1"
dstintfrole="undefined" srccountry="Reserved" dstinetsvc="FQDN-Microsoft-Microsoft.Update"
dstcountry="United States" dstregion="Washington" dstcity="Quincy" sessionid=10391 proto=1
action="accept" policyid=100 policytype="security-policy" poluuid="4473436c-9b01-51f0-eea3-
44af202bf3a1" centralnatid=1 service="FQDN-Microsoft-Microsoft.Update" trandisp="snat"
transip=172.16.200.68 appid=24466 app="Ping" appcat="Network.Service" apprisk="elevated"
duration=119 sentbyte=4956 rcvdbyte=0 sentpkt=59 rcvdpkt=0 srchwvendor="Fortinet"
devtype="Router" srcfamily="FortiGate" osname="FortiOS" srchwversion="200F"
mastersrcmac="d4:76:a0:03:23:38" srcmac="d4:76:a0:03:23:38" srcserver=0 dstdevtype="Unknown"
dstosname="Unknown" masterdstmac="90:6c:ac:c2:75:cb" dstmac="90:6c:ac:c2:75:cb" dstserver=0
```

Security Profiles

This section contains information about configuring FortiGate security features, including:

- Inspection modes on page 2212
- AI protection on page 2224
- Antivirus on page 2240
- Web filter on page 2311
- Video filter on page 2372
- DNS filter on page 2386
- Application control on page 2428
- Inline CASB examples on page 2449
- Intrusion prevention on page 2478
- File filter on page 2518
- Email filter on page 2531
- VoIP solutions on page 2548
- ICAP on page 2582
- Web application firewall on page 2599
- Data loss prevention on page 2602
- Virtual patching on page 2675
- SSL & SSH Inspection on page 2690
- Custom signatures on page 2724
- Overrides on page 2736
- IP ban on page 2751
- Profile groups on page 2757

If you are unable to view a security profile feature, go to *System > Feature Visibility* to enable it.

Some profiles and features are not supported on FortiGate models with 2 GB RAM or less. See Proxy-related features not supported on FortiGate 2 GB RAM models on page 126 and Proxy-based inspection for email protocols supported on models with 2 GB RAM on page 2218 for details.

Inspection modes

FortiOS supports flow-based and proxy-based inspection in firewall policies. You can select the inspection mode when configuring a policy.

Flow-based inspection takes a snapshot of content packets and uses pattern matching to identify security threats in the content.

Proxy-based inspection reconstructs content that passes through the FortiGate and inspects the content for security threats.

Certain security profiles allows users to display flow-based or proxy-based feature sets.

This following topics provide information about inspection modes for various security profile features:

- Flow mode inspection (default mode) on page 2213
- Proxy mode inspection on page 2213
- Inspection mode feature comparison on page 2214

See also Proxy-based inspection for email protocols supported on models with 2 GB RAM on page 2218.

Flow mode inspection (default mode)

When a firewall policy's inspection mode is set to flow, traffic flowing through the policy will not be buffered by the FortiGate. Unlike proxy mode, the content payload passing through the policy will be inspected on a packet by packet basis with the very last packet held by the FortiGate until the scan returns a verdict. If a violation is detected in the traffic, a reset packet is issued to the receiver, which terminates the connection and prevents the payload from being sent successfully.

Flow-based inspection identifies and blocks security threats in real time as they are identified. All applicable flow-based security modules are applied simultaneously in one single pass, using Direct Filter Approach (DFA) pattern matching to identify possible attacks or threats. Pattern matching is offloaded and accelerated by CP8 or CP9 processors.

Flow-based inspection typically requires lower processing resources than proxy-based inspection and does not change packets, unless a threat is found and packets are blocked. Flow-based inspection is selected by default on new firewall policies. It is the recommended inspection mode, unless proxy-specific features are required. For more information, see Inspection mode feature comparison on page 2214.

Proxy mode inspection

When a firewall policy’s inspection mode is set to proxy, traffic flowing through the policy will be buffered by the FortiGate for inspection. This means that the packets for a file, email message, or web page will be held by the FortiGate until the entire payload is inspected for violations (virus, spam, or malicious web links). After FortiOS finishes the inspection, the payload is either released to the destination (if the traffic is clean) or dropped and replaced with a replacement message (if the traffic contains violations).

To optimize inspection, the policy can be configured to block or ignore files or messages that exceed a certain size. To prevent the receiving end user from timing out, you can apply client comforting. This allows small portions of the payload to be sent while it is undergoing inspection.

In proxy-based antivirus scanning, certain techniques are used to streamline scanning with either in-process or stream-based scanning. For more information, see Proxy mode stream-based scanning on page 2259.

Proxy mode provides some security profile capabilities that are not available to flow-based scanning:

- Video Filter
- Inline CASB
- Web Application Firewall (WAF)

- Content Disarm and Reconstruction (CDR)
- Web quota
- Sandbox Inline Scanning

For a complete list, see Inspection mode feature comparison on page 2214.

Some features are exclusively proxy-based:

- SSL Offloading
- Explicit Web Proxy
- ZTNA

Verify the capabilities that you need when deciding to use proxy-based or flow-based policy. Applying the same type of scan mode in all your policies also helps optimize your performance.

FortiOS supports the Zstandard (ZSTD) compression algorithm for web content. FortiOS can use proxy-based policies to decode ZSTD-encoded web content, scan it, and forward the web content to a browser. Then the web content can be passed to the user or blocked from the user based on UTM profile settings, ensuring a seamless and secure browsing experience.

Most proxy features are not supported on FortiGate models with 2 GB RAM or less. See Proxy-related features not supported on FortiGate 2 GB RAM models on page 126 and Proxy-based inspection for email protocols supported on models with 2 GB RAM on page 2218.

Inspection mode feature comparison

The following table shows which UTM profile can be configured on a flow mode or proxy mode inspection policy.

Some UTM profiles are hidden in the GUI and can only be configured using the CLI. To configure profiles in a firewall policy in CLI, enable the `utm-status` setting.

Some profiles might have feature differences between flow-based and proxy-based Inspection. From the GUI and CLI, you can set the *Feature set* option to be *Flow-based* or *Proxy-based* to display only the settings for that mode.

Some profiles and features are not supported on FortiGate models with 2 GB RAM or less. See Proxy-related features not supported on FortiGate 2 GB RAM models on page 126 for the list of models. See also Proxy-based inspection for email protocols supported on models with 2 GB RAM on page 2218.

	Flow Mode Inspection Policy		Proxy Mode Inspection Policy		Feature set option
UTM Profile	GUI	CLI	GUI	CLI	
AntiVirus**	Yes	Yes	Yes	Yes	GUI/CLI
Web Filter**	Yes	Yes	Yes	Yes	GUI/CLI
Video Filter*	No	No	Yes	Yes	N/A
DNS Filter***	Yes	Yes	Yes	Yes	N/A
Application Control	Yes	Yes	Yes	Yes	N/A

UTM Profile	Flow Mode Inspection Policy		Proxy Mode Inspection Policy		Feature set option
	GUI	CLI	GUI	CLI	
Inline CASB*	No	No	Yes	Yes	N/A
Intrusion Prevention System	Yes	Yes	Yes	Yes	N/A
File Filter**	Yes	Yes	Yes	Yes	GUI/CLI
Email Filter**	Yes	Yes	Yes	Yes	GUI/CLI
VoIP	Yes	Yes	Yes	Yes	N/A
ICAP*	No	No	Yes	Yes	N/A
Web Application Firewall*	No	No	Yes	Yes	N/A
Data Loss Prevention**	No	Yes	Yes	Yes	CLI
Virtual Patching	Yes	Yes	Yes	Yes	N/A
SSL/SSH Inspection	Yes	Yes	Yes	Yes	N/A
SSH Filter*	No	No	No	Yes	N/A

* Proxy-only UTM profiles are not supported on FortiGate models with 2 GB RAM or less.

** Feature set option is not available on FortiGate models with 2 GB RAM or less. Profile only supports flow mode.

*** The transparent conditional DNS forwarder feature only works with a proxy-based firewall policy. The feature uses DNS filters with `transparent-dns-database` enabled and is not available on FortiGate models with 2 GB RAM or less.

The following sections outline differences between flow-based and proxy-based inspection for a security profile.

Feature comparison between Antivirus inspection modes

The following table indicates which Antivirus features are supported by their designated scan modes.

Part1	Replacement Message	Content Disarm	Mobile Malware	Virus Outbreak	Sandbox Post-Transfer Scanning	Sandbox Inline Scanning	NAC Quarantine
Proxy (2)	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Flow	Yes (1)	No	Yes	Yes	Yes	Yes	Yes

1. IPS Engine caches the URL and a replacement message is presented after the second attempt.
2. Not available on FortiGate models with 2 GB RAM or less, except when the firewall policy inspects email protocols SMTP(s), POP3(s), IMAP(s), and NNTP.

Part 2	Archive Blocking	Emulator	Client Comforting	Infection Quarantine	Heuristics	Treat EXE as Virus
Proxy (3)	Yes	Yes	Yes	Yes (1)	Yes	Yes (2)
Flow	Yes	Yes	No	Yes (1)	Yes	Yes (2)

1. Only available on FortiGate models with HDD or when FortiAnalyzer or FortiGate Cloud is connected and enabled.
2. Only applies to inspection on IMAP, POP3, SMTP, and MAPI protocols.
3. Not available on FortiGate models with 2 GB RAM or less, except when the firewall policy inspects email protocols SMTP(s), POP3(s), IMAP(s), and NNTP.

Part 3	External Blocklist	EMS External Feed	AI/ML Based Detection	FortiNDR Inline Detection
Proxy (1)	Yes	Yes	Yes	Yes
Flow	Yes	Yes	Yes	No

1. Not available on FortiGate models with 2 GB RAM or less, except when the firewall policy inspects email protocols SMTP(s), POP3(s), IMAP(s), and NNTP.

Feature comparison between Web Filter inspection modes

The following table indicates which Web Filter features are supported by their designated inspection modes.

	FortiGuard Category-Based Filter	Category Usage Quota	Override Blocked Categories	Search Engines	Static URL Filter	Rating Option	Proxy Option	Web Profile Override
Proxy (4)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Flow	Yes (1)	No	Yes (2)	Yes	Yes (5)	Yes	Limited (3)	No

1. Local Category and Remote Category filters do not support the warning and authenticate actions.
2. Local Category and Remote Category filters cannot be overridden.
3. Only HTTP POST Action and Remove Cookies are supported.
4. Not available on FortiGate models with 2 GB RAM or less.
5. File filter is not supported. See Configuring web filter to exempt URLs from other security profiles on page 2369.

Feature comparison between Email Filter inspection modes

The following tables indicate which Email Filters are supported by the specified inspection modes for local filtering and FortiGuard-assisted filtering.

Local Filtering	Banned Word Check	Block/Allow List	HELO/ EHLO DNS Check	Return Address DNS Check	DNSBL/ ORBL Check	MIME Header Check
Proxy	Yes	Yes	Yes	Yes	Yes	Yes
Flow	Yes	Yes	No	No	No	Yes

FortiGuard-Assisted Filtering	Phishing URL Check	Anti-Spam IP Check	Submit Spam to FortiGuard	Spam Email Checksum Check	Spam URL Check
Proxy	Yes	Yes	Yes	Yes	Yes
Flow	No	No	No	No	No

Feature comparison between DLP inspection modes

The following table indicates which DLP filters are supported by their designated inspection modes.

	Credit Card Filter	SSN Filter	Regex Filter	File-Type Filter	File-Pattern Filter	Fingerprint Filter	Watermark Filter	Encrypted Filter	File-Size Filter
Proxy (2)	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Flow	Yes	Yes	Yes	Yes	Yes	No	No	Yes	Yes (1)

1. File-size filtering only works if file size is present in the protocol exchange.
2. Not available on FortiGate models with 2 GB RAM or less, except when the firewall policy inspects email protocols SMTP(s), POP3(s), IMAP(s), and NNTP.

Inspection of SMBv3 multichannel is not supported. To inspect SMBv3 traffic, it is advisable to disable multichannel support first. See the vendor specific documentation for more information on disabling multichannel support.

Proxy-based inspection for email protocols supported on models with 2 GB RAM

Previously in FortiOS 7.4.4, proxy-related features were disabled on FortiGate models with 2 GB RAM or less. See Proxy-related features no longer supported on FortiGate 2 GB RAM models for more information.

FortiOS 7.6.5 and later supports proxy-based inspection for email protocols on FortiGate models with 2 GB RAM for the following services:

- SMTP(s)
- POP3(s)
- IMAP(s)
- NNTP

Firewall policies support proxy-based inspection mode when you select one or more of the supported services in a firewall policy.

When configuring a firewall policy in the GUI, a warning is displayed when the selected services, protocol options, and SSL inspection profile contain non-mail protocols. Therefore, at least one of the services, protocol options, and SSL inspection profile must contain strictly mail protocols.

Even if you select a non-mail protocol as a service and FortiOS allows the configuration, traffic hitting the policy for the non-mail protocol is dropped. Therefore, it is recommended to define only the mail protocols that you want to inspect in the proxy-based inspection policy under the services field.

Furthermore, the following proxy-based UTM profiles are not supported in either proxy-based or flow-based policies:

- casb-profile
- waf-profile
- videofilter-profile
- icap-profile
- ssh-filter-profile

The following profiles are not supported in proxy-based policies, but are supported in flow-based policies:

- dnsfilter-profile
- webfilter-profile
- voip-profile

Other UTM profiles, such as AV, IPS, application control, file filter, DLP, and so on have no restrictions.

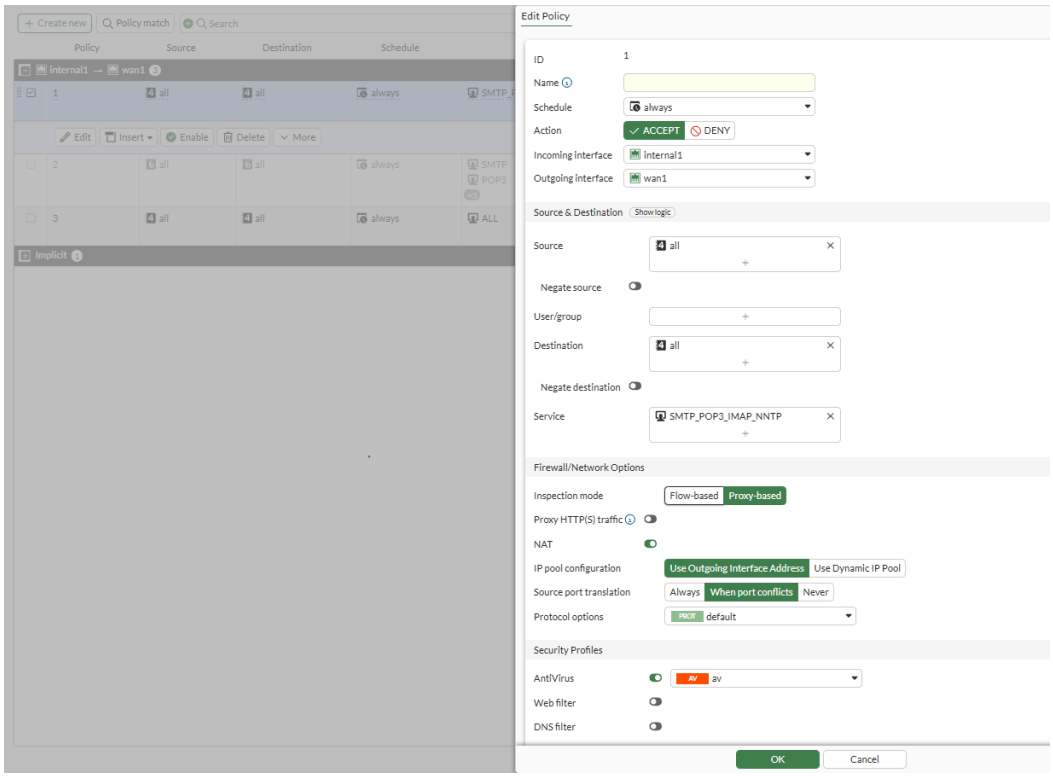
Finally, if `inspect-all` is enabled in an SSL inspection profile, then the profile cannot be used in a proxy-based firewall policy.

These changes impact the FortiGate/FortiWiFi 40F, 60E, 60F, 80E, and 90E series of devices and their variants and FortiGate-Rugged 60F (2 GB versions only).

The following screen shot shows a firewall policy correctly configured with email/NNTP services, proxy-based inspection mode, and a proxy UTM profile:

- *Service* is set to *SMTP_POP3_IMAP_NNTP*.
- *Inspection mode* is set to *Proxy-based*.
- *Protocol Options* is set to the *default* profile.
- Under *Security Profiles*, *AntiVirus* is enabled, and the *av* profile is selected.
- *SSL Inspection* is set to the *deep-inspection* profile.

Conclusion: Because the *Service* defined contains only mail protocols, this configuration is accepted.

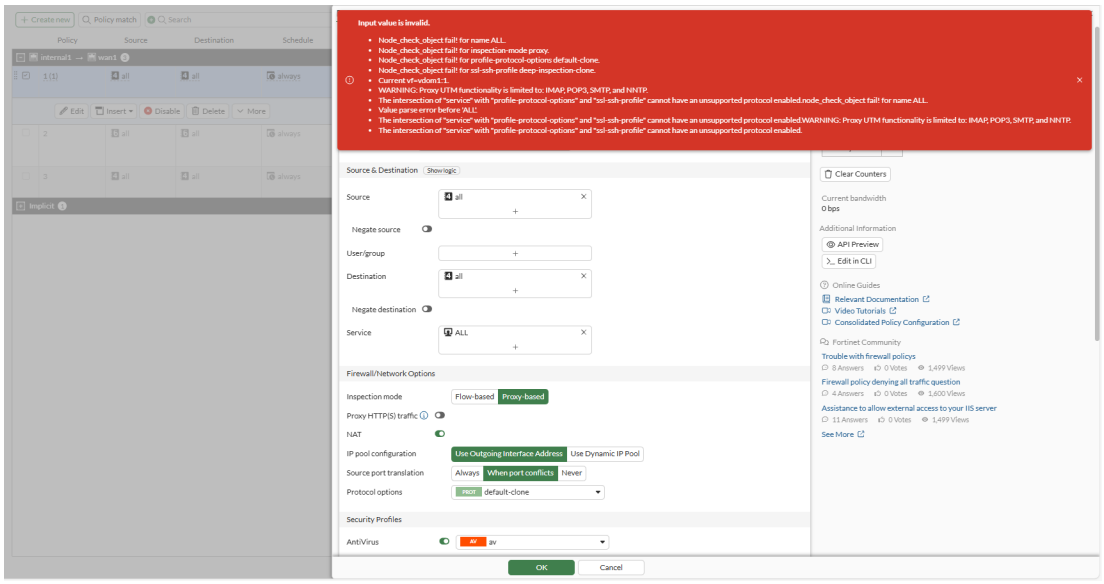


The following GUI example shows the warning message that displays with an unsupported firewall policy configuration:

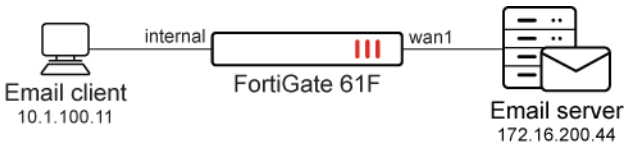
- *Service* is set to *ALL*.

  The *ALL* setting includes ports, such as 80 and 443, that are used for protocols other than the supported email protocols of SMTP, POP3, IMAP, and NNTP.
- *Inspection mode* is set to *Proxy-based*.
- *Protocol Options* is set to the *default-clone* profile, which inspects HTTP.
- Under *Security Profiles*, *AntiVirus* is enabled, and the *av* profile is selected.
- *SSL Inspection* is set to the *deep-inspection-clone* profile, which inspects HTTP.

Conclusion: Between the *Service* selected, the *Protocol Options*, and *SSL Inspection* profile, all three include inspection on non-mail protocols.



Example



In this example, a firewall policy is configured to scan email traffic sent from an email client through FortiGate to an email server. The firewall policy is configured with a proxy-based inspection mode and mail services and profiles. The FortiGate 61F has less than 2 GB RAM.

To configure a firewall policy:

1. Configure the service to include mail protocols in the TCP port range:

```
config firewall service custom
    edit "SMTP_POP3_IMAP_NNTP"
        set uuid 8fa0520e-afcd-51f0-ecf3-f5bc0e446b3b
        set category "General"
        set tcp-portrange 25 110 143 119 465 995 993
    next
end
```

2. Configure the protocol profile:

```
config firewall profile-protocol-options
    edit "default-clone"
        set comment "All default services."
        config http
            set ports 80 8080
            unset options
```

```
            unset post-lang
        end
        config ftp
            set ports 21
            set options splice
        end
        config imap
            set ports 143
            set options fragmail
        end
        config pop3
            set ports 110
            set options fragmail
        end
        config smtp
            set ports 25
            set options fragmail splice
        end
        config nntp
            set ports 119
            set options splice
        end
        config dns
            set ports 53
        end
        config cifs
            set ports 445
            unset options
        end
    next
end
```

3. Configure an SSL inspection profile:
   This example uses the default `deep-inspection-clone` profile.

```
config firewall ssl-ssh-profile
    edit "deep-inspection-clone"
        set comment "Read-only deep inspection profile."
        config https
            set ports 443
            set status deep-inspection
        end
        config ftps
            set ports 990
            set status deep-inspection
        end
        config imaps
            set ports 993
            set status deep-inspection
        end
        config pop3s
            set ports 995
```

```
            set status deep-inspection
        end
        config smtps
            set ports 465
            set status deep-inspection
        end
        config dot
            set status disable
        end
        set ssl-exemption-log enable
        set ssl-server-cert-log enable
        set ssl-handshake-log enable
    next
end
```

4. Configure an antivirus profile with proxy-only feature and content-disarm:

```
config antivirus profile
    edit "av"
        set feature-set proxy
        config http
            set av-scan block
        end
        config ftp
            set av-scan block
        end
        config imap
            set av-scan block
        end
        config pop3
            set av-scan block
        end
        config smtp
            set av-scan block
        end
        config content-disarm
            set analytics-suspicious disable
        end
    next
end
```

5. Configure the firewall policy:

```
config firewall policy
    edit 1
        set name "1"
        set uuid a85736ea-79dd-51ef-103d-0791ae4d6152
        set srcintf "internal"
        set dstintf "wan1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
```

```
        set schedule "always"
        set service "SMTP_POP3_IMAP_NNTP"
        set utm-status enable
        set inspection-mode proxy
        set profile-protocol-options "default-clone"
        set ssl-ssh-profile "deep-inspection-clone"
        set av-profile "av"
        set logtraffic all
        set nat enable
    next
end
```

When email traffic of SMTP(s), POP3(s), or IMAP(s) contains a virus and is sent through the firewall policy, the antivirus profile blocks the traffic. Following are examples of the generated utm-av logs for SMTP(s), POP3(s), and IMAP(s):

- SMTP(s):

```
3: date=2025-10-23 time=12:21:34 eventtime=1761178894206730660 tz="+1200" logid="0211008194"
type="utm" subtype="virus" eventtype="infected" level="warning" vd="vdom1" policyid=1
poluuid="a85736ea-79dd-51ef-103d-0791ae4d6152" policytype="policy" msg="MIME data detected to
have a virus and blocked by Antivirus." action="blocked" service="SMTPS" sessionid=849
srcip=10.1.100.11 dstip=172.16.200.44 srcport=52746 dstport=465 srccountry="Reserved"
dstcountry="Reserved" srcintf="internal" srcintfrole="lan" dstintf="wan1" dstintfrole="wan"
srcuuid="427663be-79dd-51ef-b8ee-20c97371a015" dstuuid="427663be-79dd-51ef-b8ee-20c97371a015"
proto=6 direction="outgoing" filename="eicar.com" quarskip="Quarantine-disabled" virus="EICAR_
TEST_FILE" viruscat="Virus" dtype="av-engine" itype="infected"
ref="https://fortiguard.com/encyclopedia/virus/2172" virusid=2172 profile="av"
from="pc1smtp@kunal.com" to="pc4@kunal.com" sender="pc4user2@qa.fortinet.com"
recipient="pc4user1@qa.fortinet.com" messageid="<46095C0D.5000203@kunal.com>"
subject="attachment with eicar.com" attachment="yes"
analyticscksum="275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f"
analyticssubmit="false" crscore=50 craction=2 crlevel="critical"
```

- POP3(s):

```
2: date=2025-10-23 time=12:21:48 eventtime=1761178908394447280 tz="+1200" logid="0211008194"
type="utm" subtype="virus" eventtype="infected" level="warning" vd="vdom1" policyid=1
poluuid="a85736ea-79dd-51ef-103d-0791ae4d6152" policytype="policy" msg="MIME data detected to
have a virus and blocked by Antivirus." action="attachment-removed" service="POP3S"
sessionid=850 srcip=10.1.100.11 dstip=172.16.200.44 srcport=40020 dstport=995
srccountry="Reserved" dstcountry="Reserved" srcintf="internal" srcintfrole="lan"
dstintf="wan1" dstintfrole="wan" srcuuid="427663be-79dd-51ef-b8ee-20c97371a015"
dstuuid="427663be-79dd-51ef-b8ee-20c97371a015" proto=6 direction="incoming"
filename="eicar.com" quarskip="Quarantine-disabled" virus="EICAR_TEST_FILE" viruscat="Virus"
dtype="av-engine" itype="infected" ref="https://fortiguard.com/encyclopedia/virus/2172"
virusid=2172 profile="av" from="test1@qa.fortinet.com" to="test1@qa.fortinet.com"
recipient="pc4user2" messageid="<1426193044.8910.11.camel@nan-desktop>" subject="eicar file"
attachment="yes"
analyticscksum="275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f"
analyticssubmit="false" crscore=50 craction=2 crlevel="critical"
```

- IMAP(s):

```
1: date=2025-10-23 time=12:22:04 eventtime=1761178923814916760 tz="+1200" logid="0211008194"
type="utm" subtype="virus" eventtype="infected" level="warning" vd="vdom1" policyid=1
poluuid="a85736ea-79dd-51ef-103d-0791ae4d6152" policytype="policy" msg="MIME data detected to
have a virus and blocked by Antivirus." action="attachment-removed" service="IMAPS"
sessionid=854 srcip=10.1.100.11 dstip=172.16.200.44 srcport=45458 dstport=993
srccountry="Reserved" dstcountry="Reserved" srcintf="internal" srcintfrole="lan"
dstintf="wan1" dstintfrole="wan" srcuuid="427663be-79dd-51ef-b8ee-20c97371a015"
dstuuid="427663be-79dd-51ef-b8ee-20c97371a015" proto=6 direction="incoming"
filename="eicar.com" quarskip="Quarantine-disabled" virus="EICAR_TEST_FILE" viruscat="Virus"
dtype="av-engine" itype="infected" ref="https://fortiguard.com/encyclopedia/virus/2172"
virusid=2172 profile="av" from="test1@qa.fortinet.com" to="test1@qa.fortinet.com"
recipient="pc4user2" messageid="<1426193044.8910.11.camel@nan-desktop>" subject="eicar file"
attachment="yes"
analyticscksum="275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f"
analyticssubmit="false" crscore=50 craction=2 crlevel="critical"
```

AI protection

Artificial Intelligence (AI) is rapidly transforming business operations, cybersecurity, and digital infrastructure. Among its most prominent forms are Large Language Models (LLMs), such as GPT, which are a type of Generative AI (GenAI) widely used in chatbot applications, such as ChatGPT, DeepSeek, and others. As these chatbots become more common in corporate environments, new risks emerge. AI applications, including but not limited to chatbots, can inadvertently expose sensitive data. For instance, an employee might paste a confidential email into a chatbot to generate a customized reply, potentially exposing sensitive information.

Securing GenAI has become critical in this evolving landscape. FortiGate offers several methods to restrict or block access to GenAI tools, helping prevent data leaks and protect company information.

The following table outlines the techniques and highlights scenarios where each method may be most effective.

Technique	Description
Web filter	Using a web filter profile with the FortiGuard Artificial Intelligence Technology category enables users to monitor and block access to sites in this category by matching the requested URL against FortiGuard's classification. See Protecting GenAI access using web filter on page 2225 for more information.
Application control	Using an application control profile with the Generative AI category enables users to monitor and block access to sites in this category by matching traffic patterns against FortiGuard's Generative AI signatures. This approach goes beyond URL matching by employing deep packet inspection to identify and control application-level traffic, offering enhanced visibility and insight into AI-related activities. See Protecting GenAI access using application control on page 2227 for more information.

Technique	Description
Data Loss Prevention (DLP)	Using a DLP profile, administrators can block access to LLM applications when specific keywords are detected in user prompts. This technique uses keyword matching and FQDN filtering to prevent sensitive or restricted data from being sent to external AI services. This is particularly useful for enforcing compliance and protecting intellectual property. See Protecting GenAI Access using DLP on page 2232 for more information.

Protecting GenAI access using web filter

By leveraging FortiGuard's Artificial Intelligence Technology category within a web filter profile, organizations can proactively restrict access to sites offering AI-related solutions, insights, and resources. This capability is particularly valuable for enterprises aiming to mitigate risks associated with unauthorized AI tool usage.

To confirm that the URL of the LLM application that you need to block is in the Artificial Intelligence Technology category, the URL category can be verified using the FortiGuard Web Filter Lookup.

The `Artificial intelligence technology` FortiGuard category has been added as category `100` in the CLI.

This topic includes examples that incorporate several web filter use cases:

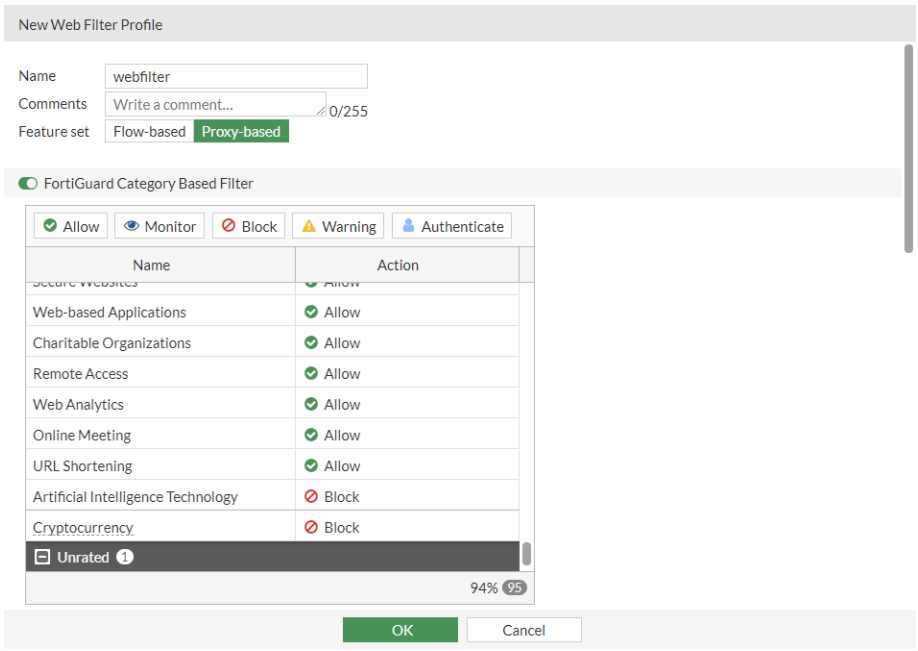
- Example 1: block access to Artificial Intelligence Technology category on page 2225

Example 1: block access to Artificial Intelligence Technology category

In this example, access to all websites classified under the Artificial Intelligence Technology category is blocked using a web filter profile

To configure a web filter profile to block the AI category in the GUI:

1. Go to *Security Profiles* > *Web Filter* and click *Create New*.
2. Enter a name for the web filter profile.
3. In the category table, locate the *General Interest - Business* section. Select the *Artificial Intelligence Technology*, and set the *Action* to *Block*.



4. Configure the remaining settings as needed.
5. Click *OK*.

To configure a web filter profile to block the AI category in the CLI:

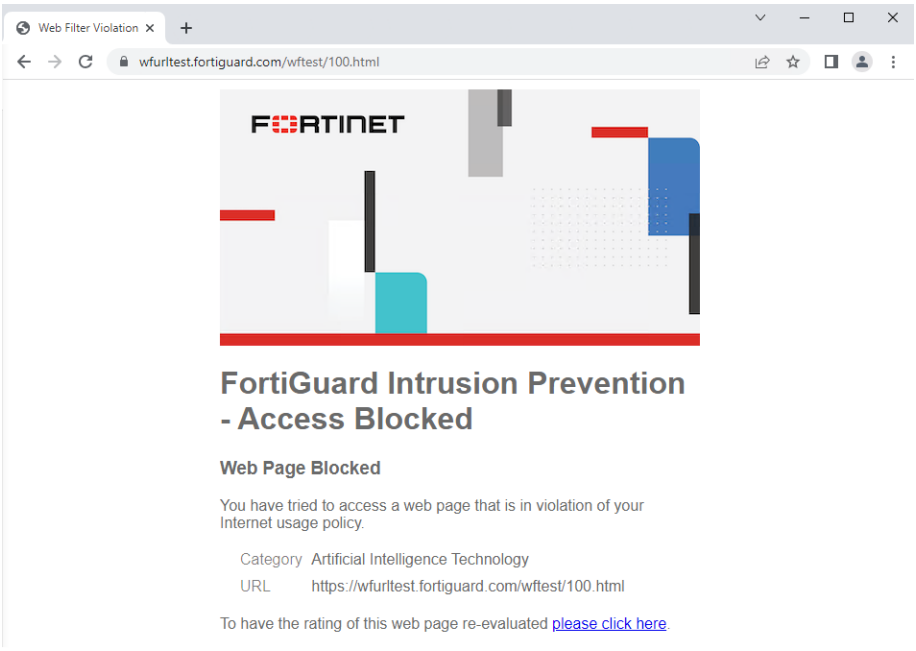
```
config webfilter profile
    edit "webfilter"
        set feature-set proxy
        config ftgd-wf
            unset options
            config filters
                edit 100
                    set category 100
                    set action block
                next
            end
        end
        set log-all-url enable
    next
end
```

To verify that the categories are blocked:

1. Apply the web filter profile in a firewall policy.
2. On a device that is connected through the FortiGate and uses the policy, visit the following test URL for the category:
   a. `https://wfurltest.fortiguard.com/wftest/100.html`

      The browser displays a replacement message that the URL is blocked based on the FortiGuard category.

- Artificial intelligence technology:



To verify the web filter logs:

1. In the GUI, go to *Log & Report* > *Security Events* and click *Web Filter*.
2. In the CLI, enter the following:

```
2: date=2023-07-12 time=10:39:13 eventtime=1689183553021358734 tz="-0700" logid="0316013056"
type="utm" subtype="webfilter" eventtype="ftgd_blk" level="warning" vd="vdom1" policyid=1
poluuid="996b0a68-2055-51ee-b841-2b3f373c9b37" policytype="policy" sessionid=3255
srcip=10.1.100.31 srcport=35102 srccountry="Reserved" srcintf="port2" srcintfrole="undefined"
srcuuid="124f368a-2055-51ee-c7d6-857ab36dd6cb" dstip=154.52.5.202 dstport=443
dstcountry="United States" dstintf="port1" dstintfrole="undefined" dstuuid="124f368a-2055-
51ee-c7d6-857ab36dd6cb" proto=6 httpmethod="GET" service="HTTPS"
hostname="wfurltest.fortiguard.com" agent="curl/7.68.0" profile="webfilter" action="blocked"
reqtype="direct" url="https://wfurltest.fortiguard.com/wftest/100.html" sentbyte=849
rcvdbyte=3633 direction="outgoing" msg="URL belongs to a denied category in policy"
ratemethod="domain" cat=100 catdesc="Artificial Intelligence Technology"
```

Protecting GenAI access using application control

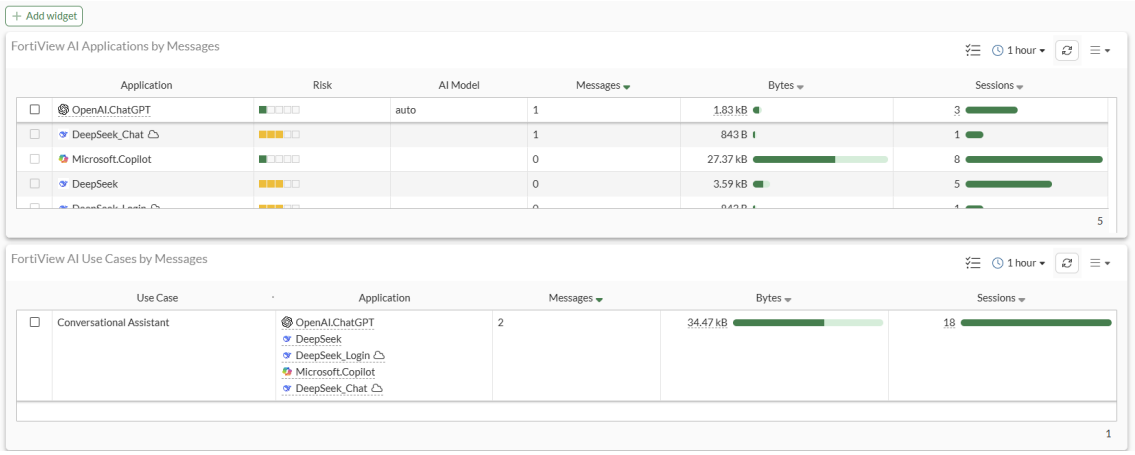
By leveraging FortiGuard's GenAI application category within security profiles, organizations can monitor and manage AI-related activities, such as ChatGPT usage, ensuring compliance and reducing risk.

This technique offers visibility and insight to AI-related activities with deep packet inspection enabled. Following is a table of log fields available in the CLI and GUI for generative AI monitoring:

CLI log field	GUI log field
`aiuser`	AI User
`model`	Model
`dcgeo`	Data Center's Geographical Location
`usecase`	Use Case
`prompt`	Prompt
`cloudgenai`	Generate AI Application

The following FortiView widgets are available to support real-time monitoring of AI traffic:

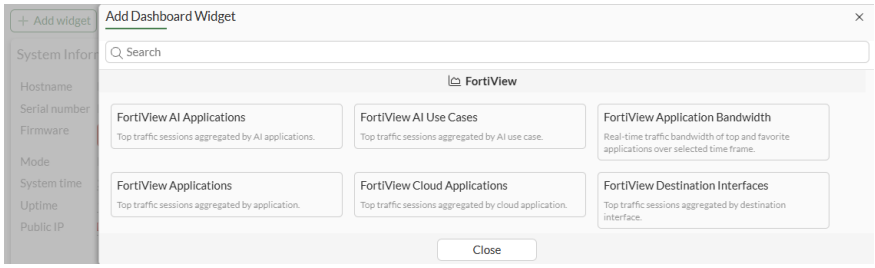
- *AI Applications*: Displays top traffic sessions aggregated by AI application, including details such as Risk, AI model, Sessions, and more. Selecting Drill Down on an application reveals specific insights like Users, Use Cases, and Data Center Locations.
- *AI Use Cases*: Categorizes AI activity based on business or operational context, showing top traffic sessions by use case along with related Application, Sessions, and other metrics. Drill Down provides further details such as Users, Applications, and Data Center Locations.



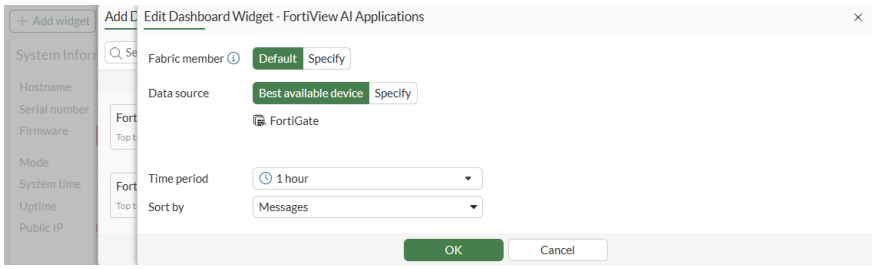
These widgets provide security teams with actionable insights and help enforce governance over AI usage within the network.

To add the FortiView AI widgets to the Dashboard:

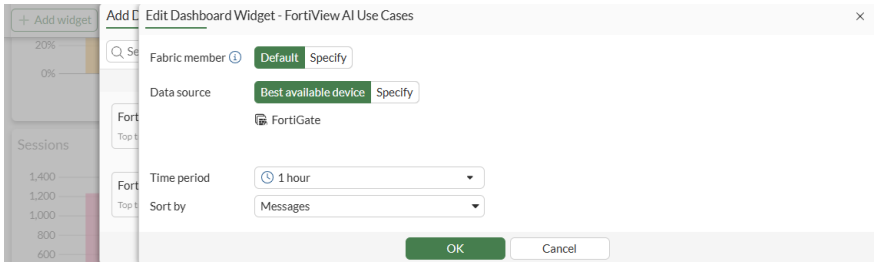
1. Go to *Dashboard* > *Status*.
2. Click *Add Widget*.



3. Select *FortiView AI Applications*.



4. Configure the widget.
5. Click *OK*. The widget will be added to the *Dashboard*.
6. Repeat the configuration steps for *FortiView AI Use Cases*.



7. Click *OK*. The widget will be added to the *Dashboard*.
8. Click *Close*.

To review the generative AI database:

```
# diagnose autoupdate versions | grep -A 6 GenAI
GenAI Application Definitions
---------
Version: 33.00033 signed
Contract Expiry Date: Thu Jan  3 2030
Last Updated using scheduled update on Tue Jun 24 20:59:43 2025
Last Update Attempt: Tue Jun 24 23:10:03 2025
Result: No Updates
```

If you see `Version: 0.00000`, this implies that the AIAP database is not updated yet. For AIAP database to be updated, ensure that at least one firewall policy is utilizing an application control profile. Database updates require FMWR contract.

The `GenAI` application category has been added as `category 36` in the CLI.

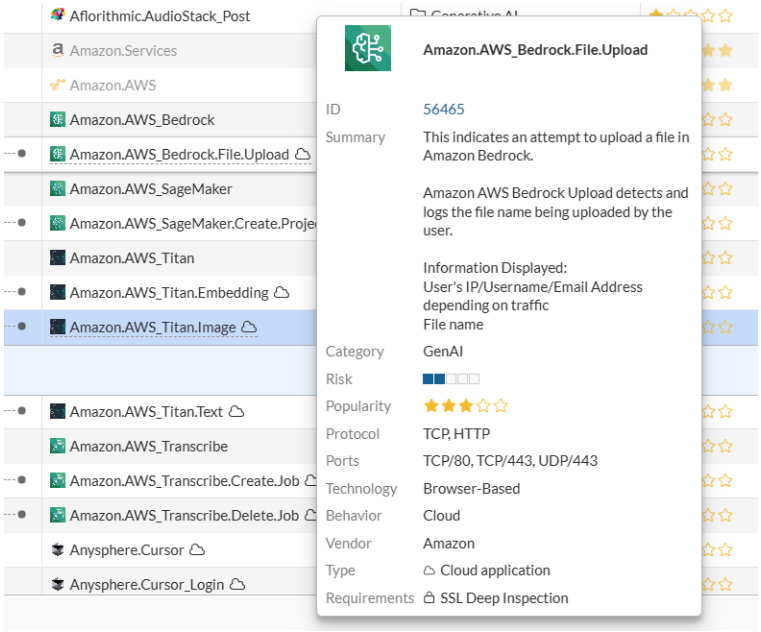
Limitations and advanced configurations

Keep the following limitations and advanced configurations in mind when using Generative AI application control:

- To work with proxy policies while using explicit proxy, for example, inline IPS must be disabled:

```
config ips settings
    set proxy-inline-ips disable
end
```

- Some signatures require deep inspection and some do not.
  To identify which signatures require deep inspection:
  a. Go to *Security Profiles* > *Application Signatures*, and locate the *Generative AI* signature.
  b. Hover over the signature to display the popup.



  c. Locate the *Requirements* field to see whether *SSL Deep Inspection* is required.
- Extended UTM log should be enabled to capture extra information in the log, such as model, message, GenAI user, DC location, and so on. This requires deep inspection.

This topic includes examples that incorporate several application control use cases:

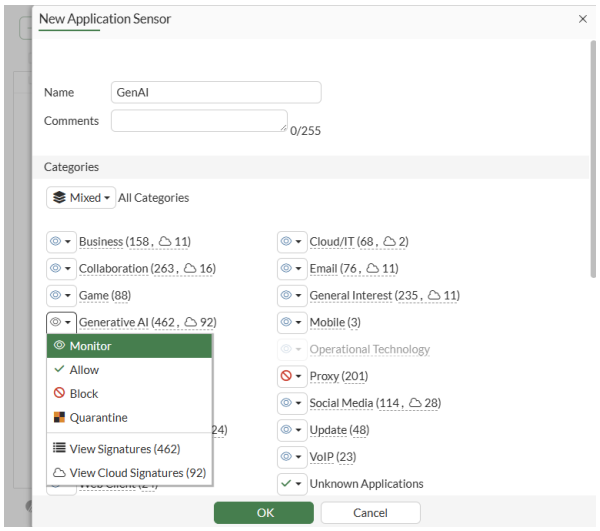
- Example 1: monitor access to Generative AI category on page 2230

Example 1: monitor access to Generative AI category

In this example, access to all signatures classified under the Generative AI category are monitored using an application control profile.

To create an application control sensor to monitor generative AI in the GUI:

1. Monitor generative AI in an application control sensor:
   a. Go to *Security Profiles* > *Application Control*.
   b. Click *Create New*.
   c. Enter a *Name*.
   d. Set the *Generative AI* category to *Monitor*.



e. Configure other fields, as needed.

f. Click *OK*.

2. Apply the sensor to a firewall policy with deep inspection:

a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.

b. In the *Security Profiles* section, enable *Application control* and select the profile that you previously created. In this example, the GenAI profile is selected.

c. Set *SSL Inspection* to a profile where deep-inspection is enabled. In this example *new-deep-inspection* is selected.

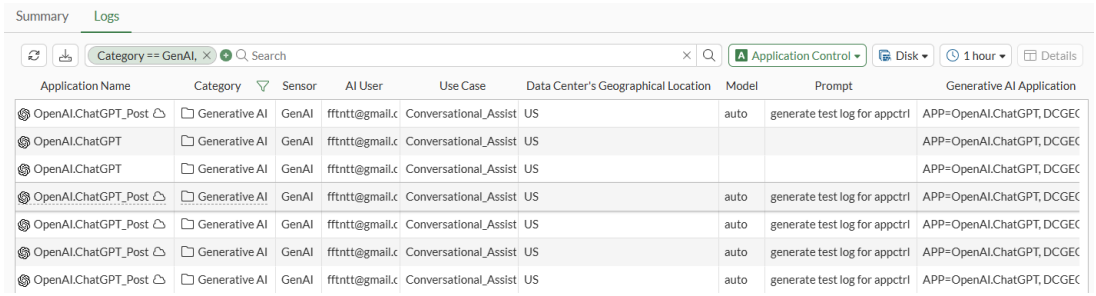
d. Configure the other settings as needed.

e. Click *OK*.

3. On a Client Windows PC, log in to www.chatgpt.com and enter a test prompt.

4. Review the application control UTM log for OpenAI.ChatGPT_Post:

Logs are listed in *Log & Report* > *Security Events*.



To implement generative AI application control in the CLI:

1. Monitor generative AI in an application control sensor:

```
config application list
    edit "GenAI"
        config entries
            edit 1
                set category 36
```

```
                set action pass
            next
        end
    next
end
```

2. Apply the sensor to a firewall policy with deep inspection:

```
config firewall policy
    edit 1
        set name "1"
        set srcintf "port2"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "new-deep-inspection"
        set application-list "GenAI"
        set nat enable
    next
end
```

3. On a Client Windows PC, log into www.chatgpt.com and enter a test prompt.
4. Review the application control UTM log for OpenAI.ChatGPT_Post:

```
# execute log display
1: date=2025-06-25 time=00:13:07 eventtime=1750810387445006017 logid="1059028704" type="utm"
subtype="app-ctrl" eventtype="signature" level="information" vd="vd1" appid=53323
srcip=10.1.100.126 srccountry="Reserved" dstip=110.18.32.47 dstcountry="United States"
srcport=59001 dstport=443 srcintf="port2" srcintfrole="undefined" dstintf="port1"
dstintfrole="undefined" proto=6 service="HTTPS" direction="incoming" policyid=1
policytype="policy" sessionid=1299 applist="GenAI" action="pass" appcat="GenAI"
app="OpenAI.ChatGPT_Post" hostname="110.18.32.47" incidentserialno=61866712 url="/"
httpmethod="POST" msg="GenAI: OpenAI.ChatGPT_Post" usecase="Conversational_Assistant"
aiuser="fftntt@gmail.com" model="auto" dcgeo="US" prompt="generate test log for appctrl"
cloudgenai="APP=OpenAI.ChatGPT, DCGEO=US, UseCase=Conversational_Assistant,
User=fftntt@gmail.com, UserOrganization=org-F2ZasxjlplRvKPgeVfbXbPeM, HistoryTraining=true,
Model=auto, Title='AppCtrl Test Log', Prompt='generate test log for appctrl'" apprisk="low"
```

Protecting GenAI Access using DLP

The FortiGate Data Loss Prevention (DLP) system helps safeguard your network by inspecting traffic for sensitive data patterns. It scans both inbound and outbound traffic, and based on predefined rules, it can block, log, allow, or quarantine data that matches these patterns.

To prevent the upload of sensitive information to Large Language Model (LLM) applications over HTTPS, FortiGate can block traffic containing specific keywords using the predefined "keyword" data type in the DLP dictionary.

Since web-based chatbot implementations are highly dynamic and can vary significantly, it is recommended to configure both message-based and file-based DLP rules to improve detection accuracy. While this approach enhances protection, it may not be completely foolproof.

For stronger security, it is advisable to restrict access to LLM applications entirely. This can be achieved by:

- Applying a web filter profile with the FortiGuard Artificial Intelligence Technology category set to block, or
- Using an application control profile with the Generative AI category

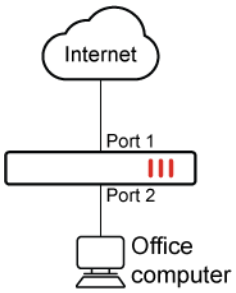
This topic includes examples that incorporate several DLP use cases:

- Example 1: prevent sensitive LLM searches in the browser on page 2233

Example 1: prevent sensitive LLM searches in the browser

In this example, a user is conducting a search on an LLM application in the Chrome browser on an office computer, using a sensitive keyword that has been configured in the DLP dictionary. The FortiGate intercepts this traffic using deep inspection and prevents the search that contains sensitive keywords because it matches the DLP profile that has been set up on this FortiGate.

When a sensitive keyword is included in HTTPS upload traffic, the request is blocked and a DLP log is generated.



To block HTTPS upload traffic that includes sensitive keywords in the GUI:

1. Configure the DLP dictionary:
   a. Go to *Security Profiles* > *Data Loss Prevention*, select the *Dictionary* tab, then click *Create New*.
   b. Set *Name* to *llmapps*.
   c. In the *Dictionary Entries* table click *Create New*:
      i. Set *Type* to *keyword*.
      ii. Set *Pattern* to *fortinet*.
      iii. Enable *Case sensitive*.
      iv. Click *OK*.
   d. Repeat step c and set *Pattern* to *source code*.
   e. Click *OK*.
2. Configure the DLP sensor:
   a. Go to *Security Profiles* > *Data Loss Prevention*, select the *Sensor* tab, then click *Create New*.
   b. Set *Name* to *llmapps*.
   c. In the *Sensor Entries* section, click *Add*. The *Select Entries* pane is displayed.
   d. Select *Managed Locally* from the dropdown menu.

e. Select *llmapps* and click *Apply*.
f. Click *OK*.

3. Configure the DLP profile:
   a. Go to *Security Profiles > Data Loss Prevention*, select the *Profile* tab, then click *Create New*.
   b. Set *Name* to *llmapps*.
   c. In the *Rules* section, click *Create New*:
      i. Configure the following settings:

Name	llmapps1
Sensors	llmapps
Severity	Critical
Action	Block
Type	File
File type	builtin-patterns
Protocol	HTTP-POST

      ii. Click *OK*.
   d. In the *Rules* section, click *Create New* again:
      i. Configure the following settings:

Name	llmapps2
Sensors	llmapps
Severity	Critical
Action	Block
Type	Message
Protocol	HTTP-POST

      ii. Click *OK*.
   e. Click *OK* to save the profile.
   f. Unset the file type option to enable filtering of all file types, including unknown ones:

```
config dlp profile
    edit "llmapps"
        config rule
            edit 1
                unset file-type
            next
        end
    next
end
```

4. Configure the firewall address for the LLM application:
   a. Go to *Policy & Objects > Addresses*, select the *Standard* tab, then click *Create New*.
   b. Set the following:

Name	<name>
Type	FQDN
FQDN	See FQDN on page 2237 for the FQDN on the specific LLM application.

c. Click *OK*.

5. Add the firewall address to a group:
   a. Go to *Policy & Objects* > *Addresses*, select the *Address Group* tab, then click *Create New*.
   b. Set the following:

Group name	llmapps
Members	ChatGpt, Gemini, DeepSeek

6. Add the DLP profile and the address group to a firewall policy:
   a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
   b. Set the *Inspection Mode* to *Proxy-based*.
   c. Set the *Destination* to *llmapps*.
   d. In the *Security Profiles* section:
      i. Enable *Application control* and select the profile that *QUIC* is blocked in. In this example, the *default* profile is selected.
      ii. Enable *DLP Profile* and select *llmapps*.
   e. Set *SSL Inspection* to *deep-inspection*.
   f. Configure the other settings as needed.
   g. Click *OK*.

To block HTTPS upload traffic that includes sensitive keywords in the CLI:

1. Configure the DLP dictionary:

```
config dlp dictionary
    edit "llmapps"
        config entries
            edit 1
                set type "keyword"
                set pattern "source code"
                set ignore-case enable
            next
            edit 2
                set type "keyword"
                set pattern "fortinet"
                set ignore-case enable
            next
        end
    next
end
```

2. Configure the DLP sensor:

```
config dlp sensor
    edit "llmapps"
        config entries
            edit 1
                set dictionary "llmapps"
            next
        end
    next
end
```

3. Configure the DLP profile:

```
config dlp profile
    edit "llmapps"
        set feature-set proxy
        config rule
            edit 1
                set name "llmapps1"
                set severity critical
                set proto http-post
                set filter-by sensor
                set sensor "llmapps"
                set action block
            next
            edit 2
                set name "llmapps2"
                set type message
                set proto http-post
                set filter-by sensor
                set sensor "llmapps"
                set action block
            next
        end
    next
end
```

4. Configure the firewall address for the LLM application:

```
config firewall address
    edit <name>
        set type fqdn
        set fqdn <string>
    next
end
```

See FQDN on page 2237 for the FQDN on the specific LLM application.

5. Add the firewall addresses to a group:

```
config firewall addrgrp
    edit "llmapps"
        set member "ChatGpt" "Gemini" "DeepSeek"
```

```
    next
end
```

6. Add the DLP profile and the FQDN address to a firewall policy:

```
config firewall policy
    edit 1
        set name "llmapps "
        set srcintf "port2"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "llmapps"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "deep-inspection"
        set dlp-profile "llmapps"
        set application-list "default"
        set nat enable
    next
end
```

FQDN

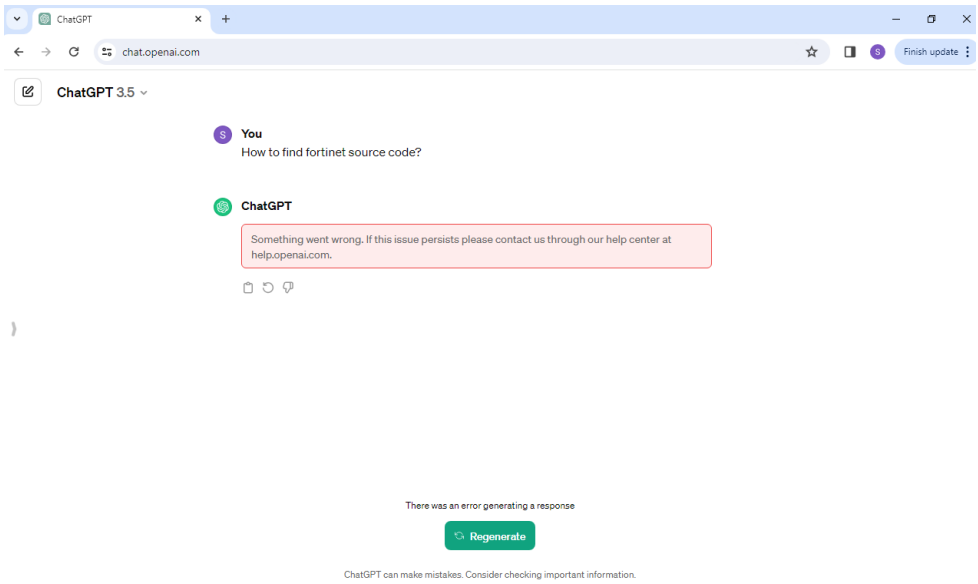
A firewall address should be created for each LLM application:

LLM application	FQDN
ChatGPT	*chat.openai.com*
Gemini	*gemini.google.com*
DeepSeek	*chat.deepseek.com*

Verification

ChatGPT:

1. Visit https://chat.openai.com.
2. Search for any phrase that includes keywords set up in the DLP dictionary.

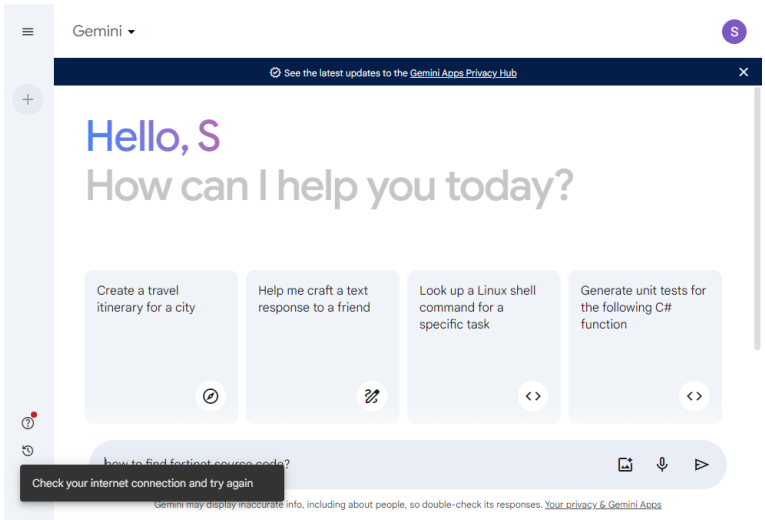


3. Verify that the request failed and an error was generated.
4. Review the log that was generated when the attempt was made to send an HTTP POST request containing sensitive keywords:

```
1: date=2024-03-15 time=09:59:35 eventtime=1710453575538415503 tz="+1200" logid="0954024576"
type="utm" subtype="dlp" eventtype="dlp" level="warning" vd="root" ruleid=1
rulename="llmapps1" dlpextra="Sensor 'llmapps' matching any: ('llmapps'=1) >= 1; match."
filtertype="sensor" filtercat="file" severity="critical" policyid=1 poluuid="eea32b46-db4e-
51ee-92a9-b46e5580db33" policytype="policy" sessionid=69254 epoch=424445846 eventid=1
srcip=13.13.13.13 srcport=56747 srccountry="United States" srcintf="port2"
srcintfrole="undefined" srcuuid="d2f06fda-15e7-51ee-0d22-faaf5170dad2" dstip=104.18.37.228
dstport=443 dstcountry="United States" dstintf="port1" dstintfrole="undefined"
```

Gemini:

1. Visit https://gemini.google.com .
2. Search for any phrase that includes keywords set up in the DLP dictionary.

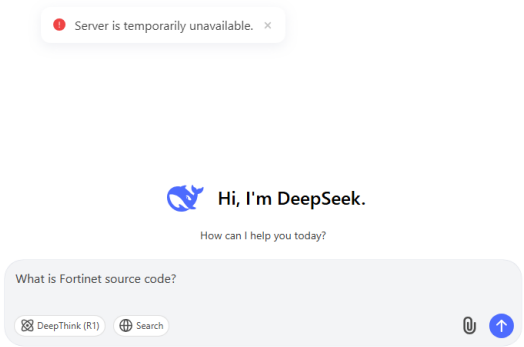


3. Verify that the request failed and an error was generated.
4. Review the log that was generated when the attempt was made to send an HTTP POST request containing sensitive keywords:

```
1: date=2024-03-15 time=12:46:08 eventtime=1710463568053453203 tz="+1200" logid="0954024576"
type="utm" subtype="dlp" eventtype="dlp" level="warning" vd="root" ruleid=1
rulename="llmapps1" dlpextra="Sensor 'llmapps' matching any: ('llmapps'=1) >= 1; match."
filtertype="sensor" filtercat="file" severity="critical" policyid=1 poluuid="eea32b46-db4e-
51ee-92a9-b46e5580db33" policytype="policy" sessionid=77832 epoch=424449372 eventid=1
srcip=13.13.13.13 srcport=58137 srccountry="United States" srcintf="port2"
srcintfrole="undefined" srcuuid="d2f06fda-15e7-51ee-0d22-faaf5170dad2" dstip=142.251.33.110
dstport=443 dstcountry="United States" dstintf="port1" dstintfrole="undefined"
dstuuid="d2f06fda-15e7-51ee-0d22-faaf5170dad2" proto=6 service="HTTPS" filetype="unknown"
direction="outgoing" action="block" hostname="gemini.google.com"
url="https://gemini.google.com/_
/BardChatUi/data/assistant.lamda.BardFrontendService/StreamGenerate?bl=boq_assistant-bard-web-
server_20240313.09_p0&f.sid=2103257702826212605&hl=en&_reqid=1474614&rt=c" agent="Mozilla/5.0
(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/122.0.0.0
Safari/537.36" httpmethod="POST" referralurl="https://gemini.google.com/"
filename="StreamGenerate" filesize=2211 profile="llmapps"
```

DeepSeek:

1. Visit https://chat.deepseek.com.
2. Search for any phrase that includes keywords set up in the DLP dictionary.



3. Verify that the request failed and an error was generated.
4. Review the log that was generated when the attempt was made to send an HTTP POST request containing sensitive keywords:

```
1: date=2025-03-13 time=05:42:35 eventtime=1741797755180776410 tz="+1200" logid="0954024576"
type="utm" subtype="dlp" eventtype="dlp" level="warning" vd="root" ruleid=1
rulename="llmapps1" dlpextra="Sensor 'lmapps' matching any: ('llmapps'=2) >= 1; match."
filtertype="sensor" filtercat="file" severity="critical" policyid=3 poluuid="27681982-ff5c-
51ef-61f2-c3db5a98100c" policytype="policy" sessionid=22163 epoch=1199369450 eventid=1
srcip=10.10.10.13 srcport=65353 srccountry="Reserved" srcintf="port2" srcintfrole="undefined"
srcuuid="d2f06fda-15e7-51ee-0d22-faaf5170dad2" dstip=104.18.27.90 dstport=443
dstcountry="United States" dstintf="port1" dstintfrole="undefined" dstuuid="bb60c556-ff59-
51ef-216d-86ad81ccf871" proto=6 service="HTTPS" filetype="unknown" direction="outgoing"
action="block" hostname="chat.deepseek.com"
```

```
url="https://chat.deepseek.com/api/v0/chat/completion" agent="Mozilla/5.0 (Windows NT 10.0;
Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36"
httpmethod="POST" referralurl="https://chat.deepseek.com/a/chat/s/fcd6fce1-bdd8-4999-b49f-
c27eacb1f960" filename="completion" filesize=187 profile="llmapps"
```

Antivirus

An antivirus software's primary function is to detect and stop viruses that could cause harm to your system or compromise the security of your connected devices. It can be installed on individual endpoints or it can operate as an antivirus engine (AV engine) to perform traffic inspection inside a Next Generation Firewall (NGFW).

The FortiGate's AV engine operates by leveraging the information stored in signature databases that is updated in real-time by the FortiGuard AV services. These databases are essentially vast repositories that contain detailed profiles of known and previously unknown viruses. The AV engine cross-references these profiles with the activities and files on your system to determine if any known or previously unknown viruses are active or attempting to infiltrate your network.

The scope of threats that the antivirus engine can neutralize extends beyond just viruses. It is equipped to deal with a wide array of malicious software, often called malware. This encompasses, but is not confined to, infected files that may carry harmful code, Trojans that disguise themselves as legitimate software, worms that can replicate themselves and spread across networks, and spyware that can collect and transmit your personal information without your consent. Furthermore, inline malware prevention powered by Sandboxing and AI extends protection to even new zero-day malware found in the wild.

This section includes information about antivirus techniques and configurations:

- Antivirus introduction on page 2240
- Advanced configurations on page 2267
- Configuration examples on page 2295

Antivirus introduction

FortiOS offers antivirus solutions in two modes: flow-based and proxy-based. Users can select the mode that best suits their needs. The default setting for a new antivirus profile is flow-based inspection, which is generally recommended unless there is a need for features specific to proxy. For more information, see Protocol comparison between antivirus inspection modes on page 2241.

FortiOS includes two preloaded antivirus profiles:

- *default*
- *wifi-default*

You can customize these profiles, or you can create your own to inspect certain protocols, remove viruses, analyze suspicious files with FortiSandbox, and apply botnet protection to network traffic. Once configured, you can add the antivirus profile to a firewall policy.

This functionality requires a subscription to FortiGuard Antivirus.

In order to download updated AV definitions, at least 1 policy with a security profile that has Antivirus scanning must be enabled.

Protocol comparison between antivirus inspection modes

The following table indicates which protocols can be inspected by the designated antivirus scan modes.

	HTTP	FTP	IMAP	POP3	SMTP	NNTP	MAPI	CIFS	SSH
Proxy	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes*	Yes
Flow	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	No

* Proxy mode antivirus inspection on CIFS protocol has the following limitations:

- Cannot detect infections within some archive files.
- Cannot detect oversized files.

Other antivirus differences between inspection modes

Starting from 6.4.0, the scan mode option is no longer available for flow-based AV.

This means that AV no longer exclusively uses the default or legacy scan modes when handling traffic on flow-based firewall policies. Instead, AV in flow-based policies uses a hybrid of the two scan modes. Flow AV may use a pre-filtering database for malware detection in some circumstances as opposed to the full AV signature database in others. The scan method is determined by the IPS engine algorithm that is based on the type of file being scanned. When handling oversized files in flow-based AV, the action can either be pass (default) or block. When the action is pass, IPS appends to-be-scan data into the AV scan buffer. If the appended file size exceeds the oversize-limit that is defined in the protocol option profile, then the AV session is cleared and the file is bypassed from AV scanning.

In contrast, proxy mode maintains the scan mode option, which can be toggled between default or legacy mode. In default mode, the WAD daemon receives the file and then decides if it can do an in-process scan of the file in simple AV configuration scenarios. If the file is in an oversized archive that is supported by the stream-based decompressor, then it is sent to stream-based scan for best effort inspection. Stream-based scan decompresses and scans the entire archive without archiving the file. If the file is not supported by stream-based scan, then it is buffered and then sent to the scanunit daemon for inspection on content that is under the oversize limit.

In legacy mode, stream-based scanning is disabled, so oversized archive files and files that cannot be handled by WAD in-process scan are buffered and sent to the scanunit daemon for processing.

Antivirus techniques

The security of digital systems is a top priority for organizations. A range of techniques and tools are employed to ensure the integrity and reliability of these systems.

The following table describes some of the industry standard techniques that are used for Antivirus protection, and if they can be configured in the GUI or CLI.

Technique	Description	GUI	CLI
Signature-based detection	Antivirus scan detects and compares malicious file against virus signatures database. The FortiGuard Antivirus Service uses content pattern recognition language (CPRL), which is more efficient and accurate than traditional signature-based detection methods.	✓	✓
Content Disarm and Reconstruction (CDR)	CDR sanitizes Office, OpenOffice, PDF, RTF, and XLSB files by removing active content and active media, preserving only the text. See Content disarm and reconstruction on page 2243 for more information.	✓	✓
Virus Outbreak Prevention (VOS)	VOS enhances FortiGate's antivirus database with third-party malware hashes. It checks file hashes against FortiGuard's database. See Virus outbreak prevention on page 2243 for more information.	✓	✓
Zero-day malware stream scanning	Enables real-time delivery of malware IOCs (Indicators of Compromise) to FortiGate devices using the fortimq daemon and notifications, eliminating the need for frequent cloud polling and reducing server load. See Zero-day malware stream scanning on page 2299 for more information.	✓	✓
External Malware Block List	Users can add their own malware signatures to an external list. See External malware block list on page 2243 for more information.	✓	✓
EMS External Feed	FortiGate receives malware feeds from FortiClient EMS, which itself gathers detected malware hashes from FortiClients. See EMS external feed on page 2244 for more information.	✓	✓
Behavior-based detection	Submit suspected malicious files to FortiSandbox for inspection. See Using FortiSandbox post-transfer scanning with antivirus on page 2267 and Using FortiSandbox inline scanning with antivirus on page 2269 for more information.	✓	✓
CIFS Scanning	File filtering and antivirus scanning on Common Internet File System (CIFS) traffic is supported. See CIFS support on page 2289 for more information.	✓	✓
Heuristic Analysis	Identify malicious files such as Windows Portable Executables (PEs) to combat zero-day attacks. See AI-based malware detection on page 2244 for more information.		✓
AI/ML, behavioral, and human analysis	Helps identify, classify, and respond to threats. See Using FortiNDR inline scanning with antivirus on page 2284 for more information.	✓	✓

See Configuring an antivirus profile on page 2244 and Testing an antivirus profile on page 2247 for more information.

Content disarm and reconstruction

Content disarm and reconstruction (CDR) allows the FortiGate to sanitize Microsoft Office documents and PDF files (including those that are in ZIP archives) by removing active content, such as hyperlinks, embedded media, JavaScript, macros, and so on from the files (disarm) without affecting the integrity of its textual content (reconstruction).

CDR is supported on HTTP, SMTP, POP3, and IMAP.

HTTP GET is supported, but not HTTP POST.

SMTP splice and client-comfort mode are not supported.

CDR does not support flow-based inspection modes.

It allows network administrators to protect their users from malicious document files. See Content disarm and reconstruction on page 2295 for a configuration example.

Virus outbreak prevention

FortiGuard VOS allows the FortiGate antivirus database to be supplemented with third-party malware hash signatures curated by FortiGuard. This allows VOS to manage zero-day threats effectively. The hash signatures are obtained from FortiGuard's Global Threat Intelligence database. Any signature that is added to FortiGuard becomes immediately active, eliminating the need to wait for AVDB (antivirus database) update. The AVDB queries FortiGuard with the hash of a scanned file. If FortiGuard returns a match, the scanned file is deemed to be malicious. Enabling the AV engine scan is not required to use this feature.

FortiGuard VOS can be used in both proxy-based and flow-based policy inspections across all supported protocols.

The FortiGate must be registered with a valid FortiGuard outbreak prevention license.

See FortiGuard outbreak prevention on page 2297 for a configuration example.

External malware block list

The external malware block list allows users to add their own malware signatures in the form of MD5, SHA1, and SHA256 hashes. The FortiGate's antivirus database retrieves an external malware hash list from a remote server and polls the hash list every n minutes for updates. Enabling the AV engine scan is not required to use this feature.

The external malware block list can be used in both proxy-based and flow-based policy inspections, but it is not supported in AV quick scan mode.

Note that using different types of hashes simultaneously may slow down the performance of malware scanning. It is recommended to use one type of hash.

See External malware block list on page 2306 and Malware hash external feed on page 4720 for more details and configuration examples.

EMS external feed

A FortiGate can pull malware external feeds from FortiClient EMS, which in turn receives malware hashes detected by FortiClients. The malware hash can be used in an antivirus profile when AV scanning is enabled with block or monitor actions. This feature is supported in proxy and flow mode.

If an external malware blocklist and the FortiGuard outbreak prevention database are also enabled in the antivirus profile, the checking order is: AV local database, EMS external feed, external malware blocklist, FortiGuard outbreak prevention database. If the EMS external feed and external malware blocklist contain the same hash value, then the EMS infection will be reported if both of them are blocked.

See Malware external feed from EMS on page 2287 for more details and configuration examples.

AI-based malware detection

The AV Engine AI malware detection model integrates into regular AV scanning to help detect potentially malicious files, such as malicious Windows Portable Executables or ELF files, in order to mitigate zero-day attacks. Previously, this type of detection was handled by heuristics that analyzed file behavior. With AV Engine AI, the module is trained by FortiGuard AV against many malware samples to identify file features that make up the malware. The AV Engine AI package can be downloaded by FortiOS via FortiGuard on devices with an active AV subscription. The machine-learning-detection setting is enabled by default at a per-VDOM level. Files detected by the AV Engine AI are identified with the W32/AI.Pallas.Suspicious virus signature.

To configure machine learning-based malware detection:

```
config antivirus settings
     set machine-learning-detection {enable| monitor | disable}
end
```

FortiGuard provides several sample files to test the AV configuration on the FortiGate, which are available to download from https://www.fortiguard.com/sample-files. Test the AI-based malware detection feature by downloading *AI Sample* file. See Example 2: AI sample file on page 2252.

Configuring an antivirus profile

In an antivirus profile, the FortiGate can be configured to apply antivirus protection to HTTP, FTP, IMAP, POP3, SMTP, CIFS, and NNTP sessions. Proxy-based profiles also support MAPI and SSH. Antivirus inspection prevents potentially unwanted and malicious files from entering the network. Antivirus profiles include multiple different functions, such as scanning files for virus signatures, scanning for advanced persistent threats, checking external malware hash lists and external feeds, and others. Malicious files can be blocked or monitored, and can be quarantined. Some antivirus profile options require a license and/or other Fortinet products. Some antivirus profile options can only be configured in the CLI (refer to the FortiOS CLI Reference).

The feature set setting (proxy or flow) in the antivirus profile must match the inspection mode setting (proxy or flow) in the associated firewall policy. For example, a flow-based antivirus profile must be used with a flow-based firewall policy.

To configure an antivirus profile:

1. Go to *Security Profiles* > *AntiVirus* and click *Create New*.
2. Configure the following settings:

Name	Enter a unique name for the profile.
Comments	Enter a comment (optional).
AntiVirus scan	Enable one or more protocols for inspection, then enable *AntiVirus scan* for the selected protocols with a specified action. • *Block*: block the malicious traffic. • *Monitor*: log malicious traffic and allow it to pass inspection.

Feature set	Select the feature set for the profile. The feature set mode must match the inspection mode used in the associated firewall policy.
	• *Flow-based*
	• *Proxy-based*
	Additional options are available in proxy-based mode and are identified in the GUI with a *P* icon. See Inspection mode feature comparison on page 2214 for more details.
	If the *Feature set* option is not visible, enter the following in the CLI:
	`config system settings`
	`  set gui-proxy-inspection enable`
	`end`

Inspected Protocols	Enable to inspect the protocol for session inspection: HTTP, SMTP, POP3, IMAP, FTP, and CIFS. Disabled protocols are not inspected. MAPI and SSH can be inspected in proxy-based mode.
APT Protection Options	This section includes options available with FortiGuard to mitigate advanced persistent threats (APT) in file-based attacks.
Content Disarm and Reconstruction	This option is available in proxy-based mode when at least one protocol is enabled for inspection and *AntiVirus scan* is enabled. See Content disarm and reconstruction on page 2243 for more details.
Allow transmission when an error occurs	Enable to allow traffic to pass when an inspection error occurs. Disable to block traffic when an inspection error occurs.
Original File Destination	Specify how to quarantine files processed by content disarm and reconstruction. • *FortiSandbox*: quarantine files on FortiSandbox. The FortiSandbox must be enabled. See Using FortiSandbox post-transfer scanning with antivirus on page 2267 for more details. • *File Quarantine*: quarantine files on FortiGate models with a hard disk. • *Discard*: discard suspicious files.

Treat Windows executables in email attachments as viruses	Enable to deem all Windows executable files located in email traffic as viruses.
Send Files to FortiSandbox for Inspection	Enable to send files to FortiSandbox for inspection. The FortiSandbox must be enabled.
Scan strategy	FortiSandbox scans files inline for flow-based mode (*Inline*) and after the file transfer is complete for proxy-based mode (*Post Transfer*). See Using FortiSandbox inline scanning with antivirus on page 2269 and Using FortiSandbox post-transfer scanning with antivirus on page 2267 for more details.
File types	Specify which files to FortiSandbox for inspection. • *Suspicious Files Only*: only send suspicious files to FortiSandbox for inspection. • *All Supported Files*: send all supported files to FortiSandbox for inspection.
Do not submit files matching types	Click the + to exclude certain file types from being sent to FortiSandbox.
Do not submit files matching file name patterns	Click the + to enter a wildcard pattern to exclude files from being sent to FortiSandbox.
Use FortiSandbox database	Enable to use the signature database from FortiSandbox. The FortiSandbox must be enabled.
Send files to FortiNDR for inspection	This option is available in proxy-based mode when at least one protocol is enabled for inspection, *AntiVirus scan* is enabled, and FortiNDR is enabled. See Using FortiNDR inline scanning with antivirus on page 2284 for more details.
Include mobile malware protection	Enable to use the mobile malware protection database from FortiGuard for content scanning.
Quarantine	This option is available when at least one protocol is enabled for inspection and *AntiVirus scan* is enabled. Enable to quarantine infected files. See also Downloading quarantined files in archive format on page 2310.
Virus Outbreak Prevention	This section includes options available with the FortiGuard Virus Outbreak Protection Service (VOS). See Virus outbreak prevention on page 2243 and FortiGuard outbreak prevention on page 2297 for more details.
Use FortiGuard outbreak prevention database	Enable to use the outbreak prevention database that is available with Advanced Malware Protection on FortiGuard. A license is required. • *Block*: block the malicious traffic. • *Monitor*: log malicious traffic and allow it to pass inspection.
Use 0-day malware stream scanning	Enable to use real-time delivery of malware IOCs to FortiGate devices using fortimq daemon.

	• *Block*: block the malicious traffic. • *Monitor*: log malicious traffic and allow it to pass inspection. See Zero-day malware stream scanning on page 2299 for more details.
Use external malware block list	Enable to use one or more external blocklist file hashes. • *Block*: block the malicious traffic. • *Monitor*: log malicious traffic and allow it to pass inspection. • *All*: use all malware block lists. • *Specify*: select specific malware block lists. See External malware block list on page 2243and Malware hash external feed on page 4720 for more details.
Use EMS external feed	This option is available when at least one protocol is enabled for inspection and *AntiVirus scan* is enabled. Enable to use malware external feeds from FortiClient EMS. A FortiClient EMS Fabric connector with EMS external feed enabled is required. See EMS external feed on page 2244 for more details.

3. Click *OK*.

Protocol options

When applying an antivirus profile to a firewall policy, the protocol options profile defines parameters for handling protocol-specific traffic. These parameters affect functions such as the port mapping for inspecting each protocol, whether to log or block oversized files when performing AV scanning, enabling comfort client, and more. Protocol options profiles are configured by going to *Policy & Objects* > *Protocol Options*, or in the CLI under `config firewall profile-protocol-options`. See Protocol options on page 2101 for more information.

Scan mode

In proxy-based antivirus profiles, the scan mode can be set to either default or legacy. This setting can only be configured in the CLI. See Proxy mode stream-based scanning on page 2259 for more information.

To configure the scan mode:

```
config antivirus profile
    edit <name>
        set feature-set proxy
        set scan-mode {default | legacy}
    next
end
```

Testing an antivirus profile

Antivirus (AV) profiles can be tested using various file samples to confirm whether AV is correctly configured. In this topic, an AV profile is configured, applied to a firewall policy, and a user attempts to download sample virus test files hosted on eicar.org and fortiguard.com.

Different sample files are used to verify different features on the AV profile. The expectation is these files must be blocked by the AV profile, and the user should be presented with a block page.

File	Test case
EICAR test file	A plain text EICAR test file (hosted on eicar.org over a HTTPS connection) to test basic AV scanning on the FortiGate using deep inspection.
AI sample file	A machine learning sample file to test AI-based malware detection on the FortiGate.
Virus outbreak (VO) sample file	A zero-day sample virus file to test the outbreak prevention feature of the AV profile.
Behavioral-based samples	Files that are detected by a sandbox. This requires FortiSandbox integration with the FortiGate.

For the following AV test cases, the test PC has an IP of 192.168.1.110/24 and is connected to the internal1 interface. It accesses the internet through the wan1 interface.

Configuring the AV profile

The *default* AV profile is used, and the *Use FortiGuard outbreak prevention database* setting is enabled with the action set to block.

To configure the AV profile:

1. Go to *Security Profiles* > *AntiVirus* and edit the *default* profile.
2. In the *Virus Outbreak Prevention* section, enable *Use FortiGuard outbreak prevention database* and select *Block*. See FortiGuard outbreak prevention on page 2297 for more information about this setting.
3. Configure the other settings as needed (see Configuring an antivirus profile on page 2244).
4. Click *OK*.

By default, the FortiOS AV Engine has AI-based malware detection enabled (`set machine-learning-detection enable`) . The AV Engine AI malware detection model integrates into regular AV scanning to help detect potentially malicious Windows Portable Executables (PEs) in order to mitigate zero-day attacks. See AI-based malware detection on page 2244 for more information.

To verify the status of the AV Engine AI contract:

```
# diagnose autoupdate versions | grep AI -A6
AI/Machine Learning Malware Detection Model
---------
Version: 2.12588 signed
Contract Expiry Date: Tue Jul  9 2024
Last Updated using scheduled update on Tue Sep  5 08:23:15 2023
```

```
Last Update Attempt: Tue Sep  5 09:23:00 2023
Result: No Updates
```

Configuring the SSL SSH profile and firewall policy

The PC will be accessing and downloading the test files using HTTPS from the EICAR and the FortiGuard websites. Since HTTPS traffic is encrypted traffic, in order for the FortiGate to scan the encrypted traffic and inspect it for viruses and malware, it should act as the machine-in-the-middle to decrypt this communication and then re-encrypt it to send it to the website. Deep inspection must be enabled in the SSL SSH profile that will be applied to the firewall policy (see Deep inspection on page 2697). The *custom-deep-inspection* profile is modified to remove the *fortinet* FQDN address from the exemption list.

To configure the SSL SSH profile:

1. Go to *Security Profiles* > *SSL/SSH Inspection* and edit the *custom-deep-inspection* profile.
2. In the *Exempt from SSL Inspection* section, locate the *fortinet* FQDN entry in the *Addresses* field, and click the *X* to delete it.
3. Click *OK*.

To configure the firewall policy:

1. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
2. Configure the following settings:

Name	*To Internet*
Incoming Interface	*internal1*
Outgoing Interface	*wan1*
AntiVirus	Enable and select *default*.
SSL Inspection	Select *custom-deep-inspection*.

3. Configure the other settings as needed (see Firewall policy on page 1842).

   The feature set setting (proxy or flow) in the antivirus profile must match the inspection mode setting (proxy or flow) in the associated firewall policy. For example, a flow-based antivirus profile must be used with a flow-based firewall policy.

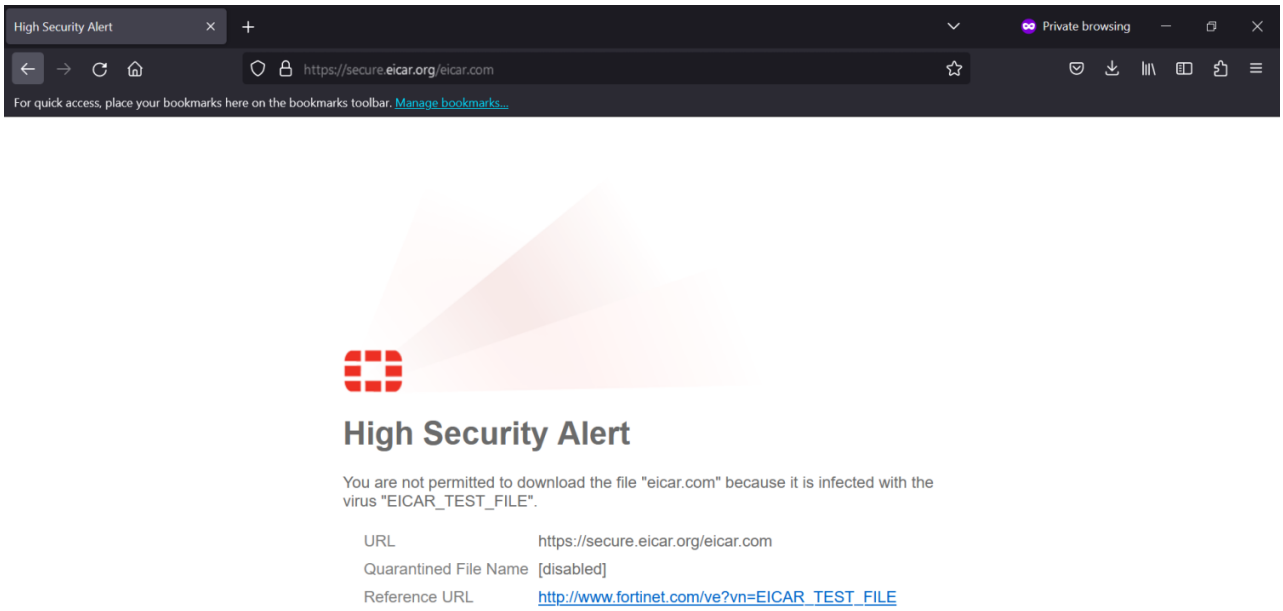
4. Click *OK*.

Example 1: EICAR test file

EICAR hosts anti-malware test files, which are available to download from https://www.eicar.org/download-anti-malware-testfile.

To test the AV profile with the EICAR test file:

1. On the PC, go to the EICAR website and download the *eicar.com* file.
2. The download attempt is blocked by the FortiGate's *default* AV profile, and a block page appears in the PC's browser.



3. Check the antivirus statistics on the FortiGate, `HTTP virus detected` increases by one:

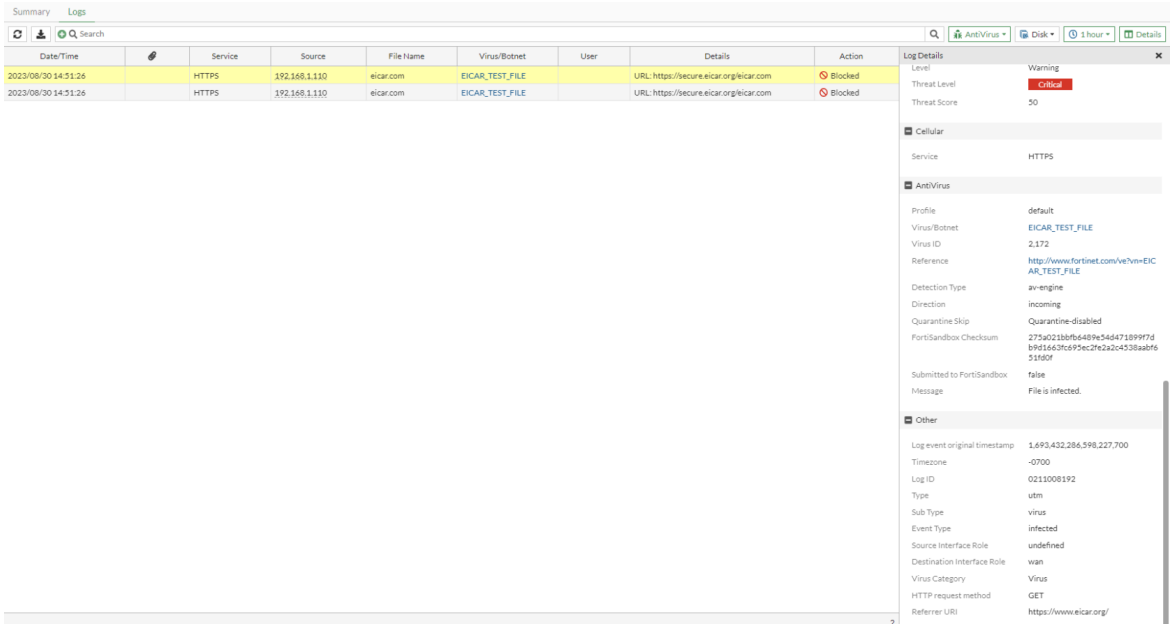
```
# diagnose ips av stats show
 AV stats:
 HTTP virus detected: 1
 HTTP virus blocked: 0
 SMTP virus detected: 0
 SMTP virus blocked: 0
 POP3 virus detected: 0
 POP3 virus blocked: 0
 IMAP virus detected: 0
 IMAP virus blocked: 0
 NNTP virus detected: 0
 NNTP virus blocked: 0
 FTP virus detected: 0
 FTP virus blocked: 0
 SMB virus detected: 0
 SMB virus blocked: 0
```

4. Check the antivirus statistics using an SNMP walk:

```
root:~# snmpwalk -c public -v 1 10.1.100.6 1.3.6.1.4.1.12356.101.8.2.1.1
 iso.3.6.1.4.1.12356.101.8.2.1.1.1 = Counter32: 2  (fgAvVirusDetected)
 iso.3.6.1.4.1.12356.101.8.2.1.1.2 = Counter32: 1  (fgAvVirusBlocked)
 iso.3.6.1.4.1.12356.101.8.2.1.1.3 = Counter32: 1  (fgAvHTTPVirusDetected)
 iso.3.6.1.4.1.12356.101.8.2.1.1.4 = Counter32: 0
```

```
iso.3.6.1.4.1.12356.101.8.2.1.1.5.1 = Counter32: 0
iso.3.6.1.4.1.12356.101.8.2.1.1.6.1 = Counter32: 0
iso.3.6.1.4.1.12356.101.8.2.1.1.7.1 = Counter32: 0
iso.3.6.1.4.1.12356.101.8.2.1.1.8.1 = Counter32: 0
iso.3.6.1.4.1.12356.101.8.2.1.1.9.1 = Counter32: 0
iso.3.6.1.4.1.12356.101.8.2.1.1.10.1 = Counter32: 0
iso.3.6.1.4.1.12356.101.8.2.1.1.11.1 = Counter32: 0
iso.3.6.1.4.1.12356.101.8.2.1.1.12.1 = Counter32: 0
iso.3.6.1.4.1.12356.101.8.2.1.1.13.1 = Counter32: 0
iso.3.6.1.4.1.12356.101.8.2.1.1.14.1 = Counter32: 0
iso.3.6.1.4.1.12356.101.8.2.1.1.15.1 = Counter32: 0
iso.3.6.1.4.1.12356.101.8.2.1.1.16.1 = Counter32: 0
iso.3.6.1.4.1.12356.101.8.2.1.1.17.1 = Counter32: 0
iso.3.6.1.4.1.12356.101.8.2.1.1.18.1 = Counter32: 0
iso.3.6.1.4.1.12356.101.8.2.1.1.19.1 = Counter32: 0
iso.3.6.1.4.1.12356.101.8.2.1.1.20.1 = Counter32: 0
iso.3.6.1.4.1.12356.101.8.2.1.1.21.1 = Counter32: 0
iso.3.6.1.4.1.12356.101.8.2.1.1.22.1 = Counter32: 0
```

5. Verify the AV log.
   a. In the GUI, go to *Log & Report* > *Security Events* and select the *AntiVirus* card. Select the log entry and click *Details*.



   b. In the CLI, enter the following:

```
# execute log filter category 2
# execute log display
date=2023-08-30 time=14:51:26 eventtime=1693432286598227820 tz="-0700" logid="0211008192"
type="utm" subtype="virus" eventtype="infected" level="warning" vd="root" policyid=1
poluuid="c65fa590-4758-51ee-4d28-f2cc75f14979" policytype="policy" msg="File is infected."
action="blocked" service="HTTPS" sessionid=15797 srcip=192.168.1.110 dstip=89.238.73.97
srcport=64641 dstport=443 srccountry="Reserved" dstcountry="Germany" srcintf="internal1"
```

```
srcintfrole="undefined" dstintf="wan1" dstintfrole="wan" srcuuid="ab8d1c24-30b1-51ee-138a-
f7be846c205d" dstuuid="ab8d1c24-30b1-51ee-138a-f7be846c205d" proto=6 direction="incoming"
filename="eicar.com" quarskip="Quarantine-disabled" virus="EICAR_TEST_FILE"
viruscat="Virus" dtype="av-engine" ref="http://www.fortinet.com/ve?vn=EICAR_TEST_FILE"
virusid=2172 url="https://secure.eicar.org/eicar.com" profile="default" agent="Mozilla/5.0
(Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/116.0" httpmethod="GET"
referralurl="https://www.eicar.org/"
analyticscksum="275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f"
analyticssubmit="false" crscore=50 craction=2 crlevel="critical"
```

6. Optionally, reset the antivirus statistics to zero:

```
# diagnose ips av stats clear
```

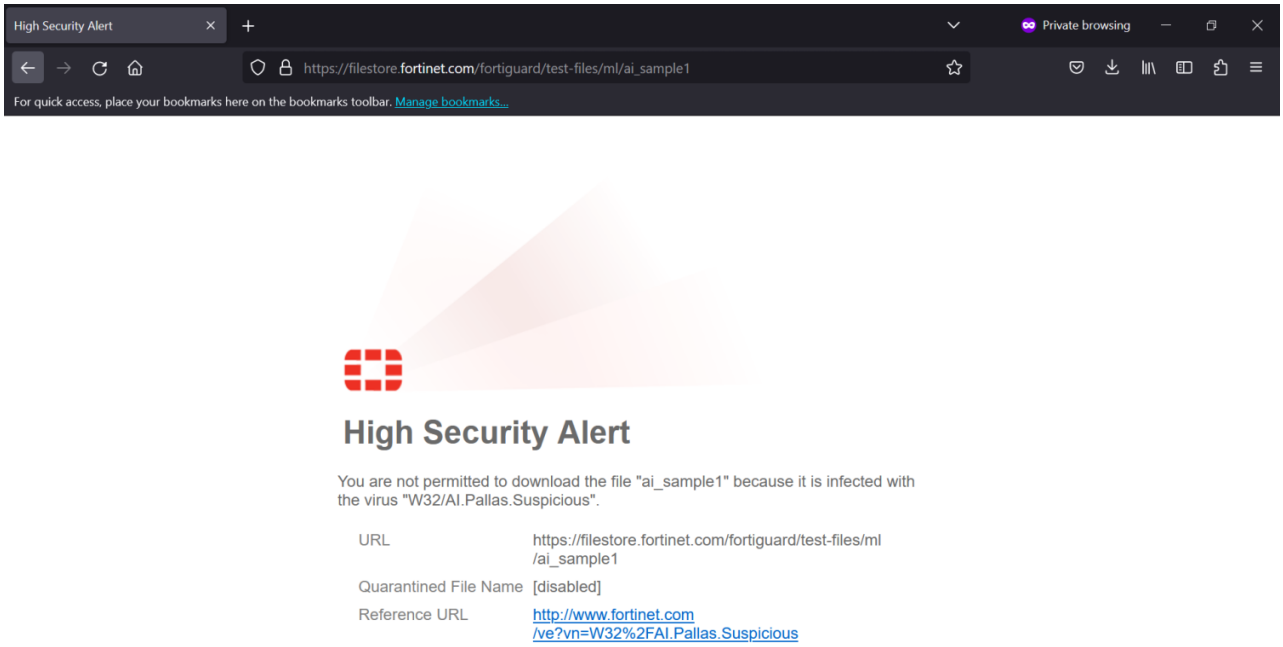
See CLI troubleshooting cheat sheet for log gathering, analysis, and troubleshooting.

Example 2: AI sample file

FortiGuard provides several sample files to test the AV configuration on the FortiGate, which are available to download from https://www.fortiguard.com/sample-files.

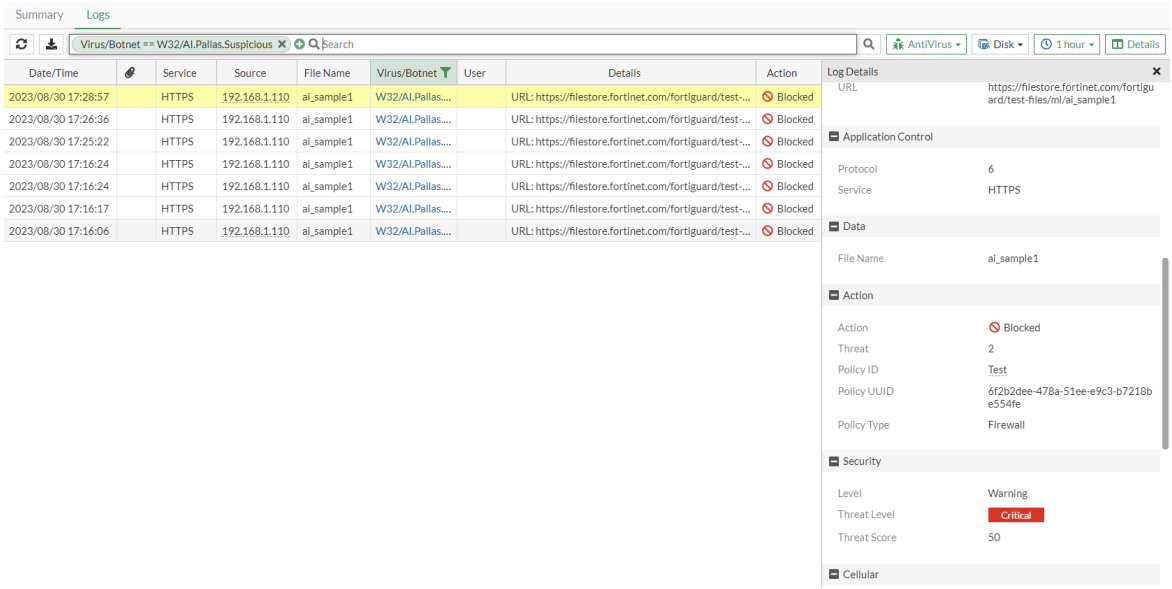
To test the AV profile with the AI sample file:

1. On the PC, go to the FortiGuard website and download the *AI Sample* file.
2. The download attempt is blocked by the FortiGate's *default* AV profile, and a block page appears in the PC's browser.



   The file is blocked due to AI-based malware detection and will be logged. Files detected by the AV Engine AI are identified with the W32/AI.Pallas.Suspicious virus signature in the AV logs.
3. Verify the AV log.

a. In the GUI, go to *Log & Report* > *Security Events* and select the *AntiVirus* card. Select the log entry and click *Details*.



b. In the CLI, enter the following:

```
# execute log filter category 2
# execute log display
date=2023-08-30 time=17:28:57 eventtime=1693441737721077640 tz="-0700" logid="0211008192"
type="utm" subtype="virus" eventtype="infected" level="warning" vd="root" policyid=1
poluuid="c65fa590-4758-51ee-4d28-f2cc75f14979" policytype="policy" msg="File is infected."
action="blocked" service="HTTPS" sessionid=1179 srcip=192.168.1.110 dstip=209.52.38.129
srcport=63117 dstport=443 srccountry="Reserved" dstcountry="Canada" srcintf="internal1"
srcintfrole="undefined" dstintf="wan1" dstintfrole="wan" srcuuid="6c43f8d6-478a-51ee-95d8-
31177232e869" dstuuid="6c43f8d6-478a-51ee-95d8-31177232e869" proto=6 direction="incoming"
filename="ai_sample1" quarskip="Quarantine-disabled" virus="W32/AI.Pallas.Suspicious"
viruscat="Virus" dtype="av-engine"
ref="http://www.fortinet.com/ve?vn=W32%2FAI.Pallas.Suspicious" virusid=8187637
url="https://filestore.fortinet.com/fortiguard/test-files/ml/ai_sample1" profile="default"
agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0) Gecko/20100101 Firefox/117.0"
httpmethod="GET" referralurl="https://www.fortiguard.com/"
analyticscksum="7057e364dbf09b6de7a6cc152b8967e50ed86a0edf97cfd2e88b142ac41873f0"
analyticssubmit="false" crscore=50 craction=2 crlevel="critical"
```

4. Verify the AV (scanunit daemon) real-time debug:

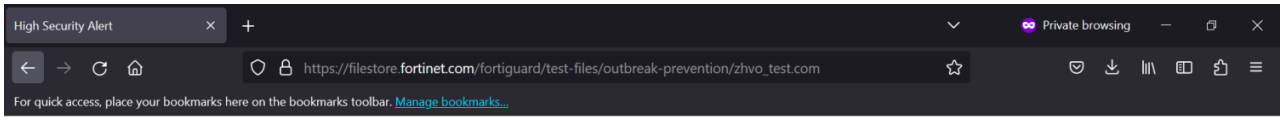
```
# diagnose sys scanunit debug  all
# diagnose sys scanunit debug level verbose
su 4655 req vfid 0 id 2 ep 0 new request from ipsengine pid 4998, size 4096, fwd-pol 1,
oversize 0, url-exempt 0x0, ff-done 0, partial-data 0, dir srv->clt, http-block 0
su 4655 job 157 req vfid 0 id 2 ep 0 received; ack 157, data type: 2
su 4655 job 157 request info:
su 4655 job 157   client N/A server N/A
su 4655 job 157   object_name 'ai_sample1'
su 4655 job 157 heuristic scan enabled
su 4655 job 157 enable databases 0f (core avai mmdb extended)
```

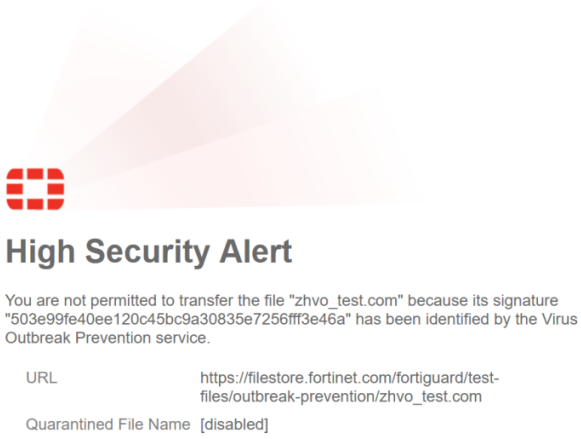
```
su 4655 job 157 scan file 'ai_sample1' bytes 4096
su 4655 job 157 file-hash query, level 0, filename 'ai_sample1' size 4096
su 4655 job 157   sha1 'e027a991fd3f03961d05d25cd27617d2945be10b'
su 4655 job 157 scan return status 2
su 4655 job 157 scan status 2 infection 2 virus 8187637 'W32/AI.Pallas.Suspicious' s_type 4
cate 0 fsize 4096 hr 100 checksum 1619585399
su 4655 job 157 add quarantine file 'ai_sample1' virus 'W32/AI.Pallas.Suspicious' infection_
type 2
su 4655 job 157 settings are such that file won't be quarantined
su 4655 job 157 not wanted for analytics: post-transfer scan submission is disabled at
protocol level (m 2 r 2)
su 4655 job 157 report HEURISTIC infection priority 1
su 4655 job 157 insert infection HEURISTIC SUCCEEDED loc (nil) off 0 sz 0 at index 0 total
infections 1 error 0
su 4655 job 157 send result
su 4655 job 157 close
su 4654 open
```

Example 3: VO sample file

To test the AV profile with the VO sample file:

1. On the PC, go to the FortiGuard website and download the *VO Sample* file.
2. The download attempt is blocked by the FortiGate's *default* AV profile, and a block page appears in the PC's browser.

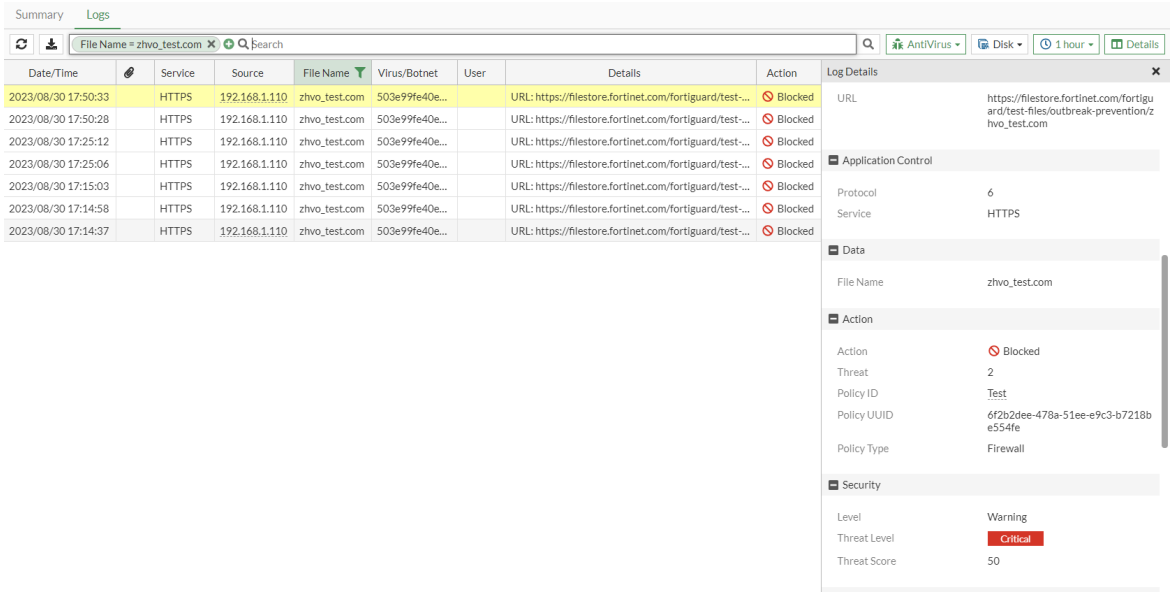




The file is blocked due to the virus outbreak protection service and database that is enabled in the *default* AV profile.

3. Verify the AV log.

a. In the GUI, go to *Log & Report* > *Security Events* and select the *AntiVirus* card. Select the log entry and click *Details*.



b. In the CLI, enter the following:

```
# execute log filter category 2
# execute log display
date=2023-08-30 time=17:50:33 eventtime=1693443033509250120 tz="-0700" logid="0204008202"
type="utm" subtype="virus" eventtype="outbreak-prevention" level="warning" vd="root"
policyid=1 poluuid="c65fa590-4758-51ee-4d28-f2cc75f14979" policytype="policy" msg="Blocked
by Virus Outbreak Prevention service." action="blocked" service="HTTPS" sessionid=2501
srcip=192.168.1.110 dstip=209.52.38.129 srcport=63450 dstport=443 srccountry="Reserved"
dstcountry="Canada" srcintf="internal1" srcintfrole="undefined" dstintf="wan1"
dstintfrole="wan" srcuuid="6c43f8d6-478a-51ee-95d8-31177232e869" dstuuid="6c43f8d6-478a-
51ee-95d8-31177232e869" proto=6 direction="incoming" filename="zhvo_test.com"
quarskip="Quarantine-disabled" virus="503e99fe40ee120c45bc9a30835e7256fff3e46a"
viruscat="File Hash" dtype="outbreak-prevention"
filehash="503e99fe40ee120c45bc9a30835e7256fff3e46a" filehashsrc="fortiguard"
url="https://filestore.fortinet.com/fortiguard/test-files/outbreak-prevention/zhvo_
test.com" profile="default" agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:109.0)
Gecko/20100101 Firefox/117.0" httpmethod="GET" referralurl="https://www.fortiguard.com/"
analyticssubmit="false" crscore=50 craction=2 crlevel="critical"
```

Example 4: behavioral-based samples detected by a sandbox

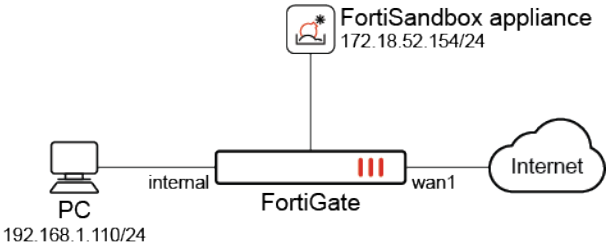
The FortiGate can be integrated with FortiSandbox appliance (used in this example), FortiSandbox Cloud, or FortiGate Cloud Sandbox. See Configuring sandboxing on page 4354 for more information about configuring the different FortiSandbox sandbox solutions.

There are two types of scan strategies that can be used in an AV profile.

- Inline: files are sent to the FortiSandbox, and the FortiGate waits for a verdict before completing the file transfer to the client (see Using FortiSandbox inline scanning with antivirus on page 2269).

- Post-transfer: files are sent to the FortiSandbox, but the FortiGate does not wait for a verdict and completes the file transfer to the client (see Using FortiSandbox post-transfer scanning with antivirus on page 2267).
  - After the FortiSandbox scans and presents a verdict, it updates its malware signature database and the FortiGate retrieves the malware signature database from FortiSandbox if the FortiSandbox database on page 2265 is enabled.
  - If a user attempts to download the file again, the FortiGate will either block or allow the download depending on the FortiSandbox verdict.

This example uses the inline AV scan strategy with a Windows executable sample file, and assumes that the scan profile has already been configured in FortiSandbox. See Verify the FortiSandbox Analysis in the FortiSandbox Administration Guide for more information.



To test the AV profile with a Windows executable sample file:

1. Integrate the FortiGate with the FortiSandbox appliance using the Security Fabric (see Configuring sandboxing on page 4354).
2. Update the AV profile to use inline scanning (see FortiSandbox appliance inline scanning).
3. On the PC, go to the FortiGuard website, hover over the *Windows Executable* link, right-click, and select the browser's option to copy the link.
4. Open another browser tab, paste the URL, and append the URL with `&s=<string>`, such as `https://filegen.fortinet.com/v1/sandbox-file?file_name=windows.exe&s=ftnt`.

   Every file download attempt from the FortiGuard website downloads a new file. Downloading a file with same `<string>` ensures that the downloaded file is the same, and not a new file. A file named windows.exe starts to download.

   Since inline AV scanning is in use, the client's file is held while it is sent to FortiSandbox for inspection. During this time, the FortiGate may apply client comforting (see Protocol options on page 2101) by leaking a certain amount of bytes at a certain time interval to the client.

   Once a verdict is returned from FortiSandbox, the appropriate action (allow or block) is performed on the held file. If there is an error connecting to the FortiSandbox or a timeout on the FortiSandbox scanning the file within the default 50 seconds, the file can be passed, logged, or blocked based on FortiGate's configuration. In this example, the action is set to the default (`log-only`), which allows the file to be downloaded if FortiSandbox takes more than 50 seconds to present a verdict.

   In the first attempt, the windows.exe file is downloaded on the client's PC if FortiSandbox takes more than 50 seconds for scanning the file, after which the FortiGate encounters a scan timeout.

   a. To change the FortiSandbox error and timeout settings in the AV profile, enter the following in the CLI:

```
config antivirus profile
   edit <name>
       set fortisandbox-error-action {ignore | log-only | block}
```

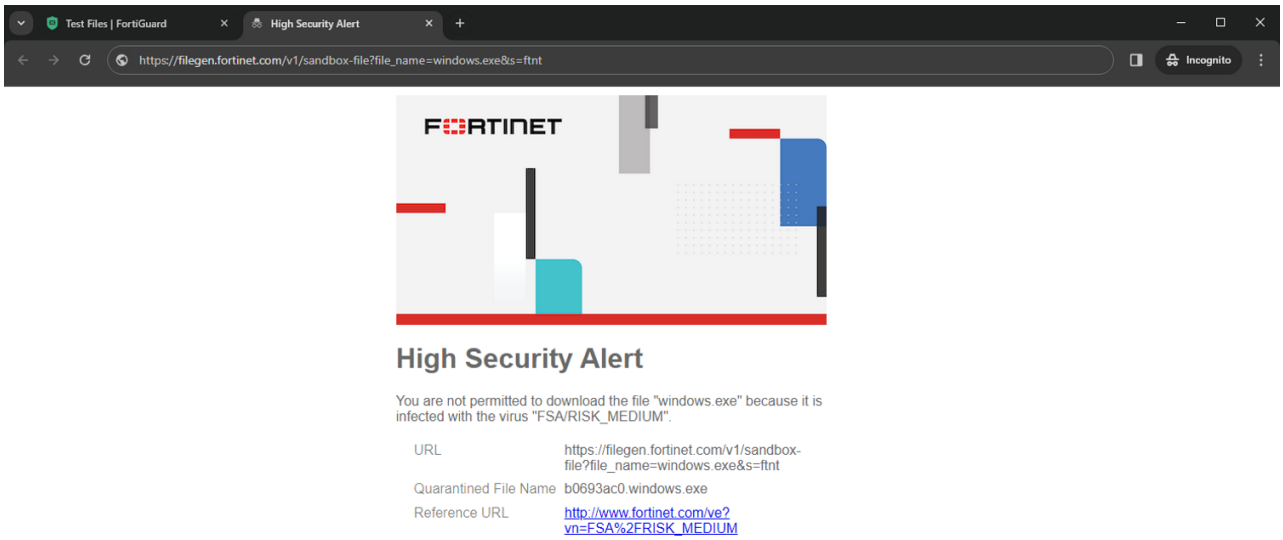
```
        set fortisandbox-timeout-action {ignore | log-only | block}
    next
end
```

`fortisandbox-error-action {ignore | log-only | block}`	Set the action to take if FortiSandbox inline scanning encounters an error reaching the FortiSandbox: • `ignore`: take no action • `log-only`: log the FortiSandbox inline scan error, but allow the file (default) • `block`: block the file upon FortiSandbox inline scan error
`fortisandbox-timeout-action {ignore | log-only | block}`	Set the action to take if FortiSandbox inline scanning encounters a scan timeout: • `ignore`: take no action • `log-only`: log the FortiSandbox inline scan timeout, but allow the file (default) • `block`: block the file upon FortiSandbox inline scan timeout

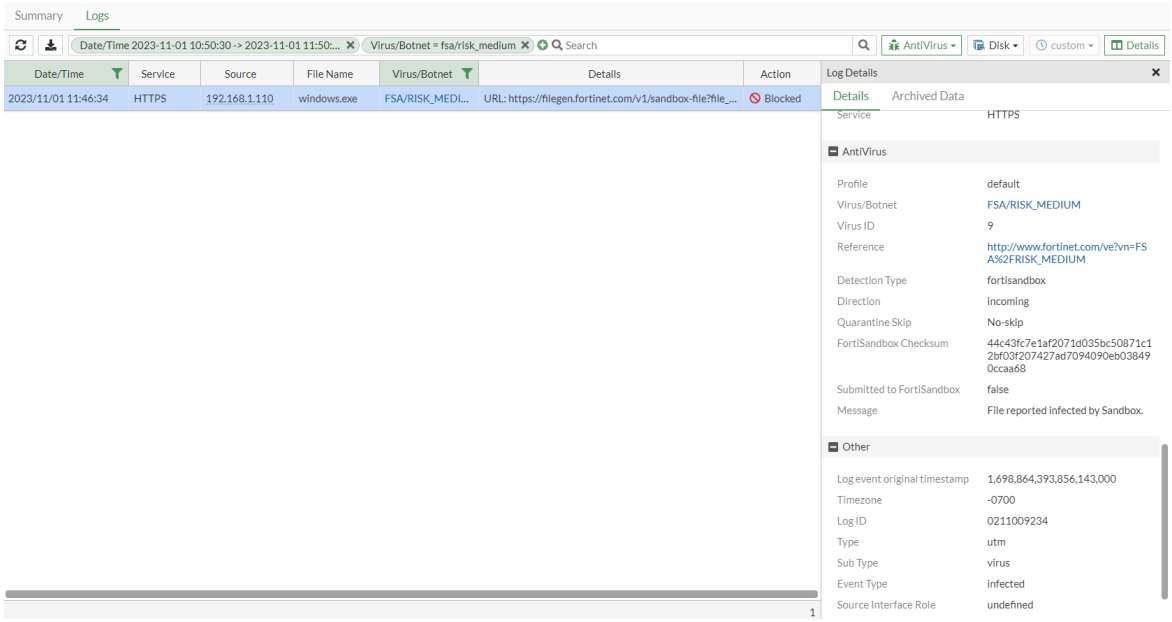
5. Verify the AV log:

```
# execute log filter category 2
# execute log display
date=2023-11-01 time=11:30:45 eventtime=1698863444363194189 tz="-0700" logid="0209008225"
type="utm" subtype="virus" eventtype="inline-block" level="notice" vd="root" policyid=1
poluuid="6f2b2dee-478a-51ee-e9c3-b7218be554fe" policytype="policy" msg="Inline Block scan
timeout." action="monitored" service="HTTPS" sessionid=946497 srcip=192.168.1.110
dstip=149.5.234.147 srcport=57497 dstport=443 srccountry="Reserved" dstcountry="France"
srcintf="internal1" srcintfrole="undefined" dstintf="wan1" dstintfrole="wan"
srcuuid="a051eeb2-5284-51ee-99d0-d8b19cb7439d" dstuuid="6c43f8d6-478a-51ee-95d8-31177232e869"
proto=6 direction="incoming" filename="windows.exe" quarskip="No-quarantine-for-inline-block-
error" url="https://filegen.fortinet.com/v1/sandbox-file?file_name=windows.exe&s=ftnt
profile="default" agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML,
like Gecko) Chrome/118.0.0.0 Safari/537.36" httpmethod="GET" analyticssubmit="false"
icbaction="timeout" crscore=50 craction=2 crlevel="critical"
```

6. After FortiSandbox finishes scanning the file (typically between one to three minutes), attempt to re-download the file using the same URL and separator (`https://filegen.fortinet.com/v1/sandbox-file?file_name=windows.exe&s=ftnt`).
7. The file is now blocked by the FortiGate, and a *High Security Alert* block page appears in the PC's browser.



8. Verify the AV log again.
   a. In the GUI, go to *Log & Report* > *Security Events* and select the *AntiVirus* card. Select the log entry and click *Details*.



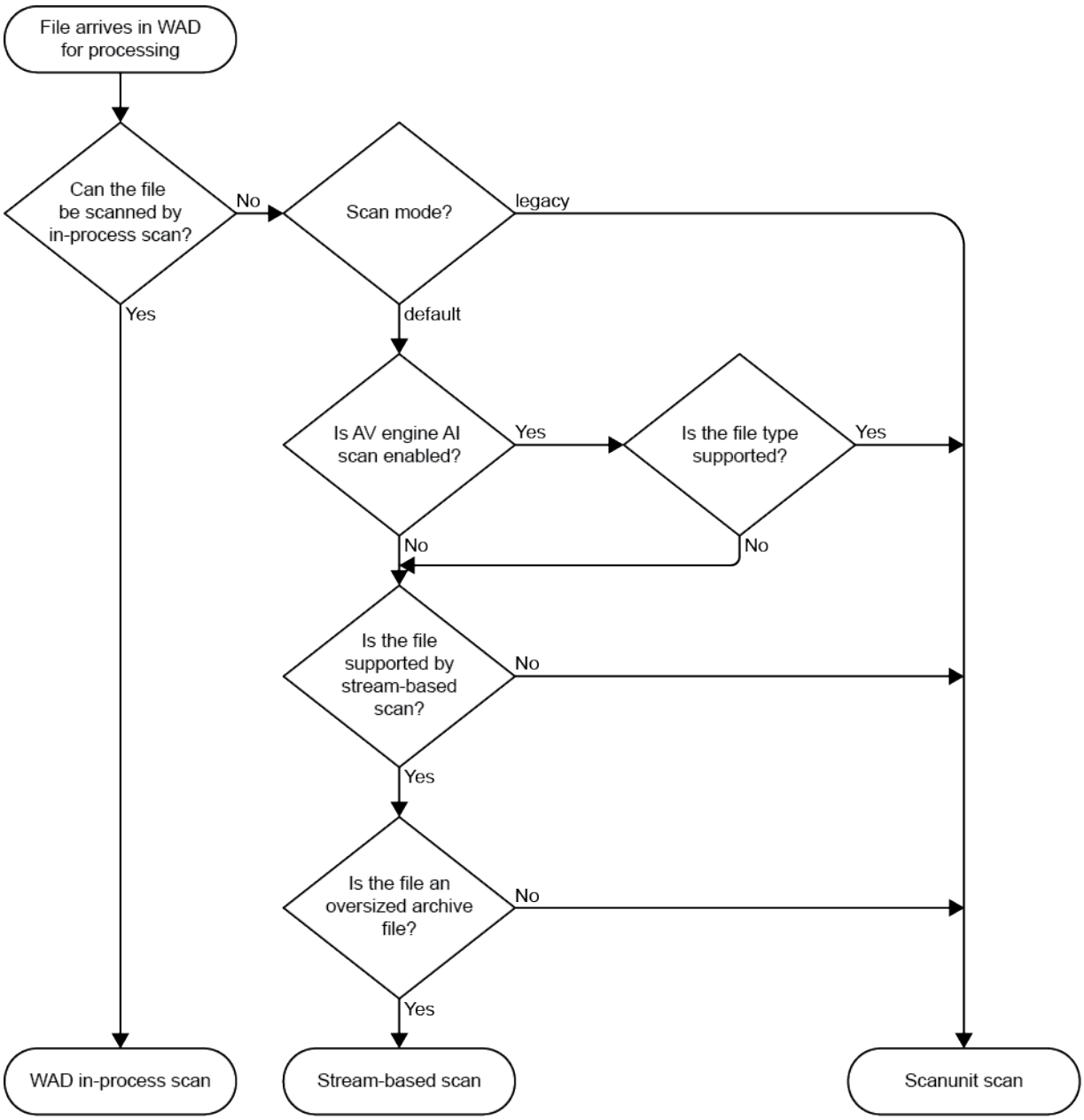
   b. In the CLI, enter the following:

```
# execute log filter category 2
# execute log display
date=2023-11-01 time=11:46:34 eventtime=1698864393856143209 tz="-0700" logid="0211009234"
type="utm" subtype="virus" eventtype="infected" level="warning" vd="root" policyid=1
poluuid="6f2b2dee-478a-51ee-e9c3-b7218be554fe" policytype="policy" msg="File reported
infected by Sandbox." action="blocked" service="HTTPS" sessionid=946497
srcip=192.168.1.110 dstip=149.5.234.147 srcport=57497 dstport=443 srccountry="Reserved"
dstcountry="France" srcintf="internal1" srcintfrole="undefined" dstintf="wan1"
dstintfrole="wan" srcuuid="a051eeb2-5284-51ee-99d0-d8b19cb7439d" dstuuid="6c43f8d6-478a-
51ee-95d8-31177232e869" proto=6 direction="incoming" filename="windows.exe"
```

```
checksum="b0693ac0" quarskip="No-skip" virus="FSA/RISK_MEDIUM" viruscat="Virus"
dtype="fortisandbox" ref="http://www.fortinet.com/ve?vn=FSA%2FRISK_MEDIUM" virusid=9
url="https://filegen.fortinet.com/v1/sandbox-file?file_name=windows.exe&s=ftnt"
profile="default" agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/118.0.0.0 Safari/537.36" httpmethod="GET"
analyticscksum="44c43fc7e1af2071d035bc50871c12bf03f207427ad7094090eb038490ccaa68"
analyticssubmit="false" crscore=10 craction=2 crlevel="medium"
```

Proxy mode stream-based scanning

In proxy mode, AV scanning is processed as follows:



Can the file be scanned by in-process scan?

- This is determined by the WAD daemon.
- In-process scan can be used for simple AV configurations to quickly scan a file without handing it off to another process.
- The following, more complex feature sets cannot be processed by in-process scan:
  - AV engine AI scan
  - DLP
  - Quarantine
  - FortiGuard outbreak prevention, external block list, and EMS external feed
  - Content disarm

Scan mode?

- To configure the scan mode:

```
config antivirus profile
    edit <name>
        set feature-set proxy
        set scan-mode {default | legacy}
    next
end
```

`default`	Enable stream-based scanning (default).
`legacy`	Disable stream-based scanning.

Is AV engine AI scan enabled?

- When enabled, supported files (such as EXE, PDF, and MS Office) are forwarded to the scanunit scan.
- AV engine AI scan is enabled by default. To disable it:

```
config antivirus settings
    set machine-learning-detection disable
end
```

Is the file supported by stream-based scan?

- Stream-based scan supports the following archive file types: ZIP, GZIP, BZIP2, TAR, and ISO (ISO 9660).
- In FortiOS 7.0, stream-based scan is supported in HTTP(S), FTP(S), and SCP/SFTP.
- In FortiOS 6.4 and 6.2, stream-based scan is only supported in HTTP(S).
- Stream-based scan does not support HTTP POST.
- Stream-based scan is not supported when the following features are enabled:
  - DLP
  - Quarantine
  - FortiGuard outbreak prevention, external block list, and EMS external feed
  - Content Disarm
- If a file is not supported, it is buffered and sent to scanunit for scanning.

Is the file an oversized archive file?

- An oversized archive file is a compressed file that is oversized according to the following setting:

```
config firewall profile-protocol-options
    edit <profile>
        config <protocol>
            set oversize-limit <size>
        end
    next
end
```

- If the file is not oversized, it is buffered and sent to scanunit for scanning.

Notes

This feature is not supported on FortiGate models with 2 GB RAM or less. See Proxy-related features not supported on FortiGate 2 GB RAM models on page 126 for more information.

Stream-based scans:

- Are performed with no oversize limits on a best effort basis.
- Can inspect the contents of large archive files without buffering the entire file.
- Decompress and scan the entire archive.
- Can cache infected scan results and clean the scan results (this is enabled by default):

```
config antivirus settings
    set cache-infection-result enable
    set cache-clean-result enable
end
```

Legacy scan mode:

- Used to disable stream-based scanning for troubleshooting purposes.
- Limited by the oversize and uncompressed-oversize limits:

```
config firewall profile-protocol-options
    edit <profile>
        config <protocol>
            set oversize-limit <size>
            set uncompressed-oversize-limit <size>
        end
    next
end
```

TCP windows

Some file transfer applications can negotiate large TCP windows. For example, WinSCP can negotiate an initial TCP window size of about 2 GB.

The TCP window options can be used to prevent overly large initial TCP window sizes, helping avoid channel flow control issues. It allows stream-based scan's flow control to limit peers from sending data that exceeds a policy's configured oversize limit.

To configure TCP window size options:

```
config firewall profile-protocol-options
    edit <string>
        config {http | ftp | ssh | cifs}
            set stream-based-uncompressed-limit <integer>
            set tcp-window-type {auto-tuning | system | static | dynamic}
            set tcp-window-size <integer>
            set tcp-window-minimum <integer>
            set tcp-window-maximum <integer>
        end
    next
end
```

{http | ftp | ssh | cifs}	• http: Configure HTTP protocol options. • ftp: Configure FTP protocol options. • ssh: Configure SFTP and SCP protocol options. • cifs: Configure CIFS protocol options.
stream-based-uncompressed-limit <integer>	The maximum stream-based uncompressed data size that will be scanned, in MB (default = 0 (unlimited)). Stream-based uncompression used only under certain conditions.).
tcp-window-type {auto-tuning | system | static | dynamic}	The TCP window type to use for this protocol. • auto-tuning: Allow the system to auto-tune TCP window size (default). • system: Use the system default TCP window size for this protocol. • static: Manually specify the TCP window size. • dynamic: Vary the TCP window size based on available memory within the limits configured in tcp-window-minimum and tcp-window-maximum.
tcp-window-size <integer>	The TCP static window size (65536 - 16777216, default = 262144). This option is only available when tcp-window-type is static.
tcp-window-minimum <integer>	The minimum TCP dynamic window size (65536 - 1048576, default = 131072). This option is only available when tcp-window-type is dynamic.
tcp-window-maximum <integer>	The maximum TCP dynamic window size (1048576 - 16777216, default = 8388608). This option is only available when tcp-window-type is dynamic.

Flow mode stream-based scanning

In flow mode with AV engine 7.0, FortiOS automatically uses stream-based antivirus scanning for HTML and Javascript files. The AV engine determines the necessary amount of file payload to buffer and scans the partial buffer in certain instances, eliminating the need to cache the entire file, and potentially improving memory usage.

Prior to AV engine 7.0, flow AV operates in a hybrid mode where the IPS engine will attempt an in-process AV scan by default. If the built-in AV engine in the IPS process indicates a full scan is required, the file is sent to the

`scanunit` process for a full scan. In this scenario, the whole file is cached before `scanunit` can begin scanning the file.

With this stream-based AV scanning enhancement, the built-in AV engine in the IPS process can attempt to scan HTML and Javascript files as it buffers the file. This provides better performance and potentially less memory usage overall compared to a full scan.

The full antivirus scanning method is retained for file types and configurations unsupported by stream-based scanning.

The following table summarizes the types of scans and when they are automatically used:

Default antivirus scan	• Automatically uses stream-based scanning in flow mode for HTML and Javascript files. • Automatically scans other supported files using the flow DB. • Triggers a legacy scan for unsupported configurations and file types.

Full antivirus scan	• Automatically used for files types unsupported by default antivirus scans.
	• Automatically used when any of the following antivirus scanning features are enabled:
	  • Machine learning-based malware detection (`set machine-learning-detection`)
	  • Extreme antivirus database (`set use-extreme-db`)
	  • Antivirus PUP/PUA grayware checks
	  • Mobile malware database (`set mobile-malware-db`)
	  • External block list (`set external-blocklist`)
	  • EMS external feed
	  • FortiGuard outbreak prevention
	• Automatically used when any of the following scanning features are used:
	  • Data loss prevention (DLP)
	  • File filter

Example

When the default antivirus scan is used, the AV engine uses stream-based scanning to partially buffer the file and scan it:

```
# diagnose debug application ipsengine 0x1000
diagnose sys scanunit debug all
diagnose debug enable
...
[flav-402] open file size: 68, ftype: 0
[flav-402] flowav config allows quickscan: yes
[flav-402] fsa_enabled=0 fsa_mode=2
[flav-402] heur=0 (0) bz2=1 fsa_ft=0 grayware=0 scantypes: 1
[flav-402] av_flow_write_0 flav_ctx=0x7f35a61a1000, buflen=68, rc = 2
[flav-402] file is infected or suspicious. Wait until file close due to av_exempt enabled
[flav-402] ips_avscan_file_close
[flav-402] [118]: quickscan_close() flav_ctx=0x7f35a61a1000, rc = 2
[flav-402] [118]: cached length 0, flow_bytes 68
```

```
[flav-402] [118]: virus EICAR_TEST_FILE detected!
[flav-402] quickscan_destroy(), flow_writes=1, flow_bytes=68, flav_ctx=0x7f35a61a1000
[flav-402] [118]: quickscan finalized with action 1
[flav-402] ips_avscan_file_close, action=1
[flav-402] ips_avscan_file_destroy
```

The above debug is taken while a user attempts to download an EICAR file. Partial buffering occurred and the file is scanned inside the IPS engine.

When the default antivirus scan (stream-based scanning) cannot be used for a file, the full antivirus scan is used, and the IPS engine buffers the entire file before sending it to `scanunit` for scanning:

```
# diagnose debug application ipsengine 0x1000
# diagnose sys scanunit debug all
# diagnose debug enable
...
[flav-496] [41]: quickscan_close() flav_ctx=0x7fdfc41a1000, rc = -7
[flav-496] [41]: file requires fullscan
[flav-496] attempting switch to fullscan
[flav-496] succesfully switched to fullscan
[flav-496] got FlowAV fullscan request: query_id=41 view_id=3 file_size=12939
[flav-496] quickscan_destroy(), flow_writes=10, flow_bytes=12939, flav_ctx=0x7fdfc41a1000
su 2388 open
```

The Flow AV statistics monitor can be used to view whether the default or full (legacy) scan method was used:

```
# diagnose test app ipsmonitor 24

pid: 23498 from 20240404-09:23:05 to 20240404-09:23:59
av_failopen: enabled
    FlowAV mmap                   : 0
    FlowAV file open              : 0
    FlowAV timeout                : 0
    FlowAV req success            : 0
    FlowAV req fail               : 0
    FlowAV req retry success      : 0
    FlowAV req retry fail         : 0
    FlowAV bypassed scan          : 0
    FlowAV buffer scan            : 0
    FlowAV file scan              : 0
    FlowAV interface file open    : 0
    FlowAV interface file close   : 0
    FlowAV interface file destroy : 0
    FlowAV ignored files          : 0
    FlowAV legacy scan            : 1
    FlowAV default scan           : 1
```

Databases

The antivirus scanning engine uses a virus signatures database to record the unique attributes of each infection. The antivirus scan searches for these signatures and when one is discovered, the FortiGate determines if the file is infected and takes action.

All FortiGates have the normal antivirus signature database. Some models have additional databases that you can use. The database you use depends on your network and security needs, and on your FortiGate model.

The extended virus definitions database is the default setting and provides comprehensive antivirus protection. Entry-level and some mid-range FortiGates cannot support the extreme database. The FortiGate 121G is the lowest model that supports the extreme database. All VMs support the extreme database. The `use-extreme-db` setting is only available on models that support the extreme database.

Extended	This is the default setting. This database includes currently spreading viruses, as determined by the FortiGuard Global Security Research Team, plus recent viruses that are no longer active. These viruses may have been spreading within the last year but have since nearly or completely disappeared.
Extreme	This includes the extended database, plus a large collection of zoo viruses. These are viruses that have not spread in a long time and are largely dormant. Some zoo viruses might rely on operating systems and hardware that are no longer widely used.

To change the antivirus database:

```
config antivirus settings
    set use-extreme-db {enable | disable}
end
```

FortiSandbox database

The *Use FortiSandbox database* setting in the Antivirus profile enables the FortiGate's antivirus engine to receive the latest malware signatures discovered by FortiSandbox that is stored inside FortiSandbox's malware database. By enabling *Use FortiSandbox database*, FortiGate uses these signatures from the malware database along with its existing antivirus signature database for scanning.The antivirus engine scan searches for the malware signature database and antivirus signature database in tandem to check for a match. Once a signature match is discovered, the FortiGate determines if the file is infected and takes action.

The malware signature database supplements the existing antivirus signature database on the FortiGate. This setting is useful if a FortiSandbox solution (either FortiGate Sandbox Cloud, FortiSandbox Cloud, or the FortiSandbox appliance) is deployed.

If you have multiple FortiGates deployed and FortiSandbox is in use, if *Use FortiSandbox database* is enabled in the Antivirus profile, it will enable all FortiGates to download the malware signature database from your FortiSandbox. This can prevent zero-day attacks discovered by the FortiSandbox. FortiSandbox can also be configured to submit its malware signature database with Fortinet Inc. Community by enabling the required *Contribute* settings under your scan profile. See Scan Profile Advanced Tab in the FortiSandbox Administration Guide for information on the scan profile.

FortiGuard labs later release the required submitted signatures in the form of Antivirus updates which can be downloaded by the FortiGates worldwide through FortiGuard updates. See Configuring FortiGuard updates on page 3926.

To enable using the FortiSandbox database in an antivirus profile in the GUI:

1. Go to *Security Profile* > *AntiVirus*.
2. Select the default profile and click *Edit*.
3. Under the *APT Protection Options*, enable *Use FortiSandbox database*.
4. Click *OK* to save the changes.
5. Apply this default profile to the respective firewall policy.

To enable using the FortiSandbox database in an antivirus profile in the CLI:

```
config antivirus profile
    edit “default”
        set analytics-db enable
    next
end
```

It is best practice to keep the `analytics-db` enabled.

To use the antivirus profile in a firewall policy:

```
config firewall policy
    edit 1
        set name "policyid-1"
        set srcintf "lan"
        set dstintf "wan"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set ssl-ssh-profile "certificate-inspection"
        set av-profile "default"
        set nat enable
    next
end
```

Once the Antivirus profile is configured to use the FortiSandbox database and submit files to FortiSandbox, and the antivirus profile is used in a firewall policy, the sharing of malware database from the FortiSandbox to the FortiGate needs to be configured. For information on submitting files to FortiSandbox, see Using FortiSandbox post-transfer scanning with antivirus on page 2267.

The configuration depends on the type of FortiSandbox in use. The table below shows key differences in configuration:

Type of FortiSandbox	Malware database sharing with the FortiGate
FortiSandbox Appliance/FortiSandbox VM (On-Premise)	Enabled using the Global network. See Global Network in the FortiSandbox Administration Guide.

Type of FortiSandbox	Malware database sharing with the FortiGate
FortiSandbox Cloud (PaaS)	Enabled by default.
FortiGate Cloud Sandbox (SaaS)	Enabled by default.

Advanced configurations

This section includes the following:

- Using FortiSandbox post-transfer scanning with antivirus on page 2267
- Using FortiSandbox inline scanning with antivirus on page 2269
- Using FortiNDR inline scanning with antivirus on page 2284
- Malware external feed from EMS on page 2287
- CIFS support on page 2289

Using FortiSandbox post-transfer scanning with antivirus

Antivirus profiles can submit potential zero-day viruses to FortiSandbox for inspection. Based on FortiSandbox's analysis, the FortiGate can supplement its own antivirus database with FortiSandbox's threat intelligence to detect files determined as malicious or suspicious. This augments the FortiGate antivirus with zero-day detection.

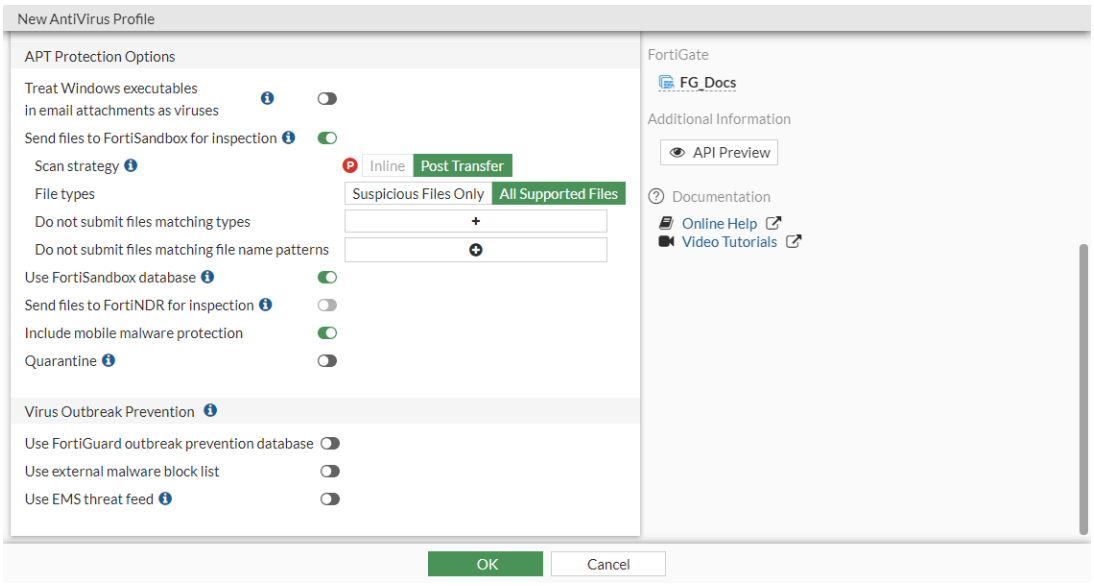
FortiSandbox can be used with antivirus in both proxy-based and flow-based inspection modes. The FortiGate first examines the file for any known viruses. When a match is found, the file is tagged as known malware. If no match is found, the files are forwarded to FortiSandbox using the following options:

- *All Supported Files*: all files matching the file types defined in the scan profile of the FortiSandbox are forwarded.
- *Suspicious Files Only*: files classified by the antivirus as having any possibility of active content are forwarded to FortiSandbox. When using FortiGate Cloud Sandbox, we recommend selecting this option due to its submission limits.
- *None*: files are not forwarded to FortiSandbox.

For more information, see Configuring sandboxing on page 4354.

To enable FortiSandbox inspection in an antivirus profile:

1. Go to *Security Profiles* > *AntiVirus*.
2. Create, edit, or clone an antivirus profile.
3. In the *APT Protection Options* section, set *Send Files to FortiSandbox for Inspection* to either *Suspicious Files Only* or *All Supported Files*.
4. Optionally, for *Do not submit files matching types*, click the *+* to exclude certain file types from being sent to FortiSandbox.
5. Optionally, for *Do not submit files matching file name patterns*, click the *+* to enter a wildcard pattern to exclude files from being sent to FortiSandbox.



6. Enable *Use FortiSandbox Database*.
7. Click *OK*.

FortiGate diagnostics

To view the detection count:

```
# diagnose test application quarantined 7
Total: 0

Statistics:
        vfid: 0, detected: 2, clean: 1252, risk_low: 6, risk_med: 2, risk_high: 1, limit_reached:0
```

To verify the address is configured correctly:

```
# diagnose test application quarantined 1
…
fortisandbox-fsb1 is enabled: analytics, realtime=yes, taskfull=no
addr=172.18.52.154/514, source-ip=0.0.0.0, keep-alive=no. ssl_opt=3, hmac_alg=0
…
```

To run the diagnostics for real-time debugging:

```
# diagnose debug application quarantined -1
# diagnose debug enable
```

To check the FortiGate Cloud server status:

```
# diagnose test application forticldd 3
…
```

```
    Active APTServer status:  up
```

To view FortiGate Cloud Sandbox submission statistics for advanced debugging:

```
# diagnose test application quarantined 2
```

FortiSandbox diagnostics

To run the OFTP debug for advanced debugging:

```
# diagnose-debug device <client serial number>
```

Using FortiSandbox inline scanning with antivirus

FortiSandbox inline scanning can be used in proxy and flow mode. When inline scanning is enabled, the client's file is held while it is sent to FortiSandbox for inspection. During this time, the FortiGate may apply client comforting (see Protocol options on page 2101). For example, leaking a certain amount of bytes at a certain time interval to the client. Once a verdict is returned, the appropriate action (allow or block) is performed on the held file. If there is an error connecting to the FortiSandbox or a timeout on the FortiSandbox scanning the file, the file can be passed, logged, or blocked based on FortiGate's configuration. For more information, see Configuring sandboxing on page 4354.

When using FortiSandbox inline scanning in flow mode, you can utilize sandboxing alongside other flow-mode features, such as IPS; streamlining security operations by improving threat detection capabilities without the need to switch to proxy mode.

This topic describes how to configure the FortiSandbox inline scanning as follows:

- FortiSandbox appliance inline scanning on page 2269
- Configuring proxy-mode FortiSandbox inline scanning on page 2271
- FortiGuard AI-based Inline Malware Prevention Service on page 2274
- FortiSandbox scanning error and timeout actions on page 2279
- Verifying flow-mode FortiSandbox inline scanning for HTTP actions on page 2279
- Verifying flow-mode FortiSandbox inline scanning for FTP actions on page 2281
- Verifying flow-mode FortiSandbox inline scanning for POP3 actions on page 2282

FortiSandbox appliance inline scanning

Inline scanning requires a FortiSandbox appliance running version 4.2 or later, and the FortiSandbox must be reachable by port 4443. See Understanding Inline Block feature in the FortiSandbox Best Practices for more information.

FortiSandbox inline scanning is disabled by default. FortiSandbox inline scanning is best used in conjunction with AV engine scanning since there is a higher rate of detection by using both at the same time.

To enable FortiSandbox inline scanning:

```
config system fortisandbox
    set status enable
    set inline-scan {enable | disable}
    set server <fortisandbox_server_ip>
end
```

To configure the FortiSandbox scanning options in an antivirus profile:

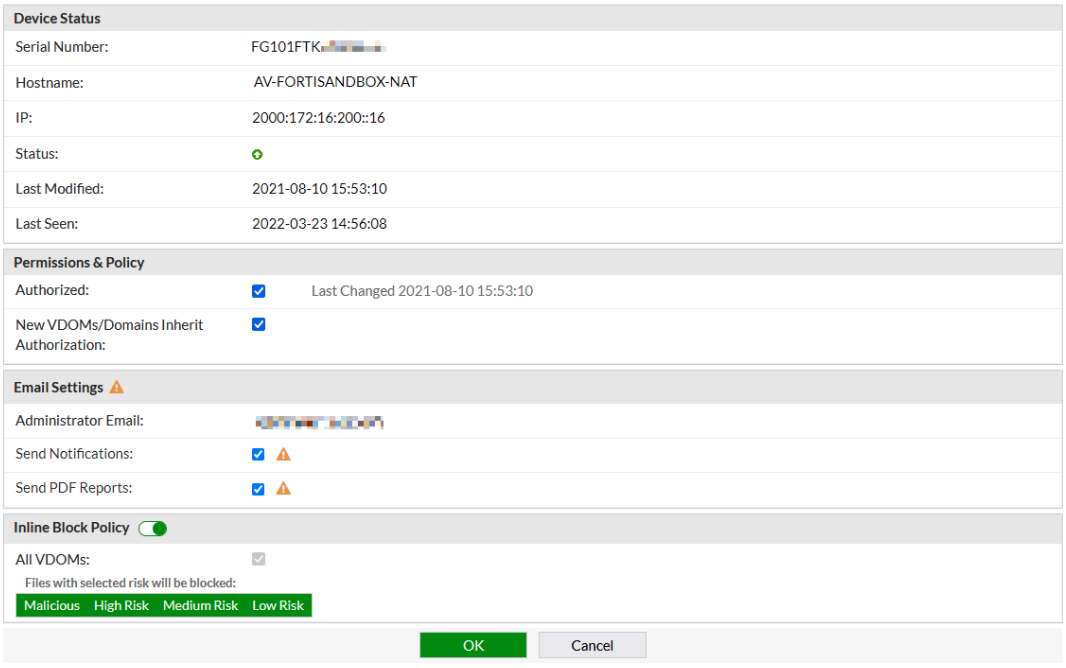
```
config antivirus profile
    edit <name>
        set feature-set {proxy | flow}
        set fortisandbox-mode {inline | analytics-suspicious | analytics-everything}
        set fortisandbox-error-action {ignore | log-only | block}
        set fortisandbox-timeout-action {ignore | log-only | block}
        set fortisandbox-max-upload <integer>
        set fortisandbox-scan-timeout <integer>
        config {http | ftp | imap | pop3 | smtp | mapi | cifs | ssh}
            set av-scan {disable | block | monitor}
            set fortisandbox {disable | block | monitor}
        end
    next
end
```

Command	Description
`feature-set {proxy | flow}`	Set the inline scanning mode type.
`fortisandbox-mode {inline | analytics-suspicious | analytics-everything}`	Set the FortiSandbox scan mode: • `inline`: FortiSandbox inline scanning • `analytics-suspicious`: FortiSandbox post-transfer scanning; submit supported files if heuristics or other methods determine they are suspicious • `analytics-everything`: FortiSandbox post-transfer scanning; submit supported files and known infected files (default)
`fortisandbox-error-action {ignore | log-only | block}`	Set the action to take if FortiSandbox inline scanning encounters an error reaching the FortiSandbox: • `ignore`: Allow the file to pass, do not log the error. • `log-only`: Allow the file to pass, but log the connection error. (Default) • `block`: Block the file if a sandboxing error occurs.
`fortisandbox-timeout-action {ignore | log-only | block}`	Set the action to take if FortiSandbox inline scanning encounters a scan timeout: • `ignore`: Allow the file to pass, do not log the timeout. • `log-only`: Allow the file to pass, but log the timeout event. (Default) • `block`: Block the file if the scan times out.
`fortisandbox-max-upload <integer>`	Set the maximum size of files that can be uploaded to FortiSandbox (1 - 396, default = 10).

Command	Description
`fortisandbox-scan-timeout <integer>`	FortiSandbox inline scan timeout, in seconds (30 - 180, default = 60). For on-premise Sandbox and FortiSandbox Cloud, setting `fortisandbox-scan-timeout` value lower than the timeout setting on the remote device may result in session timeout. Lowering `fortisandbox-scan-timeout` from the default `60` seconds may result in session timeout when connecting to FortiGate Cloud Sandbox.
`av-scan {disable | block | monitor}`	Enable the antivirus scan service. Set to `block` or `monitor` to work with FortiSandbox (default = `disable`).
`fortisandbox {disable | block | monitor}`	Set the protocol level parameter for FortiSandbox file scanning: • `disable` (default), `block`, and `monitor` are available for inline scanning • `disable` (default) and `monitor` are available for post-transfer scanning

Configuring proxy-mode FortiSandbox inline scanning

This example assumes that *Inline Block Policy* is already enabled in FortiSandbox for the FortiGate with selected risk levels (see FortiGate devices in the FortiSandbox Administration Guide for more information). The inline block policy in this example blocks all risk levels: malicious, high risk, medium risk, and low risk.



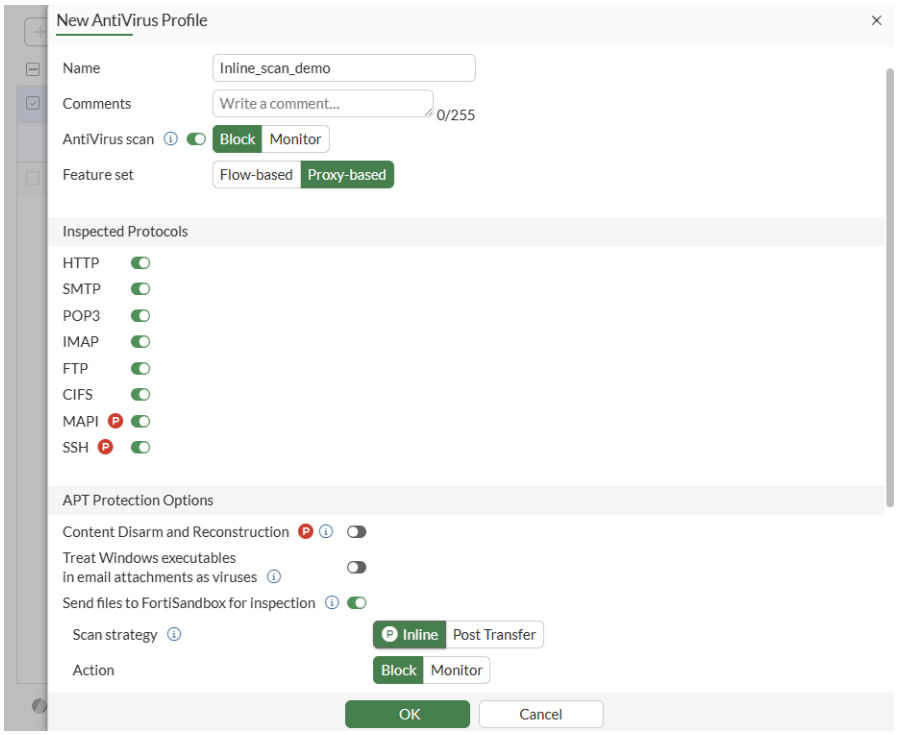
To configure proxy-mode FortiSandbox inline scanning in the GUI:

1. Enable FortiSandbox inline scanning globally:

```
config system fortisandbox
    set status enable
    set inline-scan enable
```

```
    set server "172.18.70.76"
end
```

2. Configure the antivirus profile:
   a. Go to *Security Profiles* > *AntiVirus* and click *Create New*.
   b. Set the *Feature set* to *Proxy-based*.
   c. Enable the protocols to inspect.
   d. Enable *AntiVirus scan* and set it to *Block*.
   e. Enable *Send files to FortiSandbox* for inspection and set the *Action* to *Block*. The *Scan strategy* appears as *Inline* because it was configured in the CLI.



   f. Click *OK*.

To configure proxy-mode FortiSandbox inline scanning in the CLI:

1. Enable FortiSandbox inline scanning globally:

```
config system fortisandbox
    set status enable
    set inline-scan enable
    set server "172.18.70.76"
end
```

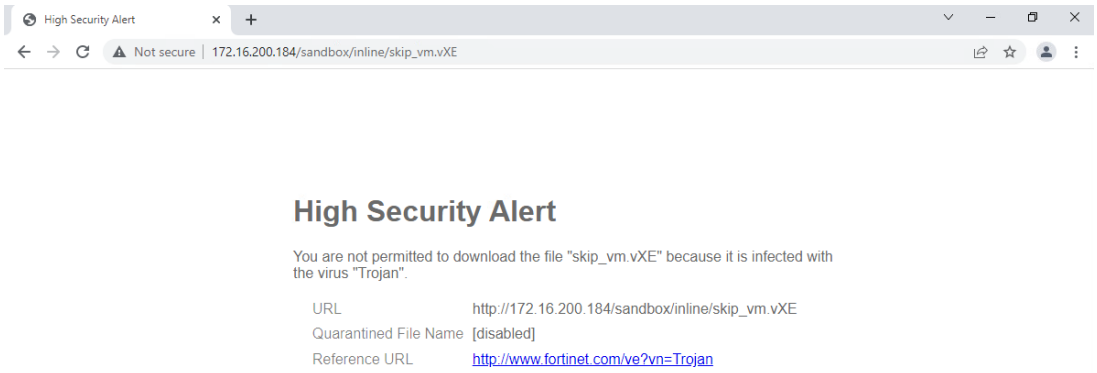
2. Configure the antivirus profile:

```
config antivirus profile
    edit "Inline_scan_demo"
        set feature-set proxy
```

```
        set fortisandbox-mode inline
        config http
            set av-scan block
            set fortisandbox block
        end
        config ftp
            set av-scan block
            set fortisandbox block
        end
        config imap
            set av-scan block
            set fortisandbox block
        end
        config pop3
            set av-scan block
            set fortisandbox block
        end
        config smtp
            set av-scan block
            set fortisandbox block
        end
        config mapi
            set av-scan block
            set fortisandbox block
        end
        config cifs
            set av-scan block
            set fortisandbox block
        end
        config ssh
            set av-scan block
            set fortisandbox block
        end
    next
end
```

To verify that infected files are blocked inline:

1. On a client, open a web browser and download an infected file.
2. The file is held while being scanned by FortiSandbox. Once FortiSandbox determines that file's risk level is not tolerated by the inline block policy, the FortiGate drops the connection and displays a replacement message that the file cannot be downloaded.



3. In FortiOS, view the antivirus log.
   - In the GUI, go to *Log & Report* > *Security Events* and click the *AntiVirus* card.
   - In the CLI:

```
# execute log filter category 2
# execute log display
1 logs found.
1 logs returned.

1: date=2022-03-23 time=16:19:37 eventtime=1648077577156255080 tz="-0700"
logid="0210008232" type="utm" subtype="virus" eventtype="fortisandbox" level="warning"
vd="vdom1" policyid=1 poluuid="9170ca3e-aade-51ec-772b-1d31f135fe26" policytype="policy"
msg="Blocked by FortiSandbox." action="blocked" service="HTTP" sessionid=10545
srcip=10.1.100.181 dstip=172.16.200.184 srcport=37046 dstport=80 srccountry="Reserved"
dstcountry="Reserved" srcintf="port1" srcintfrole="undefined" dstintf="port9"
dstintfrole="undefined" srcuuid="5b426c60-aade-51ec-f020-b3d334ba18d3" dstuuid="5b426c60-
aade-51ec-f020-b3d334ba18d3" proto=6 direction="incoming" filename="skip_vm.vXE"
quarskip="File-was-not-quarantined" virus="Trojan" viruscat="Unknown" dtype="fortisandbox"
ref="http://www.fortinet.com/ve?vn=Trojan" virusid=0
url="http://172.16.200.184/sandbox/inline/skip_vm.vXE" profile="Inline_scan_demo"
agent="curl/7.68.0" httpmethod="GET" analyticssubmit="false" fsaaction="deny"
fsaseverity="high-risk" fsaverdict="block" fsafileid=0 fsafiletype="exe" crscore=50
craction=2 crlevel="critical
```

FortiGuard AI-based Inline Malware Prevention Service

Inline scanning is supported when the FortiGate is licensed with the FortiGuard AI-based Inline Malware Prevention Service (IL MPS). It works similar to inline scanning for the FortiSandbox appliance, where the file that is being processed is first checked locally by the FortiGate. If it requires further analysis, the file is sent to the IL MPS for inspection. The file is submitted along with the following, additional information:

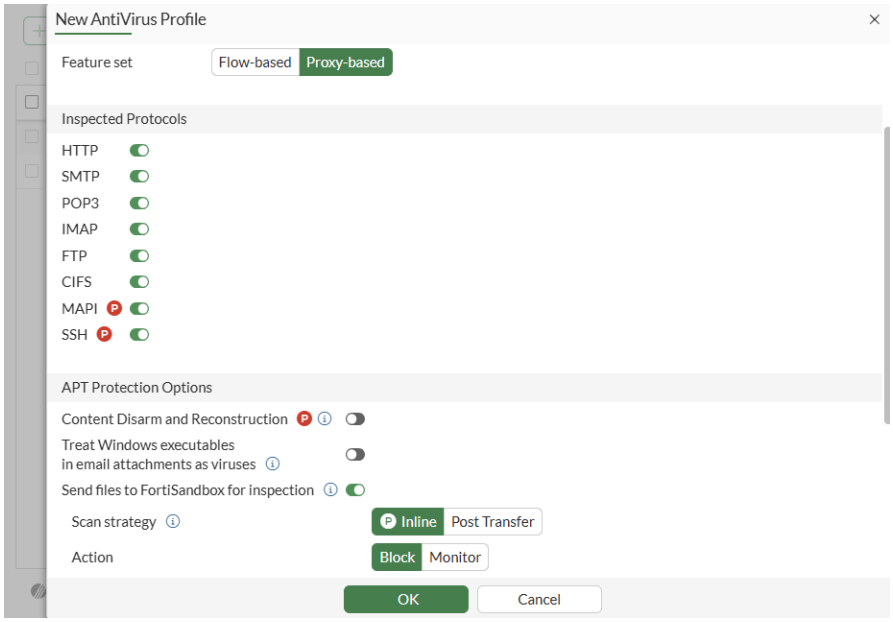
- Device configuration (VDOM, service)
- Username
- Device identifiers
- IP address
- Suspicious URL
- Suspicious files

The file and the above additional information remain in the FortiSandbox where they are evaluated by its multi-layered process leveraging its contained AI model. The information is not shared outside the FortiSandbox service with other large learning models. After the rating of either suspicious or clean is sent back to the FortiGate, the cloud-based service keeps the file, collected information, and derived information based on the retention period of 3 days and 60 days for files rated as Clean and Suspicious, respectively.

Inline scanning is supported for FortiSandbox appliances, FortiNDR, and IL MPS. On a FortiGate, only a single inline scanning type can be configured at one time.

To configure IL MPS scanning in the GUI:

1. Enable the FortiGate Cloud feature visibility:
   a. Go to *System* > *Feature Visibility*.
   b. In the *Additional Features* section, enable *FortiGate Cloud Sandbox*.
   c. Click *Apply*.
2. Configure the *Sandbox* Fabric connector:
   a. Go to *Security Fabric* > *Fabric Connectors* and double-click the *Sandbox* card.
   b. In the *Settings* tab, set the *Type* to *FortiGate Cloud*.
   c. Select a *Region*.
   d. Enable *Inline scan*.
   e. Click *OK*.
3. Configure the antivirus profile:
   a. Go to *Security Profiles* > *AntiVirus* and click *Create New*.
   b. Set the *Feature set* to *Proxy-based*.
   c. Enable the protocols to inspect.
   d. Enable *Send files to FortiSandbox* for inspection.
   e. Set the *Scan strategy* to *Inline*, and set the *Action* to *Block*.



   f. Click *OK*.

To configure IL MPS scanning in the CLI:

1. Disable FortiSandbox appliance and FortiSandbox Cloud:

```
config system fortisandbox
    set status disable
end
```

2. Configure FortiGate Cloud Sandbox:

```
# execute forticloud-sandbox region
0  Global
1  Europe
2  Japan
3  US
Please select cloud sandbox region[0-3]:0
Cloud sandbox region is selected: Global
```

3. Enable inline scanning for FortiGate Cloud:

```
config system fortiguard
    set sandbox-region "Global"
    set sandbox-inline-scan enable
end
```

4. Configure the antivirus profile:

```
config antivirus profile
    edit "av"
        set feature-set proxy
        set fortisandbox-mode inline
        config http
            set fortisandbox block
        end
        config ftp
            set fortisandbox block
        end
        config imap
            set fortisandbox block
        end
        config pop3
            set fortisandbox block
        end
        config smtp
            set fortisandbox block
        end
        config mapi
            set fortisandbox block
        end
        config cifs
            set fortisandbox block
        end
        config ssh
```

```
            set fortisandbox block
        end
        set scan-mode default
    next
end
```

To verify that infected files are blocked inline:

1. On a client, open a web browser and download an infected file using HTTP.
2. The file is held while being scanned by FortiGate Cloud Sandbox. Once FortiGate Cloud Sandbox determines that file's risk level is not tolerated, the FortiGate drops the connection and displays a replacement message that the file cannot be downloaded.
3. Verify the antivirus log:

```
# execute log display
1 logs found.
1 logs returned.

1: date=2022-07-12 time=16:31:26 eventtime=1657668686245018328 tz="-0700" logid="0210008232"
type="utm" subtype="virus" eventtype="fortisandbox" level="warning" vd="vdom1" policyid=1
poluuid="54c06312-01fd-51ed-0db5-10c9586a0c2e" policytype="policy" msg="Blocked by
FortiSandbox." action="blocked" service="HTTP" sessionid=19934 srcip=10.1.100.191
dstip=172.16.200.194 srcport=51688 dstport=80 srccountry="Reserved" dstcountry="Reserved"
srcintf="port1" srcintfrole="undefined" dstintf="port9" dstintfrole="undefined"
srcuuid="1cb467b6-01fd-51ed-8abf-72abd959c0d0" dstuuid="1cb467b6-01fd-51ed-8abf-72abd959c0d0"
proto=6 direction="incoming" filename="skip_vm.vXE" quarskip="Quarantine-disabled"
virus="Unknown" viruscat="Trojan" dtype="fortisandbox"
ref="http://www.fortinet.com/ve?vn=Unknown" virusid=0
url="http://172.16.200.194/sandbox/inline/skip_vm.vXE" profile="av" agent="curl/7.68.0"
httpmethod="GET" analyticssubmit="false" fsaaction="deny" fsaverdict="block"
fsaseverity="high-risk" fsafileid=0 fsafiletype="exe" crscore=50 craction=2 crlevel="critical"
```

To verify that infected files are monitored:

1. Edit the antivirus profile to monitor files over HTTP:

```
config antivirus profile
    edit "av"
        set feature-set proxy
        set fortisandbox-mode inline
        config http
            set fortisandbox monitor
        end
    next
end
```

2. On a client, open a web browser and download an infected file using HTTP.
3. Verify the antivirus log:

```
# execute log display
1 logs found.
```

```
1 logs returned.

1: date=2022-07-12 time=16:34:25 eventtime=1657668865371976563 tz="-0700" logid="0210008233"
type="utm" subtype="virus" eventtype="fortisandbox" level="notice" vd="vdom1" policyid=1
poluuid="54c06312-01fd-51ed-0db5-10c9586a0c2e" policytype="policy" msg="Detected by
FortiSandbox." action="monitored" service="HTTP" sessionid=20002 srcip=10.1.100.191
dstip=172.16.200.194 srcport=51724 dstport=80 srccountry="Reserved" dstcountry="Reserved"
srcintf="port1" srcintfrole="undefined" dstintf="port9" dstintfrole="undefined"
srcuuid="1cb467b6-01fd-51ed-8abf-72abd959c0d0" dstuuid="1cb467b6-01fd-51ed-8abf-72abd959c0d0"
proto=6 direction="incoming" filename="skip_vm.vXE" quarskip="Quarantine-disabled"
virus="Unknown" viruscat="Trojan" dtype="fortisandbox"
ref="http://www.fortinet.com/ve?vn=Unknown" virusid=0
url="http://172.16.200.194/sandbox/inline/skip_vm.vXE" profile="av" agent="curl/7.68.0"
httpmethod="GET" analyticssubmit="false" fsaaction="deny" fsaverdict="block"
fsaseverity="high-risk" fsafileid=0 fsafiletype="exe" crscore=50 craction=2 crlevel="critical"
```

To verify that infected files are blocked inline if a scan timeout occurs:

1. Edit the antivirus profile to block files over HTTP and when there is a scan timeout:

```
config antivirus profile
    edit "av"
        set feature-set proxy
        set fortisandbox-mode inline
        config http
            set fortisandbox block
        end
        set fortisandbox-timeout-action block
    next
end
```

2. On a client, open a web browser and download a large ZIP file (clean file).
3. When the scan timeout occurs, a replacement message appears that *The file "zipfile.zip" is still being scanned and will be released once complete. Please try the transfer again in a few minutes.*
4. Verify the antivirus log:

```
# execute log display
1 logs found.
1 logs returned.

1: date=2022-07-12 time=16:44:51 eventtime=1657669491697816069 tz="-0700" logid="0210008236"
type="utm" subtype="virus" eventtype="fortisandbox" level="warning" vd="vdom1" policyid=1
poluuid="54c06312-01fd-51ed-0db5-10c9586a0c2e" policytype="policy" msg="FortiSandbox scan
timeout." action="blocked" service="HTTP" sessionid=20258 srcip=10.1.100.191
dstip=172.16.200.194 srcport=51830 dstport=80 srccountry="Reserved" dstcountry="Reserved"
srcintf="port1" srcintfrole="undefined" dstintf="port9" dstintfrole="undefined"
srcuuid="1cb467b6-01fd-51ed-8abf-72abd959c0d0" dstuuid="1cb467b6-01fd-51ed-8abf-72abd959c0d0"
proto=6 direction="incoming" filename="zipfile.zip" quarskip="Quarantine-disabled"
url="http://172.16.200.194/sandbox/zipfile.zip" profile="av" agent="curl/7.68.0"
httpmethod="GET" analyticssubmit="false" fsaaction="timeout" fsafileid=0 crscore=50 craction=2
crlevel="critical"
```

5. After a few minutes, download the ZIP file again.
6. When the scan is complete on the FortiSandbox side, the file is downloaded and no log is generated because the scan deemed that the file is clean.

FortiSandbox scanning error and timeout actions

In this example, the HTTP protocol settings for `av-scan` and `fortisandbox` in the AV profile are both set to `block`. All files traversing HTTP in this configuration are scanned by the AV engine first, and then by FortiSandbox inline scanning for further file analysis. Based on the FortiSandbox results, FortiOS will take the appropriate action.

Files can be blocked if they contain a scan error or timeout. The scan timeout is configured in FortiSandbox and set to 50 seconds. If the file scan takes longer than 50 seconds, FortiSandbox returns a timeout to the FortiGate, and file is dropped with the current configuration. If a user tries to download the same file again, the cached result is provided by FortiSandbox to the FortiGate based on the previous file scan.

This example assumes FortiSandbox inline scanning has been configured globally. The FortiGate will block the file if there is an inline scanning error or timeout.

To configure the antivirus profile to block files if there is an inline scanning error or timeout:

```
config antivirus profile
    edit "av"
        set feature-set proxy
        set fortisandbox-mode inline
        config http
            set av-scan block
            set fortisandbox block
        end
        set fortisandbox-error-action block
        set fortisandbox-timeout-action block
    next
end
```

If the administrator decides to take more risk and scan all files traversing HTTP, but log or ignore an inline scanning error or timeout, the profile is modified as follows:

```
config antivirus profile
    edit "av"
        set fortisandbox-error-action {log-only | ignore}
        set fortisandbox-timeout-action {log-only | ignore}
    next
end
```

The AV engine is still used first, followed by FortiSandbox inline scanning. The FortiGate will log or ignore the file if there is an inline scanning error or timeout, and the file is allowed to pass through.

Verifying flow-mode FortiSandbox inline scanning for HTTP actions

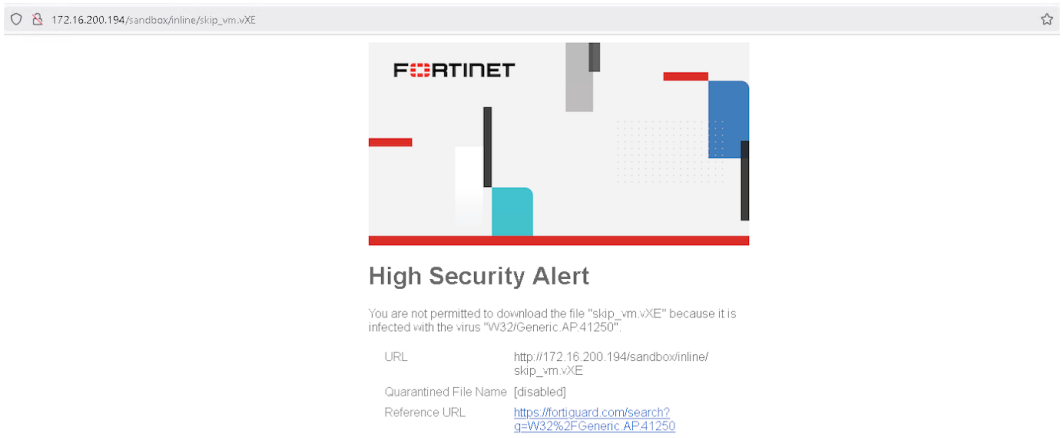
In the following example, FortiSandbox inline scanning in flow mode is set to block HTTP actions. Downloading a malicious file sample on HTTP is blocked.

To block HTTP actions through FortiSandbox inline scanning in flow mode:

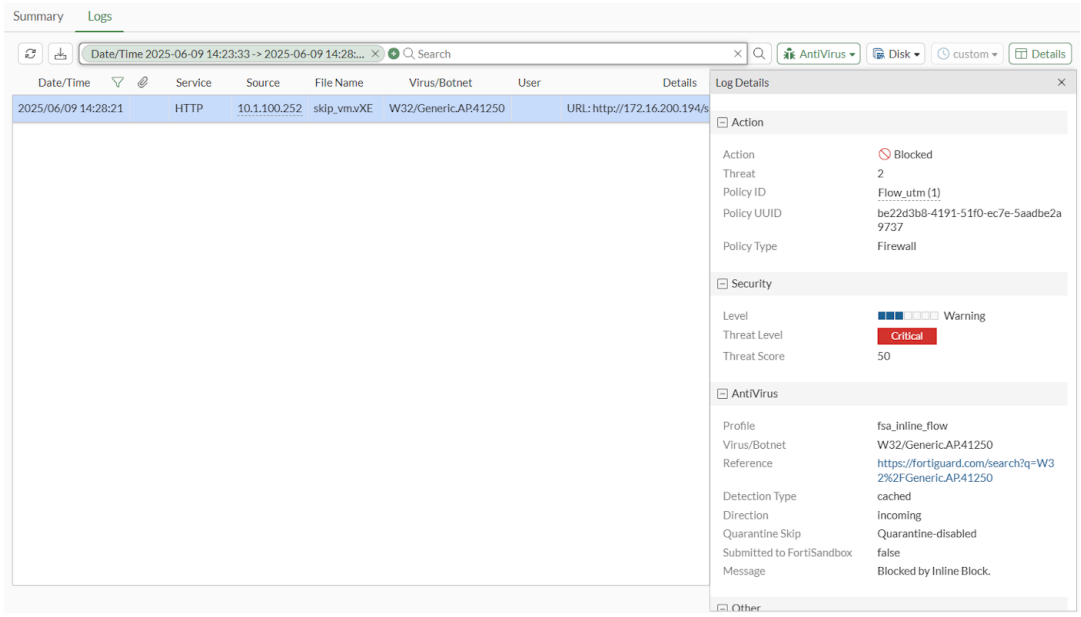
1. Configure FortiSandbox inline scanning in flow mode:

```
config antivirus profile
    edit "fsa_inline_flow"
        set fortisandbox-mode inline
        config http
            set fortisandbox block
        end
    next
end
```

2. Attempt to download a malicious file sample on HTTP. The action is blocked.



3. View the logs in the GUI:



4. View the logs in the CLI:

```
# execute log display

1:date=2025-06-09 time=14:24:09 eventtime=1749504249131410853 tz="-0700" logid="0209008220"
type="utm" subtype="virus" eventtype="inline-block" level="warning" vd="vdom1" policyid=1
poluuid="be22d3b8-4191-51f0-ec7e-5aadbe2a9737" policytype="policy" msg="Blocked by Inline
Block." action="blocked" service="HTTP" sessionid=93035 srcip=10.1.100.252
dstip=172.16.200.194 srcport=56671 dstport=80 srccountry="Reserved" dstcountry="Reserved"
srcintf="port1" srcintfrole="undefined" dstintf="port9" dstintfrole="undefined"
srcuuid="79ed46d8-4191-51f0-3dd9-dc7cd3922e47" dstuuid="79ed46d8-4191-51f0-3dd9-dc7cd3922e47"
proto=6 direction="incoming" filename="skip_vm.vXE" filetype="exe" quarskip="Quarantine-
disabled" virus="W32/Generic.AP.41250" viruscat="Unknown" dtype="cached" itype="inline-block"
ref="https://fortiguard.com/search?q=W32%2FGeneric.AP.41250"
url="http://172.16.200.194/sandbox/inline/skip_vm.vXE" profile="fsa_inline_flow"
agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:135.0) Gecko/20100101 Firefox/135.0"
httpmethod="GET" analyticssubmit="false" icbaction="deny" icbverdict="block"
icbseverity="malicious"
icbfileid="ab5d58debc11e8c78dac5097267d472927f1d185cb21ffbedd00db9d6049c8d3" icbfiletype="exe"
crscore=50 craction=2 crlevel="critical"
```

Verifying flow-mode FortiSandbox inline scanning for FTP actions

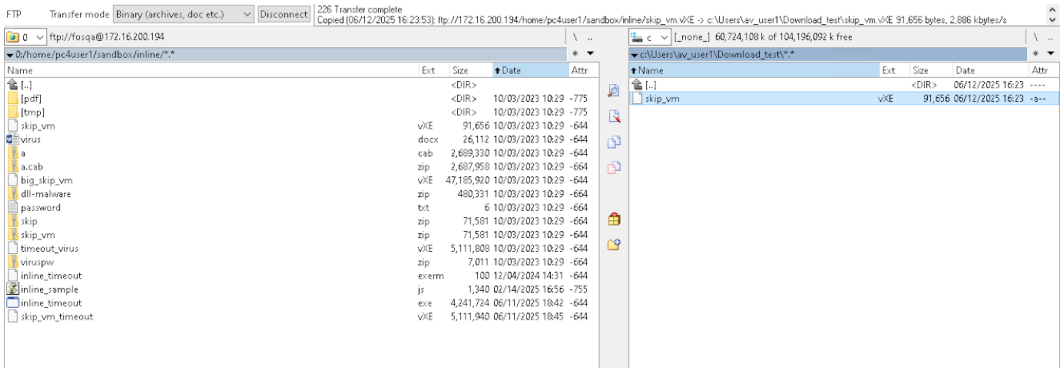
In the following example, FortiSandbox inline scanning in flow mode is set to monitor FTP actions. Uploading a malicious file sample on FTP is monitored.

To monitor FTP actions through FortiSandbox inline scanning in flow mode:

1. Configure FortiSandbox inline scanning in flow mode:

```
config antivirus profile
    edit "fsa_flow_inline"
        set fortisandbox-mode inline
        config ftp
            set fortisandbox monitor
        end
    next
end
```

2. Attempt to upload a malicious file sample on FTP. The action is monitored.



3. View the logs:

```
# execute log display

1: date=2025-06-12 time=16:19:56 eventtime=1749770396462744989 tz="-0700" logid="0209008221"
type="utm" subtype="virus" eventtype="inline-block" level="notice" vd="vdom1" policyid=1
poluuid="f820fc12-f07d-51ef-10a7-9ce92adafc22" policytype="policy" msg="Detected by Inline
Block." action="monitored" service="FTP" sessionid=1305 srcip=172.16.200.194
dstip=10.1.100.252 srcport=20 dstport=65463 srccountry="Reserved" dstcountry="Reserved"
psrcport=65453 pdstport=21 srcintf="port3" srcintfrole="lan" dstintf="port2" dstintfrole="lan"
srcuuid="eb7a7402-4489-51ef-4afd-ca286e50ca04" dstuuid="eb7a7402-4489-51ef-4afd-ca286e50ca04"
proto=6 direction="incoming" filename="skip_vm.vXE" filetype="exe" quarskip="Quarantine-
disabled" virus="W32/Generic.AP.41250" viruscat="NotApplicable" dtype="fortisandbox"
itype="inline-block" ref="https://fortiguard.com/search?q=W32%2FGeneric.AP.41250"
profile="fsa_flow_inline" analyticssubmit="false" icbaction="deny" icbverdict="block"
icbseverity="malicious"
icbfileid="ab5d58debc11e8c78dac5097267d472927f1d185cb21ffbedd00db9d6049c8d3" icbfiletype="exe"
crscore=50 craction=2 crlevel="critical"
```

Verifying flow-mode FortiSandbox inline scanning for POP3 actions

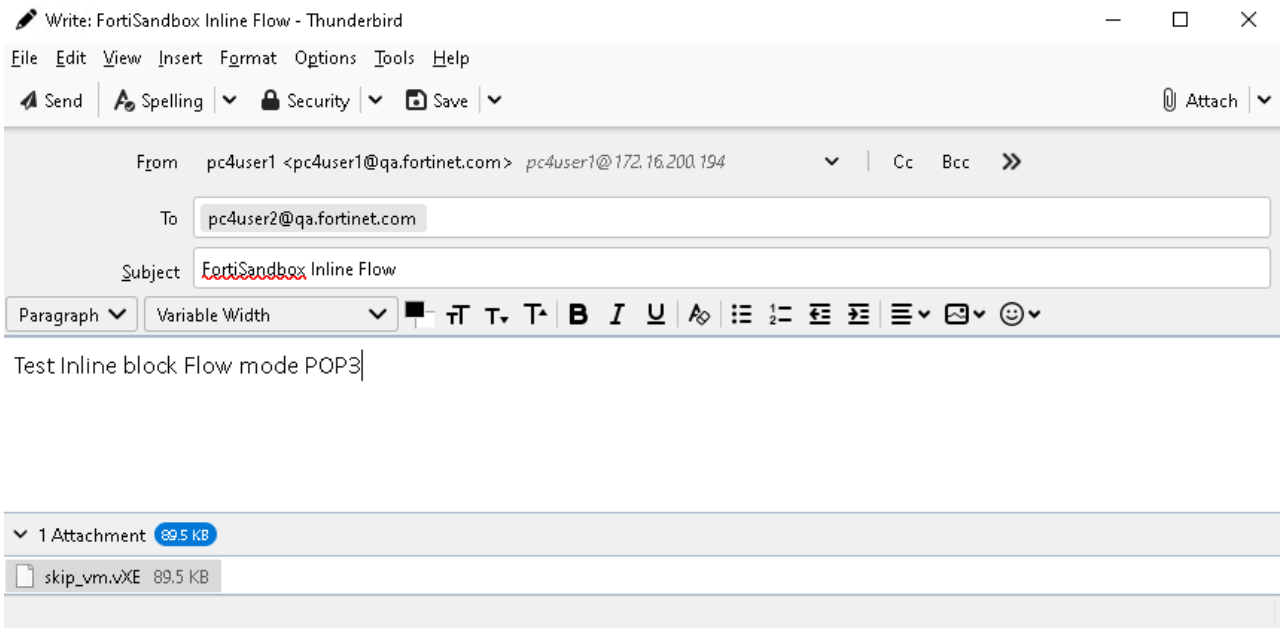
In the following example, FortiSandbox inline scanning in flow mode is set to block POP3 actions. Downloading a malicious file sample on POP3 is blocked.

To block POP3 actions through FortiSandbox inline scanning in flow mode:

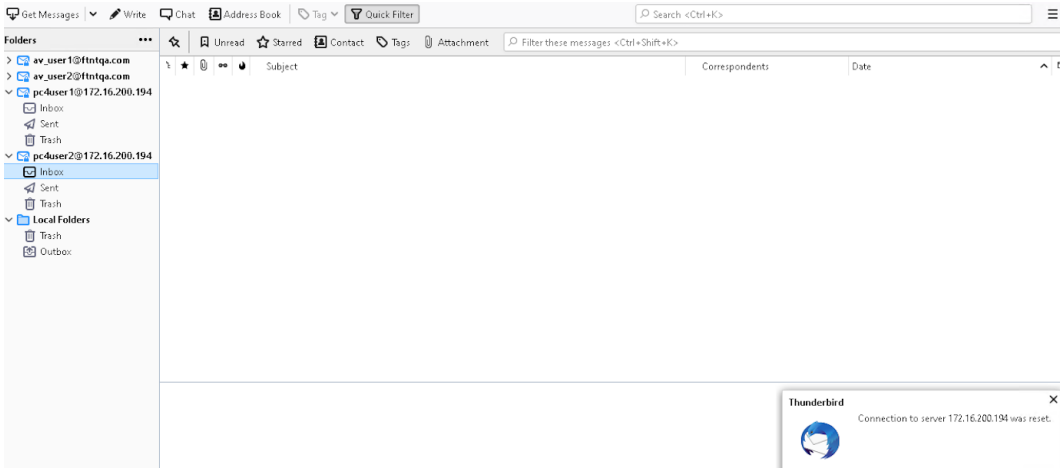
1. Configure FortiSandbox inline scanning in flow mode:

```
config antivirus profile
    edit "fsa_inline_flow"
        set fortisandbox-mode inline
        config pop3
            set fortisandbox block
        end
    next
end
```

2. A malicious file is emailed to the client on POP3.



The email is blocked.



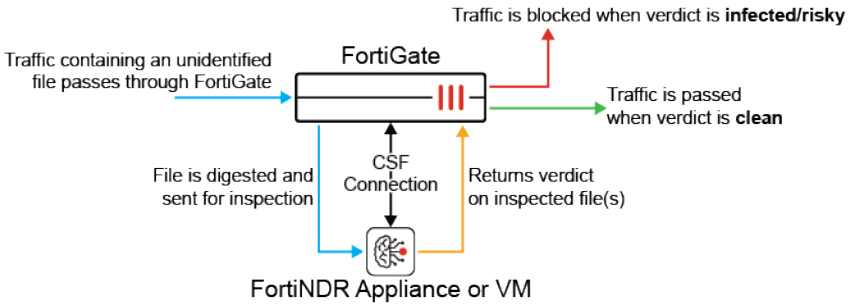
3. View the logs:

```
# execute log display

1: date=2025-06-12 time=17:02:06 eventtime=1749772926320332178 tz="-0700" logid="0209008220"
type="utm" subtype="virus" eventtype="inline-block" level="warning" vd="vdom1" policyid=1
poluuid="f820fc12-f07d-51ef-10a7-9ce92adafc22" policytype="policy" msg="Blocked by Inline
Block." action="blocked" service="POP3S" sessionid=3036 srcip=10.1.100.252
dstip=172.16.200.194 srcport=49642 dstport=110 srccountry="Reserved" dstcountry="Reserved"
srcintf="port2" srcintfrole="lan" dstintf="port3" dstintfrole="lan" srcuuid="eb7a7402-4489-
51ef-4afd-ca286e50ca04" dstuuid="eb7a7402-4489-51ef-4afd-ca286e50ca04" proto=6
direction="incoming" filename="skip_vm.vXE" filetype="exe" quarskip="Quarantine-disabled"
virus="W32/Generic.AP.41250" viruscat="Unknown" dtype="fortisandbox" itype="inline-block"
ref="https://fortiguard.com/search?q=W32%2FGeneric.AP.41250" profile="fsa_inline_flow"
from="pc4user1@qa.fortinet.com" to="pc4user2@qa.fortinet.com" messageid="<a9de312a-91a8-3340-
0f2e-4b19f0238e50@qa.fortinet.com>" analyticssubmit="false" icbaction="deny"
```

```
icbverdict="block" icbseverity="malicious"
icbfileid="ab5d58debc11e8c78dac5097267d472927f1d185cb21ffbedd00db9d6049c8d3" icbfiletype="exe"
crscore=50 craction=2 crlevel="critical"
```

Using FortiNDR inline scanning with antivirus

FortiNDR can be used with antivirus profiles in proxy inspection mode (flow mode is currently not supported). FortiNDR inspects high-risk files and issues a verdict to the firewall based on how close the file features match those of malware. When enabled, FortiNDR can log, block, ignore, or monitor (allow) the file based on the verdict.



A licensed FortiNDR appliance is required to use this feature. Check the FortiNDR release notes for support information.

To configure FortiNDR inline inspection with an AV profile:

1. Configure FortiNDR to join a Security Fabric in FortiOS (see Configuring FortiNDR on page 4387).
2. On the FortiOS, enable inline inspection:

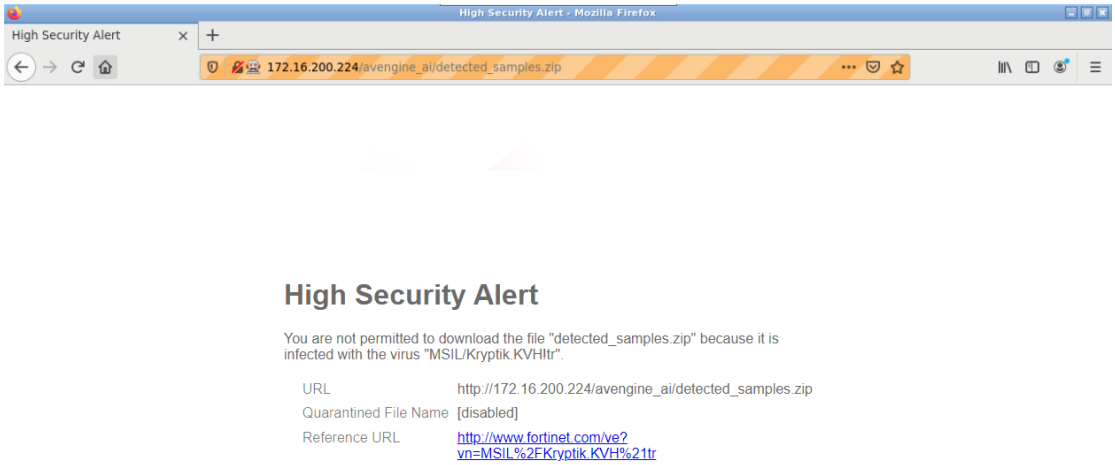
```
config system fortindr
    set status enable
end
```

3. Configure an antivirus profile to use inline inspection and block detected infections:

```
config antivirus profile
    edit "av"
        set feature-set proxy
        config http
            set fortindr block
        end
        config ftp
            set fortindr block
        end
        config imap
            set fortindr block
        end
        config pop3
            set fortindr block
```

```
        end
        config smtp
            set fortindr block
        end
        config mapi
            set fortindr block
        end
        config nntp
            set fortindr block
        end
        config cifs
            set fortindr block
        end
        config ssh
            set fortindr block
        end
    next
end
```

4. Add the antivirus profile to a firewall policy. When potential infections are blocked by FortiNDR inline inspection, a replacement message appears (see Replacement messages on page 3899 for more information). An infection blocked over HTTP looks similar to the following:



Sample log

```
date=2021-04-29 time=15:12:07 eventtime=1619734327633022960 tz="-0700" logid="0209008221"
type="utm" subtype="virus" eventtype="fortindr" level="notice" vd="vdom1" policyid=1 msg="Detected
by FortiNDR." action="monitored" service="HTTP" sessionid=13312 srcip=10.1.100.221
dstip=172.16.200.224 srcport=50792 dstport=80 srcintf="wan2" srcintfrole="wan" dstintf="wan1"
dstintfrole="wan" proto=6 direction="incoming" filename="detected_samples.zip" quarskip="File-was-
not-quarantined" virus="MSIL/Kryptik.KVH!tr" dtype="fortindr"
ref="http://www.fortinet.com/ve?vn=MSIL%2FKryptik.KVH%21tr" virusid=0
url="http://172.16.200.224/avengine_ai/detected_samples.zip" profile="av" agent="curl/7.68.0"
analyticssubmit="false" crscore=50 craction=2 crlevel="critical"
```

FortiNDR inline inspection with other AV inspection methods

The following inspection logic applies when FortiNDR inline inspection is enabled simultaneously with other AV inspection methods. The AV engine inspection and its verdict always takes precedence because of performance. The actual behavior depends on which inspected protocol is used.

HTTP, FTP, SSH, and CIFS protocols:

1. AV engine scan; AV database and FortiSandbox database (if applicable).
   a. FortiNDR inline inspection occurs simultaneously.
2. AV engine machine learning detection for WinPE PUPs (potentially unwanted programs).
   a. FortiNDR inline inspection occurs simultaneously.
3. Outbreak prevention and external hash list resources.
   a. FortiNDR inline inspection occurs simultaneously.

If any AV inspection method returns an infected verdict, the FortiNDR inspection is aborted.

POP3, IMAP, SMTP, NNTP, and MAPI protocols:

1. AV engine scan; AV database and FortiSandbox database (if applicable).
2. AV engine machine learning detection for WinPE PUPs (potentially unwanted programs).
   a. FortiNDR inline inspection occurs simultaneously.
3. Outbreak prevention and external hash list resources.
   a. FortiNDR inline inspection occurs simultaneously.

In an AV profile, use `set fortindr-error-action {log-only | block | ignore}` to configure the action to take if FortiNDR encounters an error.

Accepted file types

The following file types are sent to FortiNDR for inline inspection:

7Z	HTML	RTF
ARJ	JS	TAR
BZIP	LZH	VBA
BZIP2	LZW	VBS
CAB	MS Office documents (XML and non-XML)	WinPE (EXE)
ELF	PDF	XZ
GZIP	RAR	ZIP

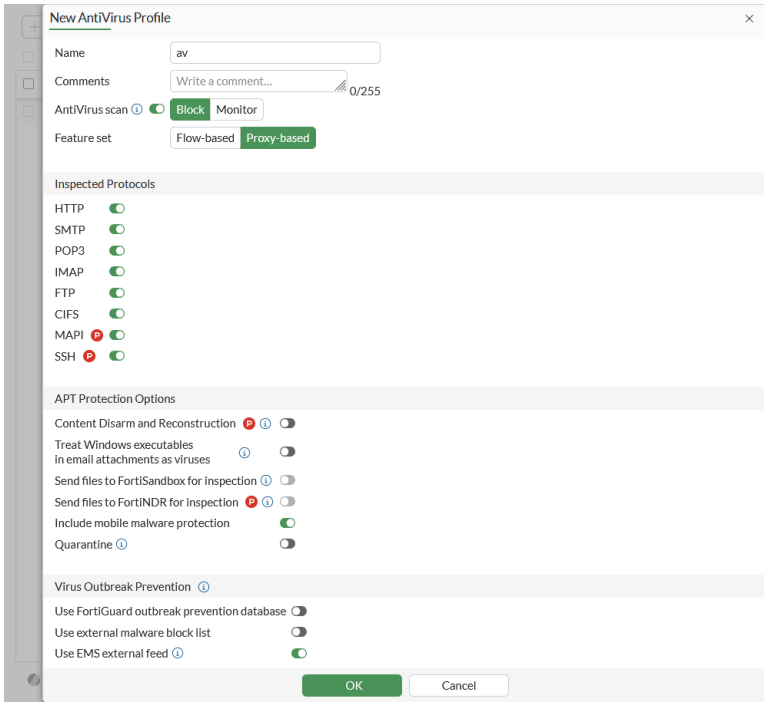
Malware external feed from EMS

A FortiGate can pull malware external feeds from FortiClient EMS, which in turn receives malware hashes detected by FortiClients. The malware hash can be used in an antivirus profile when AV scanning is enabled with block or monitor actions. This feature is supported in proxy and flow mode.

If an external malware blocklist and the FortiGuard outbreak prevention database are also enabled in the antivirus profile, the checking order is: AV local database, EMS external feed, external malware blocklist, FortiGuard outbreak prevention database. If the EMS external feed and external malware blocklist contain the same hash value, then the EMS infection will be reported if both of them are blocked.

To configure an EMS external feed in an antivirus profile in the GUI:

1. Enable the EMS external feed:
   a. Go to *Security Fabric* > *Fabric Connectors* and double-click the *FortiClient EMS* card.
   b. Enable *EMS external feed* for an EMS server.
   c. Configure the other settings if needed (see Configuring FortiClient EMS on page 4322 for more details).
   d. Click *OK*.
2. Create the antivirus profile:
   a. Go to *Security Profiles* > *AntiVirus* and click *Create New*.
   b. In the *Virus Outbreak Prevention* section, enable *Use EMS external feed*.
   c. Configure the other settings as needed.



   d. Click *OK*.

To configure an EMS external feed in an antivirus profile in the CLI:

1. Enable the EMS external feed:

```
config endpoint-control fctems
    edit 2
        set status enable
        set name "WIN10-EMS"
        set server "192.168.20.10"
        set https-port 443
        set source-ip 0.0.0.0
        set pull-sysinfo enable
        set pull-vulnerabilities enable
        set pull-tags enable
        set pull-malware-hash enable
        unset capabilities
        set call-timeout 30
        set websocket-override disable
    next
end
```

2. Create the antivirus profile:

```
config antivirus profile
    edit "av"
        config http
            set av-scan block
        end
        config ftp
            set av-scan block
        end
        config imap
            set av-scan block
        end
        config pop3
            set av-scan block
        end
        config smtp
            set av-scan block
        end
        config cifs
            set av-scan block
        end
        set external-blocklist-enable-all enable
        set ems-threat-feed enable
    next
end
```

Sample log

```
# execute log filter category utm-virus
# execute log display
```

```
1: date=2021-03-19 time=16:06:46 eventtime=1616195207055607417 tz="-0700" logid="0208008217"
type="utm" subtype="virus" eventtype="ems-threat-feed" level="notice" vd="vd1" policyid=1
msg="Detected by EMS threat feed." action="monitored" service="HTTPS" sessionid=1005
srcip=10.1.100.24 dstip=172.16.200.214 srcport=54674 dstport=443 srcintf="port2"
srcintfrole="undefined" dstintf="port1" dstintfrole="undefined" proto=6 direction="incoming"
filename="creditcardSSN.pdf" quarskip="Quarantine-disabled" virus="Email scan" dtype="File Hash"
filehash="22466078c2d52dfd5ebbbd6c4207ddec6ac61aa82f960dc54cfbc83b8eb42ed1" filehashsrc="test"
url="https://172.16.200.214/hash/creditcardSSN.pdf" profile="av" agent="curl/7.68.0"
analyticssubmit="false" crscore=10 craction=2 crlevel="medium"
```

```
2: date=2021-03-19 time=16:06:13 eventtime=1616195173832494609 tz="-0700" logid="0208008216"
type="utm" subtype="virus" eventtype="ems-threat-feed" level="warning" vd="vd1" policyid=1
msg="Blocked by EMS threat feed." action="blocked" service="HTTPS" sessionid=898 srcip=10.1.100.24
dstip=172.16.200.214 srcport=54672 dstport=443 srcintf="port2" srcintfrole="undefined"
dstintf="port1" dstintfrole="undefined" proto=6 direction="incoming" filename="BouncingButton.pdf"
quarskip="Quarantine-disabled" virus="Email scan" dtype="File Hash"
filehash="a601431acd5004c37bf8fd02fccfdacbb54b27c8648d1d41ad14fa3eaf8651d3" filehashsrc="test"
url="https://172.16.200.214/hash/BouncingButton.pdf" profile="av" agent="curl/7.68.0"
analyticssubmit="false" crscore=10 craction=2 crlevel="medium"
```

CIFS support

Antivirus scanning on Common Internet File System (CIFS) traffic is supported in flow-based and proxy-based inspection. The file filter profile handles the configuration of file filtering on CIFS. The antivirus profile handles the antivirus configuration for CIFS scanning.

File filtering for CIFS is performed by inspecting the first 4 KB of the file to identify the file's magic number. If a match occurs, CIFS file filtering prevents the CIFS command that contains that file from running. The file filter functions differently for un-encrypted and encrypted CIFS traffic:

- For un-encrypted CIFS traffic, the standalone file filter works in flow and proxy mode.
- For encrypted CIFS traffic, the CIFS profile must be enabled in the firewall policy because the SMB server's credential settings are still be configured in CIFS profile. Using the standalone file filter only works in proxy mode.

For a CIFS profile to be available for assignment in a policy, the policy must use proxy inspection mode. See Proxy mode inspection on page 2213 for details. Note that in proxy inspection mode, special condition archive files (encrypted, corrupted, mailbomb, and so on) marked by the antivirus engine are blocked automatically.

Messages that are compressed with LZNT1, LZ77, and LZ77+Huffman algorithms can be scanned in proxy mode.

To configure file-type filtering and antivirus scanning on CIFS traffic:

1. Configure a CIFS domain controller on page 2290
2. Configure a CIFS profile on page 2290
3. Configure an antivirus profile on page 2292

Configure a CIFS domain controller

The domain controller must be configured when CIFS traffic is encrypted. The configuration tells the FortiGate the network location of the domain controller and the superuser credentials.

For FortiGate to retrieve the domain information, the user needs to grant Replicating Directory Changes permissions in the Domain Controller (DC). See How to grant the "Replicating Directory Changes" permission for the Microsoft Metadirectory Services ADMA service account for more information.

To configure the CIFS domain controller:

```
config user domain-controller
    edit "SERVER_NAME"
        set hostname "host"
        set domain-name "EXAMPLE.COM"
        set username "admin-super"
        set password *********
        set ip 172.16.201.40
    next
end
```

Configure a CIFS profile

To create a CIFS profile, configure the server credential type and create a file filter profile.

Set the CIFS server credential type

The CIFS server credential type can be none, `credential-replication`, or `credential-keytab`.

none

The CIFS profile assumes the CIFS traffic is unencrypted. This is the default value.

```
config firewall profile-protocol-options
    edit "cifs"
        config cifs
            set server-credential-type none
        end
    next
end
```

credential-replication

To decrypt CIFS traffic, FortiOS obtains the session key from the domain controller by logging in to the superuser account. The domain controller must be configured.

```
config firewall profile-protocol-options
    edit "cifs"
```

```
        config cifs
            set server-credential-type credential-replication
            set domain-controller "SERVER_NAME"
        end
    next
end
```

Variable	Description
domain-controller <string>	The previously configured domain to decrypt CIFS traffic for.

credential-keytab

To decrypt CIFS traffic, FortiOS uses a series of keytab values. This method is used when the SMB connection is authenticated by Kerberos. Keytab entries must be configured, and are stored in FortiOS in plaintext.

```
config firewall profile-protocol-options
    edit "cifs"
        config cifs
            set server-credential-type credential-keytab
            config server-keytab
                edit "keytab1"
                    set keytab
"BQIAAABFAAEAC0VYQU1QTEUuQ09NAAdleGFtcGxlAAAAAVUmAlwBABIAILdV5P6NXT8RrTvapcMJQxDYCjRQiD0BzxhwS9h0V
gyM"
                next
            end
        end
    next
end
```

Variable	Description
keytab <keytab>	Base64 encoded keytab file containing the credentials of the server.

Configure CIFS file filtering

Multiple rules can be added to a file filter profile. See File filter on page 2518.

To configure a file filter for CIFS traffic:

```
config file-filter profile
    edit "cifs"
        set comment "block zip files on unencrypted cifs traffic"
        set feature-set flow
        set replacemsg-group ''
        set log enable
        config rules
            edit "rule1"
                set protocol cifs
                set action block
```

```
                set direction any
                set password-protected any
                set file-type zip
            next
        end
    next
end
```

Variable	Description
comment <string>	A brief comment describing the entry.
feature-set {flow | proxy}	Flow or proxy mode feature set (default = `flow`).
replacemsg-group <string>	Replacement message group.
log {enable | disable}	Enable/disable file filter logging (default = `enable`).
scan-archive-contents [enable | disable]	Enable/disable scanning of archive contents (default = `enable`).
protocol {http ftp smtp imap pop3 mapi cifs ssh}	Filter based on the specified protocol(s).
action {log-only | block}	The action to take for matched files: • `log-only`: Allow the content and write a log message (default). • `block`: Block the content and write a log message.
direction {incoming | outgoing | any}	Match files transmitted in the session's originating (`incoming`) and/or reply (`outgoing`) direction (default = `any`).
password-protected [yes | any]	Match only password-protected files (`yes`) or any file (default = `any`).
file-type <file_type>	The file types to be matched. See Supported file types on page 2520 for details.

Configure an antivirus profile

The antivirus profile handles the antivirus configuration for CIFS scanning.

To configure an antivirus profile:

```
config antivirus profile
    edit "av"
        ...
        config cifs
            set av-scan {disable | block | monitor}
            set outbreak-prevention {disable | block | monitor}
            set external-blocklist {disable | block | monitor}
            set quarantine {enable | disable}
            set archive-block {encrypted corrupted partiallycorrupted multipart nested mailbomb
fileslimit timeout unhandled}
            set archive-log {encrypted corrupted partiallycorrupted multipart nested mailbomb
```

```
fileslimit timeout unhandled}
            set emulator {enable | disable}
        end
    next
end
```

Variable	Description
av-scan	Enable antivirus scan service: • `disable`: Disable (default). • `block`: Block the virus infected files. • `monitor`: Log the virus infected files.
outbreak-prevention {disable | block | monitor}	Enable the virus outbreak prevention service: • `disable`: Disable (default). • `block`: Block the matched files. • `monitor`: Log the matched files.
external-blocklist {disable | block | monitor}	Enable the external blocklist: • `disable`: Disable (default). • `block`: Block the matched files. • `monitor`: Log the matched files.
quarantine {enable | disable}	Enable/disable quarantine for infected files (default = `disable`).
archive-block {encrypted corrupted partiallycorrupted multipart nested mailbomb fileslimit timeout unhandled}	Select the archive types to block: • `encrypted`: Block encrypted archives. • `corrupted`: Block corrupted archives. • `partiallycorrupted`: Block partially corrupted archives. • `multipart`: Block multipart archives. • `nested`: Block nested archives. • `mailbomb`: Block mail bomb archives. • `fileslimit`: Block exceeded archive files limit. • `timeout`: Block scan timeout. • `unhandled`: Block archives that FortiOS cannot open.
archive-log {encrypted corrupted partiallycorrupted multipart nested mailbomb fileslimit timeout unhandled}	Select the archive types to log: • `encrypted`: Log encrypted archives. • `corrupted`: Log corrupted archives. • `partiallycorrupted`: Log partially corrupted archives. • `multipart`: Log multipart archives. • `nested`: Log nested archives. • `mailbomb`: Log mail bomb archives. • `fileslimit`: Log exceeded archive files limit. • `timeout`: Log scan timeout. • `unhandled`: Log archives that FortiOS cannot open.
emulator {enable | disable}	Enable/disable the virus emulator (default = `enable`).

Log samples

File-type detection events generated by CIFS profiles are logged in the `utm-file-filter` log category. Antivirus detection over the CIFS protocol generates logs in the `utm-virus` category. See the FortiOS Log Message Reference for more information.

Logs generated by CIFS profile file filter:

```
date=2024-01-02 time=10:47:30 eventtime=1704221249421998774 tz="-0800" logid="1900064001"
type="utm" subtype="file-filter" eventtype="file-filter" level="notice" vd="root" policyid=1
poluuid="f4fe48a4-938c-51ee-8856-3e84e3b24af4" policytype="policy" sessionid=3985
srcip=13.13.13.13 srcport=49481 srccountry="United States" srcintf="port2" srcintfrole="undefined"
srcuuid="d2f06fda-15e7-51ee-0d22-faaf5170dad2" dstip=104.21.235.207 dstport=443 dstcountry="United
States" dstintf="port1" dstintfrole="undefined" dstuuid="d2f06fda-15e7-51ee-0d22-faaf5170dad2"
proto=6 service="HTTPS" profile="default" direction="incoming" action="log-only"
url="https://png2.cleanpng.com/dy/transparent.png" hostname="png2.cleanpng.com" agent="Mozilla/5.0
(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0
Safari/537.36" httpmethod="GET" referralurl="https://www.cleanpng.com/" rulename="2"
filename="transparent.png" filesize=803672 filetype="png" msg="File was detected by file filter."
```

```
date=2024-01-02 time=10:52:01 eventtime=1704221520563734347 tz="-0800" logid="1900064000"
type="utm" subtype="file-filter" eventtype="file-filter" level="warning" vd="root" policyid=1
poluuid="f4fe48a4-938c-51ee-8856-3e84e3b24af4" policytype="policy" sessionid=4371
srcip=13.13.13.13 srcport=49639 srccountry="United States" srcintf="port2" srcintfrole="undefined"
srcuuid="d2f06fda-15e7-51ee-0d22-faaf5170dad2" dstip=104.21.235.207 dstport=443 dstcountry="United
States" dstintf="port1" dstintfrole="undefined" dstuuid="d2f06fda-15e7-51ee-0d22-faaf5170dad2"
proto=6 service="HTTPS" profile="default" direction="incoming" action="blocked"
url="https://png2.cleanpng.com/dy/realistic.png" hostname="png2.cleanpng.com" agent="Mozilla/5.0
(Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/120.0.0.0
Safari/537.36" httpmethod="GET" referralurl="https://www.cleanpng.com/" rulename="2"
filename="realistic.png" filesize=3371220 filetype="png" msg="File was blocked by file filter."
```

Logs generated by AV profile for infections detected over CIFS:

```
date=2019-04-09 time=15:19:02 logid="0204008202" type="utm" subtype="virus" eventtype="outbreak-
prevention" level="warning" vd="vdom1" eventtime=1554848342519005401 msg="Blocked by Virus
Outbreak Prevention service." action="blocked" service="SMB" sessionid=177 srcip=10.1.100.11
dstip=172.16.200.44 srcport=37444 dstport=445 srcintf="wan2" srcintfrole="wan" dstintf="wan1"
dstintfrole="wan" policyid=1 proto=6 direction="incoming" filename="outbreak\\zhvo_test.com"
quarskip="File-was-not-quarantined." virus="503e99fe40ee120c45bc9a30835e7256fff3e46a" dtype="File
Hash" filehash="503e99fe40ee120c45bc9a30835e7256fff3e46a" filehashsrc="fortiguard" profile="av"
analyticssubmit="false" crscore=50 craction=2 crlevel="critical"
```

```
date=2019-04-09 time=15:18:59 logid="0211008192" type="utm" subtype="virus" eventtype="infected"
level="warning" vd="vdom1" eventtime=1554848339909808987 msg="File is infected." action="blocked"
service="SMB" sessionid=174 srcip=10.1.100.11 dstip=172.16.200.44 srcport=37442 dstport=445
srcintf="wan2" srcintfrole="wan" dstintf="wan1" dstintfrole="wan" policyid=1 proto=6
direction="incoming" filename="sample\\eicar.com" quarskip="File-was-not-quarantined."
virus="EICAR_TEST_FILE" dtype="Virus" ref="http://www.fortinet.com/ve?vn=EICAR_TEST_FILE"
virusid=2172 profile="av"
```

```
analyticscksum="275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f"
analyticssubmit="false" crscore=50 craction=2 crlevel="critical"
```

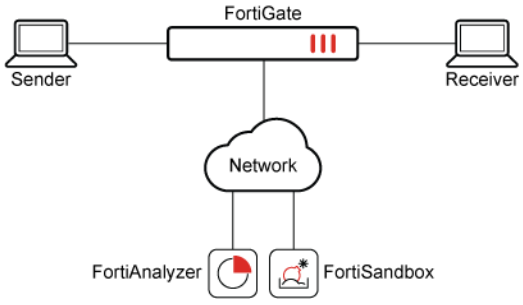
Configuration examples

This section includes the following examples:

- Content disarm and reconstruction on page 2295
- FortiGuard outbreak prevention on page 2297
- External malware block list on page 2306
- Exempt list for files based on individual hash on page 2309
- Downloading quarantined files in archive format on page 2310

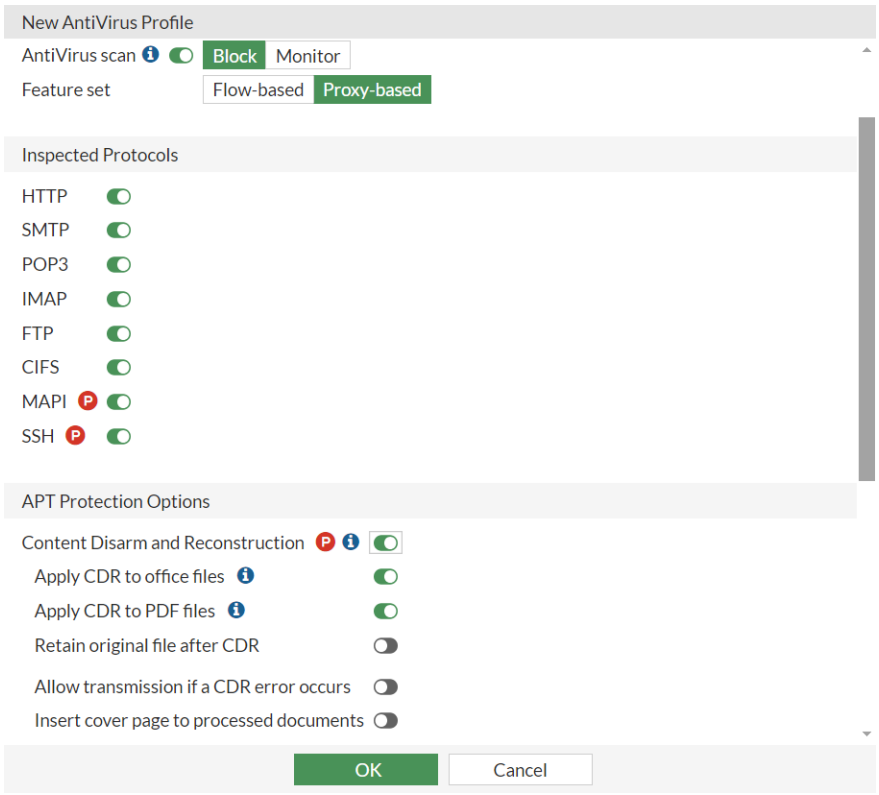
Content disarm and reconstruction

In this example, a Microsoft Office document, such as a Microsoft OneNote file, with an embedded hyperlink (that redirects to an external website) is sent to the receiver. When the user receives the file, the hyperlink in the document is deactivated. See Content disarm and reconstruction on page 2243 for more information.



To configure CDR:

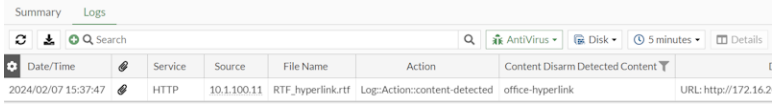
1. Go to *Security Profiles* > *AntiVirus*.
2. Edit an antivirus profile, or create a new one.
3. Set *Feature set* to *Proxy-based*.
4. Under *Inspected Protocols*, enable one or more protocols to inspect.
5. Enable *AntiVirus scan*.
6. Under *APT Protection Options*, enable *Content Disarm and Reconstruction*.



7. Enable *Retain original file after CDR*, and select a quarantine location from the available options:

Quarantine locally	Saves the original document file to disk (if possible) or a connected FortiAnalyzer based on the FortiGate log settings (`config log fortianalyzer setting`).
Send to FortiSandbox	Saves the original document file to a connected FortiSandbox.

   When *Retain original file after CDR* is disabled, the original document file is discarded.
8. Click *OK*.
9. Go to *Log & Report > Security Events* to view CDR events in Antivirus logs.
   In the following example, an AntiVirus log file includes a CDR event for an embedded hyperlink in an RTF file.



To edit the CDR detection parameters:

By default, stripping of all active Microsoft Office and PDF content types are enabled. In this example, stripping macros in Microsoft Office documents will be disabled.

```
config antivirus profile
    edit av
        config content-disarm
            set office-macro disable
            set detect-only {enable | disable}
```

```
            set cover-page {enable | disable}
        end
    next
end
```

Where:

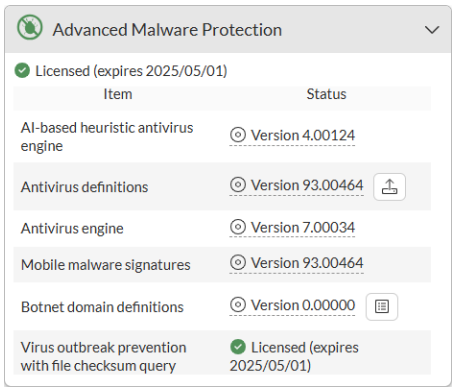
`detect-only`	Only detect disarmable files, do not alter content. Disabled by default.
`cover-page`	Attach a cover page to the file's content when the file has been processed by CDR. Enabled by default.

FortiGuard outbreak prevention

This example demonstrates how to enable FortiGuard Virus Outbreak Protection Service (VOS). See Virus outbreak prevention on page 2243 for more information.

To verify FortiGuard antivirus license information:

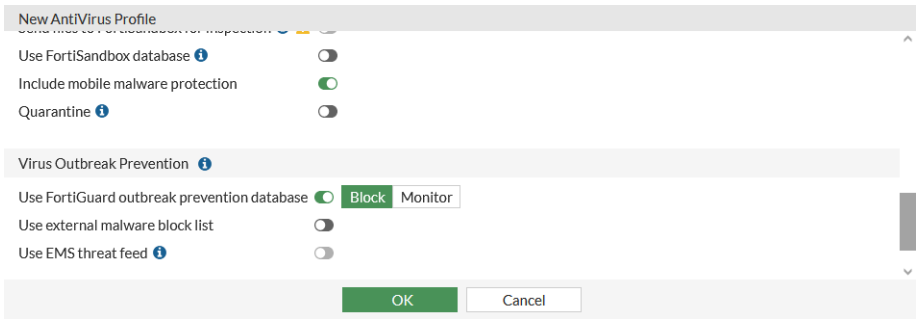
1. Go to *System* > *FortiGuard*.
2. In the *Subscriptions* > *Licensed* tab, expand the *Advanced Malware Protection* entitlement section and locate the *Virus outbreak prevention with file checksum query* entry.



3. See the instructions in the video How to Purchase or Renew FortiGuard Services, if required.

To enable FortiGuard outbreak prevention:

1. Go to *Security Profiles* > *AntiVirus*.
2. Edit an antivirus profile, or create a new one.
3. Under *Virus Outbreak Prevention*, enable *Use FortiGuard outbreak prevention database* and select *Block* or *Monitor*.



4. Click *OK*.

To verify FortiGuard antivirus license information:

```
# diagnose debug rating
Locale       : english

Service      : Web-filter
Status       : Enable
License      : Contract

Service      : Antispam
Status       : Disable

Service      : Virus Outbreak Prevention
Status       : Enable
License      : Contract

-=- Server List (Tue Feb 19 16:36:15 2019) -=-

IP                    Weight    RTT Flags  TZ    Packets  Curr Lost Total Lost
Updated Time
192.168.100.185          -218      2 DI     -8        113          0          0 Tue Feb 19
16:35:55 2019
```

To enable all scanunit debug categories:

```
# diagnose sys scanunit debug all
Set meta-category: all(0xffffffff)
Enabled categories(0xffffffff): daemon job quarantine analytics outbreak-prevention dlp antispam
file-filter
```

```
# diagnose debug enable
# su 4739 open
su 4739 req vfid 1 id 1 ep 0 new request, size 313, policy id 1, policy type 0
su 4739 req vfid 1 id 1 ep 0 received; ack 1, data type: 0
su 4739 job 1 request info:
su 4739 job 1   client 10.1.100.11:39412 server 172.16.200.44:80
su 4739 job 1   object_name 'zhvo_test.com'
su 4739 file-typing NOT WANTED options 0x0 file_filter no
su 4739 enable databases 0b (core mmdb extended)
```

```
su 4739 job 1 begin http scan
su 4739 scan file 'zhvo_test.com' bytes 68
su 4739 job 1 outbreak-prevention scan, level 0, filename 'zhvo_test.com'
su 4739 scan result 0
su 4739 job 1 end http scan
su 4739 job 1 inc pending tasks (1)
su 4739 not wanted for analytics: analytics submission is disabled (m 0 r 0)
su 4739 job 1 suspend
su 4739 outbreak-prevention recv error
su 4739 ftgd avquery id 0 status 1
su 4739 job 1 outbreak-prevention infected entryid=0
su 4739 report AVQUERY infection priority 1
su 4739 insert infection AVQUERY SUCCEEDED loc (nil) off 0 sz 0 at index 0 total infections 1
error 0
su 4739 job 1 dec pending tasks 0
su 4739 job 1 send result
su 4739 job 1 close
su 4739 outbreak-prevention recv error
```

FortiGuard provides several sample files to test the AV configuration on the FortiGate, which are available to download from https://www.fortiguard.com/sample-files. Test the *Virus Outbreak Prevention* feature by downloading *VO Sample* file. See Example 3: VO sample file on page 2254.

Zero-day malware stream scanning

Zero-day malware stream scanning enables real-time delivery of malware IOCs (Indicators of Compromise) to FortiGate devices using the fortimq daemon and notifications.

FortiGate automatically maintains an up-to-date malware hash database, removing outdated entries and optimizing performance without manual intervention.

A FortiSandbox Advanced (FSAC) or FortiSandbox Advanced Plus (FSAP) license is required for FortiGate to receive malware hash updates.

This topic includes the following information:

- Enabling fortimq daemon and notifications on page 2299
- How FortiGate receives malware-hash updates on page 2300
- Example 1: blocking malicious files on HTTP protocol on page 2301
- Example 2: monitoring malicious files on MAPI protocol on page 2304
- Troubleshooting commands on page 2306

Enabling fortimq daemon and notifications

The fortimq deamon can be enabled in the CLI and requires the following configuration:

- FortiGuard Anycast must be enabled.
- Cloud communications must be enabled.
- Notifications must be enabled.

To enable fortimq daemon in the CLI:

1. Enable fortimq daemon:

```
config system fortiguard
   set fortiguard-anycast enable
end
config system global
    set cloud-communication enable
end
```

2. Enable notifications:

```
config system fortiguard
   set subscribe-update notification enable
end
```

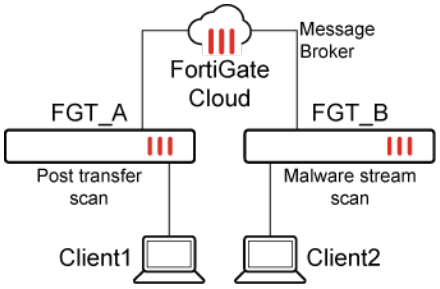
3. Confirm FortiGate is connected to fortimq daemon and able to receive messages:

```
# diagnose test application fortimq  1
connection status: connected
    acct: 901713
    fqdn: globalfortimq.fortinet.net
    port: 5671
    next-channel: 2
    msg count: 46266
    attempts: 0
```

How FortiGate receives malware-hash updates

In this example, FGT_A is configured to use post-transfer scanning to FortiGate Cloud Sandbox to monitor for malware. FGT_B is configured to use zero-day malware stream scanning and receive messages through fortimq daemon.

This example uses a sample malicious file with the hash value 71e4d6a0cc41d47e49ac0eeb45a26b2afefdd920538078eeda89b0c8f01bf622.



The user on Client1, downloads a malicious file, and FGT_A submits the file to FortiGate Cloud Sandbox for analysis. FortiGate Cloud Sandbox determines the file is malicious, and sends the verdict to FGT_A. FGT_B also receives the malware-hash update from fortimg daemon. Following is a summary of this process:

1. FGT_A generates a log for the submission to FortiGate Cloud Sandbox:

```
1: date=2025-03-25 time=14:06:39 eventtime=1742936798824763123 tz="-0700" logid="0201009233"
type="utm" subtype="virus" eventtype="analytics" level="information" vd="vdom1" policyid=1
poluuid="f820fc12-f07d-51ef-10a7-9ce92adafc22" policytype="policy" msg="File submitted to
Sandbox." action="analytics" service="HTTP" sessionid=2176 srcip=10.1.100.201
dstip=192.241.205.137 srcport=33454 dstport=80 srccountry="Reserved" dstcountry="United
States" srcintf="port2" srcintfrole="lan" dstintf="port3" dstintfrole="lan" srcuuid="eb7a7402-
4489-51ef-4afd-ca286e50ca04" dstuuid="eb7a7402-4489-51ef-4afd-ca286e50ca04" proto=6
direction="incoming" filename="fsa_downloader_1bf622.exe" filetype="exe"
url="http://rb3.ftnt.io/downloader/0.21385215657678536" profile="fgt_a_analytics_submit"
agent="Wget/1.20.3 (linux-gnu)" httpmethod="GET"
analyticscksum="71e4d6a0cc41d47e49ac0eeb45a26b2afefdd920538078eeda89b0c8f01bf622"
analyticssubmit="true"
```

2. FGT_A receives the verdict from FortiGate Cloud Sandbox:

```
1: date=2025-03-25 time=14:12:13 eventtime=1742937133549596694 tz="-0700" logid="0201009238"
type="utm" subtype="virus" eventtype="analytics" level="notice" vd="vdom1" srcip=10.1.100.201
dstip=192.241.205.137 srcport=33454 dstport=80 action="monitored" service="HTTP"
filename="fsa_downloader_1bf622.exe" fsaverdict="high risk"
analyticscksum="71e4d6a0cc41d47e49ac0eeb45a26b2afefdd920538078eeda89b0c8f01bf622"
dtype="fortisandbox"
```

3. FGT_B receives the malware-hash update from fortimq daemon within one minute and adds the new value to the scanunit file-has list.

```
# diagnose sys scanunit file-hash query sha256
71e4d6a0cc41d47e49ac0eeb45a26b2afefdd920538078eeda89b0c8f01bf622
0-day malware-stream '71e4d6a0cc41d47e49ac0eeb45a26b2afefdd920538078eeda89b0c8f01bf622'
description 'FSA/RISK_HIGH|FSA3KET320000157|7609597837562012709'
```

Example 1: blocking malicious files on HTTP protocol

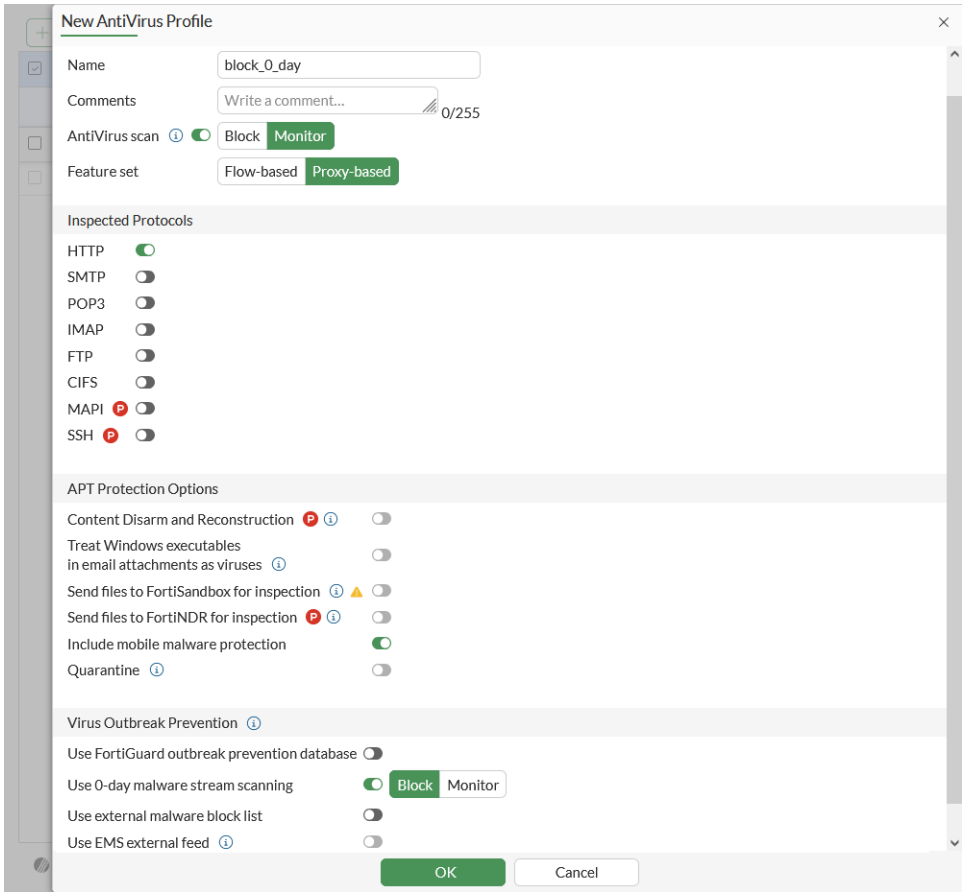
In this example, zero-day malware stream scanning is enabled on FGT_B and set to block and log malicious files on the HTTP protocol. The configuration includes:

- Enabling fortimq daemon and notifications. See Enabling fortimq daemon and notifications on page 2299.
- Configuring an AntiVirus profile to use zero-day malware stream scanning to block malicious files

To configure an AntiVirus profile in the GUI:

1. Go to *Security Profiles* > *AntiVirus*, and click *Create New*.
2. Set the following options:
   - Enter a name, such as block_0_day.
   - Enable *AntiVirus scan* and select *Monitor*.
   - Set *Feature set* to *Proxy-based*.
   - Enable *HTTP*.
   - Enable *Use 0-day malware stream scanning* and select *Block*.

- Set the other options as desired.

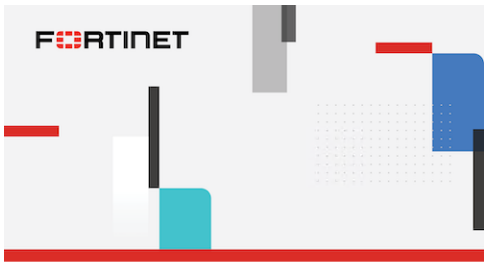


3. Click *OK* to save the profile.

To configure an AntiVirus profile in the CLI:

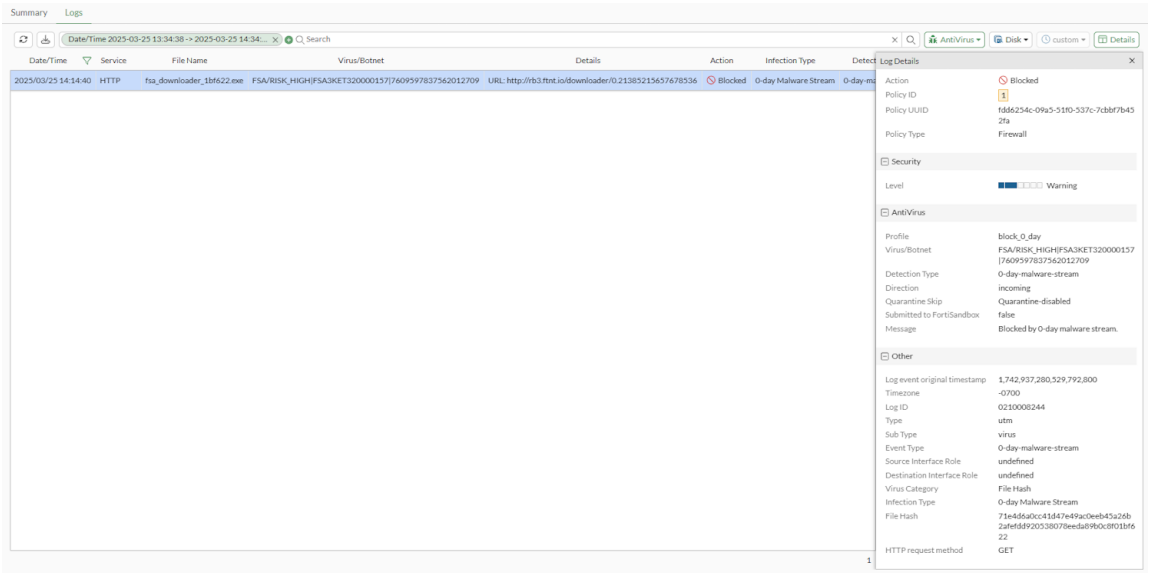
```
config antivirus profile
    edit "block_0_day"
        set feature-set proxy
        config http
            set av-scan monitor
            set malware-stream block
        end
    next
end
```

When the user on Client2 attempts to download a malicious file, FGT_B blocks the file and displays the HTTP block replacement message:



FGT_B adds the *block* action to the security event logs:

- In the GUI, go to *Log & Report* > *Security Events*:



- In the CLI:

```
1: date=2025-03-25 time=14:14:40 eventtime=1742937280529792875 tz="-0700" logid="0210008244"
type="utm" subtype="virus" eventtype="0-day-malware-stream" level="warning" vd="vdom1"
policyid=1 poluuid="fdd6254c-09a5-51f0-537c-7cbbf7b452fa" policytype="policy" msg="Blocked by
0-day malware stream." action="blocked" service="HTTP" sessionid=5921 srcip=10.1.100.252
dstip=192.241.205.137 srcport=50448 dstport=80 srccountry="Reserved" dstcountry="United
States" srcintf="port1" srcintfrole="undefined" dstintf="port9" dstintfrole="undefined"
srcuuid="b97d1360-09a5-51f0-7394-2ac219c6ac54" dstuuid="b97d1360-09a5-51f0-7394-2ac219c6ac54"
proto=6 direction="incoming" filename="fsa_downloader_1bf622.exe" quarskip="Quarantine-
disabled" virus="FSA/RISK_HIGH|FSA3KET320000157|7609597837562012709" viruscat="File Hash"
dtype="0-day-malware-stream" itype="0-day-malware-stream"
filehash="71e4d6a0cc41d47e49ac0eeb45a26b2afefdd920538078eeda89b0c8f01bf622"
url="http://rb3.ftnt.io/downloader/0.21385215657678536" profile="block_0_day"
agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/134.0.0.0 Safari/537.36 Edg/134.0.0.0" httpmethod="GET" analyticssubmit="false
```

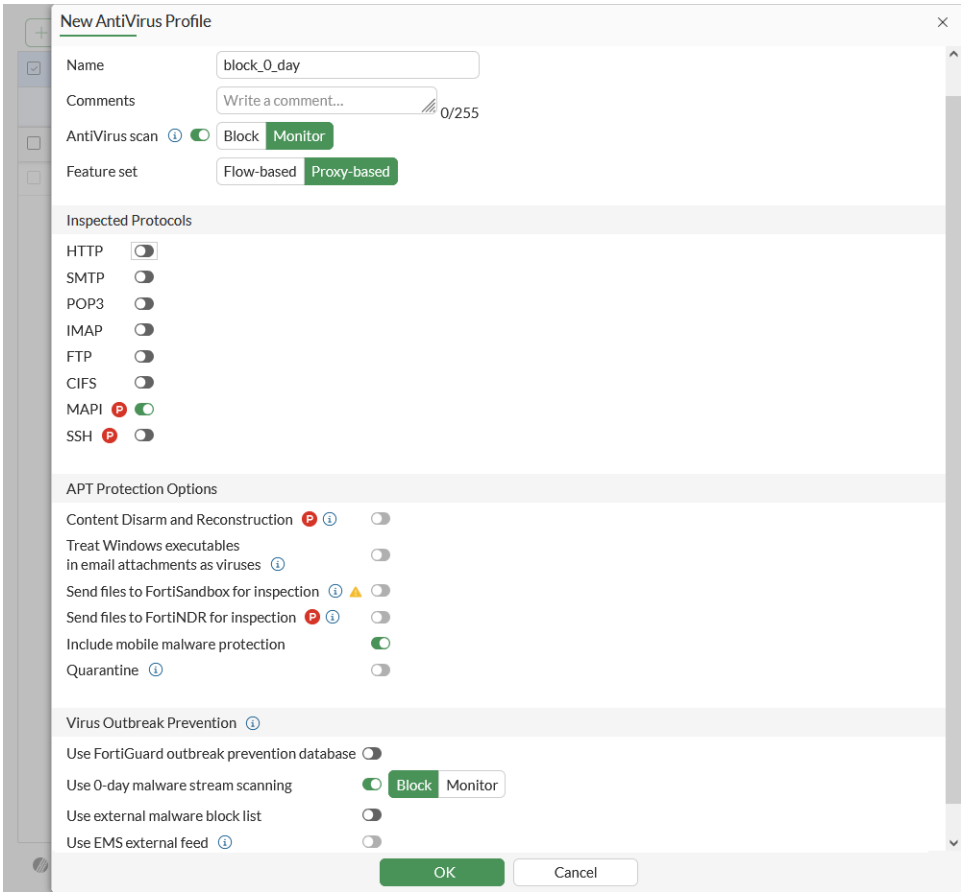
Example 2: monitoring malicious files on MAPI protocol

In this example, zero-day malware stream scanning is enabled on FGT_B and set to monitor and log malicious files on the MAPI protocol. The configuration includes:

- Enabling fortimq daemon and notifications. See Enabling fortimq daemon and notifications on page 2299.
- Configuring an AntiVirus profile to use zero-day malware stream scanning and monitor malicious files

To configure an AntiVirus profile in the GUI:

1. Go to *Security Profiles* > *AntiVirus*, and click *Create New*.
2. Set the following options:
   - Enter a name, such as monitor_0_day.
   - Enable *AntiVirus scan* and select *Monitor*.
   - Set *Feature set* to *Proxy-based*.
   - Enable *MAPI*.
   - Enable *Use 0-day malware stream scanning* and select *Monitor*.
   - Set the other options as desired.



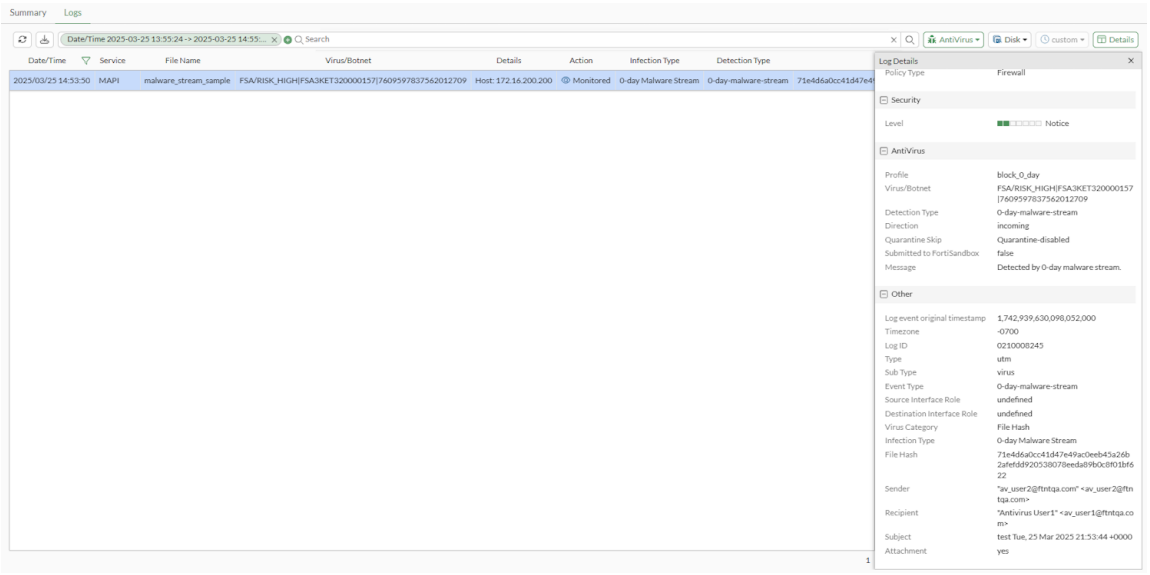
3. Click *OK* to save the profile.

To configure FGT_B in the CLI:

```
config antivirus profile
    edit "monitor_0_day"
        set feature-set proxy
        config mapi
            set av-scan monitor
            set malware-stream monitor
        end
    next
end
```

When the user on Client2 receives an email through MAPI protocol, FGT_B adds the *monitored* action to the security event logs:

- In the GUI, go to *Log & Report > Security Events*:



- In the CLI:

```
1: date=2025-03-25 time=14:53:50 eventtime=1742939630098052082 tz="-0700" logid="0210008245"
type="utm" subtype="virus" eventtype="0-day-malware-stream" level="notice" vd="vdom1"
policyid=1 poluuid="fdd6254c-09a5-51f0-537c-7cbbf7b452fa" policytype="policy" msg="Detected by
0-day malware stream." action="monitored" service="MAPI" sessionid=6797 srcip=10.1.100.252
dstip=172.16.200.200 srcport=50616 dstport=443 srccountry="Reserved" dstcountry="Reserved"
srcintf="port1" srcintfrole="undefined" dstintf="port9" dstintfrole="undefined"
srcuuid="b97d1360-09a5-51f0-7394-2ac219c6ac54" dstuuid="b97d1360-09a5-51f0-7394-2ac219c6ac54"
proto=6 direction="incoming" filename="malware_stream_sample" quarskip="Quarantine-disabled"
virus="FSA/RISK_HIGH|FSA3KET320000157|7609597837562012709" viruscat="File Hash" dtype="0-day-
malware-stream" itype="0-day-malware-stream"
filehash="71e4d6a0cc41d47e49ac0eeb45a26b2afefdd920538078eeda89b0c8f01bf622" profile="block_0_
day" from="\"av_user2@ftntqa.com\" <av_user2@ftntqa.com>" to="\"Antivirus  User1\" <av_
user1@ftntqa.com>" sender="\"av_user2@ftntqa.com\" <av_user2@ftntqa.com>"
recipient="\"Antivirus  User1\" <av_user1@ftntqa.com>" subject="test Tue, 25 Mar 2025 21:53:44
+0000" attachment="yes" analyticssubmit="false"
```

Troubleshooting commands

The following diagnose commands can be used for troubleshooting:

```
# diagnose test application fortimq <level>
```

Level	Description
1	fortimq status
2	dump ipc links
3	dump bindings
4	dump queue info
5	dump dns info
6	dump state info
7	dump timer info
8	dump unbound payloads
9	dump amqp client errors
99	restart fortimq daemon

External malware block list

This example demonstrates creating and implementing an external malware block list. See External malware block list on page 2243 for more information.

To create the external block list:

1. Create the malware hash list.

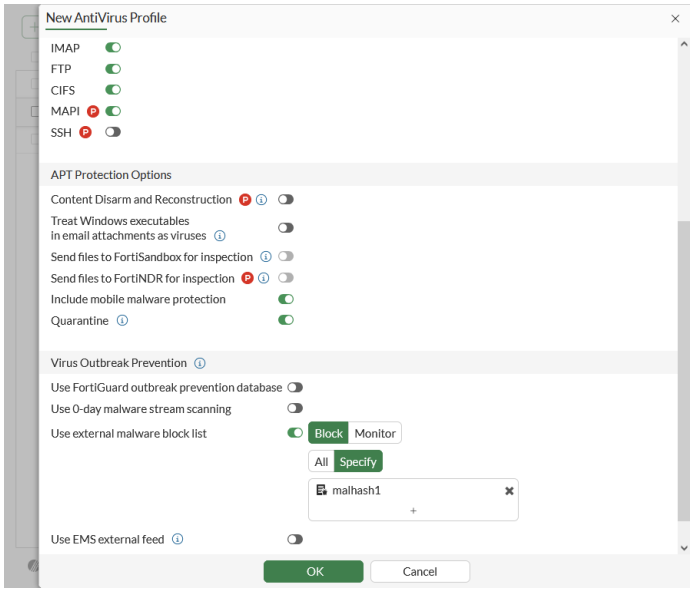
   The malware hash list follows a strict format in order for its contents to be valid. Malware hash signature entries must be separated into each line. A valid signature needs to follow this format:

```
# MD5 Entry with hash description
aa67243f746e5d76f68ec809355ec234  md5_sample1

# SHA1 Entry with hash description
a57983cb39e25ab80d7d3dc05695dd0ee0e49766  sha1_sample2

# SHA256 Entry with hash description
ae9bc0b4c5639d977d720e4271da06b50f7c60d1e2070e9c75cc59ab30e49379  sha256_sample1

# Entry without hash description
0289b0d967cb7b1fb1451339c7b9818a621903090e0020366ab415c549212521

# Invalid entries
7688499dc71b932feb126347289c0b8a_md5_sample2
7614e98badca10b5e2d08f8664c519b7a906fbd5180ea5d04a82fce9796a4b87sha256_sample3
```

2. Configure the external malware block list source:
   a. Go to *Security Fabric* > *External Connectors* and click *Create New*.
   b. Click *Malware Hash*.
   c. Configure the settings as needed. The URI must point to the malware hash list on the remote server.
   d. Click *OK*.
3. To view entries inside the malware block list on the *External Connectors* page, hover over the malware hash card and click *View Entries*.

To configure antivirus to use an external block list in the GUI:

1. Go to *Security Profiles* > *AntiVirus* and edit the antivirus profile.
2. In the *Virus Outbreak Prevention* section, enable *Use external malware block list* and click *Specify*.
3. Click the *+* in the field and select an external feed.
4. Optionally, enable *Quarantine*.



5. Configure the other settings as needed.
6. Click *OK*.

To configure antivirus to use an external block list in the CLI:

```
config antivirus profile
    edit "Demo"
        set feature-set proxy
        set mobile-malware-db enable
        config http
            set av-scan disable
            set outbreak-prevention block
            set external-blocklist block
            set quarantine enable
            set emulator enable
            set content-disarm disable
        end
```

```
        config ftp
            set av-scan disable
            set outbreak-prevention block
            set external-blocklist block
            set quarantine enable
            set emulator enable
        end
        config imap
            set av-scan monitor
            set outbreak-prevention block
            set external-blocklist block
            set quarantine enable
            set emulator enable
            set executables default
            set content-disarm disable
        end
        config pop3
            set av-scan monitor
            set outbreak-prevention block
            set external-blocklist block
            set quarantine enable
            set emulator enable
            set executables default
            set content-disarm disable
        end
        config smtp
            set av-scan monitor
            set outbreak-prevention block
            set external-blocklist block
            set quarantine enable
            set emulator enable
            set executables default
            set content-disarm disable
        end
        config mapi
            set av-scan monitor
            set outbreak-prevention block
            set external-blocklist block
            set quarantine enable
            set emulator enable
            set executables default
        end
        config nntp
            set av-scan disable
            set outbreak-prevention disable
            set external-blocklist disable
            set quarantine disable
            set emulator enable
        end
        config cifs
            set av-scan monitor
            set outbreak-prevention block
```

```
            set external-blocklist block
            set quarantine enable
            set emulator enable
        end
        config ssh
            set av-scan disable
            set outbreak-prevention disable
            set external-blocklist disable
            set quarantine disable
            set emulator enable
        end
        set outbreak-prevention-archive-scan enable
        set external-blocklist-enable-all disable
        set external-blocklist "malhash1"
        set av-virus-log enable
        set av-block-log enable
        set extended-log disable
        set scan-mode default
    next
end
```

The quarantine setting is configured in each protocol (`set quarantine`). The malware external feed is also specified (`set external-blocklist-enable-all disable`) to the external connector, malhash1 (`set external-blocklist "malhash1"`).

To verify the scanunit daemon updated itself with the external hashes:

```
# diagnose sys scanunit malware-list list
md5 'aa67243f746e5d76f68ec809355ec234' profile 'malhash1' description 'md5_sample1'
sha1 'a57983cb39e25ab80d7d3dc05695dd0ee0e49766' profile 'malhash1' description 'sha1_sample2'
sha256 '0289b0d967cb7b1fb1451339c7b9818a621903090e0020366ab415c549212521' profile 'malhash1'
description ''
sha256 'ae9bc0b4c5639d977d720e4271da06b50f7c60d1e2070e9c75cc59ab30e49379' profile 'malhash1'
description 'sha256_sample1'
```

Exempt list for files based on individual hash

The antivirus exempt list allows users to exempt known safe files that happen to be incorrectly classified as malicious by the AV signature and AV engine scan. Users can specify file hashes in MD5, SHA1, or SHA256 for matching, which are applied at a per-VDOM level. When matched, the FortiGate ignores the AV scan verdict so that the corresponding UTM behavior defined in the AV profile is not performed.

```
config antivirus exempt-list
    edit <name>
        set hash-type {md5 | sha1 | sha256}
        set hash <string>
        set status {enable | disable}
    next
end
```

The exempt list does not apply to results from outbreak prevention, machine learning, FortiNDR, or FortiSandbox inline scans.

In this example, an antivirus exempt list is configured for the EICAR anti-malware test file. Although the antivirus profile is configured to block HTTP, the client is able to download the file.

To configure an antivirus exempt list:

1. Configure the antivirus profile:

```
config antivirus profile
    edit "av"
        set feature-set proxy
        config http
            set av-scan block
        end
    next
end
```

2. Configure the antivirus exempt list:

```
config antivirus exempt-list
    edit "test-hash"
        set comment "eicar.com"
        set hash-type md5
        set hash "44d88612fea8a8f36de82e1278abb02f"
        set status enable
    next
end
```

3. Get a client to access https://www.eicar.com/ and download the anti-malware test file.
   The FortiGate exempts the AV scan verdict and bypasses the file. The client can download the file and no replacement message is displayed.

Downloading quarantined files in archive format

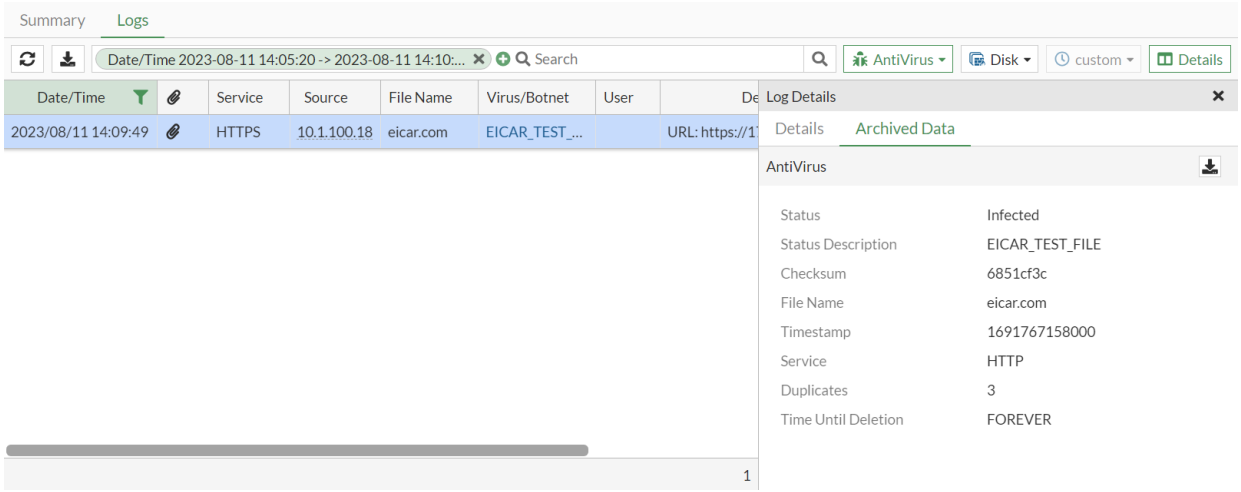
The FortiGate can download quarantined files in an archive format (.TGZ) instead of the original raw file. This allows for a more detailed analysis of the quarantined files and reduces the risk of malware infection.

The FortiGate must have a disk logging capacity or be connected to FortiAnalyzer for logging.

To download a quarantined archive file:

1. Ensure that quarantining files is enabled in the AV profile:
   a. Go to *Security Profiles* > *AntiVirus* and edit the AV profile.
   b. In the *APT Protection Options* section, verify that *Quarantine* is enabled. At least one protocol must be enabled in the AV profile for inspection, and *AntiVirus scan* must be enabled for the *Quarantine* option to work.
2. Go to *Log & Report* > *Security Events* and select the *AntiVirus* card.

3. Select a log entry and click *Details*. The *Log Details* pane opens.
4. Select the *Archived Data* tab and click the download icon (in the *AntiVirus* title bar).



Web filter

Web filtering feature offers a flexible and effective solution to control and manage internet usage within your organization, tailored to your specific requirements and objectives. It's a powerful tool for enhancing productivity and maintaining a secure digital environment. This feature has the ability to restrict users from accessing certain websites or Uniform Resource Locators (URLs) by inhibiting the users' browsers from loading the pages linked to these particular sites onto their devices.

Web filtering serves as the primary shield against attacks originating from the web. The FortiGuard URL Filtering Service provides comprehensive threat protection to address threats including ransomware, credential-theft, phishing, and other web-borne attacks. It uses AI-driven behavior analysis and correlation to block unknown malicious URLs almost immediately, with near-zero false negatives. The FortiGate's WAD daemon sends the URLs to FortiGuard in real-time for category determination.

This section includes information about web filter techniques and configurations:

- Web filter introduction on page 2311
- Advanced CLI configuration on page 2342
- Configuration examples on page 2362

Web filter introduction

Web filtering restricts or controls user access to web resources and can be applied to firewall policies using either policy-based or profile-based NGFW mode.

In FortiOS, there are three main components of web filtering:

- Web content filter: blocks web pages containing words or patterns that you specify.
- URL filter: uses URLs and URL patterns to block or exempt web pages from specific sources, or block malicious URLs discovered by FortiSandbox.
- FortiGuard Web Filtering service: provides many additional categories you can use to filter web traffic.

These components interact with each other to provide maximum control over what users on your network can view and protect your network from many internet content threats.

Web filters are applied in the following order:

1. URL filter
2. FortiGuard Web Filtering
3. Web content filter
4. Web script filter
5. Antivirus scanning

FortiOS includes three preloaded web filter profiles:

- *default*
- *monitor-all* (monitors and logs all URLs visited, flow-based)
- *wifi-default* (default configuration for offloading WiFi traffic)

You can customize these profiles, or you can create your own to manage network user access.

Some features of this functionality require a subscription to FortiGuard Web Filtering.

Web filter techniques

Web filtering is a crucial aspect of cybersecurity, helping organizations protect their digital systems from threats and ensure safe internet usage. A range of techniques and tools are employed to ensure a secure, reliable, and safe digital environment for organizations.

The following table describes some of the industry standard techniques that are used for Web filtering and if they can be configured in the GUI or CLI.

Technique	Description	GUI	CLI
URL filtering	Blocks access to websites based on their URLs. See for more information.	✓	✓
Category-based filtering	The FortiGuard URL Filtering Service provides comprehensive threat protection to address threats including ransomware, credential-theft, phishing, and other web-borne attacks. See FortiGuard filter on page 2315 for more information.	✓	✓
Image and video filtering	Remove explicit results, such as sexual content or graphic violence, that a user may encounter when conducting a search through a search engine. See Search engines on page 2326 for more information.	✓	✓
Content filtering	Block access based on specific keywords or patterns present on a	✓	✓

Technique	Description	GUI	CLI
	web page. See Web content filter on page 2336 for more information.		
Credential phishing preventing	Scans for corporate credentials submissions to external websites and cross-references those entries with legitimate corporate credentials. See Credential phishing prevention on page 2343 for more information.		✓
Blocklists and allowlists	Curate lists of prohibited and permitted websites. See FortiGuard category external feed on page 4709 for more information.	✓	✓
Web filter plugin	Extension that you can install on your web browser to enforce Web Filter rules. See Web browser plugin for HTTPS web filtering in the FortiClient Administration Guide for more information.	✓	✓
Endpoint filtering	Control and regulate access to websites at the level of individual devices in a network. See Web & Video Filter in the FortiClient Administration Guide for more information.	✓	✓
Websense web filtering service	Send traffic to the third-party web filtering service for rating and approval checking. See Websense Integrated Services Protocol on page 2351 for more information.		✓

See Configuring a web filter profile on page 2313 for more information.

Configuring a web filter profile

Web filtering restricts or controls user access to web resources and can be applied to firewall policies using either policy-based or profile-based NGFW mode.

The feature set setting (proxy or flow) in the web filter profile must match the inspection mode setting (proxy or flow) in the associated firewall policy. For example, a flow-based web filter profile must be used with a flow-based firewall policy.

An SSL inspection profile (such as the certificate-inspection profile) and a web filter profile must both be selected in the associated firewall policy. See SSL & SSH Inspection on page 2690.

Some web filter profile options can only be configured in the CLI. See Advanced CLI configuration on page 2342 and the FortiOS CLI Reference for more information.

To configure a web filter profile:

1. Go to *Security Profiles* > *Web Filter* and click *Create New*.
2. Configure the following settings:

Name	Enter a unique name for the profile.
Comments	(Optional) Enter a comment.

Feature set	Select the feature set for the profile. The feature set mode must match the inspection mode used in the associated firewall policy.
	• Flow-based
	• Proxy-based
	Additional options are available in proxy-based mode and are identified in the GUI with a P icon. See Inspection mode feature comparison on page 2214.
	If the *Feature set* option is not visible, enter the following in the CLI:
	`config system settings`
	`set gui-proxy-inspection enable`
	`end`

FortiGuard Category Based Filter	Enable to use the category-based filters from FortiGuard. A default action is assigned to each category, and you can change the action. See FortiGuard filter on page 2315.
Category Usage Quota	This option is available in proxy-based mode and can be applied to categories set to *Monitor*, *Warning*, and *Authenticate*. See Category usage quota on page 2323.
Allow users to override blocked categories	Enable to allow certain users or user groups to override websites blocked by web filtering profiles for a specified length of time. See Web profile override on page 2747.
Groups that can override	Select one or more user groups that can override blocked websites. The user group must be specified as the Source in the firewall policies using this profile.
Profile Name	Select what web filter profiles can be overridden.
Switch applies to	Specify whether the override applies to a User, User Group, or IP address. Alternately select Ask to prompt the user to log in to access the web page.
Switch duration	Select *Predefined* to specify how many days, hours, and minutes to allow the override. Select Ask to prompt the user to specify how long to allow the override.
Search Engines	
Enfore 'Safe Search' on Google, Yahoo!, Bing, Yandex	Enable to prevent explicit websites and images from appearing in search results. See Safe search on page 2326.
Restrict YouTube Access	Enable to filter out potentially mature videos. See Restrict Google account usage to specific domains on page 2341.
Log all search keywords	This option is available in proxy-based mode. Enable to log all search phrases. See Log all search keywords on page 2328.
Static URL Filter	

	Block invalid URLs	Enable to block websites when their SSL certificate CN field lacks a valid domain name. See Block invalid URLs on page 2329.
	URL Filter	Enable to specify URL patterns and an action for FortiGate to take when matching URL patterns are found in traffic. See URL filter on page 2329.
	Block malicious URLs discovered by FortiSandbox	Enable to block malicious URLs found by FortiSandbox. Requires FortiGate to be connected to a registered FortiSandbox. See Block malicious URLs discovered by FortiSandbox on page 2335.
	Content Filter	Enable to specify word or patterns to be used to identify and control access to web pages. See Web content filter on page 2336.
Rating Options		
	Allow websites when a rating error occurs	Enable to allow access to websites that return a rating error from the FortiGuard Web Filter service. See Allow websites when a rating error occurs on page 2339.
	Rate URLs by domain and IP address	Enable for FortiGate to always send both the URL domain name and the TCP/IP packet's IP address (except for private IP addresses) to FortiGuard for rating. See Rate URLs by domain and IP address on page 2340.
Proxy Options		
	Restrict Google account usage to specific domains	This option is available in proxy-based mode. Enable to block access to certain Google accounts and services. See Restrict Google account usage to specific domains on page 2341.
	HTTP POST Action	Enable to specify how to handle HTTP POST traffic. See HTTP POST action on page 2341.
	Remove Java Applets	This option is available in proxy-based mode. Enable to remove Java applets from web traffic. See Remove Java applets, ActiveX, and cookies on page 2342.
	Remove ActiveX	This option is available in proxy-based mode. Enable to remove ActiveX from web traffic. See Remove Java applets, ActiveX, and cookies on page 2342.
	Remove Cookies	Enable to remove cookies from web traffic. See Remove Java applets, ActiveX, and cookies on page 2342.

3. Click *OK*.

FortiGuard filter

The FortiGuard filter enhances the web filter features by sorting billions of web pages into a wide range of categories that users can allow or block.

The FortiGuard Web Filtering service includes over 45 million individual website ratings that apply to more than two billion pages. When the FortiGuard filter is enabled in a web filter profile and applied to firewall policies, if a request for a web page appears in traffic controlled by one of the firewall policies, the URL is sent to the nearest FortiGuard server. The URL category or rating is returned. If the category is blocked, the FortiGate shows a

replacement message in place of the requested page. If the category is not blocked, the page request is sent to the requested URL as normal.

To use this service, you must have a valid FortiGuard license.

The following actions are available:

FortiGuard web filter action	Description
Allow	Permit access to the sites in the category.
Monitor	Permit and log access to sites in the category. User quotas can be enabled for this option (see Category usage quota on page 2323).
Block	Prevent access to the sites in the category. Users trying to access a blocked site see a replacement message indicating the site is blocked.
Warning	Display a message to the user allowing them to continue if they choose.
Authenticate	Require the user to authenticate with the FortiGate before allowing access to the category or category group.
Disable	Remove the category from the from the web filter profile. This option is only available for local or remote categories from the right-click menu.

FortiGuard web filter categories

FortiGuard has many web filter categories, including two local categories and a special remote category. Refer to the following table for more information:

FortiGuard web filter category	Where to find more information
All URL categories	See Web Filter Categories.
Local categories	See Web rating override on page 2736.
Remote category	See External feeds on page 4695.

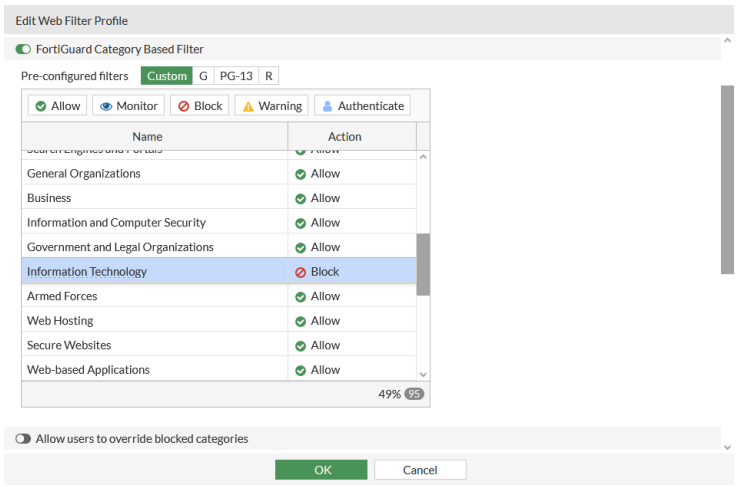
The priority of categories is local category > external category > FortiGuard built-in category. If a URL is configured as a local category, it only follows the behavior of the local category and not the external or FortiGuard built-in category.

Blocking a web category

The following example shows how to block a website based on its category. The Information Technology category (category 52) will be blocked.

To block a category in the GUI:

1. Go to *Security Profiles* > *Web Filter* and click *Create New*, or edit an existing profile.
2. In the *FortiGuard category based filter* section, select *Information Technology*, then click *Block*.



3. Configure the remaining settings as needed.
4. Click *OK*.

To block a category in the CLI:

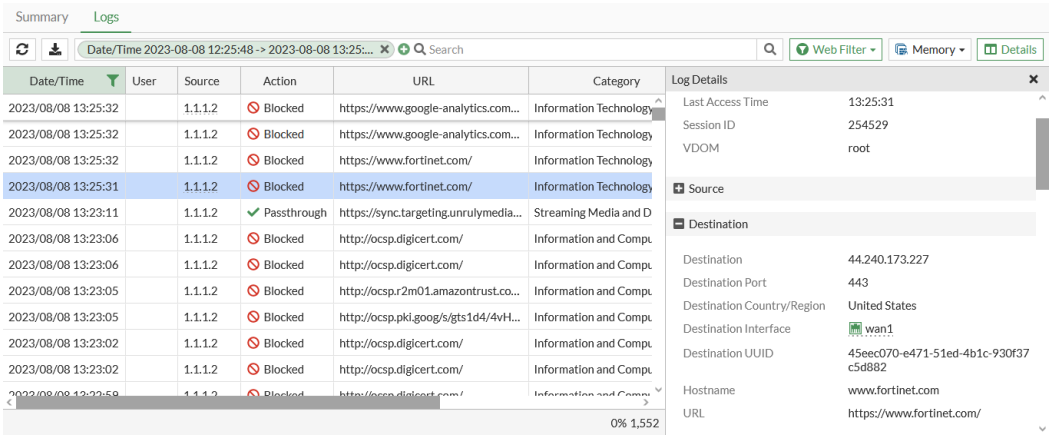
```
config webfilter profile
    edit "webfilter"
        config ftgd-wf
            unset options
            config filters
                edit 1
                    set category 52
                    set action block
                next
            end
        end
    next
end
```

You can use the `get webfilter categories` command to determine the web filtering category that corresponds to a given category ID.

To verify that the category is blocked:

1. Go to a website that belongs to the blocked category, such as www.fortinet.com.
   The page should be blocked and display a replacement message.

To view the log of a blocked website in the GUI:

1. Go to *Log & Report* > *Security Events*.
2. Click the *Web Filter* card name.
3. Select an entry with *blocked* in the *Action* column and click *Details*.
4. 

To view the log of a blocked website in the CLI:

```
# execute log filter category utm-webfilter
# execute log display

4: date=2023-08-08 time=13:25:31 eventtime=1691526331836645153 tz="-0700" logid="0316013056"
type="utm" subtype="webfilter" eventtype="ftgd_blk" level="warning" vd="root" policyid=1
poluuid="4a4b9d00-e471-51ed-71ec-c1a3bc8f773c" policytype="policy" sessionid=254529 srcip=1.1.1.2
srcport=60836 srccountry="Australia" srcintf="internal7" srcintfrole="lan" srcuuid="45eec070-e471-
51ed-4b1c-930f37c5d882" dstip=44.240.173.227 dstport=443 dstcountry="United States" dstintf="wan1"
dstintfrole="wan" dstuuid="45eec070-e471-51ed-4b1c-930f37c5d882" proto=6 service="HTTPS"
hostname="www.fortinet.com" profile="default" action="blocked" reqtype="direct"
```

```
url="https://www.fortinet.com/" sentbyte=517 rcvdbyte=0 direction="outgoing" msg="URL belongs to a
denied category in policy" ratemethod="domain" cat=52 catdesc="Information Technology"
```

Allowing users to override blocked categories

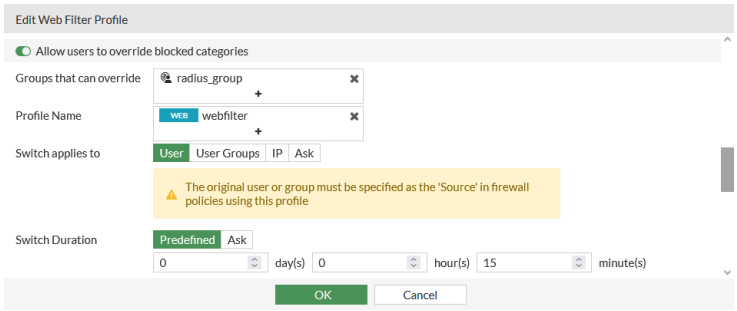
There is an option to allow users with valid credentials to override blocked categories.

To allow users to override blocked categories in the GUI:

1. Go to *Security Profiles* > *Web Filter* and click *Create New*, or edit an existing profile.
2. Enable *Allow users to override blocked categories*.
3. Enter information in the following fields:

Groups that can override	Add the user group that will be allowed to override.
Profile Name	Add the web filter profile the overridden group will use. This cannot be the same profile as its own.
Switch applies to	Select *User*, *User Groups*, *IP*, or *Ask*.
Switch Duration	Select either *Predefined* to specify a duration, or *Ask* for user input.

4. Configure the other settings as needed.



5. Click *OK*.

To allow users to override blocked categories in the CLI:

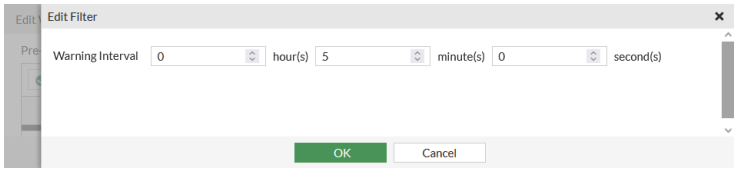
```
config webfilter profile
    edit "webfilter"
        set ovrd-perm bannedword-override urlfilter-override fortiguard-wf-override contenttype-
check-override
        config override
            set ovrd-user-group "radius_group"
            set profile "webfilter"
        end
        config ftgd-wf
            unset options
        end
    next
end
```

Issuing a warning on a web category

The following example shows how to issue a warning when a user visits a website in a specific category (Information Technology, category 52).

To configure a warning for a category in the GUI:

1. Go to *Security Profiles* > *Web Filter* and click *Create New*, or edit an existing profile.
2. In the *FortiGuard category based filter* section, select *Information Technology*, then click *Warning*.
3. Set the *Warning Interval*, then click *OK*.
   The warning interval is the amount of time until the warning appears again after the user proceeds past it.



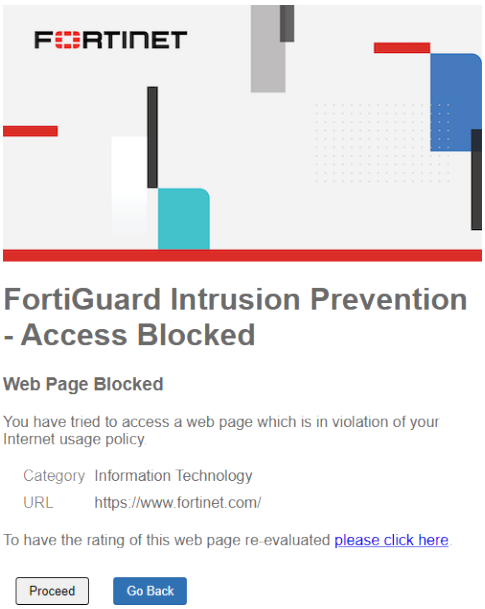
4. Configure the remaining settings as needed.
5. Click *OK*.

To configure a warning for a category in the CLI:

```
config webfilter profile
    edit "webfilter"
        config ftgd-wf
            unset options
            config filters
                edit 1
                    set category 52
                    set action warning
                next
            end
        end
    next
end
```

To verify that the warning works:

1. Go to a website that belongs to the category, such as www.fortinet.com.
2. On the warning page, click *Proceed* or *Go Back*.



Authenticating a web category

The following example shows how to authenticate a website based on its category (Information Technology, category 52).

To authenticate a category in the GUI:

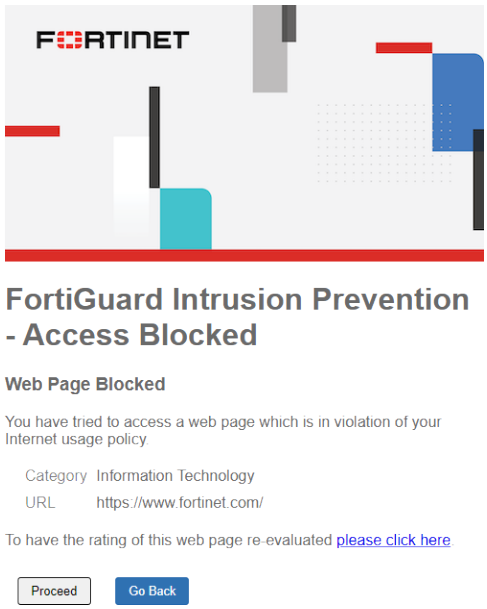
1. Go to *Security Profiles* > *Web Filter* and edit or create a new web filter profile.
2. In the *FortiGuard category based filter* section, select *Information Technology*, then click *Authenticate*.
3. Set the *Warning Interval* and select one or more user groups, then click *OK*.
4. Configure the remaining settings as needed.
5. Click *OK*.

To authenticate a category in the CLI:

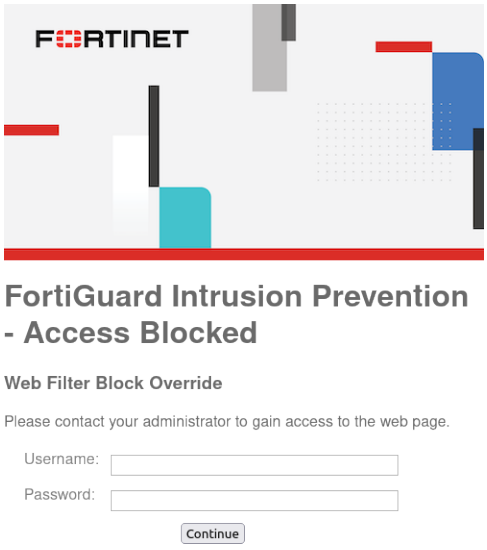
```
config webfilter profile
    edit "webfilter"
        config ftgd-wf
            unset options
            config filters
                edit 1
                    set category 52
                    set action authenticate
                    set auth-usr-grp "local_group"
                next
            end
        end
    next
end
```

To verify that you have configured authentication:

1. Go to a website that belongs to the category, such as www.fortinet.com.
2. On the warning page, click *Proceed*.



3. Enter the username and password for the configured user group, then click *Continue*.



Customizing the replacement message page

When the category action is *Block*, *Warning*, or *Authenticate*, you can customize the replacement message page that a user sees.

To customize the replacement message page:

1. Go to *Security Profiles* > *Web Filter* and edit or create a new web filter profile.
2. In the *FortiGuard category based filter* section, right-click on a category and select *Customize*.

3. Select a *Replacement Message Group*. See Replacement message groups on page 3903 for details.
4. Optionally, click *Edit FortiGuard Block Page* or *Edit FortiGuard Warning Page* to make modifications.
5. Click *Save*.
6. Configure the remaining settings as needed.
7. Click *OK*.

Customizing the CA certificate

When accessing a HTTPS webpage, in order to intercept the connection and perform an override, warning, or authentication, the connection must be proxied and the warning and/or authentication page must be signed with FortiGate's CA certificate. The client accessing the page must trust the CA in order to avoid certificate errors while browsing.

When applying the web filter profile to a firewall policy, an SSL inspection profile must also be selected.

To apply a custom certificate to the SSL inspection profile in the GUI:

1. Go to *Security Profiles* > *SSL/SSH Inspection* and click *Create New*, or edit an existing profile.
2. Under *SSL Inspection Options*, set *CA certificate* to the desired custom CA certificate.
3. Click *OK*.
4. On the client endpoints, ensure this custom CA is trusted.

To apply a custom certificate to the SSL inspection profile in the CLI:

```
config firewall ssl-ssh-profile
    edit <name>
        set caname <custom_CA_certificate>
    next
end
```

Category usage quota

In addition to using category and classification blocks and overrides to limit user access to URLs, you can set a daily quota by category, category group, or classification. Quotas allow access for a specified length of time or a specific bandwidth, and are calculated separately for each user. Quotas are reset daily at midnight.

Quotas can be set for the *Monitor*, *Warning*, or *Authenticate* actions. Once the quota is reached, the traffic is blocked and the replacement message page displays.

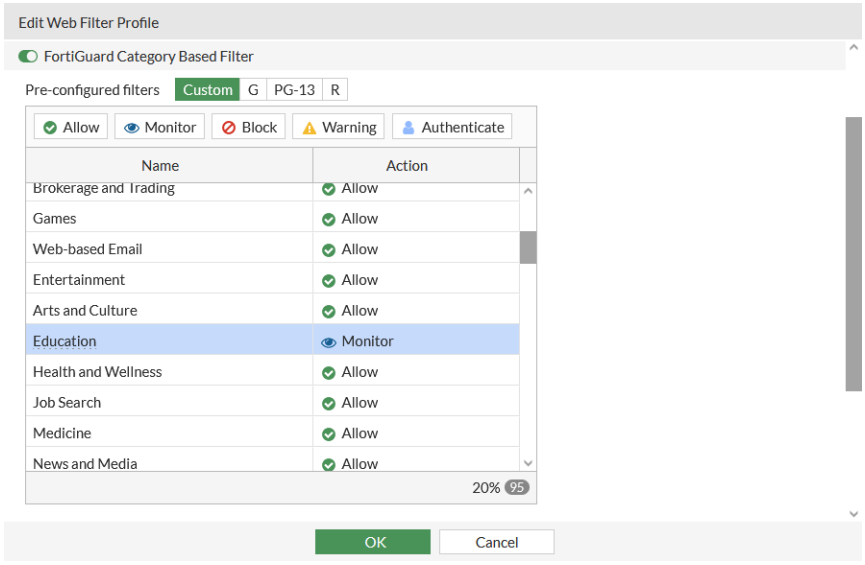
Quotas are only available in proxy-based inspection mode.

Configuring a quota

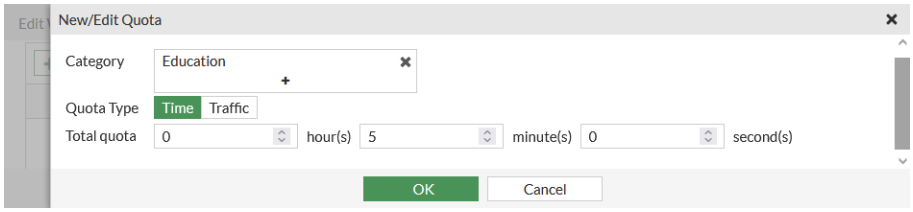
The following example shows how to set a time quota for the education category (category 30).

To configure a quota in the GUI:

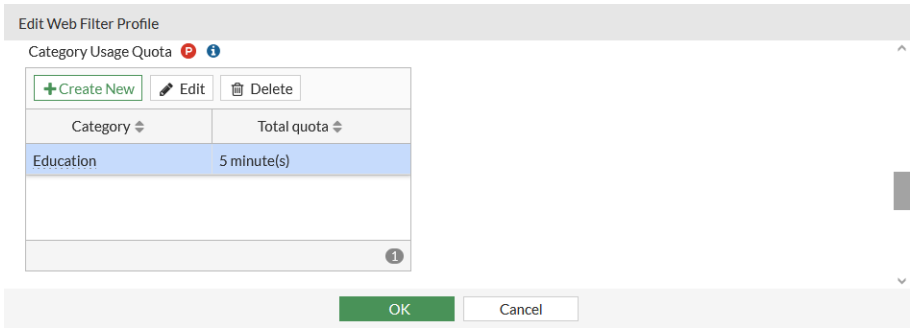
1. Go to *Security Profiles* > *Web Filter* and click *Create New*, or edit an existing profile.
2. For *Feature set*, select *Proxy-based*.
3. In the *FortiGuard category based filter* section, scroll to the *General Interest - Personal* and click the *+* to expand the section.
4. Select *Education*, then click *Monitor*.



5. In the *Category Usage Quota* section, click *Create New*.
   The *New/Edit Quota* pane opens.
6. In the *Category* field, select *Education*.
7. For the *Quota Type*, select *Time* and set the *Total quota* to 5 minutes.



8. Click *OK*. The entry appears in the table.



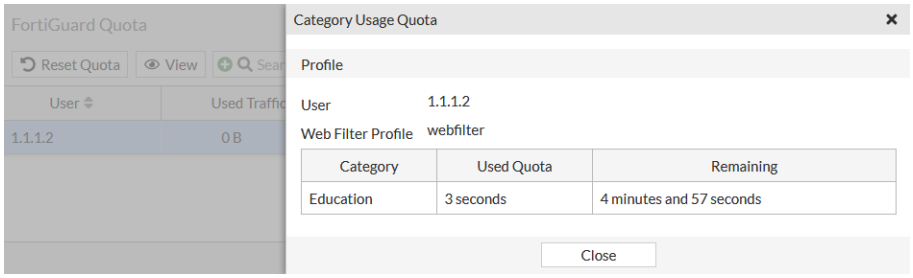
9. Configure the other settings as needed.
10. Click *OK*.

To configure a quota in the CLI:

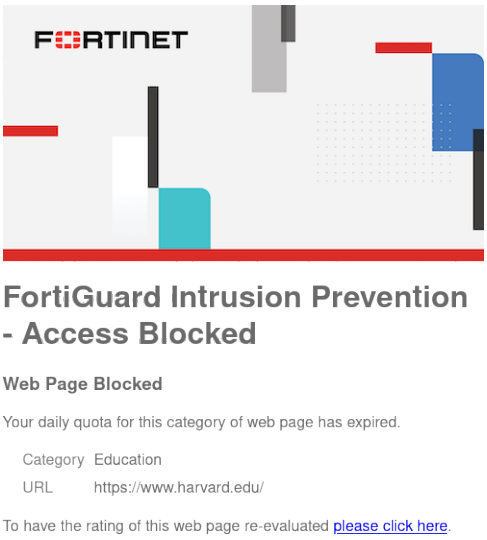
```
config webfilter profile
    edit "webfilter"
        config ftgd-wf
           unset options
           config filters
              edit 1
                 set category 30
              next
           end
           config quota
              edit 1
                 set category 30
                 set type time
                 set duration 5m
              next
           end
        end
    next
end
```

To verify the quota usage:

1. Go to a website that belongs to the education category, such https://www.harvard.edu/. You can view websites in that category at the moment.
2. In FortiOS, go to *Dashboard* > *FortiGuard Quota Monitor* to check the used and remaining time.



3. When the quota reaches its limit, traffic is blocked and the replacement page displays.



Search engines

This topic gives examples of the following advanced filter features:

- Safe search on page 2326
- Restrict YouTube and Vimeo access on page 2327
- Log all search keywords on page 2328

Safe search

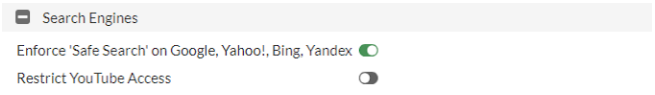
This setting applies to popular search sites and prevents explicit websites and images from appearing in search results.

The supported search sites are:

- Google
- Yahoo
- Bing
- Yandex

To enable safe search in the GUI:

1. Go to *Security Profiles* > *Web Filter* and click *Create New*, or edit an existing profile.
2. In the *Search Engines* section, enable *Enforce 'Safe Search' on Google, Yahoo!, Bing, Yandex*.



3. Click *OK*.

To enable safe search in the CLI:

```
config webfilter profile
    edit "webfilter"
        config web
            set safe-search url header
        end
    next
end
```

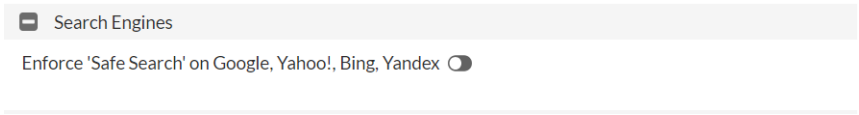
Restrict YouTube and Vimeo access

The *Restrict YouTube access* setting in the video filter profile adds the HTTP header `YouTube-Restrict: Strict` or `YouTube-Restrict: Moderate` into the HTTP request when enabled. When YouTube reads this header, it applies the appropriate content restriction based on the selected mode. YouTube Restricted Mode is an optional setting that filters out potentially mature videos while leaving a large number of videos still available (see Restrict YouTube content available to users and Manage your organization's YouTube settings for more information). Google defines the restricted YouTube access modes as follows:

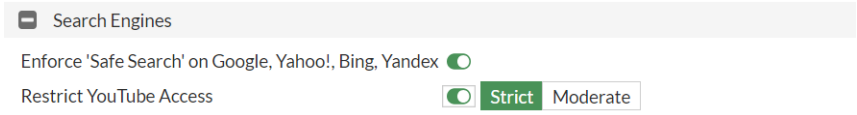
- Strict Restricted YouTube access: this setting is the most restrictive. Strict Restricted Mode does not block all videos, but works as a filter to screen out many videos based on an automated system, while leaving some videos still available for viewing.
- Moderate Restricted YouTube access: this setting is similar to Strict Restricted Mode but makes a much larger collection of videos available.

To restrict YouTube access in the GUI:

1. Go to *Security Profiles* > *Web Filter* and click *Create New*, or edit an existing profile.
2. In the *Search Engines* section, enable *Enforce 'Safe Search' on Google, Yahoo!, Bing, Yandex* to display the *Restrict YouTube Access* option.



   When `safe-search` is set to `header` in the CLI, the *Restrict YouTube Access* option is visible in the GUI.
3. Enable *Restrict YouTube Access* and select either *Strict* or *Moderate*.



4. Click *OK*.

> It is recommended to set `safe-search` to `url` and `header` because some search engines, such as Google, use the URL, and other search engines, such as Bing, use the header.
>
> When you enable *Enforce 'Safe Search' on Google, Yahoo!, Bing, Yandex* in the GUI, `safe-search` is set to `url header` in the CLI.

To restrict YouTube access in the CLI:

```
config webfilter profile
    edit <name>
        config web
            set safe-search url header
            set youtube-restrict {none | strict | moderate}
        end
    next
end
```

The file filter profile includes a setting to restrict Vimeo access, which can only be configured in the CLI.

To restrict Vimeo access:

```
config webfilter profile
    edit <name>
        config web
            set safe-search url header
            set vimeo-restrict {7 | 134}
        end
    next
end
```

`vimeo-restrict {7 | 134}`	Set the Vimeo restriction: • 7: do not show mature content • 134: do not show unrated and mature content

Log all search keywords

Use this setting to log all search phrases.

This filter is only available in proxy-based inspection mode.

To enable logging search keywords in the GUI:

1. Go to *Security Profiles* > *Web Filter* and click *Create New*, or edit an existing profile.
2. In the *Search Engines* section, enable *Log all search keywords*.
3. Click *OK*.

To enable logging search keywords in the CLI:

```
config webfilter profile
    edit "webfilter"
        config web
            set log-search enable
        end
```

```
    next
end
```

Static URL filter

This topic gives examples of the following advanced filter features:

- Block invalid URLs on page 2329
- URL filter on page 2329
- Disabling simple type URL filter from blocking subdomains on page 2335
- Block malicious URLs discovered by FortiSandbox on page 2335
- Web content filter on page 2336

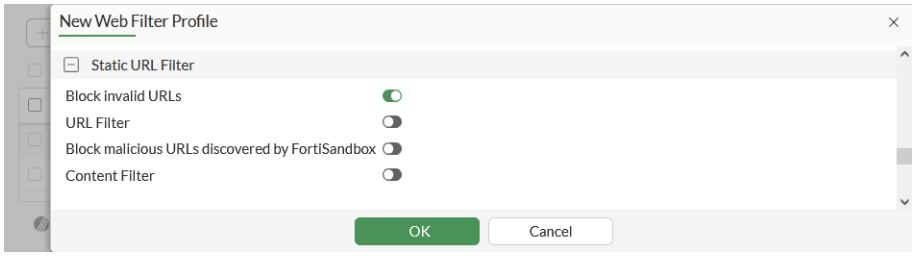
Block invalid URLs

Use this setting to block websites when their SSL certificate CN field does not contain a valid domain name.

This option also blocks URLs that contains spaces. If there is a space in the URL, it must be written as *%20* in the URL path.

To block invalid URLs in the GUI:

1. Go to *Security Profiles* > *Web Filter* and click *Create New*, or edit an existing profile.
2. In the *Static URL Filter* section, enable *Block invalid URLs* .



3. Click *OK*.

To block invalid URLs in the CLI:

```
config webfilter profile
    edit "webfilter"
        set options block-invalid-url
    next
end
```

URL filter

The URL filter uses specific URLs with patterns containing text and regular expressions so the FortiGate can process the traffic based on the filter action (exempt, block, allow, monitor) and web pages that match the criteria. Once a URL filter is configured, it can be applied to a firewall policy.

The following filter types are available:

URL filter type	Description
Simple	The FortiGate tries to strictly match the full context. For example, if you enter *www.facebook.com* in the *URL* field, it only matches traffic with *www.facebook.com*. It won't match *facebook.com* or *message.facebook.com*. One exception is subdomains. For example, if you specify *slack.com* in the URL field, this will match *hooks.slack.com* as well. When the FortiGate finds a match, it performs the selected URL action.
Regular expression/ wildcard	The FortiGate tries to match the pattern based on the rules of regular expressions or wildcards. For example, if you enter **fa** in the *URL* field, it matches all the content that has *fa* such as *www.facebook.com*, *message.facebook.com*, *fast.com*, and so on. When the FortiGate finds a match, it performs the selected URL action.

For more information, see the URL Filter expressions technical tip in the Knowledge Base.

The following actions are available:

URL filter action	Description
Exempt	The traffic is allowed to bypass the remaining FortiGuard web filters, web content filters, web script filters, antivirus scanning, and DLP proxy operations.
Block	The FortiGate denies or blocks attempts to access any URL that matches the URL pattern. A replacement message is displayed.
Allow	The traffic is passed to the remaining FortiGuard web filters, web content filters, web script filters, antivirus proxy operations, and DLP proxy operations. If the URL does not appear in the URL list, the traffic is permitted.
Monitor	The traffic is processed the same way as the *Allow* action. For the *Monitor* action, a log message is generated each time a matching traffic pattern is established.

The exempt URL filter action can be configured to bypass all or certain scanning and filtering operations. This setting can only be configured in the CLI.

If the `action` is set to `exempt`, use `set exempt` to select the scanning and filtering operations that exempt URLs skip.

```
config webfilter urlfilter
    edit <id>
        config entries
            edit <id>
                set action exempt
                set exempt {av web-content activex-java-cookie dlp fortiguard range-block pass
antiphish all}
            next
        end
    next
end
```

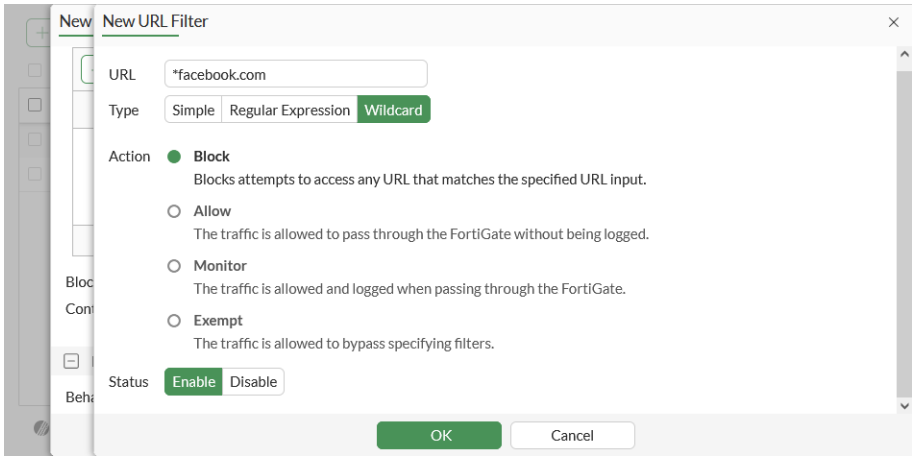
Option	Description
av	Antivirus scanning
web-content	Web filter content matching
activex-java-cookie	ActiveX, Java, and cookie filtering
dlp	DLP scanning
fortiguard	FortiGuard web filtering
range-block	Range block feature
pass	Pass single connection from all
antiphish	Antiphish credential checking
all	Exempt from all scanning and filtering operations listed above

These exempt options are not visible in the GUI. Setting the URL filter *Action* to *Exempt* will exempt URLs from all security profiles.

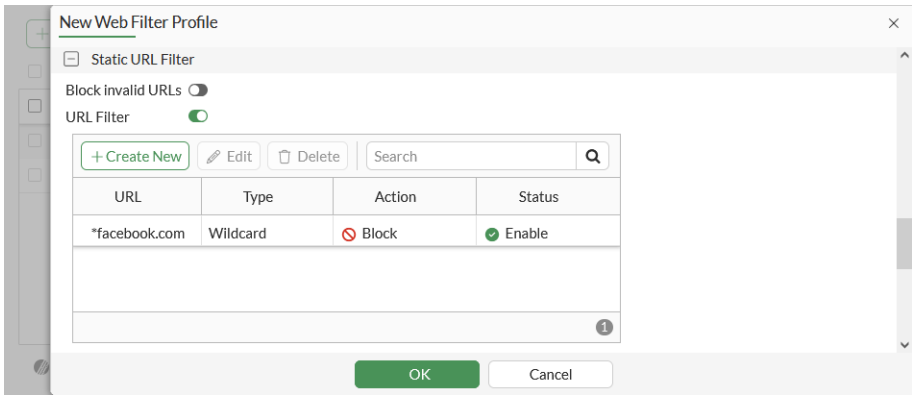
In the following example, a URL filter will be created to block the facebook.com URL using a wildcard.

To create a URL filter for Facebook in the GUI:

1. Go to *Security Profiles* > *Web Filter* and click *Create New*, or edit an existing profile.
2. In the *Static URL Filter* section, enable *URL Filter*.
3. Click *Create New*. The *New URL Filter* pane opens.
4. For URL, enter **facebook.com*, for *Type*, select *Wildcard*, and for *Action*, select *Block*.



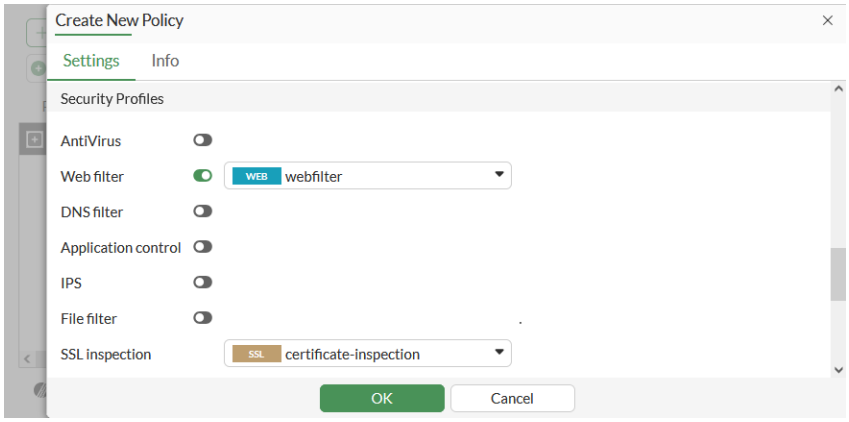
5. Click *OK*. The entry appears in the table.



6. Configure the other settings as needed.
7. Click *OK*.

To apply the web filter profile to a firewall policy in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Edit a policy, or create a new one.
3. In the *Security Profiles* section, enable *Web Filter* and select the profile that you created.
4. Set *SSL Inspection* to *certificate-inspection*.



> The *no-inspection* profile does not perform SSL inspection, so it should not be selected with other UTM profiles.

5. Configure the other settings as needed.
6. Click *OK*.

To create a URL filter for Facebook in the CLI:

```
config webfilter urlfilter
    edit 1
        set name "webfilter"
        config entries
            edit 1
                set url "*facebook.com"
```

```
                set type wildcard
                set action block
            next
        end
    next
end
```

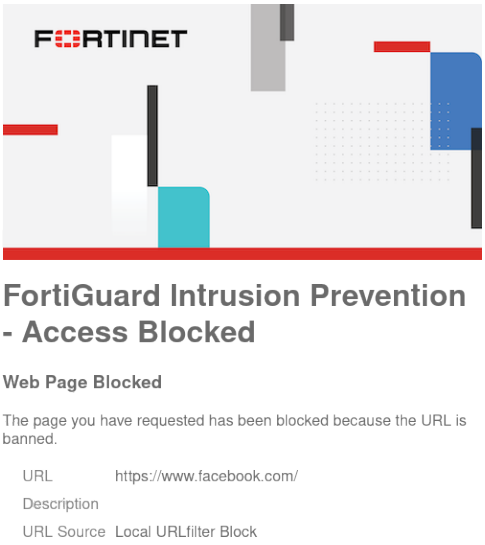
To apply the URL filter to a web filter profile in the CLI:

```
config webfilter profile
    edit "webfilter"
        config web
            set urlfilter-table 1
        end
        config ftgd-wf
            ...
        end
    next
end
```

To apply the web filter profile to a firewall policy in the CLI:

```
config firewall policy
    edit 1
        set name "WF"
        set srcintf "wan2"
        set dstintf "wan1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set logtraffic all
        set webfilter-profile "webfilter"
        set ssl-ssh-profile "certificate-inspection"
        set nat enable
    next
end
```

Verify the URL filter results by going to a blocked website. For example, when you go to the Facebook website, the replacement message appears:

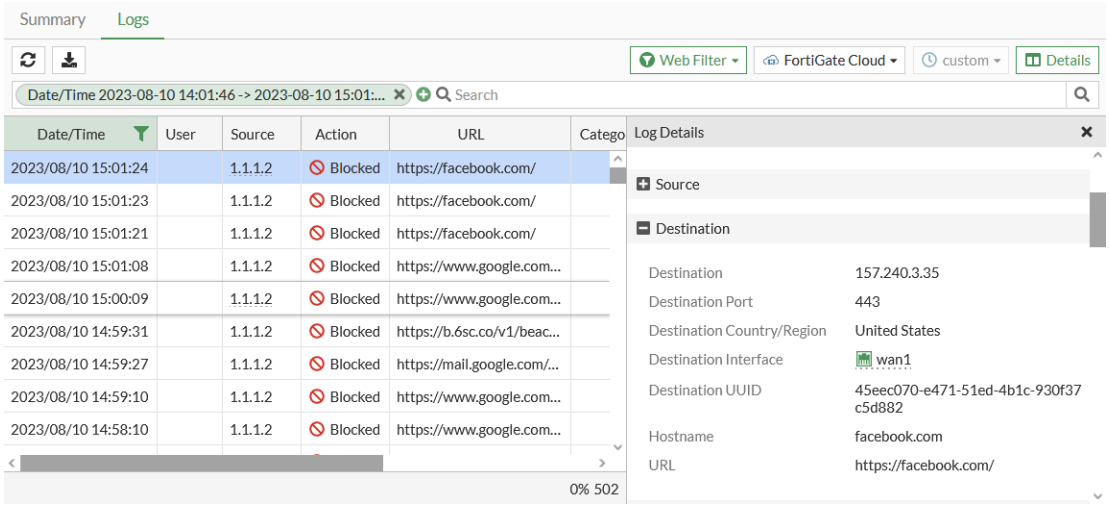


To customize the URL web page blocked message:

1. Go to *System* > *Replacement Messages*.
2. In the *HTTP* section, select *URL Block Page* and click *Edit*.
3. Edit the HTML to customize the message. See Replacement messages on page 3899 for more information.

To check web filter logs in the GUI:

1. Go to *Log & Report* > *Security Events*.
2. Click the *Web Filter* card name.
3. If there are a lot of log entries, click *Add Filter* and select *Event Type* > *urlfilter* to display logs generated by the URL filter.



To check web filter logs in the CLI:

```
# execute log filter category utm-webfilter
# execute log display

2: date=2023-08-10 time=15:02:25 eventtime=1691704944982929658 tz="-0700" logid="0315012544"
type="utm" subtype="webfilter" eventtype="urlfilter" level="warning" vd="root" urlfilteridx=1
urlfilterlist="webfilter" policyid=1 poluuid="4a4b9d00-e471-51ed-71ec-c1a3bc8f773c"
policytype="policy" sessionid=4198 srcip=1.1.1.2 srcport=55044 srccountry="Australia"
srcintf="internal7" srcintfrole="lan" srcuuid="45eec070-e471-51ed-4b1c-930f37c5d882"
dstip=157.240.3.35 dstport=443 dstcountry="United States" dstintf="wan1" dstintfrole="wan"
dstuuid="45eec070-e471-51ed-4b1c-930f37c5d882" proto=6 service="HTTPS" hostname="www.facebook.com"
profile="webfilter" action="blocked" reqtype="direct" url="https://www.facebook.com/" sentbyte=812
rcvdbyte=0 direction="outgoing" urlsource="Local URLfilter Block" msg="URL was blocked because it
is in the URL filter list" crscore=30 craction=8 crlevel="high"
```

Disabling simple type URL filter from blocking subdomains

By default, a URL filter of type Simple will block subdomains. For example, specifying *slack.com* will match *hooks.slack.com* as well as *help.slack.com*.

To disable the inclusion of subdomains:

```
config webfilter urlfilter
    edit <id>
        set include-subdomains {enable | disable}
    next
end
```

The set `include-subdomain` option only applies to Simple type entries.

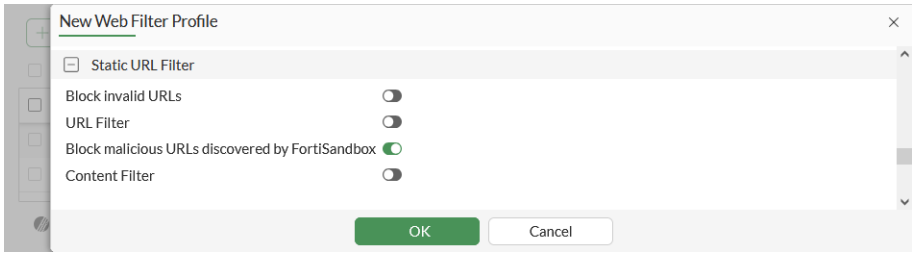
Block malicious URLs discovered by FortiSandbox

This setting blocks malicious URLs that FortiSandbox finds. Your FortiGate must be connected to a registered FortiSandbox.

For information on configuring FortiSandbox, see Using FortiSandbox post-transfer scanning with antivirus on page 2267 and Using FortiSandbox inline scanning with antivirus on page 2269.

To block malicious URLs discovered by FortiSandbox in the GUI:

1. Go to *Security Profiles* > *Web Filter* and click *Create New*, or edit an existing profile.
2. In the *Static URL Filter* section, enable *Block malicious URLs discovered by FortiSandbox*.



3. Click *OK*.

To block malicious URLs discovered by FortiSandbox in the CLI:

```
config webfilter profile
    edit "webfilter"
        config web
            set blocklist enable
        end
    next
end
```

Web content filter

You can control access to web content by blocking webpages containing specific words or patterns. This helps to prevent access to pages with questionable material. You can specify words, phrases, patterns, wildcards, and regular expressions to match content on webpages. You can use multiple web content filter lists and select the best one for each web filter profile.

The maximum number of web content patterns in a list depends on the model of the device. To find the maximum number of web content patterns allowed for a device, go to the Maximum Values Table (https://docs.fortinet.com/max-value-table). Select the software version and models, and click *Go*. Maximum values are displayed. In the *Search* box, enter *webfilter.content:entries* to find the maximum number.

When configuring a web content filter list, the following patterns are available:

Web content pattern type	Description
Wildcard	Use this setting to block or exempt one word or text strings of up to 80 characters. You can also use wildcard symbols such as *?* or * to represent one or more characters. For example, a wildcard expression *forti*.com* matches *fortinet.com* and *fortiguard.com*. The * represents any character appearing any number of times.
Regular expression	Use this setting to block or exempt patterns of regular expressions that use some of the same symbols as wildcard expressions, but for different purposes. In regular expressions, * represents the character before the symbol. For example, *forti*.com* matches *fortiii.com* but not *fortinet.com* or *fortiice.com*. In this case, the symbol * represents *i* appearing any number of times.

The web content filter scans the content of every webpage that is accepted by a firewall policy. The system administrator can specify banned words and phrases and attach a numerical value (or score) to the importance of those words and phrases. When the web content filter scan detects banned content, it adds the scores of banned words and phrases found on that page. If the sum is higher than a threshold set in the web filter profile, the FortiGate blocks the page.

The default score for web content filter is 10 and the default threshold is 10. This means that by default, a webpage is blocked by a single match. These settings can only be configured in the CLI.

Banned words or phrases are evaluated according to the following rules:

- The score for each word or phrase is counted only once, even if that word or phrase appears many times in the webpage.

- The score for any word in a phrase without quotation marks is counted.
- The score for a phrase in quotation marks is counted only if it appears exactly as written.

The following table is an example of how rules are applied to the webpage contents . For example, a webpage contains only this sentence:

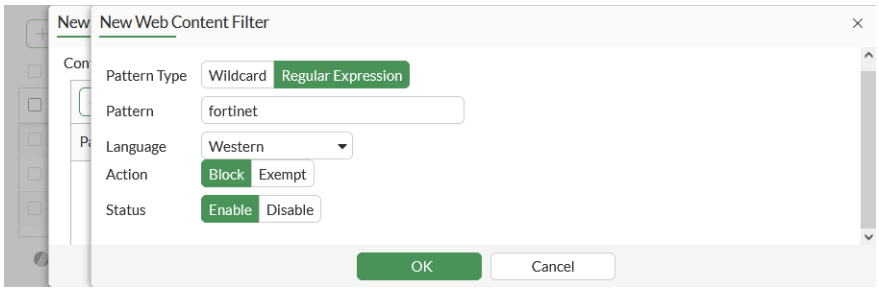
The score for each word or phrase is counted only once, even if that word or phrase appears many times in the webpage.

Banned pattern	Assigned score	Score added to the sum for the entire page	Threshold score	Comment
word	20	20	20	Appears twice but is only counted once. The webpage is blocked.
word phrase	20	40	20	Each word appears twice but is only counted once, giving a total score of 40. The webpage is blocked.
word sentence	20	20	20	*word* appears twice and *sentence* does not appear, but since any word in a phrase without quotation marks is counted, the score for this pattern is 20. The webpage is blocked.
"word sentence"	20	0	20	This phrase does not appear exactly as written. The webpage is allowed.
"word or phrase"	20	20	20	This phrase appears twice but is only counted once. The webpage is blocked.

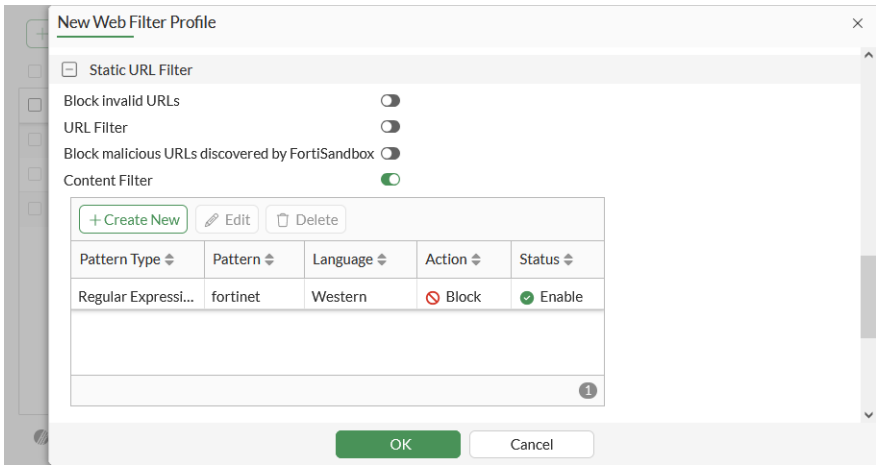
To configure a web content filter in the GUI:

1. Go to *Security Profiles* > *Web Filter* and click *Create New*, or edit an existing profile.
2. In the *Static URL Filter* section, enable *Content Filter*.
3. In the table, click *Create New*. The *New Web Content Filter* pane opens.
4. Configure the following settings:

Pattern Type	Regular Expression
Pattern	fortinet
Language	Western
Action	Block
Status	Enable



5. Click *OK*. The entry appears in the table.



6. Configure the other settings as needed.
7. Click *OK*.

To configure a web content filter in the CLI:

1. Create the content (banned word) table:

```
config webfilter content
   edit 1
      set name "webfilter"
      config entries
         edit "fortinet"
           set pattern-type regexp
           set status enable
           set lang western
           set score 10
           set action block
         next
      end
   next
end
```

2. Apply the content table to the web filter profile:

```
config webfilter profile
   edit "webfilter"
      config web
```

```
            set bword-threshold 10
            set bword-table 1
        end
        config ftgd-wf
            unset options
        end
    next
end
```

To verify the content filter:

1. Go to a website with the word *fortinet*, such as www.fortinet.com.
   The website is blocked and a replacement page displays:

Rating options

This topic gives examples of the following advanced filter features:

- Allow websites when a rating error occurs on page 2339
- Rate URLs by domain and IP address on page 2340

Allow websites when a rating error occurs

If you do not have a FortiGuard license, but you have enabled services that need a FortiGuard license (such as FortiGuard filter), then you will get a rating error message.

Use this setting to allow access to websites that return a rating error from the FortiGuard Web Filter service.

To allow websites with rating errors in the GUI:

1. Go to *Security Profiles* > *Web Filter* and click *Create New*, or edit an existing profile.
2. In the *Rating Options* section, enable *Allow websites when a rating error occurs*.
3. Click *OK*.

To allow websites with rating errors in the CLI:

```
config webfilter profile
    edit "webfilter"
        config ftgd-wf
            set options error-allow
        end
    next
end
```

Rate URLs by domain and IP address

If you enable this setting, in addition to only sending domain information to FortiGuard for rating, the FortiGate always sends both the URL domain name and the TCP/IP packet's IP address (except for private IP addresses) to FortiGuard for the rating.

The FortiGuard server might return a different category of IP address and URL domain. If they are different, the FortiGate uses the rating weight of the IP address or domain name to determine the rating result and decision. This rating weight is hard-coded in FortiOS.

For example, if we use a spoof IP of Google as www.irs.gov, the FortiGate will send both the IP address and domain name to FortiGuard to get the rating. We get two different ratings: one is the search engine and portals that belong to the Google IP, the second is the government and legal organizations that belongs to www.irs.gov. Because the search engine and portals rating has a higher weight than government and legal organizations, the traffic is rated as search engine and portals.

To rate URLs by domain and IP address in the GUI:

1. Go to *Security Profiles* > *Web Filter* and click *Create New*, or edit an existing profile.
2. In the *Rating Options* section, enable *Rate URLs by domain and IP address*.
3. Click *OK*.

To rate URLs by domain and IP address in the CLI:

```
config webfilter profile
    edit "webfilter"
        config ftgd-wf
            set options rate-server-ip
        end
    next
end
```

Proxy options

This topic gives examples of the following advanced filter features:

- Restrict Google account usage to specific domains on page 2341
- HTTP POST action on page 2341
- Remove Java applets, ActiveX, and cookies on page 2342

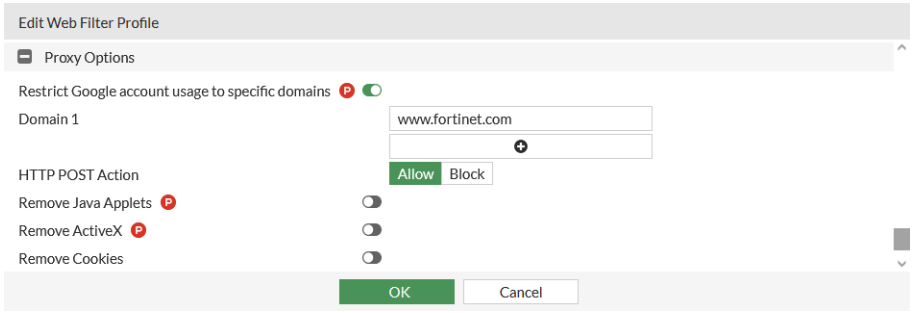
These advanced filters are only available in proxy-based inspection mode.

Restrict Google account usage to specific domains

Use this setting to block access to certain Google accounts and services, while allowing access to accounts with domains in the exception list.

To enable Google account restriction:

1. Go to *Security Profiles* > *Web Filter* and click *Create New*, or edit an existing profile.
2. In the *Proxy Options* section, enable *Restrict Google account usage to specific domains*.
3. Click the + and enter the domains that Google can access, such as www.fortinet.com.



4. Click *OK*.

When you try to use Google services like Gmail, only traffic from the domain of www.fortinet.com can go through. Traffic from other domains is blocked.

HTTP POST action

Use this setting to select the action to take with HTTP POST traffic. HTTP POST is the command used by the browser when you send information, such as a completed form or a file you are uploading to a web server. The action options are allow or block. The default is allow.

To configure HTTP POST in the GUI:

1. Go to *Security Profiles* > *Web Filter* and click *Create New*, or edit an existing profile.
2. In the *Proxy Options* section, for *HTTP POST Action*, select *Allow* or *Block*.
3. Click *OK*.

To configure HTTP POST in the CLI:

```
config webfilter profile
    edit "webfilter"
        set post-action {normal | block}
        config ftgd-wf
            unset options
        end
```

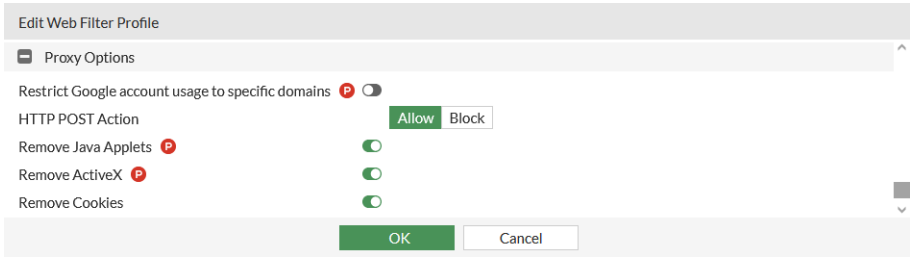
```
    next
end
```

Remove Java applets, ActiveX, and cookies

Web filter profiles have settings to filter Java applets, ActiveX, and cookies from web traffic. Note that if these filters are enabled, websites using Java applets, ActiveX, and cookies might not function properly.

To enable these filters in the GUI:

1. Go to *Security Profiles* > *Web Filter* and click *Create New*, or edit an existing profile. and go to the *Proxy Options* section.
2. In the *Proxy Options* section, enabled the filters you want to use: *Remove Java Applets*, *Remove ActiveX*, or *Remove Cookies*.



To enable these filters in the CLI:

```
config webfilter profile
   edit "webfilter"
      set options {activexfilter cookiefilter javafilter}
      config ftgd-wf
         unset options
      end
   next
end
```

Advanced CLI configuration

This section includes the following:

- Credential phishing prevention on page 2343
- Additional antiphishing settings on page 2345
- Web filter statistics on page 2349
- URL certificate blocklist on page 2350
- Websense Integrated Services Protocol on page 2351
- Inspecting HTTP3 traffic on page 2352
- Warning prompt per domain on page 2353
- Using URL risk-scores in determining policy action on page 2354

Credential phishing prevention

When credential phishing prevention is enabled, the FortiGate scans for corporate credentials submitted to external websites and compares them to sensitive credentials stored in the corporate domain controller. Based on the configured antiphishing rules in proxy mode web filter profiles, the FortiGate will block the URL or alert the user if the credentials match ones that are stored on the corporate domain controller.

- The corporate domain controller must be configured in the `domain controller`.
- For FortiGate to retrieve the domain information, the user needs to grant Replicating Directory Changes permissions in the Domain Controller (DC). See How to grant the "Replicating Directory Changes" permission for the Microsoft Metadirectory Services ADMA service account for more information.
- Credentials can be matched based on sAMAccountName, user principal name (UPN), or down-level logon name.
- The antiphishing profile defines the corporate domain controller, antiphishing check option, default action if no rules match, antiphishing status, and so on.
- Inspection entries in the profile define what action occurs when the submission request matches the specified FortiGuard categories.
- The profile scans for predefined and custom username and password fields in the HTTP request, such as `username`, `auth`, and `password`. You can evaluate custom fields by configuring custom patterns.
- The URL filter defines individual URLs that the antiphish action (block or log) is applied to when the URL submission request matches.

Web-based URL filter actions and FortiGuard category-based filtering have higher priority than antiphishing URL filter actions and FortiGuard filtering:

- If a request is blocked by the web-based URL filter or FortiGuard filter, there is no further antiphishing scanning. Antiphishing scanning only happens after the web-based URL filtes and FortiGuard filters allow the traffic.
- If a submission matches an entry in the URL filter table that has an antiphishing action, the defined action is taken. No further FortiGuard category-based rules are applied.
- Like firewall rules, the URL filter table and Fortiguard category-based antiphishing rules use a top-down priority. The rule that matches first is the one that is used.

In this example, URLs that match FortiGuard category 37 (social networking) will be blocked and other categories will be logged.

To configure credential phishing prevention:

1. Configure the corporate domain controller:

```
config user domain-controller
    edit "win2016"
        set hostname "win2016"
        set domain-name "corpserver.local"
        set username "Administrator"
        set password **********
        set ip <server_ip>
```

```
    next
end
```

The hostname and the domain-name are case sensitive.

2. Configure the antiphishing profile, which includes the FortiGuard category rule:

```
config webfilter profile
    edit <profile-name>
        set feature-set proxy
        ...
        config web
            ...
        end
        config antiphish
            set status enable
            set domain-controller "win2016"
            set default-action block
            set check-uri enable
            set check-basic-auth enable
            set max-body-len 65536
            config inspection-entries
                edit "inspect-37"
                    set fortiguard-category 37
                    set action block
                next
                edit "inspect-others"
                    set fortiguard-category all
                    set action log
                next
            end
            config custom-patterns
                edit "customer-name"
                    set category username
                next
                edit "customer-passwd"
                   set category password
                next
            end
        end
        ...
        set web-antiphishing-log enable
    next
end
```

- `check-uri` enables support for scanning HTTP GET URI parameters.
- `check-basic-auth` enables support for scanning the HTTP basic authentication field.

3. Configure the URL filter to scan specific URLs.

   The antiphish action is added to the URL filter table entry, and the URL filter is applied to the web filter profile:

```
config webfilter urlfilter
    edit 1
        set name "antiphish-table"
        config entries
            edit 1
                set url "www.example.com"
                set type simple
                set antiphish-action block
                set status enable
                set referrer-host ''
            next
        end
    next
end
config webfilter profile
    edit "<profile-name>"
        config web
            set urlfilter-table 1
        end
        ...
    next
end
```

4. Optionally, define custom patterns to scan fields other than the built-in username and password keywords:

```
config webfilter profile
    edit "<profile-name>"
        config custom-patterns
                edit "customer-name"
                    set category username
                next
                edit "customer-passwd"
                   set category password
                next
            end
        end
    next
end
```

Additional antiphishing settings

The following settings are available for antiphishing:

- Enable DNS service lookup in the domain controller so that the domain controller IP does not need to be configured. The DNS server will resolve the domain controller IP.
- Specify a source IP or port for the fetching domain controller.
- Use an LDAP server as a credential source (only the OpenLDAP server is supported).
- Block or log valid usernames regardless of password match.
- Use literal custom patterns type for username and password.
- Active Directory Lightweight Directory Services (AD LDS) support

Configuration examples

To enable DNS service lookup:

```
config user domain-controller
    edit "win2016"
        set ad-mode ds
        set dns-srv-lookup enable
        set hostname "win2016"
        set username "replicate"
        set password **********
        set domain-name "SMB2016.LAB"
    next
end
```

To specify the source IP and port for the fetching domain controller:

```
config user domain-controller
    edit "win2016"
        set ad-mode ds
        set hostname "win2016"
        set username "replicate"
        set password **********
        set ip-address 172.18.52.188
        set source-ip-address 172.16.100.1
        set source-port 2000
        set domain-name "SMB2016.LAB"

    next
end
```

To use an LDAP server as a credential store:

1. Configure the LDAP server:

```
config user ldap
    edit "openldap"
        set server "172.18.60.214"
        set cnid "cn"
        set dn "dc=qafsso,dc=com"
        set type regular
        set username "cn=Manager,dc=qafsso,dc=com"
        set password **********
        set antiphish enable
        set password-attr "userPassword"
    next
end
```

2. Configure the web filter profile:

```
config webfilter profile
    edit "webfilter"
        set feature-set proxy
        config ftgd-wf
            unset options
            config filters
                edit 1
                    set action block
                next
            end
        end
        config antiphish
            set status enable
            config inspection-entries
                edit "cat34"
                    set fortiguard-category 34
                    set action block
                next
            end
            set authentication ldap
            set ldap "openldap"
        end
        set log-all-url enable
    next
end
```

To configure username-only credential matching:

```
config webfilter profile
    edit "webfilter"
        set feature-set proxy
        config ftgd-wf
            unset options
            ...
        end
        config antiphish
            set status enable
            set check-username-only enable
            config inspection-entries
                edit "cat34"
                    set fortiguard-category 34
                    set action block
                next
            end
            set domain-controller "win2016"
        end
        set log-all-url enable
    next
end
```

To configure different custom pattern types for usernames and passwords:

```
config webfilter profile
    edit "webfilter"
        set feature-set proxy
        config ftgd-wf
            unset options
            ...
        end
        config antiphish
            set status enable
            config inspection-entries
                edit "cat34"
                    set fortiguard-category 34
                    set action block
                next
            end
            config custom-patterns
                edit "qwer"
                    set type literal
                next
                edit "[0-6]Dat*"
                next
                edit "dauw9"
                    set category password
                    set type literal
                next
                edit "[0-5]foo[1-4]"
                    set category password
                next
            end
            set domain-controller "win2016"
        end
        set log-all-url enable
    next
end
```

In this example, the `qwer` and `dauw9` entries use the literal type, while `[0-6]Dat*` and `[0-5]foo[1-4]` use the default regex type.

To configure Active Directory in LDS mode:

```
config user domain-controller
    edit "win2016adlds"
        set hostname "win2016adlds"
        set username "foo"
        set password **********
        set ip-address 192.168.10.9
        set domain-name "adlds.local"
        set ad-mode lds
        set adlds-dn "CN=adlds1part1,DC=ADLDS,DC=COM"
        set adlds-ip-address 192.168.10.9
```

```
        set adlds-port 3890
    next
end
```

Web filter statistics

FortiOS provides diagnostics commands to view web filter statistics reports, which are either proxy-based or flow-based. The commands are available in both VDOM and global command lines.

Proxy-based web filter statistics report

Use the `diagnose wad filter vd {<VDOM> | global}` command to filter for per-VDOM or global statistics reports.

In the following example, there are two VDOMs (root and vdom1) using proxy-based policies that have web filter profiles enabled.

To view per-VDOM statistics reports:

```
(global) # diagnose wad filter vd root
Drop_unknown_session is enabled.
```

```
(global) # diagnose wad stats filter list
filtering of vdom root
  dlp          = 0
  content-type = 0
  urls:
    examined = 6
    allowed = 3
    blocked = 0
    logged = 0
    overridden = 0
```

```
(global) # diagnose wad filter vd vdom1
(global) # diagnose wad stats filter list
filtering of vdom vdom1
  dlp          = 0
  content-type = 0
  urls:
    examined = 13
    allowed = 2
    blocked = 9
    logged = 8
    overridden = 0
```

```
(global) # diagnose wad filter vd ALL
(global) # diagnose wad stats filter list
filtering of all accessible vdoms
  dlp          = 0
```

```
  content-type = 0
  urls:
    examined = 19
    allowed = 5
    blocked = 9
    logged = 8
    overridden = 0
```

Flow-based web filter statistics report

Use the `diagnose webfilter stats list {<VDOM> | global}` command to check the flow-based web filter statistics.

In the following example, the VDOM is using flow-based policies that have web filter profiles enabled.

To view web filter statistics:

```
# diagnose webfilter stats list root
Proxy/flow URL filter stats:
request:   9474
blocked:   8606
allowed:   868
overridden:0
logged:    8606
pending:   0
```

URL certificate blocklist

As increasing numbers of malware have started to use SSL to attempt to bypass IPS, maintaining a fingerprint-based certificate blocklist is useful to block botnet communication that relies on SSL.

This feature adds a dynamic package that is distributed by FortiGuard and is part of the Web Filtering service. It is enabled by default for SSL/SSH profiles, and can be configured using the following CLI commands:

```
config vdom
    edit <vdom>
        config firewall ssl-ssh-profile
            edit "certificate-inspection"
                set block-blocklisted-certificates enable
            next
            edit "deep-inspection"
                set block-blocklisted-certificates enable
            next
        end
    next
end
```

Websense Integrated Services Protocol

Websense Integrated Services Protocol (WISP) is supported on the FortGate, which allows the firewall to send traffic to the third-party web filtering service for rating and approval checking.

When WISP is enabled, the FortiGate maintains a pool of TCP connections to the WISP server. The TCP connections are used to forward HTTP request information and log information to the WISP server and receive policy decisions.

When a WISP server is used in a web filter profile, in flow or proxy mode, the following web filter scanning priority sequence is used:

1. Local URL filter
2. Websense web filtering service
3. FortiGuard web filtering service

The following example uses a WISP server configured in a flow mode web filter profile.

To use a WISP server in flow mode:

1. Configure the WISP servers:

```
config web-proxy wisp
    edit "wisp1"
        set server-ip 10.2.3.4
    next
    edit "wisp2"
        set server-ip 10.2.3.5
    next
    edit "wisp3"
        set server-ip 192.168.1.2
    next
    edit "wisp4"
        set server-ip 192.168.3.4
    next
end
```

2. Configure the web filter profile:

```
config webfilter profile
    edit "webfilter_flowbase"
        set feature-set flow
        config ftgd-wf
            unset options
            config filters
                edit 64
                    set category 64
                    set action block
                next
            end
        end
        set wisp enable
        set wisp-servers "wisp1" "wisp2"
```

```
        set wisp-algorithm {primary-secondary | round-robin | auto-learning}
        set log-all-url enable
    next
end
```

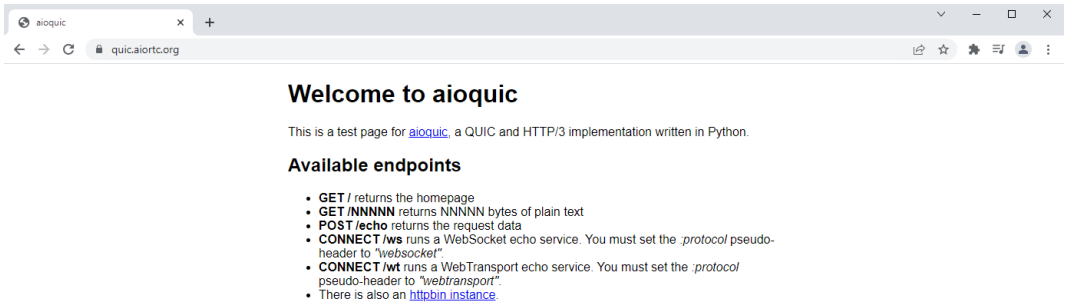
Inspecting HTTP3 traffic

HTTP/3 traffic can be inspected on the FortiGate in flow mode inspection.

When using Chrome, the browser may switch the HTTP/3 connection to HTTP/2 when deep inspection is applied, due to its sensitivity to delays caused by deep inspection.

Example

In this example, a web filter profile is created to block the words *Welcome to aioquic*, which appear in a website that uses HTTP/3.



To block content in HTTP/3 traffic:

1. Configure the web filter banned word table:

```
config webfilter content
    edit 1
        set name "aioquic"
        config entries
            edit "Welcome to aioquic"
                set status enable
            next
        end
    next
end
```

2. Apply the banned word table in the web filter profile:

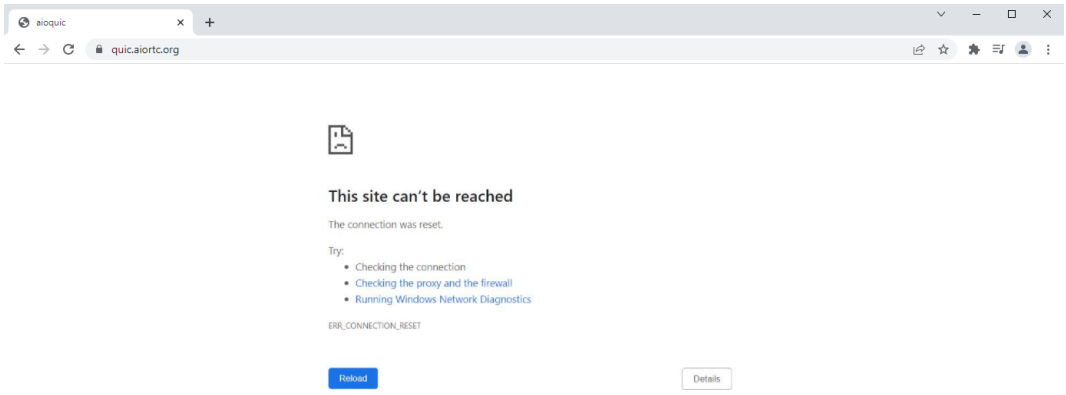
```
config webfilter profile
    edit "flow-webfilter"
        config web
            set bword-table 1
```

```
        end
        config ftgd-wf
            unset options
        end
    next
end
```

3. Configure the firewall policy:

```
config firewall policy
    edit 1
        set utm-status enable
        set ssl-ssh-profile "deep-inspection"
        set webfilter-profile "flow-webfilter"
        set logtraffic all
        set nat enable
    next
end
```

4. Access the website using a supported HTTP/3 client, such as Chrome or Firefox. The website is blocked by the FortiGate.



Warning prompt per domain

When you are configuring web filter profiles and setting the action to warn for a web filter category, the default action is to allow access to that category for the configured time. See Web profile override on page 2747.

Alternatively, the CLI may be used to set the presentation of a warning prompt for individual domains within a given category. By requiring users to acknowledge the desire to proceed on each domain within a category, subsequent visits to domains in the same category within the warn-duration provide the same experience as the initial domain. They are not transparently allowed and maintain a consistent experience for the category.

This feature requires deep inspection. Using certificate inspection is not supported and will provide inconsistent results and logs.

To configure a warning prompt:

```
config webfilter profile
    edit <web_filter_name>
        config ftgd-wf
            config filters
                edit <filter_id>
                    set category <category_id>
                    set action warning
                    set warning-prompt per-domain
                next
            end
        end
    next
end
```

Using URL risk-scores in determining policy action

Risk level ratings can be added to the FortiGuard URL rating service. A FortiGate can query the rating service to retrieve the risk score for a URL. This risk score rates the likelihood that a website has malicious intent. It combines the results of machine learning models and human analysis together to predict how likely a given URL is malicious.

The risk score returned from FortiGuard is a value from 0-100, where:

Risk Score	Risk Level	Description
91-100	High	Strong confidence of malicious intent.
71-90	Suspicious	Medium confidence of malicious intent.
51-70	Moderate	Generally benign with a potential risk of attack.
21-50	Low	Low predictive risk of attack.
1-20	Trustworthy	Very low predictive risk of attack.
0	Unrated	The URL does not exist in FortiGuard DB or the risk score of the URL is unknown.

The FortiGate can utilize this risk score and risk level in two different ways.

1. In a web filter profile, a risk level can be associated with the action *Block* or *Monitor*. When traffic hits a policy with the web filter profile applied, the URL will be used to query the FortiGuard URL rating service. If the risk score matches a level defined in the profile, the action is taken on that website.

   The firewall policy must use proxy-based inspection. Either certificate or deep inspection will work with this feature.

2. In an explicit or transparent web proxy, a proxy-policy can be configured with a risk level. The risk level becomes a matching criteria for the policy.

Furthermore, the risk score range associated with each predefined level above cannot be modified on the FortiGate. However, new risk levels can be created with a custom range. These risk levels can be used within a web filter profile or within a proxy-policy.

Tie-breaker

In a web filter profile, when both risk level and web filter category are used, the action for the matched risk level will be ranked against the action for the matched web category.

Actions have the following weight order:

1. Block
2. Warning (Authenticate)
3. Monitor
4. Allow

When the action resulting from the risk level query and web filter category query return different actions, the action higher in the weight order will be performed. For example, a block and a warning action will result in the page being blocked.

When the actions returned are the same, then that action will be applied, and the replacement message and UTM log will indicate the decision was made by web filter category check.

Finally, if multiple risk levels are matched within a web filter profile, the action that has the higher weight will be applied.

Risk score override

A risk score can be overridden by a local risk-score override value. This override applies to a single URL specified in the object name.

CLI Syntax

The follow CLI syntax are included for introducing URL risk-scores to determine policy action:

- Risk level configuration in a web filter profile:

```
config webfilter profile
    edit <name>
        set feature-set proxy
        config ftgd-wf
            unset options
            config risk
                edit <id>
                    set risk-level <pre-defined or custom level>
                    set action {block | monitor}
                    set log {enable | disable}
                next
            end
        end
    next
end
```

- Built-in read-only web filter local FortiGuard risk-level definition:

```
config webfilter ftgd-risk-level
    edit "high"
        set high 100
        set low 91
    next
    edit "suspicious"
        set high 90
        set low 71
    next
    edit "moderate"
        set high 70
        set low 51
    next
    edit "low"
        set high 50
        set low 21
    next
    edit "trustworthy"
        set high 20
        set low 1
    next
    edit "unrated"
        set high 0
        set low 0
    next
end
```

- Custom user-defined web filter local FortiGuard risk-level:

```
config webfilter ftgd-risk-level
    edit <name>
        set high <score>
        set low <score>
    next
end
```

- Web filter local FortiGuard risk-score override:

```
config webfilter ftgd-local-risk
    edit <url>
        set status {enable | disable}
        set comment <string>
        set risk-score <score>
    next
end
```

- Applying risk-level to proxy-policy match:

```
config firewall proxy-policy
    edit <id>
        set url-risk <pre-defined or custom level>
```

```
    next
end
```

Examples

The following examples demonstrate applying risk levels in different scenarios.

Two demo URLs are used for the examples:

Demo URL	Risk score	Default risk level
www.example.com	58	Moderate
www.httpbin.org	46	Low

In the web filter examples, the profile is applied to a firewall policy that utilizes proxy-based inspection and deep inspection.

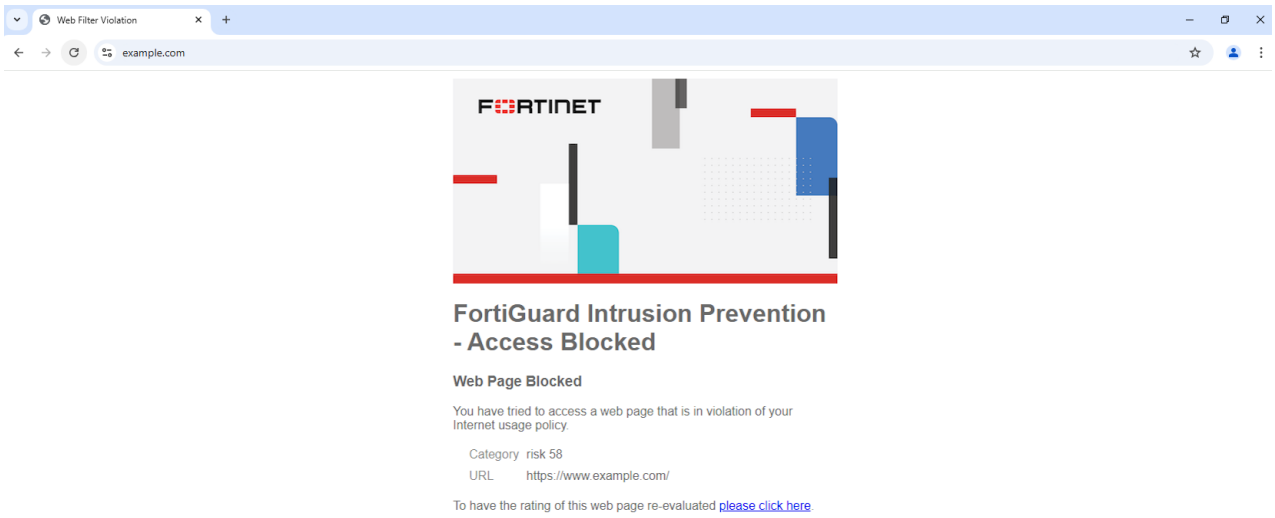
Example 1: Applying the action block to the moderate risk level

To apply the action block to the moderate risk level:

```
config webfilter profile
    edit "webfilter"
        set feature-set proxy
        config ftgd-wf
            unset options
            config risk
                edit 1
                    set risk-level "moderate"
                    set action block
                next
            end
        end
    next
end
config firewall policy
    edit 1
        set name "WF"
        set srcintf "port2"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set profile-protocol-options "protocol"
        set ssl-ssh-profile "protocols"
        set webfilter-profile "webfilter"
        set logtraffic all
```

```
        set nat enable
    next
end
```

When a client accesses www.example.com, the URL is blocked.



The following UTM log can be observed from CLI:

```
# execute log filter category utm-webfilter
# execute log display
1: date=2024-11-20 time=09:45:19 eventtime=1732124719100876715 tz="-0800" logid="0316013058"
type="utm" subtype="webfilter" eventtype="ftgd_blk" level="warning" vd="vdom1" policyid=1
poluuid="717176ae-97e7-51ef-2203-1f0b21e90462" policytype="policy" sessionid=6273
srcip=10.1.100.123 srcport=61545 srccountry="Reserved" srcintf="port2" srcintfrole="undefined"
srcuuid="8770b128-97cd-51ef-63d8-082966d66657" dstip=93.184.215.14 dstport=443
dstcountry="Belgium" dstintf="port1" dstintfrole="undefined" dstuuid="8770b128-97cd-51ef-63d8-
082966d66657" proto=6 httpmethod="GET" service="HTTPS" hostname="www.example.com"
agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KH" profile="webfilter"
action="blocked" reqtype="direct" url="https://www.example.com/" sentbyte=4559 rcvdbyte=6070
direction="outgoing" msg="URL belongs to a denied risk-level in policy" ratemethod="domain"
cat=255 urlrisk=58
```

Example 2: Overriding the URL's FortiGuard risk score with local risk score

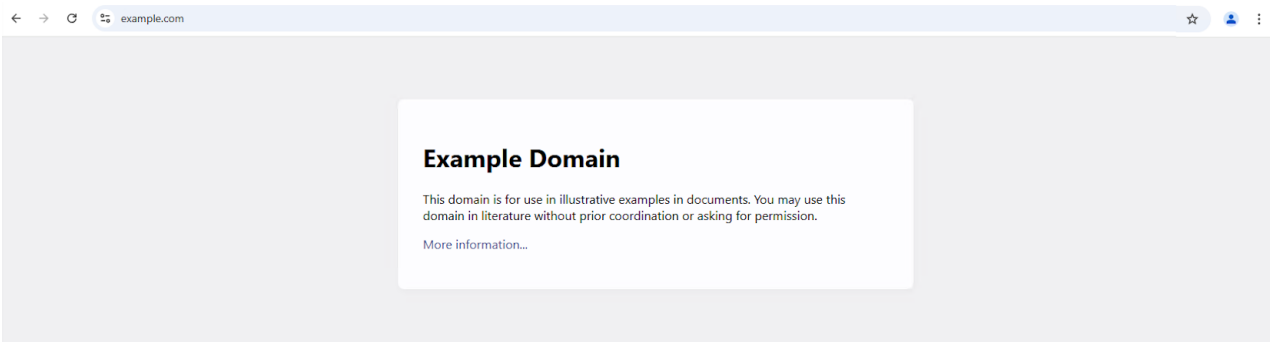
To override the URL's FortiGuard risk score with local risk score:

```
config webfilter ftgd-local-risk
    edit "www.example.com"
        set risk-score 30
    next
end
config webfilter profile
    edit "webfilter"
        set feature-set proxy
```

```
        config ftgd-wf
            unset options
            config risk
                edit 1
                    set risk-level "moderate"
                    set action block
                next
                edit 2
                    set risk-level "low"
                next
            end
        end
        set log-all-url enable
    next
end
```

The `low` risk-level is added to the web filter profile, with the action monitor.

When a client accesses www.example.com, the URL is allowed.



The following UTM log can be observed from CLI:

```
1: date=2024-11-20 time=11:25:55 eventtime=1732130754321650374 tz="-0800" logid="0317013313"
type="utm" subtype="webfilter" eventtype="ftgd_allow" level="notice" vd="vdom1" policyid=1
poluuid="717176ae-97e7-51ef-2203-1f0b21e90462" policytype="policy" sessionid=7648
srcip=10.1.100.123 srcport=61943 srccountry="Reserved" srcintf="port2" srcintfrole="undefined"
srcuuid="8770b128-97cd-51ef-63d8-082966d66657" dstip=93.184.215.14 dstport=443
dstcountry="Belgium" dstintf="port1" dstintfrole="undefined" dstuuid="8770b128-97cd-51ef-63d8-
082966d66657" proto=6 httpmethod="GET" service="HTTPS" hostname="www.example.com"
agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KH" profile="webfilter"
action="passthrough" reqtype="direct" url="https://www.example.com/" sentbyte=3118 rcvdbyte=4927
direction="outgoing" msg="URL belongs to an allowed risk-level in policy" ratemethod="domain"
cat=255 urlrisk=30
```

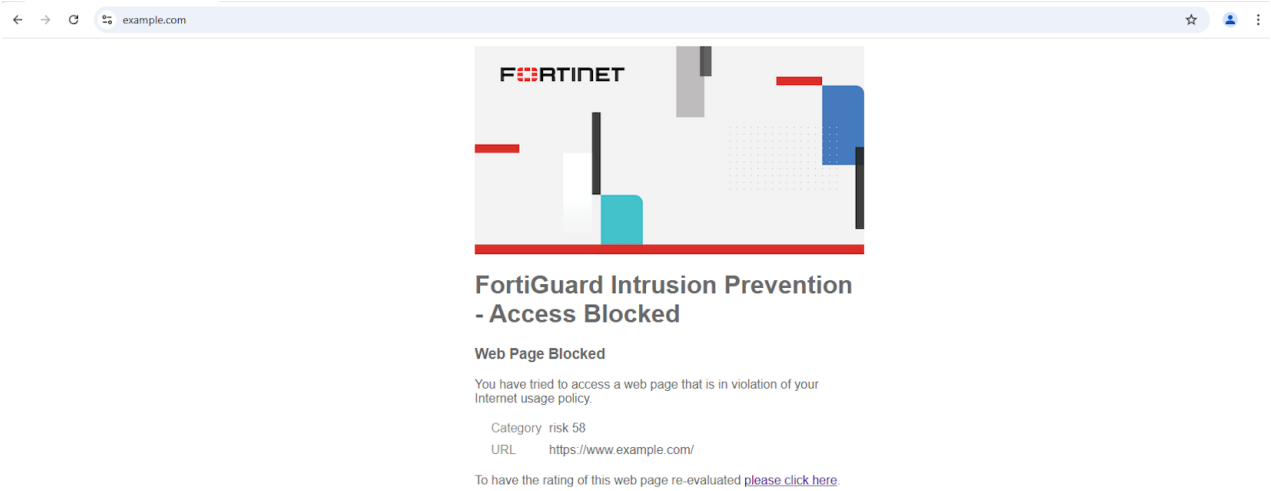
Example 3: Applying a user-defined risk level to the web filter profile

To apply a user-defined risk level to the web filter profile:

```
config webfilter ftgd-risk-level
    edit "medium-safety"
        set high 60
```

```
        set low 30
    next
end
config webfilter profile
    edit "webfilter"
        set feature-set proxy
        config ftgd-wf
            unset options
            config risk
                edit 1
                    set risk-level "medium-safety"
                    set action block
                next
            end
        end
        set log-all-url enable
    next
end
```

When a client accesses www.example.com, the URL is blocked.



The following UTM log can be observed from CLI:

```
1: date=2024-11-20 time=11:58:46 eventtime=1732132726031604734 tz="-0800" logid="0316013058"
type="utm" subtype="webfilter" eventtype="ftgd_blk" level="warning" vd="vdom1" policyid=1
poluuid="717176ae-97e7-51ef-2203-1f0b21e90462" policytype="policy" sessionid=8028
srcip=10.1.100.123 srcport=62051 srccountry="Reserved" srcintf="port2" srcintfrole="undefined"
srcuuid="8770b128-97cd-51ef-63d8-082966d66657" dstip=93.184.215.14 dstport=443
dstcountry="Belgium" dstintf="port1" dstintfrole="undefined" dstuuid="8770b128-97cd-51ef-63d8-
082966d66657" proto=6 httpmethod="GET" service="HTTPS" hostname="www.example.com"
agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KH" profile="webfilter"
action="blocked" reqtype="direct" url="https://www.example.com/" sentbyte=2962 rcvdbyte=4927
direction="outgoing" msg="URL belongs to a denied risk-level in policy" ratemethod="domain"
cat=255 urlrisk=58
```

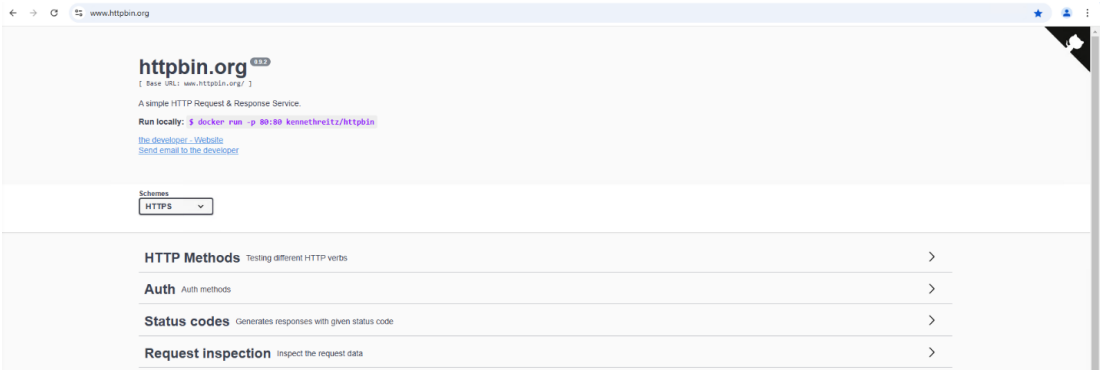
Example 4: Matching an explicit web proxy policy by the URL's risk level

A proxy-policy is configured to allow URLs that have low risk level.

To match an explicit web proxy policy by the URL's risk level:

```
config firewall proxy-policy
    edit 1
        set name "Test"
        set proxy explicit-web
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set url-risk "low"
        set service "webproxy"
        set action accept
        set schedule "always"
        set logtraffic all
        set ssl-ssh-profile "protocols"
        set log-http-transaction enable
    next
end
```

When a client accesses www.httpbin.org, the URL is allowed.



Traffic log indicates the traffic matched proxy-policy 1:

```
# execute log filter category traffic
# execute log display
2: date=2024-11-20 time=14:14:43 eventtime=1732140883609560183 tz="-0800" logid="0006000026"
type="traffic" subtype="http-transaction" level="notice" vd="vdom1" srcip=10.1.100.123
srcport=64445 dstip=52.20.148.183 dstport=443 sessionid=1798921931 transid=150995356
action="accept" policyid=1 policytype="proxy-policy" poluuid="7b6afd8e-a78b-51ef-0ae5-
5ddbd0defd25" url="https://www.httpbin.org/" agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36" appcat="unscanned"
duration=1 reqlength=474 resplength=9751 rcvdbyte=15379 sentbyte=2809 scheme="https"
hostname="www.httpbin.org" resptype="normal" httpmethod="GET" statuscode="200" reqtime=1732140882
resptime=1732140883 respfinishtime=1732140883
```

When a client accesses www.example.com, no proxy policy is matched. The URL is blocked.

Traffic log indicates the traffic is denied:

```
2: date=2024-11-20 time=14:19:17 eventtime=1732141157742504064 tz="-0800" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="vdom1" srcip=10.1.100.123 srcport=64593
srcintf="port2" srcintfrole="undefined" dstip=93.184.215.14 dstport=443 dstintf="port1"
dstintfrole="undefined" srccountry="Reserved" dstcountry="Belgium" sessionid=1798922057 proto=6
action="deny" policyid=0 policytype="proxy-policy" service="HTTPS" trandisp="noop"
url="https://www.example.com/" agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/131.0.0.0 Safari/537.36" appcat="unscanned" duration=0 sentbyte=222
rcvdbyte=0 sentpkt=0 rcvdpkt=0 crscore=30 craction=131072 crlevel="high" msg="Traffic denied
because of explicit proxy policy"
```

Configuration examples

This section includes the following configuration examples:

- Configuring web filter profiles with Hebrew domain names on page 2362
- Configuring web filter profiles to block cryptocurrency on page 2367
- Configuring web filter to exempt URLs from other security profiles on page 2369

Configuring web filter profiles with Hebrew domain names

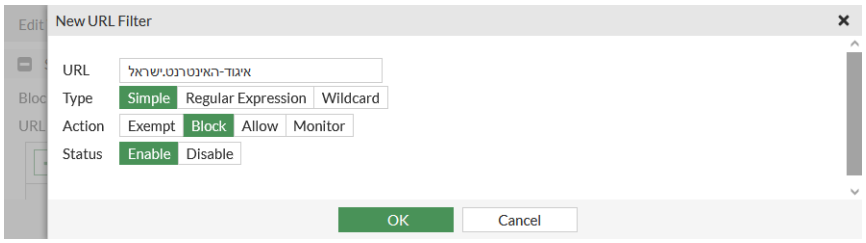
The domain name URLs in web filter profiles can be configured with non-ASCII characters, such as in Hebrew. Any configured domain name in non-ASCII characters is encoded into Punycode format, then the domain name in Punycode format is used to match the domain name in the HTTP request for URL filtering purposes.

In the following example, a Hebrew URL (`אינטרנט-אגוד.ישראל`) is blocked in by a static URL filter. The URL translates to:

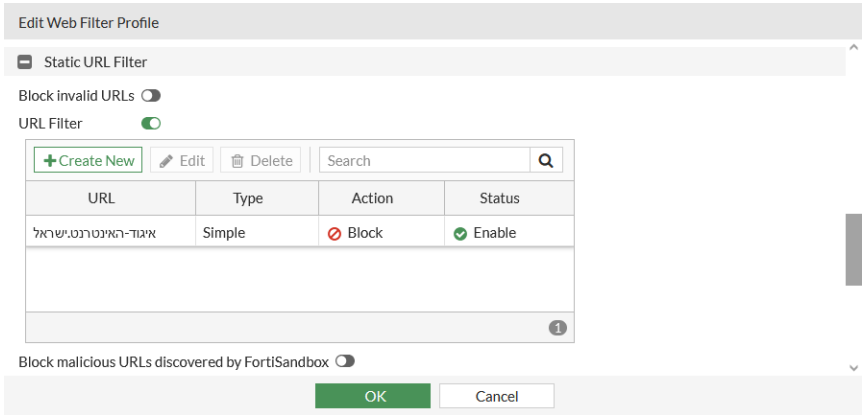
- `xn----zhcbgfhe2aacg8fb5i.xn--4dbrk0ce` in Punycode
- `en.isoc.org.il` in English

To configure the web filter profile in the GUI:

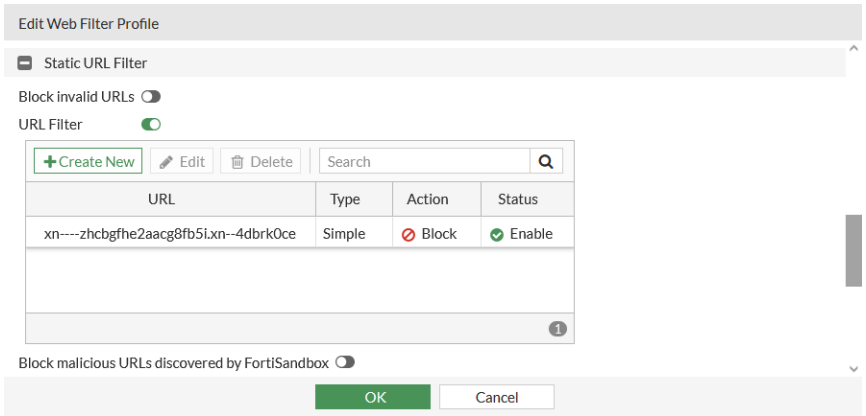
1. Go to *Security Profiles* > *Web Filter* and click *Create New*.
2. Enter a profile *Name*.
3. In the *Static URL Filter* section, enable *URL Filter* and click *Create New*.
4. Enter the Hebrew URL and set the *Action* to *Block*. The URL can be entered in Hebrew.



5. Click *OK* to save the filter. The URL appears in Hebrew in the URL filter table.



6. Click *OK* to save the web filter profile.
7. Edit the web filter profile. The URL in the table has been converted by the FortiGate into Punycode.



In the CLI, Punycode must be used to configure the Hebrew URL.

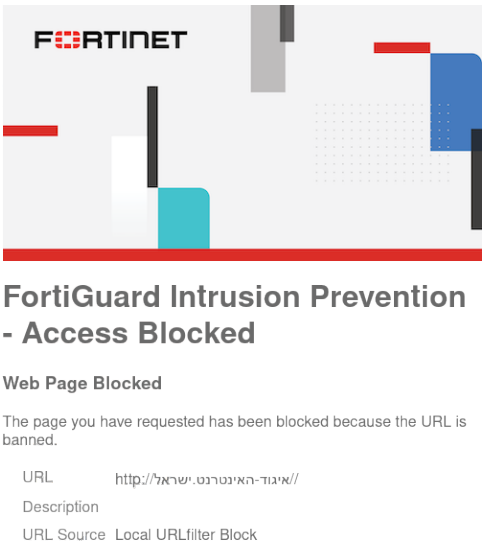
To configure the web filter profile in the CLI:

```
config webfilter urlfilter
    edit 1
        set name "Auto-webfilter-urlfilter_0wedo5f1c"
        config entries
            edit 1
                set url "xn----zhcbgfhe2aacg8fb5i.xn--4dbrk0ce"
                set action block
            next
```

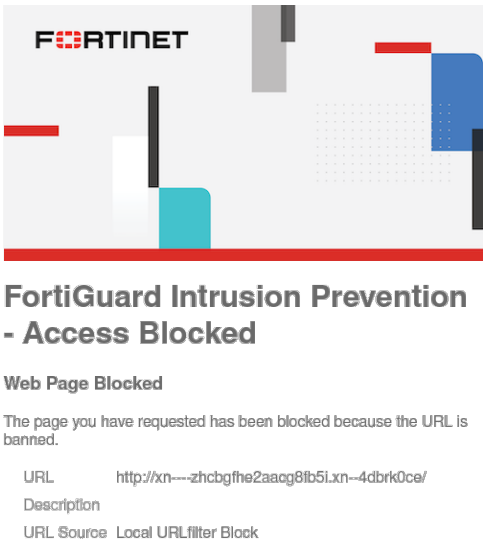
```
        end
    next
end
```

To verify the configuration:

1. From a client, access the Hebrew URL over HTTPS. The website is blocked by the FortiGate.
2. The content of the replacement message displayed in the browser depends on the inspection mode.
   - In flow mode (current configuration), the URL is displayed in Hebrew.



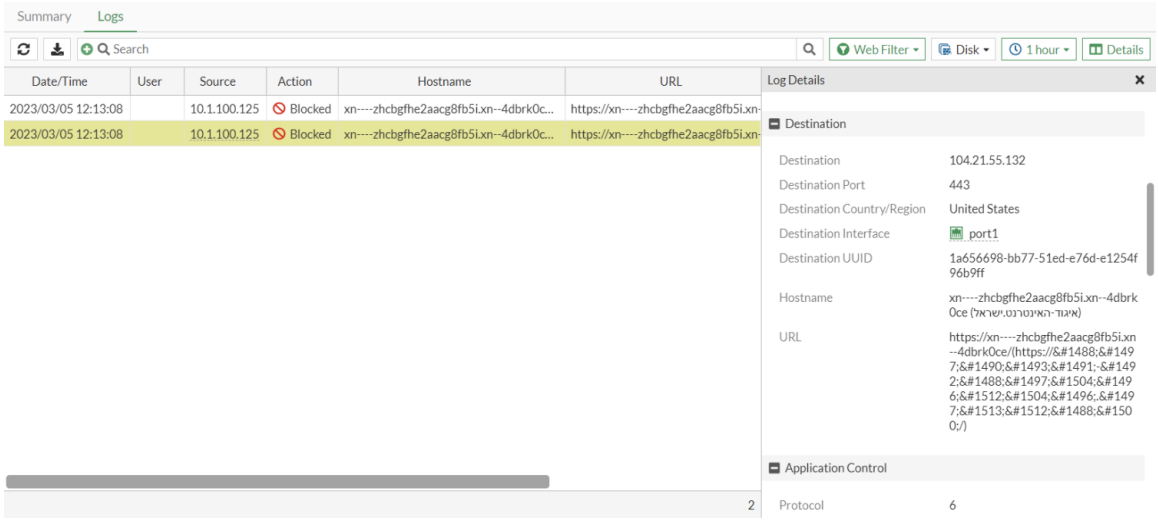
   - In proxy mode, the URL is displayed in Punycode.



To verify the logs:

1. Go to *Log & Report* > *Security Events* and select the *Web Filter* card.
2. Select a log and click *Details*. The format of the *Hostname* and *URL* fields depends on the inspection mode.

- In flow mode, the *Hostname* is displayed in Punycode with Hebrew in parentheses. The *URL* is displayed Punycode.

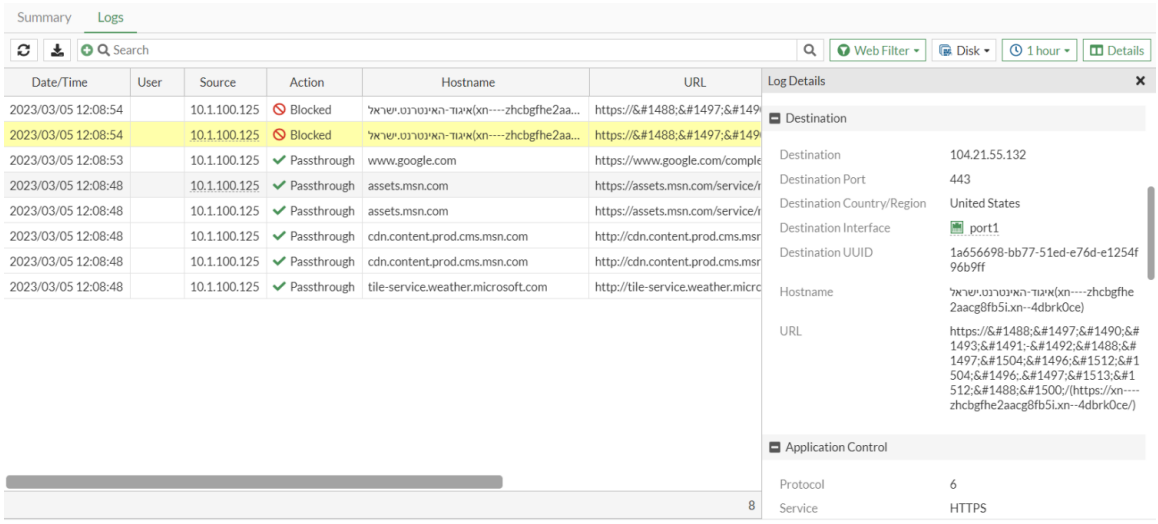


When the log file is downloaded, the hostname in the raw file cannot be displayed. Paste the log into a text editor (such as Word or Notepad) to view the URL in Hebrew.

```
# execute log display
2 logs found.
2 logs returned.

1: date=2023-03-20 time=09:43:57 eventtime=1679330638045179264 tz="-0700"
logid="0315012544" type="utm" subtype="webfilter" eventtype="urlfilter" level="warning"
vd="vdom1" urlfilteridx=10 urlfilterlist="Hebrew-url" policyid=1 poluuid="d0c84854-c736-
51ed-d761-71527ca0b446" policytype="policy" sessionid=6987 srcip=10.1.100.125
srcport=54542 srccountry="Reserved" srcintf="port2" srcintfrole="undefined"
srcuuid="4074dca4-c736-51ed-0e0e-7a6b5ec6b7b9" dstip=172.67.148.48 dstport=443
dstcountry="United States" dstintf="port1" dstintfrole="undefined" dstuuid="4074dca4-c736-
51ed-0e0e-7a6b5ec6b7b9" proto=6 httpmethod="GET" service="HTTPS" hostname="-אינוד
האינטרנט.ישראל" agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/110.0.0.0 Safari/537.36" profile="webfilter_flowbase"
action="blocked" reqtype="referral" url="https://אינוד-האינטרנט.ישראל/favicon.ico"
referralurl="https://xn----zhcbgfhe2aacg8fb5i.xn--4dbrk0ce/" sentbyte=496 rcvdbyte=0
direction="outgoing" urlsource="Local URLfilter Block" msg="URL was blocked because it is
in the URL filter list" crscore=30 craction=8 crlevel="high"

2: date=2023-03-20 time=09:43:57 eventtime=1679330637755477060 tz="-0700"
logid="0315012544" type="utm" subtype="webfilter" eventtype="urlfilter" level="warning"
vd="vdom1" urlfilteridx=10 urlfilterlist="Hebrew-url" policyid=1 poluuid="d0c84854-c736-
51ed-d761-71527ca0b446" policytype="policy" sessionid=6982 srcip=10.1.100.125
srcport=54540 srccountry="Reserved" srcintf="port2" srcintfrole="undefined"
srcuuid="4074dca4-c736-51ed-0e0e-7a6b5ec6b7b9" dstip=172.67.148.48 dstport=443
dstcountry="United States" dstintf="port1" dstintfrole="undefined" dstuuid="4074dca4-c736-
51ed-0e0e-7a6b5ec6b7b9" proto=6 httpmethod="GET" service="HTTPS" hostname="-אינוד
האינטרנט.ישראל" agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/110.0.0.0 Safari/537.36" profile="webfilter_flowbase"
action="blocked" reqtype="direct" url="https://אינוד-האינטרנט.ישראל/" sentbyte=546
```

```
rcvdbyte=0 direction="outgoing" urlsource="Local URLfilter Block" msg="URL was blocked
because it is in the URL filter list" crscore=30 craction=8 crlevel="high"
```

- In proxy mode, the *Hostname* is displayed in Hebrew with Punycode in parentheses. The *URL* is displayed Punycode.



When the log file is downloaded, the hostname in the raw file is displayed in Punycode.

```
# execute log display
2 logs found.
2 logs returned.

1: date=2023-03-20 time=09:38:44 eventtime=1679330324572766407 tz="-0700"
logid="0315012544" type="utm" subtype="webfilter" eventtype="urlfilter" level="warning"
vd="vdom1" urlfilteridx=1 urlfilterlist="Hebrew-url" policyid=1 poluuid="d0c84854-c736-
51ed-d761-71527ca0b446" policytype="policy" sessionid=6782 srcip=10.1.100.125
srcport=50527 srccountry="Reserved" srcintf="port2" srcintfrole="undefined"
srcuuid="4074dca4-c736-51ed-0e0e-7a6b5ec6b7b9" dstip=104.21.55.132 dstport=443
dstcountry="United States" dstintf="port1" dstintfrole="undefined" dstuuid="4074dca4-c736-
51ed-0e0e-7a6b5ec6b7b9" proto=6 httpmethod="GET" service="HTTPS" hostname="xn----
zhcbgfhe2aacg8fb5i.xn--4dbrk0ce" agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.0.0 Safari/537.36" profile="webfilter"
action="blocked" reqtype="referral" url="https://xn----zhcbgfhe2aacg8fb5i.xn--
4dbrk0ce/favicon.ico" referralurl="https://xn----zhcbgfhe2aacg8fb5i.xn--4dbrk0ce/"
sentbyte=1402 rcvdbyte=5473 direction="outgoing" urlsource="Local URLfilter Block"
msg="URL was blocked because it is in the URL filter list" crscore=30 craction=8
crlevel="high"

2: date=2023-03-20 time=09:38:44 eventtime=1679330324438504542 tz="-0700"
logid="0315012544" type="utm" subtype="webfilter" eventtype="urlfilter" level="warning"
vd="vdom1" urlfilteridx=1 urlfilterlist="Hebrew-url" policyid=1 poluuid="d0c84854-c736-
51ed-d761-71527ca0b446" policytype="policy" sessionid=6782 srcip=10.1.100.125
srcport=50527 srccountry="Reserved" srcintf="port2" srcintfrole="undefined"
srcuuid="4074dca4-c736-51ed-0e0e-7a6b5ec6b7b9" dstip=104.21.55.132 dstport=443
dstcountry="United States" dstintf="port1" dstintfrole="undefined" dstuuid="4074dca4-c736-
51ed-0e0e-7a6b5ec6b7b9" proto=6 httpmethod="GET" service="HTTPS" hostname="xn----
zhcbgfhe2aacg8fb5i.xn--4dbrk0ce" agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64)
```

```
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/110.0.0.0 Safari/537.36" profile="webfilter"
action="blocked" reqtype="direct" url="https://xn----zhcbgfhe2aacg8fb5i.xn--4dbrk0ce/"
sentbyte=1171 rcvdbyte=4930 direction="outgoing" urlsource="Local URLfilter Block"
msg="URL was blocked because it is in the URL filter list" crscore=30 craction=8
crlevel="high"
```

If a FortiGuard category-based filter is configured in a web filter profile, the same behavior for replacement messages and logs applies based on the inspection mode.

Configuring web filter profiles to block cryptocurrency

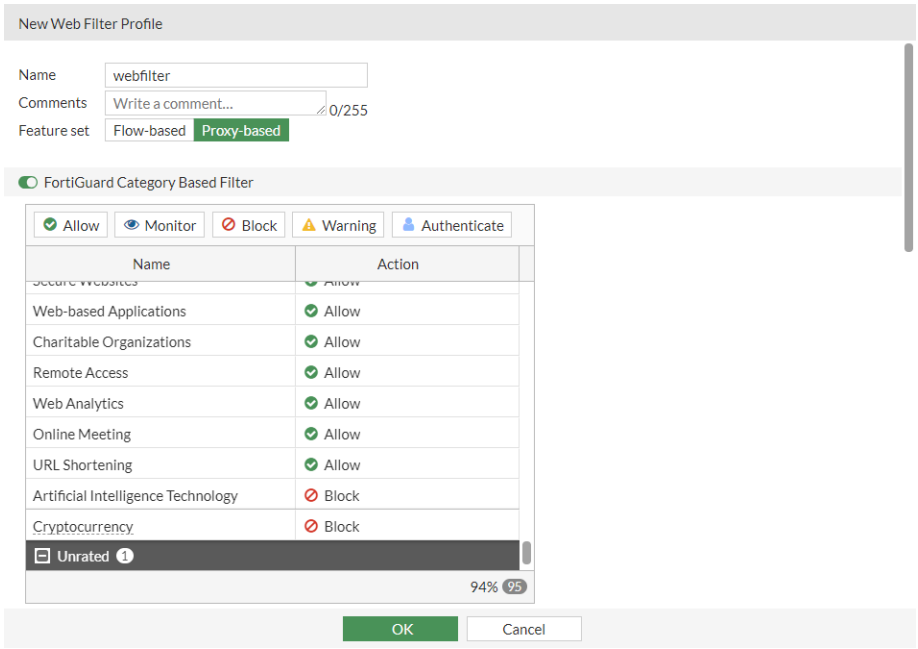
The following FortiGuard web filter category is available:

- Cryptocurrency (category 101): sites that specialize in digital or virtual currencies that are secured by cryptography and operate on decentralized networks.

See also Protecting GenAI access using web filter on page 2225.

To configure a web filter profile to block the cryptocurrency categories in the GUI:

1. Go to *Security Profiles* > *Web Filter* and click *Create New*.
2. Enter a name for the web filter profile.
3. In the category table, locate the *General Interest - Business* section. Select the *Cryptocurrency* category, and set the *Action* to *Block*.



4. Configure the remaining settings as needed.
5. Click *OK*.

To configure a web filter profile to block the cryptocurrency category in the CLI:

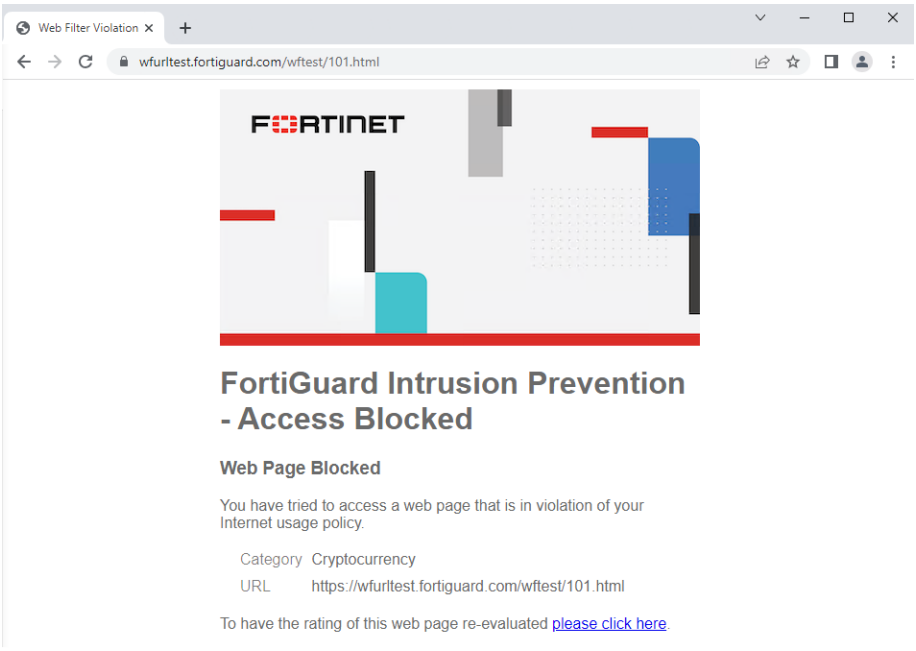
```
config webfilter profile
    edit "webfilter"
        set feature-set proxy
        config ftgd-wf
            unset options
            config filters
                edit 101
                    set category 101
                    set action block
                next
                edit 52
                    set category 52
                next
            end
        end
        set log-all-url enable
    next
end
```

To verify that the categories are blocked:

1. Apply the web filter profile in a firewall policy.
2. On a device that is connected through the FortiGate and uses the policy, visit the following test URL for the category:
   a. `https://wfurltest.fortiguard.com/wftest/101.html`

      The browser displays a replacement message that the URL is blocked based on the FortiGuard category.

- Cryptocurrency:



To verify the web filter logs:

1. In the GUI, go to *Log & Report* > *Security Events* and click *Web Filter*.
2. In the CLI, enter the following:

```
# execute log filter category utm-webfilter
# execute log display
1: date=2023-07-12 time=10:39:18 eventtime=1689183557968026063 tz="-0700" logid="0316013056"
type="utm" subtype="webfilter" eventtype="ftgd_blk" level="warning" vd="vdom1" policyid=1
poluuid="996b0a68-2055-51ee-b841-2b3f373c9b37" policytype="policy" sessionid=3258
srcip=10.1.100.31 srcport=35116 srccountry="Reserved" srcintf="port2" srcintfrole="undefined"
srcuuid="124f368a-2055-51ee-c7d6-857ab36dd6cb" dstip=154.52.5.202 dstport=443
dstcountry="United States" dstintf="port1" dstintfrole="undefined" dstuuid="124f368a-2055-
51ee-c7d6-857ab36dd6cb" proto=6 httpmethod="GET" service="HTTPS"
hostname="wfurltest.fortiguard.com" agent="curl/7.68.0" profile="webfilter" action="blocked"
reqtype="direct" url="https://wfurltest.fortiguard.com/wftest/101.html" sentbyte=849
rcvdbyte=3633 direction="outgoing" msg="URL belongs to a denied category in policy"
ratemethod="domain" cat=101 catdesc="Cryptocurrency"
```

Configuring web filter to exempt URLs from other security profiles

When configuring other security profiles, specific URLs sometimes need to be exempt from the defined rules.

For example, a rule might exist to block all executable files, but an executable file needs to be downloadable from a specific website. In this case, a web filter static URL filter can be used to exempt the specific URL from further processing by the other security profiles.

Static URL filters can be applied to other security profiles as follows:

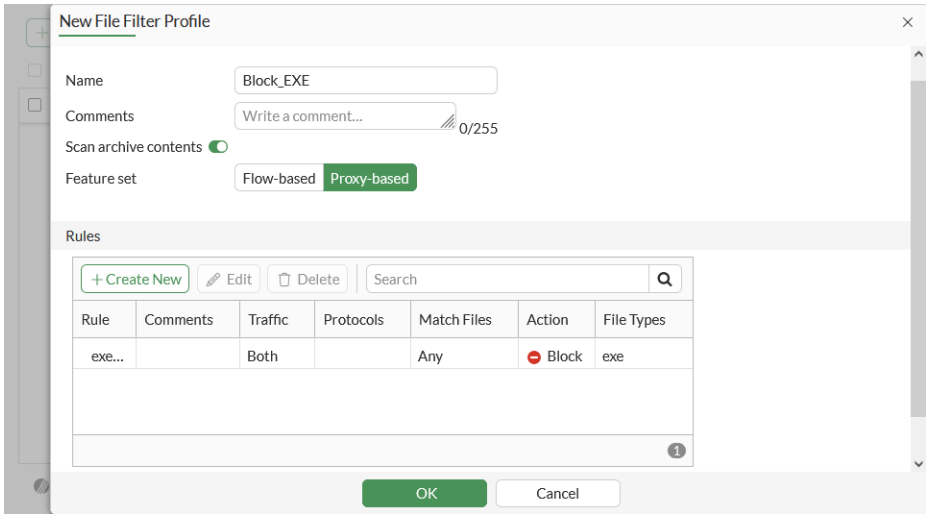
Security profile	Proxy support	Flow support
File filter	Yes	No
Antivirus	Yes	Yes
Data loss prevention	Yes	Yes
Video filter	No	(Proxy only security profile)

Example

In this example, a file filter has been configured to block downloads of .exe files and is applied to a firewall policy that has deep inspection enabled.

The file filter has the following settings:

Field	Value
Name	Block_EXE
Feature set	Proxy-based
Rule	exe_block
Traffic	Both
Match Files	Any
Action	Block
File Types	exe

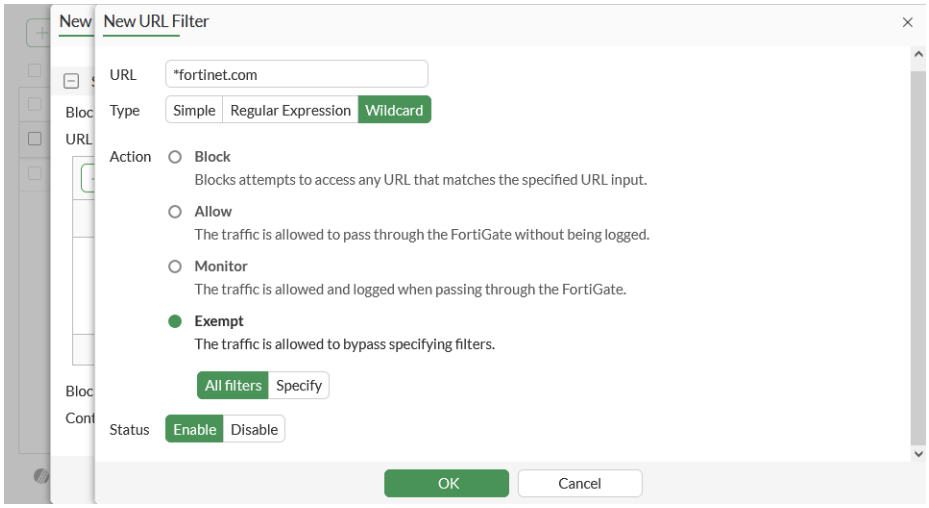


See File filter on page 2518 for more information about configuring file filters.

To configure the web filter exemption in the GUI:

1. Go to *Security Profiles* > *Web Filter* and click *Create New*.
2. Enter a profile *Name*.

3. In the *Static URL Filter* section, enable *URL Filter* and click *Create New* in the table.
4. Enter the URL you need to make an exception for. In this example the *Wildcard* type is used.
5. Set *Action* to *Exempt* and *Status* to *Enable*.



6. Click OK.
7. Click OK to save the web filter profile.
8. Enable the web filter profile in the firewall policy that includes the file filter that you need to add the exemption to. See Firewall policy on page 1842 for more information.

To configure the web filter exemption in the CLI:

1. Configure the URL filter:

```
config webfilter urlfilter
    edit 99
        set name "exempt_URL"
        config entries
            edit 1
                set url "*fortinet.com"
                set type wildcard
            next
        end
    next
end
```

2. Configure the web filter profile to use the URL filter:

```
config webfilter profile
    edit "exempt_exe"
        config web
            set urlfilter-table 99
        end
    next
end
```

3. Apply the web filter in the firewall policy:

```
config firewall policy
    edit 1
        set webfilter-profile "exempt_exe"
    next
end
```

Video filter

The video filter profile can be used to filter YouTube videos based on several criteria, including: FortiGuard categories, video titles, video descriptions, and channel IDs. These criteria provide a more granular override of a single channel, user, or video. The video filter profile is currently supported in proxy-based policies and requires SSL deep inspection. The FortiGuard Video filtering service is based on a valid FortiGuard web filter license.

This feature is not supported on FortiGate models with 2 GB RAM or less. See Proxy-related features not supported on FortiGate 2 GB RAM models on page 126 for more information.

Configuring a video filter profile

In the GUI (*Security Profiles* > *Video Filter* > *Video Filter Profile*), four types of filters can be created on the *New Video Filter Profile* page: *Category*, *Title*, *Description*, and *Channel*.

When adding a *Category* type filter, the various FortiGuard categories, including *Any*, can be set to allow, monitor, or block videos in those categories. See Filtering based on FortiGuard categories on page 2373 for a detailed example and explanation of how the WAD daemon inspects videos.

The *Title* and *Description* type filters can be used to filter video based on keywords. See Configuring a video filter keyword list on page 2382 for more information. When a video's title or description matches a defined keyword, the video filter will take the corresponding action of allow, monitor, or block. See Filtering based on title on page 2380 and Filtering based on description on page 2381 for detailed examples.

The *Channel* type filter can be used to filter all or specific YouTube channels. When a video matches a YouTube channel, the video filter will take the corresponding action of allow, monitor, or block. See Filtering based on YouTube channel on page 2378 for a detailed example.

Users can prioritize configured filters within the video filter profiles based on their sequence order. To change the sequence order, drag the selected filter to the desired position. An implicit rule within the video filter profile is set to *Allow*. If a video does not match any of the other filters, it will be subject to this implicit rule and allowed to pass through.

To configure a video filter profile:

```
config videofilter profile
    edit <name>
        config filters
            edit <id>
                set type {category* | channel | title | description}
                set log {enable* | disable}
```

```
                set action {allow | block | monitor*}
            next
        end
    next
end
```

The default values are marked with an asterisk (*).

YouTube API key

The YouTube API key is required when filtering by a:

- YouTube video title
- YouTube video description
- YouTube channel in conjunction with filtering based on FortiGuard categories

To configure the YouTube API key in the GUI:

1. Go to *Security Profiles* > *Video Filter* and select the *Video Filter Settings* tab.
2. Click the + to add an API key.
3. Click *OK*.

To configure the YouTube API key in the CLI:

```
config videofilter youtube-key
    edit <id>
        set key <string>
    next
end
```

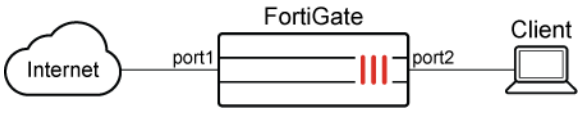
Filtering based on FortiGuard categories

Video filtering is only proxy-based and uses the WAD daemon to inspect the video in four phases:

1. When the WAD receives a video query from a client, it extracts the video ID (`vid`) and tries to check the category and channel from the local cache.
2. If there is no match from the local cache, it connects to the FortiGuard video rating server to query the video category.
3. If the FortiGuard rating fails, it uses the `videofilter.youtube-key` to communicate with the Google API server to get its category and channel ID. This is the API query setting and it requires the user's own YouTube API key string. This configuration is optional.
4. If all steps fail to match the video, the WAD calls on the IPS engine to match the video ID and channel ID from the application signature database.

The FortiGuard anycast service must be enabled to use this feature.

In this example, a new video filter profile is created to block the Knowledge category.



To configure a video filter based on FortiGuard categories in the GUI:

1. Create the video filter profile:
   a. Go to *Security Profiles* > *Video Filter*, select the *Video Filter Profile* tab, and click *Create new*.
   b. Enter a name (*category_filter*).
   c. Create the filter:
      i. In the *Filters* table, click *Create new*.
      ii. Set the *Type* to the *Category*.
      iii. Set the *Action* to *Block*.
      iv. Set the *Category* to *Knowledge*.
      v. Click *OK* to save the filter.
   d. Click *OK* to save the video filter profile.
2. Create the firewall policy:
   a. Enter the following:

Incoming Interface	port2
Outgoing Interface	port1
Source	All
Destination	All
Service	All
Inspection Mode	Proxy-based
NAT	Enable
Video Filter	Enable and select *category_filter*
Application Control	Enable and select *default*
SSL Inspection	deep-inspection
Log Allowed Traffic	All Sessions

   b. Configure the other settings as needed and click *OK*.

To configure a video filter based on FortiGuard categories in the CLI:

1. Create the video filter profile:

```
config videofilter profile
    edit "category_filter"
        config filters
            edit 1
```

```
                set type category
                set category "4"
                set action block
                set log enable
            next
        end
    next
end
```

2. Create the firewall policy:

```
config firewall policy
    edit 10
        set name "client_yt_v4"
        set srcintf "port2"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "deep-inspection"
        set application-list "default"
        set videofilter-profile "category_filter"
        set logtraffic all
        set nat enable
    next
end
```

Verifying that the video is blocked

When a user browses to YouTube and selects a video based in the Knowledge category, a replacement message will appear (see Example 3: blocking the video based on FortiGuard category on YouTube for an example replacement message). On the FortiGate, verify the forward traffic and web filter logs.

Sample forward traffic log:

```
2: date=2023-12-05 time=09:05:32 eventtime=1701796727673178582 tz="-0800" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="root" srcip=10.1.100.11 srcport=50568
srcintf="port2" srcintfrole="undefined" dstip=142.251.179.93 dstport=443 dstintf="port1"
dstintfrole="undefined" srccountry="United States" dstcountry="United States" sessionid=480384
proto=6 action="client-rst" policyid=1 policytype="policy" poluuid="f4fe48a4-938c-51ee-8856-
3e84e3b24af4" policyname="client_yt_v4" service="HTTPS" trandisp="snat" transip=172.16.200.1
transport=50568 appcat="unknown" applist="default" duration=821 sentbyte=303404 rcvdbyte=3601568
sentpkt=1824 rcvdpkt=2688 wanin=3493278 wanout=201892 lanin=126344 lanout=3493868
utmaction="block" countweb=2 countapp=3 sentdelta=0 rcvddelta=0 utmref=65514-4674
```

Sample web filter log:

```
1: date=2023-12-05 time=09:05:37 eventtime=1701795937361806440 tz="-0800" logid="0347013664"
type="utm" subtype="webfilter" eventtype="videofilter-category" level="warning" vd="root"
msg="Video category is blocked." policyid=1 poluuid="f4fe48a4-938c-51ee-8856-3e84e3b24af4"
sessionid=480384 srcip=10.1.100.11 dstip=142.251.179.93 srcport=50568 dstport=443 srcintf="port2"
srcintfrole="undefined" dstintf="port1" dstintfrole="undefined" proto=6 httpmethod="POST"
service="HTTPS" action="blocked" videoinfosource="FortiGuard" profile="category_filter"
videoid="hG-rVFM62J4" videocategoryid=4 videocategoryname="Knowledge" hostname="www.youtube.com"
agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KH"
referralurl="https://www.youtube.com/results?search_query=udemy"
url="https://www.youtube.com/youtubei/v1/player?key=AIzaSyAO_FJ2SlqU8Q4STEHLGCilw_Y9_
11qcW8&prettyPrint=false"
```

Troubleshooting and debugging

To verify if the FortiGuard video filtering license is valid:

```
# get system fortiguard

fortiguard-anycast  : enable
fortiguard-anycast-source: fortinet
protocol            : https
port                : 443
...
webfilter-license   : Contract
webfilter-expiration: Mon Oct 28 2024
...
```

To verify the WAD worker is running:

```
# diagnose test app wad 1000
Process [0]: WAD manager type=manager(0) pid=232 diagnosis=yes.
Process [1]: type=worker(2) index=0 pid=294 state=running
              diagnosis=no debug=enable valgrind=supported/disabled
...
Process [6]: type=YouTube-filter-cache-service(9) index=0 pid=290 state=running
              diagnosis=no debug=enable valgrind=unsupported/disabled
...
```

To display and debug video filter cache:

```
# diagnose test app wad ?
....
        321:  Display Video Filter Cache stats.
        322:  Reset Video Filter Cache stats.
        323:  Flush Video Filter Cache entries.
        324:  Display Video Filter module stats.
        325:  Request category list from Youtube API.
```

```
        326:  Display FTGD agent module stats.
        327:  Reset FTGD agent module stats.
        328:  Toggle Video Filter Cache Check.
        329:  Toggle Video Filter FTGD Query.
        330:  Toggle Video Filter API Check.
```

To enable real-time WAD debugs:

```
# diagnose wad debug enable level verbose
# diagnose wad debug enable category video
# diagnose debug enable
```

Sample output

```
[p:274][s:8754][r:186] wad_http_req_exec_video_filter_check(167): hreq=0x7f1184f288e0, check video
filter check videofilter
[p:274][s:8754][r:186] wad_vf_req_submit(1869): node=0x7f1186694640, ctx=0x7f118502d1f8, youtube_
channel_filter_id=0
[p:274][s:8754][r:186] wad_vf_match_pattern_cb(1551): ctx=0x7f118502d1f8 matched type video
[p:274][s:8754][r:186] wad_vf_extract_video_id(297): str='v=EAyo3_zJj5c', start='v=', end='&'
[p:274][s:8754][r:186] wad_vf_extract_video_id(297): str='v=EAyo3_zJj5c', start='v=', end=''
[p:274][s:8754][r:186] wad_vf_extract_video_id(322): video-id: start=2, end=13
[p:274][s:8754][r:186] wad_vf_sync_task_trigger_async_task(1602): extracted vid=EAyo3_zJj5c
ctx=0x7f118502d1f8
[p:274][s:8754][r:186] wad_vf_sync_task_trigger_async_task(1622): video filter ctx=0x7f118502d1f8
creates new task=0x7f118657e7a0
[p:274][s:8754][r:186] wad_vfc_client_lookup(159): oid=15194313278609724406
[p:274][s:8754][r:186] wad_vfc_core_lookup(277): youtube-filter-cache core(0x7f11864d2078) found
the item!
[p:274][s:8754][r:186] wad_vfc_client_lookup(174): local lookup: ret=0 result=hit, hit_cnt=51
local hit item, item's value:
  oid=15194313278609724406
  vid="EAyo3_zJj5c"
  category="4"
  title="Youtube Data API V3 Video Search Example"
  channel="UCR6d0EiC3G4WA8-Rqji6a8g"
  desc(first 100 characters)="Youtube Data API V3 Video Search Example

Welcome Folks My name is Kiki and Welcome to Coding Shik......"
[p:274][s:8754][r:186] wad_vf_task_proc_cache_resp(1048): vf filter cache hit, item=0x7f116dacc060
[p:274][s:8754][r:186] wad_vf_async_task_run(1491): end of async task ret=0
[p:274][s:8754][r:186] wad_vf_sync_task_proc_async_result(1686): task=0x7f118657e7a0
item=0x7f116dacc060
[p:274][s:8754][r:186] wad_vf_sync_task_proc_async_result(1721): ctx(0x7f118502d1f8) channel
UCR6d0EiC3G4WA8-Rqji6a8g not match
[p:274][s:8754][r:186] wad_vf_sync_task_proc_async_result(1733): ctx(0x7f118502d1f8) category
result is block
[p:274][s:8754][r:186] wad_vfc_client_add(230): oid=15194313278609724406
```

Filtering based on YouTube channel

Video filtering can be configured to filter any or specific YouTube channels. When a video matches a YouTube channel, the video will take the corresponding action of allow, monitor, or block. Video filtering is only supported in proxy-based inspection mode, and deep inspection must be enabled in the firewall policy.

The YouTube API key must be configured when using this feature in conjunction with filtering based on FortiGuard categories. See YouTube API key for more information.

Videos are filtered based on their sequence order, which could allow for specific categorization. A category can be blocked, but certain channels within that category can be allowed. If no match is found, then the video will be subjected to an implicit rule and allowed to pass through See Basic configuration for an example.

Identifying the YouTube channel ID

The following table lists how to identify the YouTube channel ID based on different YouTube video URLs formats:

Video URL	Channel ID
www.youtube.com/channel/<channel-id>	<channel-id> indicates the ID for the channel.
www.youtube.com/user/<user-id>	Open the page source and locate: `<meta itemprop="channelId" content="<channel-id>">` <channel-id> indicates the channel ID for the user page.
www.youtube.com/watch?v=<string>	Open the page source and locate: `<meta itemprop="channelId" content="<channel-id>">` <channel-id> indicates the channel ID for the video.

Basic configuration

In this example, the Knowledge category in the video filter is configured to be blocked. The YouTube channel filter list is configured with the action set to monitor, which effectively creates an allowlist. The Fortinet YouTube channel ID UCJHo4AuVomwMRzgkA5DQEOA is allowed. This example uses the category_filter video filter profile and client_yt_v4 firewall policy configured in Filtering based on FortiGuard categories on page 2373.

To configure a video filter based on a YouTube channel in the GUI:

1. Go to *Security Profiles* > *Video Filter*, select the *Video Filter Profile* tab, and edit *channel_filter*.
2. In the *Filters* table, click *Create new*.
3. Set the *Type* to the *Channel*.

4. Set the *Action* to *Monitor*.
5. Set the *Channel* to *Specify* and enter the *Channel ID*, *UCJHo4AuVomwMRzgkA5DQEOA*.
6. Click *OK* to save the filter.
7. Click *OK* to save the video filter profile.

To configure a video filter based on a YouTube channel in the CLI:

```
config videofilter profile
    edit "category_filter"
        config filters
            edit 2
                set type channel
                set channel "UCJHo4AuVomwMRzgkA5DQEOA"
                set action monitor
                set log enable
            next
        end
    next
end
```

Verifying the configuration

Perform a search from a client for a video on YouTube that falls under the Knowledge category, such as any video from the Udemy channel. The result is a blocked video. However, searching for a video from the Fortinet YouTube channel results in an accessible video.

Sample logs:

```
1: date=2023-12-13 time=09:33:20 eventtime=1702488800873289004 tz="-0800" logid="0347013664"
type="utm" subtype="webfilter" eventtype="videofilter-category" level="warning" vd="root"
msg="Video category is blocked." policyid=1 poluuid="f4fe48a4-938c-51ee-8856-3e84e3b24af4"
sessionid=33099 srcip=13.13.13.13 dstip=108.177.98.136 srcport=58844 dstport=443 srcintf="port2"
srcintfrole="undefined" dstintf="port1" dstintfrole="undefined" proto=6 httpmethod="POST"
service="HTTPS" action="blocked" videoinfosource="FortiGuard" profile="category_filter"
videoid="hG-rVFM62J4" videocategoryid=4 videocategoryname="Knowledge" hostname="www.youtube.com"
agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KH"
referralurl="https://www.youtube.com/results?search_query=udemy"
url="https://www.youtube.com/youtubei/v1/player?key=AIzaSyAO_FJ2SlqU8Q4STEHLGCilw_Y9_
11qcW8&prettyPrint=false"
```

```
2: date=2023-12-13 time=09:30:29 eventtime=1702488629127408526 tz="-0800" logid="0348013681"
type="utm" subtype="webfilter" eventtype="videofilter-channel" level="notice" vd="root" msg="Video
channel is monitored." policyid=1 poluuid="f4fe48a4-938c-51ee-8856-3e84e3b24af4" sessionid=32740
srcip=13.13.13.13 dstip=142.251.211.238 srcport=58685 dstport=443 srcintf="port2"
srcintfrole="undefined" dstintf="port1" dstintfrole="undefined" proto=6 httpmethod="POST"
service="HTTPS" action="passthrough" videoinfosource="API" profile="category_filter"
videoid="PZzW7rYMUJw" videochannelid="UCJHo4AuVomwMRzgkA5DQEOA" hostname="www.youtube.com"
agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KH"
referralurl="https://www.youtube.com/watch?v=PZzW7rYMUJw"
```

```
url="https://www.youtube.com/youtubei/v1/player?key=AIzaSyAO_FJ2SlqU8Q4STEHLGCilw_Y9_
11qcW8&prettyPrint=false"
```

Filtering based on title

Video filtering can be configured to filter using keyword-based filters for video titles. When a video's title matches the configured keyword, the video filter will take the corresponding action of allow, monitor, or block. Video filtering is only supported in proxy-based inspection mode, and deep inspection must be enabled in the firewall policy.

The YouTube API key must be configured to use this feature. Otherwise, the title filter will not retrieve the video information and bypass the traffic. See YouTube API key for more information.

Basic configuration

In this example, videos are blocked that contain the keyword, game. For information about configuring video filter keyword lists, see Example configuration.

To configure the video filter profile in the GUI:

1. Configure the video filter profile:
   a. Go to *Security Profiles* > *Video Filter*, select the *Video Filter Profile* tab, and click *Create new*.
   b. Enter a name (*title-filter-profile*).
   c. In the *Filters* table, click *Create new*.
   d. Configure the filter with the following settings:
      i. Set the *Type* to *Title*.
      ii. Set the *Action* to *Block*.
      iii. Set the *Keyword* to *test-keyword-match-or*.
      iv. Click *OK*.
   e. Click *OK* to save the video filter profile.
2. Apply the video filter in a firewall policy.

To configure the video filter profile in the CLI:

1. Configure the video filter profile:

```
config videofilter profile
    edit "title-filter-profile"
        config filters
            edit 1
                set type title
                set keyword 1
                set action block
                set log enable
            next
        end
```

```
    next
end
```

2. Apply the video filter in a firewall policy.

Verifying the configuration

From a client, search for a video in YouTube named "How To Use Python Steam API || Steam game API python". The video is blocked.

Sample log:

```
6: date=2023-11-24 time=09:51:30 eventtime=1700848289598975941 tz="-0800" logid="0350013712"
type="utm" subtype="webfilter" eventtype="unknown" level="warning" vd="vdom1" msg="Video title is
blocked." policyid=1 poluuid="19841eb8-841c-51ee-7047-6a6860eb3522" sessionid=384813810
srcip=10.1.100.141 dstip=142.251.33.110 srcport=21473 dstport=443 srcintf="port2"
srcintfrole="undefined" dstintf="port1" dstintfrole="undefined" proto=6 httpmethod="GET"
service="HTTPS" action="blocked" videoinfosource="API" profile="title-filter-profile"
videoid="LaRHkSVvDjI" videotitle="How To Use Python Steam API  || Steam game API python"
hostname="www.youtube.com" agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KH" url="https://www.youtube.com/watch?v=LaRHkSVvDjI"
```

Filtering based on description

Video filtering can be configured to filter using keyword-based filters for video descriptions. When a video's description matches the configured keyword, the video filter will take the corresponding action of allow, monitor, or block. The description filter supports the first 100 characters of the video description. Video filtering is only supported in proxy-based inspection mode, and deep inspection must be enabled in the firewall policy.

The YouTube API key must be configured to use this feature. Otherwise, the description filter will not retrieve the video information and bypass the traffic. See YouTube API key for more information.

Basic configuration

In this example, videos are blocked where the description contains the keyword, API. For information about configuring video filter keyword lists, see Example configuration.

To configure the video filter profile in the GUI:

1. Configure the video filter profile:
   a. Go to *Security Profiles* > *Video Filter*, select the *Video Filter Profile* tab, and click *Create new*.
   b. Enter a name (*test-description-filter*).
   c. In the *Filters* table, click *Create new*.
   d. Configure the filter with the following settings:
      i. Set the *Type* to *Description*.
      ii. Set the *Action* to *Block*.

iii. Set the *Keyword* to *test-keyword-match-all*.
iv. Click *OK*.

e. Click *OK* to save the video filter profile.

2. Apply the video filter in a firewall policy.

To configure the video filter profile in the CLI:

1. Configure the video filter profile:

```
config videofilter profile
    edit "test-description-filter"
        config filters
            edit 1
                set type description
                set keyword 2
                set action block
                set log enable
            next
        end
    next
end
```

2. Apply the video filter in a firewall policy.

Verifying the configuration

From a client, search for a video in YouTube named "Postman Tutorial #7 - HTTP Methods GET and POST in Postman". The description contains the text, "POSTMAN TUTORIAL - Complete API Testing and API Test Automation Course using Postman Tool...", so the video is blocked.

Sample log:

```
4: date=2023-11-24 time=16:08:51 eventtime=1700870931146681788 tz="-0800" logid="0351013728"
type="utm" subtype="webfilter" eventtype="unknown" level="warning" vd="vdom1" msg="Video
description is blocked." policyid=1 poluuid="090ca600-83e4-51ee-158a-a920fcf8f892"
sessionid=100211 srcip=10.1.100.141 dstip=142.250.69.206 srcport=24948 dstport=443 srcintf="port2"
srcintfrole="undefined" dstintf="port1" dstintfrole="undefined" proto=6 httpmethod="GET"
service="HTTPS" action="blocked" videoinfosource="API" profile="test-description-filter"
videoid="pUGmhtqVJRk" videodesc="Get all my courses for USD 5.99/Month - https://bit.ly/all-c..."
hostname="www.youtube.com" agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KH" url="https://www.youtube.com/watch?v=pUGmhtqVJRk"
```

Configuring a video filter keyword list

Users have the ability to filter videos based on the video titles and descriptions by defining a video filter keyword. Multiple keywords can be defined using a wildcard or regular expression, and can be used with AND/OR logic options.

Unicode emoji character code is currently not supported.

Configure the video filter keyword list in the GUI:

1. Go to *Security Profiles* > *Video Filter*, select the *Video Filter Keyword* tab, and click *Create new*.
2. Configure the following:

Name	Enter a name.
Match operator	Select the match operator. • *Any*: match any keyword (OR logic) • *All*: match all keywords (AND logic)

3. Create the keywords:
   a. In the *Keywords* table, click *Create new*.
   b. Configure the following:

Pattern	Enter a regular expression or wildcard pattern string.
Pattern type	Select the pattern type. • *Wildcard*: suitable for basic search • *Regular Expression*: suitable for advanced search
Status	Set the status. • *Enable*: consider this keyword • *Disable*: ignore this keyword

   c. Click *OK*.
4. Click *OK* to save the keyword list.

Configure the video filter keyword list in the CLI:

```
config videofilter keyword
    edit <id>
        set name <string>
        set match {or | and}
        config word
           edit <name>
                set pattern-type {wildcard | regex}
                set status {enable | disable}
            next
        end
    next
end
```

Example configuration

In this example, two keywords, API and game, are created with the wildcard and regular expression types, respectively. The match operator is set to *Any*.

Configure the video filter keyword list in the GUI:

1. Go to *Security Profiles* > *Video Filter*, select the *Video Filter Keyword* tab, and click *Create new*.
2. Enter a name (*test-keyword-match-or*) and set *Match operator* to *Any*.
3. In the *Keywords* table, click *Create new*.
4. Configure the API keyword with the following settings:
   a. In the *Pattern* field, enter *API*.
   b. Set the *Pattern type* to *Wildcard*.
   c. Click *OK*.
5. Click *Create new*.
6. Configure the game keyword with the following settings:
   a. In the *Pattern* field, enter *Game*.
   b. Set the *Pattern type* to *Regular Expression*.
   c. Click *OK*.
7. Click *OK* to save the keyword list.

Configure the video filter keyword list in the CLI:

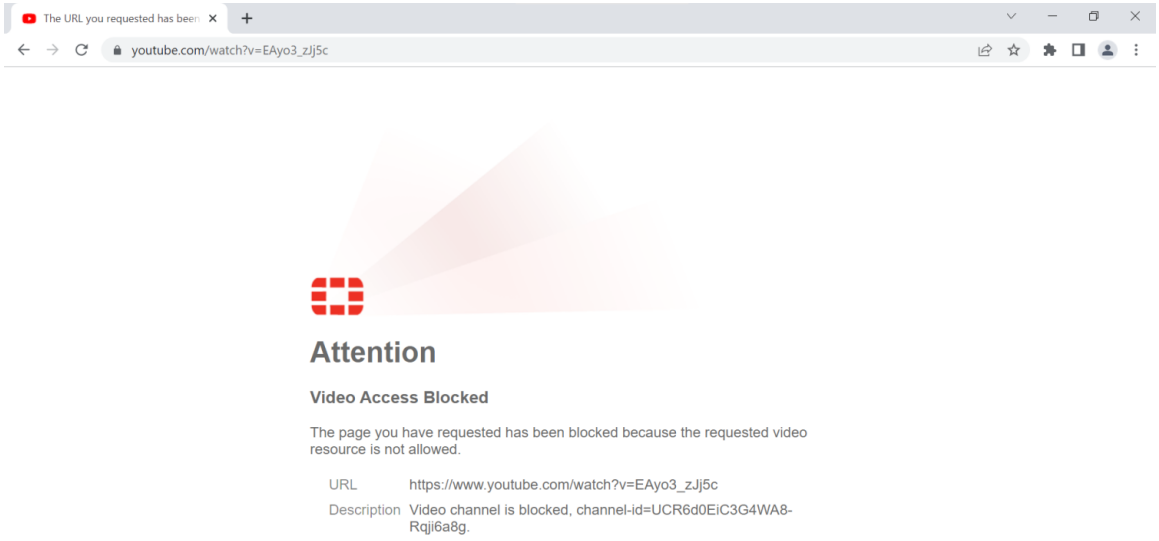
```
config videofilter keyword
    edit 1
        set name "test-keyword-match-or"
        set match or
        config word
           edit "API"
                set pattern-type wildcard
                set status enable
            next
            edit "Game"
                set pattern-type regex
                set status enable
            next
        end
    next
end
```

Replacement messages displayed in blocked videos

When a user visits a video directly by a URL, a full page replacement message is displayed. When a user loads a video from the YouTube website (homepage or recommended videos), the page loads and the replacement message is displayed in the video frame. For more information about configuring video filters, see Filtering based on FortiGuard categories on page 2373 and Filtering based on YouTube channel on page 2378.

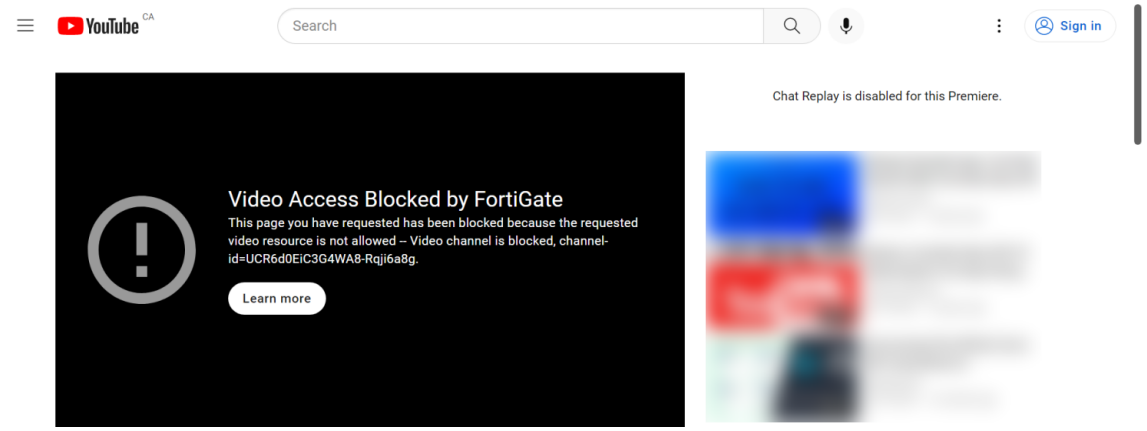
Example 1: blocking the video based on the URL

In this example, the user entered the URL of a blocked channel ID in their browser. The replacement message is displayed in the browser (full page).



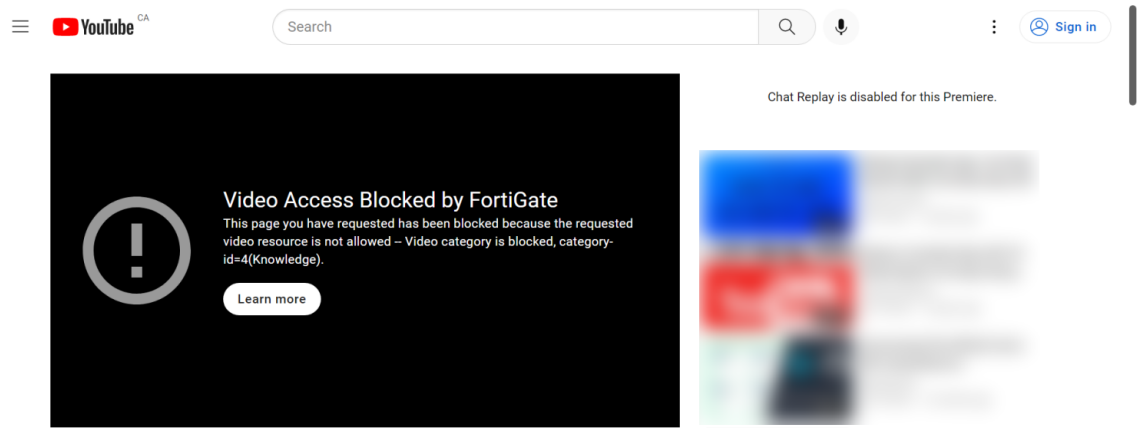
Example 2: blocking the video based on channel ID on YouTube

In this example, the user visited a blocked channel ID on the YouTube website. The replacement message is displayed in the video frame.



Example 3: blocking the video based on FortiGuard category on YouTube

In this example, the user visited a video on the YouTube website that belongs to a blocked FortiGuard category. The replacement message is displayed in the video frame.



DNS filter

You can apply DNS category filtering to control user access to web resources. You can customize the default profile, or create your own to manage network user access and apply it to a firewall policy, or you can add it to a DNS server on a FortiGate interface. For more information about configuring DNS, see DNS on page 358.

When both a DNS and a web filter are configured on a firewall policy, the DNS filter takes precedence. If certificate inspection is also used, the web filter replacement message might be shown instead of the SDNS block page. If both filters are set to block, SSL inspection is set to certificate inspection, and the destination web traffic is HTTPS, then the client receives a redirect to the SDNS server, which is then blocked by the web filter because the redirect URL cannot pass the certificate inspection.

DNS filtering has the following features:

- FortiGuard Filtering: filters the DNS request based on the FortiGuard domain rating.
- Botnet C&C domain blocking: blocks the DNS request for the known botnet C&C domains.
- External dynamic category domain filtering: allows you to define your own domain category.
- DNS safe search: enforces Google, Bing, and YouTube safe addresses for parental controls.
- Local domain filter: allows you to define your own domain list to block or allow.
- External IP block list: allows you to define an IP block list to block resolved IPs that match this list.
- DNS translation: maps the resolved result to another IP that you define.

Some DNS filter features require a subscription to FortiGuard Web Filtering.

DNS filtering connects to the FortiGuard secure DNS server over anycast by default. For more information about this configuration, see DNS over TLS and HTTPS on page 392.

The IPS engine handles the DNS filter in flow mode policies and queries the FortiGuard web filter server for FortiGuard categories. In proxy mode, the DNS proxy daemon handles the DNS filter and queries the FortiGuard SDNS server for FortiGuard categories. When a DNS filter profile is enabled in `config system dns-server`, the DNS proxy daemon handles the traffic.

A DNS filter profile can be applied in a policy to scan DNS traffic traversing the FortiGate (see Configuring a DNS filter profile on page 2388), or applied on the DNS server interface (see Applying DNS filter to FortiGate DNS server on page 2411).

DNS filter behavior in proxy mode

In cases where the DNS proxy daemon handles the DNS filter (described in the preceding section) and if DNS caching is enabled (this is the default setting), then the FortiGate will respond to subsequent DNS queries using the result in the DNS cache and will not forward these queries to a real DNS server.

There are two options to disable this behavior:

- Disable DNS caching globally.
- Remove the DNS filter profile from the proxy mode firewall policy or from the DNS server configured on a FortiGate interface.

To disable DNS caching globally:

```
config system dns
    set dns-cache-limit 0
end
```

There will be a performance impact to DNS queries since each query will not be cached, and will be forwarded to a real DNS server.

FortiGuard DNS rating service

DNS over TLS connections to the FortiGuard secure DNS server is supported. The CLI options are only available when `fortiguard-anycast` is enabled. DNS filtering connects to the FortiGuard secure DNS server over anycast by default.

To configure DoT to the secure DNS server in the CLI:

```
config system fortiguard
    set fortiguard-anycast enable
    set fortiguard-anycast-source fortinet
    set anycast-sdns-server-ip 0.0.0.0
    set anycast-sdns-server-port 853
end
```

The following topics provide information about DNS filters:

- Configuring a DNS filter profile on page 2388
- FortiGuard category-based DNS domain filtering on page 2393
- Botnet C&C domain blocking on page 2396

- DNS safe search on page 2400
- Local domain filter on page 2402
- DNS translation on page 2406
- Applying DNS filter to FortiGate DNS server on page 2411
- DNS inspection with DoT and DoH on page 2412
- DNS over QUIC and DNS over HTTP3 for transparent and local-in DNS modes on page 2416
- Troubleshooting for DNS filter on page 2421

Configuring a DNS filter profile

A DNS filter profile contains settings that enable or disable various forms of DNS filtering, including:

- FortiGuard filtering
- Botnet C&C domain blocking
- DNS safe search
- External dynamic category domain filtering
- Local domain filter
- External IP block list
- DNS translation

Once a DNS filter is configured, it can be applied to a firewall policy, or on a FortiGate DNS server if one is configured. In the following basic example, a DNS filter is created and applied to a firewall policy to scan DNS queries that pass through the FortiGate.

To configure a DNS filter profile in the GUI:

1. Go to *Security Profiles* > *DNS Filter* and click *Create New*, or edit an existing profile.
2. Configure the settings as needed.

Name	Enter a unique name for the profile.
Comments	Enter a comment (optional).
Redirect botnet C&C requests to Block Portal	Enable to block botnet website access at the DNS name resolution stage. See Botnet C&C domain blocking on page 2396 for more details.
Enforce 'Safe Search' on Google, Bing, YouTube	Enable to avoid explicit and inappropriate results in the Google, Bing, and YouTube search engines. See DNS safe search on page 2400 for more details.
Restrict YouTube Access	When *Enforce 'Safe Search' on Google, Bing, YouTube* is enabled, select either *Strict* or *Moderate* to restrict YouTube access by responding to DNS resolutions with CNAME restrict.youtube.com and restrictmoderate.youtube.com respectively.
FortiGuard Category Based Filter	Enable to use the FortiGuard domain rating database to inspect DNS traffic. A FortiGuard Web Filter license is required to use this option.

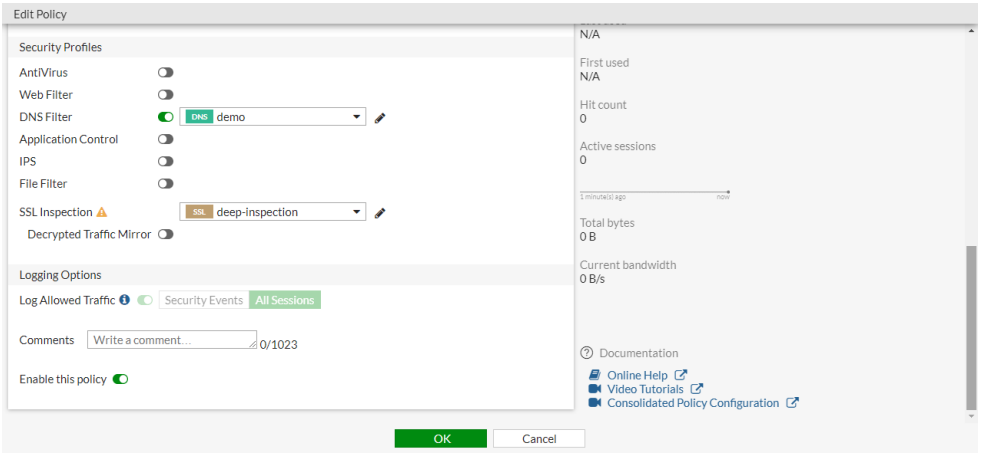
		Expand the category groups in the table to view and edit the FortiGuard category settings to *Allow*, *Monitor*, or *Redirect to Block Portal*. See FortiGuard category-based DNS domain filtering on page 2393 for more details.
Static Domain Filter		This section includes options related to the static domain filter.
	Domain Filter	Enable to define local static domain filters to allow or block specific domains. The local domain filter has a higher priority than the FortiGuard category-based domain filter. Click *Create New* in the table to add a domain filter and configure the following settings. • *Domain*: enter a domain. • *Type*: select *Simple*, *Reg. Expression*, or *Wildcard*. • *Action*: select *Redirect to Block Portal*, *Allow*, or *Monitor*. • *Status*: select *Enable* or *Disable*. See Local domain filter on page 2402 for more details.
	External IP Block Lists	Enable to add one or more external IP block lists. See IP address external feed on page 4712 for more details.
	DNS Translation	Enable to translate a DNS resolved IP address to another IP address specified on a per-policy basis. Click *Create New* in the table to add a DNS translation and configure the following settings. • *Type*: select *IPv4* or *IPv6*. • *Original Destination*: enter the address of a host or subnet that you want translated. When a resolved address in a DNS response matches this destination, the FortiGate will replace the address with the address in *Translated Destination*. • *Translated Destination*: enter the address of a host or subnet that you want the resolved address to be translated to. • *Network Mask*: enter the netmask for the original and translated destination. If a single host is used for the original and translated destination, set the netmask to *255.255.255.255*. • *Status*: select *Enable* or *Disable*. Enabling DNS translation will override matching DNS responses with translated IPs. See DNS translation on page 2406 for more details.
Options		This section includes other options related to the DNS filter.
	Redirect Portal IP	Set the IP address of the SDNS redirect portal. Select *Use FortiGuard Default*, or *Specify* and enter the IP address. When *FortiGuard Category Based Filter* categories are set to *Redirect to Block Portal*, the DNS response will use this IP address in its response to the client. If the client is accessing the domain on a web browser, they will be redirected to the block portal page on this address.

	Allow DNS requests when a rating error occurs	Enable to allow all domains when FortiGuard DNS servers fail, or they are unreachable from the FortiGate. When this happens, a log message is recorded in the DNS logs by default.
	Log all DNS queries and responses	Enable to log all domains visited (detailed DNS logging).
	Strip Encrypted Client Hello service parameters	Enable removal of the ECH service parameter from supporting DNS RRs. ECH information is stripped from DoH responses, forcing the browser to not use ECH for TLS connections.

3. Click *OK*.

To apply a DNS filter profile to a policy in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*, or edit an existing policy.
2. In the *Security Profiles* section, enable *DNS Filter* and select the DNS filter.



3. Configure the other settings as needed.
4. Click *OK*.

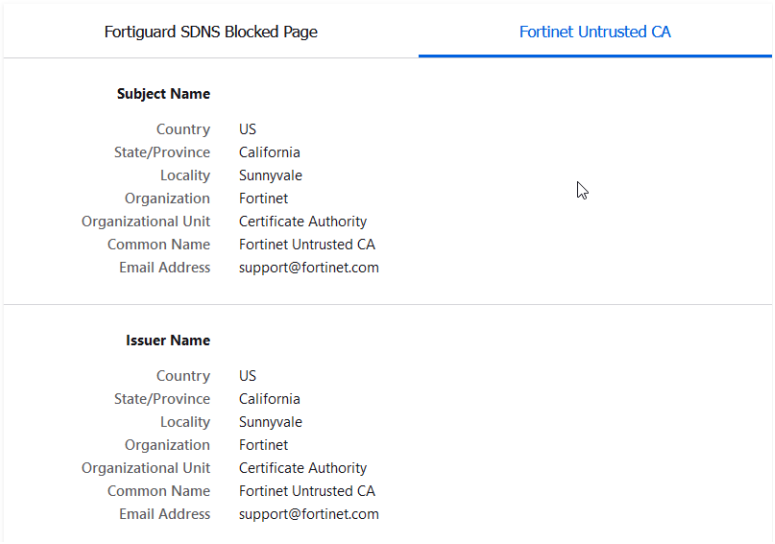
Redirecting to default Block Portal

By default, FortiGate redirects traffic to the FortiGuard SDNS block portal at 208.91.112.55, which is owned and hosted by Fortinet. When opening a blocked destination in the browser, two behaviors may occur:

- When opening a HTTP page, the blocked page is hosted on HTTP/80 and displays without errors.
- When opening a HTTPS page, the blocked page is hosted on HTTPS/443 and results in a warning.

  The warning is expected because the blocked page is redirected to the blocked portal on 208.91.112.55, but the page is served in HTTPS using the domain of the destination that the user is trying to reach. Therefore, a Fortinet CA must resign the SSL certificate in order to display the block page content. The Fortinet CA cannot be trusted natively because it is not a publicly trusted CA. As such, when you review the CA issuer, you find that the Common Name of the CA specifically indicates Fortinet Untrusted CA.

Certificate



CLI-only settings

The following DNS filter profile settings can only be configured in the CLI:

```
config dnsfilter profile
    edit <name>
        set block-action {block | redirect | block-servfail}
        set sdns-ftgd-err-log {enable | disable}
    next
end
```

`block-action {block | redirect | block-servfail}`	Set the action to take for blocked domains: • `block`: return NXDOMAIN for blocked domains. • `redirect`: redirect blocked domains to SDNS portal (default). • `block-servfail`: return SERVFAIL for blocked domains. When a FortiGuard or local domain filter category is set to *Redirect to Block Portal* in the GUI, the action is set to `block` in the CLI. By default, the `block-action` applied to a DNS profile is set to `redirect`.
`sdns-ftgd-err-log {enable | disable}`	Enable/disable FortiGuard SDNS rating error logging (default = enable).

To configure a DNS filter profile in the CLI:

```
config dnsfilter profile
    edit "demo"
        set comment ''
        config domain-filter
            unset domain-filter-table
        end
```

```
        config ftgd-dns
            set options error-allow
            config filters
                edit 2
                    set category 2
                    set action monitor
                next
                edit 7
                    set category 7
                    set action block
                next
                ...
                edit 22
                    set category 0
                    set action monitor
                next
            end
        end
        set log-all-domain enable
        set sdns-ftgd-err-log enable
        set sdns-domain-log enable
        set block-action redirect
        set block-botnet enable
        set safe-search enable
        set redirect-portal 93.184.216.34
        set youtube-restrict strict
        set strip-ech enable
    next
end
```

To apply a DNS filter profile to a policy in the CLI:

```
config firewall policy
    edit 1
        set name "Demo"
        set srcintf "port10"
        set dstintf "port9"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set logtraffic all
        set fsso disable
        set dnsfilter-profile "demo"
        set profile-protocol-options "default"
        set ssl-ssh-profile "deep-inspection"
        set nat enable
    next
end
```

FortiGuard category-based DNS domain filtering

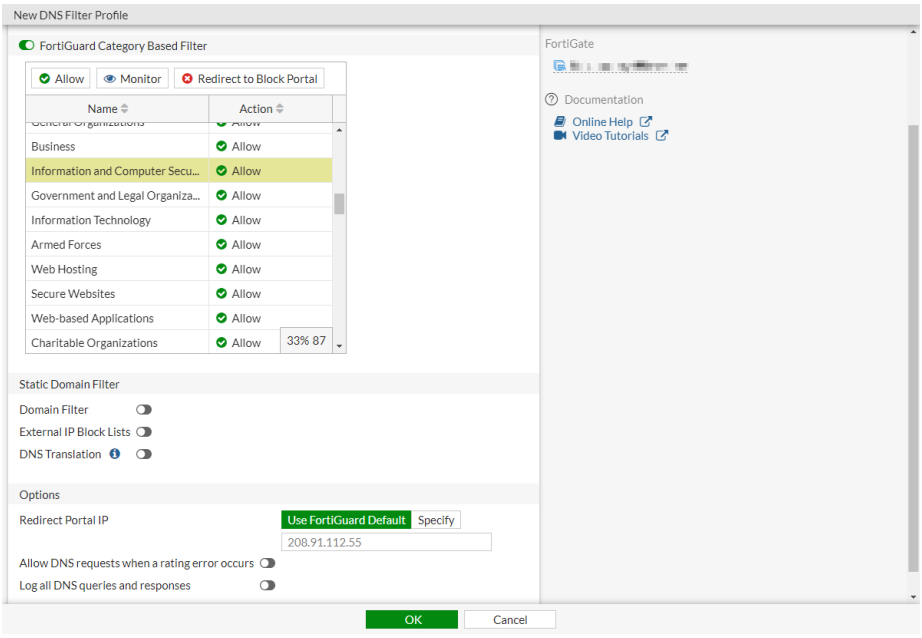
You can use the FortiGuard category-based DNS domain filter to inspect DNS traffic. This makes use of FortiGuard's continuously updated domain rating database for more reliable protection.

A DNS filter profile can be applied in a policy to scan DNS traffic traversing the FortiGate (see Configuring a DNS filter profile on page 2388), or applied on the DNS server interface (see Applying DNS filter to FortiGate DNS server on page 2411).

The FortiGate must have a FortiGuard Web Filter license to use the FortiGuard category-based filter.

To configure FortiGuard category-based DNS domain filtering in the GUI:

1. Go to *Security Profiles* > *DNS Filter* and click *Create New*, or edit an existing profile.
2. Enable *FortiGuard Category Based Filter*.
3. Select the category and then select *Allow*, *Monitor*, or *Redirect to Block Portal* for that category.
4. In the *Options* section, select a setting for *Redirect Portal IP*. Select either *Use FortiGuard Default* (208.91.112.55) or click *Specify* and enter another portal IP. The FortiGate will use the portal IP to replace the resolved IP in the DNS response packet.



5. Click *OK*.

To configure FortiGuard category-based DNS domain filtering in the CLI:

```
config dnsfilter profile
   edit "demo"
      set comment ''
      config domain-filter
         unset domain-filter-table
```

```
      end
      config ftgd-dns
         set options error-allow
         config filters
             edit 2
                 set category 2
                 set action monitor
             next
             edit 7
                 set category 7
                 set action monitor
             next
            ...
             edit 22
                 set category 0
                 set action monitor
             next
         end
      end
      set log-all-domain enable
      set sdns-ftgd-err-log enable
      set sdns-domain-log enable
      set block-action {redirect | block}
      set block-botnet enable
      set safe-search enable
      set redirect-portal 93.184.216.34
      set youtube-restrict strict
   next
end
```

You can use the `get webfilter categories` command to determine the web filtering category that corresponds to a given category ID.

Verifying the logs

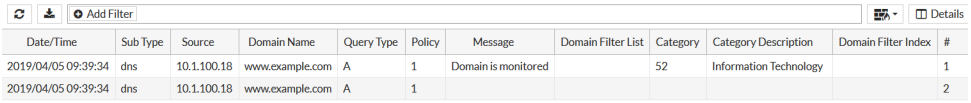
From your internal network PC, use a command line tool, such as dig or nslookup, to do a DNS query for some domains. For example:

```
#dig www.example.com
;; ->>HEADER<<- opcode: QUERY; status: NOERROR; id: 61252
;; Flags: qr rd ra; QUERY: 1; ANSWER: 1; AUTHORITY: 13; ADDITIONAL: 11

;; QUESTION SECTION:
;; www.example.com.             IN      A

;; ANSWER SECTION:
www.example.com.        17164   IN      A       93.184.216.34

;; AUTHORITY SECTION:
com.                    20027   IN      NS      h.gtld-servers.net.
com.                    20027   IN      NS      i.gtld-servers.net.
```

```
com.                    20027   IN      NS      f.gtld-servers.net.
com.                    20027   IN      NS      d.gtld-servers.net.
com.                    20027   IN      NS      j.gtld-servers.net.
com.                    20027   IN      NS      l.gtld-servers.net.
com.                    20027   IN      NS      e.gtld-servers.net.
com.                    20027   IN      NS      a.gtld-servers.net.
com.                    20027   IN      NS      k.gtld-servers.net.
com.                    20027   IN      NS      g.gtld-servers.net.
com.                    20027   IN      NS      m.gtld-servers.net.
com.                    20027   IN      NS      c.gtld-servers.net.
com.                    20027   IN      NS      b.gtld-servers.net.

;; ADDITIONAL SECTION:
a.gtld-servers.net.     21999   IN      A       192.5.6.30
a.gtld-servers.net.     21999   IN      AAAA    2001:503:a83e::2:30
b.gtld-servers.net.     21997   IN      A       192.33.14.30
b.gtld-servers.net.     21997   IN      AAAA    2001:503:231d::2:30
c.gtld-servers.net.     21987   IN      A       192.26.92.30
c.gtld-servers.net.     20929   IN      AAAA    2001:503:83eb::30
d.gtld-servers.net.     3340    IN      A       192.31.80.30
d.gtld-servers.net.     3340    IN      AAAA    2001:500:856e::30
e.gtld-servers.net.     19334   IN      A       192.12.94.30
e.gtld-servers.net.     19334   IN      AAAA    2001:502:1ca1::30
f.gtld-servers.net.     3340    IN      A       192.35.51.30

;; Received 509 B
;; Time 2019-04-05 09:39:33 PDT
;; From 172.16.95.16@53(UDP) in 3.8 ms
```

To check the DNS filter log in the GUI:

1. Go to *Log & Report* > *Security Events*.
2. Click the *DNS Query* card name. There are logs for the DNS traffic that just passed through the FortiGate with the FortiGuard rating for the domain name.



Date/Time	Sub Type	Source	Domain Name	Query Type	Policy	Message	Domain Filter List	Category	Category Description	Domain Filter Index	#
2019/04/05 09:39:34	dns	10.1.100.18	www.example.com	A	1	Domain is monitored		52	Information Technology		1
2019/04/05 09:39:34	dns	10.1.100.18	www.example.com	A	1						2

To check the DNS filter log in the CLI:

```
# execute log filter category utm-dns

# execute log display
2 logs found.
2 logs returned.

1: date=2019-04-05 time=09:39:34 logid="1501054802" type="utm" subtype="dns" eventtype="dns-
response" level="notice" vd="vdom1" eventtime=1554482373 policyid=1 sessionid=50868
srcip=10.1.100.18 srcport=34308 srcintf="port10" srcintfrole="undefined" dstip=172.16.95.16
dstport=53 dstintf="port9" dstintfrole="undefined" proto=17 profile="demo" xid=17647
```

```
qname="www.example.com" qtype="A" qtypeval=1 qclass="IN" ipaddr="93.184.216.34" msg="Domain is
monitored" action="pass" cat=52 catdesc="Information Technology"

2: date=2019-04-05 time=09:39:34 logid="1500054000" type="utm" subtype="dns" eventtype="dns-query"
level="information" vd="vdom1" eventtime=1554482373 policyid=1 sessionid=50868 srcip=10.1.100.18
srcport=34308 srcintf="port10" srcintfrole="undefined" dstip=172.16.95.16 dstport=53
dstintf="port9" dstintfrole="undefined" proto=17 profile="demo" xid=17647 qname="www.example.com"
qtype="A" qtypeval=1 qclass="IN"
```

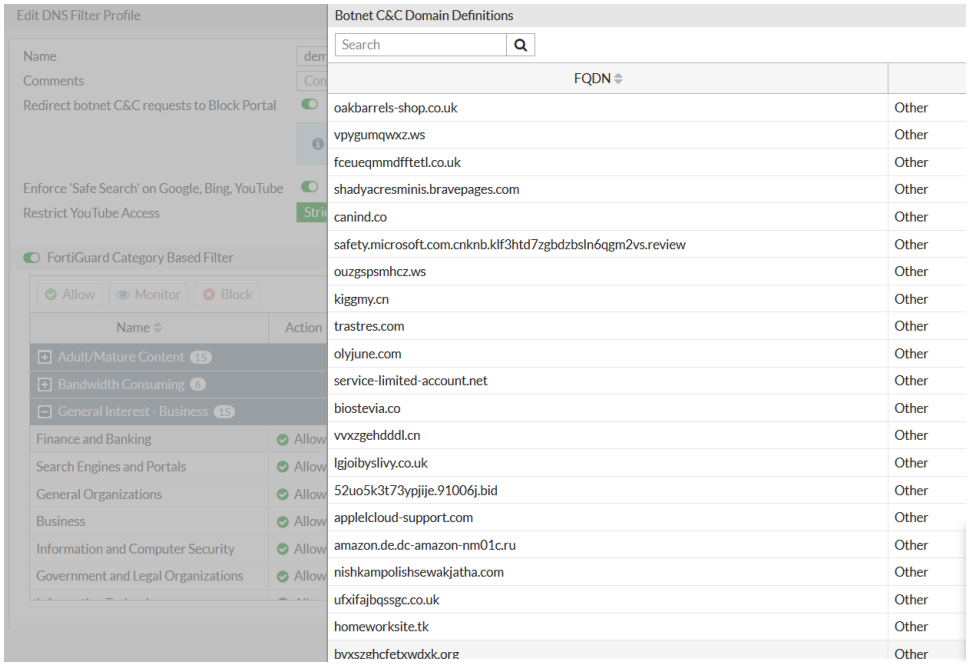
Botnet C&C domain blocking

FortiGuard Service continually updates the botnet C&C domain list. The botnet C&C domain blocking feature can block the botnet website access at the DNS name resolving stage. This provides additional protection for your network.

A DNS filter profile can be applied in a policy to scan DNS traffic traversing the FortiGate (see Configuring a DNS filter profile on page 2388), or applied on the DNS server interface (see Applying DNS filter to FortiGate DNS server on page 2411).

To configure botnet C&C domain blocking in the GUI:

1. Go to *Security Profiles* > *DNS Filter* and click *Create New*, or edit an existing profile.
2. Enable *Redirect botnet C&C requests to Block Portal*.
3. Optionally, click the *botnet package* link. The *Botnet C&C Domain Definitions* pane opens, which displays the latest list.



4. Configure the other settings as needed.
5. Click *OK*.

To configure botnet C&C domain blocking in the CLI:

```
config dnsfilter profile
   edit "demo"
      set comment ''
      config domain-filter
         unset domain-filter-table
      end
      config ftgd-dns
         set options error-allow
         config filters
            ...
         end
      end
      set log-all-domain enable
      set sdns-ftgd-err-log enable
      set sdns-domain-log enable
      set block-action block
      set block-botnet enable
      set safe-search enable
      set redirect-portal 208.91.112.55
      set youtube-restrict strict
   next
end
```

Verifying the logs

Select a botnet domain from that list. From your internal network PC, use a command line tool, such as dig or nslookup, to send a DNS query to traverse the FortiGate. For example:

```
#dig canind.co
;; ->>HEADER<<- opcode: QUERY; status: NOERROR; id: 997
;; Flags: qr rd; QUERY: 1; ANSWER: 1; AUTHORITY: 0; ADDITIONAL: 0

;; QUESTION SECTION:
;; canind.co.                   IN      A

;; ANSWER SECTION:
canind.co.              60      IN      A       208.91.112.55

;; Received 43 B
;; Time 2019-04-05 09:55:21 PDT
;; From 172.16.95.16@53(UDP) in 0.3 ms
```

The botnet domain query was blocked and redirected to the portal IP (208.91.112.55) .

To check the DNS filter log in the GUI:

1. Go to *Log & Report > Security Events*.
2. Click the *DNS Query* card name to view the DNS query blocked as a botnet domain.

Date/Time	Sub Type	Source	Domain Name	Query Type	Policy	Message	Domain Filter List	Category	Category Description
2019/04/04 16:43:59	dns	10.1.100.18	canind.co	A	1	Domain was blocked by dns botnet C&C			
2019/04/04 16:43:59	dns	10.1.100.18	canind.co	A	1				

To check the DNS filter log in the CLI:

```
(vdom1) # execute log filter category utm-dns

(vdom1) # execute log display
2 logs found.
2 logs returned.

1: date=2019-04-04 time=16:43:59 logid="1501054601" type="utm" subtype="dns" eventtype="dns-
response" level="warning" vd="vdom1" eventtime=1554421439 policyid=1 sessionid=14135
srcip=10.1.100.18 srcport=57447 srcintf="port10" srcintfrole="undefined" dstip=172.16.95.16
dstport=53 dstintf="port9" dstintfrole="undefined" proto=17 profile="demo" xid=24339
qname="canind.co" qtype="A" qtypeval=1 qclass="IN" msg="Domain was blocked by dns botnet C&C"
action="redirect" botnetdomain="canind.co"

2: date=2019-04-04 time=16:43:59 logid="1500054000" type="utm" subtype="dns" eventtype="dns-query"
level="information" vd="vdom1" eventtime=1554421439 policyid=1 sessionid=14135 srcip=10.1.100.18
srcport=57447 srcintf="port10" srcintfrole="undefined" dstip=172.16.95.16 dstport=53
dstintf="port9" dstintfrole="undefined" proto=17 profile="demo" xid=24339 qname="canind.co"
qtype="A" qtypeval=1 qclass="IN"
```

Botnet C&C IPDB blocking

FortiOS also maintains a botnet C&C IP address database (IPDB). If a DNS query response IP address (resolved IP address) matches an entry inside the botnet IPDB, this DNS query is blocked by the DNS filter botnet C&C.

To view the botnet IPDB list in the CLI:

```
(global) # diagnose sys botnet list 9000 10
9000. proto=TCP ip=103.228.28.166, port=80, rule_id=7630075, name_id=3, hits=0
9001. proto=TCP ip=5.9.32.166, port=481, rule_id=4146631, name_id=7, hits=0
9002. proto=TCP ip=91.89.44.166, port=80, rule_id=48, name_id=96, hits=0
9003. proto=TCP ip=46.211.46.166, port=80, rule_id=48, name_id=96, hits=0
9004. proto=TCP ip=77.52.52.166, port=80, rule_id=48, name_id=96, hits=0
9005. proto=TCP ip=98.25.53.166, port=80, rule_id=48, name_id=96, hits=0
9006. proto=TCP ip=70.120.67.166, port=80, rule_id=48, name_id=96, hits=0
9007. proto=TCP ip=85.253.77.166, port=80, rule_id=48, name_id=96, hits=0
9008. proto=TCP ip=193.106.81.166, port=80, rule_id=48, name_id=96, hits=0
9009. proto=TCP ip=58.13.84.166, port=80, rule_id=48, name_id=96, hits=0
```

Select an IP address from the IPDB list and use a reverse lookup service to find its corresponding domain name. From your internal network PC, use a command line tool, such as dig or nslookup, to query this domain and verify that it is blocked by the DNS filter botnet C&C. For example:

```
# dig cpe-98-25-53-166.sc.res.rr.com
;; ->>HEADER<<- opcode: QUERY; status: NOERROR; id: 35135
```

```
;; Flags: qr rd; QUERY: 1; ANSWER: 1; AUTHORITY: 0; ADDITIONAL: 0

;; QUESTION SECTION:
;; cpe-98-25-53-166.sc.res.rr.com.              IN      A

;; ANSWER SECTION:
cpe-98-25-53-166.sc.res.rr.com. 60      IN      A       208.91.112.55

;; Received 64 B
;; Time 2019-04-05 11:06:47 PDT
;; From 172.16.95.16@53(UDP) in 0.6 ms
```

Since the resolved IP address matches the botnet IPDB, the query was blocked and redirected to the portal IP (208.91.112.55) .

To check the DNS filter log in the GUI:

1. Go to *Log & Report* > *DNS Query* to view the DNS query blocked by botnet C&C IPDB.

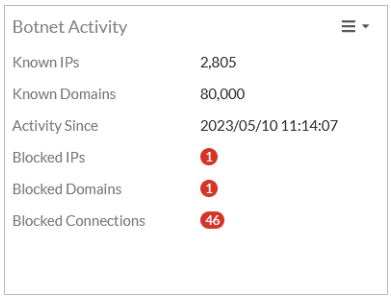
Date/Time	Sub Type	Source	Domain Name	Query Type	Policy	Message	Domain Filter List	Category	Category Description
2019/04/05 11:06:48	dns	10.1.100.18	cpe-98-25-53-166.sc.res.rr.com	A	1	Domain was blocked by dns botnet C&C			
2019/04/05 11:06:48	dns	10.1.100.18	cpe-98-25-53-166.sc.res.rr.com	A	1				

To check the DNS filter log in the CLI:

```
(global) # execute log filter category utm-dns

(global) # execute log display
2 logs found.
2 logs returned.

1: date=2019-04-05 time=11:06:48 logid="1501054600" type="utm" subtype="dns" eventtype="dns-
response" level="warning" vd="vdom1" eventtime=1554487606 policyid=1 sessionid=55232
srcip=10.1.100.18 srcport=60510 srcintf="port10" srcintfrole="undefined" dstip=172.16.95.16
dstport=53 dstintf="port9" dstintfrole="undefined" proto=17 profile="demo" xid=16265 qname="cpe-
98-25-53-166.sc.res.rr.com" qtype="A" qtypeval=1 qclass="IN" ipaddr="93.184.216.34" msg="Domain
was blocked by dns botnet C&C" action="redirect" botnetip=98.25.53.166

2: date=2019-04-05 time=11:06:48 logid="1500054000" type="utm" subtype="dns" eventtype="dns-query"
level="information" vd="vdom1" eventtime=1554487606 policyid=1 sessionid=55232 srcip=10.1.100.18
srcport=60510 srcintf="port10" srcintfrole="undefined" dstip=172.16.95.16 dstport=53
dstintf="port9" dstintfrole="undefined" proto=17 profile="demo" xid=16265 qname="cpe-98-25-53-
166.sc.res.rr.com" qtype="A" qtypeval=1 qclass="IN"
```

To check botnet activity:

1. Go to *Dashboard* > *Status* and locate the *Botnet Activity* widget.



2. If you do not see the widget, click *Add Widget*, and add the *Botnet Activity* widget.

DNS safe search

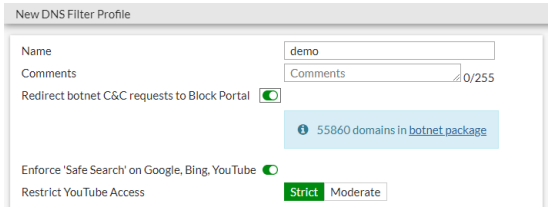
The DNS safe search option helps avoid explicit and inappropriate results in the Google, Bing, DuckDuckGo, Qwant, and YouTube search engines. The FortiGate responds with content filtered by the search engine.

For individual search engine safe search specifications, refer to the documentation for Google, Bing, DuckDuckGo, Qwant, and YouTube.

A DNS filter profile can be applied in a policy to scan DNS traffic traversing the FortiGate (see Configuring a DNS filter profile on page 2388), or applied on the DNS server interface (see Applying DNS filter to FortiGate DNS server on page 2411).

To configure safe search in the GUI:

1. Go to *Security Profiles* > *DNS Filter* and click *Create New*, or edit an existing profile.
2. Enable *Enforce 'Safe search' on Google, Bing, YouTube* (this setting also applies safe search on DuckDuckGo and Qwant).
3. For *Restrict YouTube Access*, click *Strict* or *Moderate*.



4. Configure the other settings as needed.
5. Click *OK*.

To configure safe search in the CLI:

```
config dnsfilter profile
    edit "demo"
        config ftgd-dns
```

```
            set options error-allow
            config filters
                edit 2
                    set category 2
                next
               ...
            end
        end
        set log-all-domain enable
        set block-botnet enable
        set safe-search enable
        set youtube-restrict strict
    next
end
```

Verifying the logs

From your internal network PC, use a command line tool, such as dig or nslookup, and perform a DNS query on www.bing.com. For example:

```
# dig www.bing.com
;; ->>HEADER<<- opcode: QUERY; status: NOERROR; id: 46568
;; Flags: qr rd ra; QUERY: 1; ANSWER: 2; AUTHORITY: 0; ADDITIONAL: 0

;; QUESTION SECTION:
;; www.bing.com.                IN      A

;; ANSWER SECTION:
www.bing.com.           103     IN      CNAME   strict.bing.com
strict.bing.com.        103     IN      A       204.79.197.220

;; Received 67 B
;; Time 2019-04-05 14:34:52 PDT
;; From 172.16.95.16@53(UDP) in 196.0 ms
```

The DNS query for www.bing.com returns with a CNAME strict.bing.com, and an A record for the CNAME. The user's web browser then connects to this address with the same search engine UI, but any explicit content search is filtered out.

To check the DNS filter log in the GUI:

1. Go to *Log & Report > Security Events*.
2. Click the *DNS Query* card name.

Date/Time	Sub Type	Source	Domain Name	Query Type	Policy	Message	Domain Filter List	Category	Category Description
2019/04/05 14:34:53	dns	10.1.100.18	www.bing.com	A	1	DNS Safe Search enforced		41	Search Engines and Portals
2019/04/05 14:34:53	dns	10.1.100.18	www.bing.com	A	1				

   The DNS filter log in FortiOS shows a message of *DNS Safe Search enforced*.

To check the DNS filter log in the CLI:

```
# execute log filter category utm-dns
# execute log display
2 logs found.
2 logs returned.

1: date=2019-04-05 time=14:34:53 logid="1501054804" type="utm" subtype="dns" eventtype="dns-
response" level="notice" vd="vdom1" eventtime=1554500093 policyid=1 sessionid=65955
srcip=10.1.100.18 srcport=36575 srcintf="port10" srcintfrole="undefined" dstip=172.16.95.16
dstport=53 dstintf="port9" dstintfrole="undefined" proto=17 profile="demo" xid=59573
qname="www.bing.com" qtype="A" qtypeval=1 qclass="IN" ipaddr="204.79.197.220" msg="DNS Safe Search
enforced" action="pass" sscname="strict.bing.com" cat=41 catdesc="Search Engines and Portals"

2: date=2019-04-05 time=14:34:53 logid="1500054000" type="utm" subtype="dns" eventtype="dns-query"
level="information" vd="vdom1" eventtime=1554500092 policyid=1 sessionid=65955 srcip=10.1.100.18
srcport=36575 srcintf="port10" srcintfrole="undefined" dstip=172.16.95.16 dstport=53
dstintf="port9" dstintfrole="undefined" proto=17 profile="demo" xid=59573 qname="www.bing.com"
qtype="A" qtypeval=1 qclass="IN"
```

Local domain filter

In addition to the FortiGuard category-based domain filter, you can define a local static domain filter to allow or block specific domains.

In a DNS filter profile, the local domain filter has a higher priority than FortiGuard category-based domain filter. DNS queries are scanned and matched first with the local domain filter.

- If the local domain filter list has no match, then the FortiGuard category-based domain filter is used. If a DNS query domain name rating belongs to the block category, the query is blocked and redirected. If the FortiGuard category-based filter has no match, then the original resolved IP address is returned to the client DNS resolver.
- If the local domain filter action is set to block and an entry matches, then that DNS query is blocked and redirected.
- If the local domain filter action is set to allow and an entry matches, it will skip the FortiGuard category-based domain filter and directly return to the client DNS resolver.
- If the local domain filter action is set to monitor and an entry matches, it will skip the FortiGuard category-based domain filter, directly return to the client DNS resolver, and log the resolution.

A DNS filter profile can be applied in a policy to scan DNS traffic traversing the FortiGate (see Configuring a DNS filter profile on page 2388), or applied on the DNS server interface (see Applying DNS filter to FortiGate DNS server on page 2411).

In this example, a DNS filter profile is configured and applied to a firewall policy running proxy-based inspection mode.

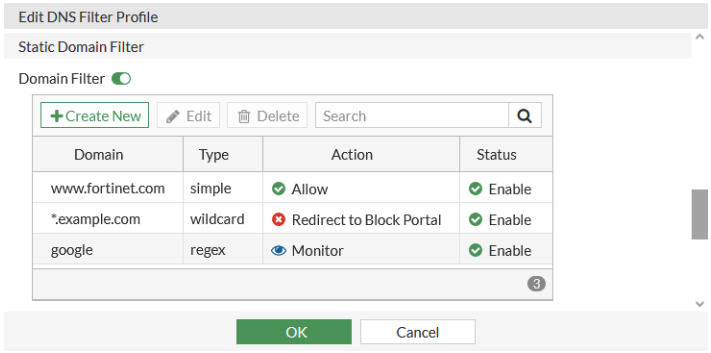
To configure the local domain filter in the GUI:

1. Go to *Security Profiles* > *DNS Filter* and click *Create New*, or edit an existing profile.
2. Set *Name* to *demo*.

3. In the *Static Domain Filter* section, enable *Domain Filter*.
4. Click *Create New*. The *Create Domain Filter* pane opens.
5. Enter a domain, and select a *Type* and *Action*. This example has three filters:

Domain	Type	Action
www.fortinet.com	Simple	Allow
*.example.com	Wildcard	Redirect to Block Portal
google	Reg. Expression	Monitor

6. Click *OK*. The entry appears in the table.



7. In the *FortiGuard Category Based Filter* table, set *General Interest - Business > Search Engines and Portals* to *Redirect to Block Portal*.
8. Configure the remaining settings as required.
9. Click *OK*.

To apply the DNS filter to a policy-mode policy in the GUI:

1. Go to *Policy & Objects > Firewall Policy* and click *Create New*, or edit an existing policy.
2. Configure the *Incoming Interface*, *Outgoing Interface*, *Source*, *Destination*, and *Service* as required.
3. Set *Inspection Mode* to *Proxy-based*.
4. Enable *DNS Filter* and select the *demo* filter.
5. Set *SSL Inspection* to *certificate-inspection*.
6. Configure the remaining settings as required.
7. Click *OK*.

To configure the local domain filter in the CLI:

```
config dnsfilter domain-filter
    edit 1
        set name "demo"
        set comment ''
        config entries
            edit 1
                set domain "www.fortinet.com"
                set type simple
                set action allow
```

```
                set status enable
            next
            edit 2
                set domain "*.example.com"
                set type wildcard
                set action block
                set status enable
            next
            edit 3
                set domain "google"
                set type regex
                set action monitor
                set status enable
            next
        end
        config domain-filter
            set domain-filter-table 1
        end
        config ftgd-dns
            config filters
                edit 23
                    set category 41
                    set action block
                next
            end
        end
    next
end
```

Wildcard entries are converted to regular expressions by FortiOS. As a result, wildcards will match any suffix, as long as there is a word boundary following the search term.

For example:

```
config entries
    edit 1
        set domain "*.host"
        set type wildcard
    next
end
```

will match `wp36.host` and `wp36.host.pressdns.com`, but not `wp36.host123.pressdnds.com`.

To avoid this, use an explicit regular expression search string:

```
config entries
    edit 1
        set domain "^.*\\.host$"
        set type regexp
    next
end
```

To apply the DNS filter to a proxy-mode policy in the CLI:

```
config firewall policy
    edit 1
        set name "port3-port1"
        set srcintf "port3"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "certificate-inspection"
        set dnsfilter-profile "demo"
        set logtraffic all
        set nat enable
    next
end
```

Testing and Verification

On a client computer, perform DNS lookup on the three domains:

Domain	DNS query result	Log
www.fortinet.com	Allowed. Resolved to correct IP.	None
www.example.com	Blocked. Redirected to IP of block page.	Deny log
www.google.com	Allowed. Resolved to correct IP.	Allow log

To check the DNS filter log in the GUI:

1. Go to *Log & Report* > *Security Events*.
2. Click the *DNS Query* card name to show the logs.

To check the DNS filter log in the CLI:

```
# execute log display
71 logs found.
10 logs returned.

1: date=2022-08-17 time=18:16:50 eventtime=1660785410733825945 tz="-0700" logid="1501054401"
type="utm" subtype="dns" eventtype="dns-response" level="information" vd="root" policyid=3
poluuid="6b80057c-1e76-51ed-c629-5fe117f24362" policytype="policy" sessionid=820031
srcip=192.168.0.10 srcport=52674 srccountry="Reserved" srcintf="port3" srcintfrole="lan"
dstip=8.8.8.8 dstport=53 dstcountry="United States" dstintf="port1" dstintfrole="wan" proto=17
profile="demo" xid=4352 qname="www.google.com" qtype="AAAA" qtypeval=28 qclass="IN"
```

```
ipaddr="2607:f8b0:400a:803::2004" msg="Domain was allowed because it is in the domain-filter list"
action="pass" domainfilteridx=1 domainfilterlist="demo"

2: date=2022-08-17 time=18:16:50 eventtime=1660785410718697625 tz="-0700" logid="1501054401"
type="utm" subtype="dns" eventtype="dns-response" level="information" vd="root" policyid=3
poluuid="6b80057c-1e76-51ed-c629-5fe117f24362" policytype="policy" sessionid=820030
srcip=192.168.0.10 srcport=52673 srccountry="Reserved" srcintf="port3" srcintfrole="lan"
dstip=8.8.8.8 dstport=53 dstcountry="United States" dstintf="port1" dstintfrole="wan" proto=17
profile="demo" xid=4096 qname="www.google.com" qtype="A" qtypeval=1 qclass="IN"
ipaddr="172.217.14.228" msg="Domain was allowed because it is in the domain-filter list"
action="pass" domainfilteridx=1 domainfilterlist="demo"

3: date=2022-08-17 time=18:16:40 eventtime=1660785401007448812 tz="-0700" logid="1501054400"
type="utm" subtype="dns" eventtype="dns-response" level="warning" vd="root" policyid=3
poluuid="6b80057c-1e76-51ed-c629-5fe117f24362" policytype="policy" sessionid=820019
srcip=192.168.0.10 srcport=59950 srccountry="Reserved" srcintf="port3" srcintfrole="lan"
dstip=8.8.8.8 dstport=53 dstcountry="United States" dstintf="port1" dstintfrole="wan" proto=17
profile="demo" xid=3840 qname="www.example.com" qtype="AAAA" qtypeval=28 qclass="IN"
ipaddr="2620:101:9000:53::55" msg="Domain was blocked because it is in the domain-filter list"
action="redirect" domainfilteridx=1 domainfilterlist="demo"

4: date=2022-08-17 time=18:16:40 eventtime=1660785401006872790 tz="-0700" logid="1501054400"
type="utm" subtype="dns" eventtype="dns-response" level="warning" vd="root" policyid=3
poluuid="6b80057c-1e76-51ed-c629-5fe117f24362" policytype="policy" sessionid=820018
srcip=192.168.0.10 srcport=59949 srccountry="Reserved" srcintf="port3" srcintfrole="lan"
dstip=8.8.8.8 dstport=53 dstcountry="United States" dstintf="port1" dstintfrole="wan" proto=17
profile="demo" xid=3584 qname="www.example.com" qtype="A" qtypeval=1 qclass="IN"
ipaddr="208.91.112.55" msg="Domain was blocked because it is in the domain-filter list"
action="redirect" domainfilteridx=1 domainfilterlist="demo"
```

DNS translation

DNS translation is a technique that allows the translation of a DNS-resolved IP address to another specified IP address on a per-policy basis.

One of its use cases is as an alternative to hairpin NAT. By utilizing DNS translation, internal clients can access internal servers using external DNS names. This is achieved by translating DNS responses to internal IP addresses, which can streamline configurations and mitigate some of the complexities associated with Hairpin NAT. However, hairpin NAT can offer faster performance in certain scenarios by eliminating the need for DNS queries, thereby providing direct routing and potentially reducing overhead. The performance difference may not always be significant and largely depends on the specific network setup. For more information, see Hairpin NAT on page 1977.

How DNS translation works:

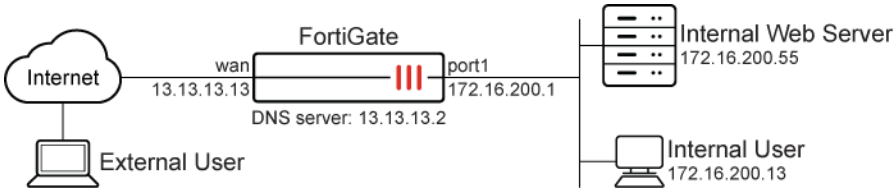
1. DNS Query Interception: When an internal client queries the DNS for the external IP address of an internal server the DNS server intercepts this request.
2. DNS Response Modification: Instead of returning the external IP address, the DNS server responds with the internal IP address of the server.
3. Direct Communication: The internal client then communicates directly with the internal server using its internal IP address, bypassing the need for NAT hairpinning.

DNS translation supports Service (SRV) records over the DNS filter profile for IPv4 and IPv6 and for flow and proxy inspection modes. The translation detects and logs DNS QTYP=SRV. You can view the records in the GUI by going to *Log & Report* > *Security Events* and filtering by *Query Type = SRV*.

A DNS filter profile can be applied in a policy to scan DNS traffic traversing the FortiGate (see Configuring a DNS filter profile on page 2388), or applied on the DNS server interface (see Applying DNS filter to FortiGate DNS server on page 2411).

Example

This configuration allows both internal and remote employees to use the same domain name to access the server, such as www.pochiya.com. DNS translation ensures consistency, avoiding confusion and complications in network configurations. This setup is particularly useful for companies that save bookmarks to internal web servers using public domain names in users' profiles, ensuring access whether connected to the internal LAN or the internet.



In this scenario, the user can access saved bookmarks containing public domain names from both their office computer on the internal network and their personal computer over the internet.

Before continuing, make sure that the port1 and wan interfaces have been configured with valid IP addresses and some publicly accessible domain names are saved as bookmarks in the user profile.

It is assumed that you have administrative access, the FortiGate is incorporated into your network, and the external IP address is the same as the FortiGate wan interface.

The configuration has three steps:

1. Create and attach DNS translation object: Map the external IP address to the internal IP address and attach the DNS translation object to a DNS filter profile, allowing internal clients to receive the internal IP address when querying the external DNS name.
2. Apply the DNS filter profile to a policy: DNS traffic is filtered through the DNS translation profile, allowing internal users to access the internal server using the external DNS name.
3. Verify the result: Confirm that the employee can access the internal web server using the external DNS name.

To create and attach a DNS translation object in the GUI:

1. Go to *Security Profiles* > *DNS Filter* and click *Create New*.
2. Set the *Name* of the filter, such as *Reflective DNS*.
3. Under *Static Domain Filter*, enable *DNS Translation*.
4. In the *DNS Translation* table, click *Create New* and configure the new translation:

Original Destination	13.13.13.13
Translated Destination	172.16.200.55

Network Mask	255.255.255.255
Status	Enable

5. Click *OK*.
6. Enable *Log all DNS queries and responses*.
7. Click *OK*.

To configure a firewall policy for DNS translation in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy*, and click *Create New*.
2. Configure the following:

Field	Value
Name	lan-wan
Schedule	always
Action	ACCEPT
Type	Standard
Incoming interface	port1
Outgoing interface	wan
Source	all
Destination	all
Service	ALL
NAT	Enabled
IP pool configuration	Use Outgoing Interface Address
DNS Filter	Enable and select *Reflective DNS*
Log allowed traffic	All sessions

3. Click *OK*.

To configure DNS translation in the CLI:

1. Create and attach a DNS translation object :

```
config dnsfilter profile
    edit "Reflective DNS"
        set log-all-domain enable
        config dns-translation
            edit 1
                set src 13.13.13.13
                set dst 172.16.200.55
                set netmask 255.255.255.255
            next
```

```
        end
    next
end
```

2. Configure a firewall policy for DNS translation:

```
config firewall policy
    edit 1
        set name "lan-wan"
        set srcintf "port1"
        set dstintf "wan"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set dnsfilter-profile "Reflective DNS"
        set logtraffic all
        set nat enable
   next
end
```

To verify the results:

1. Access the internal machine and use the saved bookmarks that contain the public domain name of the internal web server to access it. This should be successful, and the web server should load correctly. On the FortiGate, this can be verified using the logs and session list
2. On the FortiGate, go to *Log & Report > Security Events*, expand the *DNS Query* widget, and double click the desired log entry to view its details, or use the CLI to view the logs:

```
#execute log filter category 15
#execute log display
1: date =2024-12-13 time=07:39:37 eventtime=1734032377518565638 tz="+1200" logid="1501054802"
type="utm" subtype="dns" eventtype="dns-response" level="notice" vd="root" policyid=1
poluuid="62063350-b324-51ef-75c1-7c13a319762c" policytype="policy" sessionid=5147
srcip=172.16.200.13 srcport=60866 srccountry="Reserved" srcintf="port1"
srcintfrole="undefined" dstip=13.13.13.2 dstport=53 dstcountry="United States" dstintf="wan"
dstintfrole="undefined" proto=17 profile="Reflective DNS" xid=64 qname="www.pochiya.com"
qtype="A" qtypeval=1 qclass="IN" ipaddr="172.16.200.55" msg="Domain is monitored"
action="pass" cat==49 catdesc="Business" translationid=1

2: date=2024-12-13 time=07:39:37 eventtime=1734032377517238918 tz="+1200" logid="1500054000"
type="utm" subtype="dns" eventtype="dns-query" level="information" vd="root" policyid=1
poluuid="62063350-b324-51ef-75c1-7c13a319762c" policytype="policy" sessionid=5147
srcip=172.16.200.13 srcport=60866 srccountry="Reserved" srcintf="port1"
srcintfrole="undefined" dstip=13.13.13.2 dstport=53 dstcountry="United States" dstintf="wan"
dstintfrole="undefined" proto=17 profile="Reflective DNS" xid=64 qname="www.pochiya.com"
qtype="A" qtypeval=1 qclass="IN"
```

If the logs are missing, go to *Log & Report > Forward Traffic* and increase the time frame to ensure that all pertinent log entries are available.

DNS translation network mask

The following is an example of DNS translation that uses a network mask other than /32 (255.255.255.255):

To configure DNS translation in the CLI:

```
config dns-translation
        edit 1
           set src 93.184.216.34
           set dst 1.2.3.4
           set netmask 255.255.224.0
        next
    end
```

To check DNS translation using a command line tool after DNS translation:

```
# dig www.example.com
;; ->>HEADER<<- opcode: QUERY; status: NOERROR; id: 6736
;; Flags: qr rd ra; QUERY: 1; ANSWER: 1; AUTHORITY: 2; ADDITIONAL: 0

;; QUESTION SECTION:
;; www.example.com.             IN      A

;; ANSWER SECTION:
www.example.com.        29322   IN      A       1.2.24.34

;; AUTHORITY SECTION:
example.com.            13954   IN      NS      a.iana-servers.net.
example.com.            13954   IN      NS      b.iana-servers.net.

;; Received 97 B
;; Time 2019-04-08 12:04:30 PDT
;; From 172.16.95.16@53(UDP) in 2.0 ms
```

The binary arithmetic to convert 93.184.216.34 to 1.2.3.4 with the subnet mask is as follows:

1. AND src(Original IP) with negative netmask (93.184.216.34 & ~255.255.224.0):

   ```
   01011101.10111000.11011000.00100010 93.184.216.34
   00000000.00000000.00011111.11111111 ~255.255.224.0
   ----------------------------------------------------- &
   00000000.00000000.00011000.00100010 0.0.24.34
   ```

2. AND dst(Translated IP) with netmask:

```
00000001.00000010.00000011.00000100 1.2.3.4
11111111.11111111.11100000.00000000 255.255.224.0
--------------------------------------------------- &
00000001.00000010.00000000.00000000 1.2.0.0
```

3. Final step 2 bitwise-OR 3:

```
00000000.00000000.00011000.00100010 0.0.24.34
00000001.00000010.00000000.00000000 1.2.0.0
--------------------------------------------------- |
00000001.00000010.00011000.00100010 1.2.24.34
```

Applying DNS filter to FortiGate DNS server

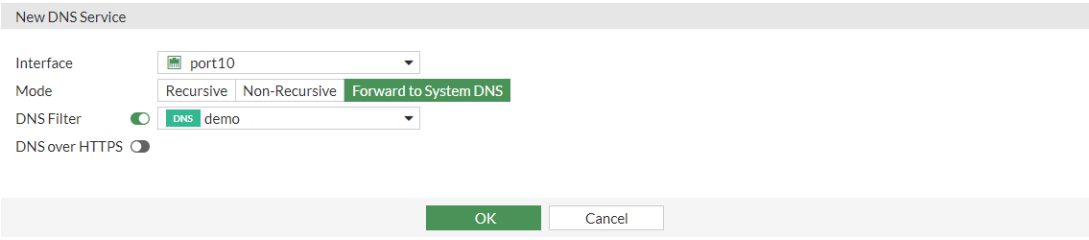
You can configure a FortiGate as a DNS server in your network. When you enable DNS service on a specific interface, the FortiGate will listen for DNS service on that interface.

Depending on the configuration, DNS service works in three modes: *Recursive*, *Non-Recursive*, or *Forward to System DNS* (server). For details on how to configure the FortiGate as a DNS server and configure the DNS database, see FortiGate DNS server on page 365.

You can apply a DNS filter profile to *Recursive* and *Forward to System DNS* mode. This is the same as the FortiGate working as a transparent DNS proxy for DNS relay traffic.

To configure DNS service in the GUI:

1. Go to *Network* > *DNS Servers* (if this option is not available, go to *System* > *Feature Visibility* and enable *DNS Database*).
2. In the *DNS Service on Interface* section, click *Create New* and select an *Interface* from the dropdown.
3. For *Mode*, select *Forward to System DNS*.
4. Enable *DNS Filter* and select a profile from the dropdown.



5. Click *OK*.

To configure DNS service in the CLI:

```
config system dns-server
    edit "port10"
        set mode forward-only
        set dnsfilter-profile "demo"
    next
end
```

To check DNS service with a DNS filter profile using a command line tool:

In this example, port10 is enabled as a DNS service with the DNS filter profile demo. The IP address of port10 is 10.1.100.5 , and the DNS filter profile is configured to block category 52 (information technology). From your internal network PC, use a command line tool, such as dig or nslookup, to perform a DNS query. For example:

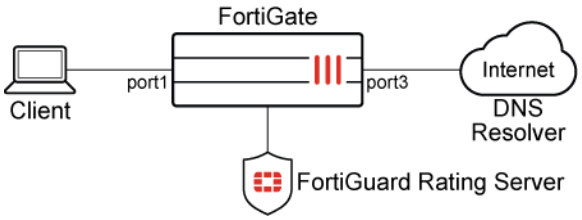
```
# dig @10.1.100.5 www.fortinet.com
;; ->>HEADER<<- opcode: QUERY; status: NOERROR; id: 52809
;; Flags: qr rd; QUERY: 1; ANSWER: 1; AUTHORITY: 0; ADDITIONAL: 0

;; QUESTION SECTION:
;; www.fortinet.com.            IN      A

;; ANSWER SECTION:
www.fortinet.com.       60      IN      A       208.91.112.55

;; Received 50 B
;; Time 2019-04-08 14:36:34 PDT
;; From 10.1.100.5@53(UDP) in 13.6 ms
```

The relay DNS traffic was filtered based on the DNS filter profile configuration. It was blocked and redirected to the portal IP (208.91.112.55).

DNS inspection with DoT and DoH

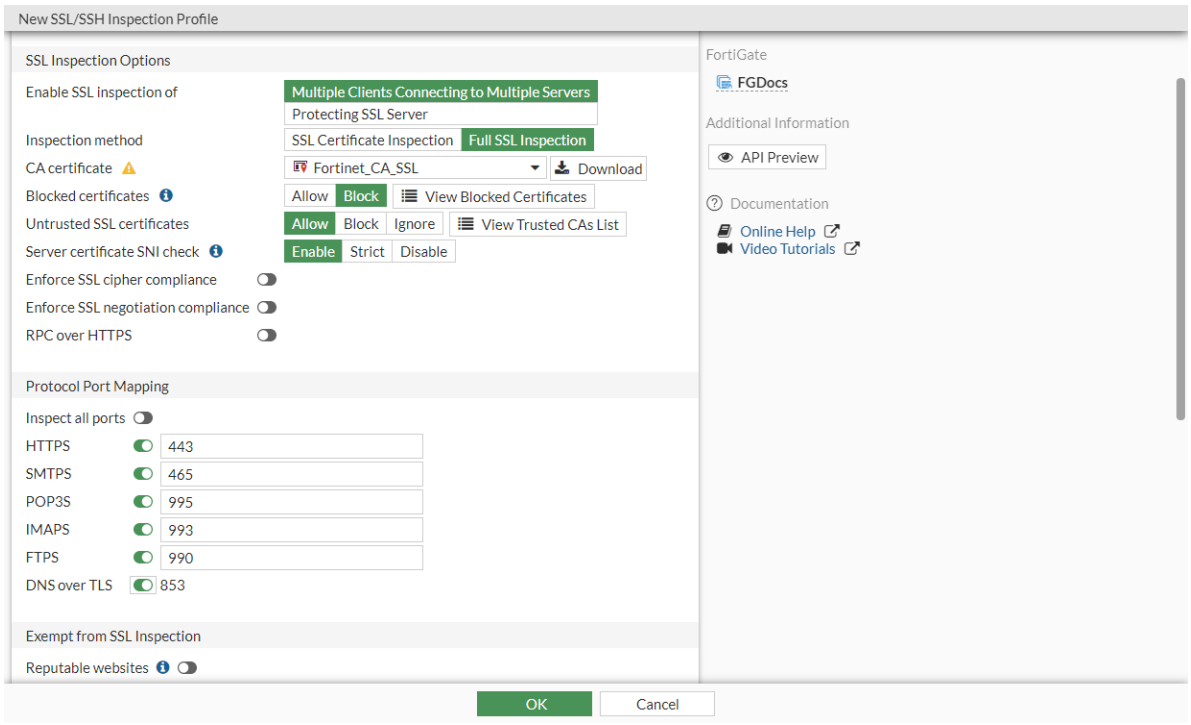
DNS over TLS (DoT) and DNS over HTTPS (DoH) are supported in DNS inspection. Prior to 7.6.0, DoT and DoH inspection can only be processed in proxy mode. Starting in 7.6.0, both proxy and flow inspection modes are able to handle DoT and DoH.

In the following examples, the FortiGate inspects DNS queries made over DoT and DoH to a Cloudflare DNS server. The DNS filter profile blocks the education category.



To configure DNS inspection of DoT and DoH queries in the GUI:

1. Configure the SSL-SSH profile:
   a. Go to *Security Profiles* > *SSL/SSH Inspection* and click *Create New*.
   b. Set *Inspection method* to *Full SSL Inspection*. DoT and DoH can only be inspected using doing deep inspection.
   c. In the *Protocol Port Mapping* section, enable *DNS over TLS*.



d. Configure the other settings as needed.

e. Click *OK*.

2. Configure the DNS filter profile:
   a. Go to *Security Profiles* > *DNS Filter* and click *Create New*.
   b. Enable *Redirect botnet C&C requests to Block Portal*.
   c. Enable *FortiGuard Category Based Filter* and set the *Action* for the *Education* category to *Redirect to Block Portal*.
   d. Configure the other settings as needed.
   e. Click *OK*.
3. Configure the firewall policy:
   a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
   b. Enable *DNS Filter* and select the profile you created.
   c. For *SSL Inspection*, select the profile you created.
   d. Configure the other settings as needed.
   e. Click *OK*.

To configure DNS inspection of DoT and DoH queries in the CLI:

1. Configure the SSL-SSH profile:

```
config firewall ssl-ssh-profile
    edit "ssl"
        config dot
            set status deep-inspection
            set client-certificate bypass
            set unsupported-ssl-cipher allow
```

```
            set unsupported-ssl-negotiation allow
            set expired-server-cert block
            set revoked-server-cert block
            set untrusted-server-cert allow
            set cert-validation-timeout allow
            set cert-validation-failure block
        end
    next
end
```

2. Configure the DNS filter profile:

```
config dnsfilter profile
    edit "dnsfilter"
        config ftgd-dns
            config filters
                edit 1
                    set category 30
                    set action block
                next
            end
        end
        set block-botnet enable
    next
end
```

3. Configure the firewall policy:

```
config firewall policy
    edit 1
        set srcintf "port1"
        set dstintf "port3"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set profile-protocol-options "protocol"
        set ssl-ssh-profile "ssl"
        set webfilter-profile "webfilter"
        set dnsfilter-profile "dnsfilter"
        set nat enable
    next
end
```

Testing the connection

To query DNS over TLS:

1. Send a DNS query over TLS to the Cloudflare server 1.1.1.1 (this example uses kdig on an Ubuntu client). The www.ubc.ca domain belongs to the education category:

```
~$ kdig -d @1.1.1.1 +tls-ca +tls-host=cloudflare-dns.com www.ubc.ca
;; DEBUG: Querying for owner(www.ubc.ca.), class(1), type(1), server(1.1.1.1), port(853),
protocol(TCP)
;; DEBUG: TLS, imported 128 system certificates
;; DEBUG: TLS, received certificate hierarchy:
;; DEBUG:  #1, C=US,ST=California,L=San Francisco,O=Cloudflare\, Inc.,CN=cloudflare-dns.com
;; DEBUG:      SHA-256 PIN: elpYCnCs9ZtkQBI4+cb2QtZcyOl5UI9jMkSvbTsTad0=
;; DEBUG:  #2, C=US,ST=California,L=Sunnyvale,O=Fortinet,OU=Certificate
Authority,CN=FG3H1E5818903681,EMAIL=support@fortinet.com
;; DEBUG:      SHA-256 PIN: s48VtdODlNZfAG2g/92hMLhitU51qsP9pkHAUtTJ+f4=
;; DEBUG: TLS, skipping certificate PIN check
;; DEBUG: TLS, The certificate is trusted.
;; TLS session (TLS1.3)-(ECDHE-SECP256R1)-(ECDSA-SECP256R1-SHA256)-(AES-256-GCM)
;; ->>HEADER<<- opcode: QUERY; status: NOERROR; id: 56850
;; Flags: qr rd; QUERY: 1; ANSWER: 1; AUTHORITY: 0; ADDITIONAL: 0

;; QUESTION SECTION:
;; www.ubc.ca.                  IN      A

;; ANSWER SECTION:
www.ubc.ca.             60      IN      A       208.91.112.55

;; Received 44 B
;; Time 2021-03-12 06:53:37 UTC
;; From 1.1.1.1@853(TCP) in 6.0 ms
```

   In this query, the FortiGate inspects the DNS query to the Cloudflare DNS server. It replaces the result with the IP of the FortiGuard block page, which successfully blocks the query.

To query DNS over HTTPS:

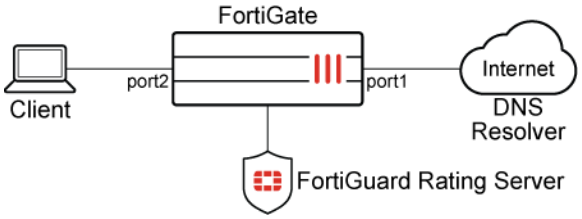
1. In your browser, enable DNS over HTTPS.
2. Go to www.ubc.ca. The website is redirected to the block page.

DNS over QUIC and DNS over HTTP3 for transparent and local-in DNS modes

DNS over QUIC (DoQ) and DNS over HTTP3 (DoH3) are supported in proxy mode inspection for transparent and local-in explicit modes. With DoQ and DoH3, connections can be established faster than with DNS over TLS (DoT) or DNS over HTTPS (DoH). The FortiGate can also handle the QUIC/TLS handshake and perform deep inspection for HTTP3 and QUIC traffic. This allows for faster and more secure DNS resolution, with improved privacy and reduced latency.

In transparent mode, the FortiGate is acting as a proxy, forwarding DNS queries, and not as a DNS server. In local-in DNS mode, the FortiGate acts as the DNS server and a DNS filter profile is applied in the system DNS server.

The firewall policy must be in proxy mode.

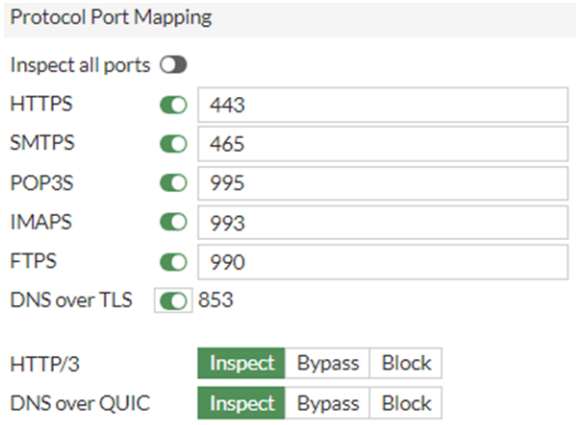


DoQ transparent and local-in query can be achieved using tools or applications in Linux, such as the q tiny command line DNS client from Natesales.

DoH3 transparent and local-in query can be achieved in Linux using q or Curl. In Windows, change the client network DNS server to the FortiGate and treat the FortiGate as an HTTP3 DNS server listening for DoH3 connections.

To enable QUIC in SSL/SSH inspection profiles in the GUI:

1. Go to *Security Profiles* > *SSL/SSH Inspection* and click *Create New*.
2. Under *Protocol Port Mapping*, set *HTTP/3* and *DNS over QUIC* to *Inspect*.



3. Configure the remaining settings as required.
4. Click *OK*.

To configure DoQ in transparent mode in the CLI:

1. Enable QUIC in the ssl-ssh-profile:

```
config firewall ssl-ssh-profile
    edit "protocols"
        config dot
            set status deep-inspection
            set quic inspect
        end
    next
end
```

2. Configure a DNS filter profile:

```
config dnsfilter profile
    edit "dnsfilter_fgd"
        config ftgd-dns
            config filters
                edit 1
                    set category 30
                    set action block
                next
            end
        end
    next
end
```

3. Apply the profiles to a proxy firewall policy:

```
config firewall policy
    edit 1
        set name "dnsfilter"
        set srcintf "port2"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set profile-protocol-options "protocol"
        set ssl-ssh-profile "protocols"
        set dnsfilter-profile "dnsfilter_fgd"
        set logtraffic all
        set nat enable
    next
end
```

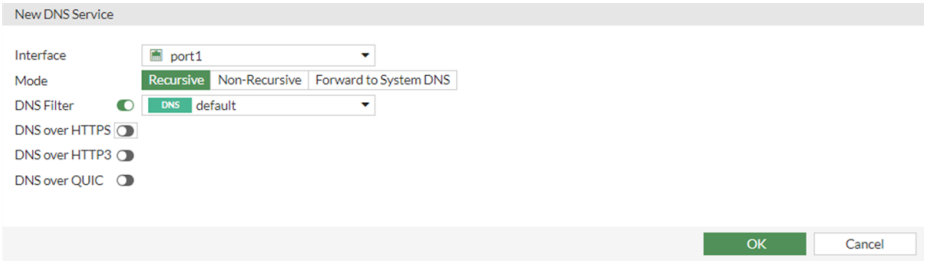
4. Test the configuration:

   On the client, use q to query a FortiGuard category30 domain with the Adguard DNS server over QUIC. The default redirect block IP address should be returned:

```
pc03:~# q www.sfu.ca @quic://dns.adguard.com --tls-no-verify
2023/08/18 18:53:44 failed to sufficiently increase receive buffer size (was: 208 kiB, wanted:
2048 kiB, got: 416 kiB). See https://github.com/quic-go/quic-go/wiki/UDP-Receive-Buffer-Size
for details.
www.sfu.ca. 1m0s A 208.91.112.55
www.sfu.ca. 1m0s AAAA 2620:101:9000:53::55
```

To enable DNS over HTTP3 or DNS over QUIC on the DNS server in the GUI:

1. Go to *Network* > *DNS Servers*.
   If this option is not available, go to *System* > *Feature Visibility* and enable *DNS Database*.
2. In the *DNS Service on Interface* section, edit an existing interface, or create a new one.
3. Set the *Mode* option as desired.
4. Enable *DNS Filter* and select a profile.
5. Enable *DNS over HTTP3* or *DNS over QUIC*.



6. Click *OK*.

To configure DoQ in local-in mode in the CLI:

1. In the FortiGate DNS server configuration, enable DoQ for a port with the previously configured DNS filter profile applied:

```
config system dns-server
    edit "port2"
        set dnsfilter-profile "dnsfilter_fgd"
        set doq enable
    next
end
```

2. Test the configuration:
   On the client, use q to query a FortiGuard category30 domain with the FortiGate interface over QUIC. The default redirect block IP address should be returned:

```
pc03:~# q www.mcgill.ca @quic://10.1.100.150 --tls-no-verify
2023/08/18 20:05:53 failed to sufficiently increase receive buffer size (was: 208 kiB, wanted:
2048 kiB, got: 416 kiB). See https://github.com/quic-go/quic-go/wiki/UDP-Receive-Buffer-Size
for details.
www.mcgill.ca. 1m0s A 208.91.112.55
www.mcgill.ca. 1m0s AAAA 2620:101:9000:53::55
```

To configure DoH3 in transparent mode in the CLI:

1. Enable QUIC in the ssl-ssh-profile:

```
config firewall ssl-ssh-profile
    edit "protocols"
        config https
            set ports 443 8443
            set status deep-inspection
            set quic inspect
        end
    next
end
```

2. Configure a DNS filter profile:

```
config dnsfilter profile
    edit "dnsfilter_fgd"
        config ftgd-dns
            config filters
                edit 1
                    set category 30
                    set action block
                next
            end
        end
    next
end
```

3. Apply the profiles to a proxy firewall policy:

```
config firewall policy
    edit 1
        set name "dnsfilter"
        set srcintf "port2"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set profile-protocol-options "protocol"
        set ssl-ssh-profile "protocols"
        set dnsfilter-profile "dnsfilter_fgd"
        set logtraffic all
        set nat enable
    next
end
```

4. Test the configuration:

On the client with HTTP3 support, use q or Curl to query a FortiGuard category30 domain with the Adguard DNS server or Cloudflare DNS server over QUIC. The default redirect block IP address should be returned:

```
pc03:~# q www.mcgill.ca --http3 @https://dns.adguard.com --tls-no-verify
2023/08/18 21:04:02 failed to sufficiently increase receive buffer size (was: 208 kiB, wanted:
2048 kiB, got: 416 kiB). See https://github.com/quic-go/quic-go/wiki/UDP-Receive-Buffer-Size
for details.
www.mcgill.ca. 1m0s A 208.91.112.55
www.mcgill.ca. 1m0s AAAA 2620:101:9000:53::55
```

```
pc03:~# curl -H 'accept: application/dns-message' -v -k --http3 'https://1.1.1.1/dns-
query?dns=q80BAAABAAAAAAAAA3d3dwN1YmMCY2EAAAEAAQ' | hexdump
*   Trying 1.1.1.1:443...
* Connect socket 5 over QUIC to 1.1.1.1:443
* Sent QUIC client Initial, ALPN: h3,h3-29,h3-28,h3-27
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0* Connected to
1.1.1.1 (1.1.1.1) port 443 (#0)
* h3 [:method: GET]
* h3 [:path: /dns-query?dns=q80BAAABAAAAAAAAA3d3dwN1YmMCY2EAAAEAAQ]
* h3 [:scheme: https]
* h3 [:authority: 1.1.1.1]
* h3 [user-agent: curl/7.80.0-DEV]
* h3 [accept: application/dns-message]
* Using HTTP/3 Stream ID: 0 (easy handle 0x558fdd1c2220)
> GET /dns-query?dns=q80BAAABAAAAAAAAA3d3dwN1YmMCY2EAAAEAAQ HTTP/3
> Host: 1.1.1.1
> user-agent: curl/7.80.0-DEV
> accept: application/dns-message
>
< HTTP/3 200
< content-type: application/dns-message
< content-length: 44
<
{ [44 bytes data]
100    44  100    44    0     0   1305      0 --:--:-- --:--:-- --:--:--  1375
* Connection #0 to host 1.1.1.1 left intact
0000000 cdab 0081 0100 0100 0000 0000 7703 7777
0000010 7503 6362 6302 0061 0100 0100 0cc0 0100
0000020 0100 0000 3c00 0400 5bd0 3770
000002c
```

To configure DoH3 in local-in mode in the CLI:

1. In the FortiGate DNS server configuration, enable DoH3 for a port with the previously configured DNS filter profile applied:

```
config system dns-server
    edit "port2"
        set dnsfilter-profile "dnsfilter_fgd"
        set doh3 enable
```

```
    next
end
```

2. Test the configuration:

   On the client with HTTP3 support, use q or Curl to query a FortiGuard category30 domain with the FortiGate interface over HTTP3. The default redirect block IP address should be returned:

```
pc03:~# q www.mcgill.ca --http3 @https://10.1.100.150 --tls-no-verify
2023/08/18 20:37:55 failed to sufficiently increase receive buffer size (was: 208 kiB, wanted:
2048 kiB, got: 416 kiB). See https://github.com/quic-go/quic-go/wiki/UDP-Receive-Buffer-Size
for details.
www.mcgill.ca. 1m0s A 208.91.112.55
www.mcgill.ca. 1m0s AAAA 2620:101:9000:53::55
```

```
pc03:~# curl -H 'accept: application/dns-message' -v -k --http3 'https://10.1.100.150/dns-
query?dns=q80BAAABAAAAAAAAA3d3dwN1YmMCY2EAAAEAAQ' | hexdump
*   Trying 10.1.100.150:443...
* Connect socket 5 over QUIC to 10.1.100.150:443
* Sent QUIC client Initial, ALPN: h3,h3-29,h3-28,h3-27
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:-- --:--:--     0* Connected to
10.1.100.150 (10.1.100.150) port 443 (#0)
* h3 [:method: GET]
* h3 [:path: /dns-query?dns=q80BAAABAAAAAAAAA3d3dwN1YmMCY2EAAAEAAQ]
* h3 [:scheme: https]
* h3 [:authority: 10.1.100.150]
* h3 [user-agent: curl/7.80.0-DEV]
* h3 [accept: application/dns-message]
* Using HTTP/3 Stream ID: 0 (easy handle 0x55ced8274250)
> GET /dns-query?dns=q80BAAABAAAAAAAAA3d3dwN1YmMCY2EAAAEAAQ HTTP/3
> Host: 10.1.100.150
> user-agent: curl/7.80.0-DEV
> accept: application/dns-message
>
< HTTP/3 200
< content-type: application/dns-message
< content-length: 44
<
{ [44 bytes data]
100    44  100    44    0     0   1893      0 --:--:-- --:--:-- --:--:--  2000
* Connection #0 to host 10.1.100.150 left intact
0000000 cdab 0081 0100 0100 0000 0000 7703 7777
0000010 7503 6362 6302 0061 0100 0100 0cc0 0100
0000020 0100 0000 3c00 0400 5bd0 3770
000002c
```

Troubleshooting for DNS filter

If you have trouble with the DNS filter profile in your policy, start with the following troubleshooting steps:

- Check the connection between the FortiGate and FortiGuard DNS rating server (SDNS server).
- Check that the FortiGate has a valid FortiGuard web filter license.
- Check the FortiGate DNS filter configuration.

Checking the connection between the FortiGate and FortiGuard SDNS server

You need to ensure the FortiGate can connect to the FortiGuard SDNS server. By default, the FortiGate uses DNS over TLS (DoT, TCP port 853) to connect to the SDNS server. See DNS over TLS and HTTPS on page 392 for more information.

To check the connection between the FortiGate and SDNS server:

1. Verify the FortiGuard SDNS server information:

```
# diagnose test application dnsproxy 3
...
SDNS servers:
173.243.140.53:853 vrf=0 tz=-480 encrypt=dot req=0 to=0 res=0 rt=34 ready=1 timer=0 probe=0
failure=0 last_failed=0
```

   The SDNS server IP address might be different depending on location (in this example, it is 173.243.140.53:853).
2. In the management VDOM, check the communication between the FortiGate and the SDNS server:

```
# execute ping 173.243.140.53
```

3. If FortiGuard is not reachable using anycast, configure the default FortiGuard SDNS (unicast) server (208.91.112.220):

```
config system fortiguard
    set fortiguard-anycast disable
    set sdns-server-ip "208.91.112.220"
end
```

4. Verify the list of SDNS servers again:

```
# diagnose test application dnsproxy 3
FGD_DNS_SERVICE_LICENSE:
server=208.91.112.220:53, expiry=2023-10-28, expired=0, type=2
server=83.231.212.53:53, expiry=2023-10-28, expired=0, type=2
```

   The default FortiGuard SDNS server should work in most cases; however, you can switch to another server to see if it improves latency.

By default, DNS filtering connects to the FortiGuard secure DNS server over anycast and uses DoT (TCP port 853) when the default settings of `fortiguard-anycast enable` and `fortiguard-anycast-source fortinet` are configured. Disabling `fortiguard-anycast` will force the FortiGate to use cleartext (UDP port 53) instead of DoT (TCP port 853) in addition to disabling FortiGuard secure DNS over anycast.

Checking the FortiGuard DNS rating service license

The FortiGuard DNS rating service shares the license with the FortiGuard web filter, so you must have a valid web filter license for the DNS rating service to work. While the license is shared, the DNS rating service uses a separate connection mechanism from the web filter rating.

To check the DNS rating service license in the CLI:

1. View the DNS settings:

```
# diagnose test application dnsproxy 3
```

2. Find the `FGD_DNS_SERVICE_LICENSE` line and check that the license has not expired:

```
FGD_DNS_SERVICE_LICENSE:
server=173.243.140.53:853, expiry=2023-10-28, expired=0, type=2
```

3. Find the `SDNS servers` line to view the functioning servers:

```
SDNS servers:
173.243.140.53:853 vrf=0 tz=-480 encrypt=dot req=0 to=0 res=0 rt=34 ready=1 timer=0 probe=0
failure=0 last_failed=0
```

Checking the FortiGate DNS filter profile configuration

To check the DNS filter profile configuration:

1. In FortiOS, create a local domain filter and set the *Action* to *Redirect to Block Portal* (see Local domain filter on page 2402).
2. Apply this DNS filter profile to the policy.
3. From the client PC, perform a DNS query on this domain. If you get the profile's redirected portal address, this means that the DNS filter profile works as expected.

Additional troubleshooting

Use `diagnose test application dnsproxy <test level>` to troubleshoot further DNS proxy information, where:

Test level	Action
1	Clear DNS cache
2	Show statistics
3	Dump DNS setting
4	Reload FQDN
5	Requery FQDN
6	Dump FQDN
7	Dump DNS cache
8	Dump DNS database
9	Reload DNS database
10	Dump secure DNS policy/profile
11	Dump botnet domain
12	Reload secure DNS setting
13	Show hostname cache
14	Clear hostname cache
15	Show SDNS rating cache
16	Clear SDNS rating cache
17	Show DNS debug bit mask
18	Show DNS debug object members
99	Restart the dnsproxy worker

To debug DNS proxy details:

```
# diagnose debug application dnsproxy -1
# diagnose debug {enable | disable}
```

Telemetry profile

This topic applies only to a FortiGate acting as a FortiTelemetry Controller with *FortiTelemetry* enabled on the *System > Feature Visibility* page. For a summary of FortiTelemetry-related tasks that you can do in FortiOS, see FortiTelemetry on page 139. See also *FortiTelemetry Administration Guide*.

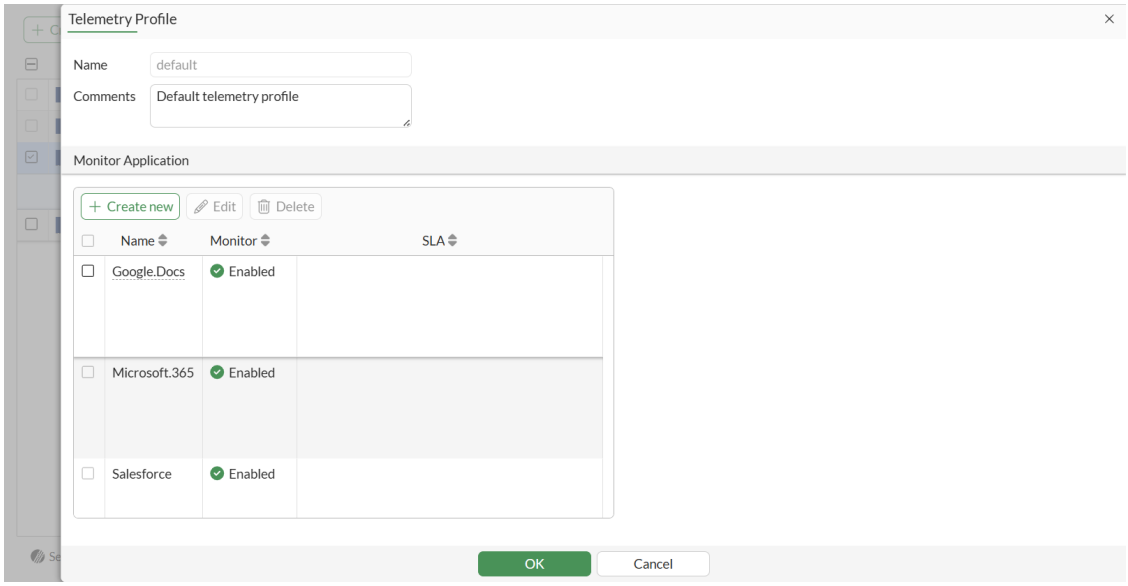
Telemetry profiles are used by FortiTelemetry agent(s). Telemetry profiles instruct FortiTelemetry agent(s) what predefined applications to monitor.

A telemetry profile must be configured in a firewall policy to create a telemetry policy. See Telemetry policy on page 1909 for more information.

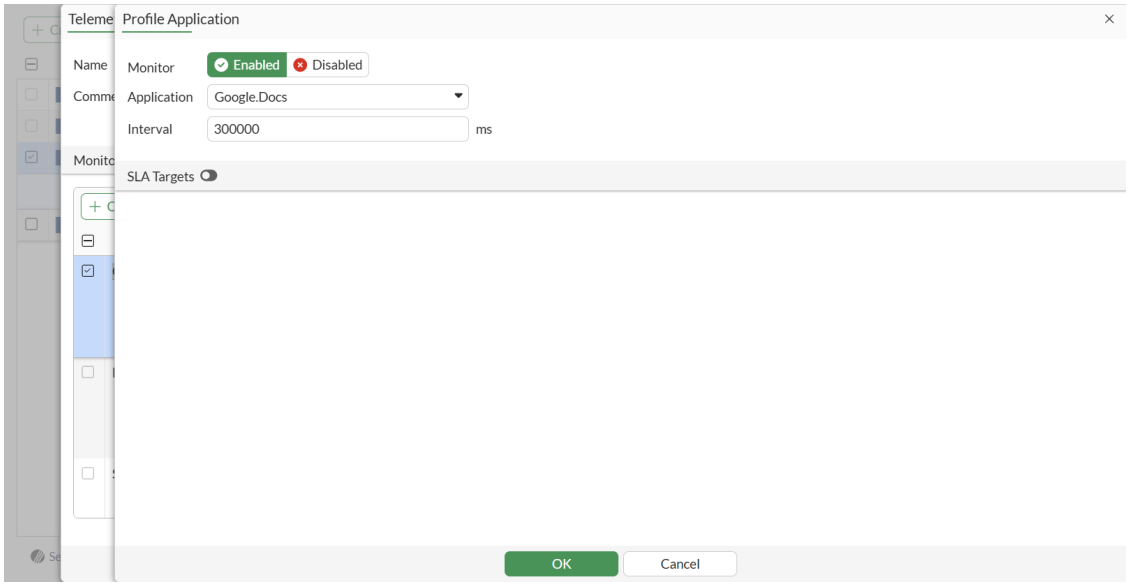
To view Telemetry profiles:

1. Go to *Security Profiles* > *Telemetry Profile*. The list of telemetry profiles is displayed, including a *default* telemetry profile.
2. Select a profile to access additional buttons, such as *Edit*, *Clone*, *Delete*, and *More*.
3. Select the *default* profile, and click *Edit*. The *Telemetry Profile* pane opens.

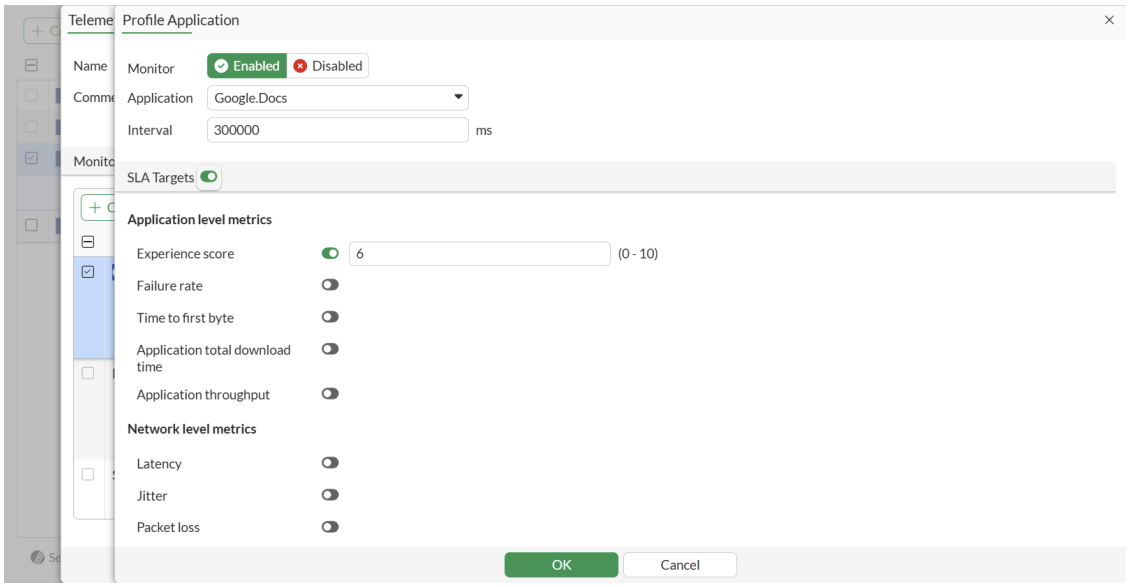
   The default profile monitors the following predefined applications: Google docs, Microsoft 365, and Salesforce. No SLA targets are defined.



4. Select an application, and click *Edit*. The *Profile Application* pane opens.



5. Enable *SLA Targets* to view what data will be collected. Although you can enable and set SLA targets, the functionality does not work yet



6. Click *OK* to save profile application settings.
7. Click *OK* to save telemetry profile settings.

Predefined applications

FortiOS 8.0.0 includes the following SaaS applications that FortiTelemetry agents can monitor:

- Adobe
- Atlassian Cloud
- Dropbox
- Elastic Search
- Google Docs
- Google Drive
- Google Maps
- Go To Meeting
- Microsoft 365
- Microsoft SharePoint
- Microsoft Teams
- Sales Force
- Slack
- Twilio
- Webex
- Yahoo
- Zendesk
- Zoom

On the FortiTelemetry Controller, use the `config telemetry-controller application predefine` command to view the list.

Creating telemetry profiles

This topic applies only to a FortiGate acting as a FortiTelemetry Controller with *FortiTelemetry* enabled on the *System > Feature Visibility* page. For a summary of FortiTelemetry-related tasks that you can do in FortiOS, see FortiTelemetry on page 139. See also *FortiTelemetry Administration Guide*.

FortiOS includes a *default* telemetry security profile. You can edit or clone the profile, or you can create a new profile.

Although you can enable SLA targets, the functionality does not work yet.

To edit the default telemetry profile in the GUI:

1. Go to *Security Profiles* > *Telemetry Profile*.
2. Select the default profile, and click *Edit*. The profile opens for editing.
3. Enter your changes, and click *OK* to save the changes.

To clone a telemetry profile in the GUI:

1. Go to *Security Profiles* > *Telemetry Profile*.
2. Select a profile, and click *Clone*. The profile opens for editing.
3. Set *Name* to a unique name.
4. Enter your changes, and click *OK* to save the new profile.

To create a telemetry profile in the GUI:

1. Go to *Security Profiles* > *Telemetry Profile*, and click *Create New*.
2. Enter a name and optional comment for the profile.
3. In the *Monitored Applications* section, add to the profile the applications to monitor:
   a. Click *Create New*.
   b. Choose the *Monitor* status for the application:
      - Choose *Enabled* to monitor the application.
      - Choose *Disabled* to disable monitoring of the application.
   c. From the *Applications* dropdown, select a predefined application to monitor. See Predefined applications on page 2426 for more information.
   d. Set the monitoring *Interval* in milliseconds.
   e. Enable *SLA Targets* to view what data will be collected. Although you can enable and set SLA targets, the functionality does not work yet.

Application level metrics:	
Experience score	Overall application experience score threshold between 0 and 10. Experience scores are categorized as follows: • Good: 8 to 10 • OK: 5 to 8 • Poor: 3 to 5 • Broken: 0 to 3
Failure rate	Failure rate threshold for monitoring HTTP requests as a percentage (%).
Time to first byte	Time to first byte threshold monitor requests in milliseconds (ms).
Application total download time	Application total download time threshold for monitoring HTTP requests in milliseconds (ms).
Application throughput	The throughput threshold for monitoring HTTP requests in megabytes (MBps).
Network level metrics:	

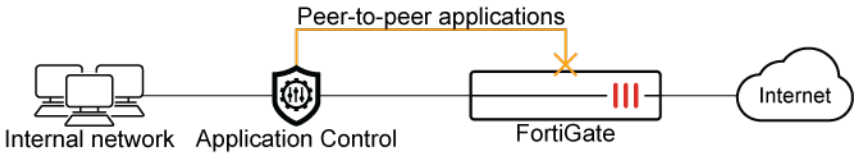
Latency	Average latency threshold for network probes in milliseconds (ms).
Jitter	Jitter threshold for network probes in milliseconds (ms).
Packet loss	Packet-loss threshold for network probes as a percentage.
TCP round trip time	TCP round trip time threshold for monitoring HTTP requests in milliseconds (ms).
95% DNS resolving time	DNS resolving time threshold for monitoring HTTP requests in milliseconds.
95% TLS handshake time	TLS time threshold in milliseconds.

f. Click *OK* to save the monitored application.
g. Repeat the steps to include additional applications as needed in the profile.

4. Click *OK* to save the telemetry profile.

Application control

FortiGates can recognize network traffic generated by a large number of applications. Application control sensors specify what action to take with the application traffic. Application control uses IPS protocol decoders that can analyze network traffic to detect application traffic, even if the traffic uses non-standard ports or protocols. Application control supports traffic detection using the HTTP protocol (versions 1.0, 1.1, and 2.0).



FortiOS includes three preloaded application sensors:

- *default* (monitors all applications)
- *wifi-default* (default configuration for offloading WiFi traffic)
- *block-high-risk*

You can customize these sensors, or you can create your own to log and manage the applications on your network.

Once configured, you can add the application sensor to a firewall policy.

This functionality requires a subscription to FortiGuard Application Control.

The following topics provide information about application control:

- Configuring an application sensor on page 2429
- Application matching signature priority on page 2430
- Basic category filters and overrides on page 2432
- Excluding signatures in application control profiles on page 2437

- Port enforcement check on page 2439
- Protocol enforcement on page 2440
- SSL-based application detection over decrypted traffic in a sandwich topology on page 2442
- Matching multiple parameters on application control signatures on page 2443
- Application signature dissector for DNP3 on page 2446
- Application control on page 2428

Configuring an application sensor

FortiGates can recognize network traffic generated by a large number of applications using application control, which relies on IPS protocol decoders. Application sensors control what action is taken with application traffic.

To configure an application sensor:

1. Go to *Security Profiles* > *Application Control* and click *Create New*.
2. Configure the following settings:

Name	Enter a unique name for the sensor.
Comments	Enter a comment (optional).
Categories	Configure the action to take on groups of signatures based on their category type. Applications belonging to the category trigger the configured action: monitor, allow, block, or quarantine. See Basic category filters and overrides on page 2432 for more information.
Network Protocol Enforcement	Enable/disable the enforcement of protocols over selected ports. See Protocol enforcement on page 2440 for more information.
Application and Filter Overrides	Configure multiple applications signatures with a dedicated action for a single sensor. Filters can be added based on the application category, behavior, popularity, protocol, risk, technology, or vendor subtype. For more information, see • Configuring application and filter overrides on page 2434 • Matching multiple parameters on application control signatures on page 2443 • Blocking applications with custom signatures on page 2726
Block applications detected on non-default ports	When enabled: • For monitor and allow actions, applications will be blocked if detected on non-default ports (as defined in FortiGuard application signatures). • Block actions still block traffic for the application regardless of the port. See Port enforcement check on page 2439 for more information.

Allow and Log DNS Traffic	The intended behavior is to allow and log DNS traffic. However, to fully enable logging, configure the following: 1. Within the application sensor profile, create a new *Application and Filter Overrides* entry. 2. Set the *Action* to *Monitor*. 3. Search for the DNS application, select it, and click *Add Selected*. 4. Click *OK* to save the override, then click *OK* to save the application sensor profile.
Replacement Messages for HTTP-based Applications	Enable/disable replacement messages for blocked applications. See Replacement messages on page 3899 for information about replacement messages.

3. Click *OK*.

Application matching signature priority

Many applications will match more than one application control signature. When attempting to match a signature to the application traffic, FortiGate evaluates several factors and selects a signature based on the following tie breakers:

1. Segments are considered in sequential order. The order of the applications within the segment does not matter.

   Segments are groupings of applications and protocols that are all evaluated at the same time for the best match. The default segment is defined by the Category section. Additional segments may be defined using the Application and Filter Overrides section. In the CLI these segments are called entries.
2. Custom signatures take precedence over FortiGuard signatures.
3. The signature with the most pre/post-match signature actions is preferred.

   For example,one signature could have a lot of the --deep_ctrl option (used for pattern matching) and each one would increment the post-match counter. Similarly, options like --quiet are considered a pre-match action that would suppress logging of the match. See Creating IPS and application control signatures for more attributes.
4. The application weight.

   For predefined signatures, the weight is defined by FortiGuard signature analysts and is not configurable. Generally, more specific application signatures will have a higher weight than more broad protocol signatures. Not all application signature types (such as protocols) have the same weight.
5. Signature visibility options. Non-hidden signatures are preferred.

   A signature could have *hidden* visibility if it is a beta or test signature that is still under false-positive detection evaluation, marked --quiet, or a built-in protocol dissector/peer-to-peer rule.
6. Pattern counts.

   A comparison between signature --pattern and --pcre count. The signature with more of them is preferred.
7. ID comparison.

   A comparison between signature IDs. Select the one with a larger, and therefore most likely newer, ID.

Application weight is used the most often to decide the match. To see the weight of applications and protocols, use the `get application name status` command.

The following examples use grep to find specific applications by name. The -A flag is used to include a specific number of lines after the match is found.

```
# get application name status | grep -A10 "app-name: \"Facebook\""
app-name: "Facebook"
id: 15832
category: "Social.Media"
cat-id: 23
popularity: 5.low
risk: 3.low
weight: 10
shaping: 0
protocol: 1.TCP, 9.HTTP, 2.UDP, 26.SSL
vendor: 3.Meta
technology: 1.Browser-Based
```

```
# get application name status | grep -A10 "app-name: \"SSL\""
app-name: "SSL"
id: 15895
category: "Network.Service"
cat-id: 15
popularity: 5.low
risk: 2.high
weight: 1
shaping: 0
protocol: 1.TCP, 26.SSL
vendor: 0.Other
technology: 0.Network-Protocol
```

Finding the applications belonging to an application category

When you have an application category and you want to find the matching applications, use the following command to list the applications filtered by the category name:

```
# get application name status | grep 'category: "Email"' -B2
app-name: "1und1.Mail"
id: 29025
category: "Email"
--
app-name: "126.Mail"
id: 16554
category: "Email"
--
app-name: "AIM.Webmail"
id: 15819
category: "Email"
--
....
```

If you only have the application category ID, you can use the following command to list the applications:

```
# get application name status | grep 'cat-id: 21' -B3
app-name: "1und1.Mail"
id: 29025
category: "Email"
cat-id: 21
--
app-name: "126.Mail"
id: 16554
category: "Email"
cat-id: 21
--
app-name: "AIM.Webmail"
id: 15819
category: "Email"
cat-id: 21
--
...
```

Basic category filters and overrides

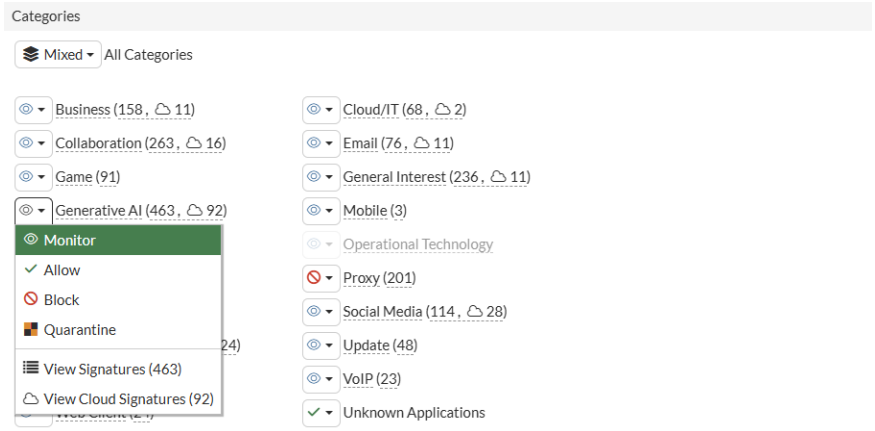
When creating an application sensor, you can define the applications that you want to control. You can add applications and filters using categories, application overrides, and/or filter overrides with designated actions (monitor, allow, block, or quarantine).

Action	Description
Monitor	Passes the traffic and generates a log message.
Allow	Passes the traffic but does not generate a log message.
Block	Drops the detected traffic and generates a log message.
Quarantine	Blocks the traffic from an attacker IP address until the expiration time is reached and generates a log message.

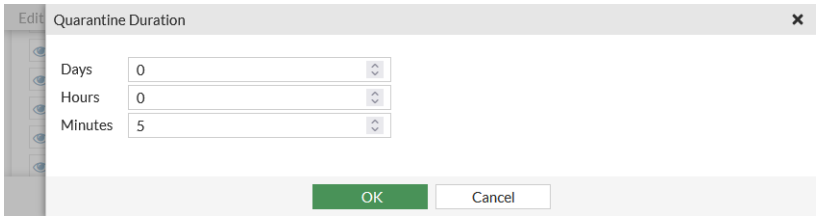
For more information about application control logs, see Security Events log page on page 4760.

To configure category filters in the GUI:

1. Go to *Security Profiles* > *Application Control* and click *Create New*, or edit an existing sensor.
2. Under *Categories*, click the icon next to the category name to set the action or view the application signatures.



3. If you select the *Quarantine* action, the *Quarantine Duration* pane will open. Enter the duration values and click *OK*.



4. Click *OK*.

To configure category filters in the CLI:

```
config application list
    edit <name>
        config entries
            edit <id>
                set category <id>
                set action {pass | block | reset}
                set quarantine {none | attacker}
                set quarantine-expiry <###d##h###m>
                set log {enable | disable}
            next
        end
    next
end
```

Command	Description
`category <id>`	Set the category by category ID. To view the category list by ID, enter `set category ?` • all (All) • 2 (P2P) • 3 (VoIP) • 5 (Video/Audio) • 6 (Proxy)

Command	Description
	• 7 (Remote.Access) • 8 (Game) • 12 (General.Interest) • 15 (Network.Service) • 17 (Update) • 21 (Email) • 22 (Storage.Backup) • 23 (Social.Media) • 25 (Web.Client) • 26 (Industrial) • 28 (Collaboration) • 29 (Business) • 30 (Cloud.IT) • 31 (Mobile) • 32 (Unknown Applications) • 36 (GenAI) Application categories may differ between different versions and models.
`action {pass | block | reset}`	Pass/block traffic or reset the connection for traffic from this application (default = block).
`quarantine {none | attacker}`	Set the quarantine method: • `none`: Quarantine is disabled. • `attacker`: Block all traffic sent from attacker's IP address. The attacker's IP address is also added to the banned user list. The target's address is not affected.
`quarantine-expiry <###d##h##m>`	Duration of quarantine, in minutes (1 - 364d23h59, default = 5). Requires `quarantine` set to `attacker`.
`log {enable | disable}`	Enable/disable logging for this application list (default = enable).

Configuring application and filter overrides

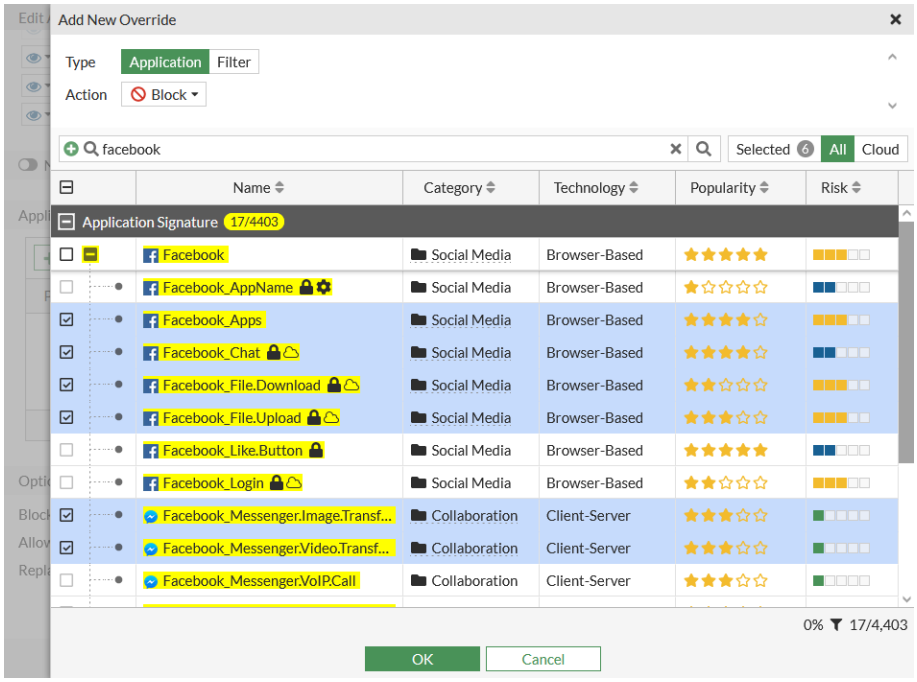
Multiple application signatures can be added for one sensor with a designated action. Filters can be added based on behavior, application category, popularity, protocol, risk, technology, or vendor subtypes.

To configure overrides in the GUI:

1. Go to *Security Profiles* > *Application Control* and click *Create New*, or edit an existing sensor.
2. Add an application:
   a. In the *Application and Filter Overrides* table, click *Create New*.
   b. For *Type*, select *Application*.
   c. Select an *Action* from the dropdown.

d. In the *Search* box, enter an application name and press `Enter`, or click the plus icon and configure a search filter.

e. In the search results, select the required applications (you can select multiple applications).



f. Click *OK*.

3. Add a filter:

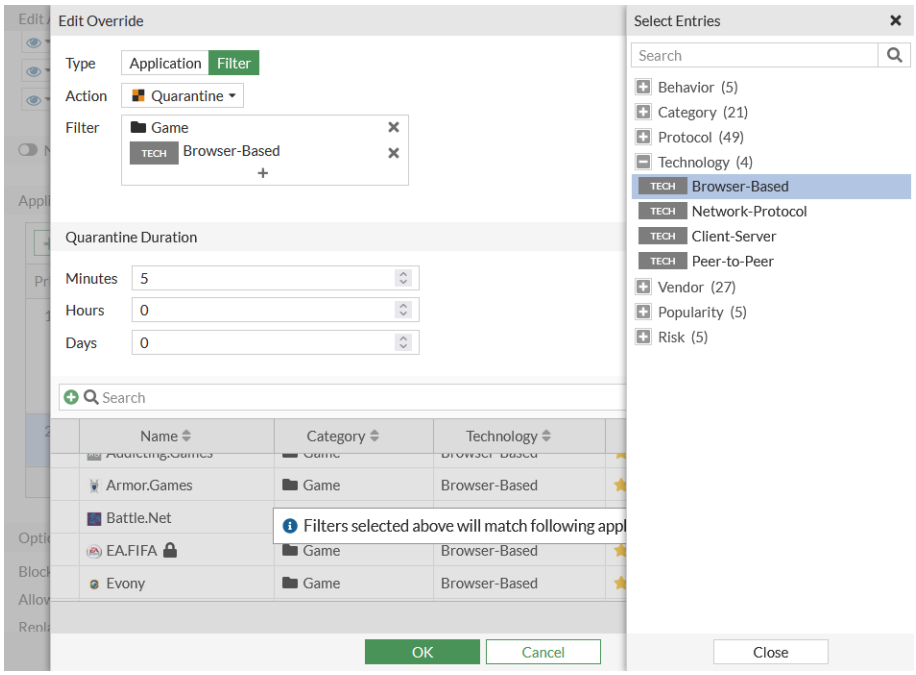
a. In the *Application and Filter Overrides* table, click *Create New*.

b. For *Type*, select *Filter*.

c. Select an *Action* from the dropdown.

If the action is set to *Quarantine*, set the duration of the quarantine.

d. In the *Filter* field, click the *+* . The *Select Entries* pane opens, and you can search based on filter subtypes. This example uses Browser-Based (under Technology) and Game (under Category).



e. Click *OK*.

4. Click *OK*.

To configure overrides in the CLI:

```
config application list
    edit <name>
        config entries
            edit <id>
                set protocols <integer>
                set risk <integer>
                set vendor <id>
                set technology <id>
                set behavior <id>
                set popularity <integer>
                set action {pass | block | reset}
                set quarantine {none | attacker}
                set log {enable | disable}
            next
        end
    next
end
```

Command	Description
`protocols <integer>`	Application protocol filter (0 - 54, or all).
`risk <integer>`	Risk or impact of allowing traffic from this application to occur (1 - 5; low (1), elevated (2), medium (3), high (4), and critical (5)).

Command	Description
`vendor <id>`	Application vendor filter (0 - 25, or all).
`technology <id>`	Application technology filter: • all • 0 (network-protocol) • 1 (browser-based) • 2 (client-server) • 4 (peer-to-peer)
`behavior <id>`	Application behavior filter: • all • 2 (botnet) • 3 (evasive) • 5 (excessive bandwidth) • 6 (tunneling) • 9 (cloud)
`popularity <integer>`	Application popularity filter (1 - 5, from least to most popular).
`action {pass | block | reset}`	Pass/block traffic or reset the connection for traffic from this application (default = block).
`quarantine {none | attacker}`	Set the quarantine method: • `none`: Quarantine is disabled. • `attacker`: Block all traffic sent from attacker's IP address. The attacker's IP address is also added to the banned user list. The target's address is not affected.
`log {enable | disable}`	Enable/disable logging for this application list (default = enable).

For more information, see the FortiOS CLI reference.

Excluding signatures in application control profiles

In an application control list, the exclusion option allows users to specify a list of applications they wish to exclude from an entry filtered by category, technology, or others. By excluding the signature, the application is no longer processed on the entry in which it is excluded, but may match subsequent entries that exist.

To configure signature exclusion:

```
config application list
    edit <name>
        config entries
            edit <id>
                set category <id>
                set exclusion <application id>
                set action {pass | block | reset}
```

```
            next
        end
    next
end
```

Sample configurations

In the following example, category 23 (social media) is blocked in the entries, and signature 34527 (Instagram) is excluded from this entry. Traffic to Instagram will pass because the signature is removed from entry 1 and the action of `other-application-action` is set to `pass`.

To configure signature exclusion:

```
config application list
    edit "test"
        set other-application-action pass
        set unknown-application-action pass
        set other-application-log enable
        set unknown-application-log enable
            config entries
                edit 1
                    set category 23
                    set exclusion 34527
                    set action block
                next
            end
    next
end
```

In the following example, entry 1 is configured so that category 23 (social media) is set to pass and signature 34527 (Instagram) is excluded. In entry 2, application 34527 (Instagram) is blocked, so the traffic to Instagram will be blocked, even though it is excluded in entry 1. Traffic to other signatures in category 23, such as Facebook, will still pass.

To configure signature exclusion:

```
config application list
    edit "test"
        set other-application-action pass
        set unknown-application-action pass
        set other-application-log enable
        set unknown-application-log enable
            config entries
                edit 1
                    set category 23
                    set exclusion 34527
                    set action pass
                next
                edit 2
```

```
                    set application 34527
                    set action block
                next
            end
    next
end
```

In the following example, an explicit proxy is behind the FortiGate with an excluded signature for 107347980 (Proxy.HTTP) and category 6 (proxy) is set to block. The client will allow normal proxy traffic to pass, but it will discard all proxy application traffic (such as KProxy, Tor, and so on).

To configure signature exclusion:

```
config application list
    edit "test"
        set other-application-action pass
        set unknown-application-action pass
        set other-application-log enable
        set unknown-application-log enable
            config entries
                edit 1
                    set category 6
                    set exclusion 107347980
                    set action block
                next
            end
    next
end
```

Port enforcement check

Most networking applications run on specific ports. For example, SSH runs on port 22, and Facebook runs on ports 80 and 443.

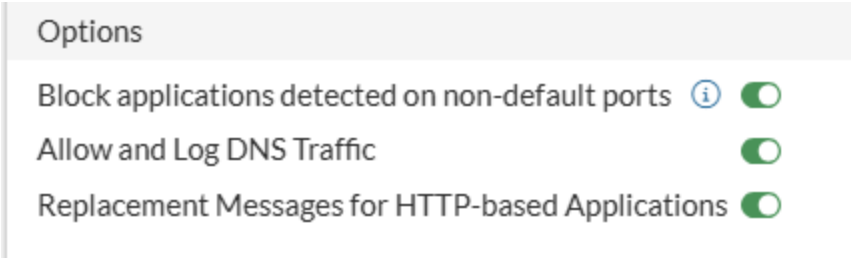
If the default network service is enabled in the application control profile, the IPS engine performs a check at the application profile level, and any detected application signatures running on the non-standard TCP/IP port are blocked. This means that each allowed application runs on its default port.

Default ports for network services are defined by FortiGuard. See Application Control Service.

In the following example, an application sensor is configured to enforce FTP on port 21.

To configure port enforcement check in the GUI:

1. Go to *Security Profiles* > *Application Control*.
2. Create a new application sensor or edit an existing one.
3. In *Options*, enable *Block applications detected on non-default ports*.



4. In *Application and Filter Overrides*, you can configure actions for individual applications, overriding the action configured for their category.
   a. Click *Create New*.
   b. Select the desired action from the dropdown list in the upper left corner. For example, select *Allow*.

      When an application is configured with an action of *Monitor* or *Allow*, when its traffic is detected on non-standard ports (as defined in FortiGuard application signatures) then the application will be blocked. An application configured with an action of *Block* will have its traffic blocked regardless of the port its traffic is detected on.

   c. Select the desired applications. For example, select *FTP*.
   d. Click *OK*. The entries are displayed in a table.
5. Click *OK*.

To configure port enforcement check in the CLI:

```
config application list
    edit <name>
        set enforce-default-app-port enable
        config entries
            edit 1
                set application 15896
                set action pass
            next
        end
    next
end
```

With the above GUI or CLI configuration, FTP traffic (application 15896 in the CLI) with the standard port (port 21) is allowed, while a non-standard port (port 2121) is blocked.

Protocol enforcement

Protocol enforcement allows you to enforce known protocols (such as FTP, HTTP, and HTTPS) on known ports (such as 21, 80, and 443). If the port of incoming traffic does not match the allowed protocols, or is not allowlisted, then the Intrusion Prevention System (IPS) engine applies the configured violation action, such as blocking or monitoring the traffic.

You can use this feature in the following scenarios:

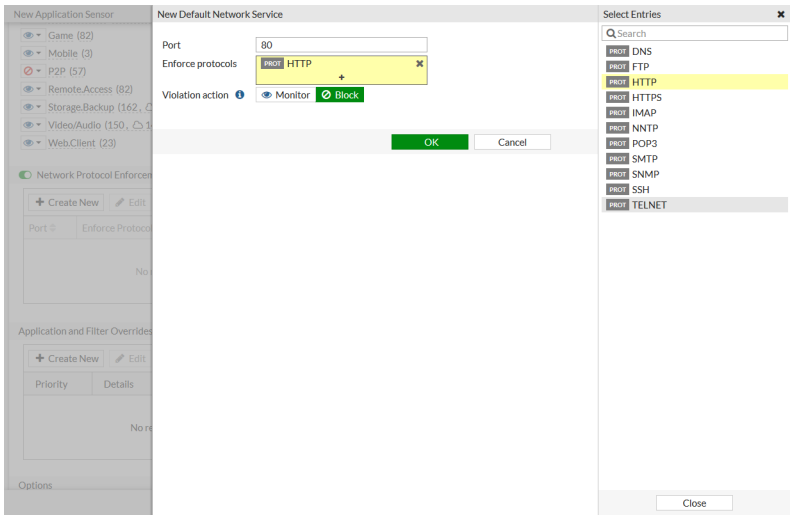
- When an IPS protocol decoder identifies a network traffic service, it checks whether the service port aligns with the enforced protocols. If the decoder finds a match, that is, the traffic is allowlisted, then the traffic proceeds without restriction. If there is no match, the traffic is classified as a violation and the specified action, monitoring or blocking, is applied.
- When there is no confirmed service for the network traffic, the traffic is classified as a violation if IPS protocol decoders rule out all of the services enforced under its server port.

In an applicable profile, a default network service list can be created to associate well known ports with accepted services. Default ports for network services are defined by FortiGuard. See Application Control Service.

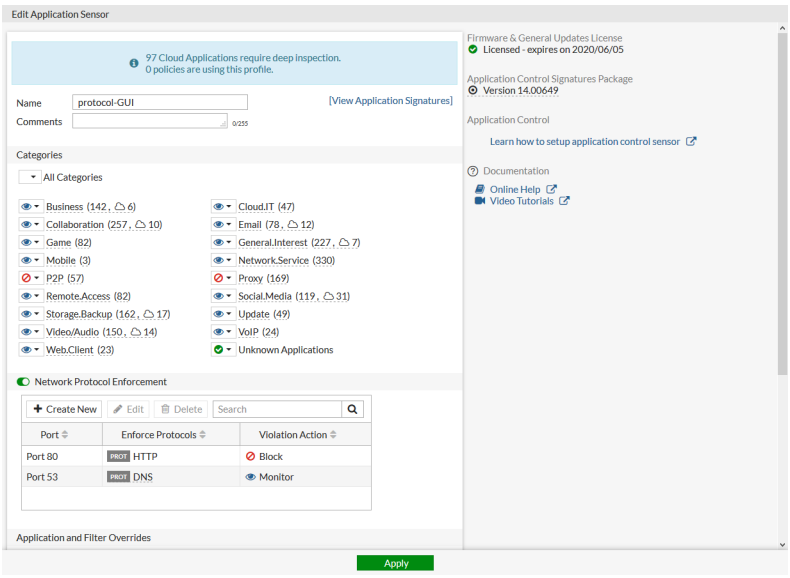
In the following example, an application sensor is configured to enforce HTTP on port 80 (block), and DNS on port 53 (monitor).

To configure protocol enforcement in the GUI:

1. Go to *Security Profiles* > *Application Control*.
2. Create a new application sensor or edit an existing one.
3. Enable *Network Protocol Enforcement*.
   Enforcement entries can be created, edited, or deleted to configure network services on certain ports and determine the violation action.
4. In the *Network Protocol Enforcement* table, click *Create New*.
5. Configure the entry for HTTP:
   a. For *Port*, enter 80.
   b. For *Enforced protocols*, select *HTTP*.
   c. For *Violation action*, select *Block*.
   d. Click *OK*.



6. Configure the entry for DNS:
   a. Click *Create New*, then for *Port*, enter 53.
   b. For *Enforced protocols*, select *DNS*.
   c. For *Violation action*, select *Monitor*.
   d. Click *OK*.
   The entries are displayed in the table.



7. Click *OK*.

To configure protocol enforcement in the CLI:

```
config application list
    edit "protocol-GUI"
        set other-application-log enable
        set control-default-network-services enable
        config default-network-services
            edit 1
                set port 80
                set services http
                set violation-action block
            next
            edit 2
                set port 53
                set services dns
                set violation-action monitor
            next
        end
    next
end
```

SSL-based application detection over decrypted traffic in a sandwich topology

When a FortiGate is sandwiched between SSL encryption and decryption devices, the FortiGate can process the decrypted traffic that passes between those devices. This feature adds support for decrypted traffic in application control. In some predefined signatures, the signature is pre-marked with the *require_ssl_di* tag. The `force-inclusion-ssl-di-sigs` option under `application list` allows users to control the inspection of dissected traffic. When this option is enabled, the IPS engine forces the pre-marked SSL-based signatures to be

applied to the decrypted traffic of the respective applications. In the following topology, SSL Proxy 1 handles the client connection and SSL Proxy 2 handles the server connection, leaving the content unencrypted as traffic passes through the FortiGate.

To configure SSL-based application detection over decrypted traffic:

```
config application list
    edit "test"
        set force-inclusion-ssl-di-sigs {enable | disable}
    next
end
```

Example pre-marked SSL-based signature:

F-SBID(--vuln_id 15722; --attack_id 42985; --name "Facebook_Chat"; --group im; --protocol tcp; --default_action pass; --revision 4446; --app_cat 23; --vendor 3; --technology 1; --behavior 9; --pop 4; --risk 2; --language "Multiple"; --weight 20; --depend-on 15832; --depend-on 38468; ***--require_ssl_di "Yes"****; --casi 1; --casi 8; --parent 15832; --app_port "TCP/443"; --severity info; --status hidden; --service http; --flow from_client; --pattern "/pull?"; --context uri; --no_case; --pattern ".facebook.com"; --context host; --no_case; --tag set,Tag.Facebook.Pull; --tag quiet; --scan-range 10m,all; --date 20190301;)*

All signatures that include the *require_ssl_di* tag are predefined and cannot be customized.

Matching multiple parameters on application control signatures

Application control signatures that support parameters (such as SCADA protocols) can have multiple parameters grouped together and matched at the same time. Multiple application parameter groups can be added to an override. Traffic will be flagged if it matches at least one parameter group.

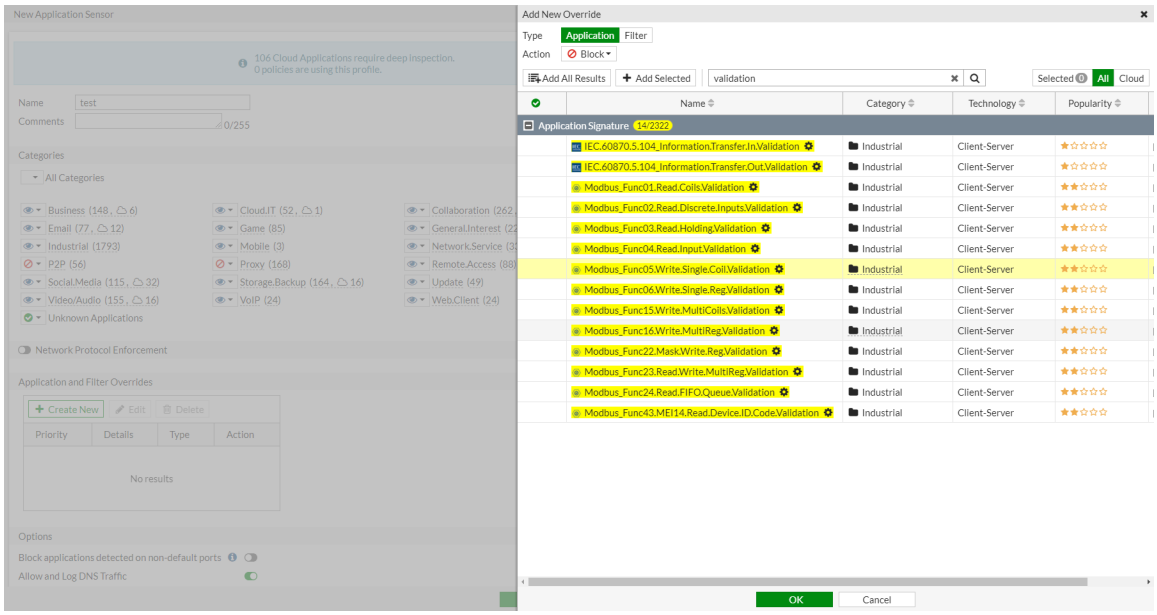
This example uses the Modbus_Func05.Write.Single.Coil.Validation signature. This is an OT signature, so ensure that no signatures are excluded:

```
config ips global
    set exclude-signatures none
end
```

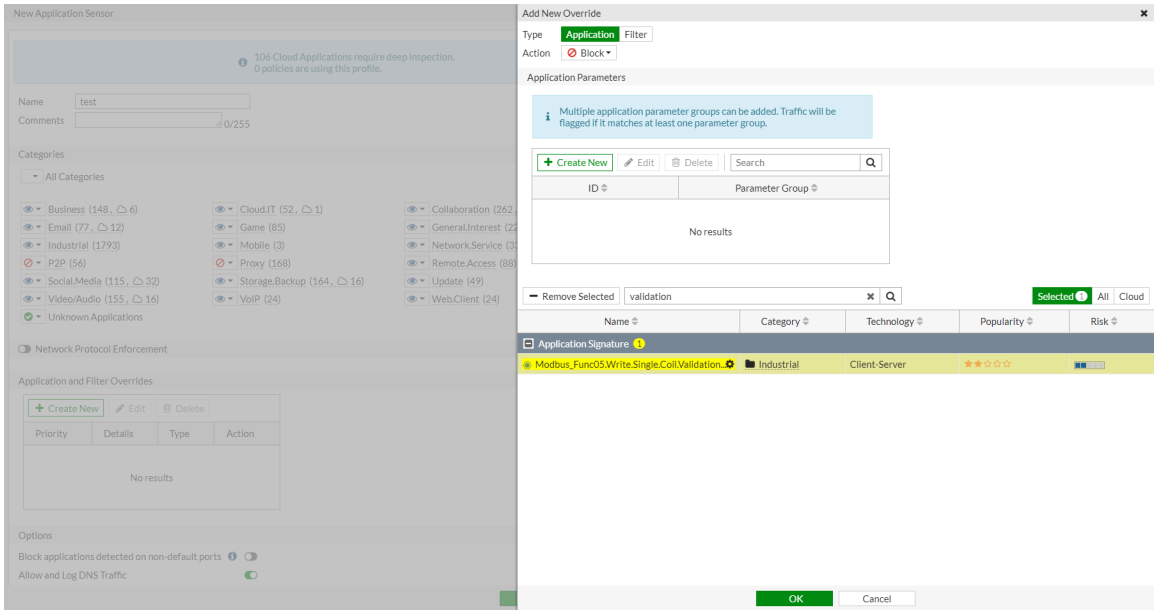
To configure an application sensor with multiple parameters in the GUI:

1. Go to *Security Profiles* > *Application Control* and click *Create New*, or edit an existing sensor.
2. In the *Application and Filter Overrides* table, click *Create New*.

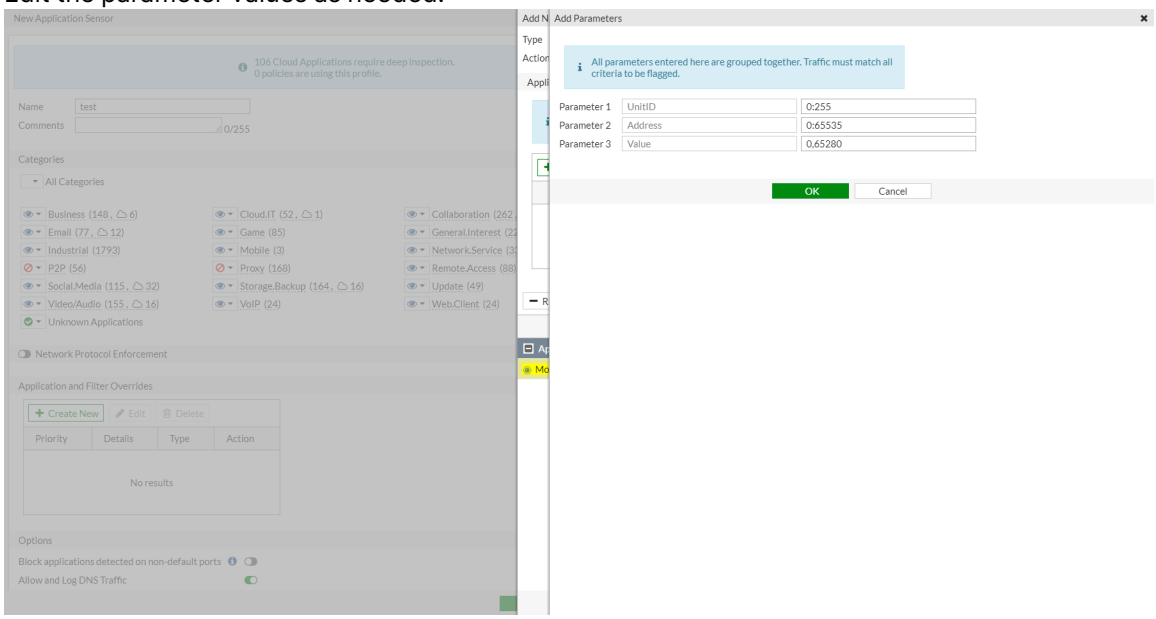
3. Search for *Modbus_Func05.Write.Single.Coil.Validation* and press `Enter`. A gear icon beside the signature name indicates it has configurable application parameters.
4. In the search results, select *Modbus_Func05.Write.Single.Coil.Validation* and click *Add Selected*.



5. Click the *Selected* tab. In the *Application Parameters* section, click *Create New*.



6. Edit the parameter values as needed.



7. Click *OK*.
8. Add more signatures if needed.
9. Click *OK*.

To configure an application sensor with multiple parameters in the CLI:

```
config application list
    edit "test"
        set other-application-log enable
        config entries
            edit 1

                set application 48885
                config parameters
                    edit 1
                        config members
                            edit 1
                                set name "UnitID"
                                set value "0:255"
                            next
                            edit 2
                                set name "Address"
                                set value "0:65535"
                            next
                            edit 3
                                set name "Value"
                                set value "0,65280"
                            next
                        end
                    next
                end
```

```
            next
            edit 2
                set category 2 6
            next
        end
    next
end
```

Application signature dissector for DNP3

The DNP3 application signature dissector supports detecting DNP3 traffic that is encapsulated by the RealPort protocol (Net.CX). DNP3 is used in industrial solutions over serial ports, USB ports, printers, and so on. RealPort encapsulation allows transportation of the underlying protocols over TCP/IP. The FortiGate OT signatures must be enabled to use RealPort.DNP3 signatures:

```
config ips global
    set exclude-signatures none
end
```

IPS engine version 7.0015 and later supports RealPort.DNP3 dissectors.

Sample logs

```
119: date=2021-03-09 time=18:56:35 eventtime=1615344995698958507 tz="-0800" logid="1059028704"
type="utm" subtype="app-ctrl" eventtype="signature" level="information" vd="vd1" appid=49890
srcip=10.1.100.191 dstip=172.16.200.159 srcport=43946 dstport=771 srcintf="port10"
srcintfrole="undefined" dstintf="port9" dstintfrole="undefined" proto=6 service="RLDNP3"
direction="incoming" policyid=1 sessionid=1204 applist="test" action="pass"
appcat="Operational.Technology" app="RealPort.DNP3" incidentserialno=88083610
msg="Operational.Technology: RealPort.DNP3," apprisk="elevated"
```

```
1: date=2021-03-09 time=18:56:08 eventtime=1615344968811546102 tz="-0800" logid="1059028704"
type="utm" subtype="app-ctrl" eventtype="signature" level="information" vd="vd1" appid=49899
srcip=10.1.100.191 dstip=172.16.200.159 srcport=43946 dstport=771 srcintf="port10"
srcintfrole="undefined" dstintf="port9" dstintfrole="undefined" proto=6 service="RLDNP3"
direction="outgoing" policyid=1 sessionid=1204 applist="test" action="pass"
appcat="Operational.Technology" app="RealPort.DNP3_Confirm" incidentserialno=88083404
msg="Operational.Technology: RealPort.DNP3_Confirm," clouduser="34 -> 34" filename="Null"
apprisk="elevated" cloudaction="others"
```

Domain fronting protection

FortiOS can protect against domain fronting in both explicit proxy and proxy-based firewall policies. In both cases, FortiGate checks whether the domain of the request matches the host domain in the HTTP header, and then allows, blocks, or monitors the traffic. This feature enhances security by preventing unauthorized access that could result from domain mismatches.

```
config firewall profile-protocol-options
    edit protocol
        config http
            set domain-fronting {allow | monitor | block | strict}
        next
    end
end
```

Command	Description
`domain-fronting {allow | block | monitor}`	Configure HTTP domain fronting (default = block). • `allow`: allow domain fronting. • `monitor`: allow and log domain fronting. • `block`: block and log domain fronting. • `strict`: block and log domain fronting. Will block potential matching IP and domain.

Domain fronting protection supports HTTP/1.1 but not HTTP/2.

Example

In this example, the server name indication (SNI) in the request is `httpbin.org`, and the host header in the request is `google.com`.

When FortiGate has an explicit proxy policy configured with `set domain-fronting block`, traffic is blocked and logged when the request domain does not match the HTTP header domain.

- Example traffic log:

```
1: date=2024-06-11 time=10:38:23 eventtime=1718127503650731465 tz="-0700" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="vdom1" srcip=10.1.100.77 srcport=41548
srcintf="port2" srcintfrole="undefined" dstip=3.211.196.247 dstport=443 dstintf="port3"
dstintfrole="undefined" srccountry="Reserved" dstcountry="United States" sessionid=1542161161
proto=6 action="deny" policyid=1 policytype="proxy-policy" poluuid="01352fb2-1370-51ef-8ac3-
c46f77827b80" service="HTTPS" trandisp="noop" duration=0 sentbyte=0 rcvdbyte=0 sentpkt=0
rcvdpkt=0 appcat="unscanned" utmaction="block" countweb=1 msg="Traffic denied because of
domain fronting" utmref=65498-0
```

- Example Web filter UTM log:

```
1: date=2024-06-11 time=10:38:23 eventtime=1718127503650663438 tz="-0700" logid="0320013318"
type="utm" subtype="webfilter" eventtype="domain-fronting" level="notice" vd="vdom1"
policyid=1 poluuid="01352fb2-1370-51ef-8ac3-c46f77827b80" policytype="proxy-policy"
sessionid=1542161161 transid=1 srcip=10.1.100.77 srcport=41548 srccountry="Reserved"
srcintf="port2" srcintfrole="undefined" dstip=3.211.196.247 dstport=443 dstcountry="United
States" dstintf="port3" dstintfrole="undefined" proto=6 httpmethod="GET" service="HTTPS"
hostname="google.com" agent="curl/7.83.1" profile="protocol" action="blocked" reqtype="direct"
url="https://google.com/" sentbyte=860 rcvdbyte=5470 direction="outgoing" msg="Domain fronting
detected" rawdata="HTTP Host <google.com> does not match SNI <httpbin.org>"
```

When FortiGate has a transparent proxy policy configured with `set domain-fronting monitor`, traffic is passed and logged when the request domain does not match the HTTP header domain.

- Example traffic log:

```
1: date=2024-06-11 time=11:14:22 eventtime=1718129661884640964 tz="-0700" logid="0000000010"
type="traffic" subtype="forward" level="notice" vd="vdom1" srcip=10.1.100.77 srcport=44250
srcintf="port2" srcintfrole="undefined" dstcountry="United States" srccountry="Reserved"
dstip=3.211.196.247 dstport=443 dstintf="port3" dstintfrole="undefined" sessionid=2024
service="web" proxyapptype="web-proxy" proto=6 action="accept" policyid=22 policytype="proxy-
policy" poluuid="05d56dfc-1370-51ef-5315-e0ee922dd3b5" trandisp="snat" transip=172.16.200.2
transport=44250 duration=0 wanin=15331 rcvdbyte=15331 wanout=578 lanin=777 sentbyte=777
lanout=12868 appcat="unscanned" utmaction="allow" countweb=1 utmref=65496-0
```

- Example Web filter UTM log:

```
1: date=2024-06-11 time=11:14:22 eventtime=1718129661808321505 tz="-0700" logid="0320013318"
type="utm" subtype="webfilter" eventtype="domain-fronting" level="notice" vd="vdom1"
policyid=22 poluuid="05d56dfc-1370-51ef-5315-e0ee922dd3b5" policytype="proxy-policy"
sessionid=2024 transid=1 srcip=10.1.100.77 srcport=44250 srccountry="Reserved" srcintf="port2"
srcintfrole="undefined" srcuuid="e3659a44-136f-51ef-2b98-b3a63e02baed" dstip=3.211.196.247
dstport=443 dstcountry="United States" dstintf="port3" dstintfrole="undefined"
dstuuid="e3659a44-136f-51ef-2b98-b3a63e02baed" proto=6 httpmethod="GET" service="HTTPS"
hostname="google.com" agent="curl/7.83.1" profile="protocol" action="passthrough"
reqtype="direct" url="https://google.com/" sentbyte=746 rcvdbyte=5470 direction="outgoing"
msg="Domain fronting detected" rawdata="HTTP Host <google.com> does not match SNI
<httpbin.org>"
```

Inline CASB

The inline CASB security profile enables the FortiGate to perform granular control over SaaS applications directly on firewall policies. The supported controls include:

Control	Description
Privilege control	Specify the action to apply to user activities per application such as upload, download, share, delete, log in, and so on. See Privilege control on page 2450 for an example.
Safe search	On SaaS applications that support searching, enable and select the level of safe search. See Safe search on page 2453 for an example.
Tenant control	Allow only users belonging to specific domains to access the SaaS application. See Tenant control on page 2455 for an example.
Advanced tenant control	Allows for much more granular manipulation of HTTP requests, including custom headers and dynamic attribute mapping.

Control	Description
	See Advanced tenant control on page 2459 for an example.
UTM bypass	For each user activity, bypass further UTM scanning any of the following security profiles: • Antivirus • DLP • File filter • Video filter • Web filter See UTM bypass on page 2462 for an example.

Administrators can customize their own SaaS applications, matching conditions, and custom controls and actions.

A firewall policy must use proxy-based inspection with a deep inspection SSL profile to apply the inline CASB profile and scan the traffic payload.

Inline CASB can be applied to a firewall policy or a proxy policy.

This feature is not supported on FortiGate models with 2 GB RAM or less. See Proxy-related features not supported on FortiGate 2 GB RAM models on page 126 for more information.

The Inline-CASB Application Definitions entitlement is licensed under the basic firmware and updates contract. To view the entitlement information, go to *System > FortiGuard* and expand the *Firmware, VM, and base subscriptions* section in the *Subscriptions > Licensed* tab.

To enable inline CASB security profiles in the GUI:

1. Go to *System > Feature Visibility*.
2. Enable *Inline-CASB* in the *Security Features* section.
3. Click *Apply*.

See Inline CASB examples on page 2449 for sample configurations.

Inline CASB examples

The following examples are included:

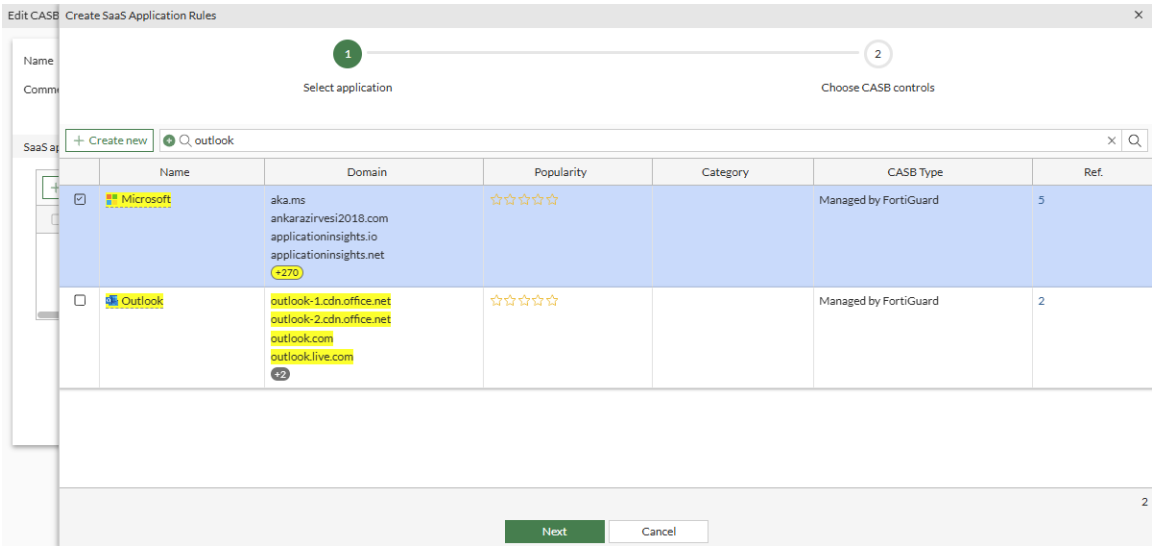
- Privilege control on page 2450
- Safe search on page 2453
- Tenant control on page 2455
- Advanced tenant control on page 2459
- UTM bypass on page 2462
- Customized SaaS application and user activity on page 2465
- Support control factors in exchanged JSON data for custom SaaS applications on page 2470

Privilege control

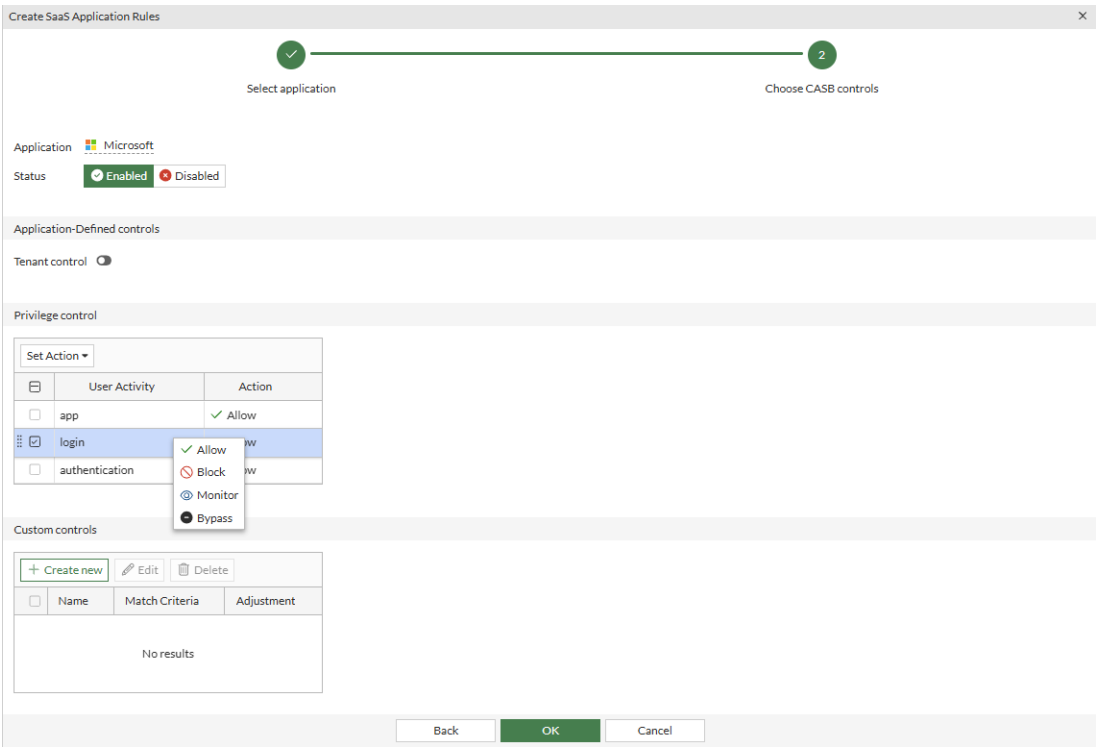
In this example, logging in to Microsoft Outlook is blocked by the privilege control settings in the inline CASB profile.

To configure an inline CASB profile with privilege control in the GUI:

1. Configure the inline CASB profile:
   a. Go to *Security Profiles* > *Inline-CASB* and click *Create new*.
   b. Enter a *Name*, such as *outlook_test*.
   c. In the *SaaS Applications* table, click *Create new*. The *Create SaaS Application Rules* pane opens.
   d. Search *Outlook* and select the *Microsoft* application, then click *Next*.



   e. In the *Privilege Control* table, select *login* and from the *Set Action* dropdown, select *Block*.



f. Click *OK* to save the rule.
g. Click *OK* to save the CASB Profile.

2. Configure the firewall policy:
   a. Go to *Policy & Objects* > *Firewall Policy*. Edit an existing policy, or create a new one.
   b. Set the *Inspection Mode* to *Proxy-based*.
   c. In the *Security Profiles* section, enable *Inline-CASB* and select the *outlook_test* profile.
   d. Set the *SSL Inspection* profile to one that uses deep inspection.
   e. Configure the other settings as needed.
   f. Click *OK*.

To configure an inline CASB profile with privilege control in the CLI:

1. Configure the inline CASB profile:

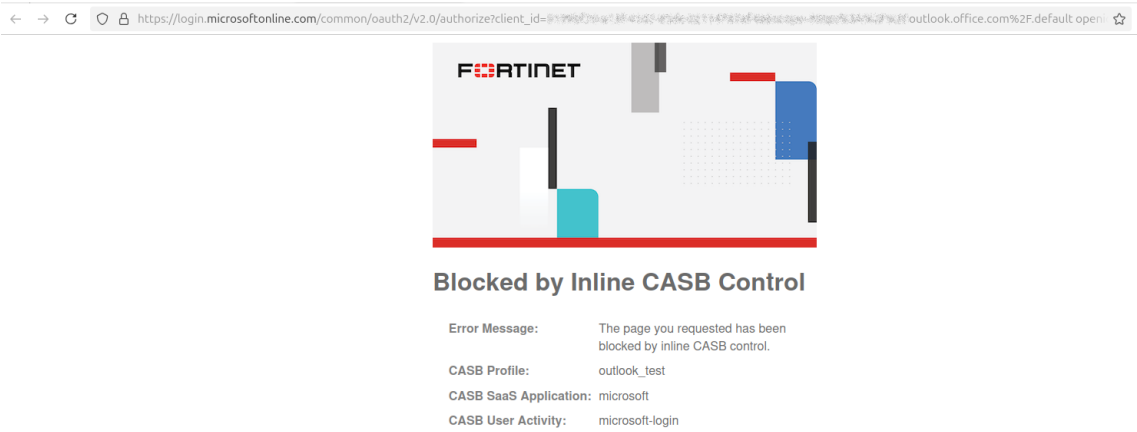
```
config casb profile
    edit "outlook_test"
        config saas-application
            edit "microsoft"
                config access-rule
                    edit "microsoft-login"
                        set action block
                    next
                end
            next
        end
    next
end
```

2. Configure the firewall policy:

```
config firewall policy
    edit 6
        set name "casb_test"
        set srcintf "port1"
        set dstintf "port3"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "ssl"
        set casb-profile "outlook_test"
        set nat enable
    next
end
```

To test the configuration:

1. Open a browser and attempt to access the Outlook login page.
2. The traffic is blocked by the firewall policy. The browser displays a replacement message: *Blocked by Inline CASB Control*.



Sample log:

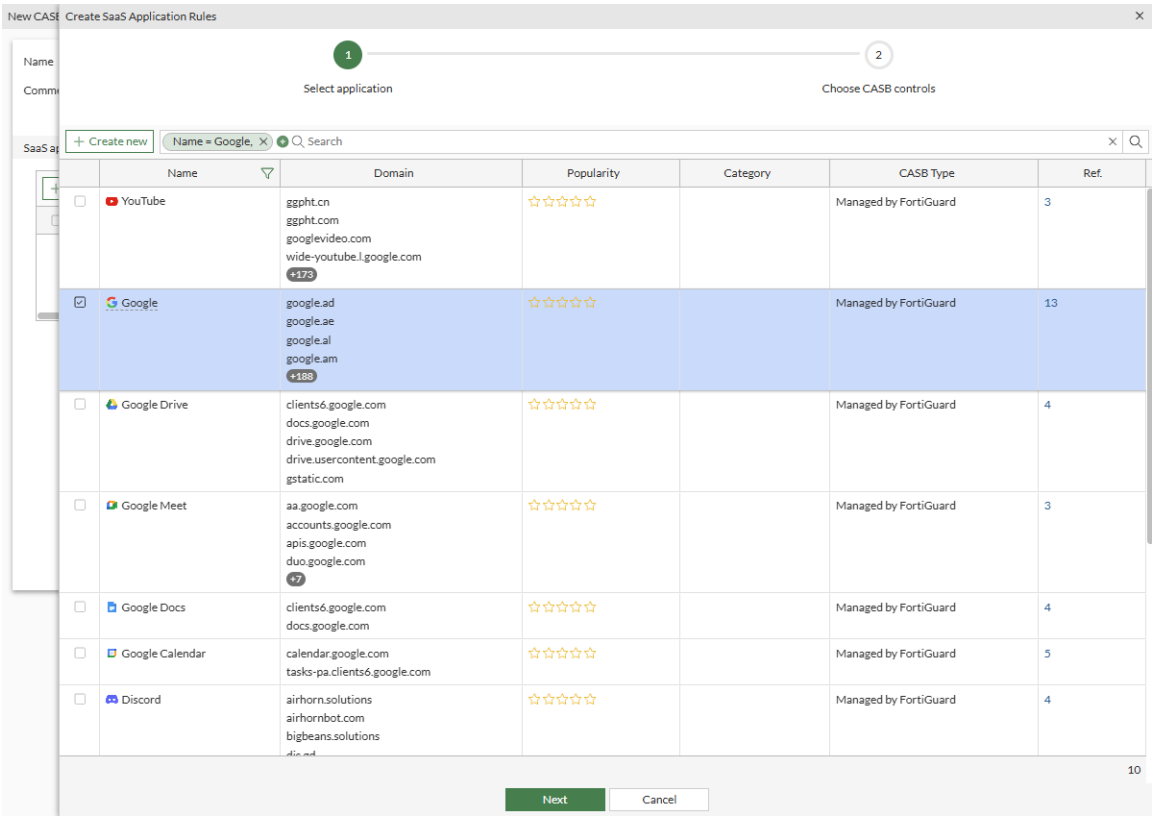
```
1: date=2026-04-07 time=17:07:45 eventtime=1775606865001260027 tz="-0700" logid="2500010000"
type="utm" subtype="casb" eventtype="casb" level="warning" vd="root" policyid=6 poluuid="23fa6e58-
fd3a-51f0-fcba-755a1c0b5610" sessionid=440036 srcip=13.13.13.13 dstip=40.126.24.83 srcport=45698
dstport=443 srcintf="port1" srcintfrole="undefined" srcuuid="1c9e034e-72bc-51f0-4866-42e7eacff7bb"
dstintf="port3" dstintfrole="undefined" dstuuid="70b43f6e-1e2b-51f1-c748-c3c24466ed2e" proto=6
url="https://login.microsoftonline.com/favicon.ico" action="block" profile="outlook_test"
saasapp="microsoft" useractivity="microsoft-login" subaction="monitor" tenantmatch="missed"
activitycategory="activity-control" msg="CASB access was blocked because it contained banned
activity."
```

Safe search

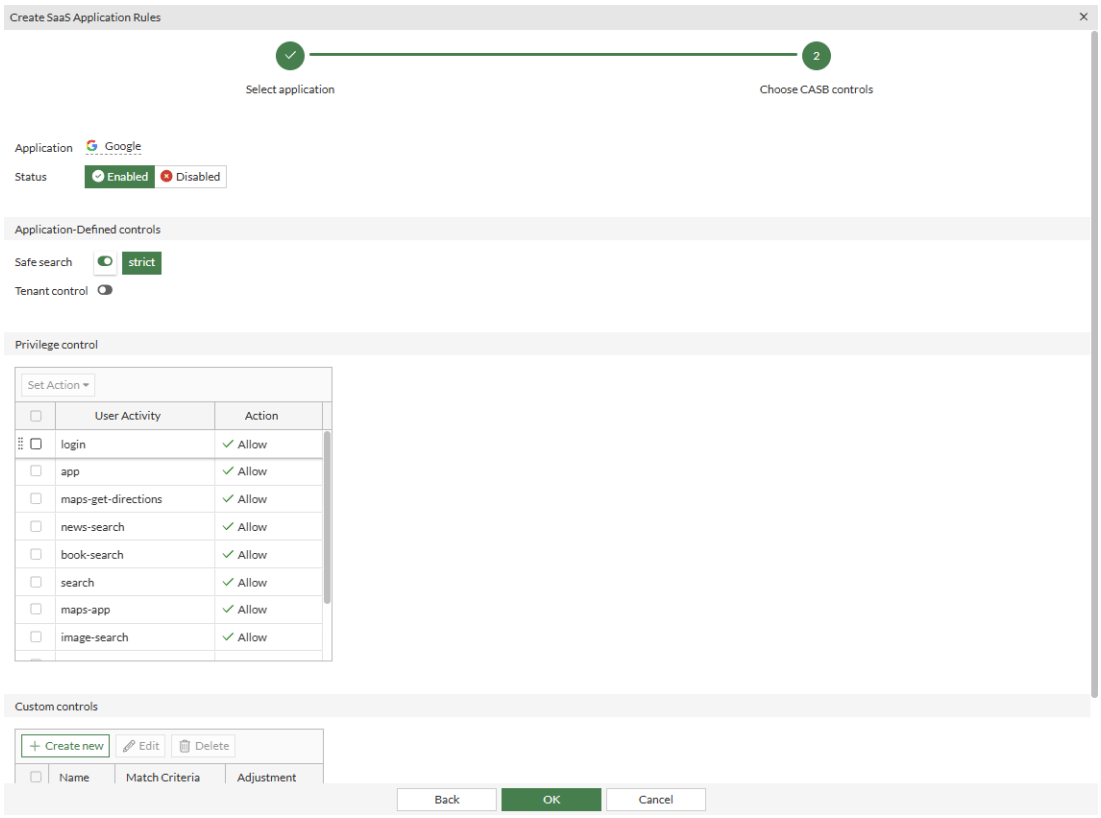
In this example, safe search is configured for Google in the inline CASB profile.

To configure an inline CASB profile with safe search in the GUI:

1. Configure the inline CASB profile:
   a. Go to *Security Profiles* > *Inline-CASB* and click *Create new*.
   b. Enter a *Name*, such as *google_test*.
   c. In the *SaaS Applications* table, click *Create new*. The *Create SaaS Application Rules* pane opens.
   d. Search *google* and select the *google* application, then click *Next*.



   e. Enable *Safe search*.



f. Click *OK* to save the rule.

g. Click *OK* to save the CASB Profile.

2. Configure the firewall policy:

a. Go to *Policy & Objects* > *Firewall Policy*. Edit an existing policy, or create a new one.

b. Set the *Inspection Mode* to *Proxy-based*.

c. In the *Security Profiles* section, enable *Inline-CASB* and select the *google_test* profile.

d. Set the *SSL Inspection* profile to one that uses deep inspection.

e. Configure the other settings as needed.

f. Click *OK*.

To configure an inline CASB profile with safe search in the CLI:

1. Configure the inline CASB profile:

```
config casb profile
    edit "google_test"
        config saas-application
            edit "google"
                set safe-search enable
                set safe-search-control "strict"
            next
        end
    next
end
```

2. Configure the firewall policy:

```
config firewall policy
    edit 7
        set name "casb_test_google"
        set srcintf "port1"
        set dstintf "port3"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "ssl"
        set casb-profile "google_test"
        set nat enable
    next
end
```

To test the configuration:

1. Open a browser and attempt to search in Google for content that is considered mature or explicit.
2. The sensitive content is filtered out in the search results.

Sample log:

```
1: date=2026-04-08 time=09:01:46 eventtime=1775664106064977832 tz="-0700" logid="2500010002"
type="utm" subtype="casb" eventtype="casb" level="information" vd="root" policyid=7
poluuid="23fa6e58-fd3a-51f0-fcba-755a1c0b5610" sessionid=482175 srcip=13.13.13.13
dstip=142.251.154.119 srcport=60126 dstport=443 srcintf="port1" srcintfrole="undefined"
srcuuid="1c9e034e-72bc-51f0-4866-42e7eacff7bb" dstintf="port3" dstintfrole="undefined"
dstuuid="70b43f6e-1e2b-51f1-c748-c3c24466ed2e" proto=6
url="https://www.google.com/complete/search?q=explicit.com&safe=active" action="monitor"
profile="google_test" saasapp="google" useractivity="google-safe-search" operation="strict"
subaction="monitor" tenantmatch="missed" activitycategory="safe-search-control" msg="CASB access
was monitored because it contained activity."
```

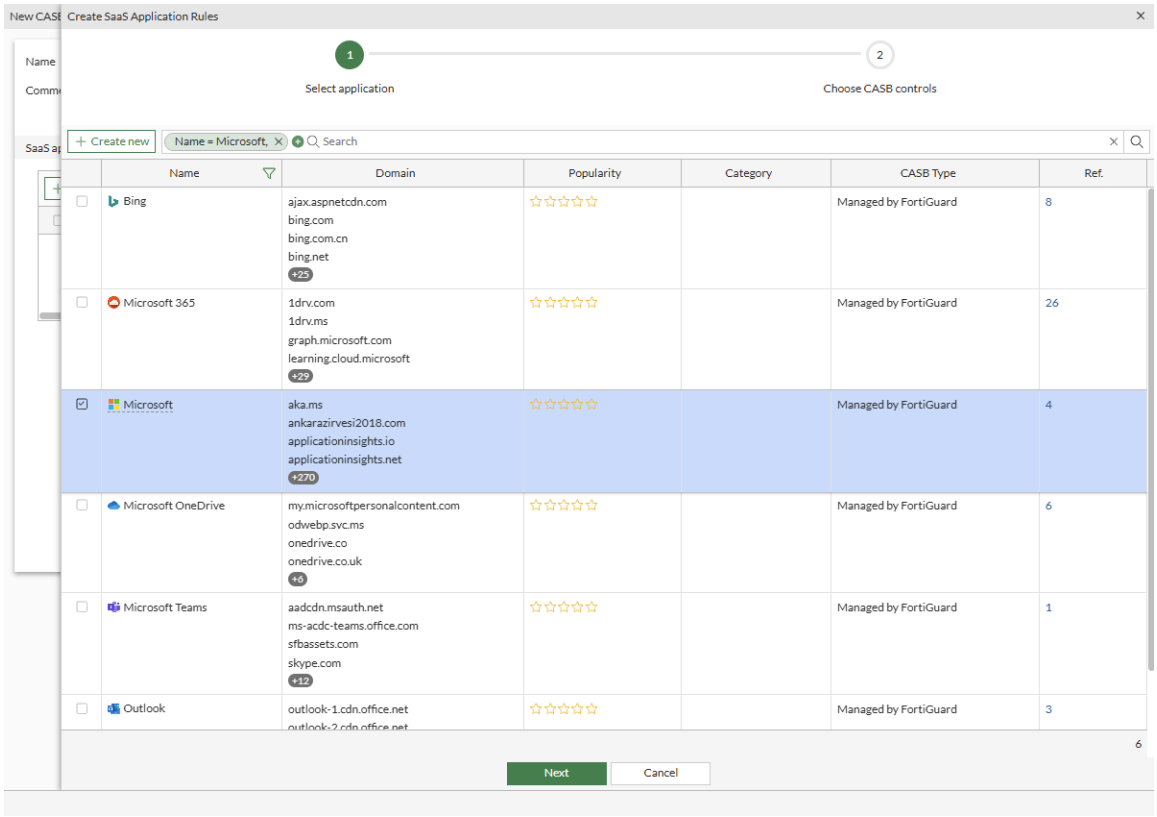
Tenant control

In this example, tenant control is configured for Microsoft in the inline CASB profile for the fortinet-us.com domain.

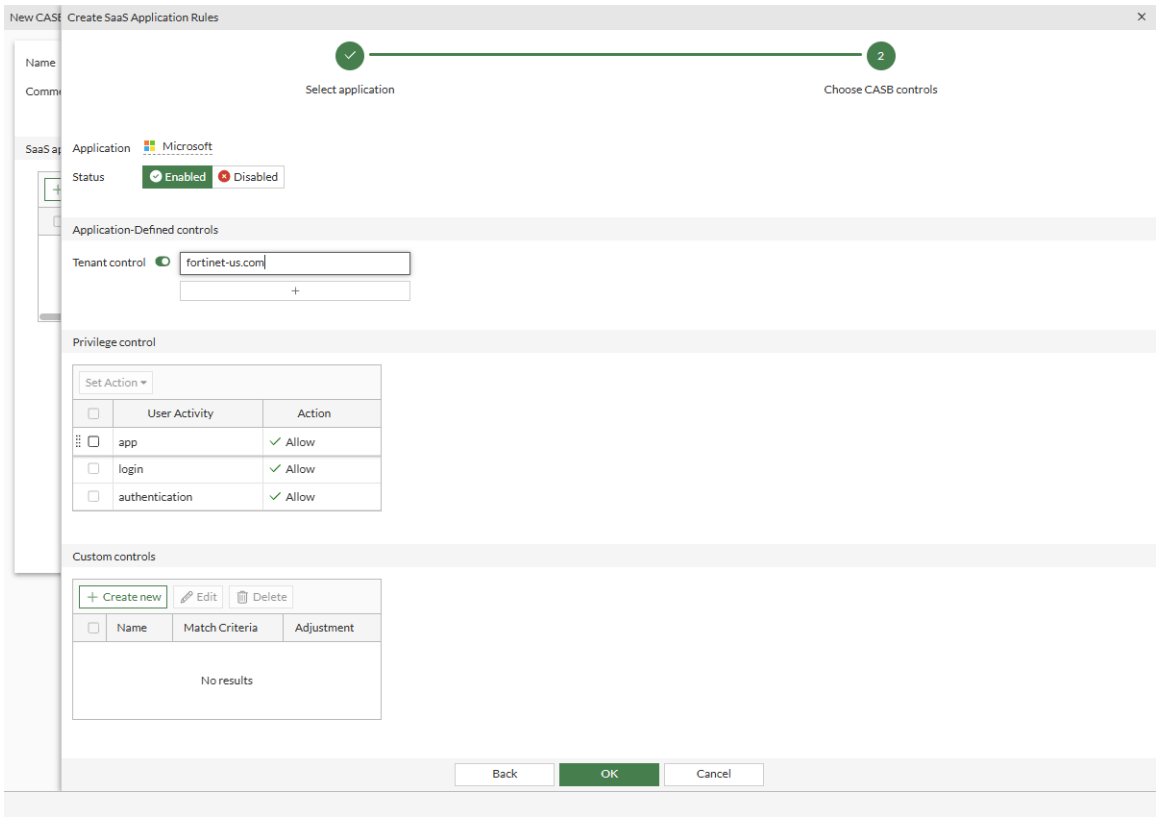
To configure an inline CASB profile with tenant control in the GUI:

1. Configure the inline CASB profile:
   a. Go to *Security Profiles* > *Inline-CASB* and click *Create new*.
   b. Enter a *Name*, such as *microsoft_test*.
   c. In the *SaaS Applications* table, click *Create new*. The *Create SaaS Application Rules* pane opens.

d. Search *Microsoft* and select the *Microsoft* application, then click *Next*.



e. Enable *Tenant control*. Click the *+* and enter *fortinet-us.com*.



f. Click *OK* to save the rule.

g. Click *OK* to save the CASB Profile.

2. Configure the firewall policy:

a. Go to *Policy & Objects* > *Firewall Policy*. Edit an existing policy, or create a new one.

b. Set the *Inspection Mode* to *Proxy-based*.

c. In the *Security Profiles* section, enable *Inline-CASB* and select the *microsoft_test* profile.

d. Set the *SSL Inspection* profile to one that uses deep inspection.

e. Configure the other settings as needed.

f. Click *OK*.

To configure an inline CASB profile with tenant control in the CLI:

1. Configure the inline CASB profile:

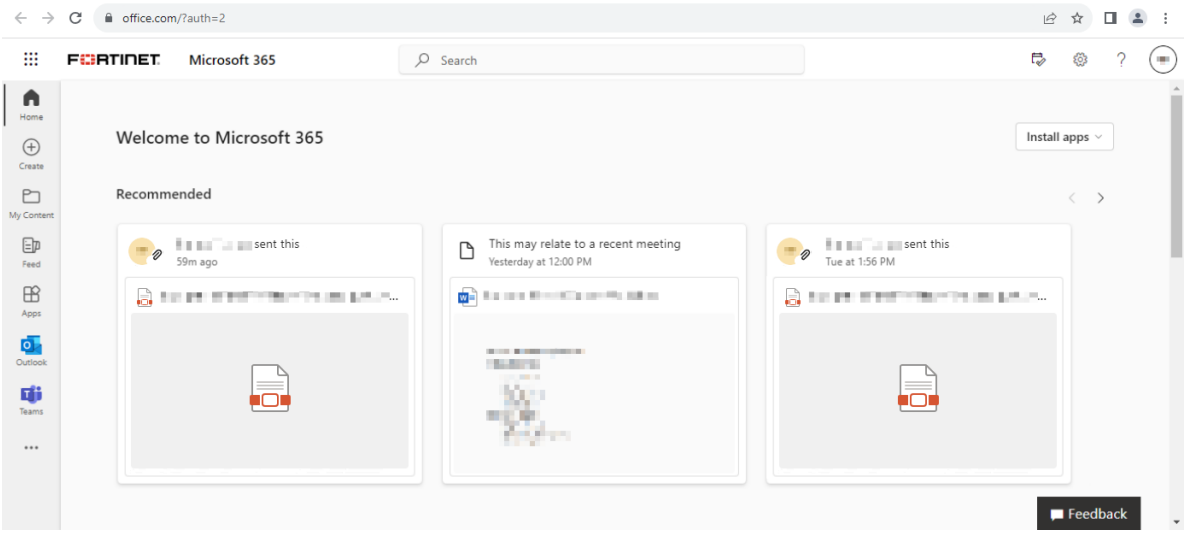
```
config casb profile
    edit "microsoft_test"
        config saas-application
            edit "microsoft"
                set tenant-control enable
                set tenant-control-tenants "fortinet-us.com"
            next
        end
    next
end
```

2. Configure the firewall policy:

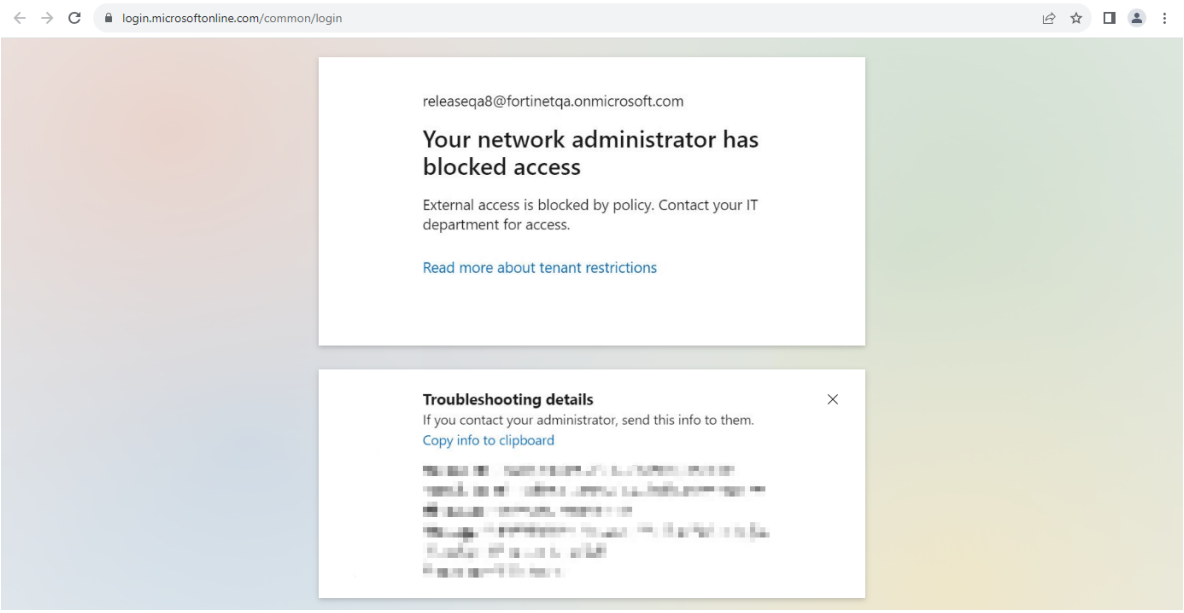
```
config firewall policy
    edit 8
        set name "casb_test_microsoft"
        set srcintf "port1"
        set dstintf "port3"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "ssl"
        set casb-profile "microsoft_test"
        set nat enable
    next
end
```

To test the configuration:

1. Open a browser and attempt to log in to Microsoft Office 365 with a fortinet-us.com account.
2. Since the domain is valid, the user can log in successfully.



3. Attempt to log in to Microsoft Office 365 with another account with a different domain.
4. The domain is invalid. The user is unable to log in, and an error message appears: *Your network administrator has blocked access*.



Sample log:

```
1: date=2026-04-08 time=09:33:27 eventtime=1775666006955935136 tz="-0700" logid="2500010002"
type="utm" subtype="casb" eventtype="casb" level="information" vd="root" policyid=8
poluuid="23fa6e58-fd3a-51f0-fcba-755a1c0b5610" sessionid=484445 srcip=13.13.13.13
dstip=20.190.190.130 srcport=59380 dstport=443 srcintf="port1" srcintfrole="undefined"
srcuuid="1c9e034e-72bc-51f0-4866-42e7eacff7bb" dstintf="port3" dstintfrole="undefined"
dstuuid="70b43f6e-1e2b-51f1-c748-c3c24466ed2e" proto=6
url="https://login.microsoftonline.com/common/login" action="monitor" profile="microsoft_test"
saasapp="microsoft" useractivity="microsoft-tenant-control" operation="tenant" subaction="monitor"
tenantmatch="missed" activitycategory="tenant-control" msg="CASB access was monitored because it
contained activity."
```

Advanced tenant control

Advanced tenant control allows administrators to go beyond simple tenant restriction by performing granular manipulation of HTTP headers—including adding, modifying, or deleting request headers. By defining custom SaaS applications and user activities, administrators can inject specific metadata (such as business unit, location, or tenant IDs) into the traffic flow for better visibility or compliance at the destination SaaS application.

To configure a customized inline CASB profile with advanced tenant control:

1. Configure the CASB SaaS application:

```
config casb saas-application
    edit "httpbin"
        set domains "httpbin.org"
        config input-attributes
            edit "business_name"
            next
```

```
            edit "city"
            next
            edit "province"
            next
        end
    next
end
```

2. Configure the CASB user activity:

```
config casb user-activity
    edit "httpbin-tenant"
        set application "httpbin"
        set category advanced-tenant-control
        config control-options
            edit "tenant-control"
                config operations
                    edit "add_header1"
                        set action new-on-not-found
                        set header-name "X-Business-Name"
                        set value-from-input enable
                        set value-name-from-input "business_name"
                    next
                    edit "add_header2"
                        set action new-on-not-found
                        set header-name "X-City"
                        set value-from-input enable
                        set value-name-from-input "city"
                    next
                    edit "add_header3"
                        set action new-on-not-found
                        set header-name "X-Province"
                        set value-from-input enable
                        set value-name-from-input "province"
                    next
                end
            next
        end
    next
end
```

3. Configure the inline CASB profile:

```
config casb profile
    edit "advanced_casb"
        config saas-application
            edit "httpbin"
                set tenant-control enable
                config advanced-tenant-control
                    edit "httpbin-tenant"
                        config attribute
                            edit "business_name"
```

```
                                set input "Pochiya Limited"
                            next
                            edit "city"
                                set input "Coquitlam"
                            next
                            edit "province"
                                set input "BC"
                            next
                        end
                    next
                end
            next
        end
    next
end
```

4. Configure the firewall policy:

```
config firewall policy
    edit 1
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "deep-inspection"
        set casb-profile "advanced_casb"
        set schedule "always"
        set service "HTTPS"
        set action accept
    next
end
```

Verification

To verify that the headers are being injected correctly, perform a request from a client behind the FortiGate to the configured domain using `curl`:

```
# curl -k https://httpbin.org/headers
{
  "headers": {
    "Accept": "*/*",
    "Host": "httpbin.org",
    "User-Agent": "curl/7.68.0",
    "X-Business-Name": "Pochiya Limited",
    "X-City": "Coquitlam",
    "X-Province": "BC"
  }
}
```

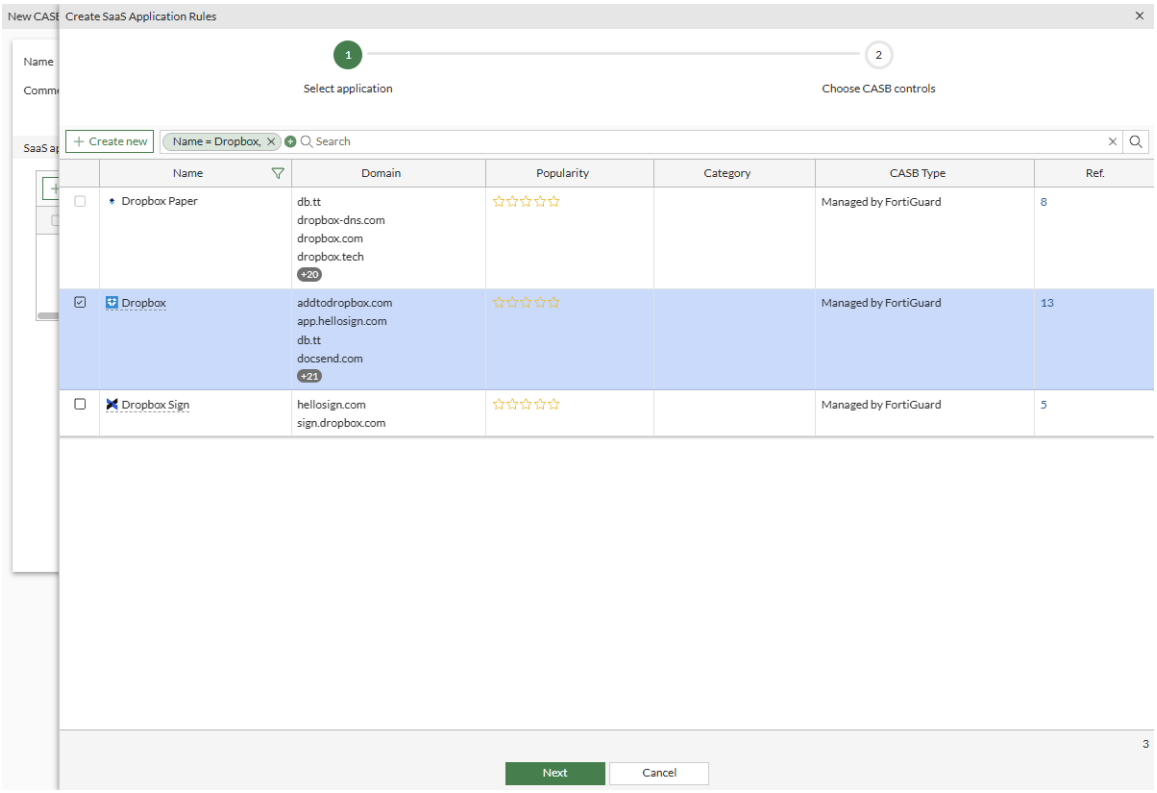
The successful verification of the custom headers (X-Business-Name, X-City, and X-Province) confirms that the FortiGate is successfully intercepting the HTTPS traffic, identifying the SaaS application context, and performing real-time header manipulation.

UTM bypass

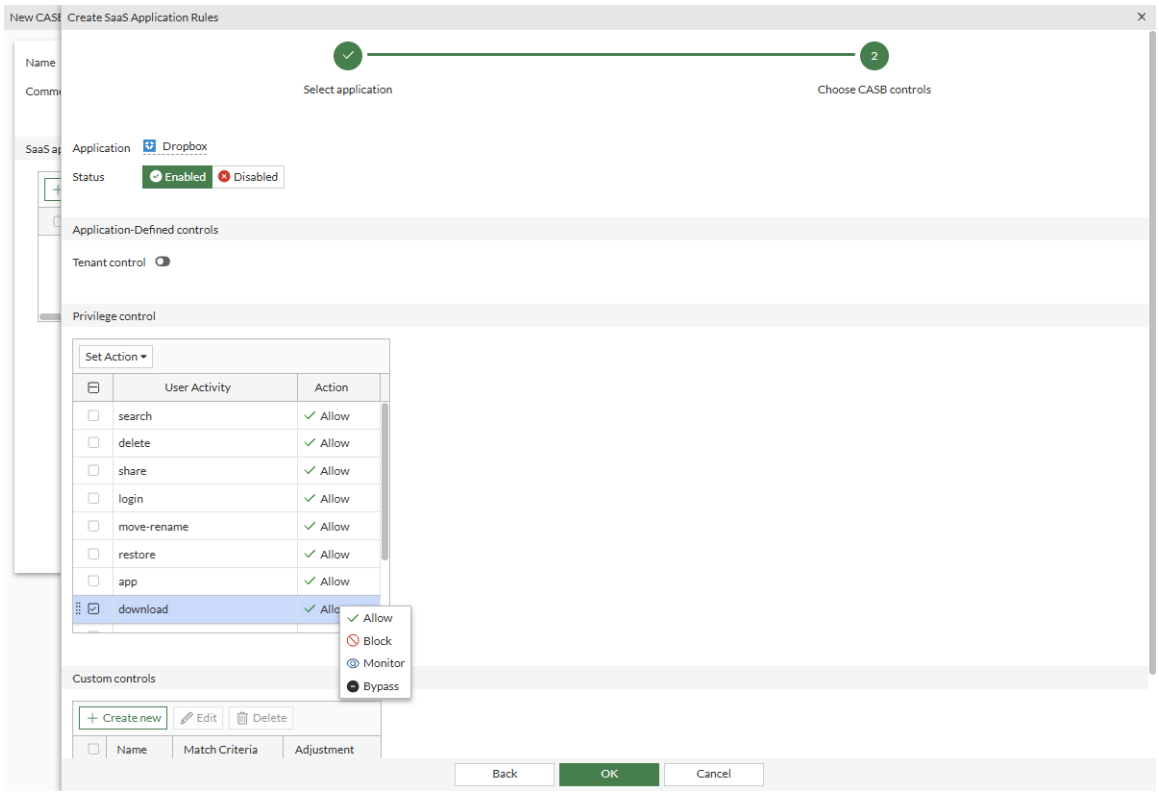
In this example, UTM bypass is configured for Dropbox file downloading in the inline CASB profile.

To configure an inline CASB profile with UTM bypass in the GUI:

1. Configure the inline CASB profile:
   a. Go to *Security Profiles* > *Inline-CASB* and click *Create new*.
   b. Enter a *Name*, such as *dropbox_test*.
   c. In the *SaaS Applications* table, click *Create new*. The *Create SaaS Application Rules* pane opens.
   d. Search *dropbox* and select the *dropbox* application, then click *Next*.

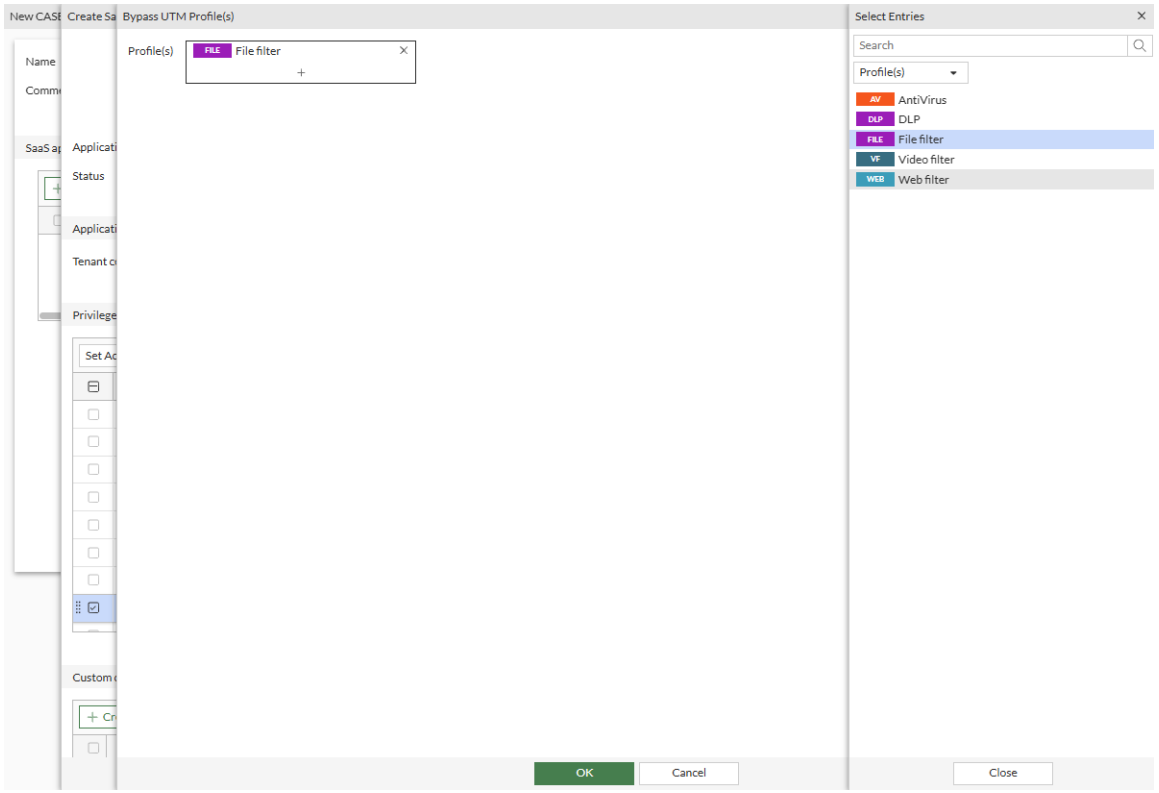


   e. In the *Privilege Control* table, select *download-file* and from the *Set Action* dropdown, select *Bypass*.



The *Bypass UTM Profile(s)* pane opens.

f. Click the *+* and set *Profile(s)* to *File Filter*.



g. Click *OK* to save the bypass UTM profile.

h. Click *OK* to save the inline CASB rule.
i. Click *OK* to save the inline CASB profile

2. Configure the firewall policy:
   a. Go to *Policy & Objects* > *Firewall Policy*. Edit an existing policy, or create a new one.
   b. Set the *Inspection Mode* to *Proxy-based*.
   c. In the *Security Profiles* section, enable *Inline-CASB* and select the *dropbox_test* profile.
   d. Set the *SSL Inspection* profile to one that uses deep inspection.
   e. Configure the other settings as needed.
   f. Click *OK*.

To configure an inline CASB profile with UTM bypass in the CLI:

1. Configure the inline CASB profile:

```
config casb profile
    edit "dropbox_test"
        config saas-application
            edit "dropbox"
                config access-rule
                    edit "dropbox-download-file"
                        set bypass file-filter
                        set action bypass
                    next
                end
            next
        end
    next
end
```

2. Configure the firewall policy:

```
config firewall policy
    edit 9
        set name "casb_test_dropbox"
        set srcintf "port1"
        set dstintf "port3"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "ssl"
        set casb-profile "dropbox_test"
        set nat enable
    next
end
```

To test the configuration:

1. Open a browser and log in to Dropbox.
2. Attempt to download a file, such as a PDF. The download is successful.

Sample log:

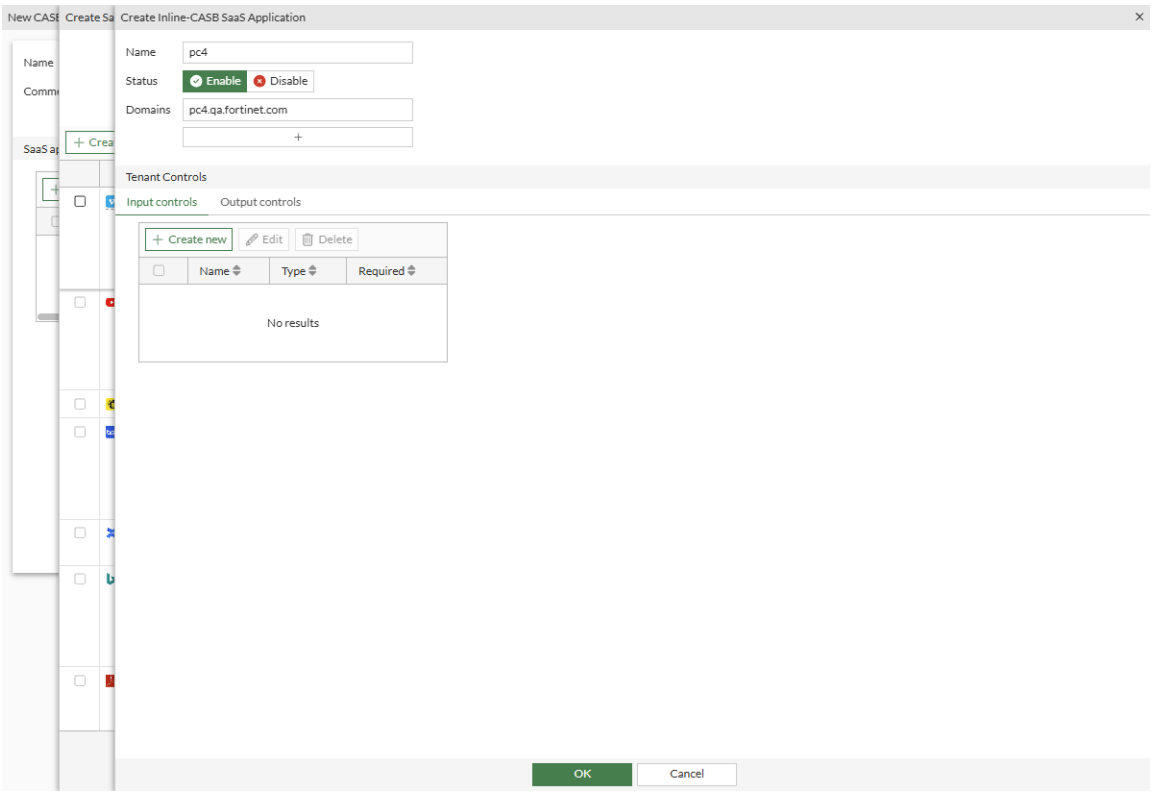
```
1: date=2026-04-08 time=10:55:23 eventtime=1775670922625406688 tz="-0700" logid="2500010001"
type="utm" subtype="casb" eventtype="casb" level="information" vd="root" policyid=9
poluuid="23fa6e58-fd3a-51f0-fcba-755a1c0b5610" policytype="policy" sessionid=3574
srcip=13.13.13.13 dstip=162.125.1.15 srcport=48562 dstport=443 srcintf="port1"
srcintfrole="undefined" srcuuid="1c9e034e-72bc-51f0-4866-42e7eacff7bb" dstintf="port3"
dstintfrole="undefined" dstuuid="70b43f6e-1e2b-51f1-c748-c3c24466ed2e" proto=6
url="https://uc3b405d6c8b72a0ba53075676f1.dl.dropboxusercontent.com/cd/0/get/C-OZIEfbqTRl_N_
lx1yMpq31pEBeubCP_
e1nsVnQuBjiDcU3LpfKGJqbl5jTpifx0NaSr271UwNjqW5ZE07rMKT3vLFWPKwBzw7hfhrRjW2MshlVg2g3qd85EtvXKCAiTXB
FiMPVci6LmjDfAnARaz2EU9qizzw0JahHTPANAiOeoQ/file?_download_
id=399831976025986201628746329668162684155635121667676120608087983697&_log_download_success=1"
action="bypass" profile="dropbox_test" saasapp="dropbox" useractivity="dropbox-download"
subaction="monitor" tenantmatch="missed" activitycategory="activity-control" msg="CASB access was
allowed although it contained activity."
```

Customized SaaS application and user activity

In this example, a custom SaaS application is created (pc4) with a custom user action. When a user accesses pc4.qa.fortinet.com/virus, they are redirected to pc4.qa.fortinet.com/testweb/testweb.htm.

To configure a customized inline CASB profile in the GUI:

1. Configure the inline CASB profile:
   a. Go to *Security Profiles* > *Inline-CASB* and click *Create new*.
   b. Enter a *Name*, such as *custom_test*.
   c. In the *SaaS Applications* table, click *Create new*. The *Create SaaS Application Rules* pane opens.
   d. Click *Create new* to create a custom entry. The *Create Inline-CASB SaaS Application* pane opens.
   e. Enter the *Name* (*pc4*) and *Domains* (*pc4.qa.fortinet.com*), then click *OK*.



f. Search *pc4* and select *pc4*, then click *Next*.

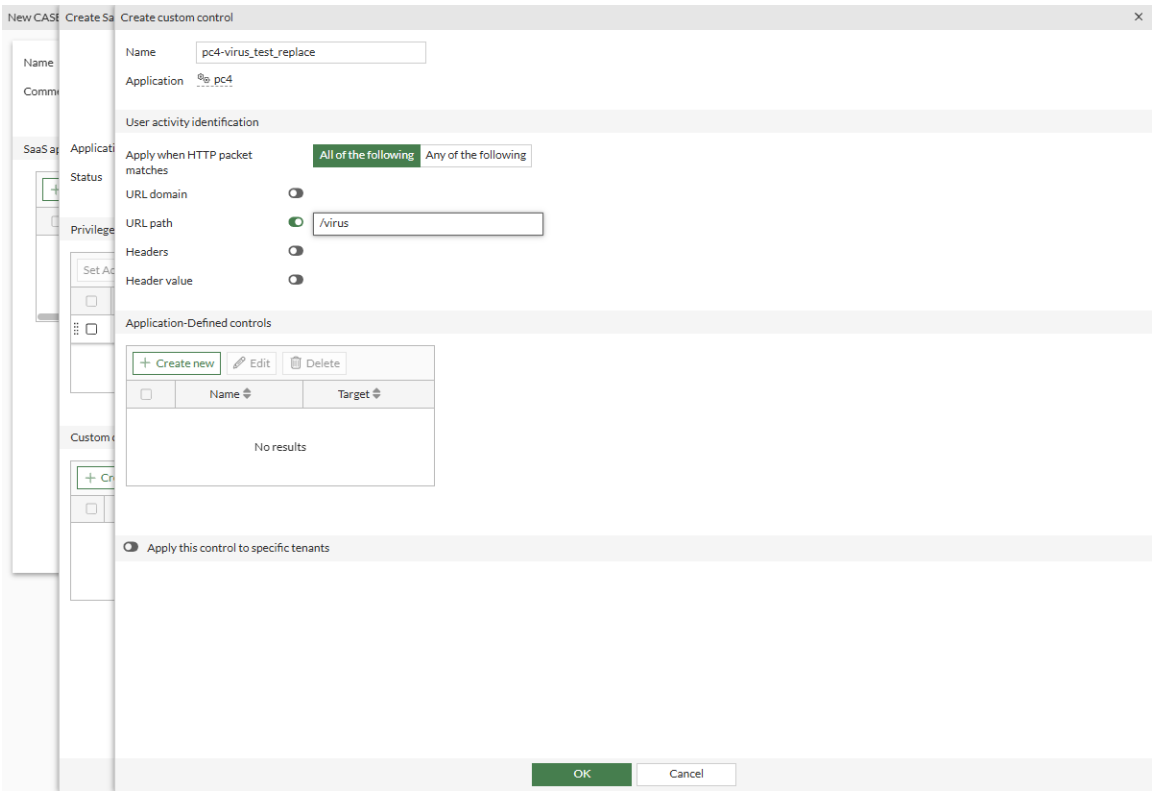
g. Configure the custom control and action:

i. In the *Custom Controls* table, *Create new*. The *Create Custom Control* pane opens.

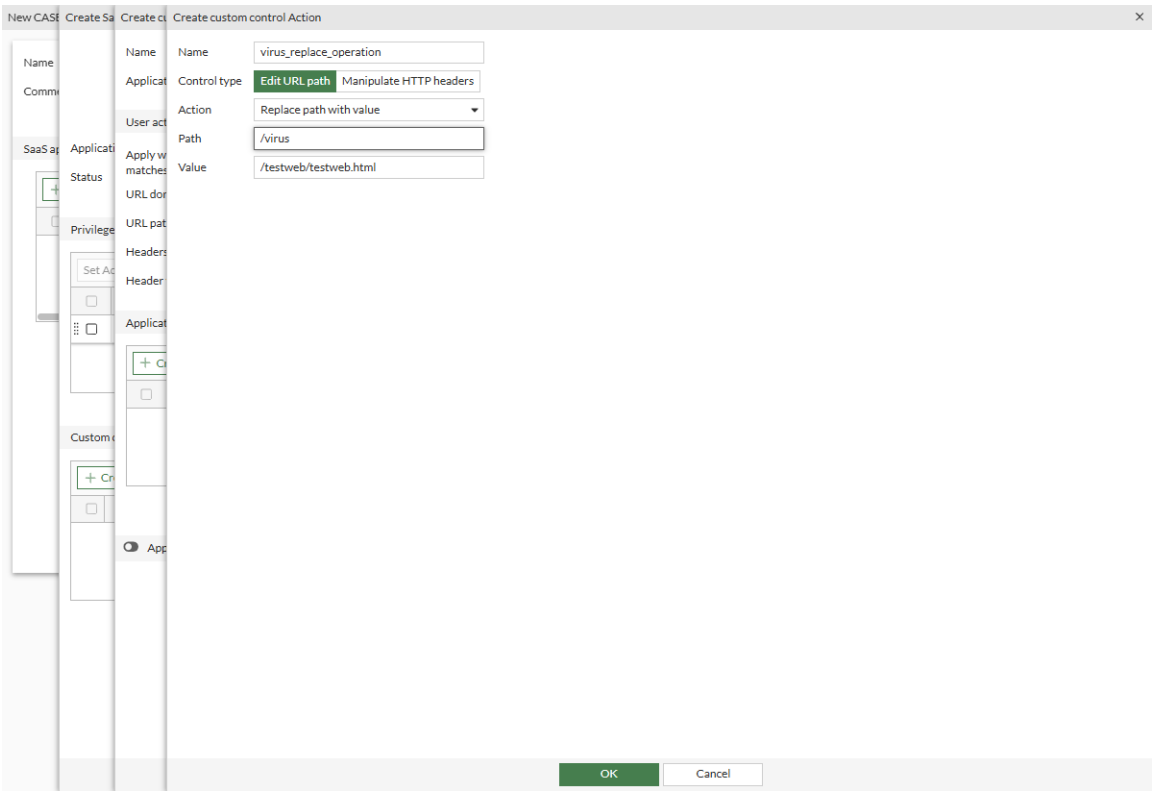
ii. Enter a *Name*, such as *pc4-virus_test_replace*.

iii. Set *Apply when HTTP packet matches* to *All of the following*.

iv. Enable *URL path* and enter */virus*.



v. In the *Application-Defined Controls* table, *Create new*. The *Create Custom Control Action* pane opens.
vi. Enter a *Name*, such as *virus_replace_operation*.
vii. Set the *Control Type* to *Edit URL path*.
viii. Set the *Action* to *Replace path with value*.
ix. Set the *Path* to */virus*.
x. Set the *Value* to */testweb/testweb.html*.



xi. Click *OK* to save the custom action.

xii. Click *OK* to save the custom control.

h. Click *OK* to save the application rule.

i. Click *OK* to save the inline CASB profile.

2. Configure the firewall policy:
   a. Go to *Policy & Objects* > *Firewall Policy*. Edit an existing policy, or create a new one.
   b. Set the *Inspection Mode* to *Proxy-based*.
   c. In the *Security Profiles* section, enable *Inline-CASB* and select the *custom_test* profile.
   d. Set the *SSL Inspection* profile to one that uses deep inspection.
   e. Configure the other settings as needed.
   f. Click *OK*.

To configure a customized inline CASB profile in the CLI:

1. Configure the CASB SaaS application:

```
config casb saas-application
    edit "pc4"
        set domains "pc4.qa.fortinet.com"
    next
end
```

2. Configure the CASB user activity:

```
config casb user-activity
    edit "pc4-virus_test_replace"
        set application "pc4"
        set category other
        config match
            edit 1
                config rules
                    edit 1
                        set type path
                        set match-value "/virus"
                    next
                end
            next
        end
        config control-options
            edit "virus_replace_operation"
                config operations
                    edit "virus_replace_operation"
                        set target path
                        set action replace
                        set search-key "/virus"
                        set values "/testweb/testweb.html"
                    next
                end
            next
        end
    next
end
```

3. Configure the inline CASB profile:

```
config casb profile
    edit "custom_test"
        config saas-application
            edit "pc4"
                config custom-control
                    edit "pc4-virus_test_replace"
                        config option
                            edit "virus_replace_operation"
                            next
                        end
                    next
                end
            next
        end
    next
end
```

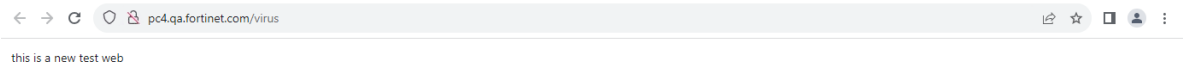
4. Configure the firewall policy:

```
config firewall policy
    edit 10
```

```
        set name "casb_test_custom"
        set srcintf "port1"
        set dstintf "port3"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "ssl"
        set casb-profile "custom_test"
        set nat enable
    next
end
```

To test the configuration:

1. Open a browser and go to pc4.qa.fortinet.com/virus.
2. Access is redirected to pc4.qa.fortinet.com/testweb/testweb.htm.



Sample log:

```
1: date=2023-08-21 time=08:31:06 eventtime=1692631866382806917 tz="-0700" logid="2500010001"
type="utm" subtype="casb" eventtype="casb" level="information" vd="vdom1" msg="CASB access was
allowed although it contained activity." policyid=10 sessionid=3139 srcip=10.1.100.195
dstip=172.16.200.44 srcport=56774 dstport=80 srcintf="port1" srcintfrole="undefined"
dstintf="port3" dstintfrole="undefined" proto=6
url="http://pc4.qa.fortinet.com/testweb/testweb.html" action="bypass" profile="custom_test"
saasapp="pc4" useractivity="pc4-virus_test_replace" activitycategory="other"
```

Support control factors in exchanged JSON data for custom SaaS applications

The inline CASB security profile supports control factors, such as tenant information in JSON data exchanged between a web browser and a custom SaaS application. For example, for some custom SaaS applications, the URL does not change to reflect the type or identity of the user or organization when logged in as such tenant information is exchanged using JSON data instead of through changes in the URL. JSON data can be extracted using JQ filters (see https://jqlang.org/manual/v1.5/#basic-filters).

In this example, the administrator wants to distinguish between two types of users in different categories, and assign different actions accordingly. The following two SaaS application requests will be made:

```
curl -k https://httpbin.org/headers -H "Partners-Data: {'company': 'contractorsA'}"
curl -k https://httpbin.org/headers -H "Partners-Data: {'company': 'contractorsB'}"
```

Sending the CURL request for contractorsA as:

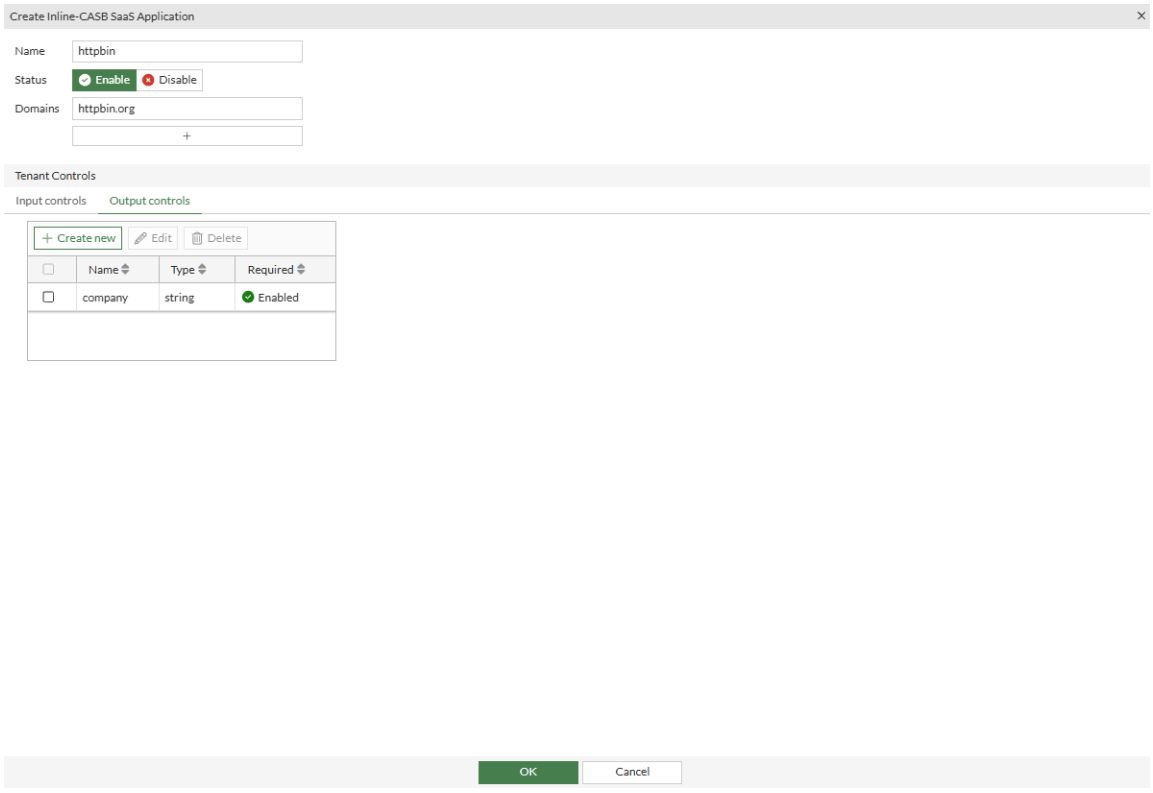
```
GET /headers HTTP/2.0
Host: httpbin.org
user-agent: curl/7.68.0
accept: */*
partners-data: {'company': 'contractorsA'}
```

Traffic is classified based on the tenant information *company* in the HTTP request header *Partners-Data*. Traffic from *contractorsA* is authorized and monitored, and traffic from *contractorsB* is blocked.

This information is transported in a JSON structure within the request, so the JQ tenant extraction feature is used.

To configure a CASB SaaS application in the GUI:

1. Go to *Security Profiles* > *Inline-CASB* and select the *SaaS Application* tab.
2. Click *Create New*.
3. Enter a *Name*, such as *httpbin*.
4. Enter the *Domains*, such as *httpbin.org*.
5. In *Tenant Controls*, select the *Output controls* tab and then click *Create New*.
6. Enter the *Attribute*, such as *company*, then click *OK*.

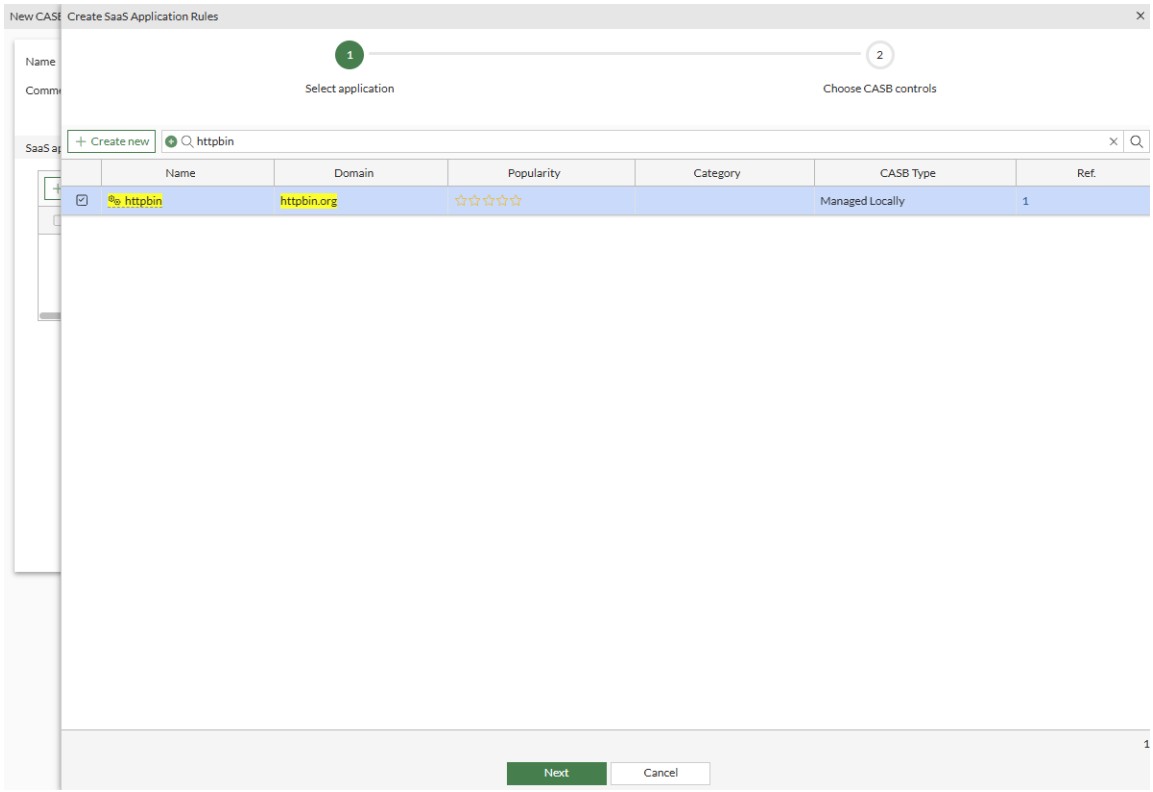


7. Click *OK*.

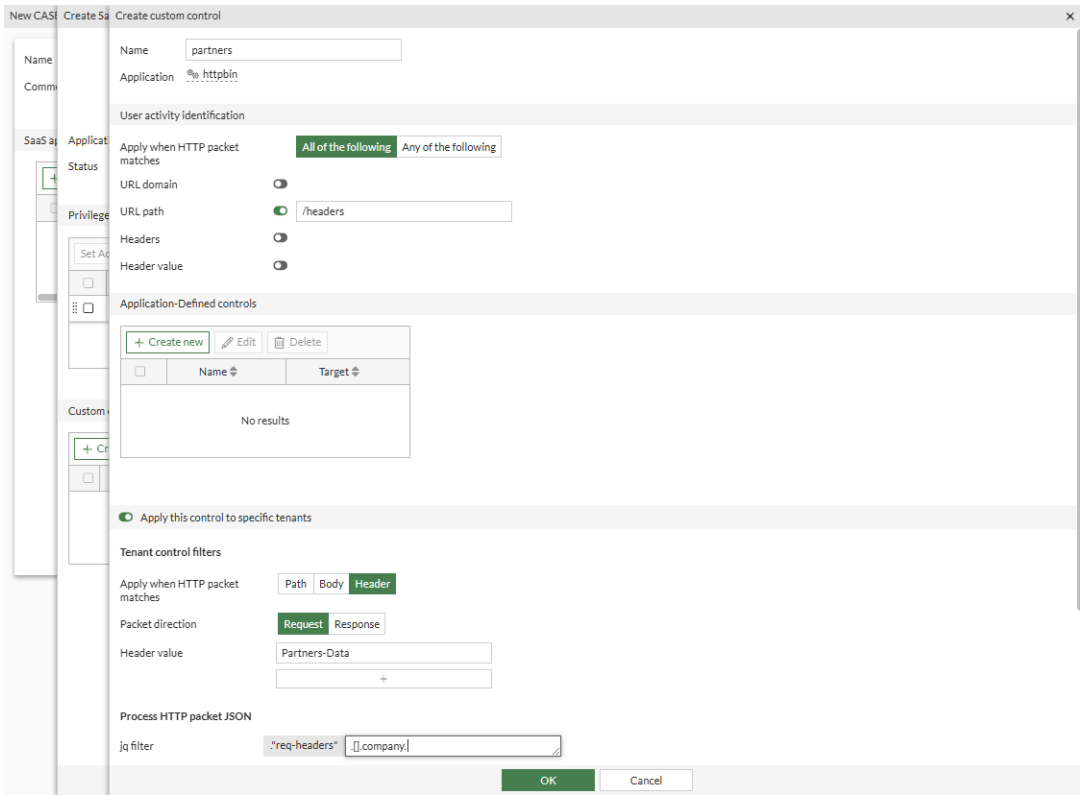
To configure a CASB profile in the GUI:

1. Go to *Security Profiles* > *Inline-CASB* and, on the *Profile* tab, click *Create New*.
2. Enter a *Name*, such as *casbProfile*.

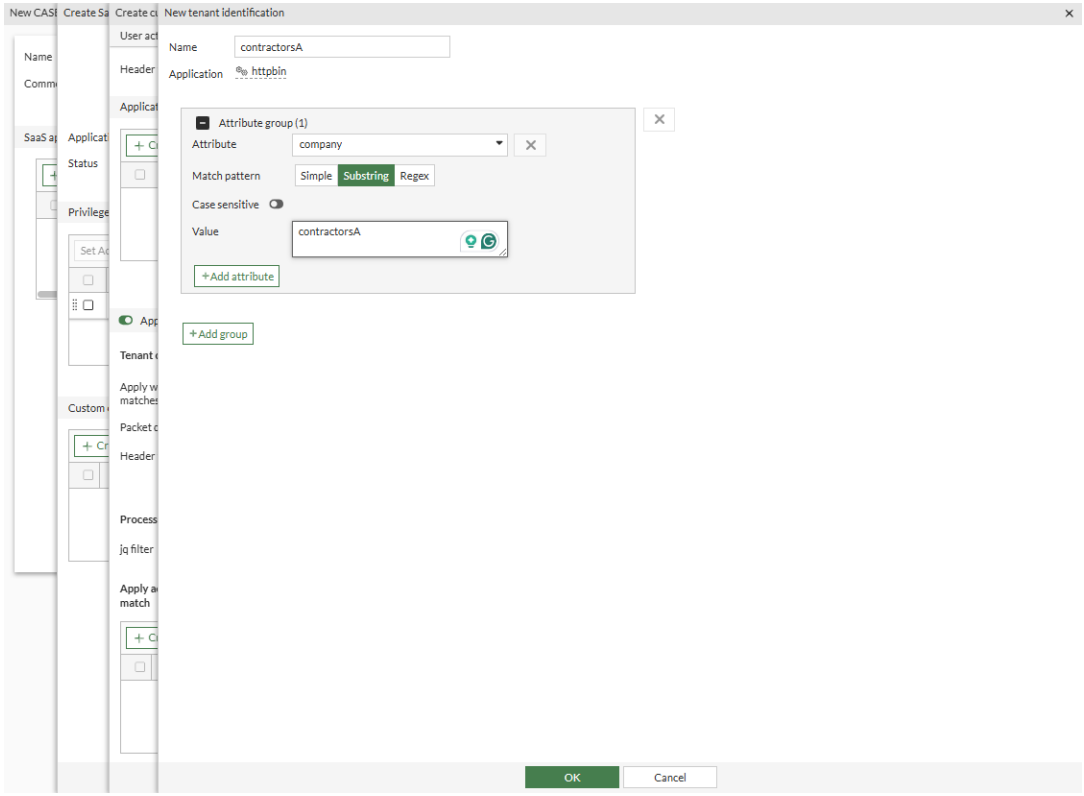
3. In the *SaaS applications* table, click *Create New*.
4. Search and select the *httpbin* custom application, then click *Next*.



5. In the *Custom controls* table, click *Create New*.
6. Enter a *Name*, such as *partners*.
7. Enable *URL path* and set it to */headers*. This how the user activity (UA) will be identified. The administrator needs to find a way to recognize this traffic, which can be done using the path, a special header/value pair, or the URL domain name.
8. Enable *Apply this control to specific tenants*.
9. Set *Apply when HTTP packet matches* to *Header*, because in this case the JSON is in the HTTP header.
10. Set *Header value* to *Partners-Data*.
11. Set *jq filter* to *.[].company*. This is the tenant extraction field; it tells the FortiGate to extract the *company* field from the JSON structure that is in the HTTP request header.



12. In the *Apply action by attribute match* table click *Create New*.
13. Configure the tenant information for contractorsA:

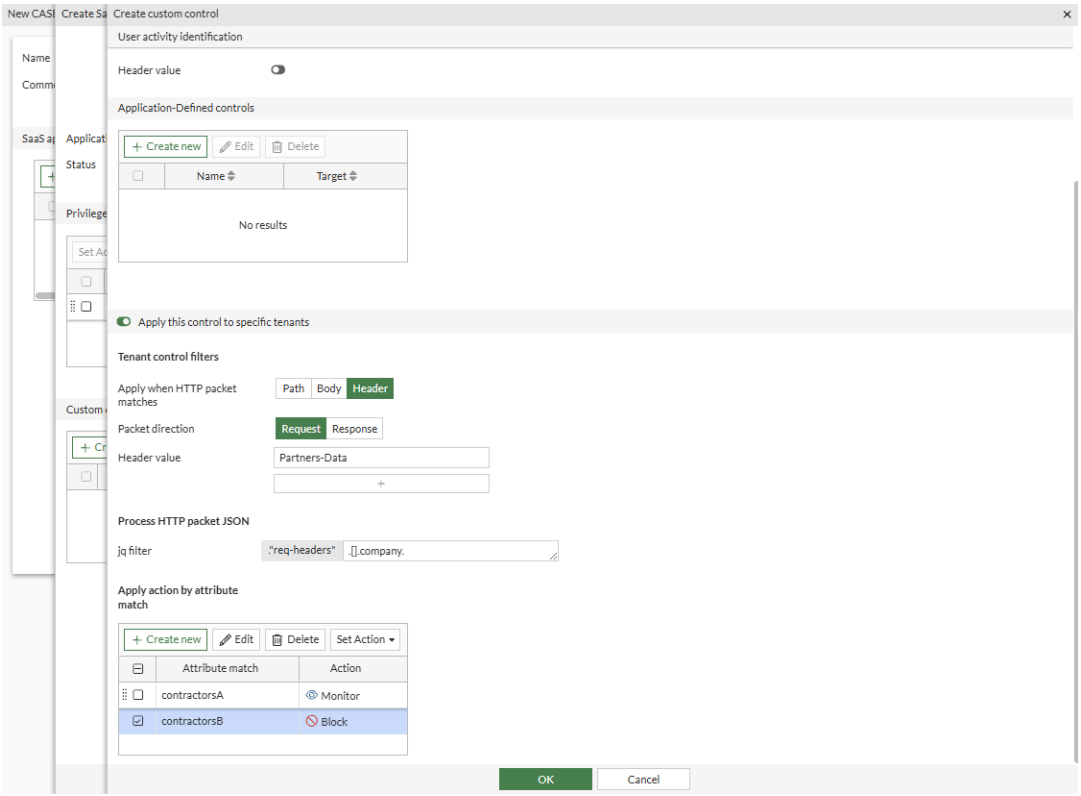


Field	Value
Name	contractorsA
Attribute	company
Match pattern	substring
Value	contractorsA

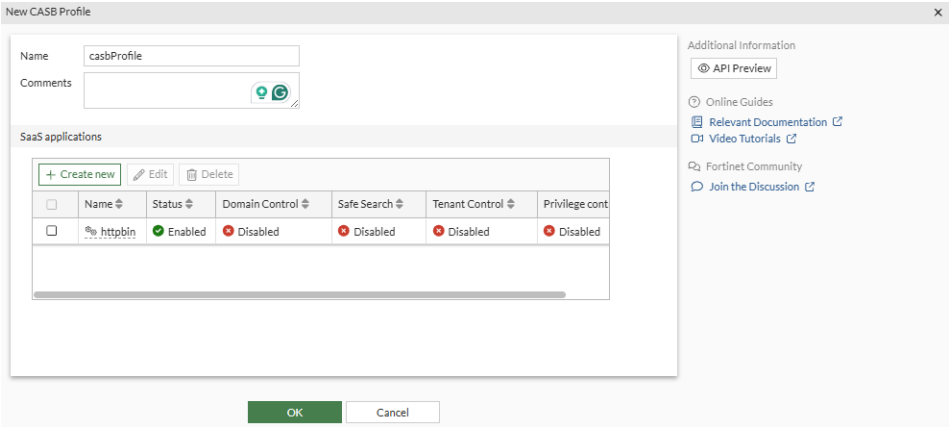
14. Click *OK*, then click *Create New* again to configure the tenant information for contractorsB:

Field	Value
Name	contractorsB
Attribute	company
Match pattern	substring
Value	contractorsB

15. Click *OK*.
16. In the *Apply action by attribute match* table, select *contractorsB* and click *Set Action* > *Block*.



17. Click *OK* to save the custom control.
18. Click *OK* to save the SaaS application rules.



19. Click *OK* to save the CASB profile.

To configure a CASB SaaS application in the CLI:

```
config casb saas-application
    edit "httpbin"
        set domains "httpbin.org"
        config output-attributes
            edit "company"
            next
        end
    next
end
```

To configure a CASB profile in the CLI:

1. Configure CASB user activity:

```
config casb user-activity
    edit "httpbin-partners"
        set application "httpbin"
        set category other
        config match
            edit 1
                config rules
                    edit 1
                        set type path
                        set match-value "/headers"
                    next
                end
                config tenant-extraction
                    set status enable
                    set jq ".\"req-headers\".[].company"
                    config filters
                        edit 1
                            set header-name "Partners-Data"
                        next
                    end
```

```
                end
            next
        end
    next
end
```

2. Configure CASB attribute match rules:

```
config casb attribute-match
    edit "httpbin-contractorsA"
        set application "httpbin"
        config match
            edit 1
                config rule
                    edit 1
                        set attribute "company"
                        set match-pattern substr
                        set match-value "contractorsA"
                    next
                end
            next
        end
    next
    edit "httpbin-contractorsB"
        set application "httpbin"
        config match
            edit 1
                config rule
                    edit 1
                        set attribute "company"
                        set match-pattern substr
                        set match-value "contractorsB"
                    next
                end
            next
        end
    next
end
```

3. Configure a CASB profile:

```
config casb profile
    edit "casbProfile"
        config saas-application
            edit "httpbin"
                config custom-control
                    edit "httpbin-partners"
                        config attribute-filter
                            edit 1
                                set attribute-match "httpbin-contractorsA"
                                set action monitor
                            next
```

```
                            edit 2
                                set attribute-match "httpbin-contractorsB"
                                set action block
                            next
                        end
                    next
                end
            next
        end
    next
end
```

To test the profile:

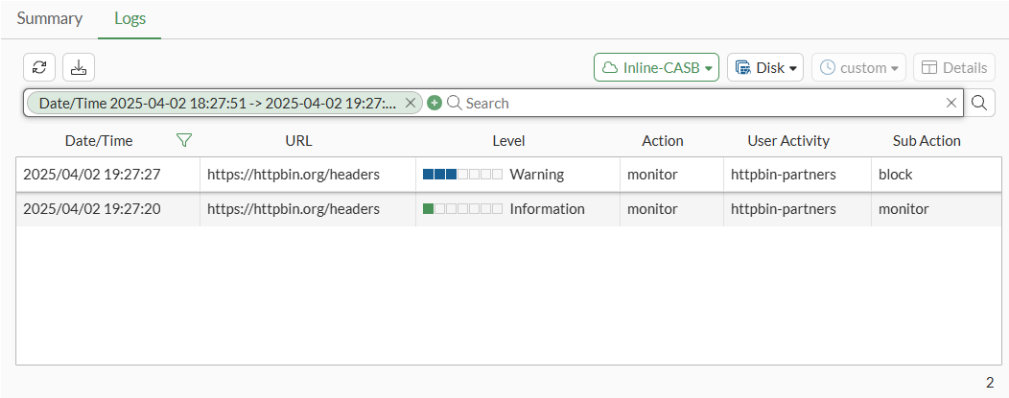
1. Make the following two SaaS application requests:

```
curl -k https://httpbin.org/headers -H "Partners-Data: {'company': 'contractorsA'}"
curl -k https://httpbin.org/headers -H "Partners-Data: {'company': 'contractorsB'}"
```

2. Check the logs to see that the SaaS application request for contractorsA is passed and a log is generated:

```
1: date=2025-03-22 time=00:25:19 eventtime=1742628319441656177 tz="-0700" logid="2500010002"
type="utm" subtype="casb" eventtype="casb" level="information" vd="vdom1" policyid=1
poluuid="e0a45778-05e0-51f0-d77d-e4e8a02811e2" policytype="policy" sessionid=7535
srcip=10.1.100.13 dstip=54.236.151.211 srcport=46476 dstport=443 srcintf="lan"
srcintfrole="undefined" srcuuid="bcbee936-05e0-51f0-5712-f0e95616dde0" dstintf="mgmt"
dstintfrole="lan" dstuuid="bcbee936-05e0-51f0-5712-f0e95616dde0" proto=6
url="https://httpbin.org/headers" action="monitor" profile="casbProfile" saasapp="httpbin"
useractivity="httpbin-partners" subaction="monitor" tenantmatch="matched"
activitycategory="other" msg="CASB access was monitored because it contained activity."
```

3. In the GUI, go to *Log & Report* > *Security Events* and view the *Inline-CASB* event logs.
4. Confirm that the SaaS application request for contractorsB is blocked, as its sub-action is set to *block*.



Intrusion prevention

Intrusion Prevention System (IPS) detects network attacks and prevents threats from compromising the network, including protected devices. IPS can be in the form of a standalone appliance, or part of the feature set of a Next Generation Firewall (NGFW), such as FortiGate. IPS utilizes signatures, protocol decoders, heuristics (or behavioral monitoring), threat intelligence (such as FortiGuard Labs), and advanced threat detection in order to prevent exploitation of known and unknown zero-day threats. FortiGate IPS is even capable of performing deep packet inspection to scan encrypted payloads in order to detect and prevent threats from attackers.

When implementing IPS profiles, Fortinet recommends tuning the sensor to reflect the environment that it is protecting. This involves selecting only signatures that match the services and devices that it is meant to protect, and adjusting thresholds for selected anomalies to suit your environment.

Networks and devices are often exploited through vulnerabilities. Software vulnerabilities are one such example where a bug or inherent weakness in the code provides attackers an opportunity to gain access to the software. More severe vulnerabilities allow unauthorized access, data loss, and execution of malicious code. Exploitation of these vulnerabilities can cause damage to the machine and infect others. While the best solution is to patch vulnerabilities as soon as patches are available, IPS signatures offer a solution to detect and block exploitation of many vulnerabilities before they enter the network.

IPS signatures

In order to download updated IPS definitions, at least 1 policy with a security profile that has IPS scanning must be enabled.

Fortinet's solution combines industry-leading threat intelligence from FortiGuard Labs with the FortiGate NGFW to identify the latest threats and prevent them from entering your network. IPS signatures are one such method for delivering the latest protection. FortiGuard Labs uses AI and Machine Learning (ML) to analyze billions of events every day. The FortiGuard Labs research team also proactively performs threat research to discover new vulnerabilities and exploitation, and produces signatures to identify such threats. These IPS signatures are delivered to each FortiGate daily, so that the IPS engine is armed with the latest databases to match the latest threats.

IPS sensors

A FortiGate IPS sensor is a collection of IPS signatures and filters that define the scope of what the IPS engine will scan when the IPS sensor is applied. An IPS sensor can have multiple sets of signatures and/or filters. A set of IPS signatures consists of manually selected signatures, while a set of IPS filters consists of filters based on signature attributes like target, severity, protocol, OS, and application. Each signature has predefined attributes and an action, such as block, allow, monitor (pass), quarantine, and reset. It is also possible to create custom IPS signatures to apply to an IPS sensor.

From the *Security Profiles* > *Intrusion Prevention* pane, you can create new IPS sensors and view a list of predefined sensors.

FortiOS includes the following predefined IPS sensors with associated predefined signatures:

Predefined IPS sensors	Description
all_default	Filters all predefined signatures, and sets action to the signature's default action.
all_default_pass	Filters all predefined signatures, and sets action to pass/monitor.
default	Filters all predefined signatures with severity of Critical/High/Medium. Sets action to signature's default action.
high_security	Filters all predefined signatures with severity of Critical/High/Medium, and sets action to Block. For Low severity signatures, sets action to signature's default action.
protect_client	Protects against client-side vulnerabilities by filtering on `Target=Client`. Sets action to signature's default action.
protect_email_server	Protects against email server-side vulnerabilities by filtering on `Target=Server` and `Protocol=IMAP`, `POP3` or `SMTP`. Sets action to signature's default action.
protect_http_server	Protects against HTTP server-side vulnerabilities by filtering on `Target=Server` and `Protocol=HTTP`. Sets action to signature's default action.
wifi-default	Filters all predefined signatures with severity of Critical/High/Medium. Sets action to signature's default action. Used in profile for offloading WiFi traffic.

New signatures

When new vulnerabilities are discovered and the FortiGuard team creates signatures to match them, the priority is to match the malicious traffic and release the signature as quickly as possible. As a result, the signatures are more broad at the start and become refined over time, incidentally causing some false positives. For this reason, the signatures are often released with an action of pass.

You must decide on the tradeoff between protection and usability with potential false positives. The high_security IPS sensor (or any custom sensor which overwrites the default action of Medium/High/Critical) should be used when false positives are acceptable to provide a higher level of security.

DDoS attacks

Besides protecting against threats and exploitation of vulnerabilities, the IPS engine is also responsible for mitigating Denial of Service (DoS) attacks where attackers attempt to bring a service down by flooding the target with traffic from distributed systems. Using anomaly-based defense, FortiGate can detect a variety of L3 and L4 anomalies and take action against these attacks. This can be configured under IPv4 and IPv6 DoS Policies, which is discussed in detail under DoS policy on page 1901.

This section contains the following topics:

- Signature-based defense on page 2480
- Configuring an IPS sensor on page 2483
- IPS configuration options on page 2486
- IPS signature filter options on page 2498
- SCTP filtering capabilities on page 2492

- Diameter protocol inspection on page 2494

This section also provides the following examples about IPS sensors:

- IPS with botnet C&C IP blocking on page 2502
- IPS signatures for the operational technology security service on page 2506
- IPS sensor for IEC 61850 MMS protocol on page 2508
- IPS Modbus TCP decoder on page 2509
- AI and ML-based IPS detection on page 2511
- Support Ethernet layer protocols in the IPS engine on page 2514

Signature-based defense

Signature-based defense is used against known attacks or vulnerability exploits. These often involve an attacker attempting to gain access to your network. The attacker must communicate with the host in an attempt to gain access, and this communication includes commands or sequences of commands and variables. The IPS signatures include these command sequences, allowing the FortiGate unit to detect and stop the attack.

This section describes the following components used in signature-based defense:

- IPS signatures on page 2480
- Protocol decoders on page 2480
- IPS engine on page 2481
- IPS sensors on page 2481
- IPS filters on page 2482
- Custom and predefined signature entries on page 2482
- Policies on page 2483

IPS signatures

IPS signatures are the basis of signature-based intrusion prevention. Every attack can be reduced to a particular string of commands or a sequence of commands and variables. Signatures include this information, and FortiGate uses the information to detect and stop attacks.

Signatures also include characteristics about the attack they describe. These characteristics include the network protocol associated with the attack, the vulnerable operating system, and the vulnerable application.

To view the complete list of signatures, go to *Security Profiles* > *IPS Signatures*. The list of signatures includes predefined and custom signatures. You can hover over the name of the IPS signature to display a pop-up window that includes an ID number. You can click the ID number to display the FortiGuard page.

Protocol decoders

Before examining network traffic for attacks, the IPS engine uses protocol decoders to identify each protocol appearing in the traffic. Attacks are protocol-specific, so your FortiGate unit conserves resources by looking for attacks only in the protocols used to transmit them. For example, the FortiGate unit will only examine HTTP traffic for the presence of a signature describing an HTTP attack.

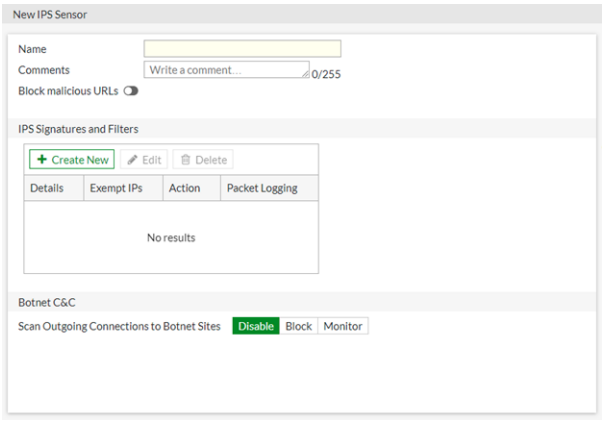
IPS engine

Once the protocol decoders separate the network traffic by protocol, the IPS engine examines the network traffic for the attack signatures by using IPS sensors.

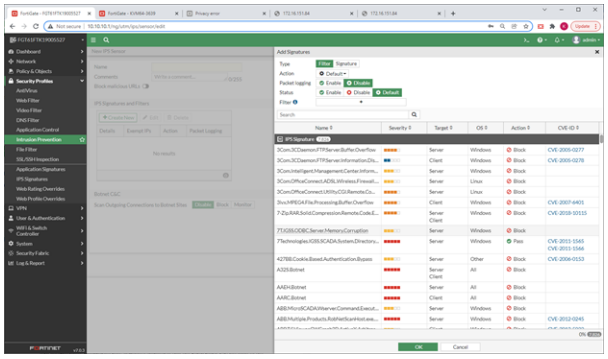
IPS sensors

The IPS engine does not examine network traffic for all signatures. The IPS engine examines network traffic for signatures specified in IPS sensors. You must first create an IPS sensor, and then you can specify what signatures the IPS sensor will use. You can add individual signatures to IPS sensors, or you can add filters to IPS sensors, and the filters automatically include the applicable signatures.

To view IPS sensors, go to *Security Profiles* > *Intrusion Prevention*. To create a new sensor, click *Create New*.



An IPS sensor is composed of IPS signatures and filters. Under *IPS Signatures and Filters*, click *Create New* to create a set of IPS signatures or a set of IPS filters.



You can create IPS sensors for specific types of traffic, and then select the IPS sensors in firewall policies designed to handle the same type of traffic. For example, you can specify all of the web-server related signatures in an IPS sensor, and select the IPS sensor in a firewall policy that controls all traffic to and from a web server that is protected by the FortiGate unit.

The FortiGuard Service periodically adds new predefined signatures to counter new threats. New predefined signatures are automatically included in IPS sensors that are configured to use filters when the new signatures match existing filter specifications. For example, if you have an IPS sensor with a filter that includes all

signatures for the Windows operating system, your filter will automatically incorporate new Windows signatures that the FortiGuard Service adds to the database.

IPS signature and filter entries are checked from top down. When a signature is found in a set of signatures or filters, the action defined for the signature is taken.

IPS filters

IPS sensors can contain one or more IPS filters. A filter is a collection of signature attributes that you specify. The signatures that have all of the attributes specified in a filter are included in the IPS filter.

Following are the attribute groups:

- Target
- Severity
- Protocol
- OS
- Application

Starting in FortiOS 6.4.2, you can also filter by CVE ID or CVE pattern by using the CLI. See FortiOS 6.4 New Features > IPS signature filter options.

When selecting multiple attributes within the same group, the selections are combined by using a logical `OR`. When selecting multiple attributes between attribute groups, each attribute group is combined by using a logical `AND`.

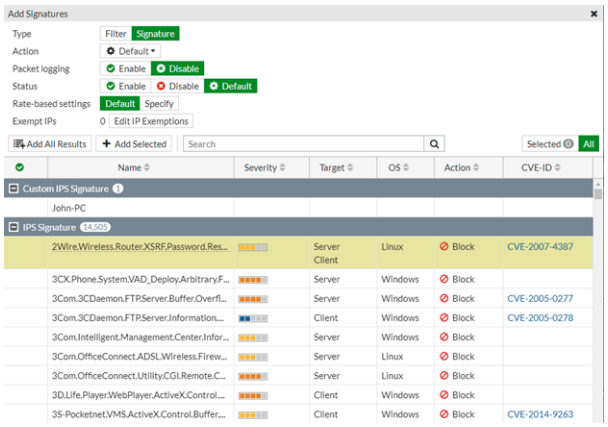
Once you select filters in the GUI, the filtered list of IPS signatures are displayed. Adjust your filters accordingly to construct a suitable list for your needs.

For example, if your FortiGate unit protects a Linux server running the Apache web server software, you could create a new filter to protect it. By setting *OS* filter attribute to *Linux*, and the filter attribute *Application* to *Apache*, the filter will include only the signatures that apply to both Linux and Apache. If you wanted to scan for all the Linux signatures and all the Apache signatures, you would create two filters, one for each.

To view the filters in an IPS sensor, go to *Security Profiles* > *Intrusion Prevention*, select the IPS sensor, and click *Edit*.

Custom and predefined signature entries

Signature entries allow you to add individual, custom or predefined IPS signatures to an IPS sensor. If you need only one signature, or you want to manually select multiple signatures that don't fall into the criteria for an IPS filter, adding a signature entry to an IPS sensor is the easiest way. Signature entries are also the only way to include custom signatures in an IPS sensor.



To select an individual signature, click a signature, and select *Add Selected*. The signature moves to the *Selected* list.

To select multiple signatures, use the *Search* bar to perform a keyword search, and then click *Add All Results* to move all entries to the *Selected* list.

Overriding the default action

Each IPS signature comes with a default action such as *Block* and *Pass*. In some scenarios, you may want to override this action. You can override a set of IPS filter or signatures. By default, a set of IPS filter or signatures has an action of *Default*, which applies a signature's default action when the signature is matched. By changing the action, you can override the setting for all signatures within the filter or signature set.

Policies

You must select an IPS sensor in a security policy or an interface policy to apply the IPS sensor to traffic. An IPS sensor that it not selected in a policy is not applied to network traffic.

Configuring an IPS sensor

You can configure IPS sensors to be used in policies in the GUI.

To configure an IPS sensor:

1. Go to *Security Profiles* > *Intrusion Prevention*.
2. Click *Create New*.
3. Configure the following settings:

Name	Enter a unique name for the sensor.
Comments	Enter a comment (optional).
Block malicious URLS	Enable to block malicious URLs based on a local malicious URL database on the FortiGate to assist in the detection of drive-by exploits. See Malicious URL database for drive-by exploits detection on page 2487.

IPS Signature and Filters		Select a signature or filter to assign to the sensor. See Configuring signatures and filters on page 2484.
Botnet C&C		
	Scan Outgoing Connections to Botnet Sites	Define the botnet scanning across traffic that matches the policy: • *Disable*: Do not scan connections to botnet servers. • *Block*: Block connections to botnet servers. • *Monitor*: Log connections to botnet servers. See IPS with botnet C&C IP blocking on page 2502.

4. Click *OK*.

For information on configuring IPS sensors in the CLI, see IPS configuration options on page 2486.

Configuring signatures and filters

Signatures and filters can be configured and added to IPS sensors. A filter is a collection of signature attributes. Any signatures that meet all of the attributes specified in a filter are automatically included in the IPS sensor. See IPS signature filter options on page 2498.

To configure a Signature entry of type Filter:

1. Go to *Security Profiles* > *Intrusion Prevention*.
2. Click *Create New*.
3. Configure the IPS sensor settings.
4. In *IPS Signatures and Filters*, click *Create New*. The *Add Signatures* pane is displayed.
5. Configure the settings as follows:

Type	Select *Filter*.
Action	Click the dropdown menu and select the action when a signature is triggered: • *Allow*: Allow traffic to continue to its destination. • *Monitor*: Allow traffic to continue to its destination and log the activity. • *Block*: Drop traffic that matches the signature. • *Reset*: Reset the session whenever the signature is triggered. • *Default*: Use the default action of the signature. Search for the signature in the *IPS Signature* pane to view the default *Action*. • *Quarantine*: Block the matching traffic. Enable packet logging. Quarantine the attacker.
Packet logging	Enable packet logging to save a copy of the packets when they match the signature. Packet copies can be analyzed later.

	Packet logging is not supported on all FortiGate devices. FortiAnalyzer logging or a hard disk are required to support this feature; see the Feature Platform Matrix.
Status	Define the signature status: • *Enable*: Enable the signature. • *Disable*: Disable the signature. • *Default*: Use the default status of the signature. Search for the signature in the *IPS Signature* pane to view the default *Status*.
Filter	Select the *+* to open the *Select Entries* field and select filter entries. There are different entry categories: • *Target*: Refers to the type of device targeted by the attack. • *Severity*: Refers to the level of the threat posed by the attack. • *Protocol*: Refers to the protocol that is the vector for the attack. • *OS*: Refers to the Operating System affected by the attack. • *Application*: Refers to the application affected by the attack.

6. Select one or more signatures from the *IPS Signatures* pane.
7. Click *OK*. The signature is added to the IPS sensor.
8. Click *OK*.

Individual signatures, custom or predefined IPS signatures can be selected for an IPS sensor. If you need only one signature, or you want to manually select multiple signatures that don't fall into the criteria for an IPS filter, adding a signature entry to an IPS sensor is the easiest way.

To configure a Signature entry of type Signature:

1. Go to *Security Profiles > Intrusion Prevention*.
2. Click *Create New*.
3. Configure the IPS sensor settings.
4. In *IPS Signatures and Filters*, click *Create New*. The *Add Signatures* pane is displayed.
5. Configure the settings as follows:

Type	Select *Signature*.
Action	Click the dropdown menu and select the action when a signature is triggered: • *Allow*: Allow traffic to continue to its destination. • *Monitor*: Allow traffic to continue to its destination and log the activity. • *Block*: Drop traffic that matches the signature. • *Reset*: Reset the session whenever the signature is triggered. • *Default*: Use the default action of the signature. Search for the signature in the *IPS Signature* pane to view the default *Action*. • *Quarantine*: Block the matching traffic. Enable packet logging. Quarantine the attacker.

Packet Logging	Enable packet logging to save a copy of the packets when they match the signature. Packet copies can be analyzed later. Packet logging is not supported on all FortiGate devices. FortiAnalyzer logging or a hard disk are required to support this feature; see the Feature Platform Matrix.
Status	Define the signature status: • *Enable*: Enable the signature. • *Disable*: Disable the signature. • *Default*: Use the default status of the signature. Search for the signature in the *IPS Signature* pane to view the default *Status*.
Rate-based settings	
Default	Use the default rate-based settings.
Specify	Specify the rate-based settings: • *Threshold*: Enter the threshold. See IPS signature rate count threshold on page 2487. • *Duration (seconds)*: Enter the duration in seconds. • *Track By*: Select the tracking method as *Any*, *Source IP*, or *Destination IP*.
Exempt IPs	Add IP addresses that are exempt from the signature rules. Click *Edit IP Exemptions* and click *Create New*. Edit the *Source IP/Netmask* and the *Destination IP/Netmask* to define the IP address for exemption. Click *OK* to add it to *Exempt IPs*.

6. Select one or more signatures from the *IPS Signatures* pane.
7. Click *OK*. The signature is added to the IPS sensor.
8. Click *OK*.

IPS configuration options

Besides configuring an IPS filter or selecting IPS signatures for an IPS sensor, you can configure additional IPS options for each sensor or globally for all sensors. This topic introduces the following available configuration options:

- Malicious URL database for drive-by exploits detection on page 2487
- IPS signature rate count threshold on page 2487
- Botnet C&C on page 2488
- Hardware acceleration for flow-based security profiles (NTurbo and IPSA) on page 2488
- Extended IPS database on page 2489
- IPS engine-count on page 2490
- OT threat definitions on page 2490
- Fail-open on page 2490
- IPS buffer size on page 2491
- Session count accuracy on page 2491

- Protocol decoders on page 2491

To configure IPS sensors, signatures, and filters in the GUI, see Configuring an IPS sensor on page 2483.

Malicious URL database for drive-by exploits detection

This feature uses a local malicious URL database on the FortiGate to assist in detection of drive-by exploits, such as adware that allows automatic downloading of a malicious file when a page loads without the user's detection. The database contains all malicious URLs active in the last one month, and all drive-by exploit URLs active in the last three months. The number of URLs controlled are in the one million range.

This feature can be enabled from an IPS sensor in the GUI by going to *Security Profiles* > *Intrusion Prevention* and editing or creating an IPS Sensor, then enabling *Block malicious URLs*. See Configuring an IPS sensor on page 2483.

To enable the blocking of malicious URLs in the CLI:

```
config ips sensor
    edit <profile>
        set block-malicious-url {enable | disable}
    next
end
```

Blocking malicious URLs is not supported on some FortiGate models, such as FortiGate 51E, 50E, or 30E.

IPS signature rate count threshold

You can use the IPS signature rate-based settings to specify a rate count threshold that must be met before the signature is triggered. A rate count threshold provides a more controlled recording of attack activity. For example, if multiple login attempts produce a failed result over a short period of time, then an alert would be sent and traffic might be blocked, which is a more manageable response than sending an alert every time a login fails.

This can be configured from the GUI by going to *Security Profiles* > *Intrusion Prevention*. Create or edit an IPS sensor. Within the sensor, edit the IPS signatures and filters. Only IPS signatures have the rate-based settings option. IPS filters do not. See Configuring an IPS sensor on page 2483.

Some settings are only available in the CLI.

To configure the IPS signature rate-based settings in the CLI:

```
config ips sensor
    edit <sensor>
        config entries
            edit <filter ID number>
```

```
                set rule <ids>
                set rate-count <integer>
                set rate-duration <integer>
                set rate-mode {continuous | periodical}
                set rate-track {none | src-ip | dest-ip | dhcp-client-mac  | dns-domain}
            next
        end
    next
end
```

`rule <ids>`	The predefined or custom IPS signatures to add to the sensor.
`rate-count <integer>`	The count of the rate (0 - 65535, default = 0). The rate-count must be configured before the other rate settings can be set.
`rate-duration <integer>`	Duration of the rate, in seconds (0 - 65535, default = 60)
`rate-mode {continuous | periodical}`	How the count threshold is met. • `continuous`: If the action is set to block, the action is engaged as soon as the rate-count is reached. For example, if the count is 10, the traffic would be blocked as soon as the signature is triggered 10 times. This is the default. • `periodical`: The FortiGate allows up to the value of the rate-count incidents where the signature is triggered during the rate-duration. For example, if the rate count is 100 and the duration is 60, the signature would need to be triggered 100 times in 60 seconds for the action to be engaged.
`rate-track {none | src-ip | dest-ip | dhcp-client-mac | dns-domain}`	Track one of the protocol fields within the packet (default = none).

Botnet C&C

See IPS with botnet C&C IP blocking on page 2502 for information on configuring settings in the CLI.

Hardware acceleration for flow-based security profiles (NTurbo and IPSA)

Some FortiGate models support a feature call NTurbo that can offload flow-based firewall sessions to network processors. See also NTurbo offloads flow-based processing in the Hardware Acceleration Guide. For IPSA enhanced pattern matching, see IPSA offloads flow-based advanced pattern matching in the Hardware Acceleration Guide.

Some FortiGate models also support offloading enhanced pattern matching for flow-based security profiles to CP8 or CP9 content processors.

To configure NTurbo and IPSA:

```
config ips global
    set np-accel-mode {none | basic}
    set cp-accel-mode {none | basic | advanced}
end
```

If the `np-accel-mode` option is available, your FortiGate supports NTurbo. The `none` option disables NTurbo, and `basic` (the default) enables NTurbo.

If the `cp-accel-mode` option is available, your FortiGate supports IPSA. The `none` option disables IPSA, and `basic` enables basic IPSA, and `advanced` enables enhanced IPSA, which can offload more types of pattern matching than basic IPSA. The `advanced` option is only available on FortiGate models with two or more CP8 processors, or one or more CP9 processors.

Starting in 7.6.5, memory usage for FortiGate models with 2 GB or 4 GB of RAM is optimized by:

- Reducing the memory reserved for Network Processors (NPs)
- Setting NTurbo max frame size to 1500. Interfaces with higher MTU will not offload to NTurbo.

Extended IPS database

Some models have access to an extended IPS Database. Because the extended database may affect FortiGate performance, the extended database package may be disabled by default on some models, such as entry-level models.

You can only enable the extended IPS database by using the CLI.

To enable the extended IPS database:

```
config ips global
    set database extended
end
```

FortiGate models with the CP9 SPU receive the IPS full extended database, and the other physical FortiGate models receive a slim version of the extended database. The slim-extended DB is a smaller version of the full extended DB that contains top active IPS signatures. It is designed for customers who prefer performance.

Customers with non-CP9 SPU models need to upgrade to a CP9 SPU model (physical FortiGate) in order to get full IPS signature coverage. All FortiGate models 200 (E and F) and higher have a CP9 SPU.

See Determining the content processor in your FortiGate unit in the FortiOS Hardware Acceleration Guide to check if your device has a CP9 SPU.

FortiGate VMs with eight or more vCPUs can be configured to have a minimum of eight cores to be eligible to run the full extended database. Any FortiGate VM with less than eight cores will receive a slim version of the extended database.

IPS engine-count

FortiGate units with multiple processors can run one or more IPS engine concurrently. The `engine-count` CLI command allows you to specify how many IPS engines to use at the same time.

To specify the number of concurrent IPS engines running:

```
config ips global
    set engine-count <int>
end
```

The recommended and default setting is `0`, which allows the FortiGate unit to determine the optimum number of IPS engines.

OT threat definitions

Operational Technology (OT) threat definitions are defined to protect Industrial Control Systems (ICS), Operational Technology (OT) and SCADA systems, which are critical infrastructure used by manufacturing industries. An OT Security Service license is required to use this signature database. These signatures are excluded by default, but can be configured in the CLI.

Enabling the OT threat definitions may impact IPS performance, since this increases the number of signatures to scan. To optimize IPS performance, enable only IPS signature packages that are needed.

To configure OT signatures:

```
config ips global
    set exclude-signatures {none | ot}
end
```

Fail-open

A fail-open scenario is triggered when IPS raw socket buffer is full. Therefore IPS engine has no space in memory to create more sessions and needs to decide whether to drop the sessions or bypass the sessions without inspection.

To enable fail-open mode:

```
config ips global
    set fail-open {enable | disable}
end
```

The default setting is `disable`, so sessions are dropped by IPS engine when the system enters fail-open mode.

When enabled, the IPS engine fails open, and it affects all protocols inspected by FortiOS IPS protocol decoders, including but not limited to HTTP, HTTPS, FTP, SMTP, POP3, IMAP, and so on. When the IPS engine fails open, traffic continues to flow without IPS scanning.

Sessions offloaded to Nturbo do not support fail-open. When Nturbo data path is overloaded, traffic is dropped regardless of fail-open setting.

IPS buffer size

If system enters fail-open mode frequently, it is possible to increase the IPS socket buffer size to allow more data buffering, which reduces the chances of overloading the IPS engine. You can set the size of the IPS buffer.

To set the socket buffer size:

```
config ips global
    set socket-size <int>
end
```

The default socket size and maximum configurable value varies by model. In short, socket-size determines how much data the kernel passes to the IPS engine each time the engine samples packets.

Take caution when modifying the default value. If the socket-size is too large, the higher memory used by the IPS engine may cause the system to enter conserve mode more frequently. If set too low, the system may enter IPS fail-open mode too frequently.

Session count accuracy

The IPS engine can track the number of open session in two ways. An accurate count uses more resources than a less accurate heuristic count.

To configure the IPS open session count mode:

```
config ips global
    set session-limit-mode {accurate | heuristic}
end
```

The default is `heuristic`.

Protocol decoders

The FortiGate Intrusion Prevention system uses protocol decoders to identify the abnormal traffic patterns that do not meet the protocol requirements and standards. For example, the HTTP decoder monitors traffic to identify any HTTP packets that do not meet the HTTP protocol standards.

To change the ports a decoder examines, you must use the CLI.

To configure protocol decoder ports:

```
config ips decoder dns_decoder
    config parameter "port_list"
        set value "100,200,300"
    end
end
```

In this example, the ports examined by the DNS decoder were changed from the default 53 to 100, 200, and 300.

You cannot assign specific ports to decoders that are set to *auto* by default. These decoders can detect their traffic on any port. Specifying individual ports is not necessary.

SCTP filtering capabilities

A Stream Control Transmission Protocol (SCTP) dissector and Payload Protocol Identifier (PPID) filter can be used to either terminate the SCTP session, or replace the offending data chunk with zeros to keep the client and server sequence numbers synchronized. The SCTP filter action can also pass the data chunk.

To configure and test an SCTP filter:

1. Configure an SCTP filter profile that uses the reset action:

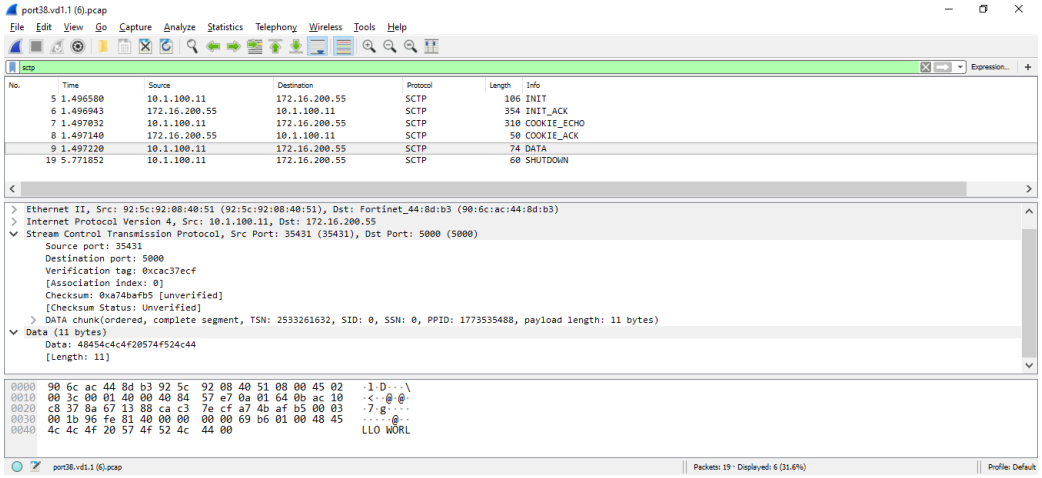
```
config sctp-filter profile
    edit "sctp"
        set comment "Demo profile"
        config ppid-filters
            edit 1
                set ppid 112233
                set action reset
                set comment "test chunk"
            next
        end
    next
end
```

2. Use the SCTP filter profile in a firewall policy:

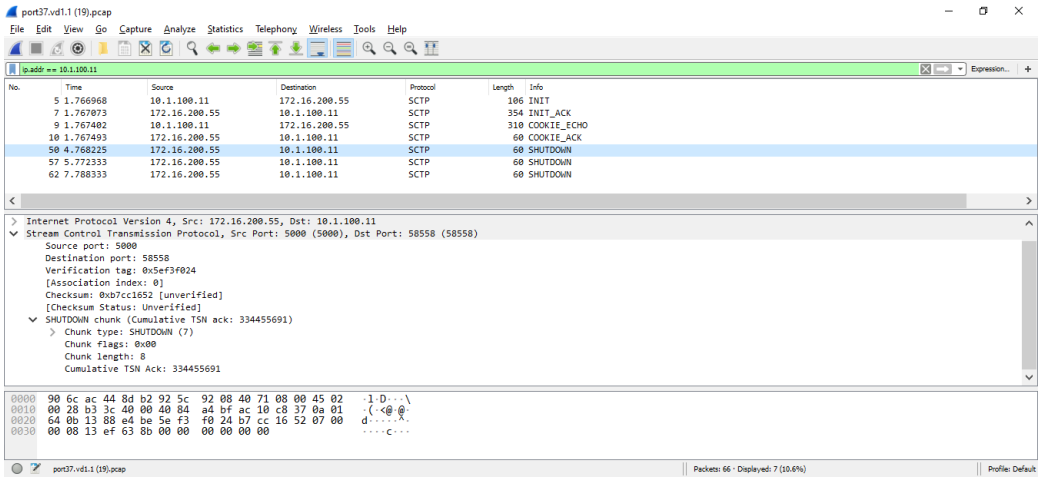
```
config firewall policy
    edit 1
        set name "1"
        set srcintf "port38"
        set dstintf "port37"
        set srcaddr "all"
        set dstaddr "all"
```

```
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "new-deep-inspection"
        set sctp-filter-profile "sctp"
        set logtraffic all
    next
end
```

3. On the SCTP client, confirm that the connection works and send a data chunk with PPID 112233.



4. The IPS engine detects the data chunk. The PPID matches the PPID filter, and the filter action is reset, so the data chunk is not received on the server, and the session is terminated.

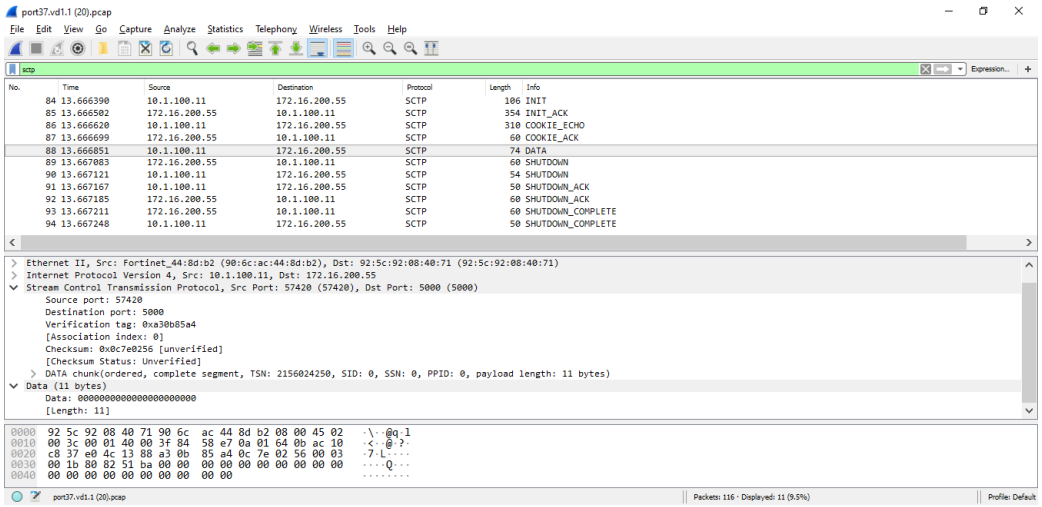


5. Change the filter action to replace:

```
config sctp-filter profile
    edit "sctp"
        config ppid-filters
            edit 1
                set action replace
            next
```

```
        end
    next
end
```

6. Resend the data chunk.
7. The IPS engine detects the data chunk. The PPID matches the PPID filter, and the filter action is replace, so the data chunk is replaced with zeros.



Diameter protocol inspection

Diameter protocol inspection is supported on the FortiGate, which offers the following capabilities.

- Diameter-based packet forwarding and routing: the FortiGate can forward and route Diameter packets that match a firewall policy with an enabled and assigned `diameter-filter profile`. These diameter packets traverse over SCTP or TCP on the reserved port 3868.
- Packet sanity checking: this feature checks if the packet passing through the FortiGate conforms to the Diameter protocol standards as defined in RFC 3588.
  - This includes checking the release version field, error command flags, message length, reserved command flag bits, command code, and tracking the request and answer of the Diameter-based packets.
- Logging: for network auditing purposes, the traffic for both dropped and forwarded Diameter-based packets of the supported commands can be logged. By default, these are disabled.

Diameter protocol is particularly important on interfaces that are used to exchange information with roaming partners, through the Internetwork Packet Exchange (IPX) network.

This feature requires a valid IPS license.

```
config diameter-filter profile
    edit <name>
        set monitor-all-messages {enable | disable}
        set log-packet {enable | disable}
```

```
        set track-requests-answers {enable | disable}
        set missing-request-action {allow | block | reset | monitor}
        set protocol-version-invalid {allow | block | reset | monitor}
        set message-length-invalid {allow | block | reset | monitor}
        set request-error-flag-set {allow | block | reset | monitor}
        set cmd-flags-reserve-set {allow | block | reset | monitor}
        set command-code-invalid {allow | block | reset | monitor}
        set command-code-range <min-max>
    next
end
```

Command	Description
`monitor-all-messages {enable | disable}`	Enable/disable logging for all User-Name and Result-Code AVP messages.
`log-packet {enable | disable}`	Enable/disable packet log for triggered Diameter settings.
`track-requests-answers {enable | disable}`	Enable/disable validation that each answer has a corresponding request.
`missing-request-action {allow | block | reset | monitor}`	Set the action to be taken for answers without a corresponding request. • `allow`: allow or pass matching traffic. • `block`: block or drop matching traffic. • `reset`: reset sessions for matching traffic. • `monitor`: allow and log matching traffic.
`protocol-version-invalid {allow | block | reset | monitor}`	Set the action to be taken for an invalid protocol version. • `allow`: allow or pass matching traffic. • `block`: block or drop matching traffic. • `reset`: reset sessions for matching traffic. • `monitor`: allow and log matching traffic.
`message-length-invalid {allow | block | reset | monitor}`	Set the action to be taken for an invalid message length. • `allow`: allow or pass matching traffic. • `block`: block or drop matching traffic. • `reset`: reset sessions for matching traffic. • `monitor`: allow and log matching traffic.
`request-error-flag-set {allow | block | reset | monitor}`	Set the action to be taken for request messages with an error flag set. • `allow`: allow or pass matching traffic. • `block`: block or drop matching traffic. • `reset`: reset sessions for matching traffic. • `monitor`: allow and log matching traffic.
`cmd-flags-reserve-set {allow | block | reset | monitor}`	Set the action to be taken for messages with a command flag reserve bits set. • `allow`: allow or pass matching traffic. • `block`: block or drop matching traffic. • `reset`: reset sessions for matching traffic.

Command	Description
	• `monitor`: allow and log matching traffic.
`command-code-invalid {allow | block | reset | monitor}`	Set the action to be taken for messages with an invalid command code. • `allow`: allow or pass matching traffic. • `block`: block or drop matching traffic. • `reset`: reset sessions for matching traffic. • `monitor`: allow and log matching traffic.
`command-code-range <min-max>`	Set the valid range for command codes (min = 0, max = 16777215, default = 256-16777213).

To configure Diameter protocol inspection:

1. Configure the Diameter filter profile:

```
config diameter-filter profile
    edit "diameter_profile"
        set monitor-all-messages disable
        set log-packet enable
        set track-requests-answers enable
        set missing-request-action block
        set protocol-version-invalid block
        set message-length-invalid block
        set request-error-flag-set block
        set cmd-flags-reserve-set block
        set command-code-invalid block
        set command-code-range 256-1677213
    next
end
```

2. Apply the Diameter filter to a firewall policy:

```
config firewall policy
    edit 1
        set srcintf "port1"
        set dstintf "port3"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "deep-inspection"
        set diameter-filter-profile "diameter_profile"
        set logtraffic all
        set auto-asic-offload disable
    next
end
```

NTurbo does not fully support SCTP, so if the configuration includes Diameter-over-SCTP, the `auto-asic-offload` setting should be disabled in the firewall policy. Otherwise, IPS does not get the full session packets.

Sample logs

No matching request:

```
1: date=2023-11-09 time=11:04:32 eventtime=1699556673071701052 logid="0419016386" type="utm"
subtype="ips" eventtype="signature" level="alert" vd="vdom1" severity="info" srcip=10.1.100.32
srccountry="Reserved" dstip=172.16.200.33 dstcountry="Reserved" srcintf="port1"
srcintfrole="undefined" dstintf="port3" dstintfrole="undefined" sessionid=163572 action="dropped"
proto=132 service="sctp/3868" policyid=1 poluuid="c17362a6-7a84-51ee-0025-80ce4c60ec49"
policytype="policy" attack="Diameter.Response.Message.No.Matching.Request.Found"
direction="outgoing" attackid=52234 ref="http://www.fortinet.com/ids/VID52234"
incidentserialno=60817776 msg="diameter_decoder:
Diameter.Response.Message.No.Matching.Request.Found, command_code=317"
```

Invalid protocol version:

```
1: date=2023-11-08 time=20:20:54 eventtime=1699503655386037801 logid="0419016386" type="utm"
subtype="ips" eventtype="signature" level="alert" vd="vdom1" severity="info" srcip=10.1.100.32
srccountry="Reserved" dstip=172.16.200.33 dstcountry="Reserved" srcintf="port1"
srcintfrole="undefined" dstintf="port3" dstintfrole="undefined" sessionid=117419 action="dropped"
proto=132 service="sctp/3868" policyid=1 poluuid="c17362a6-7a84-51ee-0025-80ce4c60ec49"
policytype="policy" attack="Diameter.Invalid.Version" direction="outgoing" attackid=52229
ref="http://www.fortinet.com/ids/VID52229" incidentserialno=60817657 msg="diameter_decoder:
Diameter.Invalid.Version, protocol_version=2"
```

Incorrect message length:

```
1: date=2023-11-08 time=19:18:10 eventtime=1699499890820325221 logid="0419016386" type="utm"
subtype="ips" eventtype="signature" level="alert" vd="vdom1" severity="info" srcip=10.1.100.32
srccountry="Reserved" dstip=172.16.200.33 dstcountry="Reserved" srcintf="port1"
srcintfrole="undefined" dstintf="port3" dstintfrole="undefined" sessionid=113487 action="dropped"
proto=132 service="sctp/3868" policyid=1 poluuid="c17362a6-7a84-51ee-0025-80ce4c60ec49"
policytype="policy" attack="Diameter.Incorrect.Message.Length" direction="outgoing" attackid=52230
ref="http://www.fortinet.com/ids/VID52230" incidentserialno=60817601 msg="diameter_decoder:
Diameter.Incorrect.Message.Length, message_length=174, packet_length=164"
```

Request error flag:

```
1: date=2023-11-08 time=19:27:29 eventtime=1699500449951027175 logid="0419016386" type="utm"
subtype="ips" eventtype="signature" level="alert" vd="vdom1" severity="info" srcip=10.1.100.32
srccountry="Reserved" dstip=172.16.200.33 dstcountry="Reserved" srcintf="port1"
srcintfrole="undefined" dstintf="port3" dstintfrole="undefined" sessionid=114134 action="dropped"
proto=132 service="sctp/3868" policyid=1 poluuid="c17362a6-7a84-51ee-0025-80ce4c60ec49"
```

```
policytype="policy" attack="Diameter.Request.Message.Error.Flag.Set" direction="outgoing"
attackid=52231 ref="http://www.fortinet.com/ids/VID52231" incidentserialno=60817619 msg="diameter_
decoder: Diameter.Request.Message.Error.Flag.Set, command_flags=A0"
```

Incorrect reserved bits:

```
1: date=2023-11-08 time=19:31:10 eventtime=1699500670891359990 logid="0419016386" type="utm"
subtype="ips" eventtype="signature" level="alert" vd="vdom1" severity="info" srcip=10.1.100.32
srccountry="Reserved" dstip=172.16.200.33 dstcountry="Reserved" srcintf="port1"
srcintfrole="undefined" dstintf="po/cdoc/ImplementationDoc5906/FGT_FileFilter_7-4_2512_
202311090951_correct config.confrt3" dstintfrole="undefined" sessionid=114400 action="dropped"
proto=132 service="sctp/3868" policyid=1 poluuid="c17362a6-7a84-51ee-0025-80ce4c60ec49"
policytype="policy" attack="Diameter.Incorrect.Reserved.Bits" direction="outgoing" attackid=52232
ref="http://www.fortinet.com/ids/VID52232" incidentserialno=60817626 msg="diameter_decoder:
Diameter.Incorrect.Reserved.Bits, command_flags=82"
```

Out-of-range command code:

```
2: date=2023-11-08 time=16:59:41 eventtime=1699491581561225681 logid="0419016386" type="utm"
subtype="ips" eventtype="signature" level="alert" vd="vdom1" severity="info" srcip=10.1.100.32
srccountry="Reserved" dstip=172.16.200.33 dstcountry="Reserved" srcintf="port1"
srcintfrole="undefined" dstintf="port3" dstintfrole="undefined" sessionid=106658 action="dropped"
proto=132 service="sctp/3868" policyid=1 poluuid="c17362a6-7a84-51ee-0025-80ce4c60ec49"
policytype="policy" attack="Diameter.Message.Command.Overlong" direction="outgoing" attackid=52233
ref="http://www.fortinet.com/ids/VID52233" incidentserialno=60817600 msg="diameter_decoder:
Diameter.Message.Command.Overlong, command_code=255, range_min=256, range_max=1677213"
```

IPS signature filter options

IPS signature filter options include hold time, CVE pattern, and IPS sensor attributes.

Hold time

The hold time option allows you to set the amount of time that signatures are held after a FortiGuard IPS signature update per VDOM. During the holding period, the signature's mode is monitor. The new signatures are enabled after the hold time to avoid false positives.

The hold time can be from 0 days and 0 hours (default) up to 21 days, in the format ##d##h.

To configure the amount of time to hold and monitor IPS signatures:

```
config system ips
    set signature-hold-time 3d12h
    set override-signature-hold-by-id enable
end
```

When a signature that is on hold is matched, the log will include the message `signature is on hold`:

```
date=2010-07-06 time=00:00:57 logid="0419016384" type="utm" subtype="ips" eventtype="signature"
level="alert" vd="vd1" eventtime=1278399657778481842 tz="-0700" severity="info" srcip=10.1.100.22
srccountry="Reserved" dstip=172.16.200.55 srcintf="port13" srcintfrole="undefined"
dstintf="port14" dstintfrole="undefined" sessionid=3620 action="detected" proto=6 service="HTTP"
policyid=1 attack="Eicar.Virus.Test.File" srcport=52170 dstport=80 hostname="172.16.200.55"
url="/virus/eicar" direction="incoming" attackid=29844 profile="test"
ref="http://www.fortinet.com/ids/VID29844" incidentserialno=25165825 msg="file_transfer:
Eicar.Virus.Test.File, (signature is on hold)"
```

To view signatures being held by rule ID 29844 on the VDOM:

```
# diagnose ips signature on-hold vd1 29844
Rule: 29844, attack_id: 58886, last updated: 20170411
Rule: 29844, attack_id: 59517, last updated: 20170411
Rule: 29844, attack_id: 60105, last updated: 20170411
...
```

To view all help signatures on the VDOM:

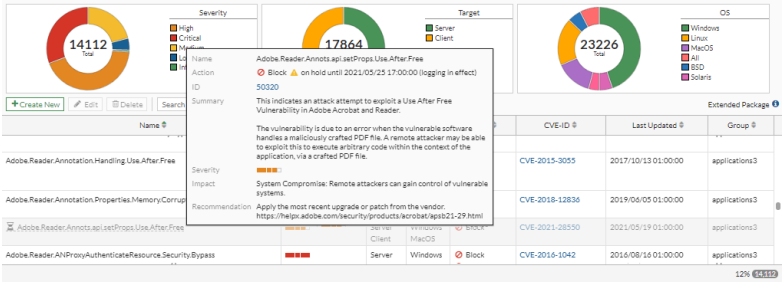
```
# diagnose ips signature on-hold vd1
Rule: 17541, attack_id: 20899, last updated: 20140423
Rule: 17557, attack_id: 20934, last updated: 20140423
Rule: 17559, attack_id: 20932, last updated: 20140423
Rule: 17560, attack_id: 20933, last updated: 20140423
Rule: 17562, attack_id: 20928, last updated: 20170908
Rule: 17677, attack_id: 21187, last updated: 20171106
Rule: 17713, attack_id: 43756, last updated: 20140424
Rule: 17759, attack_id: 21298, last updated: 20140423
...
```

When diagnosing IPS signatures in the CLI, the `Rule` field corresponds to the `attackid` log field, the *ID* in the FortiGate GUI, as well as the IPS entry ID in the FortiGuard IPS encyclopedia.

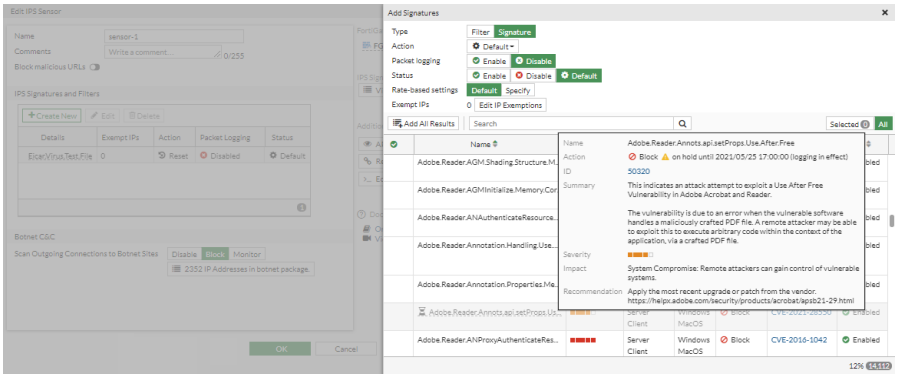
Viewing on hold information in the GUI

On hold signatures are grayed out in the GUI with an hourglass icon beside the signature name. A tooltip displays the on hold expiry time and other details.

On the *Security Profiles* > *IPS Signatures* page, for example, the *Adobe.Reader.Annots.api.setProps.Use.After.Free* signature is on hold. Hover over the grayed-out entry to view the tooltip, which includes the action and hold time expiry. On this page, all on hold signatures are displayed as on hold regardless of whether `override-signature-hold-by-id` is enabled.



The same tooltip is available on the *Edit IPS Sensor* (*Security Profiles* > *Intrusion Prevention*) page when creating or editing the IPS signatures. In the *Add Signatures* pane when the *Type* is *Signature*, signatures on hold are only displayed as on hold if `override-signature-hold-by-id` is enabled.



You can still use on hold signatures in an IPS sensor profile; however, the profile will not block matching traffic. It will monitor it instead (logging in effect) until the on hold time expires.

CVE pattern

The CVE pattern option allows you to filter IPS signatures based on CVE IDs or with a CVE wildcard, ensuring that any signatures tagged with that CVE are automatically included.

To configure CVE patterns for CVE-2010-0177 and all CVE-2017 CVEs:

```
config ips sensor
    edit "cve"
        set comment "cve"
        config entries
            edit 1
                set cve "cve-2010-0177"
                set status enable
                set log-packet enable
                set action block
            next
            edit 2
                set cve "cve-2017"
                set action reset
```

```
            next
        end
    next
end
```

For example, the CVE of the IPS signature *Mozilla.Firefox.PluginArray.NsMimeType.Code.Execution* is CVE-2010-0177. This matches the CVE filter in the IPS sensor, so traffic is blocked and logged:

```
date=2020-07-13 time=15:44:56 logid="0419016384" type="utm" subtype="ips" eventtype="signature"
level="alert" vd="vd1" eventtime=1594593896666145871 tz="-0700" severity="critical"
srcip=10.1.100.22 srccountry="Reserved" dstip=172.16.200.55 srcintf="port2"
srcintfrole="undefined" dstintf="port1" dstintfrole="undefined" sessionid=1638 action="dropped"
proto=6 service="HTTPS" policyid=1 attack="Mozilla.Firefox.PluginArray.NsMimeType.Code.Execution"
srcport=58298 dstport=443 hostname="172.16.200.55" url="/Mozilla" direction="incoming"
attackid=20853 profile="sensor-1" ref="http://www.fortinet.com/ids/VID20853"
incidentserialno=124780667 msg="web_client:
Mozilla.Firefox.PluginArray.NsMimeType.Code.Execution," crscore=50 craction=4096
crlevel="critical"
```

IPS sensor attributes

When configuring IPS sensor profiles, IPS signatures can be filtered based on the attributes: default status, default action, vulnerability type, and the last update date. When monitoring the specific, filtered signatures, logs are not generated for other, irrelevant signatures.

This avoids generating a lot of false positives due to many signatures having the pass action, which is never logged.

To configure filters in an IPS sensor profile:

```
config ips sensor
    edit "test_default"
        config entries
            edit 1
                set default-action pass
                set default-status enable
                set vuln-type 12
                set last-modified before 2020/02/02
            next
        end
    next
end
```

`default-action {pass | block | all}`	Filter by signatures' default actions (default = all).
`default-status {enable | disable | all}`	Filter by signatures' default statuses (default = all).
`vuln-type <integer> ... <integer>`	Filter by signatures' vulnerability types.

`last-modified {before | after | between} <date> [end-date]`	Filter by signatures' last modified date (default = before 00/00/00). The date format is `yyyy/mm/dd`. The year range is 2001 - 2050.

When the IPS profile is used in a firewall profile and then the EICAR virus test file signature is triggered, the signature matches the values set in the filter and logs are generated:

```
1:date=2022-02-15 time=14:07:03 eventtime=1644962823303491048 tz="-0800" logid="0419016384"
type="utm" subtype="ips" eventtype="signature" level="alert" vd="vd1" severity="info"
srcip=10.1.100.11 srccountry="Reserved" dstip=172.16.200.55 dstcountry="Reserved" srcintf="port38"
srcintfrole="undefined" dstintf="port37" dstintfrole="undefined" sessionid=1171 action="detected"
proto=6 service="HTTP" policyid=1 poluuid="623d2d28-8ea7-51ec-00ef-7549685a77c2"
policytype="policy" attack="Eicar.Virus.Test.File" srcport=47230 dstport=80
hostname="172.16.200.55" url="/virus/eicar" direction="incoming" attackid=29844 profile="test_
default" ref="http://www.fortinet.com/ids/VID29844" incidentserialno=103809025 msg="file_transfer:
Eicar.Virus.Test.File"
```

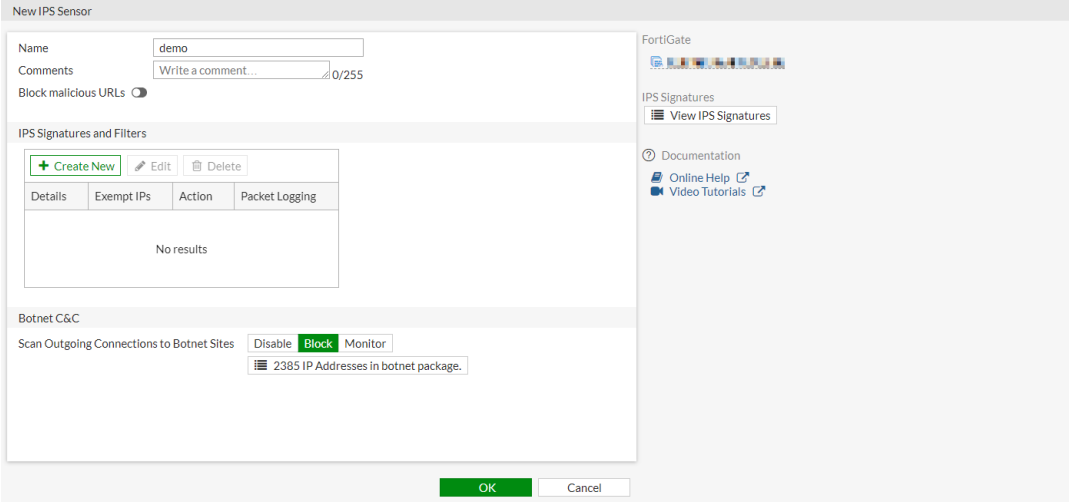
```
# get ips rule status | grep Eicar.Virus.Test.File -A 18
rule-name: "Eicar.Virus.Test.File"
rule-id: 29844
rev: 10.111
date: 1491926400
action: pass
status: enable
log: disable
log-packet: disable
severity: 0.info
service: TCP, HTTP, FTP, SMTP, POP3, IMAP, NNTP
location: server, client
os: All
application: Other
rate-count: 0
rate-duration: 0
rate-track: none
rate-mode: continuous
vuln_type: Anomaly
```

IPS with botnet C&C IP blocking

The *Botnet C&C* section consolidates multiple botnet options in the IPS profile. This allows you to enable botnet blocking across all traffic that matches the policy by configuring one setting in the GUI, or by the `scan-botnet-connections` option in the CLI.

To configure botnet C&C IP blocking in the GUI:

1. Go to *Security Profiles* > *Intrusion Prevention*, and click *Create New* to create a new IPS sensor, or double-click an existing IPS sensor to open it for editing.
2. Navigate to the *Botnet C&C* section.
3. For *Scan Outgoing Connections to Botnet Sites*, select *Block* or *Monitor*.



4. Configure the other settings as needed.
5. Click *OK* to save the IPS sensor.
6. Add the IPS sensor to a firewall policy.

   The IPS engine will scan outgoing connections to botnet sites. If you access a botnet IP address, an IPS log is generated for this attack.
7. Go to *Log & Report* > *Security Events* and click the *Intrusion Prevention* card to view the log.

To configure botnet C&C IP blocking in the CLI:

```
config ips sensor
    edit "Demo"
        set scan-botnet-connections {disable | block | monitor}
    next
end
```

The `scan-botnet-connections` option is no longer available in the following CLI commands:

- `config firewall policy`
- `config firewall interface-policy`
- `config firewall proxy-policy`
- `config firewall sniffer`

Sample log

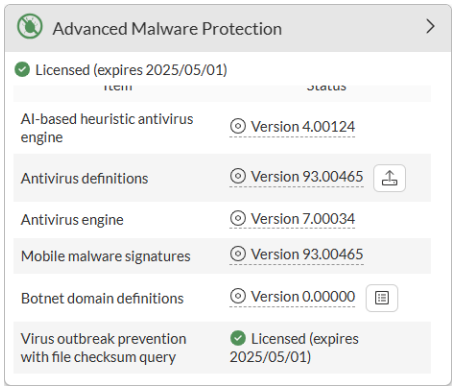
```
# execute log filter category 4
# execute log display
1 logs found.
1 logs returned.

1: date=2022-04-28 time=16:18:34 eventtime=1651187914585406621 tz="-0700" logid="0422016400"
type="utm" subtype="ips" eventtype="botnet" level="warning" vd="vd1" msg="Botnet C&C
Communication." severity="critical" srcip=10.1.100.11 srccountry="Reserved" dstip=2.58.149.169
srcintf="port13" srcintfrole="undefined" dstintf="port14" dstintfrole="undefined" sessionid=894198
```

```
action="dropped" srcport=41798 dstport=80 proto=6 service="HTTP" policyid=1 profile="sensor-1"
direction="outgoing" attack="Loki" attackid=7630239 ref="http://www.fortinet.com/be?bid=7630239"
crscore=50 craction=4 crlevel="critical"
```

Botnet IPs and domains lists

To view botnet IPs and domains lists:

1. Go to *System* > *FortiGuard*.
2. Go to the *Subscriptions* > *Licensed* tab.
3. Expand *Advanced Malware Protection* to view *Botnet domain definitions* information.

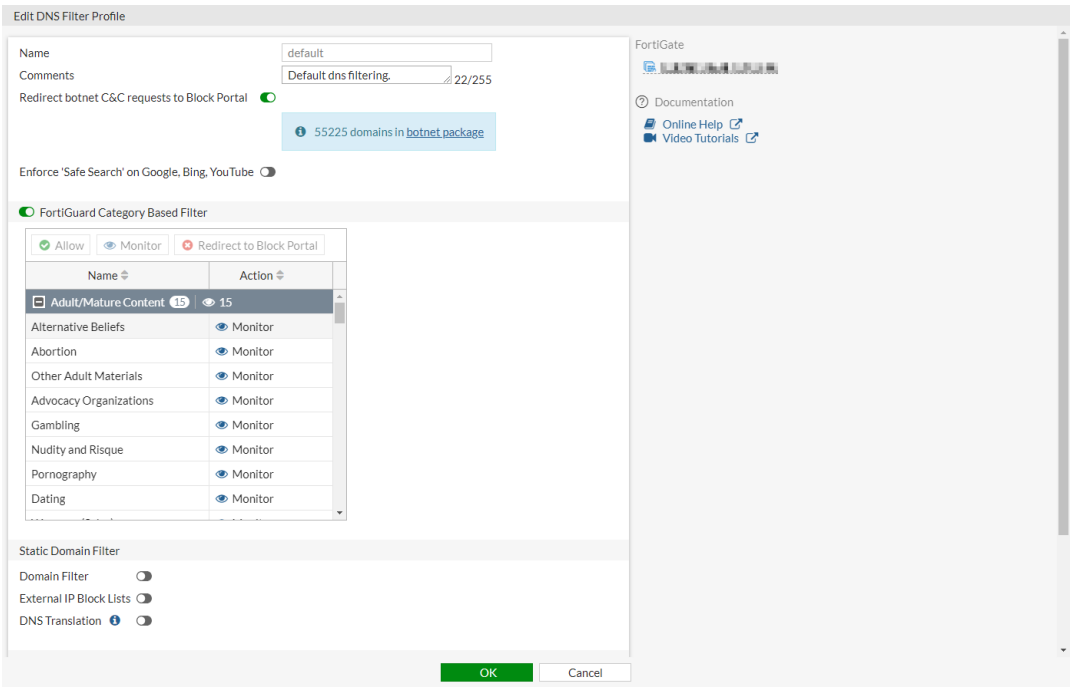


4. Click the action.

Botnet C&C domain blocking

To block connections to botnet domains:

1. Go to *Security Profiles* > *DNS Filter*, and click *Create New*, or double-click an existing filter to open it for editing.
2. Enable *Redirect botnet C&C requests to Block Portal*.

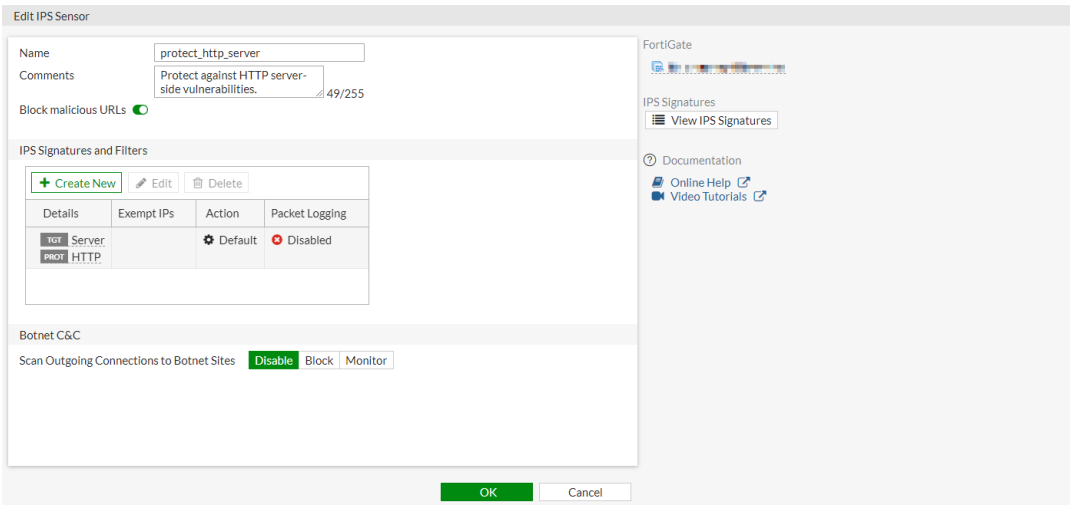


3. Configure the other settings as needed.
4. Click *OK*.
5. Add the filter profile to a firewall policy.

Botnet C&C URL blocking

To block malicious URLs:

1. Go to *Security Profiles* > *Intrusion Prevention*, and click *Create New*, or double-click an existing filter to open it for editing.
2. Enable *Block malicious URLs*.



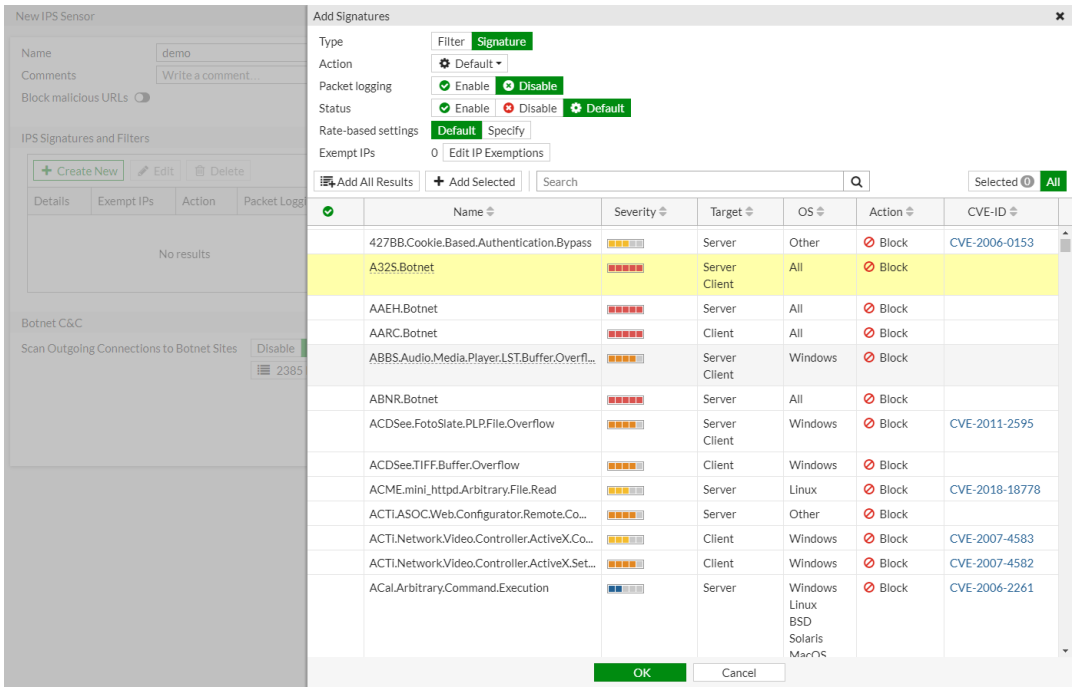
3. Configure the other settings as needed.
4. Click *OK*.

5. Add the sensor to a firewall policy.

Botnet C&C signature blocking

To add IPS signatures to a sensor:

1. Go to *Security Profiles* > *Intrusion Prevention*, and click *Create New*, or double-click an existing sensor to open it for editing.
2. In the *IPS Signatures and Filters* section, click *Create New*. A list of available signatures appears.
3. For *Type*, select *Signature*. Select the signatures you want to include from the list.
4. Configure the other settings as needed.
5. Click *Add Selected*.



6. Click *OK* to add the IPS signatures to the IPS sensor.
7. Click *OK* to save the IPS sensor.
8. Add the sensor to a firewall policy to detect or block attacks that match the IPS signatures.

IPS signatures for the operational technology security service

The FortiGuard Operational Technology (OT) includes both application control and intrusion prevention signatures for industrial applications and protocols. The OT attack definitions are only updated if the FortiGate has a valid OT Security Services license and an IPS security profile is used in a policy.

By default, OT signatures are excluded from the signature lists in the GUI.

To verify that the FortiGate has a valid OT Security Service license:

1. Go to *System* > *FortiGuard* > *Subscriptions* tab.
2. In the *Licensed* content, check the license status of *Operational Technology (OT) Security Service*.
3. Expand the *Operational Technology (OT) Security Service* entry to see the current versions.

To force the industrial DB attack definitions to update:

1. Optionally, create an IPS profile:
   a. Go to *Security Profiles* > *Intrusion Prevention* and click *Create New*.
   b. Enter a name for the profile.
   c. In the *IPS Signatures and Filters* table click *Create New*.
   d. Click *OK*.
   e. Click *OK*.
   See Intrusion prevention on page 2478 for more information.
2. Use the IPS profile in a policy:
   a. Go to *Policy & Objects* > *Firewall Policy*.
   b. Edit an existing policy, or click *Create New* to create a new policy.
   c. Under *Security Profiles*, enable *IPS* and select an IPS profile.
   d. Configure the remaining settings as needed, then click *OK*.
3. Go to *System* > *FortiGuard* and either click *Update Licenses & Definitions Now*, or wait for the next automatic update. The update could take a few minutes.
4. Refresh the page, then check the *Operational Technology (OT) Security Service* versions to confirm that they have been updated.

To make OT IPS and application control signatures available in the GUI:

```
config ips global
    set exclude-signatures none
end
```

To view the signatures in the GUI:

1. Go to *Security Profiles* > *Application Signatures* and to find signatures that identify OT protocols.
2. Go to *Security Profiles* > *IPS Signatures* to find signatures that detect networks attacks that target OT assets.

To see the entire list of OT IPS Rules and OT Application Control rules, go to the following links:

- https://www.fortiguard.com/encyclopedia?type=otips
- https://www.fortiguard.com/encyclopedia?type=otapp

To see the list of Industrial Application Control signatures, go to the following link:

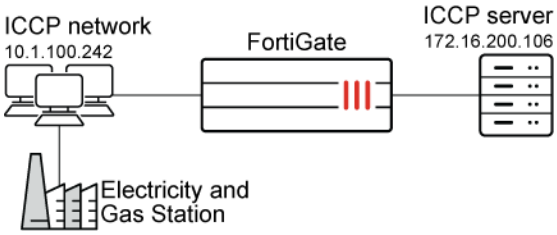
- https://www.fortiguard.com/appcontrol?category=Industrial

IPS sensor for IEC 61850 MMS protocol

IEC 61850 is a SCADA protocol whose services are mapped to a number of protocols, including MMS services. MMS/ICCP detection is supported in IPS. The purpose of the MMS dissectors is to identify every IEC 61850 service to distinguish different MMS/ICCP messages. IPS engine 6.0.12 and later support MMS dissectors.

The following scenarios are also supported:

- Multiple MMS PDUs are transferred in one TCP payload, and the IPS engine identifies individuals.
- An MMS message is split over multiple TCP segments, where MMS runs over COTP segments.
- ICCP/TASE.2 that also uses MMS transport (ISO transport over TCP for ICCP) is detected.



OT signatures must be enabled in the global IPS settings to receive MMS/ICCP signatures. By default, OT signatures are excluded.

```
config ips global
    set exclude-signatures {none | ot}
end
```

Below are some OT signatures for MMS/ICCP messages that can be detected by the IPS engine. This is not an exhaustive list.

- MMS_GetNameList.Request
- MMS_GetNamedVariableListAttributes.Request
- MMS_GetVariableAccessAttributes.Request
- MMS_Identify.Request
- MMS_Initiate.Request
- MMS_Read.Request
- MMS_Reset.Request
- ICCP_Transfer.Reporting
- ICCP_Create.Dataset
- ICCP_Abort
- ICCP_Start.Transfer.DSTransferSet
- ICCP_Get.Dataset.Element.Values
- ICCP_Get.Next.DSTransfer.Set.Value
- ICCP_Delete.Dataset
- ICCP_Start.Transfer.IMTransferSet

Diagnose command

The COTP dissector adds support for identifying every MMS PDU, and let the IPS engine separate them, like the Modbus and IEC-104 services for example.

```
# diagnose ips debug enable all
# diagnose debug enable
[284@78]ips_l7_dsct_processor: serial=8142 create: cotp
[284@78]ips_l7_dsct_processor: serial=8142 create: iec104
[284@78]ips_l7_dsct_processor: serial=8142 create: modbus
```

Log samples

MMS dissectors can be triggered, and MMS/ICCP signatures can be monitored and logged.

Log samples:

```
date=2020-03-26 time=15:51:10 logid="1059028704" type="utm" subtype="app-ctrl"
eventtype="signature" level="information" vd="vd1" eventtime=1585263070836106492 tz="-0700"
appid=43699 srcip=10.1.100.242 dstip=172.16.200.106 srcport=50963 dstport=102 srcintf="port13"
srcintfrole="undefined" dstintf="port14" dstintfrole="undefined" proto=6 service="tcp/26112"
direction="outgoing" policyid=1 sessionid=2711 applist="test" action="pass"
appcat="Operational.Technology" app="MMS_Read.Request" incidentserialno=376610508
msg="Operational.Technology: MMS_Read.Request," apprisk="elevated"
```

```
date=2020-03-26 time=16:15:45 logid="1059028704" type="utm" subtype="app-ctrl"
eventtype="signature" level="information" vd="vd1" eventtime=1585091746264983273 tz="-0700"
appid=44684 srcip=10.1.100.242 dstip=172.16.200.106 srcport=41665 dstport=102 srcintf="port13"
srcintfrole="undefined" dstintf="port14" dstintfrole="undefined" proto=6 service="tcp/26112"
direction="incoming" policyid=1 sessionid=194463 applist="test" action="pass"
appcat="Operational.Technology" app="ICCP_Transfer.Reporting" incidentserialno=762763993
msg="Operational.Technology: ICCP_Transfer.Reporting," apprisk="elevated"
```

IPS Modbus TCP decoder

Modbus TCP is a protocol used to facilitate communication between devices in the Operational Technology (OT) environment. By default, it uses TCP/502 port for communication.

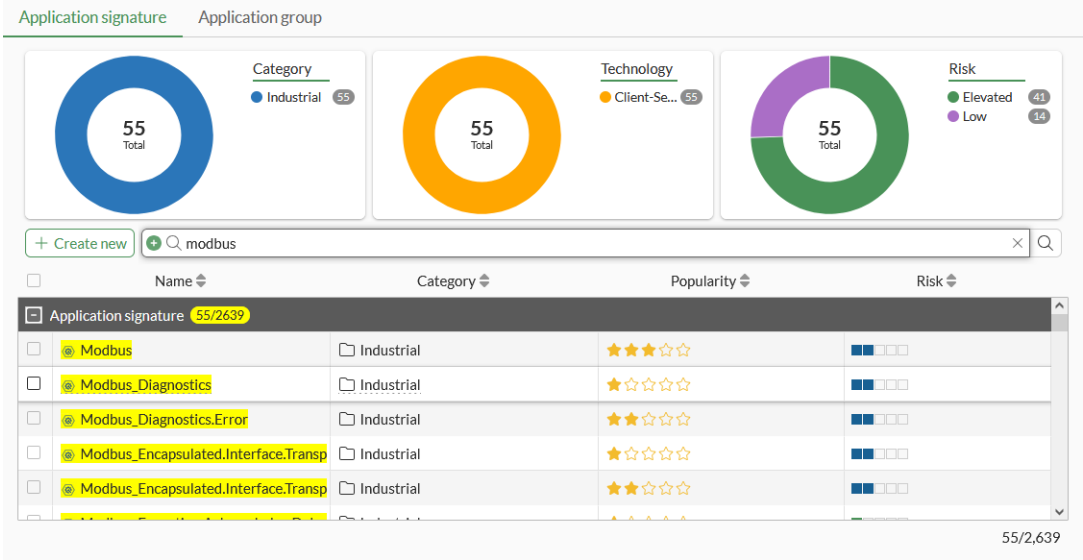
The IPS engine supports the Modbus TCP decoder, allowing it to decode Modbus protocol messages and enable application control signatures for Modbus commands and parameters. See Matching multiple parameters on application control signatures on page 2443 for information about parameter-level application policies.

OT IPS signatures must be enabled in the global IPS settings to receive Modbus signatures for application control and vulnerability protection. OT IPS signatures are part of the FortiGuard OT security service, and are excluded by default.

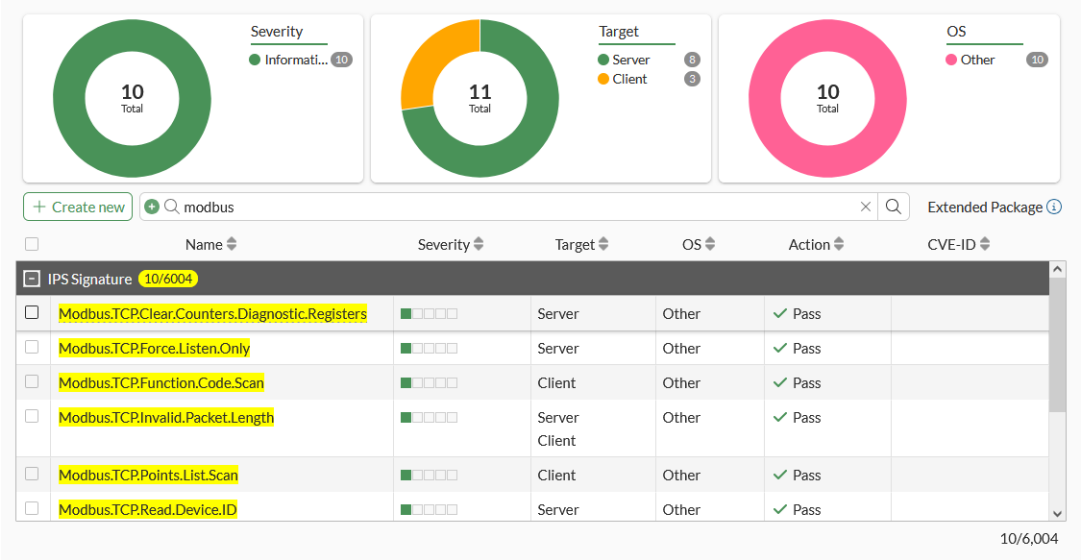
To include OT IPS signatures:

```
config ips global
    set exclude-signatures none
end
```

Modbus application control signatures are listed on *Security Profiles* > *Application Signatures*. Search for *Modbus* to see the Modbus signatures, such as *Modbus_Diagnostics* and *Modbus_Read.Coils*.



Modbus vulnerability protection signatures are listed on *Security Profiles* > *IPS Signatures*. Search for *Modbus* to see the Modbus signatures.



By default, the Modbus decoder listens on TCP/502 port only, but can be configured to listen on additional TCP ports.

To configure the Modbus decoder to listen on additional TCP ports:

```
config ips decoder "modbus_decoder"
    config parameter "port_list"
        set value "502:505"
    end
end
```

AI and ML-based IPS detection

As cyber threats become increasingly sophisticated, traditional signature-based detection is struggling to keep up. To improve it, AI and machine learning-based models are trained on features extracted during protocol decoding, such as HTTP traffic. These models act as classifiers, distinguishing exploits from clean traffic through supervised learning.

Instead of applying machine learning (ML) models blindly across all traffic, we will first use signatures for preliminary filtering, allowing AI-based detection to be more targeted and efficient. This hybrid approach will reduce false positives while maintaining high performance.

CLI syntax

The setting is enabled by default at the IPS global setting level:

```
config ips global
    set machine-learning-detection {enable | disable}
end
```

The `AI/Machine Learning IPS Definitions` package is downloaded by FortiOS from FortiGuard through FortiGuard updates. Devices with active IPS subscription can download this package. The package can be viewing in the `diagnose autoupdate versions` output.

```
# diagnose autoupdate versions | grep -A 7 AI

AI/Machine Learning IPS Definitions
---------
Version: 2503.00100 signed
Contract Expiry Date: Thu Jan  1 2032
Last Updated using manual update on Thu Mar 13 18:01:37 2025
Last Update Attempt: Thu Mar 13 18:01:37 2025
Result: Updates Installed
```

The IPS machine learning database version is displayed in the output of `get system status` command.

```
# get system status

Version: FortiGate-1101E v7.6.3,build3495,250313 (interim)
First GA patch build date: 240724
Security Level: Low
...
```

```
IPS-MLDB: 2503.00100(2025-03-13 03:43)
...
```

The IPS machine learning rules can be displayed in the output of `get ips rule status` command. For example, looking up rule `57293` returns the following:

```
# get ips rule status | grep -B 2 -A 16 57293

rule-name: "Backdoor.Cobalt.Strike"
rule-id: 57293
rev: 0.000
date: 2025-03-12 09:00:00
action: pass
status: enable
log: disable
log-packet: disable
severity: 3.high
service: TCP, HTTP
location: client
os: All
application: All
rate-count: 0
rate-duration: 0
rate-track: none
rate-mode: continuous
```

Example

In the following example, AI and ML-based IPS detection is implemented on a regular firewall policy. As the IPS machine learning detection runs alongside traditional IPS signature detection, the configuration of the IPS sensor remains the same.

To add AI and ML-based IPS detection:

1. Configure an IPS sensor with machine learning signature `Backdoor.Cobalt.Strike` set to `block`:

```
config ips sensor
    edit "MI-test"
        config entries
            edit 1
                set rule 57293
                set status enable
                set action block
            next
        end
    next
end
```

2. Apply the IPS sensory to a firewall policy:

```
config firewall policy
    edit 1
        set name "test"
        set srcintf "port2"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "certificate-inspection"
        set ips-sensor "MI-test"
    next
end
```

3. Generate traffic from the client.
4. Review the IPS event logs:

```
# execute log filter category 4

date=2025-03-14 time=11:46:12 eventtime=1741977972028174920 tz="-0700" logid="0419016384"
type="utm" subtype="ips" eventtype="signature" level="alert" vd="root" severity="high"
srcip=10.1.100.240 srccountry="Reserved" dstip=172.16.200.240 dstcountry="Reserved"
srcintf="port2" srcintfrole="undefined" dstintf="port1" dstintfrole="undefined" sessionid=4012
action="dropped" proto=6 service="HTTP" policyid=1 poluuid="9d827014-00fa-51f0-e5af-
640cfd149b4a" policytype="policy" attack="Backdoor.Cobalt.Strike" srcport=52294 dstport=80
hostname="10.0.2.78" url="/understand/CYBERDOCS31/S4IR30UL" agent="Mozilla/5.0 (Windows NT
6.1; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/52.0.2743.116 Safari/537.36"
httpmethod="GET" direction="incoming" attackid=57293 profile="MI-test"
ref="http://www.fortinet.com/ids/VID57293" incidentserialno=181403686 msg="backdoor:
Backdoor.Cobalt.Strike" forwardedfor="IK_
Z9RMF9GQECAZ5RZZ5CMHSUZZD4YTH6K8=\x07\x10Z%at=Icr>Gav=Jas?Nav?Coy?Lb<B`u?Baz=Jas<Bb~>Ga
{2Laz<Maz2Oaz>Fcz>G`t<M`}>Ca{>Fcr>F`r" crscore=30 craction=8192 crlevel="high"
```

5. Review the IPS debug logs:

```
# diagnose ips debug enable all
# diagnose debug enable

...
[699@214]ips_process_event: ctx 14: 6 => 2
[699@214]ips_process_event: ctx 14: 2 => 4
[699@214]ips_ml_classify_internal: model=0 labels=[0.9960013, 0.0]@0=0.9960013
[699@214]ips_match_rule: pattern matched 57293,99455: Backdoor.Cobalt.Strike
[699@214]ips_match_rule: matched rule 57293 99455 Backdoor.Cobalt.Strike (weight:0)
[699@214]ips_match_candidates: set best rule 57293 99455 Backdoor.Cobalt.Strike
[699@214]ips_set_pkt_verdict: action=DROP
[699@214]ips_set_pkt_verdict: turn tcp drop to DROP_SESSION
[699@214]ips_report_alert_va_internal: v_id=57293, a_id=99455, log=1, log_pkt=1
[699@214]ips_log: id=57293 conf=0x44, action=1
[699@214]ips_log_packet: aid=99455 log=0xb
```

```
[699@214]match_ips: disarm ftgd queries when request is to be blocked.
[699@214]ips_process_event: ctx 14: 4 => 3
[699@214]ips_handle_pkt_verdict: drop a session, size=296
[699@214]ips_session_sched_release: serial=7429 close session 0x7f8a84751018, reason 0
[699@214]ips_process_event: ctx 14: 3 => 5
[699@-1]ips_dsct_http_prep_release_sess: sess 214: http release proxy layer
```

To filter the debug logs to only display the bolded results, enter `diagnose ips debug enable ml` instead.

Support Ethernet layer protocols in the IPS engine

The IPS engine has been enhanced to detect industrial Ethernet protocols such as LLDP, GOOSE, EtherCAT, and PROFINET RT. Device detection starts to detect and log the Ethernet devices through the L2 protocol. Additionally, the IPS sensor detects the Ethernet protocol and logs the traffic.

Custom signature rules have been enhanced with three new rule options for `ethertype`, `mac_src`, and `mac_dst`.

The L2 protocol to be detected by the custom signature is specified by the administrator through the Ethertype hexadecimal value for the `ethertype` option.

In Examples 2 and 3 below, the `ethertype` value of `0x88cc` is used to detect LLDP protocol traffic.

The following examples are explored:

- Example 1: Ethernet protocol device detection on the interface on page 2514
- Example 2: Ethernet protocol detection with custom IPS signatures on the interface policy on page 2515
- Example 3: Ethernet protocol detection with custom IPS signatures on the sniffer policy on page 2516

Example 1: Ethernet protocol device detection on the interface

Device detection requires new signatures included in both the IoT Detection package and OT Detection package, which will be available in future FortiGuard updates.

In this example, the IPS engine detects Ethernet devices, such as those using the LLDP protocol, which contains device information.

To apply Ethernet protocol device detection on the interface:

1. Enable device detection and passive gathering of identity information about the host:

```
config system interface
    edit "port15"
        set vdom "root"
        set type physical
        set device-identification enable
        set snmp-index 17
```

```
    next
end
```

2. Apply the interface to the firewall interface policy:

```
config firewall interface-policy
    edit 1
        set interface "port15"
        set srcaddr "all"
        set dstaddr "all"
        set service "ALL"
    next
end
```

3. Generate traffic from the client.
4. Review the application control log for device detection on Ethernet protocol LLDP:

```
# diagnose log filter category 4
# diagnose log display
1: date=2025-03-20 time=18:46:52 eventtime=1742521612517801102 tz="-0700" logid="1059028738"
type="utm" subtype="app-ctrl" eventtype="signature" level="warning" vd="root" ethertype=35020
srcmac="00:0c:29:c6:ae:bf" dstmac="e0:23:ff:83:2d:2d" appid=10008004 srcintf="port15"
srcintfrole="undefined" dstintf="port15" dstintfrole="undefined" policyid=1 poluuid="c5ed61a0-
0445-51f0-1e08-ea9d568326bd" policytype="interface-policy" action="pass" appcat="IoT"
app="LLDP.Device.Test" msg="IoT: LLDP.Device.Test" clouddevice="Vendor=HP, Product=ProCurve
Swtich, Model=ProCurve Swtich 2600-8-PWR, Version=H.08.89" apprisk="low"
```

The `ethertype` field has been added in the device detection log. The `src_mac` and `dst_mac` log fields have been added instead of the source and destination IP addresses. These three new fields are not included for regular application control logs.

Example 2: Ethernet protocol detection with custom IPS signatures on the interface policy

In this example, the IPS sensor is able to detect Ethernet protocols by matching signatures in the NIDS database or by using custom-defined signatures.

To apply Ethernet protocol detection with custom IPS signatures on the interface policy:

1. Create a new custom signature for LLDP protocol with source MAC address and destination MAC address defined:

```
config ips custom
    edit "LLDP-test"
        set signature "F-SBID( --attack_id 6312;  --name \"LLDP-test-mac\"; --default_action
drop;  --ethertype 0x88cc; --mac_src 00:0c:29:c6:ae:bf; --mac_dst e0:23:ff:83:2d:2d; --
severity high; --status disable; )"
    next
end
```

2. Create a new IPS sensor and allow the new custom signatures to pass:

```
config ips sensor
    edit "l2-test"
        config entries
            edit 1
                set rule 6312
                set status enable
                set action pass
            next
        end
    next
end
```

The rule identifies the IPS signatures to add to the sensor.

3. Enable the IPS sensor in a firewall interface policy and apply the new IPS sensor:

```
config firewall interface-policy
    edit 1
        set interface "port15"
        set srcaddr "all"
        set dstaddr "all"
        set service "ALL"
        set ips-sensor-status enable
        set ips-sensor "l2-test"
    next
end
```

4. Generate traffic from the client.
5. Review the IPS sensor log for the Ethernet protocol LLDP custom signature:

```
# diagnose log filter category 4
# diagnose log display
1: date=2025-03-21 time=10:16:03 eventtime=1742577362746522425 tz="-0700" logid="0419016402"
type="utm" subtype="ips" eventtype="signature" level="warning" vd="root" ethertype=35020
srcmac="00:0c:29:c6:ae:bf" dstmac="e0:23:ff:83:2d:2d" srcintf="port15" srcintfrole="undefined"
dstintf="port15" dstintfrole="undefined" policyid=1 poluuid="8fcb4bf0-0675-51f0-fa7a-
9fec94e8115c" policytype="interface-policy" action="dropped" attack="\"LLDP-test\""
attackid=6312 profile="l2-test" incidentserialno=75497498 msg="custom: \"LLDP-test\""
crscore=30 craction=8192 crlevel="high"
```

A new log ID has been created for the L2 detection logs.

The `ethertype` field has been added in the device detection log. The `src_mac` and `dst_mac` log fields have been added instead of the source and destination IP addresses. These new fields are not included in a regular IPS sensor log.

Example 3: Ethernet protocol detection with custom IPS signatures on the sniffer policy

Ethernet protocol detection is supported in sniffer policies. In this example, the software switch with spanning is set for the sniffer detection.

To apply Ethernet protocol detection with custom IPS signatures on the sniffer policy:

1. Create a new software switch with the destination and source ports for spanning configured:

```
config system switch-interface
    edit "test-sw"
        set vdom "root"
        set member "port2" "port15"
        set span enable
        set span-dest-port "port2"
        set span-source-port "port15"
    next
end
```

2. Enable one arm sniffer on the destination interface:

```
config system interface
    edit "port2"
        set vdom "root"
        set ips-sniffer-mode enable
        set type physical
        set snmp-index 4
    next
end
```

3. Create a new custom signature for LLDP protocol with source MAC address and destination MAC address defined:

```
config ips custom
    edit "LLDP-test"
        set signature "F-SBID( --attack_id 6312;  --name \"LLDP-test-mac\"; --default_action
drop;  --ethertype 0x88cc; --mac_src 00:0c:29:c6:ae:bf; --mac_dst 01:80:c2:00:00:0e; --
severity high; --status disable; )"
    next
end
```

4. Configure the IPS sensor:

```
config ips sensor
    edit "g-sniffer-profile"
        set comment "Monitor IPS attacks."
        config entries
            edit 2
                set rule 29844
                set status enable
                set action block
            next
            edit 1
                set rule 6312
                set status enable
                set action pass
            next
        end
```

```
    next
end
```

5. Apply the interface to the firewall sniffer policy:

```
config firewall sniffer
    edit 1
        set non-ip enable
        set interface "port2"
        set ips-sensor-status enable
        set ips-sensor "g-sniffer-profile"
    next
end
```

> Confirm that non-IP address packet sniffing is also enabled. If it is not enabled, L2 traffic will not be detected.

6. Generate Ethernet traffic and mirror it to the designated interface port.
7. Review new log for sniffer policy Ethernet protocol detection:

```
# diagnose log filter category 4
# diagnose log display
1: date=2025-04-14 time=17:28:21 eventtime=1744676901127125507 tz="-0700" logid="0419016402"
type="utm" subtype="ips" eventtype="signature" level="warning" vd="root" ethertype=35020
srcmac="00:0c:29:c6:ae:bf" dstmac="01:80:c2:00:00:0e" srcintf="port2" srcintfrole="undefined"
dstintf="port2" dstintfrole="undefined" policyid=1 poluuid="39650ea6-011e-51f0-c237-
d19ed520eaf1" policytype="sniffer" action="detected" attack="LLDP-test-mac" attackid=6312
profile="g-sniffer-profile" incidentserialno=216006666 msg="custom: LLDP-test-mac" crscore=30
craction=8192 crlevel="high"
```

The Ethernet protocol detection does not support traffic logging; only an IPS log will be generated if the sniffer policy is matched.

File filter

A file filter can be configured to control the flow of different types of files passing through FortiGate. This is done by setting up rules that specify which file types are allowed or blocked. The file filter can be applied directly to firewall policies and supports various traffic protocols in proxy or flow mode. The feature set setting (proxy or flow) in the file filter profile must match the inspection mode setting (proxy or flow) in the associated firewall policy. For example, a flow-based file filter profile must be used with a flow-based firewall policy.

Protocol	Proxy mode	Flow mode
CIFS	Yes	Yes
FTP	Yes	Yes
HTTP	Yes	Yes

Protocol	Proxy mode	Flow mode
IMAP	Yes	Yes
MAPI	Yes	No
POP3	Yes	Yes
SMTP	Yes	Yes
SSH	Yes	No
WEBSOCKET	Yes	Yes

File filtering is based only on the file type (file meta data) and not on file size or content. A DLP dictionary, sensor, and profile would need to be configured to block files based on size or content, such as SSN numbers, credit card numbers, or regular expressions (see Basic DLP settings on page 2606 for more information). See Supported file types on page 2520 for a list of the supported file types.

The following options can be configured in a file filter profile:

GUI option	CLI option	Description
Basic profile settings		
Name	`name <string>`	Enter a unique name for the profile.
Comments	`comment <var-string>`	Enter a comment (optional).
Scan archive contents	`scan-archive-contents {enable | disable}`	Enable to scan archive contents.

Feature set	`feature-set {flow | proxy}`	Select the feature set for the profile. The feature set mode must match the inspection mode used in the associated firewall policy.
		• *Flow-based*
		• *Proxy-based*
		If the *Feature set* option is not visible in the GUI, enter the following in the CLI:
		`config system settings`
		`set gui-proxy-inspection enable`
		`end`

n/a	`log {enable | disable}`	Enable to use file filter logging. This setting is enabled by default.
n/a	`extended-log {enable | disable}`	Enable to use file filter extended logging. This setting is disabled by default.
n/a	`replacemsg-group <string>`	Set a replacement message group.
File filter rule settings		
Name	`name <string>`	Enter a unique name for the rule.

GUI option	CLI option	Description
Comments	`comment <var-string>`	Enter a comment (optional).
Protocols	`protocol {option1}, {option2}, ...`	Set the protocols to apply to the rule. By default, all protocols are configured: CIFS, FTP, HTTP, IMAP, POP3, and SMTP in flow mode. Additionally, MAPI and SSH are configured by default in proxy mode.
Traffic	`direction {incoming | outgoing | any}`	Set the traffic direction: • *Incoming*/`incoming`: match files transmitted in the session's reply direction. • *Outgoing*/`outgoing`: match files transmitted in the session's originating direction. • *Both*/`any`: match files transmitted in the session's originating and reply directions.
Password-protected only	`password-protected {yes | any}`	Enable (`yes`) to match password-protected files. If the setting is not enabled, any file is matched.
File types	`file-type <name1>, <name2>, ...`	Select the file type. See Supported file types on page 2520 for the list of available options.
Action	`action {log-only | block | warning}`	Set the action to take for a matched file: • *Monitor*/`log-only`: allow the content and write a log message. • *Block*/`block`: block the content and write a log message. • *Warning*/`warning`: allow content after warning the user and write a log message. This option is only available when *Feature set* is *Proxy-based*, and only applies to incoming HTTP traffic.

See the following for file filter profile configuration examples:

- Configuring a file filter profile on page 2523
- Configuring a file filter profile with the warning action NEW on page 2527

Supported file types

File filter allows the FortiGate to block files passing through based on file type based on the file's meta data only, and not on file size or file content. A DLP profile must be configured to block files based on size or content, such as SSN numbers, credit card numbers, or regexp.

The following file types are supported in file filter and DLP profiles:

Type	Description
.net	Match .NET files.
7z	Match 7-zip files.
activemime	Match activemime files.
arj	Match arj compressed files.
aspack	Match aspack files.
avi	Match avi files.
base64	Match base64 files.
bat	Match Windows batch files.
binhex	Match binhex files.
bmp	Match bmp files.
bzip	Match bzip files.
bzip2	Match bzip2 files.
c/cpp	Match c/cpp files.
cab	Match Windows cab files.
chm	Match Windows compiled HTML help files.
class	Match class files.
cod	Match cod files.
crx	Match Chrome extension files.
dll	Match dll files.
dmg	Match Apple disk image files.
elf	Match elf files.
exe	Match Windows executable files.
flac	Match flac files.
fsg	Match fsg files.
genscript	Match generic script files.
gif	Match gif files.
gzip	Match gzip files.
hlp	Match Windows help files.
hta	Match hta files.
html	Match html files.

Type	Description
hwp	Match hwp files.
iso	Match ISO archive files.
jad	Match jad files.
javascript	Match javascript files.
jnlp	Match jnlp files. Only supported in proxy inspection mode.
jpeg	Match jpeg files.
lzh	Match lzh compressed files.
lzip	Match lzip files.
mach-o	Match Mach object files.
mime	Match mime files.
mov	Match mov files.
mp3	Match mp3 files.
mpeg	Match mpeg files.
msi	Match Windows Installer msi files.
msoffice	Match legacy MS-Office files. doc, ppt, and xls.
msofficex	Match MS-Office XML files. docm, docx, pptm, pptx, xlsm, and xlsx.
pdf	Match Acrobat PDF files.
petite	Match petite files.
pfile	Match pfile files.
png	Match png files.
python	Match python files.
rar	Match rar archives.
registry	Match registry files.
rm	Match rm files.
rpm	Match rpm files.
shellscript	Match shell script files.
sis	Match sis files.
sylk	Match Microsoft symbolic link files.
tar	Match tar files.
tiff	Match tiff files.

Type	Description
torrent	Match torrent files.
unknown*	Match unknown files
upx	Match upx files.
uue	Match uue files.
wasm	Match WebAssembly files.
wav	Match wav files.
wma	Match wma files.
xar	Match xar archive files.
xz	Match xz files.
zip	Match zip files.

* This file type is only available in DLP profiles.

Configuring a file filter profile

In this example, a flow-based file filter profile is created that has two rules.

- Rule 1: applied to HTTP, FTP, SMTP, IMAP, POP3, and CIFS to monitor any matched .NET, 7-Zip, ActiveMime, ARJ, ASPack, AVI, Base64, Windows batch, BinHex, BMP, Bzip, and Bzip2 files transmitted in the session's originating and reply directions.
- Rule 2: applied to HTTP, FTP, SMTP, IMAP, POP3, and CIFS to block any matched SIS, TAR. TIFF, torrent, UPX, UUE, WAV, WMA. ZAR archive, XZ, and ZIP files transmitted in the session's originating direction.

To configure a file filter in the GUI:

1. Configure the filter profile:
   a. Go to *Security Profiles* > *File Filter* and click *Create New*.
   b. Enter a name.
   c. Set the *Feature set* to *Flow-based*.
   d. In the *Rules* table, click *Create New*.
   e. Configure rule 1 as follows:

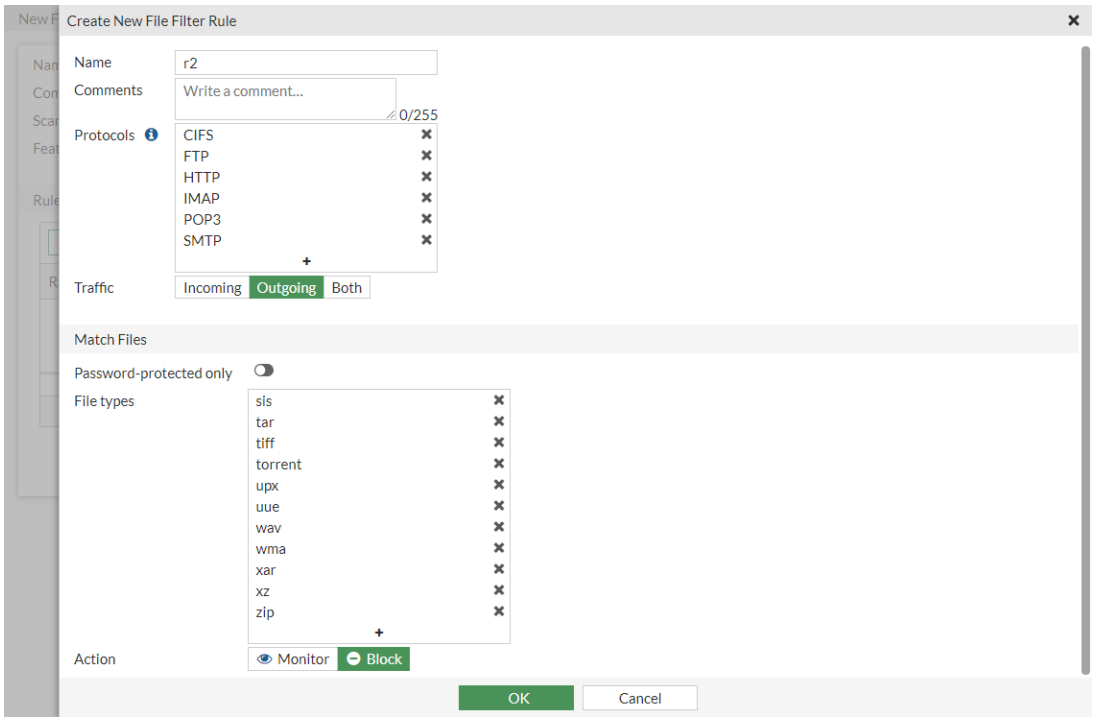
Name	r1
Protocols	HTTP, FTP, SMTP, IMAP, POP3, CIFS
Traffic	Both
Password-protected only	Deselect
File types	.net, 7z, activemime, arj, aspack, avi, base64, bat, binhex, bmp, bzip, bzip2

Action	Monitor

f. Click *OK* to save the rule.

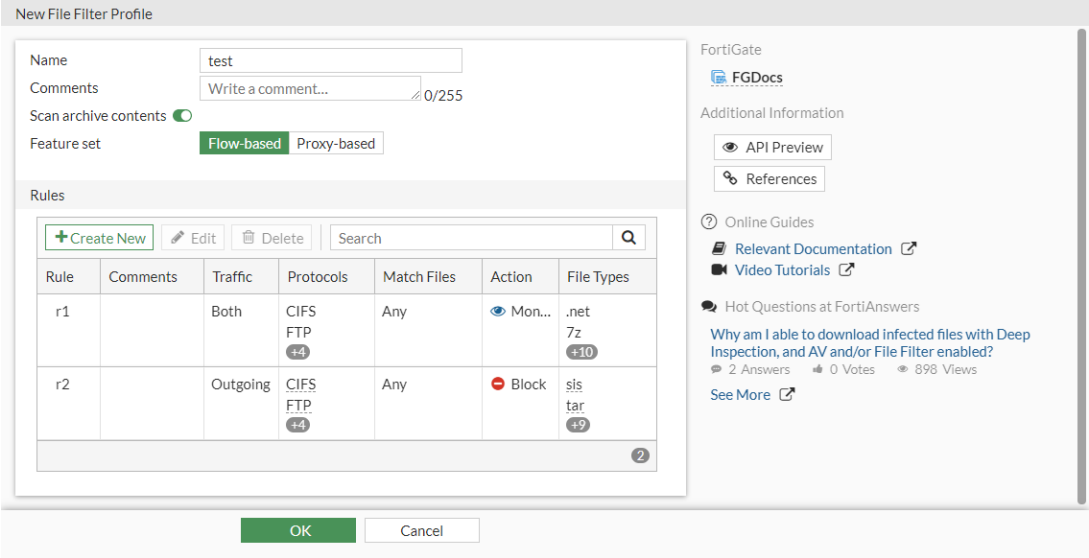
g. In the *Rules* table, click *Create New* and configure rule 2 as follows:

Name	r2
Protocols	HTTP, FTP, SMTP, IMAP, POP3, CIFS
Traffic	Outgoing
Password-protected only	Deselect
File types	sis, tar, tiff, torrent, upx, uue, wav, wma, xar, xz, zip
Action	Block



h. Click *OK* to save the rule.

i. Click *OK* to save the filter profile.



2. Apply the filter to a policy:
   a. Go to *Policy & Objects* > *Firewall Policy* and edit an existing policy or create a new one.
   b. In the *Security Profiles* section, enable *File Filter*.
   c. Select the filter from the dropdown box (*test*).
   d. Configure the other settings as needed.
   e. Click *OK*.

To configure a file filter in the CLI:

1. Configure the file filter profile:

```
config file-filter profile
    edit "test"
        set comment ''
        set feature-set flow
        set replacemsg-group ''
        set log enable
        set scan-archive-contents enable
        config rules
            edit "r1"
                set comment ''
                set protocol http ftp smtp imap pop3 cifs
                set action log-only
                set direction any
                set password-protected any
                set file-type ".net" "7z" "activemime" "arj" "aspack" "avi" "base64" "bat"
"binhex" "bmp" "bzip" "bzip2"
            next
            edit "r2"
                set comment ''
                set protocol http ftp smtp imap pop3 cifs
                set action block
                set direction outgoing
```

```
                set password-protected any
                set file-type "sis" "tar" "tiff" "torrent" "upx" "uue" "wav" "wma" "xar" "xz"
"zip"
            next
        end
    next
end
```

2. Apply the filter to a policy:

```
config firewall policy
    edit 1
        set name "filefilter-policy"
        set srcintf "port10"
        set dstintf "port9"
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set profile-protocol-options "protocol"
        set ssl-ssh-profile "protocols"
        set file-filter-profile "test"
        set auto-asic-offload disable
        set np-acceleration disable
        set nat enable
    next
end
```

To view file filter logs in the GUI:

1. Go to *Log & Report > Security Events*.
2. Select the *File Filter* card.

To view file filter logs in the CLI:

```
# execute log filter category utm-file-filter
# execute log display
```

Log samples

```
date=2020-04-21 time=17:04:02 logid="1900064000" type="utm" subtype="file-filter" eventtype="file-
filter" level="warning" vd="root" eventtime=1587513843211612684 tz="-0700" policyid=1
sessionid=1751 srcip=10.1.100.22 srcport=57382 srcintf="port21" srcintfrole="undefined"
dstip=172.16.200.44 dstport=445 dstintf="port23" dstintfrole="undefined" proto=6 service="CIFS"
profile="filefilter" direction="incoming" action="blocked" rulename="1"
filename="sample\\putty.exe" filesize=454656 filetype="exe" msg="File was blocked by file filter."
```

```
date=2020-04-21 time=17:03:54 logid="1900064000" type="utm" subtype="file-filter" eventtype="file-
filter" level="warning" vd="root" eventtime=1587513834376811325 tz="-0700" policyid=1
sessionid=1742 srcip=10.1.100.22 srcport=36754 srcintf="port21" srcintfrole="undefined"
dstip=172.16.200.44 dstport=22 dstintf="port23" dstintfrole="undefined" proto=6 service="SSH"
subservice="SCP" profile="filefilter" direction="incoming" action="blocked" rulename="1"
filename="test.pdf" filesize=571051 filetype="pdf" msg="File was blocked by file filter."
```

```
date=2020-04-21 time=17:00:30 logid="1900064000" type="utm" subtype="file-filter" eventtype="file-
filter" level="warning" vd="root" eventtime=1587513630482716465 tz="-0700" policyid=1
sessionid=1684 srcip=10.1.100.22 srcport=58524 srcintf="port21" srcintfrole="undefined"
dstip=172.16.200.44 dstport=143 dstintf="port23" dstintfrole="undefined" proto=6 service="IMAP"
profile="filefilter" direction="incoming" action="blocked" from="pc4user1@qa.fortinet.com"
to="pc4user2@qa.fortinet.com" recipient="pc4user2" subject="QA Test" rulename="1"
filename="test.JPG" filesize=48079 filetype="jpeg" msg="File was blocked by file filter."
```

```
date=2020-04-21 time=16:59:58 logid="1900064000" type="utm" subtype="file-filter" eventtype="file-
filter" level="warning" vd="root" eventtime=1587513598866551739 tz="-0700" policyid=1
sessionid=1674 srcip=10.1.100.22 srcport=39854 srcintf="port21" srcintfrole="undefined"
dstip=172.16.200.44 dstport=110 dstintf="port23" dstintfrole="undefined" proto=6 service="POP3"
profile="filefilter" direction="incoming" action="blocked" from="pc4user1@qa.fortinet.com"
to="pc4user2@qa.fortinet.com" recipient="pc4user2" subject="QA Test" rulename="1"
filename="test.JPG" filesize=48079 filetype="jpeg" msg="File was blocked by file filter."
```

```
date=2020-04-21 time=16:58:31 logid="1900064000" type="utm" subtype="file-filter" eventtype="file-
filter" level="warning" vd="root" eventtime=1587513511516745955 tz="-0700" policyid=1
sessionid=1619 srcip=10.1.100.22 srcport=53144 srcintf="port21" srcintfrole="undefined"
dstip=172.16.200.44 dstport=25 dstintf="port23" dstintfrole="undefined" proto=6 service="SMTP"
profile="filefilter" direction="outgoing" action="blocked" from="pc4user1@qa.fortinet.com"
to="pc4user2@qa.fortinet.com" sender="pc4user1@qa.fortinet.com"
recipient="pc4user2@qa.fortinet.com" subject="QA Test" rulename="1" filename="test.PNG"
filesize=65173 filetype="png" msg="File was blocked by file filter."
```

```
date=2020-04-21 time=16:58:14 logid="1900064000" type="utm" subtype="file-filter" eventtype="file-
filter" level="warning" vd="root" eventtime=1587513494608988795 tz="-0700" policyid=1
sessionid=1605 srcip=10.1.100.22 srcport=43186 srcintf="port21" srcintfrole="undefined"
dstip=172.16.200.44 dstport=21 dstintf="port23" dstintfrole="undefined" proto=6 service="FTP"
profile="filefilter" direction="incoming" action="blocked" rulename="1" filename="index.html"
filesize=21 filetype="html" msg="File was blocked by file filter."
```

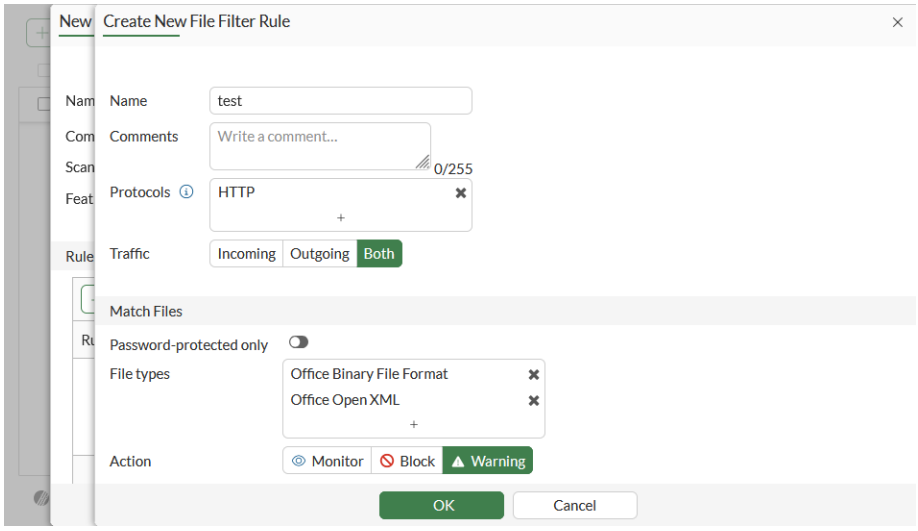
Configuring a file filter profile with the warning action NEW

In this example, a proxy-based file filter profile is created that has one rule. The rule is applied to HTTP traffic to warn about any matched Office Binary File Format or Office Open XML files.

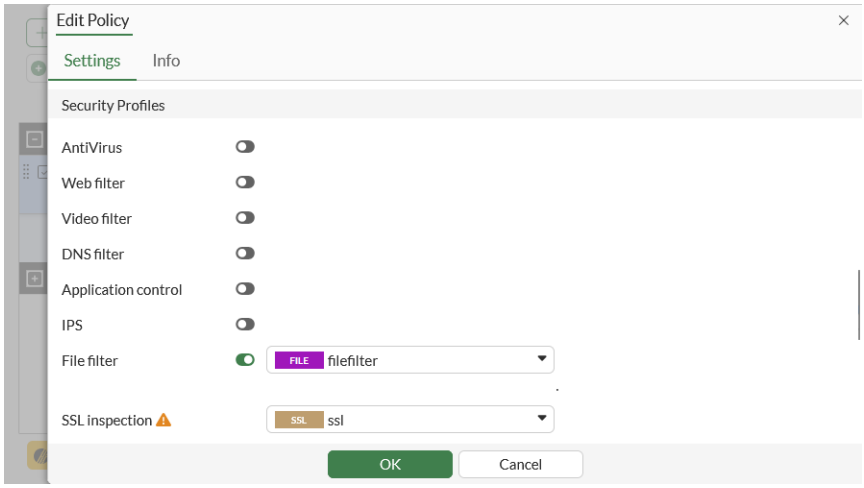
After the user acknowledges the warning, the content is allowed and access is logged. This behavior applies only to incoming HTTP traffic (eg. HTTP GET); outgoing HTTP traffic (eg. HTTP POST) and non-HTTP protocols are log-only, with no warning or blocking when applied to the same profile.

To configure and use a file filter profile with the warning action in the GUI:

1. Go to *Security Profiles* > *File Filter* and click *Create New*.
2. Set *Feature set* to *Proxy-based*.
3. In the *Rules* table, click *Create New*.
4. Set *Action* to *Warning*.
5. Configure the remaining settings as needed, then click *OK*.



6. Configure the remaining filter profile settings as needed, then click *OK*.
7. Go to *Policy & Objects* > *Firewall Policy* and create a new policy or edit an existing one.
8. In the *Security Profiles* section, enable *File filter* and select the file filter profile.



9. Configure the remaining settings as required, then click *OK*.

To configure and use a file filter profile with the warning action in the CLI:

1. Configure a file filter profile with the warning action:

```
config file-filter profile
  edit "filefilter"
        set feature-set proxy
        config rules
            edit "test"
                set protocol http
                set action warning
                set file-type "msoffice" "msofficex"
            next
        end
    next
end
```

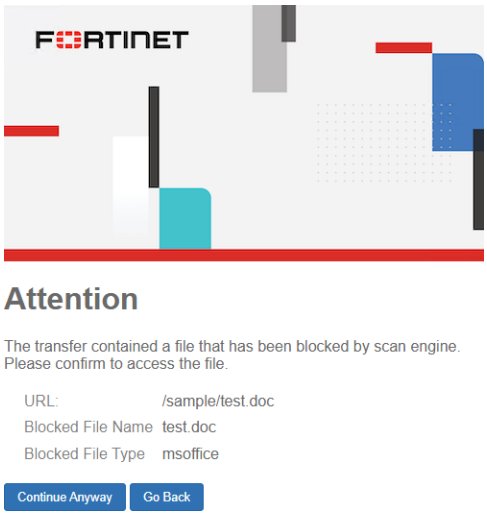
2. Apply the file filter profile to a firewall policy:

```
config firewall policy
    edit 1
        set name "filefilter"
        set srcintf "port1"
        set dstintf "port3"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set profile-protocol-options "protocol"
        set ssl-ssh-profile "ssl"
        set file-filter-profile "filefilter"
        set logtraffic all
        set nat enable
    next
end
```

To test the file filter:

1. On a Windows client, access an absolute URL (http://<iP address>/sample/test.doc) to download file *test.doc*.

   A replacement message is shown that asks the user to authorize or cancel the file download, and a blocking UTM file filter log is generated.

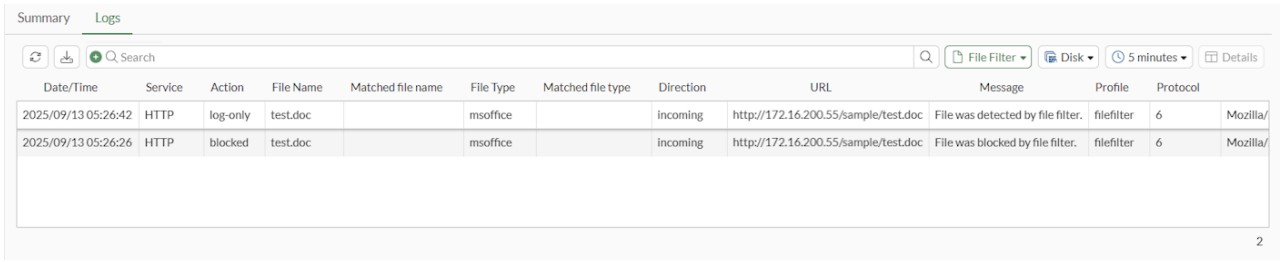



```
2: date=2025-09-13 time=05:26:26 eventtime=1757697986602674439 tz="+1200" logid="1900064000"
type="utm" subtype="file-filter" eventtype="file-filter" level="warning" vd="vdom1" policyid=1
poluuid="61c636d2-5b59-51f0-b555-76579a0b21fa" policytype="policy" sessionid=1583
srcip=10.1.100.195 srcport=51426 srccountry="Reserved" srcintf="port1" srcintfrole="undefined"
srcuuid="1027adc4-5b59-51f0-7577-e6bcd1ba3541" dstip=172.16.200.55 dstport=80
dstcountry="Reserved" dstintf="port3" dstintfrole="undefined" dstuuid="1027adc4-5b59-51f0-
7577-e6bcd1ba3541" proto=6 service="HTTP" profile="filefilter" direction="incoming"
action="blocked" url="http://172.16.200.55/sample/test.doc" hostname="172.16.200.55"
agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/140.0.0.0 Safari/537.36" httpmethod="GET" rulename="test" filename="test.doc"
filesize=19456 filetype="msoffice" msg="File was blocked by file filter."
```

2. Click *Continue Anyway* to download the file. A log-only file filter log is generated:

```
1: date=2025-09-13 time=05:26:42 eventtime=1757698001607832698 tz="+1200" logid="1900064001"
type="utm" subtype="file-filter" eventtype="file-filter" level="notice" vd="vdom1" policyid=1
poluuid="61c636d2-5b59-51f0-b555-76579a0b21fa" policytype="policy" sessionid=1602
srcip=10.1.100.195 srcport=51432 srccountry="Reserved" srcintf="port1" srcintfrole="undefined"
srcuuid="1027adc4-5b59-51f0-7577-e6bcd1ba3541" dstip=172.16.200.55 dstport=80
dstcountry="Reserved" dstintf="port3" dstintfrole="undefined" dstuuid="1027adc4-5b59-51f0-
7577-e6bcd1ba3541" proto=6 service="HTTP" profile="filefilter" direction="incoming"
action="log-only" url="http://172.16.200.55/sample/test.doc" hostname="172.16.200.55"
agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/140.0.0.0 Safari/537.36" httpmethod="GET"
referralurl="http://172.16.200.55/sample/test.doc" rulename="test" filename="test.doc"
filesize=19456 filetype="msoffice" msg="File was detected by file filter."
```

3. Go to *Log & Report* > *Security Events* and view the *File Filter* logs.



Email filter

Email filters can be configured to perform spam detection and filtering. You can customize the default profile, or create your own and apply it to a firewall policy.

Two kinds of filtering can be defined in a single profile, and they will act independent of one another.

Filter options can be organized according to the source of the decision:

- Local options: the FortiGate qualifies the email based on local conditions, such as block/allowlists, banned words, or DNS checks using FortiGuard Antispam.
- FortiGuard-based options: the FortiGate qualifies the email based on the score or verdict returned from FortiGuard Antispam.
- Third-party options: the FortiGate qualifies the email based on information from a third-party source (like an ORB list).

Local and FortiGuard block/allowlists can be enabled and combined in a single profile. When combined, the local block/allowlist has a higher priority than the FortiGuard block list during a decision making process. For example, if a client IP address is blocklisted in the FortiGuard server, but you want to override this decision and allow the IP to pass through the filter, you can define the IP address or subnet in a local block/allowlist with the clear action. Because the information coming from the local list has a higher priority than the FortiGuard service, the email will be considered clean.

Some features of this functionality require a subscription to FortiGuard Antispam.

Protocol comparison between email filter inspection modes

The following table indicates which email filters are supported by their designated inspection modes.

	SMTP	POP3	IMAP	MAPI
Proxy	Yes	Yes	Yes	Yes (1)
Flow	Yes	Yes	Yes	No

1. Not available on FortiGate models with 2 GB RAM or less.

The following topics provide information about email filter profiles:

- Configuring an email filter profile on page 2532
- Local-based filters on page 2532
- FortiGuard-based filters on page 2540
- Third-party-based filters on page 2542
- Filtering order on page 2543
- Protocols and actions on page 2545

- Configuring webmail filtering on page 2546
- Spam email header on page 2547

Configuring an email filter profile

Email filters can be configured to perform spam detection and filtering.

To configure an email filter profile:

1. Go to *Security Profiles* > *Email filter* and click *Create New*.
2. Configure the following settings:

Name		Enter a unique name for the profile.
Comments		Enter a comment (optional).
Enable spam detection and filtering		Enable/disable spam detection and filtering.
	Spam Detection by Protocol	Configure settings for SMTP, POP3, IMAP, and MAPI protocols. See Protocols and actions on page 2545 and Filtering order on page 2543 for more information.
	FortiGuard Spam Filtering	The FortiGate consults FortiGuard servers to help identify spammer IP address or emails, known phishing and spam URLs, known spam email checksums, and others. See FortiGuard-based filters on page 2540 for more information.
	Local Spam Filtering	Enable and configure local spam filters. See Local-based filters on page 2532 for more information.

3. Click *OK*.

Local-based filters

There are six types of local spam filters:

- HELO DNS lookup
- Return email DNS check
- Block/allow list
- Banned words*
- Trusted IP addresses*
- MIME header*

* These filters can only be configured in the CLI.

By default, HELO DNS and return email DNS checks are done before the block/allow list check. In some situations, such as when configuring a block/allow list to clear an email from performing further filtering, configure the following to give precedence to the block/allow list:

```
config emailfilter profile
    edit <name>
        config smtp
            set local-override enable
        next
    end
end
```

HELO DNS lookup and return email DNS checking are not supported while in flow-based inspection mode. See Inspection mode feature comparison on page 2214.

HELO DNS lookup

Whenever a client opens an SMTP session with a server, the client sends a HELO command with the client domain name. The FortiGate takes the domain name specified by the client in the HELO and performs a DNS lookup to determine if the domain exists. If the lookup fails, the FortiGate determines that any emails delivered during the SMTP session are spam. The HELO DNS lookup is only available for SMTP traffic.

Return email DNS check

The FortiGate performs a DNS lookup on the return field. If no such record exists, the email is treated as spam. When return email DNS checking is enabled, the FortiGate takes the domain in the reply-to email address and reply-to domain, and checks the DNS servers to see if there is an A or MX record for the domain. If the domain does not exist, the FortiGate treats the email as spam.

Block/allow list

Block/allow lists can be made from emails or IP subnets to forbid or allow them to send or receive emails. The following table summarizes the configurable options in a block/allow list.

Type	Description	Pattern	Action
IP/Netmask and *IPv6/Netmask*	The FortiGate compares the IP address of the client delivering the email to the addresses in the IP address block/allow list specified in the email filter profile.	The filter is an IP address with a subnet mask.	• *Mark as Reject*: the email is dropped before reaching its destination. • *Mark as Spam*: the email is

Type	Description	Pattern	Action
	If a match is found, the FortiGate takes the action configured for the matching block/allow list entry against all delivered email. By default the `hdrip` setting under `config smtp` is disabled. If enabled, the FortiGate checks all the IP addresses in the header of SMTP email against the specified IP address block/allow list.		allowed through, but it will be tagged with an indicator marking the email as spam. • *Mark as Clear*: the email is allowed to go through to its destination on the assumption that it is not spam.
Recipient Address	The FortiGate compares the sender email address to the contents of the RCPT TO envelope header and To: mail header to the specified pattern. If a match is found, the FortiGate takes the action configured for the matching block/allow list entry.	• *Wildcard*: the filter is an email address with a wildcard symbol in place of the variable characters (such as **.example.com* or *fred@*.com*). • *Regular Expression*: the filter is a regular expression. For example, *^[_a-z0-9-]+(\.[_a-z0-9-]+)*@(example|xmple|examp).(com|org|net)* can be used to filter based on a number of email domain name combinations.	• *Mark as Spam*: the email is allowed through, but it will be tagged with an indicator marking the email as spam. • *Mark as Clear*: the email is allowed to go through to its destination on the assumption that it is not spam.
Sender Address	The FortiGate compares the sender email address to the contents of the MAIL FROM envelope header, From: mail header, and Sender: mail header to the specified pattern. If a match is found, the FortiGate takes the action configured for the matching block/allow list entry.		
Subject	The FortiGate compares the sender email address to the contents of the Subject: mail header to the specified pattern. If a match is found, the FortiGate takes the action configured for the matching block/allow list entry.		

Banned words

When banned word checking is enabled, the FortiGate examines emails for words that appear in the banned word list specified in the email filter profile.

The banned word pattern can be either wildcard or Perl regular expression, which could include part of a word, a whole word, a phrase, multiple words, or multiple phrases.

Each time the banned word filter detects a pattern in an email, it adds the pattern score to the sum of scores for the message. The score is set when creating a new pattern to block content (`set score`). Higher scores indicate more offensive content. If the total score of the discovered banned words in the email exceeds the threshold value set in the email filter profile, then the FortiGate treats the email as spam. The score for each pattern is counted only once, even if that pattern appears many times in the email. The default score for banned word patterns is 10, and the default threshold in the email filter is 10. This means that by default, an email message is blocked by a single match.

For example, if the FortiGate scans an email containing only this sentence: “The score for each word or phrase is counted only once, even if that word or phrase appears many times in the email message.” and the banned word list contains the following patterns:

Banned word pattern	Pattern type	Assigned score	Score added to sum for entire page	Comments
word	Wildcard	20	20	The pattern appears twice, but it is counted once.
word phrase	Wildcard	20	0	Both words appear in the email, but they do not appear together as specified in the pattern. There are no matches.
word*phrase	Wildcard	20	20	A match occurs as long as “word” appears before “phrase” regardless of what is in between them. The pattern appears twice, but it is counted once.
mail*age	Wildcard	20	20	This pattern is a match because “email message” appears in the email.

The email would be treated as spam if the banned word threshold is set to 60 or less.

To apply a banned word filter to an email filter profile:

1. Configure the banned words list:

```
config emailfilter bword
    edit 1
        set name "banned"
        config entries
            edit 23
                set pattern-type {wildcard | regexp}
```

```
                set pattern <string>
                set score <1 - 99999>
            next
        end
    next
end
```

2. Configure the email filter profile:

```
config emailfilter profile
    edit "myBannedWordsProfile"
        set spam-filtering enable
        set options bannedword
        set spam-bword-threshold <0 - 2147483647>
        set spam-bword-table 23
    next
end
```

Once a banned word list is configured in the CLI and applied to an email filter profile, some settings can be edited in the GUI for that particular email filter profile. A banned word profile can be selected, and its *Threshold* (`spam-bword-threshold`) can be edited.

Trusted IP addresses

When the FortiGate creates a list of trusted IP addresses, any incoming email traffic from these IP address is exempt from having IP-based checks, such as DNSBL, RBL, FortiGuard Antispam service, or locally-defined IP block lists.

If the FortiGate sits behind a company's mail transfer units, it may be unnecessary to check email IP addresses because they are internal and trusted. In this case, only external IP addresses would be checked. In some cases, external IP addresses may be added to the list if they are known to not be spam sources.

To configure a trusted IP address list:

1. Define the IP address list:

```
config emailfilter iptrust
    edit 1
        set name "trustedIP"
        config entries
            edit 33
                set addr-type {ipv4 | ipv6}
                set ipv4-subnet <IPv4_classnet>
                set ipv6-subnet <IPv6_network>
            next
        end
    next
end
```

2. Add the list to the email filter profile:

```
config emailfilter profile
    edit "email_filter_profile"
        set spam-iptrust-table 1
    next
end
```

MIME header

This feature filters by the MIME header.

To configure a MIME header check:

1. Define the header content:

```
config emailfilter mheader
    edit 100
        set name "mheader"
        config entries
            edit 1
                set fieldname <string>
                set fieldbody <string>
                set pattern-type {wildcard | regexp}
                set action {spam | clear}
            next
        end
    next
end
```

2. Add the header to the email filter profile:

```
config emailfilter profile
    edit "email_filter_profile"
        set options spamhdrcheck
        set spam-mheader-table 100
    next
end
```

Configuring a local-based email filter

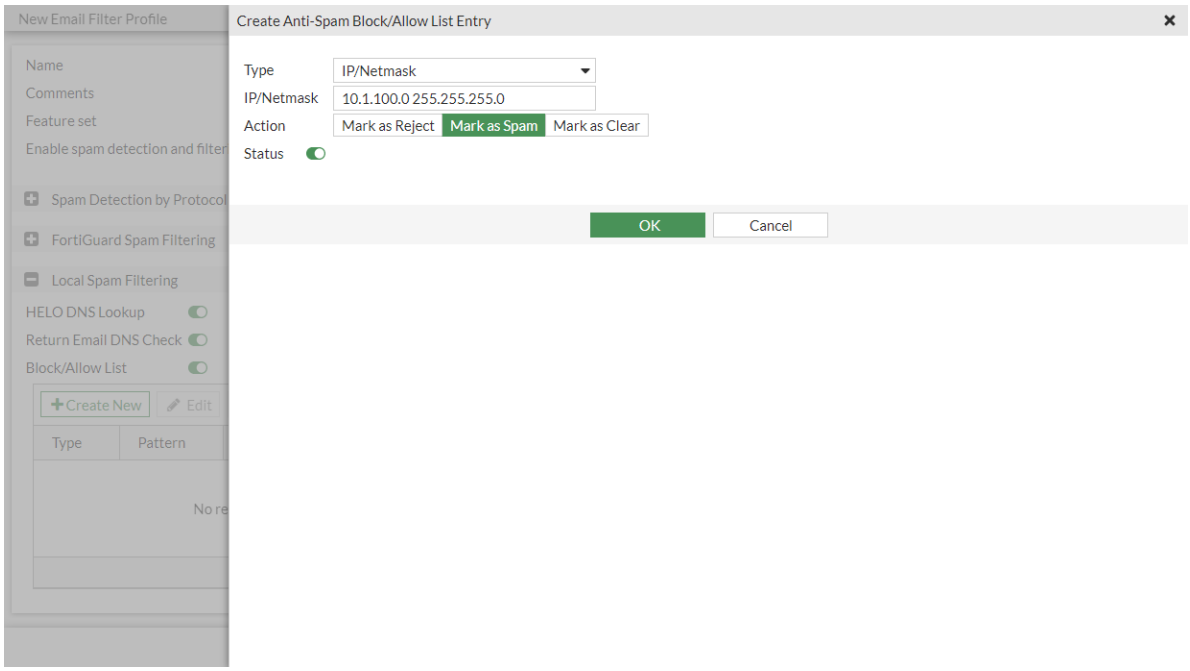
To configure a local-based email filter in the GUI:

1. Configure the email filter profile:
   a. Go to *Security Profiles* > *Email Filter* and click *Create New*, or edit an existing profile.
   b. Select a *Feature set* (*Proxy-based* is used in this example) and enable *Enable spam detection and filtering*.
   c. In the *Local Spam Filtering* section, enable the desired filters (*HELO DNS Lookup*, *Return Email DNS Check*, *Block/Allow List*).

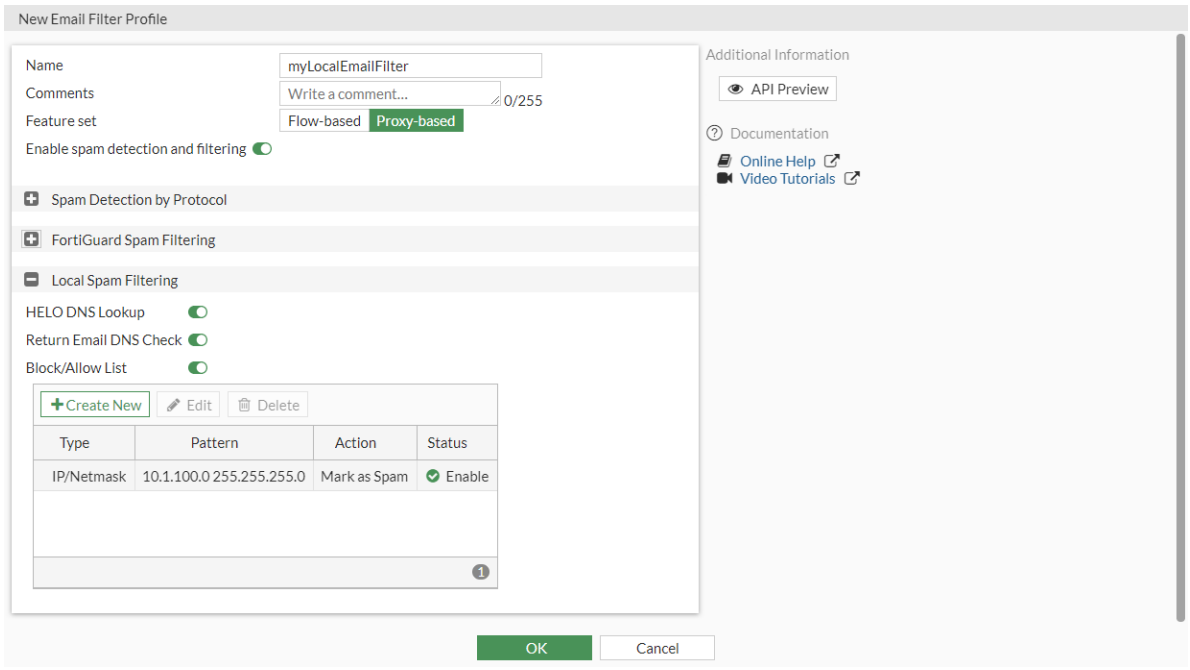
d. In the *Block/Allow List* table, click *Create New*. The *Create Anti-Spam Block/Allow List Entry* pane opens.

e. Set the *Type* to *IP/Netmask* and enter an *IP/Netmask*.

f. Select an *Action*.



g. Click *OK* to save the block/allow list.



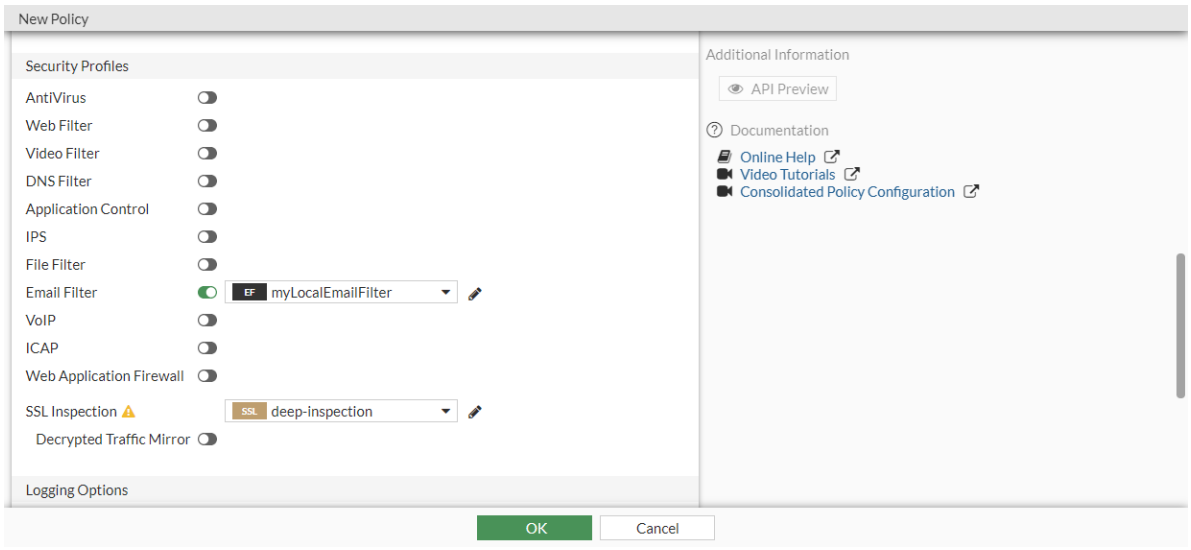
h. Click *OK* save the email filter profile.

2. Configure the firewall policy:

a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*, or edit an existing policy.

b. Set the inspection-mode to *Proxy-based*.

c. Enable the *Email Filter* option and select the previously created profile.



d. Set *SSL Inspection* to a profile that has deep SSL inspection enabled.
Deep inspection is required to filter SMTP, POP3, IMAP, or any SSL/TLS encapsulated protocol.

e. Configure the other settings as needed.

f. Click *OK*.

To configure a local-based email filter in the CLI:

1. Configure a block/allow list:

```
config emailfilter block-allow-list
    edit 1
        set name "myBAL"
        config entries
            edit 1
                set status enable
                set type ip
                set action spam
                set addr-type ipv4
                set ip4-subnet 10.1.100.0 255.255.255.0
            next
        end
    next
end
```

2. Configure an email filter profile:

```
config emailfilter profile
    edit "myLocalEmailFilter"
        set spam-filtering enable
        set options spambal spamhelodns spamraddrdns
        config smtp
            set action tag
        end
```

```
        set spam-bal-table 1
    next
end
```

3. Use the profile in a firewall policy:

```
config firewall policy
    edit 1
        set inspection-mode proxy
        set ssl-ssh-profile "deep-inspection"
        set emailfilter-profile "myLocalEmailFilter"
    next
end
```

FortiGuard-based filters

The FortiGate consults FortiGuard servers to help identify spammer IP address or emails, known phishing URLs, known spam URLs, known spam email checksums, and others. For more information, refer to the FortiGuard website.

There are five FortiGuard spam filtering options:

- IP address check
- URL check
- Detect phishing URLs in email (requires URL check to be enabled)
- Email checksum check
- Spam submission

FortiGuard-based filters are not supported while in flow-based inspection mode. See Inspection mode feature comparison on page 2214.

IP address check

The FortiGate queries the FortiGuard Antispam service to determine if the IP address of the client delivering the email is in the block list. If there is a match, the FortiGate treats delivered emails as spam.

The Spam-Spamming.Server category under the Internet Service Database (ISDB) also provides antispam protection, comprising the most recent and active spam IPs based on FortiGuard telemetry. Users may apply this ISDB address group in a firewall policy and specify the mail service to block. However, this category of the ISDB is only a subset of the IPs in the antispam service. For the most comprehensive spam IP reputation protection, use the IP address check feature.

URL check

The FortiGate submits all URLs that appear in the email body to the FortiGuard service for checking. If a URL exists in the FortiGuard URL block list, the FortiGate treats the email as spam.

Detect phishing URLs in email

The FortiGate submits all URL hyperlinks that appear in the email body to the FortiGuard service for checking. If a URL exists in the FortiGuard URL phishing list, the FortiGate removes the hyperlink from the message. The URL remains in place, but it is no longer a clickable hyperlink.

Email checksum check

The FortiGate submits a checksum of each email to the FortiGuard service for checking. If a checksum exists in the FortiGuard checksum block list, the FortiGate treats the email as spam.

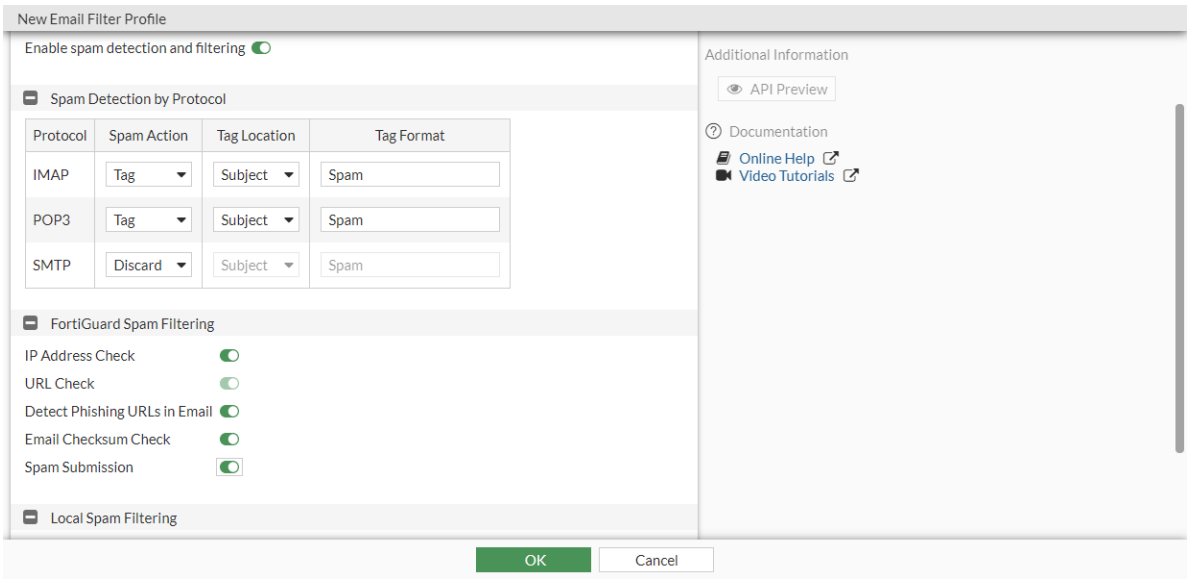
Spam submission

Spam submission is a way to inform the FortiGuard Antispam service of non-spam messages incorrectly marked as spam. When enabled, the FortiGate adds a link to the end of every email marked as spam. Click the link to notify the FortiGuard Antispam service if an email is marked incorrectly.

Configuring FortiGuard filters

To configure FortiGuard filters in the GUI:

1. Go to *Security Profiles* > *Email Filter* and click *Create New*.
2. Enable *Enable spam detection and filtering*.
3. In the *FortiGuard Spam Filtering* section, enable the following as needed:
   - *IP Address Check*
   - *URL Check*
   - *Detect Phishing URLs in Email*
   - *Email Checksum Check*
   - *Spam Submission*



4. Click *OK*.

To configure FortiGuard filters in the CLI:

```
config emailfilter profile
    edit <name>
        set spam-filtering enable
        set options spamfsip spamfsurl spamfsphish spamfschksum spamfssubmit
    next
end
```

Option	Description
`spamfsip`	Check email IP addresses
`spamfsurl`	Check email content URLs
`spamfsphish`	Check email content phishing URLs
`spamfschksum`	Check email checksums
`spamfssubmit`	Add FortiGuard Antispam spam submission text

Third-party-based filters

In addition to local and FortiGuard filters, FortiOS can leverage third-party sources, which are known as DNS-based blackhole lists (DNSBL) or Open Relay Behavior-modification Systems (ORBS). These are maintained lists of IP addresses that have been identified as associated with spamming.

The following example demonstrates how to configure a DNSBL. The `config emailfilter dnsbl` command is used to configure either DNSBL or ORBS.

To configure a DNSBL:

1. Define the server to get the DNSBL list from:

```
config emailfilter dnsbl
    edit 100
        set name "dnsbl"
        config entries
            edit 1
                set status enable
                set server <IP address or server name>
                set action {reject | spam}
            next
        end
    next
end
```

2. Add the DNSBL list to an email filter profile:

```
config emailfilter profile
    edit "email_filter_profile"
        set options spamrbl
        set spam-rbl-table 100
    next
end
```

Filtering order

The FortiGate checks for spam using various filtering techniques. The filtering order used by the FortiGate depends on which mail protocol is used.

Filters requiring a query to a server and a reply (FortiGuard Antispam service and DNSBL/ORDBL) are run simultaneously. To avoid delays, queries are sent while other filters are running. The first reply to trigger a spam action takes effect as soon as the reply is received.

Each spam filter passes the email to the next if no matches or problems are found. If the action in the filter is *Mark as Spam*, the FortiGate tags the email as spam according to the settings in the email filter profile. If the action in the filter is *Mark as Reject*, the email session is dropped. If the action in the filter is *Mark as Clear*, the email is exempt from any remaining filters. For SMTP and SMTPS, if the action is *Discard*, the email is discarded or dropped.

SMTP and SMTPS spam filtering order

The FortiGate scans SMTP and SMTPS email for spam in a specific order, which depends on whether or not the local override feature is enabled. This feature is disabled by default, but enabling it gives priority to local spam filters.

You can enable local override (`set local-override`) in an email filter profile to override SMTP or SMTPS remote checks, which includes checks for IP RBL, IP FortiGuard AntiSpam, and HELO DNS with the locally defined antispam block and/or allow lists.

SMTPS spam filtering is available on FortiGates that support SSL content scanning and inspection.

To configure local override of an antispam filter:

```
config emailfilter profile
    edit <name>
        set spam-filtering enable
        set options spambal spamfsip spamfsurl spamhelodns spamfsphish
        config smtp
            set local-override {enable | disable}
        end
        set spam-bal-table 1
    next
end
```

Local override disabled	Local override enabled
1. HELO DNS lookup, last hop IP check against ORDBL 2. Return email DNS check, FortiGuard email checksum check, FortiGuard URL check, FortiGuard IP address check, phishing URLs detection 3. Last hop IP checks local block/allow list 4. Envelope address checks local block/allow list 5. Headers IPs local block/allow list 6. Headers email address local block/allow list, MIME header checks based on local list of patterns (`mheader`) 7. Banned words (subject first, then body) based on local block/allow list (`bword`)	1. Last hop IP checks local block/allow list 2. Envelope address checks local block/allow list 3. Headers IPs local block/allow list, MIME header checks based on local list of patterns (`mheader`) 4. Headers email address local block/allow list 5. Banned words (subject first, then body) based on local list of patterns (`bword`) 6. HELO DNS lookup, last hop IP check against ORDBL 7. Return email DNS check, FortiGuard email checksum check, FortiGuard URL check, FortiGuard IP address check, phishing URLs detection

IMAP, IMAPS, POP3, and POP3S spam filtering order

The FortiGate scans IMAP, IMAPS, POP3, and POP3S email for spam in the following order:

1. MIME headers check, email address block/allow list check
2. Banned word check on email subject
3. IP block/allow list check
4. Banned word check on email body
5. Return email DNS check, FortiGuard email checksum check, FortiGuard URL check, DNSBL and ORDBL checks

IMAPS and POP3S spam filtering are available on FortiGates that support SSL content scanning and inspection.

Protocols and actions

In an email filter profile, there are options to configure settings for SMTP, POP3, IMAP, and MAPI protocols. For each protocol, you can set an action to either discard (block), tag, or pass the log for that protocol. The action options vary per protocol. For the tag action, the spam email can be tagged with configured text in the subject or header.

Some IMAP clients, such as Outlook Express and Mozilla Thunderbird, do not update the message subject when it is different from the IMAP envelope and the body. As a result, subjects are not tagged in the email list, but they are still tagged in the preview pane.

MAPI is only configurable in the CLI and with the proxy feature set.

To configure protocols in an email filer:

```
config emailfilter profile
    edit <name>
        set feature-set {flow | proxy}
        set spam-filtering enable
        set options {bannedword spambal spamfsip spamfssubmit spamfschksum spamfsurl spamhelodns
spamraddrdns spamrbl spamhdrcheck spamfsphish}
        config smtp
            set log-all {enable | disable}
            set action {pass | tag | discard}
            set tag-type {subject | header | spaminfo}
            set tag-msg <string>
            set hdrip {enable | disable}
            set local-override {enable | disable}
        end
        config imap
            set log-all {enable | disable}
            set action {pass | tag}
            set tag-type {subject | header | spaminfo}
            set tag-msg <string>
        end
        config pop3
            set log-all {enable | disable}
            set action {pass | tag}
            set tag-type {subject | header | spaminfo}
            set tag-msg <string>
        end
        config mapi
            set log-all {enable | disable}
            set action {pass | discard}
        end
    next
end
```

`options ...`	The following options are available: • bannedword: content block. • spambal: block/allow list. • spamfsip: email IP address FortiGuard antispam block list check. • spamfssubmit: add FortiGuard antispam spam submission text. • spamfschksum: email checksum FortiGuard antispam check. • spamfsurl: email content URL FortiGuard antispam check. • spamhelodns: email HELO/EHLO domain DNS check. • spamraddrdns: email return address DNS check. • spamrbl: email DNSBL and ORBL check. • spamhdrcheck: email MIME header check. • spamfsphish: email content phishing URL FortiGuard antispam check.
`tag-type {subject | header | spaminfo}`	Set the tag type: • subject: prepend text to the spam email subject. • header: append a user-defined MIME header to the spam email. • spaminfo: append spam information to the spam email header.
`tag-msg <string>`	Subject text or header added to the spam email.
`hdrip {enable | disable}`	Enable/disable SMTP email header IP checks for spamfsip, spamrbl, and spambal filters.
`local-override {enable | disable}`	Enable/disable local filter to override SMTP remote check result.

For more information, see config emailfilter profile in the FortiOS CLI Reference.

Configuring webmail filtering

You can configure an email filter to detect and log emails sent by Gmail and Hotmail. These interfaces do not use standard email protocols (SMTP, POP3, or IMAP) and use HTTPS instead. However, you can still configure the email filter to detect emails that pass through the FortiGate.

The FortiGate only detects and logs the emails, it does not discard or tag them.

To configure webmail filtering:

```
config emailfilter profile
    edit <name>
        set spam-filtering enable
        config msn-hotmail
            set log-all enable
        end
        config gmail
```

```
            set log-all enable
        end
    next
end
```

Spam email header

The following headers may be added to the x-spaminfo field of email headers:

Header	Description
ipbal, path block/allow ip <IP address>	IP address is contained in the local block/allow list
dnsbl, path block/allow ip <IP address>	Email is contained in configured third-party DNS-based blackhole lists (DNSBL), or Open Relay Behavior-modification Systems (ORBS).
FortiGuard-AntiSpam ip, path block/allow ip <IP address>	IP address is contained in the FortiGuard antispam block/allow list.
FortiGuard-AntiSpam ase,	FortiGuard antispam match (may match using: email checksum, URL check, phishing URL, and others).
email-address, no.<INT> pattern matched	Email address is contained in the local block/allow list.
mime-header, no.<INT> pattern matched	MIME header contains a match for a configured pattern.
bannedword, <listid-hexnum hexnum ... hexnum> bannedword, <hexnum hexnum ... hexnum>	Email contains one or more words listed in the local banned word list.
helo-dns	Whenever a client opens an SMTP session with a server, the client sends an HELO command with the client domain name. The FortiGate takes the domain name specified by the client in the HELO and performs a DNS lookup to determine if the domain exists. If the lookup fails, the FortiGate determines that any emails delivered during the SMTP session are spam.
return-email, domain <domain name> has neither MX nor A record	FortiGate performs a DNS lookup on the return field. If no such record exists, the email is treated as spam.
return-email, domain name <domain name> has invalid syntax	Invalid return email domain name.
return-email, DNS request error for domain <domain name>	Error when resolving domain name.

VoIP solutions

You can configure VoIP profiles to allow SIP and SCCP traffic and to protect your network from SIP- and SCCP-based attacks.

FortiOS includes two preloaded VoIP profiles:

- *default*
- *strict*

You can customize these profiles, or you can create your own and add them to firewall policies that allow VoIP.

VoIP profiles cannot be used in NGFW policy-based mode. See NGFW policy on page 1868 for more information.

The following topics provide information about VoIP profiles:

- General use cases on page 2548
- NAT46 and NAT64 for SIP ALG on page 2552
- SIP message inspection and filtering on page 2560
- SIP ALG and SIP session helper on page 2566
- SIP pinholes on page 2573
- SIP over TLS on page 2575
- Voice VLAN auto-assignment on page 2576
- Scanning MSRP traffic on page 2578

General use cases

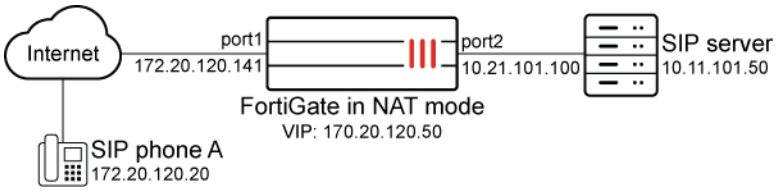
There are three scenarios in which the FortiOS session initiation protocol (SIP) solution is usually deployed:

1. The SIP server is in a private network that is protected from the internet by a FortiGate.
2. The SIP clients are in a private network that is protected from the internet by a FortiGate.
3. The SIP server is in a private network, such as a corporation's internal network or an ISP's network, that is protected from the internet by a FortiGate. The SIP clients are in a remote private network, such as a SOHO network, and behind a NAT device that is not aware of SIP applications.

The following VIP, NAT, and HNT examples show configurations for these common scenarios.

VIP

A FortiGate with SIP Application Layer Gateway (ALG) or SIP session helper protects the SIP server from the internet, while SIP phones from the internet need to register to the SIP server and establish calls through it.



A VIP needs to be configured for the SIP server, and the VIP must be applied in a firewall policy for the phones to send REGISTER messages through the FortiGate from port1 to port2.

Only one firewall policy needs to be configured for all SIP phones on both the internet and private network to register to the SIP server through port1 and set up SIP calls. This example assumes either SIP ALG or SIP session helper is enabled.

To configure the VIP for the SIP server:

```
config firewall vip
    edit "VIP_for_SIP_Server"
        set extip 172.20.120.50
        set extintf "port1"
        set mappedip "10.11.101.50"
    next
end
```

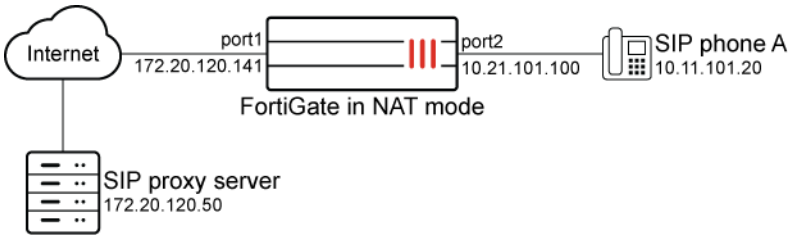
To configure the firewall policy:

```
config firewall policy
    edit 1
        set srcintf "port1"
        set dstintf "port2"
        set srcaddr "all"
        set dstaddr "VIP_for_SIP_Server"
        set action accept
        set schedule "always"
        set service "SIP"
    next
end
```

Setting `service` to `SIP` and not `all` in the firewall policy can improve protection by restricting the data traffic passing through the FortiGate to the SIP call traffic only.

NAT

A FortiGate with SIP ALG or SIP session helper protects the SIP phones and the internal network from the internet, while SIP phones in the internal network need to register to the SIP server installed on the internet and establish calls through it.



One firewall policy needs to be configured with NAT enabled for SIP phones to send REGISTER messages through the FortiGate from port2 to port1. This example assumes either SIP ALG or SIP session helper is enabled.

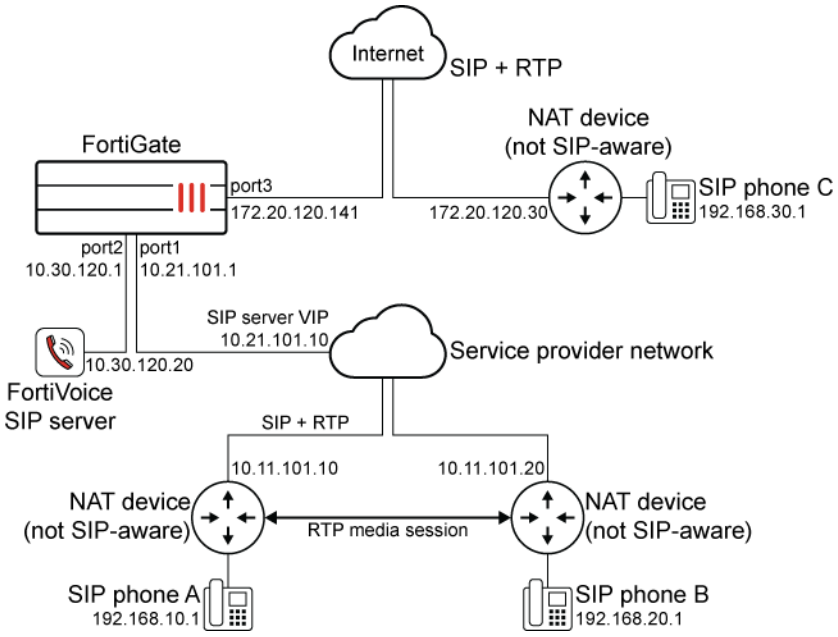
To configure the firewall policy:

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "SIP"
        set nat enable
    next
end
```

HNT

A FortiGate with SIP ALG protects the SIP server from the internet, while SIP phones are in remote private networks behind NAT devices that are not aware of the SIP application.

In this example, the SIP server is located in an ISP's service cloud that is protected by the FortiGate SIP ALG, and the SIP phones are installed in the home networks of the ISP's customers.



The SIP messages traversing the remote NAT devices might have their IP addresses translated by the NAT device at the network layer, but untranslated at the SIP application layer because those NAT devices are not aware of the SIP applications. This causes problems in a SIP session initiated process. Special configurations for the hosted NAT traversal (HNT) are required to resolve this issue.

To configure the FortiGate with HNT support for SIP phones A and B to set up calls with each other:

1. Identify port1 as the external interface:

```
config system interface
    edit "port1"
        set external enable
    next
end
```

2. Configure the VIP for the SIP server:

```
config firewall vip
    edit "VIP_for_SIP_Server"
        set extip 10.21.101.10
        set extintf "port1"
        set mappedip "10.30.120.20"
    next
end
```

3. Configure a VoIP profile with HNT enabled:

```
config voip profile
    edit "hnt"
        config sip
            set hosted-nat-traversal enable
            set hnt-restrict-source-ip enable
```

```
        end
    next
end
```

> `hosted-nat-traversal` must be enabled. `hnt-restrict-source-ip` does not have to be enabled, but can be enabled to restrict the RTP packets' source IP to be the same as the SIP packets' source IP.

4. Apply the VoIP profile and VIP in a firewall policy for phone A and B to register and set up SIP calls through the FortiGate and SIP server:

```
config firewall policy
    edit 1
        set srcintf "port1"
        set dstintf "port2"
        set srcaddr "all"
        set dstaddr "VIP_for_SIP_Server"
        set action accept
        set schedule "always"
        set service "SIP"
        set utm-status enable
        set voip-profile "hnt"
        set nat enable
    next
end
```

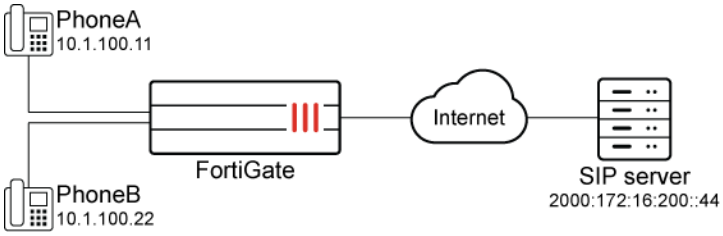
> `nat` must be enabled in the firewall policy.

NAT46 and NAT64 for SIP ALG

NAT46 and NAT64 are supported for SIP ALG. A mix of IPv4 and IPv6 networks can use SIP ALG, allowing for proper call handling.

NAT46 example

In this example, SIP phones on the internal network use IPv4, and the SIP server on an external network uses IPv6. NAT46 is used with SIP ALG to allow for seamless communication. A VoIP profile, `sip`, has already been created.



To configure the FortiGate:

1. Configure a firewall VIP with NAT46 enabled:

```
config firewall vip
    edit "vip46_server_asterisk"
        set extip 10.1.100.100
        set nat44 disable
        set nat46 enable
        set extintf "port1"
        set ipv6-mappedip 2000:172:16:200::44
    next
end
```

2. Configure an IPv6 pool:

```
config firewall ippool6
    edit "client_server_nat46"
        set startip 2000:172:16:200::200
        set endip 2000:172:16:200::207
        set nat46 enable
    next
end
```

3. Configure a firewall policy:

```
config firewall policy
    edit 1
        set name "policy46-1"
        set srcintf "port1"
        set dstintf "port9"
        set action accept
        set nat46 enable
        set srcaddr "all"
        set dstaddr "vip46_server_asterisk"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set voip-profile "sip"
        set logtraffic all
        set auto-asic-offload disable
        set ippool enable
        set poolname6 "client_server_nat46"
    next
end
```

To check the SIP calls and session lists when the phones are registering to the SIP server:

1. View the SIP proxy SIP calls:

```
# diagnose sys sip-proxy calls
sip calls
  vdom 3 (vdom1) vrf 0 call 7f64bf044b00
    call-id: 1513782757
    txn 7f64bf048f00 (REGISTER)
      cseq 2 dir 0 state 5 status 200 expiry 868 HA 0
      i_session: 7f64bf045e00  r_session: 7f64bf045e00
      register: present
      from: sip:2002@10.1.100.100
      to: sip:2002@10.1.100.100
      src: 10.1.100.22:5060
      dst: [2000:172:16:200::44]:5060

  vdom 3 (vdom1) vrf 0 call 7f64bf076700
    call-id: 1490871789
    txn 7f64bf047a00 (REGISTER)
      cseq 2 dir 0 state 5 status 200 expiry 861 HA 0
      i_session: 7f64bf045000  r_session: 7f64bf045000
      register: present
      from: sip:2001@10.1.100.100
      to: sip:2001@10.1.100.100
      src: 10.1.100.11:5060
      dst: [2000:172:16:200::44]:5060
```

2. View the IPv4 session list:

```
# diagnose sys session list

orgin->sink: org pre->post, reply pre->post dev=9->52/52->9 gwy=10.1.100.100/10.1.100.11
hook=pre dir=org act=noop 10.1.100.11:5060->10.1.100.100:5060(0.0.0.0:0)
hook=post dir=reply act=noop 10.1.100.100:5060->10.1.100.11:5060(0.0.0.0:0)
peer=2000:172:16:200::203:65476->2000:172:16:200::44:5060 naf=1
hook=pre dir=org act=noop 2000:172:16:200::203:65476->2000:172:16:200::44:5060(:::0)
hook=post dir=reply act=noop 2000:172:16:200::44:5060->2000:172:16:200::203:65476(:::0)

orgin->sink: org pre->post, reply pre->post dev=9->52/52->9 gwy=10.1.100.100/10.1.100.22
hook=pre dir=org act=noop 10.1.100.22:5060->10.1.100.100:5060(0.0.0.0:0)
hook=post dir=reply act=noop 10.1.100.100:5060->10.1.100.22:5060(0.0.0.0:0)
peer=2000:172:16:200::200:65476->2000:172:16:200::44:5060 naf=1
hook=pre dir=org act=noop 2000:172:16:200::200:65476->2000:172:16:200::44:5060(:::0)
hook=post dir=reply act=noop 2000:172:16:200::44:5060->2000:172:16:200::200:65476(:::0)
```

3. View the IPv4 expectation session list:

```
# diagnose sys session list expectation

orgin->sink: org pre->post, reply pre->post dev=9->0/52->0 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 10.1.100.100:0->10.1.100.11:5060(0.0.0.0:0)
hook=pre dir=org act=noop 0.0.0.0:0->0.0.0.0:0(0.0.0.0:0)
```

```
peer=:::0->:::0 naf=2

orgin->sink: org pre->post, reply pre->post dev=9->0/52->0 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 10.1.100.100:0->10.1.100.22:5060(0.0.0.0:0)
hook=pre dir=org act=noop 0.0.0.0:0->0.0.0.0:0(0.0.0.0:0)
peer=:::0->:::0 naf=2
```

4. View the IPv6 session list:

```
# diagnose sys session6 list

hook=pre dir=org act=noop 2000:172:16:200::203:65476->2000:172:16:200::44:5060(:::0)
hook=post dir=reply act=noop 2000:172:16:200::44:5060->2000:172:16:200::203:65476(:::0)
peer=10.1.100.100:5060->10.1.100.11:5060 naf=2
hook=pre dir=org act=noop 10.1.100.11:5060->10.1.100.100:5060(0.0.0.0:0)
hook=post dir=reply act=noop 10.1.100.100:5060->10.1.100.11:5060(0.0.0.0:0)

hook=pre dir=org act=noop 2000:172:16:200::200:65476->2000:172:16:200::44:5060(:::0)
hook=post dir=reply act=noop 2000:172:16:200::44:5060->2000:172:16:200::200:65476(:::0)
peer=10.1.100.100:5060->10.1.100.22:5060 naf=2
hook=pre dir=org act=noop 10.1.100.22:5060->10.1.100.100:5060(0.0.0.0:0)
hook=post dir=reply act=noop 10.1.100.100:5060->10.1.100.22:5060(0.0.0.0:0)
```

5. View the IPv6 expectation session list:

```
# diagnose sys session6 list expectation

orgin->sink: org pre->post, reply pre->post dev=17->0/52->0
hook=post dir=org act=noop 2000:172:16:200::44:0->2000:172:16:200::200:65476(:::0)
hook=pre dir=org act=noop :::0->:::0(:::0)
peer=10.1.100.100:0->10.1.100.22:5060 naf=1

orgin->sink: org pre->post, reply pre->post dev=17->0/52->0
hook=post dir=org act=noop 2000:172:16:200::44:0->2000:172:16:200::203:65476(:::0)
hook=pre dir=org act=noop :::0->:::0(:::0)
peer=10.1.100.100:0->10.1.100.11:5060 naf=1
```

To check the SIP calls and session lists when one phone is calling another phone:

1. View the SIP proxy SIP calls:

```
# diagnose sys sip-proxy calls

sip calls
  vdom 3 (vdom1) vrf 0 call 7f64bf057a00
    call-id: 217ac4733f80ac766c7e0f3a69d317a1@[2000:172:16:200::44]:5060
    txn 7f64bf038800 (INVITE)
      cseq 103 dir 1 state 11 status 200 expiry 252 HA 0
      i_session: 7f64bf036500  r_session: 7f64bf036500
      register: not-present
      contact[0]:  factory 7f64bf057900/4 expectation 7f64bf02cf00/2 session 7f64bf036500
      contact[1]:  factory 7f64bf057700/3 expectation 7f64bf02ca00/3 session 7f64bf036500
```

```
    from: sip:2001@[2000:172:16:200::44]
    to: sip:2002@[2000:172:16:200::200]:65476;o=10.1.100.22;line=28c59e086cac7c9
    src: [2000:172:16:200::44]:5060
    dst: 10.1.100.22:5060

vdom 3 (vdom1) vrf 0 call 7f64bf057a00
  call-id: 217ac4733f80ac766c7e0f3a69d317a1@[2000:172:16:200::44]:5060
  txn 7f64bf038100 (INVITE)
    cseq 102 dir 1 state 11 status 200 expiry 252 HA 0
    i_session: 7f64bf036500  r_session: 7f64bf036500
    register: not-present
    contact[0]:  factory 7f64bf057900/4 expectation 7f64bf02cf00/2 session 7f64bf036500
    contact[1]:  factory 7f64bf057700/3 expectation 7f64bf02ca00/3 session 7f64bf036500
    from: sip:2001@[2000:172:16:200::44]
    to: sip:2002@[2000:172:16:200::200]:65476;o=10.1.100.22;line=28c59e086cac7c9
    src: [2000:172:16:200::44]:5060
    dst: 10.1.100.22:5060

vdom 3 (vdom1) vrf 0 call 7f64bf057600
  call-id: 1876706695
  txn 7f64bf037300 (REGISTER)
    cseq 2 dir 0 state 5 status 200 expiry 856 HA 0
    i_session: 7f64bf036500  r_session: 7f64bf036500
    register: present
    from: sip:2002@10.1.100.100
    to: sip:2002@10.1.100.100
    src: 10.1.100.22:5060
    dst: [2000:172:16:200::44]:5060

vdom 3 (vdom1) vrf 0 call 7f64bf057400
  call-id: 1372246794
  txn 7f64bf035e00 (REGISTER)
    cseq 2 dir 0 state 5 status 200 expiry 853 HA 0
    i_session: 7f64bf035000  r_session: 7f64bf035000
    register: present
    from: sip:2001@10.1.100.100
    to: sip:2001@10.1.100.100
    src: 10.1.100.11:5060
    dst: [2000:172:16:200::44]:5060

vdom 3 (vdom1) vrf 0 call 7f64bf057800
  call-id: 16530657
  txn 7f64bf038f00 (INVITE)
    cseq 102 dir 1 state 11 status 200 expiry 252 HA 0
    i_session: 7f64bf035000  r_session: 7f64bf035000
    register: not-present
    contact[0]:  factory 7f64bf057900/4 expectation 7f64bf02cc80/2 session 7f64bf035000
    contact[1]:  factory 7f64bf057500/3 expectation 7f64bf02c780/3 session 7f64bf035000
    from: sip:2002@[2000:172:16:200::44]
    to: sip:2001@[2000:172:16:200::44]
    src: [2000:172:16:200::44]:5060
    dst: 10.1.100.11:5060
```

```
vdom 3 (vdom1) vrf 0 call 7f64bf057800
  call-id: 16530657
  txn 7f64bf037a00 (INVITE)
    cseq 21 dir 0 state 11 status 200 expiry 252 HA 0
    i_session: 7f64bf035000  r_session: 7f64bf035000
    register: not-present
    contact[0]:  factory 7f64bf057500/3 expectation 7f64bf02c780/3 session 7f64bf035000
    contact[1]:  factory 7f64bf057900/4 expectation 7f64bf02cc80/2 session 7f64bf035000
    from: sip:2001@10.1.100.100
    to: sip:2002@10.1.100.100
    src: 10.1.100.11:5060
    dst: [2000:172:16:200::44]:5060
```

2. View the IPv6 session list:

```
# diagnose sys session6 list

hook=pre dir=org act=noop 2000:172:16:200::203:17078->2000:172:16:200::44:17090(:::0)
hook=post dir=reply act=noop 2000:172:16:200::44:17090->2000:172:16:200::203:17078(:::0)
peer=10.1.100.100:17090->10.1.100.11:17078 naf=2

hook=pre dir=org act=noop 2000:172:16:200::200:17078->2000:172:16:200::44:17082(:::0)
hook=post dir=reply act=noop 2000:172:16:200::44:17082->2000:172:16:200::200:17078(:::0)
peer=10.1.100.100:17082->10.1.100.22:17078 naf=2
hook=pre dir=org act=noop 10.1.100.22:17078->10.1.100.100:17082(0.0.0.0:0)
hook=post dir=reply act=noop 10.1.100.100:17082->10.1.100.22:17078(0.0.0.0:0)

hook=pre dir=org act=noop 2000:172:16:200::203:65476->2000:172:16:200::44:5060(:::0)
hook=post dir=reply act=noop 2000:172:16:200::44:5060->2000:172:16:200::203:65476(:::0)
peer=10.1.100.100:5060->10.1.100.11:5060 naf=2
hook=pre dir=org act=noop 10.1.100.11:5060->10.1.100.100:5060(0.0.0.0:0)
hook=post dir=reply act=noop 10.1.100.100:5060->10.1.100.11:5060(0.0.0.0:0)

hook=pre dir=org act=noop 2000:172:16:200::200:65476->2000:172:16:200::44:5060(:::0)
hook=post dir=reply act=noop 2000:172:16:200::44:5060->2000:172:16:200::200:65476(:::0)
peer=10.1.100.100:5060->10.1.100.22:5060 naf=2
hook=pre dir=org act=noop 10.1.100.22:5060->10.1.100.100:5060(0.0.0.0:0)
hook=post dir=reply act=noop 10.1.100.100:5060->10.1.100.22:5060(0.0.0.0:0)
```

3. View the IPv6 expectation session list:

```
# diagnose sys session6 list expectation

hook=post dir=org act=noop 2000:172:16:200::44:0->2000:172:16:200::203:65476(:::0)
hook=pre dir=org act=noop :::0->:::0(:::0)
peer=10.1.100.100:0->10.1.100.11:5060 naf=1
```

4. View the IPv4 session list:

```
# diagnose sys session list

orgin->sink: org pre->post, reply pre->post dev=9->52/52->9 gwy=10.1.100.100/10.1.100.22
```

```
hook=pre dir=org act=noop 10.1.100.22:17078->10.1.100.100:17082(0.0.0.0:0)
hook=post dir=reply act=noop 10.1.100.100:17082->10.1.100.22:17078(0.0.0.0:0)
peer=2000:172:16:200::200:17078->2000:172:16:200::44:17082 naf=1
hook=pre dir=org act=noop 2000:172:16:200::200:17078->2000:172:16:200::44:17082(:::0)
hook=post dir=reply act=noop 2000:172:16:200::44:17082->2000:172:16:200::200:17078(:::0)

orgin->sink: org pre->post, reply pre->post dev=9->52/52->9 gwy=10.1.100.100/10.1.100.22
hook=pre dir=org act=noop 10.1.100.22:5060->10.1.100.100:5060(0.0.0.0:0)
hook=post dir=reply act=noop 10.1.100.100:5060->10.1.100.22:5060(0.0.0.0:0)
peer=2000:172:16:200::200:65476->2000:172:16:200::44:5060 naf=1
hook=pre dir=org act=noop 2000:172:16:200::200:65476->2000:172:16:200::44:5060(:::0)
hook=post dir=reply act=noop 2000:172:16:200::44:5060->2000:172:16:200::200:65476(:::0)

orgin->sink: org pre->post, reply pre->post dev=9->52/52->9 gwy=10.1.100.100/10.1.100.11
hook=pre dir=org act=noop 10.1.100.11:5060->10.1.100.100:5060(0.0.0.0:0)
hook=post dir=reply act=noop 10.1.100.100:5060->10.1.100.11:5060(0.0.0.0:0)
peer=2000:172:16:200::203:65476->2000:172:16:200::44:5060 naf=1
hook=pre dir=org act=noop 2000:172:16:200::203:65476->2000:172:16:200::44:5060(:::0)
hook=post dir=reply act=noop 2000:172:16:200::44:5060->2000:172:16:200::203:65476(:::0)
```

5. View the IPv4 expectation session list:

```
# diagnose sys session list expectation

orgin->sink: org pre->post, reply pre->post dev=9->0/52->0 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 10.1.100.100:0->10.1.100.11:5060(0.0.0.0:0)
hook=pre dir=org act=noop 0.0.0.0:0->0.0.0.0:0(0.0.0.0:0)
peer=:::0->:::0 naf=2

orgin->sink: org pre->post, reply pre->post dev=9->0/52->0 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 10.1.100.100:0->10.1.100.22:17078(0.0.0.0:0)
peer=:::0->:::0 naf=2

orgin->sink: org pre->post, reply pre->post dev=9->0/52->0 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 10.1.100.100:0->10.1.100.22:17079(0.0.0.0:0)
peer=:::0->:::0 naf=2

orgin->sink: org pre->post, reply pre->post dev=9->0/52->0 gwy=0.0.0.0/0.0.0.0
hook=post dir=org act=noop 10.1.100.22:0->10.1.100.100:17083(0.0.0.0:0)
peer=2000:172:16:200::200:17085->2000:172:16:200::44:17903 naf=1
```

Log messages

When the phones are registering to the SIP server:

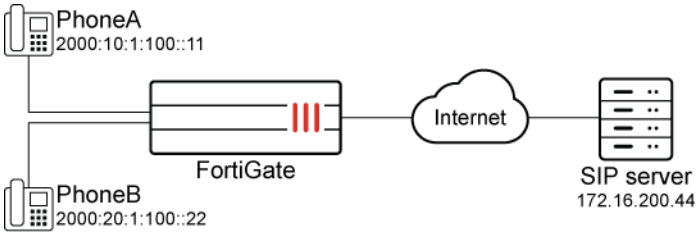
```
date=2022-02-17 time=16:44:47 eventtime=1645145087805236720 tz="-0800" logid="0814044032"
type="utm" subtype="voip" eventtype="voip" level="information" vd="vdom1" session_id=924 epoch=0
event_id=9 srcip=10.1.100.11 src_port=5060 dstip=2000:172:16:200::44 dst_port=5060 proto=17 src_
int="port1" dst_int="port9" policy_id=1 profile="sip" voip_proto="sip" kind="register"
action="permit" status="authentication-required" duration=0 dir="session_origin" call_
id="1868762230" from="sip:2001@10.1.100.100" to="sip:2001@10.1.100.100"
```

When one phone is calling another phone:

```
date=2022-02-17 time=16:44:53 eventtime=1645145093351288241 tz="-0800" logid="0814044032"
type="utm" subtype="voip" eventtype="voip" level="information" vd="vdom1" session_id=924 epoch=0
event_id=11 srcip=10.1.100.11 src_port=5060 dstip=2000:172:16:200::44 dst_port=5060 proto=17 src_
int="port1" dst_int="port9" policy_id=1 profile="sip" voip_proto="sip" kind="call" action="permit"
status="start" duration=0 dir="session_origin" call_id="133636365" from="sip:2001@10.1.100.100"
to="sip:2002@10.1.100.100"
```

NAT64 example

In this example, SIP phones on the internal network use IPv6, and the SIP server on an external network uses IPv4. NAT64 is used with SIP ALG to allow for seamless communication. A VoIP profile, `sip`, has already been created.



To configure the FortiGate:

1. Configure a firewall VIP with NAT64 enabled:

```
config firewall vip
    edit "vip64-1-asterisk"
        set extip 2000:10:1:100::100
        set nat66 disable
        set nat64 enable
        set ipv4-mappedip 172.16.200.44
    next
end
```

2. Configure an IP pool:

```
config firewall ippool
    edit "client_server_nat46"
        set startip 172.16.200.2
        set endip 172.16.200.3
        set nat64 enable
    next
end
```

3. Configure a firewall policy:

```
config firewall policy
    edit 1
        set name "policy64-1"
        set srcintf "port1"
        set dstintf "port9"
```

```
        set action accept
        set nat64 enable
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "vip64-1-asterisk"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set voip-profile "sip"
        set logtraffic all
        set auto-asic-offload disable
        set ippool enable
        set poolname "client_server_nat64"
    next
end
```

SIP message inspection and filtering

There are two types of VoIP profiles that can be configured:

```
config voip profile
    edit <name>
        set feature-set {ips | voipd}
    next
end
```

`feature-set {ips | voipd}`	Set the inspection feature set. • `ips`: use the IPS Engine feature set for the `ips-voip-filter` firewall policy option. • `voipd`: use the SIP ALG feature set for `voip-profile` firewall policy option.

SIP ALG provides users with security features to inspect and control SIP messages that are transported through the FortiGate, including:

- Verifying the SIP message syntax.
- Blocking particular types of SIP requests.
- Restricting the rate of particular SIP requests.

Proxy-based SIP ALG (`feature-set voipd`) is also able to handle features such as pin hole creation and NAT that flow-based SIP inspection cannot. Flow-based SIP (`feature-set ips`) can handle features such as MSRP decoding and scanning that proxy-based SIP ALG cannot.

The two VoIP profile types can be configured separately or at the same time on a firewall policy:

```
config firewall policy
    edit 1
        set voip-profile "voip_sip_alg"
```

```
        set ips-voip-filter "voip_sip_ips"
    next
end
```

Where:

- `voip-profile` can select a `voip-profile` with `feature-set voipd`.
- `ips-voip-filter` can select a `voip-profile` with `feature-set ips`.

The IPS-based VoIP profile (`ips-voip-filter`) allows flow-based SIP to complement SIP ALG while working together.

When both SIP ALG and SIP IPS are used and configured with same block rules, SIP IPS will take priority and do the blocking.

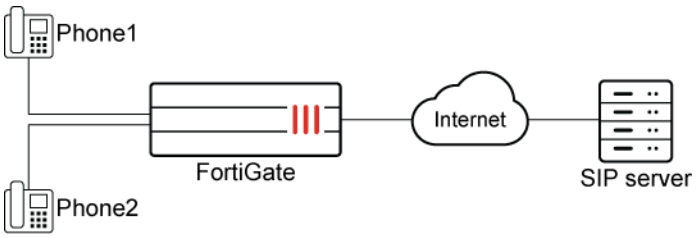
Unlike previous versions (7.0 and 7.2.0-7.2.4) where the firewall policy's inspection mode determines whether the SIP traffic is scanned by SIP ALG or flow-based SIP, the inspection mode does not matter in this version. A `voipd`-based VoIP profile will activate SIP ALG inspection, while an `ips`-based VoIP profile will activate IPS-based SIP inspection.

A `voip-profile` can be selected regardless of the `inspection-mode` used in the firewall policy.

For more information about the difference between SIP ALG and the SIP session helper, see SIP ALG and SIP session helper on page 2566.

Example

In this example, SIP ALG is required for pinhole creation, handling NAT, and controlling SIP messages that requires flow-based SIP. The administrator needs to configure two SIP profiles, one with each feature set (`voipd` and `ips`), and apply these SIP profiles in the same firewall policy.



To configure SIP ALG with SIP IPS:

1. Configure the VoIP profiles:

```
config voip profile
    edit "voip_sip_alg"
        set feature-set voipd
        set comment "sip_alg_simple"
        config sip
            set log-violations enable
            set log-call-summary enable
```

```
        end
    next
    edit "voip_sip_ips"
        set feature-set ips
        set comment "ips_voip_blocking"
        config sip
            set block-invite enable
            set log-violations enable
        end
    next
end
```

2. Configure the firewall policy:

```
config firewall policy
    edit 1
        set srcintf "port1"
        set dstintf "port9"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ips-sensor "g-default"
        set voip-profile "voip_sip_alg"
        set ips-voip-filter "voip_sip_ips"
        set logtraffic all
        set nat enable
    next
end
```

To verify the SIP proxy SIP calls:

1. Verify the register request:

```
# diagnose sys sip-proxy calls
sip calls
  vdom 1 (vdom1) vrf 0 call 7f2b99828300
    call-id: 619216389
    txn 7f2b998ad600 (REGISTER)
      cseq 2 dir 0 state 5 status 200 expiry 527 HA 0
      i_session: 7f2b998aac00  r_session: 7f2b998aac00
      register: present
      from: sip:2001@172.16.200.44
      to: sip:2001@172.16.200.44
      src: 10.1.100.11:5060
      dst: 172.16.200.44:5060
```

2. Verify the invite request:

```
# diagnose sys sip-proxy calls
sip calls
  vdom 1 (vdom1) vrf 0 call 7f2b99828300
    call-id: 619216389
    txn 7f2b998ad600 (REGISTER)
      cseq 2 dir 0 state 5 status 200 expiry 316 HA 0
      i_session: 7f2b998aac00  r_session: 7f2b998aac00
      register: present
      from: sip:2001@172.16.200.44
      to: sip:2001@172.16.200.44
      src: 10.1.100.11:5060
      dst: 172.16.200.44:5060
```

Sample logs

Register request:

```
date=2023-01-13 time=09:46:03 eventtime=1673631963477298677 tz="-0800" logid="0814044032"
type="utm" subtype="voip" eventtype="voip" level="information" vd="vdom1" session_id=17092 epoch=0
event_id=1 srcip=10.1.100.11 src_port=5060 dstip=172.16.200.44 dst_port=5060 proto=17 src_
int="port1" dst_int="port9" policy_id=1 profile="voip_sip_alg" voip_proto="sip" kind="register"
action="permit" status="succeeded" duration=0 dir="session_origin" call_id="619216389"
from="sip:2001@172.16.200.44" to="sip:2001@172.16.200.44"
```

Invite request:

```
date=2023-01-13 time=09:54:43 eventtime=1673632484065549240 tz="-0800" logid="0814044033"
type="utm" subtype="voip" eventtype="voip" level="notice" vd="vdom1" session_id=17092 epoch=0
event_id=0 srcip=10.1.100.11 src_port=5060 dstip=172.16.200.44 dst_port=5060 proto=17 src_
int="port1" dst_int="port9" policy_id=1 profile="voip_sip_ips" voip_proto="sip" kind="call"
action="block" status="N/A" reason="block-request" duration=0 dir="session_reverse" message_
type="request" request_name="INVITE" call_id="1967779864" count=0 from="<sip:2001@172.16.200.44>"
to="<sip:2002@172.16.200.44>" attackid=50083 attack="SIP.Invite.Method"
```

SIP message syntax inspection

For syntax verification, the following attributes are available for configuration in the VoIP profile to determine what action is taken when a specific syntax error or attack based on invalid syntax is detected. For example, the action can be set to pass or discard it.

```
malformed-request-line
malformed-header-via
malformed-header-from
malformed-header-to
malformed-header-call-id
malformed-header-cseq
malformed-header-rack
malformed-header-rseq
malformed-header-contact
malformed-header-record-route
```

```
malformed-header-route
malformed-header-expires
malformed-header-content-type
malformed-header-content-length
malformed-header-max-forwards
malformed-header-allow
malformed-header-p-asserted-identity
malformed-header-sdp-v
malformed-header-sdp-o
malformed-header-sdp-s
malformed-header-sdp-i
malformed-header-sdp-c
malformed-header-sdp-b
malformed-header-sdp-z
malformed-header-sdp-k
malformed-header-sdp-a
malformed-header-sdp-t
malformed-header-sdp-r
malformed-header-sdp-m
malformed-header-no-require*
malformed-header-no-proxy-require*
```

* = only available in flow mode

SIP message blocking

The following options are available in the VoIP profile to block SIP messages:

```
block-long-lines
block-unknown
block-ack
block-bye
block-cancel
block-info
block-invite
block-message
block-notify
block-options
block-prack
block-publish
block-refer
block-register
block-subscribe
block-update
block-geo-red-options**
```

** = only available in proxy mode

SIP message rate limiting

The rate of certain types of SIP requests that are passing through the SIP ALG can be restricted:

```
register-rate
invite-rate
subscribe-rate
message-rate
notify-rate
refer-rate
update-rate
options-rate
ack-rate
prack-rate
info-rate
publish-rate
bye-rate
cancel-rate
```

Additionally, flow-based SIP supports the following rate tracking features:

```
register-rate-track none
invite-rate-track none
subscribe-rate-track none
message-rate-track none
notify-rate-track none
refer-rate-track none
update-rate-track none
options-rate-track none
ack-rate-track none
prack-rate-track none
info-rate-track none
publish-rate-track none
bye-rate-track none
cancel-rate-track none
```

Call-Id and Content-Type regex

When the `ips` VoIP profile feature set is selected, options for Call-Id and Content-Type header values can be configured.

```
config voip profile
    edit <name>
        config sip
            set call-id-regex <string>
            set call-id-regex <string>
        end
    next
end
```

`call-id-regex <string>`	Enter a validation PCRE regular expression for the Call-Id header value.
`call-id-regex <string>`	Enter a validation PCRE regular expression for the Content-Type header value.

SIP ALG and SIP session helper

The SIP session helper is a legacy solution that provides basic support for SIP calls passing through the FortiGate by opening SIP and RTP pinholes, and by performing NAT of the addresses in SIP messages.

SIP Application Layer Gateway (ALG) provides the same basic SIP support as the SIP session helper. In addition, SIP ALG provides a wide range of features that protect your network from SIP attacks, apply rate limiting to SIP sessions, check the syntax of SIP and SDP content of SIP messages, and provide detailed logging and reporting of SIP activity.

By default, all SIP traffic is processed by the SIP ALG. If the policy that accepts the SIP traffic includes a VoIP profile, the SIP traffic is processed by that profile. If the policy does not include a VoIP profile, the SIP traffic is processed by the SIP ALG using the default VoIP profile.

To change between SIP ALG mode and SIP session helper mode:

```
config system settings
    set default-voip-alg-mode {proxy-based | kernel-helper-based}
end
```

`default-voip-alg-mode {proxy-based | kernel-helper-based}`	Set how the FortiGate handles VoIP traffic when a policy that accepts the traffic does not include a VoIP profile. • `proxy-based`: use SIP ALG to process SIP traffic (default). • `kernel-helper-based`: use the SIP session helper to process SIP traffic.

The `default-voip-alg-mode` setting works together with the VoIP profile configured in a firewall policy to determine whether SIP ALG, SIP ALG with IPS SIP, or the SIP session helper are used to process the SIP traffic. The following firewall policy settings correspond to the VoIP profiles (see also SIP message inspection and filtering on page 2560).

```
config firewall policy
    edit <id>
        set voip-profile <voipd-based_profile>
        set ips-voip-filter <ips-based_profile>
    next
end
```

The following table explains the results of configuring different combinations of the preceding settings.

Firewall policy setting		Default VoIP ALG mode setting	
voip-profile	**ips-voip-filter**	**kernel-helper-based**	**proxy-based**
Yes	Yes	SIP ALG + IPS SIP	SIP ALG + IPS SIP
Yes	No	SIP ALG	SIP ALG
No	Yes	SIP ALG + IPS SIP	SIP ALG + IPS SIP
No	No	SIP session helper	SIP ALG

SIP ALG configurations

SIP ALG can be enabled in several ways. The following configuration examples demonstrate different settings.

Example 1

In this example, a `voipd`-based profile is configured and applied to a firewall policy. The `default-voip-alg-mode` remains as the default setting (`proxy-based`).

To configure SIP ALG:

1. Configure the default VoIP ALG mode:

```
config system settings
    set default-voip-alg-mode proxy-based
end
```

2. Configure the VoIP profile:

```
config voip profile
    edit "sip-alg-profile"
        set feature-set voipd
        config sip
            set status enable
        end
    next
end
```

3. Configure the firewall policy:

```
config firewall policy
    edit 0
        set name "VoIP-Proxy"
        set utm-status enable
        set voip-profile "sip-alg-profile"
    next
end
```

Example 2

In this example, the `default-voip-alg-mode` is set to `kernel-helper-based`. A VoIP profile (VoIP-Proxy) has SIP enabled and is applied to a firewall policy.

To configure SIP ALG:

1. Configure the default VoIP ALG mode:

```
config system settings
    set default-voip-alg-mode kernel-helper-based
end
```

2. Configure the VoIP profile:

```
config voip profile
    edit "sip-alg-profile"
        set feature-set voipd
        config sip
            set status enable
        end
    next
end
```

3. Configure the firewall policy:

```
config firewall policy
    edit 0
        set name "VoIP-Proxy"
        set utm-status enable
        set voip-profile "sip-alg-profile"
    next
end
```

Example 3

In this example, no VoIP profile is selected in the firewall policy. However, the `default-voip-alg-mode` is set to `proxy-based`. The default `voip-profile` is implicitly applied.

To configure SIP ALG to implicitly use the default VoIP profile:

1. Configure the default VoIP ALG mode:

```
config system settings
    set default-voip-alg-mode proxy-based
end
```

2. Configure the firewall policy:

```
config firewall policy
    edit 0
        set name "VoIP-Proxy"
        set utm-status enable
        set voip-profile ""
    next
end
```

SIP session helper configurations

In some instances, SIP providers may recommend that customers disable SIP ALG on their edge firewall. This is how you can disable SIP ALG and enable the SIP session helper.

Example 1

In this example, the `default-voip-alg-mode` is set to `kernel-helper-based`, and a VoIP profile is not applied in a firewall policy. Session helper 13 is enabled by default.

To configure the SIP session helper:

1. Configure the default VoIP ALG mode:

```
config system settings
    set default-voip-alg-mode kernel-helper-based
end
```

2. Configure the firewall policy:

```
config firewall policy
    edit 0
        set name "VoIP-session-helper"
        set utm-status enable
        set voip-profile ""
    next
end
```

3. Configure the session helper:

```
config system session-helper
    edit 13
        set name sip
        set protocol 17
        set port 5060
    next
end
```

Example 2

In this example, the `default-voip-alg-mode` is set to either `proxy-based` or `kernel-helper-based`. A VoIP profile that has SIP disabled is applied to the firewall policy.

To configure the SIP session helper:

1. Configure the default VoIP ALG mode:

```
config system settings
    set default-voip-alg-mode {proxy-based | kernel-helper-based}
end
```

2. Configure the VoIP profile:

```
config voip profile
    edit "sip-disabled-profile"
        set feature-set voipd
        config sip
```

```
            set status disable
        end
    next
end
```

3. Configure the firewall policy:

```
config firewall policy
    edit 0
        set name "VoIP-session-helper"
        set utm-status enable
        set voip-profile "sip-disabled-profile"
    next
end
```

4. Configure the session helper:

```
config system session-helper
    edit 13
        set name sip
        set protocol 17
        set port 5060
    next
end
```

Example 3

In certain scenarios, especially when advised by your SIP provider, it may be necessary to disable the SIP ALG and SIP session helper to ensure proper SIP traffic handling and avoid call issues.

To disable the SIP ALG and SIP session helper from managing SIP traffic:

1. Identify the SIP session helper entry:

```
# show system session-helper
```

Look for an entry similar to the following:

```
edit 13
    set name sip
    set protocol 17
    set port 5060
next
```

2. Once identified, remove the SIP session helper:

```
config system session-helper
    delete 13
end
```

Repeat this step for other ports if SIP traffic uses different ports.

3. Change the default SIP mode to set how the FortiGate handles VoIP traffic when a policy that accepts the traffic does not include a VoIP profile.

```
config system settings
    set default-voip-alg-mode kernel-helper-based
end
```

4. Inspect the firewall policy that governs SIP traffic:

```
# show firewall policy 1
config firewall policy
    edit 1
        set name "voip to internet"
        set srcintf "port1"
        set dstintf "wan1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set voip-profile "default"
        set logtraffic all
        set nat enable
    next
end
```

If no VoIP profile is configured in the policy, skip to the next step.

5. Modify or remove the VoIP profile:
   - Disable SIP ALG in the VoIP profile:

```
config voip profile
    edit default
        config sip
            set status disable
            set rtp disable
        end
    next
end
```

   - Remove the VoIP profile from the firewall policy:

```
config firewall policy
    edit 1
        unset voip-profile
    next
end
```

6. To ensure changes take effect, clear any existing SIP sessions:

```
# diagnose sys session filter dport 5060
# diagnose sys session clear
```

```
# diagnose sys session filter sport 5060
# diagnose sys session clear
```

Clearing sessions on port 5060 will terminate all active SIP calls passing through the FortiGate.

7. To ensure your FortiGate device is no longer utilizing SIP Session Helper or the SIP ALG, check the active sessions:

```
# diagnose sys session list | grep sip
```

Check for the following flags:

Flag	Description
`ndr`	Flow SIP (IPS) handles the traffic.
`redir and local`	SIP ALG handles the traffic.
`helper=sip`	SIP kernel helper (session-helper) handles the traffic.
`helper=20 or helper=24`	SIP ALG handles the traffic.

If you see any flags other than `ndr`, it indicates that either the SIP session helper or the SIP ALG is handling the traffic.

Modifying the SIP port

Most SIP configurations use TCP or UDP port 5060 for SIP sessions and port 5061 for SIP SSL sessions. If your SIP network uses different ports for SIP sessions, the SIP port can be changed. You can also listen to two TCP and UDP ports .

To change the SIP port:

```
config system settings
    set sip-tcp-port 5064
    set sip-udp-port 5065
    set sip-ssl-port 5066
end
```

To listen to two TCP and UDP ports:

```
config system settings
    set sip-tcp-port 5060 5064
    set sip-udp-port 5061 5065
end
```

To modify the SIP ports for the default SIP session helper:

```
config system session-helper
    edit 13
        set name sip
        set protocol 17
        set port 5065
    next
end
```

To add a new session helper to listen on UDP and TCP 5064:

```
config system session-helper
    edit 0
        set name sip
        set port 5064
    next
end
```

SIP pinholes

When SIP ALG processes a SIP call, it usually opens pinholes for SIP signaling and RTP/RTCP packets. NAT usually takes place during the process at both the network and SIP application layers. SIP ALG ensures that, with NAT happening, corresponding SIP and RTP/RTCP pinholes are created during the process when it is necessary for call sessions to be established through FortiOS devices.

By default, SIP ALG manages pinholes automatically, but some special configurations can be used to restrict the pinholes if required.

SIP pinhole restriction

The `strict-register` attribute is enabled by default. When enabled, after a SIP endpoint registers to the SIP server through a firewall policy on the FortiGate, only the SIP messages sent from the same IP address as the SIP server are allowed to pass through the SIP pinhole that is created in the FortiGate to reach the SIP endpoints. If the attribute is disabled, SIP messages from any IP addresses can pass through the pinhole created after the registration.

SIP pinhole restriction is only supported by SIP ALG and in proxy mode.

To configure registrar connection ability:

```
config voip profile
    edit <name>
        config sip
            set strict-register {enable | disable}
        end
```

```
    next
end
```

RTP/RTCP pinhole restriction

The `nat-port-range` setting is used to specify a port range in the VoIP profile to restrict the NAT port range for Real-time Transport Protocol/Real-time Transport Control Protocol (RTP/RTCP) packets in a Session Initiation Protocol (SIP) call session that is handled by the SIP application layer gateway (ALG) in a FortiGate.

When NAT is enabled, or VIP is used in a firewall policy for SIP ALG to handle a SIP call session established through a FortiGate, the SIP ALG can perform NAT to translate the ports used for the RTP/RTCP packets when they are flowing through the device between the external and internal networks.

To edit the translated port range for RTP/RTCP packets:

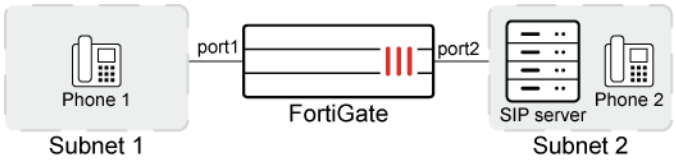
```
config voip profile
    edit <name>
        config sip
            set nat-port-range <start_port_number>-<end_port_number>
        end
    next
end
```

`nat-port-range <start_port_number>-<end_port_number>`	Enter the NAT port range (minimum port number = 5117, default = 5117-65535).

Example

In this example, Phone 1 is in Subnet 1, and the SIP server and Phone 2 are in Subnet 2. All SIP signaling messages and RTP/RTCP packets go through the SIP server. The RTP/RTCP ports on Phone 1 are configured as 17078/17079.

The FortiGate administrator wants to use NAT for the port 17078/17079 to 30000/30001. If Phone 1 and Phone 2 are registered to the SIP server, and they establish a call session between them through the FortiGate and the SIP server, then the RTP/RTCP ports 17078/17079 of Phone 1 will be translated to ports 30000/30001. All RTP/RTCP packets going out of port2 have source ports of 30000/30001, and all RTP/RTCP packets going into port2 also have destination ports of 30000/30001.



To configure the custom port range:

1. Edit the VoIP profile:

```
config voip profile
    edit "natPortRange"
        config sip
            set nat-port-range 30000-30001
        end
    next
end
```

> It is best practice to configure the starting port as an even number and the ending port as an odd number.

2. Configure the firewall policy:

```
config firewall policy
    edit 1
        set srcintf port1
        set dstintf port2
        set srcaddr all
        set dstaddr all
        set service SIP
        set action accept
        set schedule always
        set voip-profile natPortRange
        set nat enable
    next
end
```

SIP over TLS

Some SIP phones and servers can communicate using TLS to encrypt the SIP signaling traffic. To allow SIP over TLS calls to pass through the FortiGate, the encrypted signaling traffic must be unencrypted and inspected. The FortiGate SIP ALG intercepts, unencrypts, and inspects the SIP packets, which are then re-encrypted and forwarded to their destination.

The SIP ALG only supports full mode TLS. This means that the SIP traffic between SIP phones and the FortiGate, and between the FortiGate and the SIP server, is always encrypted. The highest TLS version supported by SIP ALG is TLS 1.3.

To enable SIP over TLS support, the SSL mode in the VoIP profile must be set to `full`. The SSL server and client certificates can be provisioned so that the FortiGate can use them to establish connections to SIP phones and servers, respectively.

> This configuration is only supported in proxy mode.

To configure SIP over TLS:

1. Configure a VoIP profile with SSL enabled:

```
config voip profile
    edit "tls"
        config sip
            set ssl-mode full
            set ssl-client-certificate "ssl_client_cert"
            set ssl-server-certificate "ssl_server_cert"
        end
    next
end
```

   The `ssl_server_cert`, `ssl_client_cert`, and key files can be generated using a certification tool, such as OpenSSL, and imported to the local certificate store of the FortiGate from *System* > *Certificates* in the GUI. Existing local certificates in the certificate store can also be used. As always for TLS connections, the certificates used must be verified and trusted at the other end of the connection when required.

   For example, the CA certificate of the SIP server's certificate should be imported to the FortiGate as an external CA certification, so that the FortiGate can use it to verify the SIP server's certificate when setting up the TLS connection. The CA certificate configured as the `ssl_server_cert` should be installed as the trusted certificate on the SIP phones. The deployment of the certificates across the network depends on the SIP client and server devices that are used in the system.

2. Apply the profile to the firewall policy:

```
config firewall policy
    edit 1
        set srcintf "port1"
        set dstintf "port2"
        set srcaddr "all"
        set dstaddr "vip_sip_server"
        set action accept
        set schedule "always"
        set service "SIP"
        set utm-status enable
        set voip-profile "tls"
    next
end
```

Voice VLAN auto-assignment

You can leverage LLDP-MED to assign voice traffic to the desired voice VLAN. After detection and setup, the IP phone on the network is segmented to its own VLAN for policy, prioritization, and reporting. The LLDP reception capabilities in FortiOS include LLDP-MED assignment for voice, voice signaling, guest, guest voice signaling, softphone, video conferencing, streaming video, and video signaling.

You can configureVLAN auto-assignment using the following steps:

1. Set up the VLAN for the voice device
2. Set up the DHCP server for the voice VLAN

3. Sett up the LLDP network policy
4. Enable LLDP on the physical interface that the VLAN belongs to
5. Apply the LLDP network policy on the physical interface
6. Confirm that the VLAN was assigned

To set up the VLAN for the voice device:

```
config system interface
    edit "vlan_100"
        set vdom "root"
        set ip 192.168.1.99 255.255.255.0
        set alias "voice_vlan"
        set device-identification enable
        set role lan
        set snmp-index 25
        set interface "port10"
        set vlanid 100
    next
end
```

To set up the DHCP server for the voice VLAN:

```
config system dhcp server
    edit 1
        set dns-service default
        set default-gateway 192.168.1.99
        set netmask 255.255.255.0
        set interface "vlan_100"
        config ip-range
            edit 1
                set start-ip 192.168.1.110
                set end-ip 192.168.1.210
            next
        end
    next
end
```

To set up the LLDP network policy:

```
config system lldp network-policy
    edit "1"
        config voice
            set status enable
            set tag dot1q
            set vlan 100
        end
    next
end
```

To enable LLDP on the physical interface that the VLAN belongs to:

```
config system interface
    edit "port10"
        set vdom "root"
        set type physical
        set lldp-reception enable
        set lldp-transmission enable
        set snmp-index 14
    next
end
```

To apply the LLDP network policy on the physical interface:

```
config system interface
    edit "port10"
        set lldp-network-policy "1"
    next
end
```

To confirm that the VLAN was assigned as expected:

1. Connect an IP phone to the network.
2. Check the IP address on the phone.
   The IP address should belong to the voice VLAN.
3. Sniff on the FortiGate incoming interface to see if traffic from the IP phone has the desired VLAN tag.
   In this example, the voice traffic from the IP phone should be in VLAN 100.

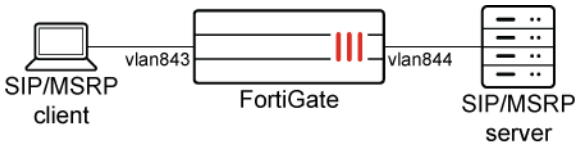
Scanning MSRP traffic

An MSRP (Message Session Relay Protocol) decoder in the IPS engine scans for IPS signatures against the application data. Malicious payload in the text message can be blocked. A VoIP profile using flow inspection mode must be configured in the firewall policy. An IPS profile must be configured in the firewall policy to inspect the payload.

```
config voip profile
    edit <name>
        set feature-set flow
        config msrp
            set status {enable | disable}
            set log-violations {enable | disable}
            set max-msg-size <integer>
            set max-msg-size-action {pass | block | reset | monitor}
        end
    next
end
```

`status {enable | disable}`	Enable/disable MSRP.
`log-violations {enable | disable}`	Enable/disable logging of MSRP violations.
`max-msg-size <integer>`	Maximum allowable MSRP message size, in bytes (0 - 65535, default = 0).
`max-msg-size-action {pass | block | reset | monitor}`	Action for violating maximum MSRP message size: • pass: pass or allow matching traffic (default) • block: block or drop matching traffic • reset: reset sessions for matching traffic • monitor: pass and log matching traffic

Examples



In this first example, MSRP messages larger than 10 bytes will be blocked. The client sends an oversized MSRP message to the server. Message Automation & Protocol Simulation (MAPS™) is used, and a client-server model was configured to use the software to send MSRP traffic from vlan843 (client) to vlan844 (server) with plain text placed in the message field. The software uses the content of the MsrpInputMessage.txt file located in the default folder, where anything in that file will be sent by MSRP. The following text is used:

> GL's Message Automation & Protocol Simulation (MAPS™) is a protocol simulation and conformance test tool that supports a variety of protocols such as SIP, MEGACO, MGCP, SS7, ISDN, GSM, MAP, CAS, LTE, UMTS, SS7 SIGTRAN, ISDN SIGTRAN, SIP I, GSM AoIP, Diameter and others. This message automation tool covers solutions for both protocol simulation and protocol analysis. The application includes various test plans and test cases to support the testing of real-time entities. Along with automation capability, the application gives users the unlimited ability to edit messages and control scenarios (message sequences).

To configure MSRP traffic scanning:

1. Configure the VoIP profile:

```
config voip profile
    edit msrp_test
        set feature-set flow
        config msrp
            set status enable
            set log-violations enable
            set max-msg-size 10
            set max-msg-size-action block
        end
    next
end
```

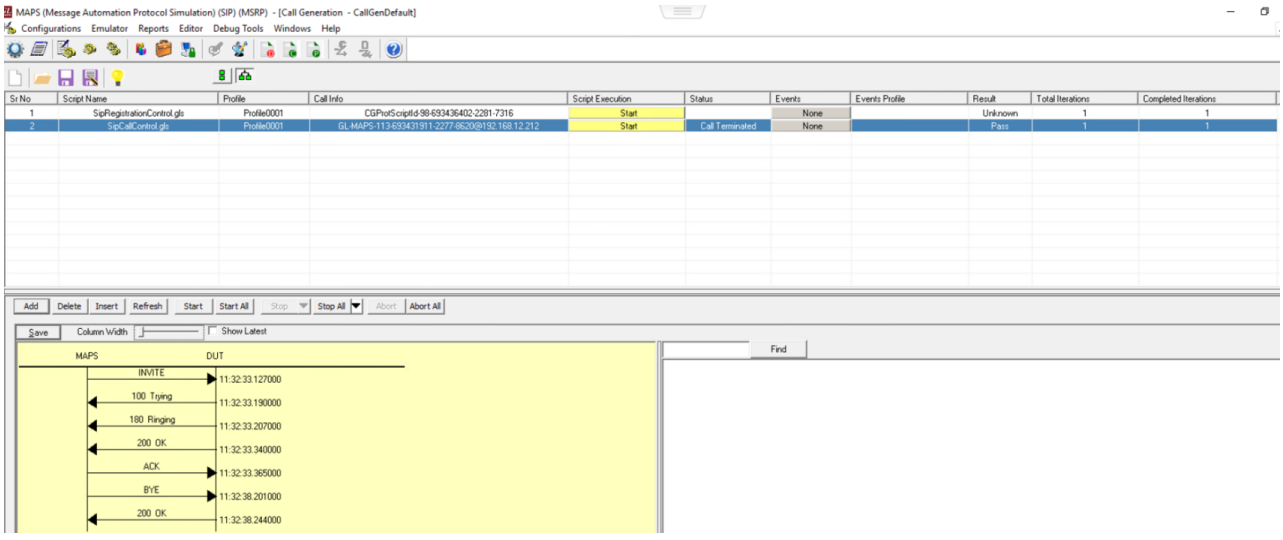
2. Configure the firewall policy:

```
config firewall policy
    edit 1
        set name "vdom3"
        set srcintf "vlan843"
        set dstintf "vlan844"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "certificate-inspection"
        set voip-profile "msrp_test"
        set logtraffic all
    next
end
```

3. Verify the log:

```
# execute log filter category 4
# execute log display
1 logs found.
1 logs returned.

1: date=2021-06-10 time=17:21:19 eventtime=1623370879840284165 tz="-0700" logid="0419016384"
type="utm" subtype="ips" eventtype="signature" level="alert" vd="vdom3" severity="info"
srcip=192.168.12.212 srccountry="Reserved" dstip=192.168.12.213 srcintf="vlan843"
srcintfrole="lan" dstintf="vlan844" dstintfrole="lan" sessionid=27700 action="dropped" proto=6
service="MSRP" policyid=1 attack="MSRP.Max.Message.Size.Exceeded" srcport=20036 dstport=20036
direction="outgoing" attackid=1000000 profile="g-default"
ref="http://www.fortinet.com/ids/VID1000000" incidentserialno=189792275 psrcport=0 pdstport=0
msg="msrp_decoder: MSRP.Max.Message.Size.Exceeded, msg_size=270 exceeds config maximum=10"
```

4. In MAPS, verify that the call was terminated:



In this second example, malicious files will be blocked. The client sends an EICAR test sample to the server in an MSRP message. Message Automation & Protocol Simulation (MAPS™) is used, and a client-server model was configured to use the software to send MSRP traffic from vlan843 (client) to vlan844 (server) with a plain text EICAR file containing a virus in the message field. The following text is used:

X5O!P%@AP[4\PZX54(P^)7CC)7}$EICAR-STANDARD-ANTIVIRUS-TEST-FILE!$H+H*

To configure MSRP traffic scanning:

1. Configure the VoIP profile:

```
config voip profile
    edit msrp_test
        set feature-set flow
        config msrp
            set status enable
            set log-violations enable
            set max-msg-size 0
            set max-msg-size-action pass
        end
    next
end
```

2. Configure the IPS profile:

```
config ips sensor
    edit "msrp"
        set extended-log enable
        config entries
            edit 1
                set rule 7470 29844
                set status enable
                set action block
            next
        end
    next
end
```

3. Configure the firewall policy:

```
config firewall policy
    edit 1
        set name "vdom3"
        set srcintf "vlan843"
        set dstintf "vlan844"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "certificate-inspection"
        set ips-sensor "msrp"
```

```
        set voip-profile "msrp_test"
        set logtraffic all
    next
end
```

4. Verify the log:

```
# execute log filter category 4
# execute log display
1 logs found.
1 logs returned.

1: date=2021-09-16 time=11:29:48 eventtime=1631816988947762597 tz="-0700" logid="0419016384"
type="utm" subtype="ips" eventtype="signature" level="alert" vd="vdom3" severity="info"
srcip=192.168.12.212 srccountry="Reserved" dstip=192.168.12.213 srcintf="vlan843"
srcintfrole="lan" dstintf="vlan844" dstintfrole="lan" sessionid=41344 action="dropped" proto=6
service="MSRP" policyid=1 attack="Eicar.Virus.Test.File" srcport=20069 dstport=20069
direction="outgoing" attackid=29844 profile="msrp" ref="http://www.fortinet.com/ids/VID29844"
incidentserialno=123731970 psrcport=0 pdstport=0 msg="file_transfer: Eicar.Virus.Test.File,"
```

ICAP

Internet Content Adaptation Protocol (ICAP) is an application layer protocol that is used to offload tasks from the firewall to separate, specialized servers. For more information see RFC 3507.

ICAP profiles can only be applied to policies that use proxy-based inspection. If you enable ICAP in a policy, HTTP and HTTPS (if HTTPS inspection is supported) traffic that is intercepted by the policy is transferred to the ICAP server specified by the selected ICAP profile. Responses from the ICAP server are returned to the FortiGate, and then forwarded to their destination.

By default, *ICAP* is not visible in the GUI. See Feature visibility on page 3952 for instructions on making it visible.

ICAP filter profiles cannot be used in NGFW policy-based mode. See NGFW policy on page 1868 for more information.

This feature is not supported on FortiGate models with 2 GB RAM or less. See Proxy-related features not supported on FortiGate 2 GB RAM models on page 126 for more information.

To configure ICAP:

1. Set up your ICAP server.
2. On the FortiGate, add an ICAP server.
3. Create an ICAP profile.
4. Use the ICAP profile in a firewall policy that covers the traffic that needs to be offloaded to the ICAP server.

The following topics provide information about ICAP:

- ICAP configuration example on page 2583
- ICAP response filtering on page 2586
- Secure ICAP clients on page 2588
- ICAP scanning with SCP and FTP on page 2589
- Domain name in XFF with ICAP on page 2592
- Protocol transparency for ICAP on page 2596
- Selective forwarding to ICAP server on page 2597

TCP connection pool for connections to ICAP server

A TCP connection pool can maintain local-out TCP connections to the external ICAP server due to a backend update in FortiOS. TCP connections will not be terminated once data has been exchanged with the ICAP server, but instead are reused in the next ICAP session to maximize efficiency.

For example, consider a scenario where an ICAP profile is used as a UTM profile in an explicit web proxy policy, and a client visits web servers through this proxy policy.

Once the WAD is initialized, when a HTTP request is sent from the client to the server through the FortiGate with an ICAP profile applied to the matched proxy policy, a TCP connection is established between the FortiGate and the ICAP server to exchange data.

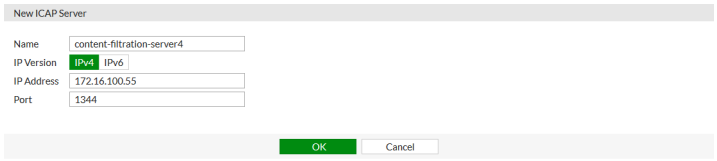
When an ICAP session is finished, the TCP connection is kept in the WAD connection pool. When another ICAP session needs to be established, the WAD will check if there are any idle connections available in the connection pool. If an idle connection is available, then it will be reused; otherwise, a new TCP connection is established for the ICAP session. This process can be checked in the WAD debug log.

ICAP configuration example

In this example, the ICAP server performs proprietary content filtering on HTTP and HTTPS requests. If the content filter is unable to process a request, then the request is blocked. Streaming media is not considered by the filter, so it is allowed through and is not processed.

To configure the ICAP setup in the GUI:

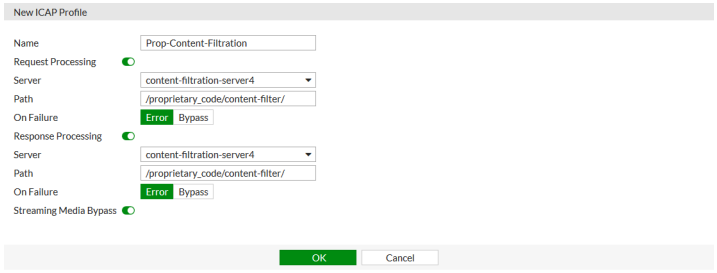
1. Add the ICAP server:
   a. Go to *Security Profiles* > *ICAP Servers* and click *Create New*.
   b. In the *Name* field, enter a name for the ICAP server, such as *content-filtration-server4*.
   c. Select the *IP Version*.
   d. In the *IP Address* field, enter the IP address of the ICAP server.
   e. In the *Port* field, enter a new port number if required. The default value is *1344*.



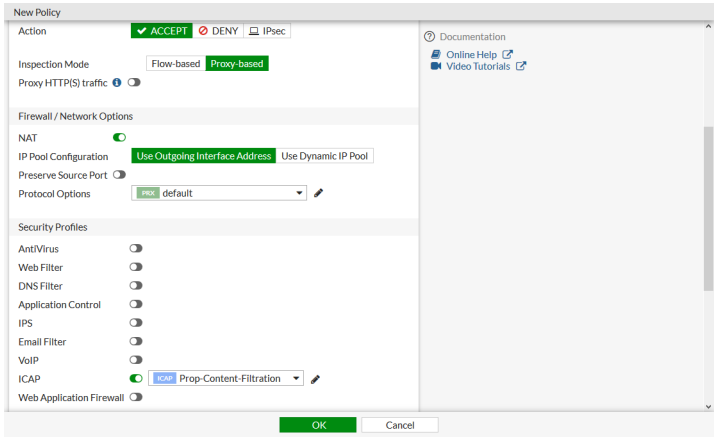
   f. Click *OK*.

> The maximum number of concurrent connections to ICAP server can be configured in the CLI (`set max-connections`). The default setting is 100 connections.

2. Create the ICAP profile:
   a. Go to *Security Profiles* > *ICAP* and click *Create New*.
   b. In the *Name* field, enter a name for the ICAP profile, such as *Prop-Content-Filtration*.
   c. Enable *Request Processing* and set the following:
      - *Server*: select the ICAP server (*content-filtration-server4*).
      - *Path*: enter the path to the processing component on the server, such as */proprietary_code/content-filter/*.
      - *On Failure*: select *Error* to block the request. If the message cannot be processed, it will be blocked.
   d. Enable *Response Processing* and set the following:
      - *Server*: select the ICAP server (*content-filtration-server4*).
      - *Path*: enter the path to the processing component on the server, such as */proprietary_code/content-filter/*.
      - *On Failure*: select *Error* to block the request. If the message cannot be processed, it will be blocked.
   e. Enable *Streaming Media Bypass* to not offload streaming media to the ICAP server.



   f. Click *OK*.
3. Add the ICAP profile to a policy:
   a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
   b. Set *Inspection Mode* to *Proxy-based*.
   c. Under *Security Profiles*, enable *ICAP* and select the ICAP server.



d. Configure the other settings as needed.

e. Click *OK*.

To configure the ICAP setup in the CLI:

1. Add the ICAP server:

```
config icap server
    edit "content-filtration-server4"
        set ip-version 4
        set ip-address 172.16.100.55
        set port 1344
        set max-connections 200
    next
end
```

2. Create the ICAP profile:

```
config icap profile
    edit "Prop-Content-Filtration"
        set request enable
        set response enable
        set streaming-content-bypass enable
        set request-server "content-filtration-server4"
        set response-server "content-filtration-server4"
        set request-failure error
        set response-failure error
        set request-path "/proprietary_code/content-filter/"
        set response-path "/proprietary_code/content-filter/"
        set methods delete get head options post put trace other
    next
end
```

3. Add the ICAP profile to a policy:

```
config firewall policy
    edit 5
        set name "icap_filter3"
        set srcintf "virtual-wan-link"
        set dstintf "virtual-wan-link"
        set srcaddr "FABRIC_DEVICE"
        set dstaddr "FABRIC_DEVICE"
        set dstaddr-negate enable
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "certificate-inspection"
        set icap-profile "Prop-Content-Filtration"
        set logtraffic disable
        set fsso disable
```

```
        set nat enable
    next
end
```

ICAP response filtering

ICAP HTTP responses can be forwarded or bypassed based on the HTTP header value and status code.

When configuring the ICAP profile, if `response` is enabled, the `respmod-default-action` option can be configured:

- If `respmod-default-action` is set to `forward`, FortiGate will treat every HTTP response and send ICAP requests to the ICAP server.
- If `respmod-default-action` is set to `bypass`, FortiGate will only send ICAP requests if the HTTP response matches the defined rules, and the rule's action is set to forward.

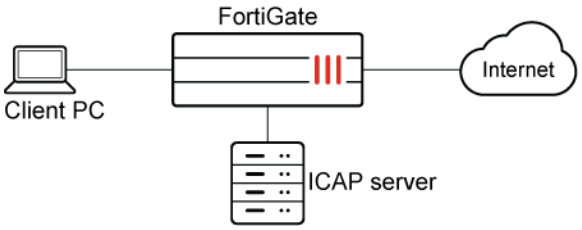
When configuring a response rule:

- The `http-resp-status-code` option is configured to specific HTTP response codes. If the HTTP response has any one of the configured values, then the rule takes effect.
- Multiple header value matching groups can be configured. If the header value matches one of the groups, then the rule takes effect.
- If both status codes and header values are specified in a rule, the response must match at least one of each.

The UTM ICAP log category is used for logging actions when FortiGate encounters errors with the ICAP server, such as no service, unreachable, error response code, or timeout. If an error occurs, a traffic log and an associated UTM ICAP log will be created.

Example

The FortiGate acts as a gateway for the client PC and connects to a reachable ICAP server. The ICAP server can be in NAT, transparent, or proxy mode.



In this example, client request HTTP responses will be forwarded to the ICAP server from all hosts if they have an HTTP status code of 200, 301, or 302, and have `content-type: image/jpeg` in the their header.

To configure an ICAP profile with HTTP response rules:

```
config icap profile
    edit "icap_profile2"
        set request disable
```

```
        set response enable
        set streaming-content-bypass disable
        set preview disable
        set response-server "icap_server1"
        set response-failure error
        set response-path ''
        set methods delete get head options post put trace other
        set response-req-hdr disable
        set respmod-default-action bypass
        config respmod-forward-rules
            edit "rule2"
                set host "all"
                set action forward
                set http-resp-status-code 200 301 302
                config header-group
                    edit 2
                        set header-name "content-type"
                        set header "image/jpeg"
                    next
                end
            next
        end
    next
end
```

To view the logs if an error occurs:

1. View the traffic log:

```
# execute log filter category 0
# execute log display
1 logs found.
1 logs returned.

 1: date=2019-10-25 time=17:43:47 logid="0000000013" type="traffic" subtype="forward"
level="notice" vd="vdom1" eventtime=1572050627037314464 tz="-0700" srcip=10.1.100.145
srcport=47968 srcintf="port1" srcintfrole="undefined" dstip=172.16.200.46 dstport=80
dstintf="port2" dstintfrole="undefined" poluuid="a4d5324e-f6c3-51e9-ce2d-f360994fb547"
sessionid=43549 proto=6 action="close" policyid=1 policytype="policy" service="HTTP"
dstcountry="Reserved" srccountry="Reserved" trandisp="snat" transip=172.16.200.1
transport=47968 duration=1 sentbyte=485 rcvdbyte=398 sentpkt=6 rcvdpkt=5 appcat="unscanned"
wanin=478 wanout=165 lanin=165 lanout=165 utmaction="block" counticap=1 crscore=5
craction=262144 crlevel="low" utmref=65532-0
```

2. View the UTM ICAP log:

```
# execute log filter category 20
# execute log display
1 logs found.
1 logs returned.

 1: date=2019-10-25 time=17:43:46 logid="2000060000" type="utm" subtype="icap"
```

```
eventtype="icap" level="warning" vd="vdom1" eventtime=1572050626010097145 tz="-0700"
msg="Request blocked due to ICAP server error" service="HTTP" srcip=10.1.100.145
dstip=172.16.200.46 srcport=47968 dstport=80 srcintf="port1" srcintfrole="undefined"
dstintf="port2" dstintfrole="undefined" policyid=1 sessionid=43549 proto=6 action="blocked"
profile="icap_profile1" url="/icap_test/"
```

The logs show that the ICAP services stopped before the access. When the client tried to access HTTP and ICAP took effect, the FortiGate sent the ICAP request to the ICAP server and received an error. The client sees a *502 Bad Gateway* message, and FortiGate writes the two logs. In the GUI, the logged traffic is displayed as *Result: Deny: UTM Blocked*.

Secure ICAP clients

A secure SSL connection from the FortiGate to the ICAP server can be configured as follows:

```
config icap server
    edit <name>
        set secure {enable | disable}
        set ssl-cert <certificate>
    next
end
```

To configure a secure ICAP client:

1. Configure the ICAP server:

```
config icap server
    edit "icap_server1"
        set ip-version 4
        set ip-address 192.168.10.2
        set port 11344
        set max-connections 100
        set secure enable
        set ssl-cert "ACCVRAIZ1"
    next
end
```

> Port 11344 is the standard port for secure ICAP. This must be configured manually if the secure connection is enabled.

2. Configure the ICAP profile:

```
config icap profile
    edit "icap_profile1"
        set request enable
        set response enable
        set streaming-content-bypass enable
        set request-server "icap_server1"
        set response-server "icap_server1"
```

```
    next
end
```

3. Configure the firewall policy:

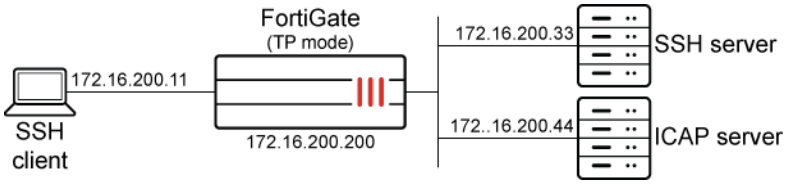
```
config firewall policy
    edit 1
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "protocols"
        set icap-profile "icap_profile1"
    next
end
```

ICAP scanning with SCP and FTP

A FortiGate can forward files transferred by SCP and FTP to an ICAP server for further scanning. Previously, only HTTP and HTTPS were supported for ICAP forwarding.

Example

The FortiGate used in this example is operating in transparent mode. The SSH client, 172.16.200.11, sends a file named `today` to the SSH server at 172.16.200.33 using SCP. Since SCP transfers are encrypted inside an SSH tunnel, for the FortiGate to scan the traffic, deep inspection must be enabled in the SSL SSH profile.



To configure ICAP scanning with SCP:

1. Configure the ICAP server settings:

```
config icap server
    edit "icap_server1"
        set ip-address 172.16.200.44
    next
end
```

2. Configure the ICAP profile for SSH:

```
config icap profile
    edit "icap_profile1"
        set file-transfer ssh
        set file-transfer-server "icap_server1"
        set file-transfer-path "ssh_test"
```

```
    next
end
```

If the file transfer is over FTP, configure the profile as follows:

```
config icap profile
    edit "icap_profile1"
        set file-transfer ftp
        set streaming-content-bypass enable
        set file-transfer-server "icap_server1"
        set file-transfer-path "ftp_test"
    next
end
```

3. Configure the SSL SSH profile:

```
config firewall ssl-ssh-profile
    edit "protocols"
        config ssh
            set ports 22
            set status deep-inspection
        end
    next
end
```

4. Configure the firewall policy:

```
config firewall policy
    edit 1
        set name "ICAP"
        set srcintf "lan"
        set dstintf "mgmt"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set profile-protocol-options "protocol"
        set ssl-ssh-profile "protocols"
        set icap-profile "icap_profile1"
    next
end
```

To test the configuration:

1. On a Linux client, copy a filed named `today` to the SSH server using SCP:

```
scp today fosqa@172.16.200.33:/home/fosqa/ssh_depot/
```

2. Capture a sniffer trace between the FortiGate and ICAP server, then verify the output from the ICAP protocol session.
   a. The client request and the file to be inspected:

```
Icap_client REQMOD:
172.016.200.200.13185-172.016.200.044.01344: REQMOD icap://172.16.200.44:1344/ssh_test
ICAP/1.0
Host: 172.16.200.44:1344
X-Client-IP: 172.16.200.11
X-Server-IP: 172.16.200.33
X-Authenticated-User: TG9jYWw6Ly9hbm9ueW1vdXM=
X-Authenticated-Groups: TG9jYWw6Ly9sb2NhbGhvc3Qvbm8gYXV0aGVudGljYXRpb24=
User-Agent: FortiOS v7.2.0
Encapsulated: req-hdr=0, req-body=116

PUT /scp/today HTTP/1.1
Host: 172.16.200.11
Content-Type: application/octet-stream
Transfer-Encoding: chunked

1d
Tue Sep 20 04:01:50 UTC 2022
```

   Where:
   - `X-Client-IP` = the client sending the file
   - `X-Server-IP` = the server receiving the file
   - `Tue Sep 20 04:01:50 UTC 2022` = the content of the file, which is in clear text after the FortiGate performs deep inspection

   b. The ICAP server response that the file is cleared and allowed to pass without modifications:

```
Icap-server reply:
172.016.200.044.01344-172.016.200.200.13185: ICAP/1.0 200 OK
ISTag: "GreasySpoon-1.0.7-b03"
Host: 0.0.0.0:1344
Encapsulated: req-hdr=0, req-body=136
Connection: keep-alive

PUT /scp/today HTTP/1.1
Host: 172.16.200.11
Content-Type: application/octet-stream
Transfer-Encoding: chunked
Content-Length: 29

1d
Tue Sep 20 04:01:50 UTC 2022
```

3. On a Linux client, copy the file from the server locally using SCP:

```
scp fosqa@172.16.200.33:/home/fosqa/ssh_depot/today2/
```

4. Similar outputs are observed. The ICAP client request indicates that the file is copied from the SSH server:

```
PUT /scp/today2 HTTP/1.1
Host: 172.16.200.33
```

Domain name in XFF with ICAP

The FortiGate can forward additional domain-related information to the ICAP server. Once domain information is gathered from an external authentication server (such as LDAP or an FSSO collector agent), FortiOS incorporates this domain information in `WinNT://DOMAIN/Username` format and forwards it to the ICAP server.

Basic ICAP configuration

The ICAP server and profile are configured on the FortiGate. The ICAP profile's header settings uses the `WinNT://$domain/$user` variable for the user information provided by the remote authentication server.

To configure the ICAP settings:

1. Configure the ICAP server:

```
config icap server
    edit "content-filtration-server4"
        set ip-address 10.1.100.41
        set max-connections 200
    next
end
```

2. Configure the ICAP profile:

```
config icap profile
    edit "Prop-Content-Filtration"
        set request enable
        set response enable
        set streaming-content-bypass enable
        set request-server "content-filtration-server4"
        set response-server "content-filtration-server4"
        set request-path "/proprietary_code/content-filter/"
        set response-path "/proprietary_code/content-filter/"
        set methods delete get head options post put trace other
        config icap-headers
            edit 1
                set name "X-Authenticated-User"
                set content "WinNT://$domain/$user"
            next
        end
    next
end
```

3. Configure the firewall policy:

```
config firewall policy
    edit 4
        set name "icap_filter3"
        set srcintf "port10"
        set dstintf "port9"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "deep-inspection"
        set icap-profile "Prop-Content-Filtration"
        set logtraffic all
        set nat enable
        set groups "ldap group" "AD-group"
    next
end
```

LDAP example

In this example, an AD LDAP server and remote user group are configured. When successful user authentication occurs, FortiOS retrieves all the user information (such as the domain name) from the UserPrincipalName attribute. A packet capture is used to compare the user and domain information before and after authentication in the ICAP REQMOD message.

To configure the LDAP authentication:

1. Configure the LDAP server:

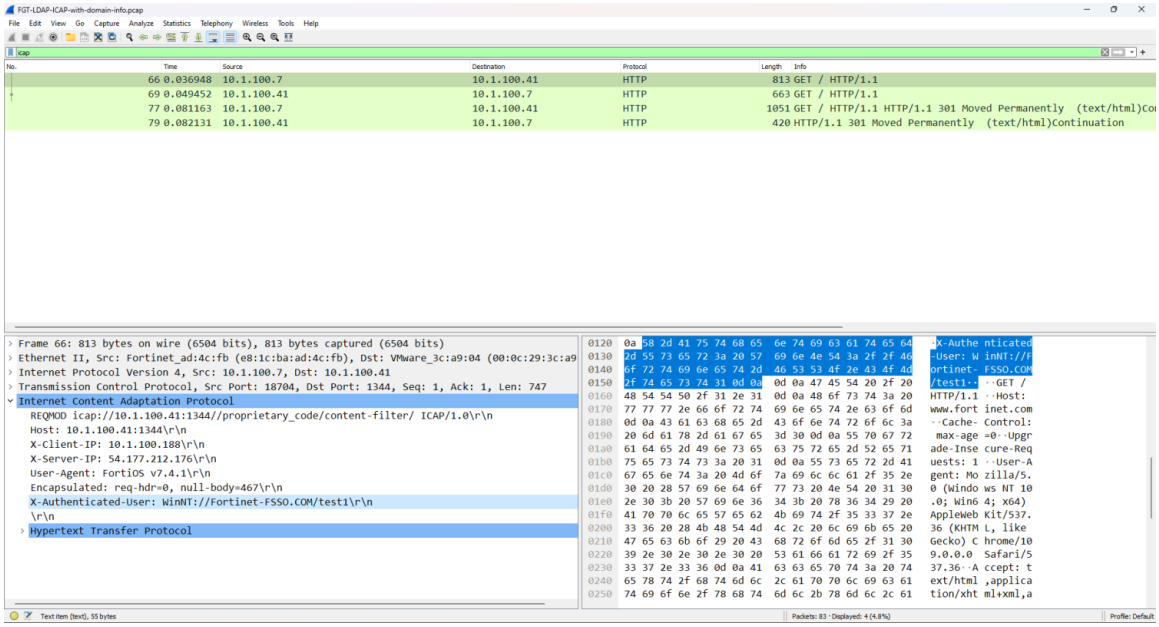
```
config user ldap
    edit "AD-ldap"
        set server "10.1.100.131"
        set cnid "cn"
        set dn "dc=fortinet-fsso,dc=com"
        set type regular
        set username "cn=Administrator,cn=users,dc=fortinet-fsso,dc=com"
        set password **********
    next
end
```

2. Configure the LDAP user group:

```
config user group
    edit "ldap group"
        set member "AD-ldap"
        config match
            edit 1
                set server-name "AD-ldap"
```

```
                set group-name "CN=group1,OU=Testing,DC=Fortinet-FSSO,DC=COM"
            next
            edit 2
                set server-name "AD-ldap"
                set group-name "CN=group2,OU=Testing,DC=Fortinet-FSSO,DC=COM"
            next
        end
    next
end
```

3. Start local traffic dump between the FortiGate and ICAP server before a user authenticates and save it in a PCAP file.
4. Verify the PCAP file. The Fortinet-fsso.com domain appears in the ICAP REQMOD message.



5. Optionally, run the following command to verify WAD debugs:

```
# diagnose wad debug enable category icap
```

FSSO example

In this example, a local FSSO agent and remote user group are configured. When successful user authentication occurs, FortiOS retrieves all the user information (such as the domain name). A packet capture is used to compare the user and domain information before and after authentication in the ICAP REQMOD message.

To configure the FSSO authentication:

1. Configure the FSSO agent:

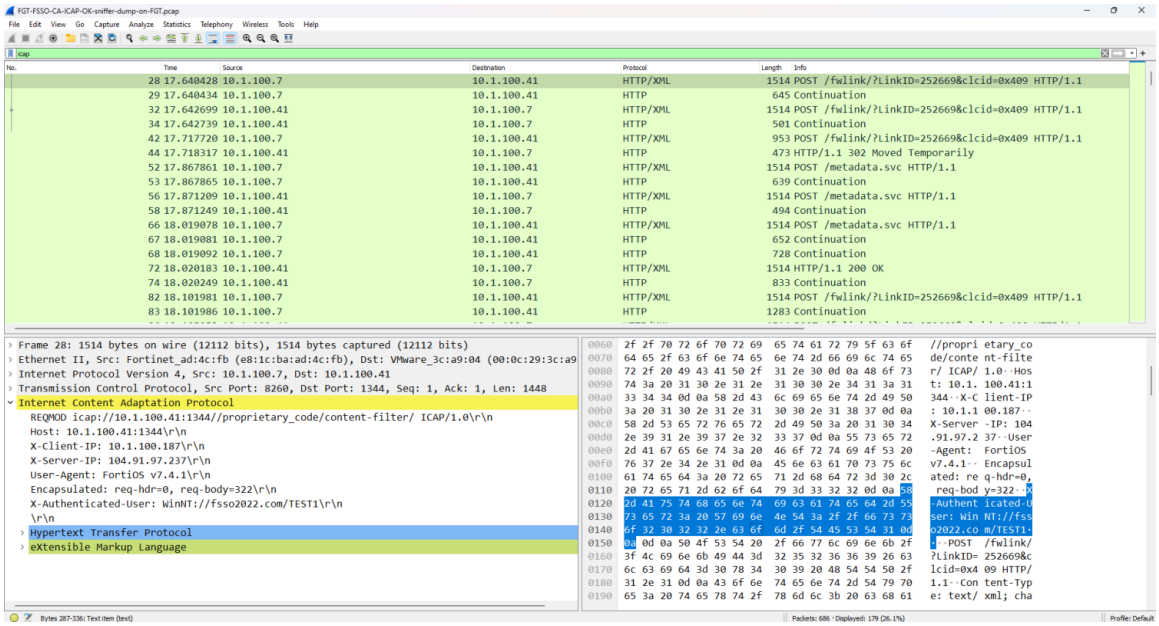
```
config user fsso
    edit "AD-fsso"
        set server "10.1.100.199"
```

```
        set password **********
    next
end
```

2. Configure the FSSO user group:

```
config user group
    edit "AD-group"
        set group-type fsso-service
        set member "FORTINET-FSSO/GROUP1" "FORTINET-FSSO/GROUP2"
    next
end
```

3. Start local traffic dump between the FortiGate and ICAP server before a user authenticates and save it in a PCAP file.
4. Verify the PCAP file. The fsso2022.com domain appears in the ICAP REQMOD message.



5. Optionally, verify the FSSO log file and search for the `get_dns_domain` lines:

```
...
06/20/2023 14:58:58 [ 1484] FortiGate connection accepted, auth OK.
06/20/2023 14:58:58 [ 1484] FortiGate:FG4H1E5819900343-root connected on socket (2004).
06/20/2023 14:58:58 [ 1484] send AUTH, len:26
06/20/2023 14:58:58 [ 1484] ready to read from socket
06/20/2023 14:58:58 [ 1484] Bytes received from FortiGate: 26
06/20/2023 14:58:58 [ 1484] process AD_INFO
06/20/2023 14:58:58 [ 1484] group filter received from FortiGate: len:26
06/20/2023 14:58:58 [ 1484] packet seq:2
06/20/2023 14:58:58 [ 1484] ad info flag:1
06/20/2023 14:58:58 [ 1484] FGT sends empty group list
06/20/2023 14:58:58 [ 1484] ready to read from socket
06/20/2023 14:58:58 [ 1484] Bytes received from FortiGate: 36
06/20/2023 14:58:58 [ 1484] packet seq:3
```

```
06/20/2023 14:58:58 [ 1484] option:00000001 ref point:00000000
06/20/2023 14:58:58 [ 1484] toFGT set to:1
06/20/2023 14:58:58 [ 1484] get_dns_domain_name:177 enable_dns_domain_name:1, netbios_domain_
name:FSSO2022
06/20/2023 14:58:58 [ 1484] get_dns_domain_name:185 dns_domain_name:FSSO2022.com
06/20/2023 14:58:58 [ 1484] send LOGON_INFO, len:187
06/20/2023 14:58:58 [ 1484] send_to_FGT() called:sock:2004 sendbuf:198f4498 sendlen:187
```

Protocol transparency for ICAP

FortiGate incorporates the original client protocol type (such as, HTTP, HTTPS, FTP, FTPS, SFTP, and SCP) into the ICAP header through the `X-Forwarded-Proto` field, enhancing integration with FortiData as the ICAP server. This enables FortiData to have full information about the client protocol used, allowing it to enforce security according to corporate security policies.

For example, the client-side ICAP packet captures includes the `X-Forwarded-Proto` field, as follows:

```
172.016.200.200.02467-172.016.200.044.01344: RESPMOD icap://172.16.200.44:1344/ ICAP/1.0
Host: 172.16.200.44:1344
X-Client-IP: 10.1.100.11
X-Server-IP: 172.16.200.33
X-Forwarded-Proto: HTTPS
User-Agent: FortiOS v7.6.1
X-Scan-Progress-Interval: 10
Encapsulated: req-hdr=0, res-hdr=103, res-body=354

GET /icap_test/redirection.html HTTP/1.1
Host: 172.16.200.33
User-Agent: curl/7.68.0
Accept: */*

HTTP/1.1 200 OK
Date: Tue, 08 Oct 2024 22:24:36 GMT
Server: Apache/2.4.41 (Ubuntu)
Last-Modified: Tue, 20 Dec 2022 00:41:05 GMT
ETag: "al-5f037b0ec71e4"
Accept-Ranges: bytes
Content-Length: 161
Vary: Accept-Encoding
Content-Type: text/html

al
<html>
.<body>
.</script>
..<p>It is the target url of redirection on the 172.16.200.33!</p>
..<p>ICAP server has redirected the request</p>
.</body>
</html>

0
```

Selective forwarding to ICAP server

The ICAP profile can allow the selective forwarding of only image files, such as JPEG, JPG, and PNG, to an ICAP (Internet Content Adaptation Protocol) server for OCR (optical character recognition) scanning. When enabled, FortiGate forwards only image files that are relevant for OCR scanning to the ICAP server. This selective forwarding applies only to image files in HTTP responses; it does not apply to image files in HTTP requests. By reducing processing time and optimizing resource usage, this feature enhances overall system efficiency.

```
config icap profile
    edit <name>
        set ocr-only {enable | disable}
    next
end
```

Command	Description
`ocr-only {enable | disable}`	Enable/disable only passing OCR scan requests of images files to ICAP server (default = disabled). When enabled, also enable `response` to allow FortiGate to forward images to the ICAP server.

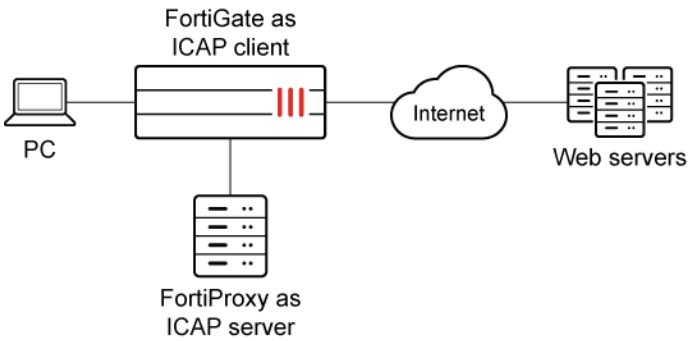
You cannot enable the `ocr-only` and `streaming-content-bypass` options in an ICAP profile at the same time. When `ocr-only` is enabled, the `streaming-content-bypass` option is removed from the CLI.

The ocr-only feature applies only to HTTP. FTP and SCP are not supported. In addition, this feature applies only to HTTP downloads. HTTP uploads are not supported.

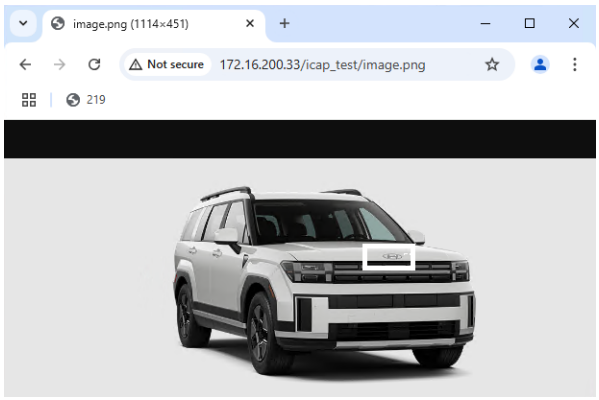
Example

In this example, FortiGate acts as the ICAP client, and FortiProxy acts as the ICAP server. An ICAP profile is configured on FortiGate with *ocr-only* enabled. An ICAP server is configured on FortiProxy with the `icap-service` configured to use an image-analyzer ICAP profile.

When a client HTTP response includes an image that is of interest to OCR, FortiGate forwards only the image file to the ICAP server for OCR scanning, and the scan results determine whether the image is passed or blocked.



When OCR scanning passes the image in the HTTP response, the image is displayed to the client, for example:



When OCR scanning blocks the image in the HTTP response, an alert message is displayed instead of the image:

Only configurations relevant to selective forwarding are described.

To enable selective forwarding on the ICAP client:

1. On FortiGate, enable `ocr-only` in the ICAP profile:
   In this example, `ocr-only` and `response` are enabled. Responses are enabled to allow FortiGate to forward images files to the ICAP server.

```
config icap profile
    edit "ocr"
        set request enable
        set response enable
        set ocr-only enable
        set request-server "icap_server1"
        set response-server "icap_server1"
    next
end
```

To enable image scanning on the ICAP server:

1. On FortiProxy acting as an ICAP server, create an image-analyzer ICAP profile:
   In this example, an image-analyzer profile named default is created.

```
config image-analyzer profile
    edit "default"
        set comment "Analyze image content"
        set alcohol-status allow
        set drugs-status allow
```

```
        set extremism-status allow
        set gambling-status allow
        set gore-status allow
        set porn-status allow
        set swim_underwear-status allow
        set weapons-status allow
        set log-option all
        set blocked-img-cache enable
        set rating-err-action block
        set optical-character-recognition enable
        set ocr-activation-threshold 100
    next
end
```

2. On FortiProxy, configure the ICAP service to use the image-analyzer profile:
   The *icap-service* is configured to use the image-analyzer profile named default.

```
config icap local-server
    edit 1
        set interface "port1"
        set incoming-ip 10.211.255.147
        set srcaddr "all"
        config icap-service
            edit 1
                set name "profile"
                set dlp-profile "default"
                set image-analyzer-profile "default"
            next
        end
    next
end
```

Web application firewall

Web application firewall (WAF) profiles can detect and block known web application attacks. You can configure WAF profiles to use signatures and constraints to examine web traffic. You can also enforce an HTTP method policy, which controls the HTTP method that matches the specified pattern.

You can customize the default profile, or you can create your own profile to apply access rules and HTTP protocol constraints to traffic. You can apply WAF profiles to firewall policies when the inspection mode is set to proxy-based.

WAF signatures are pre-loaded on the FortiGate and do not require a license to use. However, signatures are static and do not receive updates. For most updated WAF signatures, please use FortiWeb.

Web application firewall profiles cannot be used in NGFW policy-based mode. See NGFW policy on page 1868 for more information.

This feature is not supported on FortiGate models with 2 GB RAM or less. See Proxy-related features not supported on FortiGate 2 GB RAM models on page 126 for more information.

The following topic provides information about WAF profiles:

- Protecting a server running web applications on page 2600

Protecting a server running web applications

You can use a web application firewall profile to protect a server that is running a web application, such as webmail.

Web application firewall profiles are created with a variety of options called signatures and constraints. Once these options are enabled, the action can be set to allow, monitor, or block. The severity can be set to high, medium, or low.

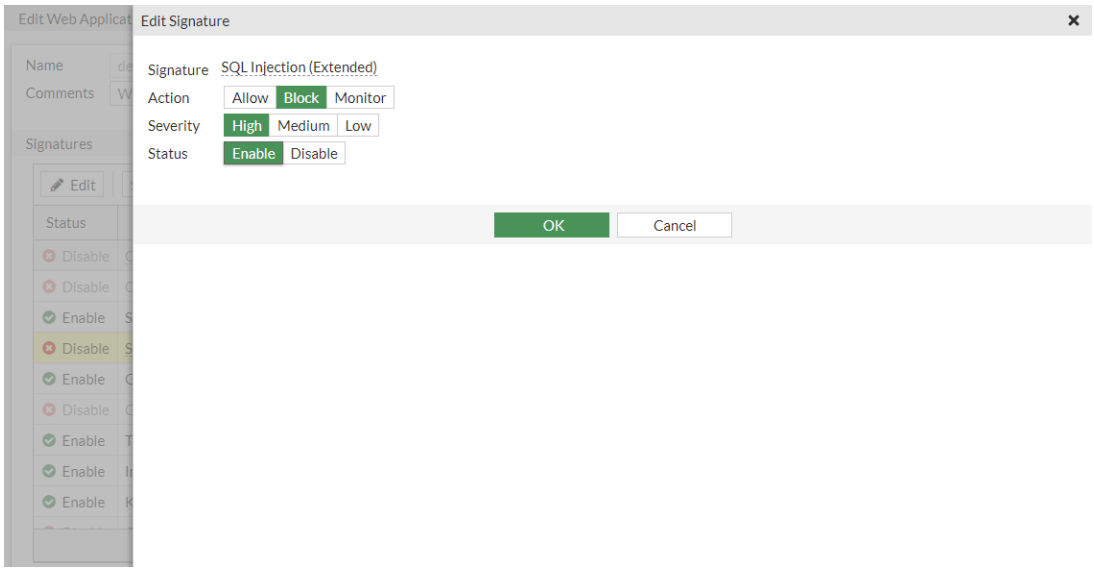
In the following example, the default profile will be targeted to block SQL injection attempts and generic attacks.

The web application firewall feature is only available when the policy inspection mode is proxy-based.

To protect a server running web applications:

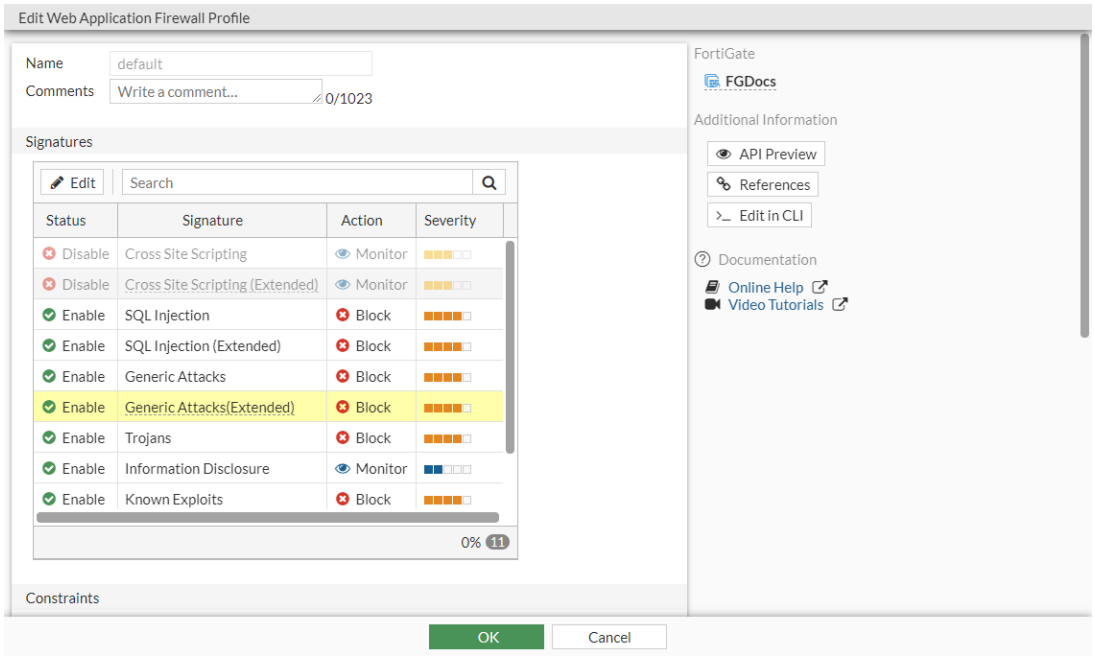
1. Enable the web application firewall:
   a. Go to *System* > *Feature Visibility*.
   b. Under *Security Features*, enable *Web Application Firewall*.
   c. Click *Apply*.
2. Edit the default web application firewall profile (*Trojans* and *Known Exploits* are blocked by default):
   a. Go to *Security Profiles* > *Web Application Firewall* and edit the *default* profile signature.
   b. Select *SQL Injection (Extended)* and edit it so that it is enabled, the *Action* is set to *Block*, and the *Severity* is set to *High*.

c. Click *OK*.



d. Enable *Generic Attacks (Extended)* and edit it so that it is enabled, the *Action* is set to *Block*, and the *Severity* is set to *High*.

e. Click *OK*.



f. Click *OK*.

3. Apply the profile to a security policy:
   a. Go to *Policy & Objects* > *Firewall Policy* and edit the policy that allows access to the web server.
   b. For *Firewall / Network Options*, select the appropriate *Protocol Option*.
   c. For *Security Profiles*, enable *Web Application Firewall* and set it to use the *default* profile.
   d. Set the *SSL Inspection* to use the *deep-inspection* profile.
   e. Configure the other settings as needed.
   f. Click *OK*.

4. Verify that the web application firewall blocks traffic:
   a. Use the following URL to simulate an attack on your web server and substitute the IP address of your server: `http://<server IP>/index.php?username=1'%20or%20'1'%20=%20'1&password=1'%20or%20'1'%20=%20'1`
   An error message appears, stating that the web application firewall has blocked the traffic:

Using FortiWeb for protection

Another way of protecting web applications is to forward HTTP traffic to a FortiWeb for scanning and inspection. A typical use case is to use a one-arm topology with FortiWeb running in reverse proxy mode to scan traffic before accessing the webpage on the web servers. See Planning the network topology in the FortiWeb Administration Guide for more information.

Data loss prevention

The FortiGate data loss prevention (DLP) system prevents sensitive data from leaving or entering your network by scanning for various patterns while inspecting traffic passing through the FortiGate. Data that matches defined sensitive data patterns is blocked, logged, allowed, or quarantined when it passes through the FortiGate.

The DLP system is configured based on the following components:

Component	Description
Data type	Define the type of pattern that DLP is trying to match. For example, this can be a predefined type such as keyword, regex, hex, credit card, US social security number (SSN), or other patterns. You can also create custom data types.
Dictionary	A collection of data type entries. When selecting a data type such as keyword, regex or hex, define the pattern that you are looking for.
Label	The following types of labels are supported to help classify and protect sensitive information: • Microsoft Information Protection (MPIP) • FortiData MPIP labels can be defined locally by configuring the Globally Unique Identifier (GUID) or synchronized automatically from a Microsoft Purview account through the Azure SDN connector. FortiData labels are defined by FortiData and used by FortiGate when the two products are integrated.

Component	Description
EDM template	An exact data match (EDM) template pairs the data from an external file, such as a data external feed file, with built-in FortiGate data types. The EDM template can link to a file on an external server to support dynamic updates, or the file can be uploaded to the EDM template.
Sensor	Define which dictionaries and/or EDM templates to check. Sensors can consist of dictionaries and EDM templates. You can match any dictionary or EDM template, all dictionaries and/or EDM templates, or a special logical combination of the dictionaries and/or EDM templates. Sensors can also count the number of matches to trigger the sensor.
File pattern	Define groups of file patterns based on predefined file types, or define your own pattern to match the file name.
DLP profile	Define rules for matching a sensor based on a file type or a message, and the type of protocol being used. It also allows you to choose the action to allow, log, block, or quarantine the address.

A DLP profile selects one or more sensors, and applies the sensor's pattern matching against the file type or message that is passing through selected protocols. The profile can be applied to a firewall policy where the traffic will be inspected.

In the backend, DLP uses Hyperscan to perform a one-parse algorithm for scanning multiple patterns. This allows DLP to scale up without any performance downgrade.

The FortiGuard Data Loss Prevention service enables the identification, monitoring, and protection of an organization's data through data breaches, insider threats, and data exfiltration. It uses a customizable database of more than 500 predefined data patterns and policies to simplify and expedite DLP deployment and integration into existing environments (see FortiGuard DLP service on page 2619 for more information).

Protocol comparison between DLP inspection modes

The following table indicates which protocols can be inspected by DLP based on the specified inspection modes.

	HTTP	FTP	IMAP	POP3	SMTP	NNTP	MAPI	CIFS	SFTP/SCP
Proxy	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
Flow	Yes	Yes	Yes	Yes	Yes	Yes	No	Yes	No

DLP can be configured in both the CLI and the GUI irrespective or firewall policy inspection mode.

When using DLP in flow-based inspection mode, only file scanning is supported. As a result, any DLP profile applied under a flow-based firewall policy must have its *Match type* set to *File* in the rule configuration under the DLP profile.

To use DLP profiles in a flow-based firewall policy, `set feature-set flow` must be set from the CLI. See Configuring DLP from the CLI on page 2609 for more information.

DLP profiles can only be added to a flow-based firewall policy from the CLI.

Archiving

DLP can archive some or all of the content that passes through the DLP system. There are two forms of DLP archiving.

- Summary only: a summary of all the activity detected by the profile is recorded. For example, when an email message is detected, the sender, recipient, message subject, and total size are recorded. When a user accesses a web browser, every URL that they visit is recorded.
- Full: detailed records of all the activity detected by the profile is recorded. For example, when an email message is detected, the message itself, including any attachments, is recorded. When a user accesses a web browser, every page that they visit is archived.

You can configure the type of archiving per protocol.

Logging and blocking files by file name

Sometimes, file names are not accurately recorded in DLP logs, even though the files are blocked correctly based on the DLP profile. This is particularly apparent on cloud-based services, such as Google Drive or SharePoint.

For HTTP file uploads, some cloud services use proprietary encodings and APIs to transfer files and exchange metadata, instead of standard HTTP mechanisms, requiring custom handling of the proprietary API. If a cloud service changes the API without notice, the custom handling becomes outdated and file names might not be logged properly. Due to this, special consideration must be taken when using DLP to block files by file pattern. To block a specific file type, it is better to block by file type, and not by file name pattern.

The following topics provide information about DLP:

- DLP techniques on page 2605
- Basic DLP settings on page 2606
- Advanced DLP configurations on page 2612
- DLP fingerprinting on page 2615
- FortiGuard DLP service on page 2619
- Sensitivity labels on page 2623
- Exact data matching on page 2631
- DLP examples on page 2640

DLP techniques

The security of sensitive data is a top priority for organizations. A range of techniques and tools are used to maintain the confidentiality and accessibility of data.

The following table describes some of the industry standard techniques that are used for data loss protection, and if they can be configured in the GUI or CLI.

Technique	Description	GUI	CLI
Indexed Document Matching (IDM)	IDM creates unique fingerprints for your organization's crucial documents that hold sensitive information. This process involves analyzing the content of these documents and generating a checksum for each one. See DLP fingerprinting on page 2615 for more information.		✓
Exact Data Matching (EDM)	EDM identifies particular data values within an indexed data source that require safeguarding. See Exact data matching on page 2631 for more information.	✓	✓
Described Content Matching (DCM)	DCM scans through data to identify the presence of specific patterns using regular expressions (Regex). See Built-in DLP data type on page 2612 for more information.	✓	✓
Optical Character Recognition (OCR)	OCR scans and analyzes the content embedded within images for sensitive information, extending data protection to image-based content.		
Predefined data patterns	Default DLP patterns that classify private and confidential data that should be regulated in accordance with regulatory compliance requirements. See Built-in DLP data type on page 2612 for more information.	✓	✓
Custom data classification tags (data pattern)	FortiGate allows you to create patterns for your custom data type. See Custom DLP data type on page 2613 for more information.	✓	✓
True file type filtering	Identify a file by the data type in its meta data. See DLP file pattern on page 2614 for more information.	✓	✓
File size filtering	Identify a file based on its size. See Block HTTPS downloads of EXE files and log HTTPS downloads of files larger than 500 KB on page 2648 for an example.		✓
Microsoft Purview sensitivity labels	Sensitivity labels provide a mechanism to categorize and safeguard your data. They function as identifiers and highlight the significance of the data that they are attached to. See Sensitivity labels on page 2623 for more information.	✓	✓

Technique	Description	GUI	CLI
FortiData labels	A Fortinet product that discovers, classifies, and labels files with sensitive data in your file storage system. Integration of FortiData with FortiGate lets you use FortiData labels for DLP processing in FortiGate policies. See FortiData labels on page 2662 for more information.		✓
FortiGuard DLP service	A database of predefined DLP patterns, such as data types, dictionaries, and sensors, that are dynamically managed by FortiGuard. A valid DLP license is required. See FortiGuard DLP service on page 2619 for more information.	✓	✓

Basic DLP settings

DLP settings can be configured for data types, dictionaries, EDM templates, sensors, file patterns, and profiles. DLP can be configured in both the CLI and the GUI irrespective of firewall policy inspection mode.

When using DLP in flow-based inspection mode, only file scanning is supported. As a result, any DLP profile applied under a flow-based firewall policy must have its *Match type* set to *File* in the rule configuration under the DLP profile.

To use DLP profiles in a flow-based firewall policy, `set feature-set flow` must be set from the CLI. See Configuring DLP from the CLI on page 2609 for more information.

DLP profiles can only be added to a flow-based firewall policy from the CLI.

On the *Security Profiles* > *Data Loss Prevention* page, there are *Profile*, *Sensor*, *Dictionary*, and *EDM Template* tabs to configure those DLP settings. DLP profiles can be added to proxy-based firewall policies and proxy policies from the GUI.

If *Data Loss Prevention* is not visible in the tree menu, go to *System* > *Feature Visibility* and enable it.

This section breaks down the DLP configuration into a sequence of steps:

1. Configure the DLP dictionary, DLP labels, and/or EDM template:
   - A DLP dictionary is a collection of data type entries. See Built-in DLP data type on page 2612 for more information.
   - A DLP label is used to configure MPIP labels, which help classify and protect sensitive information.
   - An EDM template pairs the data from an external file, such as a data external feed file, with built-in data types.
2. Configure the DLP sensor:
   - A DLP sensor defines which dictionary and/or EDM template to check. It counts the number of matches to trigger the sensor.
3. Configure the DLP profile:
   - A DLP profile allows for filtering by size and file type. See DLP file pattern on page 2614 for custom file type.
4. Add the DLP profile to a firewall policy.

All the steps mentioned above should be configured in the exact order given for ease of configuration.

Configuring DLP from the GUI

Use the following steps to configure DLP from the GUI.

To configure a DLP dictionary:

1. Go to *Security Profiles* > *Data Loss Prevention*.
2. Select the *Dictionary* tab and click *Create New*.
3. Enter a name.
4. In the *Dictionary Entries* section, click *Create New*.
5. Set the *Type* and click *OK*.
6. Click *OK* to save the dictionary.

To configure an EDM template:

1. Go to *Security Profiles* > *Data Loss Prevention*.
2. Select the *EDM Template* tab and click *Create New*.
3. Enter a name.
4. In the *Resource settings* section, select one of the following:

File upload	Select to upload an external file of data to use with built-in data types. The external file can be in text (TXT) or comma-separated value (CSV) format.
External feed	Select to provide the URL to a file of data on an external server to use with built-in data types. The external file must be in comma-separated value (CSV) format. FortiGate will periodically fetch entries from the external file using HTTPS.
External feed URL	Specify the URL to the data file in CSV format on the external server.
HTTP basic authentication	Enable to use basic HTTP authentication when accessing the file on the external server. Specify the username and password for the external server.
Refresh rate	Specify the time interval to refresh the external resource (minutes).

5. Set the *Match criteria* section:
   Each column in the external file represents data for a built-in data type. The patterns in the data file must be valid for the data type. If the patterns are invalid, FortiGate cannot use them, and no warning is displayed.

+All of these fields	Click to pair each column in the external data file with a built-in data type. All of the specified data in this section must match for FortiGate to take an action.

Column index	Specify the column number in the external file that contains the data.
Data type	Indicate which built-in data type pairs with the column index. Choose from: • credit-card • edm-keyword • mip-label • ssn-us
+Any of these fields	Click to pair the column in the external data file with a built-in data type, and to specify how many of these pairs must match for FortiGate to take an action.
Minimum number of fields matched	Specify how many of the fields in the *Any of these fields* section must match for FortiGate to take an action.
Column index	Specify the column number in the external file that contains the data.
Data type	Indicate which built-in data type pairs with the column index. Choose from: • credit-card • edm-keyword • mip-label • ssn-us • fg-edm-can-natl_id-sin The data type *fg-edm-can-natl_id-sin*, which represents the Canadian Social Insurance Number (SIN), is dynamically managed by FortiGuard. It is available for use as one of the data types in EDM templates, provided the user has a valid FortiGuard DLP service license.

6. Click *OK* to save the EDM template.

To configure a DLP sensor:

1. Go to *Security Profiles* > *Data Loss Prevention*.
2. Select the *Sensor* tab and click *Create New*.
3. Enter a name.
4. In the *Sensors Entries* section, click *Add*. The *Select Entries* pane is displayed.
5. Select the management method from the dropdown menu.
6. From the list, select the dictionary or EDM template.
7. Click *Apply*.
8. Click *OK* to save the sensor.

To configure a DLP profile:

1. Go to *Security Profiles* > *Data Loss Prevention*.
2. Select the *Profile* tab and click *Create New*.

3. Enter a name.
4. In the *Rules* table, click *Create New*.
5. Configure the following settings:

Name	Filter name.
Type	Specify what type of data source to use: • *Sensor*: Use DLP sensors, such as dictionaries or EDM templates to match content. • *Label*: Use DLP label to match content.
Sensors	Select one or more DLP sensors when *Type* is set to *Sensor*.
DLP label	Select one of the DLP label from the drop-down list when *Type* is set to *Label*.
Severity	Select the severity or threat level that matches this filter.
Action	Action to take with content that this DLP profile matches.
Match type	Select whether to check the content of messages (an email message) or files (downloaded files or email attachments).
File type	Select the number of a DLP file pattern table to match.
Protocol	Check messages or files over one or more of these protocols.

6. Click *OK*.
7. Click *OK* to save the profile.

To add the DLP profile to a firewall policy:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Click *Create New*.
3. Set the *Inspection Mode* to *Proxy-based*.
4. In the *Security Profiles* section, enable *DLP Profile* and select the desired profile.
5. Configure the other settings as needed.
6. Click *OK*.

Configuring DLP from the CLI

Use the following steps to configure DLP from the CLI.

To configure a DLP dictionary:

```
config dlp dictionary
    edit <name>
        config entries
            edit 1
                set type {credit-card | hex | keyword | regex | ssn-us}
                set pattern <string>
```

```
                set repeat {enable | disable}
                set status {enable | disable}
            next
        end
    next
end
```

To configure a DLP label:

```
config dlp label
    edit <name>
        set type mpip
        set mpip-type {remote | local}
        set connector <string>
        config entries
            edit <id>
                set guid <string>
                set mpip-label-name <string>
            next
        end
    next
end
```

`guid` is only configurable when `mpip-type` is set to `local`. `mpip-label-name` is only configurable when `mpip-type` is set to `remote`.

For the remote MPIP label, the Azure SDN connector needs to be configured with Microsoft 365 enabled. See Azure SDN connector using service principal on page 4611 for more information.

To configure an EDM template:

When configuring an EDM template from the CLI, you must link to a data file in CSV format on an external server; you cannot upload the data file to FortiGate.

1. Add the URL for the data external feed file to FortiGate.

```
config system external-resource
    edit <name>
        set type data
        set resource <URL to resource file on external server>
        end
    next
end
```

2. Configure the EDM template.

```
config dlp exact-data-match
    edit <name>
        set optional <number of optional columns that must match>
        set data <name of external resource file>
        config columns
            edit <column index number>
```

```
                set type {credit-card | edm-keyword | mip-label | ssn-us | fg-edm-can-natl_id-
sin}
            next
        end
    next
end
```

The data type `fg-edm-can-natl_id-sin`, which represents the Canadian Social Insurance Number (SIN), is dynamically managed by FortiGuard. It is available for use as one of the data types in EDM templates, provided the user has a valid FortiGuard DLP service license.

To configure a DLP sensor:

```
config dlp sensor
    edit <name>
        set match-type {match-all | match-any | match-eval}
        set eval <string>
        config entries
            edit <id>
                set dictionary <dlp dictionary or EDM template>
                set count <integer>
                set status {enable | disable}
            next
        end
    next
end
```

See Evaluation by logical relationship on page 2614 for more information about `match-eval`.

To configure a DLP profile:

```
config dlp profile
    edit <name>
        set feature-set {flow | proxy}
        config rule
            edit <id>
                set proto <protocol> <protocol> ...
                set sensor <dlp_sensor>
                set action {allow | log-only | block | quarantine-ip}
            next
        end
    next
end
```

To add the DLP profile to a firewall policy:

```
config firewall policy
    edit <id>
        set srcintf <interface>
```

```
        set dstintf <interface>
        set action accept
        set srcaddr <address>
        set dstaddr <address>
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set dlp-profile <string>
    next
end
```

See DLP examples on page 2640 for sample configurations.

Advanced DLP configurations

The following topic provides information on advanced DLP configurations.

Built-in DLP data type

Built-in DLP data type includes predefined data types to match for keyword, regex, hex, mip label, credit card, and social security number (SSN). See Predefined data patterns on page 2605 for more information. The built-in DLP data type regex employs DCM to detect patterns. See Described Content Matching (DCM) on page 2605 for more information.

```
config dlp data-type
    edit "keyword"
        set pattern "built-in"
    next
    edit "regex"
        set pattern "built-in"
    next
    edit "hex"
        set pattern "built-in"
    next
    edit "mip-label"
        set pattern "^[[:xdigit:]]{8}-[[:xdigit:]]{4}-[[:xdigit:]]{4}-[[:xdigit:]]{4}-[[:xdigit:]]
{12}$"
        set transform "built-in"
    next
    edit "credit-card"
        set pattern "\\b([2-6]{1}\\d{3})[- ]?(\\d{4})[- ]?(\\d{2})[- ]?(\\d{2})[- ]?(\\d{2,4})\\b"
        set verify "built-in"
        set look-back 20
        set transform "\\b\\1[- ]?\\2[- ]?\\3[- ]?\\4[- ]?\\5\\b"
    next
    edit "ssn-us"
        set pattern "\\b(\\d{3})-(\\d{2})-(\\d{4})\\b"
        set verify "(?<!-)\\b(?!666|000|9\\d{2})\\d{3}-(?!00)\\d{2}-(?!0{4})\\d{4}\\b(?!-)"
```

```
        set look-back 12
        set transform "\\b\\1-\\2-\\3\\b"
    next
end
```

Custom DLP data type

Custom data types can be added. See Custom data classification tags (data pattern) on page 2605 for more information.

Custom DLP data type allows for both the proximity keyword check and data validation check within the same data type. The data type simultaneously supports two verification checks and one proximity match check to significantly lower the occurrence of false positives, enhancing the precision and dependability of the search.

To configure a custom DLP data type used by DLP scans:

```
config dlp data-type
    edit <name>
        set verify <string>
        set verify2 <string>
        set look-ahead <integer>
        set look-back <integer>
        set match-around <string>
        set match-ahead <integer>
        set match-back <integer>
        set pattern <string>
    next
end
```

Command	Description
`<name>`	The name of the table containing the data type.
`pattern <string>`	Specify the regular expression pattern string without look around
`verify <string>`	Specify the regular expression pattern string used to verify the data type.
`verify2 <string>`	Specify the extra regular expression pattern string used to verify the data type.
`look-ahead <integer>`	Specify the number of character to obtain in advance for verification (1 - 255, default = 1).
`look-back <integer>`	Specify the number of characters required to save for verification (1 - 255, default = 1).
`match-around <string>`	Dictionary to check whether it has a match around (only support match-any and basic types, no repeat supported).
`match-back <integer>`	Specify the number of characters in front for `match-around` (1 - 4096, default = 1).
`match-ahead <integer>`	Specify the number of characters behind for `match-around` (1 - 4096, default = 1).

The `set pattern` command can be used to define a regular expression for use with Hyperscan. However, Hyperscan does not fully support Perl Compatible Regular Expressions (PCRE), such as `look-ahead` and `look-behind`. To use advanced features supported by PCRE, you can use the `set verify` or `set verify2` commands.

To use "?" in a regex pattern, see CLI basics on page 76. This method only supports direct console connection and SSH. It does not support the CLI console in the GUI.

See Proximity search on page 2654 for a sample configuration.

DLP file pattern

A DLP file pattern can block, allow, log, or quarantine a file based on the specified file type in the file filter list (see Supported file types on page 2520). It employs True file type filtering to identify a file. See True file type filtering on page 2605 for more information.

To configure a DLP file pattern:

```
config dlp filepattern
    edit <id>
        set name <name>
        config entries
            edit <file name pattern>
                set filter-type {type | pattern}
                set file-type <file_type>
            next
        end
    next
end
```

Command	Description
`filter-type {type | pattern}`	Filter by file name pattern or by file type.
`file-type <file_type>`	Select a file type. This option is only available when `filter-type` is set to `type`.

See config dlp filepattern in the CLI Reference guide for a comprehensive list of commands.

Evaluation by logical relationship

Evaluation by logical relationship is a powerful tool used to combine multiple dictionary entries to define an accurate DLP sensor using logical expression.

Syntax examples:

1. `set eval "dict(1) == 2"`
   Match DLP sensor only when dictionary one match count is two.

2. `set eval "(dict(1) + dict(2)) == 3"`
   Match DLP sensor only when dictionary one and dictionary two combined match count is three.
3. `set eval "(dict(1) == 2) && (dict(2) == 1)"`
   Match DLP sensor only when dictionary one match count is equal to two and dictionary two match count is equal to one.
4. `set eval "(dict(1) == 2) || (dict(2) == 1)"`
   Match DLP sensor only when dictionary one match count is equal to two or dictionary two match count is equal to one.
5. `set eval "dict(1) > dict(2)"`
   Match DLP sensor only when dictionary one match count is greater than dictionary two match count.

See Block HTTPS upload traffic that includes Visa or Mastercard information using evaluation through logical expression on page 2650.

DLP fingerprinting

DLP fingerprinting employs Indexed Document Matching (IDM) to detect sensitive data. See Indexed Document Matching (IDM) on page 2605 for more information. The file that the DLP profile filters is uploaded and the FortiGate generates and stores a checksum fingerprint. The FortiGate generates a fingerprint for all the files that are detected in network traffic, and compares all the checksums stored in its database. If a match is found, the configured action is taken. Any type of file can be detected by DLP fingerprinting, and fingerprints can be saved for each revision of a file as it is updated.

Using fingerprinting requires:

1. Creating a DLP fingerprint database by allowing the FortiGate to access a file server containing files from which to create fingerprints.
2. Adding fingerprinting filters to DLP profiles.
3. Adding the profiles to firewall policies that accept traffic that the fingerprinting will be applied on.

See Fingerprinting example on page 2617 for a sample configuration.

The document fingerprint feature requires a FortiGate that has internal storage.

To configure a DLP fingerprint document:

```
config dlp fp-doc-source
    edit <name>
        set server <string>
        set username <string>
        set password <password>
        set file-path <string>
        set sensitivity <Critical | Private | Warning>
    next
end
```

Command	Description
`server <string>`	Enter the IPv4 or IPv6 address of the file server.
`username <string>`	Enter the user name required to log into the file server.
`password <password>`	Enter the password required to log into the file server.
`file-path <string>`	Enter the path on the server to the fingerprint files.
`sensitivity <Critical | Private | Warning>`	Set the sensitivity or threat level for matches with this fingerprint database.

See config dlp fp-doc-source in the FortiOS CLI Reference for a comprehensive list of commands and supported FortiGate models.

A file server is required for the user to upload files. Each uploaded file will have a fingerprint generated by FortiGate, and will be stored locally as a checksum. Currently, only servers that are using the Samba (SMB) protocol are compatible.

To configure a DLP fingerprint profile:

```
config dlp profile
    edit <name>
        set feature-set proxy
        config rule
            edit <id>
                set proto {smtp | pop3 | imap | http-get | http-post | ftp | nntp | mapi | ssh |
cifs}
                set filter-by fingerprint
                set sensitivity {Critical | Private | Warning}
                set match-percentage <integer>
                set action {allow | log-only | block | quarantine-ip}
            next
        end
    next
end
```

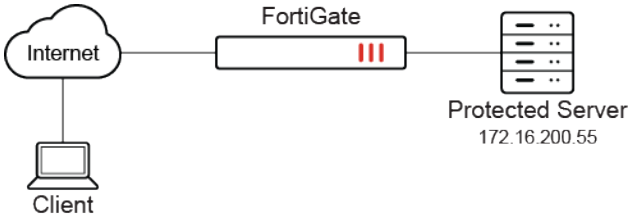
Command	Description
`proto {smtp | pop3 | imap | http-get | http-post | ftp | nntp | mapi | ssh | cifs}`	Set the protocol to inspect.
`filter-by fingerprint`	Set to match against a fingerprint sensitivity.
`sensitivity {Critical | Private | Warning}`	Set the DLP file pattern sensitivity to match.
`match-percentage <integer>`	Set the percentage of the checksum required to match before the profile is triggered.
`action {allow | log-only | block | quarantine-ip}`	Set the action to take with content that matches the DLP profile.

Fingerprinting example

This configuration will block HTTPS download traffic that matches the checksums that are stored in the FortiGate fingerprint database.

When utilizing commonly-used SSL-encrypted protocols, such as HTTPS, SMTPS, POP3S, IMAPS, and FTPS, SSL inspection must be set to Deep Inspection. See Deep inspection on page 2697 for more information.

The client machine must also have the corresponding deep inspection Certificate Authority (CA) certificate installed.



In this example, a text document with sensitive data is being downloaded by the client using the HTTP GET method. The term *Protected Server* refers to the Samba file server that stores the fingerprint files. It is assumed that you already have a configured Samba file server.

The FortiGate intercepts the traffic using deep inspection and blocks the traffic as it matches the DLP profile configured on this FortiGate. See Sample log on page 2619 for a log sample.

To block network traffic that matches the checksums stored in the FortiGate fingerprint database:

1. Configure the DLP fingerprint database:

```
config dlp fp-doc-source
    edit "test"
        set server "172.16.200.55"
        set username "kiki"
        set password *****
        set file-path "/sambashare/upload/"
        set sensitivity "Critical"
    next
end
```

   This step can only be configured in the CLI.

2. Configure the DLP profile:

```
config dlp profile
    edit "fingerprint"
        set feature-set proxy
        config rule
            edit 1
                set proto http-get
                set filter-by fingerprint
```

```
                set sensitivity "Critical"
                set action block
            next
        end
    next
end
```

DLP profiles that filter by fingerprint can only be configured in the CLI.

3. Add the DLP profile to a firewall policy:

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "deep-inspection"
        set dlp-profile "fingerprint"
        set nat enable
    next
end
```

This can also be configured in the GUI. See To add the DLP profile to a firewall policy.

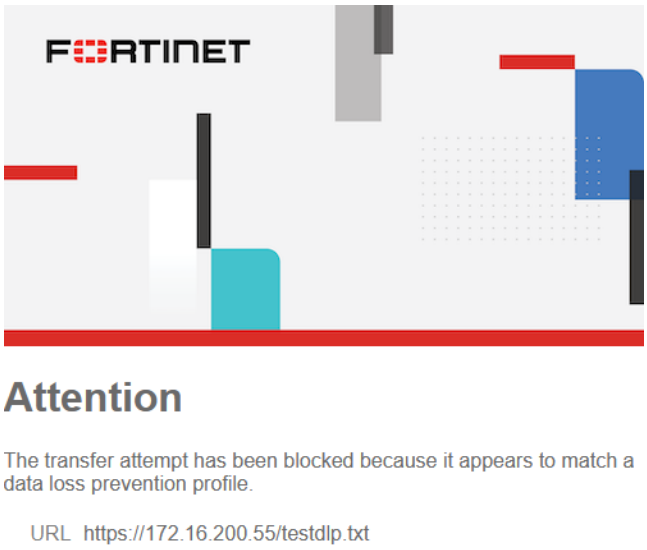
To verify the results:

1. Verify that the DLP fingerprint database is present on the FortiGate:

```
# diagnose test application dlpfingerprint 3
File DB:
----------------------------------------
id,     filename,       vdom,   archive,        deleted,        scanTime,       docSourceSrvr,
 sensitivity,    chunkCnt,       reviseCnt,
1,      /sambashare/upload/testdlp,     root,   0,      0,      1706727347,     test,   2,
 1,      0,
2,      /sambashare/upload/testdlp.txt, root,   0,      0,      1706728230,     test,   2,
 1,      0,
```

2. Verify HTTP GET traffic that matches the checksums stored in the FortiGate fingerprint database is being blocked:

   A download attempt of a text file from a Windows device was made using Chrome browser. This text file is located on the protected server and its fingerprint is saved in the FortiGate fingerprint database.



The download was unsuccessful, leading to the creation of a sample log. See Sample log on page 2619.

Sample log

To view the sample log:

1. Go to *Log & Report* > *Security Events* and select *AntiVirus*.
2. View the log details in the GUI, or download the log file:

```
1: date=2024-02-01 time=08:47:25 eventtime=1706734045777192462 tz="+1200" logid="0954024576"
type="utm" subtype="dlp" eventtype="dlp" level="warning" vd="root" ruleid=1
dlpextra="Critical" filtertype="fingerprint" filtercat="file" severity="medium" policyid=1
poluuid="f4fe48a4-938c-51ee-8856-3e84e3b24af4" policytype="policy" sessionid=308873
epoch=813849496 eventid=0 srcip=13.13.13.13 srcport=51058 srccountry="United States"
srcintf="port2" srcintfrole="undefined" srcuuid="d2f06fda-15e7-51ee-0d22-faaf5170dad2"
dstip=172.16.200.55 dstport=443 dstcountry="Reserved" dstintf="port1" dstintfrole="undefined"
dstuuid="d2f06fda-15e7-51ee-0d22-faaf5170dad2" proto=6 service="HTTPS" filetype="unknown"
direction="incoming" action="block" hostname="172.16.200.55"
url="https://172.16.200.55/testdlp.txt" agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/121.0.0.0 Safari/537.36 Edg/121.0.0.0"
httpmethod="GET" filename="testdlp.txt" filesize=20 profile="fingerprint"
```

FortiGuard DLP service

The FortiGuard DLP service offers a database of predefined DLP patterns such as data types, dictionaries, and sensors. Example include:

- Drivers licenses for various countries, various states in the USA, and various provinces in Canada
- Tax numbers for various countries
- Credit card numbers
- Bank statements

When enabled, the DLP database (DLDB) is downloaded to the FortiGate and its predefined patterns can be configured in DLP profiles.

To configure DLP database updates:

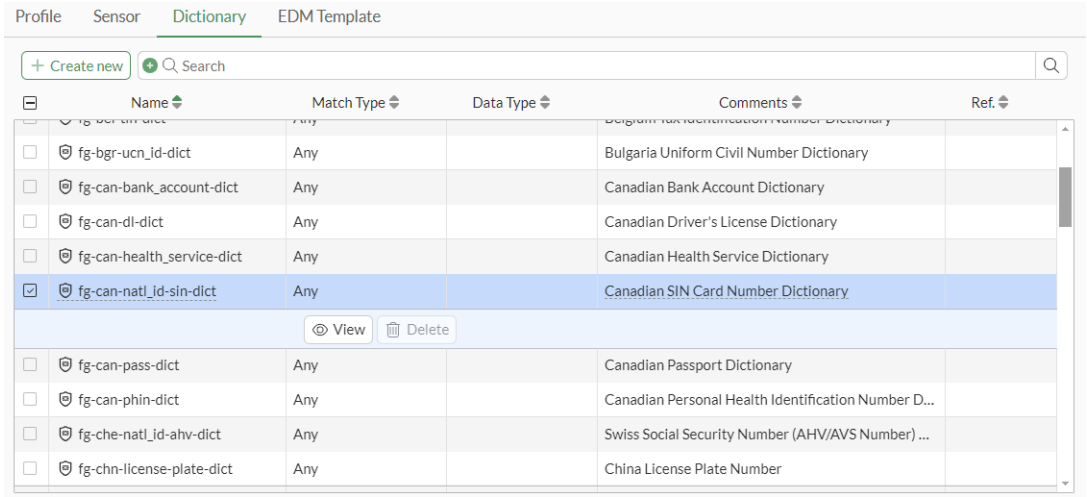
```
config system fortiguard
    set update-dldb {enable | disable}
end
```

To verify the database signature status:

```
# diagnose autoupdate versions
...
DLP Signature
---------
Version: 1.00010 signed
Contract Expiry Date: n/a
Last Updated using manual update on Fri Jan 27 15:25:00 2023
Last Update Attempt: Mon Jan 30 15:18:39 2023
Result: No Updates
```

Example

In this example, the administrator wants to look for data leakage of Canadian social insurance number (SIN) information and block this traffic. A DLP profile is created that uses the predefined dictionary, *fg-can-natl_id-sin-dict*, to check for Canadian Social Insurance Numbers (SINs).



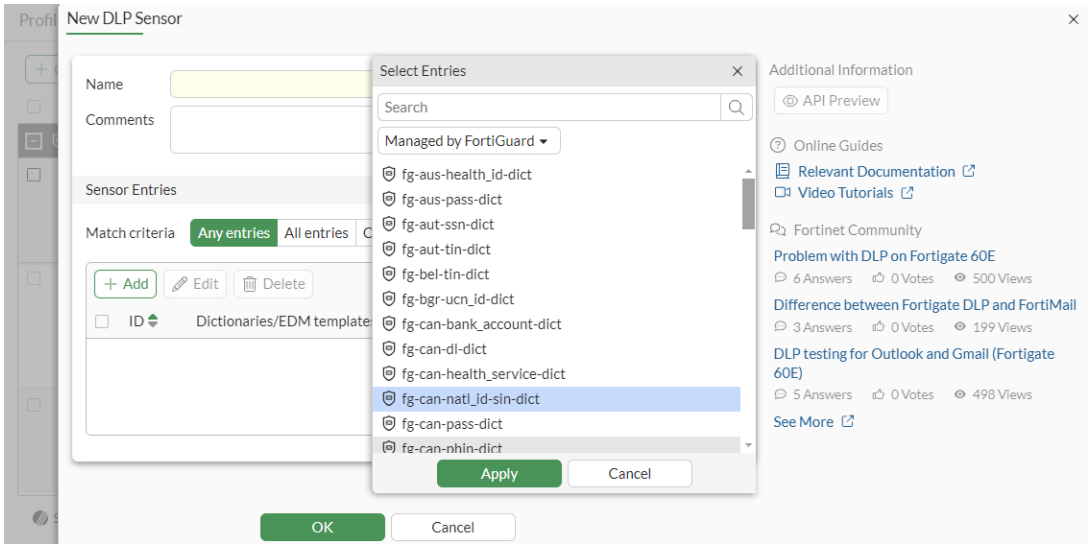
To verify that the Canadian SIN data type is added to the list of predefined data types:

```
show dlp data-type
config dlp data-type
    ...
    edit "fg-can-natl_id-proximity"
```

```
        set pattern "fortiguard dlp signature"
    next
end
```

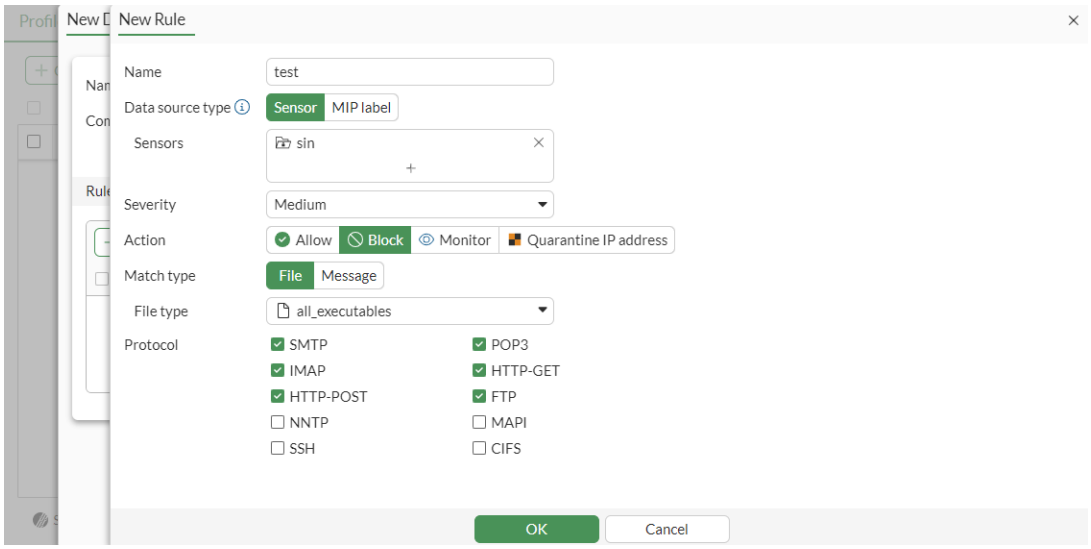
To configure the DLP profile in the GUI:

1. Configure the DLP sensor using the predefined dictionary from FortiGuard:
   a. Go to *Security Profiles* > *Data Loss Prevention*, select the *Sensor* tab, and click *Create New*.
   b. Enter a name (*sin*).
   c. In the *Sensor Entries* section, click *Add*. The *Select Entries* pane is displayed.
   d. Select *Managed by FortiGuard* from the dropdown menu.
   e. Select *fg-can-natl_id-sin-dict* and click *Apply*.



   f. Click *OK* to save the sensor.
2. Configure the DLP profile:
   a. Go to *Security Profiles* > *Data Loss Prevention*, select the *Profile* tab, and click *Create New*.
   b. Enter a name (*test*).
   c. In the *Rules* section, click *Create New*.
   d. Configure the following settings:

Name	*test*
Sensors	*sin*
Severity	*Medium*
Action	*Block*
Type	*File*
File type	*all_executables*
Protocol	*SMTP*, *POP3*, *IMAP*, *HTTP-GET*, *HTTP-POST*, *FTP*



e. Click *OK*.

f. Click *OK* to save the profile.

To configure the DLP profile in the CLI:

1. Configure the DLP sensor using the predefined dictionary from FortiGuard:

```
config dlp sensor
    edit "sin"
        config entries
            edit 1
                set dictionary "fg-can-natl_id-sin-dict"
            next
        end
    next
end
```

2. Configure the DLP profile:

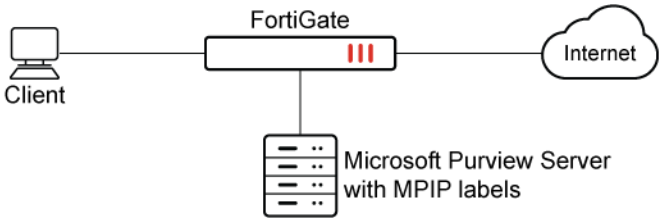
```
config dlp profile
    edit "test"
        set feature-set proxy
        config rule
            edit 1
                set name "test"
                set proto smtp pop3 imap http-get http-post ftp
                set filter-by sensor
                set file-type 2
                set sensor "sin"
                set action block
            next
        end
    next
end
```

Sensitivity labels

In order to safeguard your organization's data, labels can be employed as markers for sensitive information. Microsoft provides sensitivity labels, which act as identifiers emphasizing the importance of the data they're associated with, thereby enhancing the security measures in place. See Protect your sensitive data with Microsoft Purview (formerly MIP) for more information.

Any data traffic that includes a sensitivity label can be effectively managed using FortiGate. This is made possible through the utilization of a predefined label type, *mpip*, specifically designed for Microsoft Purview Information Protection. See Microsoft Purview sensitivity labels on page 2605 for more information.

MPIP labels have their own settings and can be used directly with DLP profiles without needing a dictionary. Additionally, remote MPIP labels can be synchronized automatically from a Microsoft Purview account through the Azure SDN connector, complementing locally defined labels.



Prerequisites

The following procedures should be completed before configuring the FortiGate:

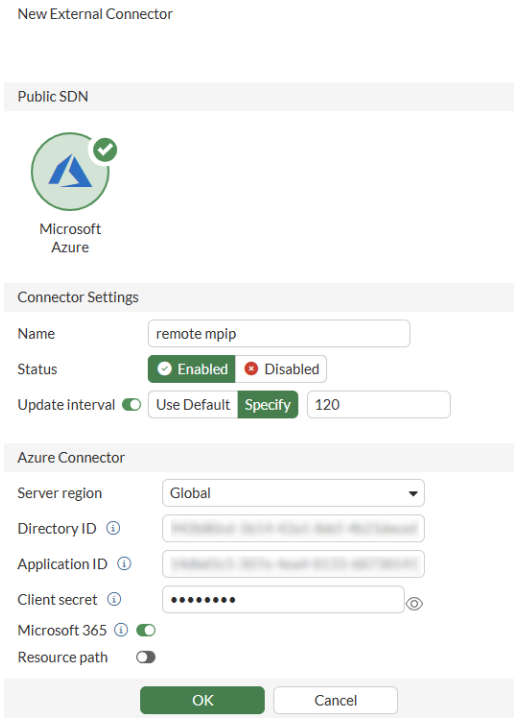
1. Create and configure sensitivity labels and their policies. See Create sensitivity labels for more information.
2. Apply a sensitivity label to content. See Apply sensitivity labels to your files and email for more information.

Example 1: Verifying that a user's attempt to post a file matching the remote MPIP label through HTTPS will be blocked

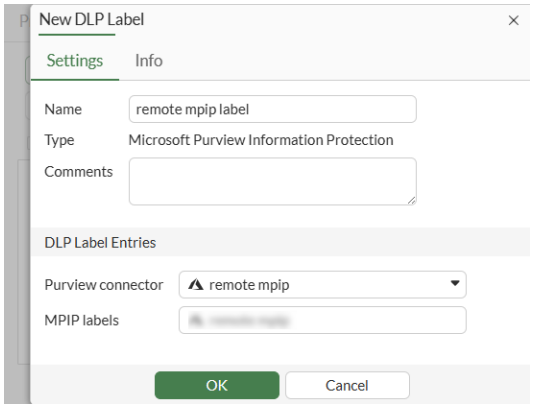
The following example demonstrates configuring a remote MPIP label to block HTTPS-POST actions.

To configure remote MPIP labels in the GUI:

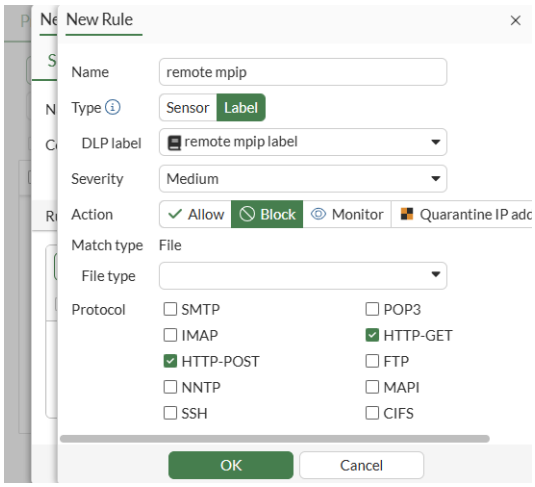
1. Configure the external connector:
   a. Go to *Security Fabric* > *External Connectors*.
   b. Click *Create New*.
   c. Configure a *Microsoft Azure* connector with *Microsoft 365* enabled. See Azure SDN connector using service principle for more information.



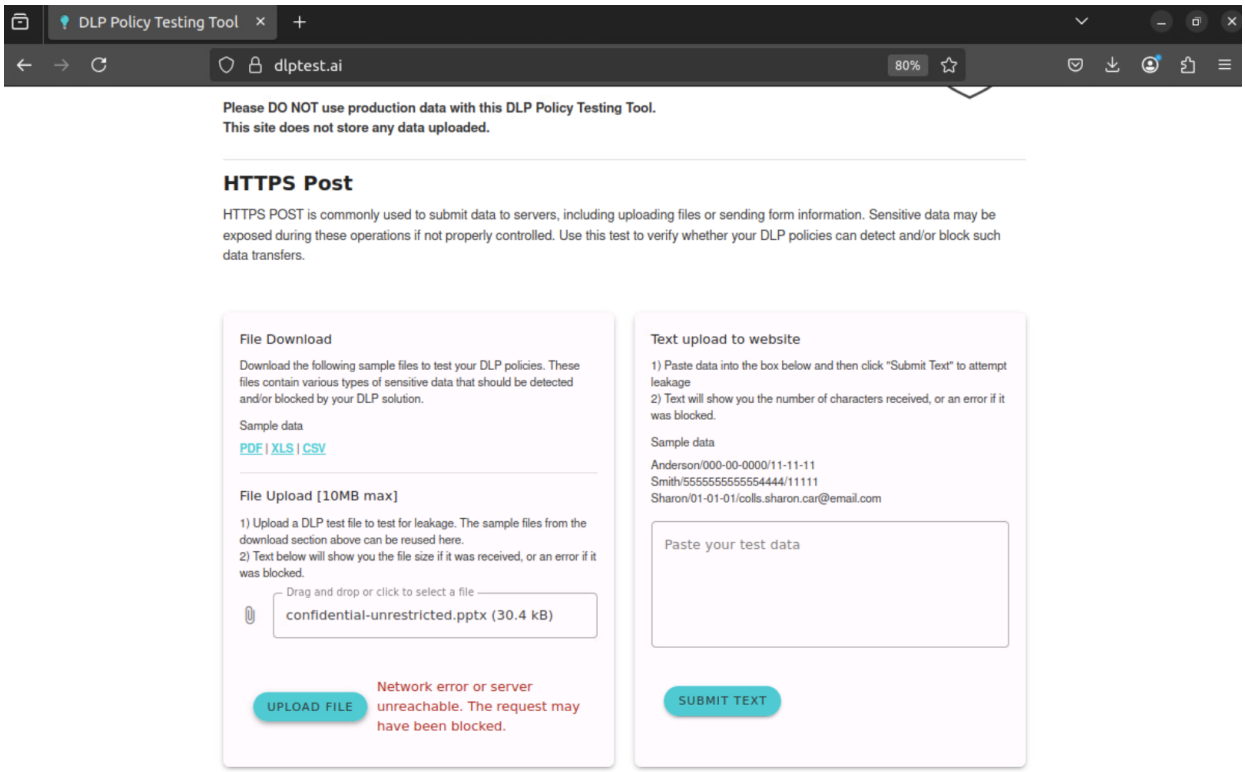
2. Configure the DLP label:
   a. Go to *Security Profiles* > *Data Loss Prevention*.
   b. Select the *Label* tab.
   c. Click *Create New*.
   d. Set *Purview connector* to the new connector.
   e. Select the *MPIP labels* synced from the connector to match sensitive files.
   f. Click *OK*.



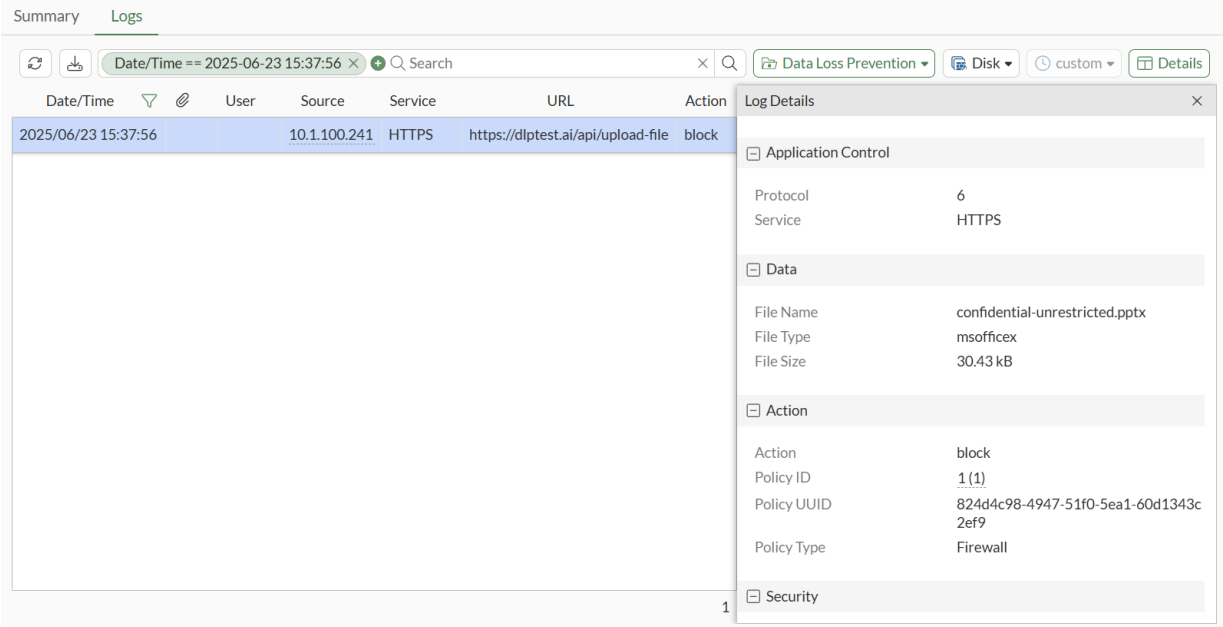
3. Configure the DLP profile:
   a. Go to *Security Profiles* > *Data Loss Prevention*.
   b. Select the *Profile* tab.
   c. Click *Create New*.
   d. Configure the DLP profile with the new DLP label applied in a rule set to block *HTTP-POST* and *HTTP-GET*. See Basic DLP settings for more information.



4. Add the DLP profile to a firewall policy:
   a. Go to *Policy & Objects* > *Firewall Policy*.
   b. Click *Create New*.
   c. Set the *Inspection Mode* to *Proxy-based*.
   d. In the *Security Profiles* section, enable *DLP Profile* and select the new DLP profile.
   e. Set *SSL Inspection* to *deep-inspection*.
   f. Configure the other settings as needed.
   g. Click *OK*.
5. Attempt an HTTPS-POST action.



The action matches the remote MPIP label and is blocked by the FortiGate.



To configure remote MPIP labels in the CLI:

1. Configure the external connector with Microsoft 365 enabled:

```
config system sdn-connector
    edit remote-mpip
        set type azure
        set azure-region global
        set tenant-id eca3011d-89ec-481d-9ed0-a0f7d2bb9b45
        set client-id 44ac574f-227a-4e90-a33b-820f2d7230b2
        set client-secret XXXXXXXX
        set update-interval 30
        set microsoft-365 enable
    next
end
```

2. Configure the DLP label:

```
config dlp label
    edit "remote-mpip-label"
        set type mpip
        set mpip-type remote
        set connector "remote-mpip"
        config entries
            edit 1
                set mpip-label-name "Confidential/Anyone (unrestricted)"
            next
        end
    next
end
```

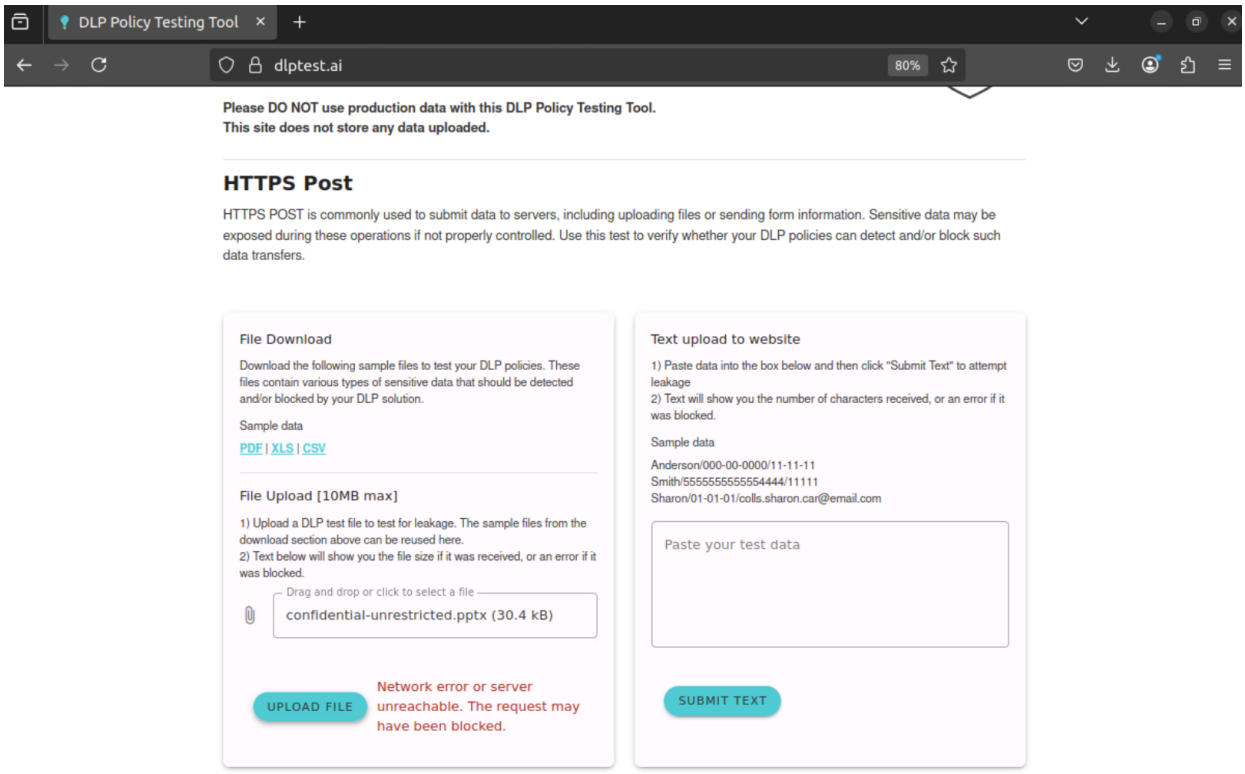
3. Configure the DLP profile to block HTTP-POST and HTTP-GET:

```
config dlp profile
    edit "remote-mpip-profile"
        config rule
            edit 1
                set proto http-get http-post
                set filter-by label
                set label "remote-mpip-label"
                set action block
            next
        end
    next
end
```

4. Add the DLP profile to a firewall policy:

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "deep-inpsection"
        set dlp-profile "remote-mpip-profile"
        set nat enable
    next
end
```

5. Attempt an HTTPS-POST action.



The action matches the remote MPIP label and is blocked by the FortiGate:

```
1: date=2025-06-23 time=15:36:53 eventtime=1750718213619030708 tz="-0700" logid="0954024576"
type="utm" subtype="dlp" eventtype="dlp" level="warning" vd="vdom1" ruleid=1 dlpextra="remote-
mpip-label" filtertype="label" filtercat="file" severity="medium" policyid=1
poluuid="824d4c98-4947-51f0-5ea1-60d1343c2ef9" policytype="policy" sessionid=720
epoch=1656654802 eventid=1 srcip=10.1.100.241 srcport=41346 srccountry="Reserved"
srcintf="port2" srcintfrole="undefined" srcuuid="4a1017e8-4947-51f0-55cb-8c18f3765a3f"
dstip=206.47.132.132 dstport=443 dstcountry="Canada" dstintf="port1" dstintfrole="undefined"
dstuuid="4a1017e8-4947-51f0-55cb-8c18f3765a3f" proto=6 service="HTTPS" filetype="msofficex"
direction="outgoing" action="block" hostname="dlptest.ai" url="https://dlptest.ai/api/upload-
file" agent="Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:139.0) Gecko/20100101 Firefox/139.0"
httpmethod="POST" referralurl="https://dlptest.ai/" filename="confidential-unrestricted.pptx"
filesize=30427 profile="remote-mpip-profile" infectedfilename="docProps/custom.xml"
infectedfilesize=2025 infectedfilelevel=1
```

Example 2: Verifying that a user's attempt to post a file matching the local MPIP label's GUID will be monitored through FTPS

The following example demonstrates configuring a local MPIP label with a Globally Unique Identifier (GUID) that will monitor files posted through FTPS.

Local MPIP label configuration is not supported in the GUI. However, once the local MPIP label has been configured in the CLI, it can be applied to a DLP profile in the GUI. See Basic DLP settings on page 2606.

To configure and apply a local MPIP label:

1. Obtain a GUID for your sensitivity labels. See Search for documents by sensitivity label for more information.
2. Configure the local MPIP label:

```
config dlp label
    edit "local-mpip-label"
        set type mpip
        set mpip-type local
        config entries
            edit 1
                set guid "c0ef495a-6714-4056-a44b-a9cb63fa9161"
            next
        end
    next
end
```

3. Configure the DLP profile:

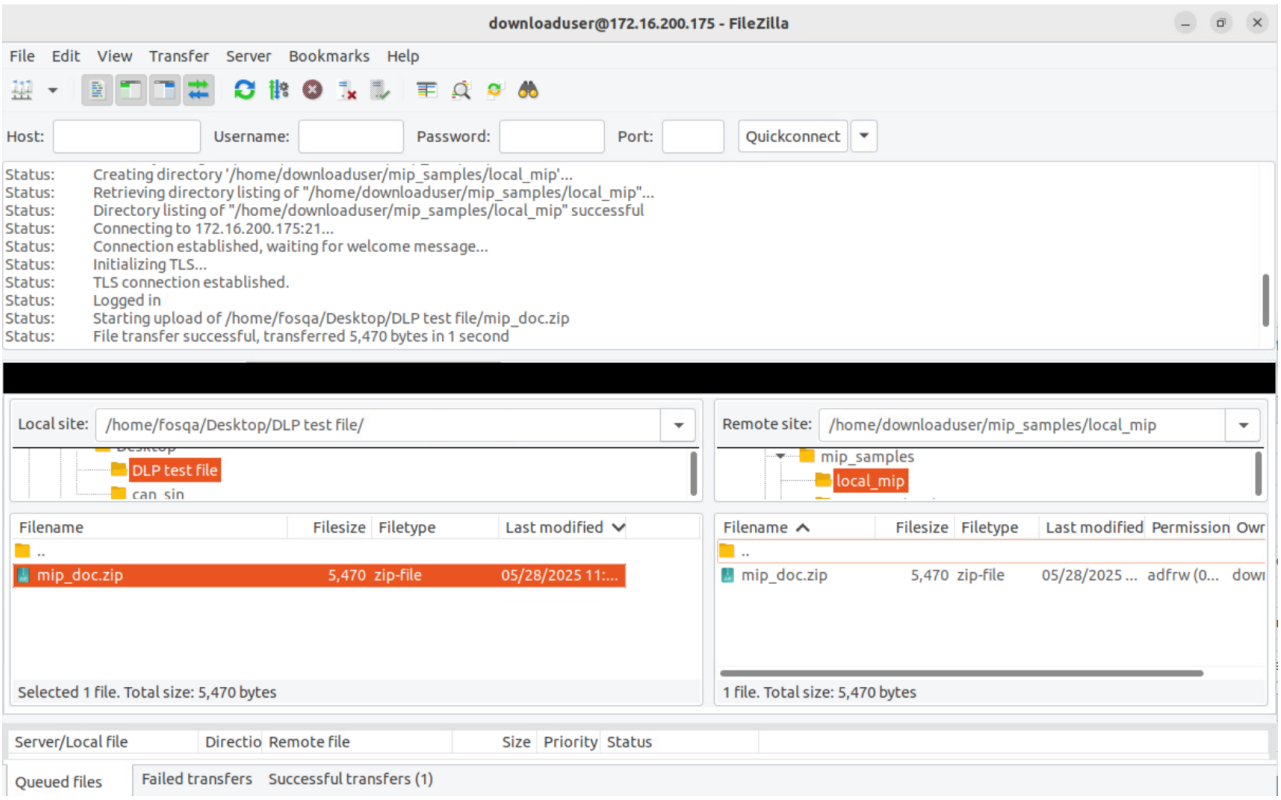
```
config dlp profile
   edit "local-mpip-profile"
        config rule
            edit 1
                set proto ftp
                set filter-by label
                set label "local-mpip-label"
                set action log-only
            next
        end
    next
end
```

4. Add the DLP profile to a firewall policy:

```
config firewall policy
    edit 2
        set srcintf "port2"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "deep-inpsection"
```

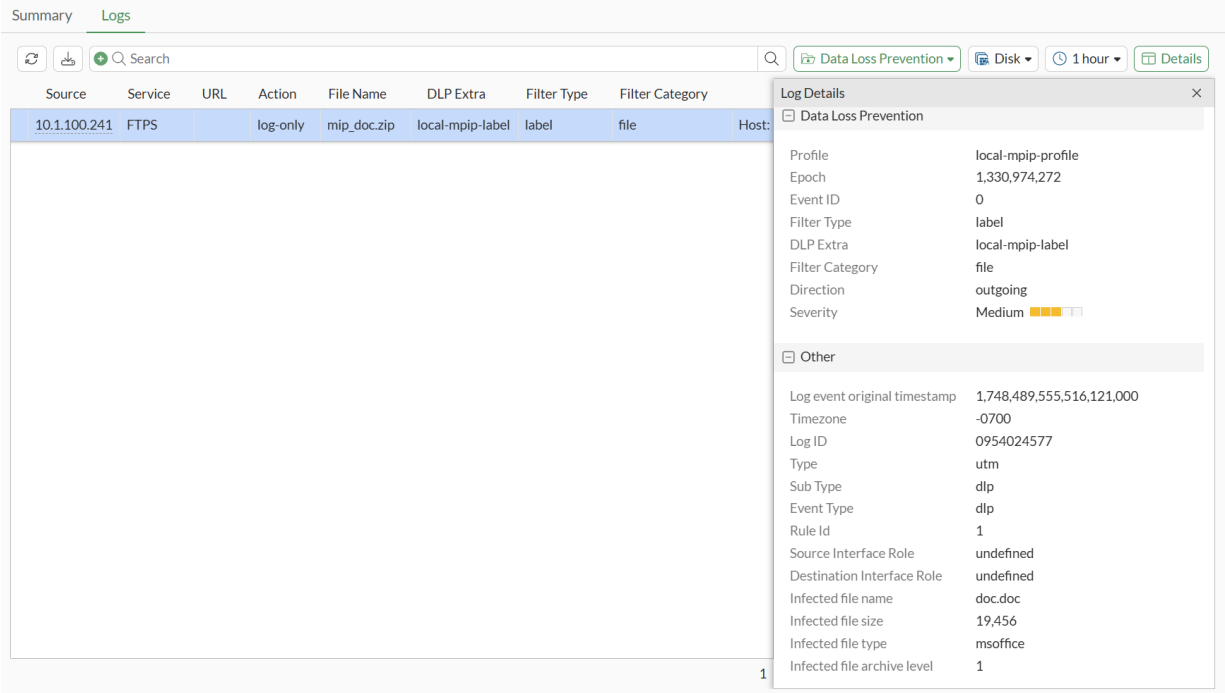
```
        set dlp-profile "local-mpip-profile"
        set nat enable
    next
end
```

5. Attempt to transfer a file matching the remote MPIP label through FTPS.



The file passes and DLP monitor logs are generated:

- A GUI log can be reviewed:



- A CLI log can be reviewed:

```
1: date=2025-05-28 time=20:32:35 eventtime=1748489555516121022 tz="-0700"
logid="0954024577" type="utm" subtype="dlp" eventtype="dlp" level="notice" vd="vdom1"
ruleid=1 dlpextra="local-mpip-label" filtertype="label" filtercat="file" severity="medium"
policyid=1 poluuid="319d6e9e-3c23-51f0-3a27-434de0e559ee" policytype="policy"
sessionid=8379 epoch=1330974272 eventid=0 srcip=10.1.100.241 srcport=57865
srccountry="Reserved" srcintf="port2" srcintfrole="undefined" srcuuid="f9cd0f6a-3c22-51f0-
b75c-6fad5f412db5" dstip=172.16.200.175 dstport=38353 dstcountry="Reserved"
dstintf="port1" dstintfrole="undefined" dstuuid="f9cd0f6a-3c22-51f0-b75c-6fad5f412db5"
proto=6 service="FTPS" filetype="zip" direction="outgoing" action="log-only"
filename="mip_doc.zip" filesize=5470 profile="local-mpip-profile"
infectedfilename="doc.doc" infectedfilesize=19456 infectedfiletype="msoffice"
infectedfilelevel=1
```

Exact data matching

Exact Data Matching (EDM) identifies particular data values within an indexed data source that require safeguarding. It offer precise detection and handling of sensitive data based on user-defined criteria. This enhances security and improves efficiency by reducing false-positive detections.

Administrators can define a dataset in a CSV or TXT file on a server, upload file directly on FortiGate or point FortiGate to the external resource. See EDM template on page 2632 for more information.

CSV file example:

Jason	Valentino	120 Jefferson St.	Riverside	CA	92504

Marry	Baxter	415 W Willow Grove Ave	Philadelphia	PA	19118
David	Solace	555 Pierce Street APT #123	Albany	CA	94706
Thomas	Jefferson	1600 Pennsylvania Avenue NW	Washington	DC	20500

TXT file example:

```
213321,john,doe
201111,karen,smith
322122,rick,wong
```

A CSV or TXT file can have a maximum of 32 columns. Each indexed column in the external file represents data (or patterns) for a built-in data type that you want to match using EDM template.

EDM template

The EDM template is used to specify the URL location of the data external feed file or upload the file directly on to the FortiGate. Once the data is imported, it can be utilized with an EDM template to map individual columns of data (or patterns) from a file to built-in data types to match credit card, keyword, mip label, social insurance number (SIN), and social security number (SSN) data. When the data passing through FortiGate matches with the EDM template, FortiGate responds according to the preset rules.

A CSV or TXT file can be uploaded directly to FortiGate using the File Upload option. However, this option is exclusively available through the GUI. It's important to note that file upload is only possible if your FortiGate unit is equipped with a hard disk. In the absence of a hard disk, the File Upload option will be grayed out. See Feature Platform Matrix.

Please note that if the CSV or TXT file is uploaded using the File Upload option, it will not be dynamically synchronized nor periodically updated. This means that any changes made to the file will not be imported by FortiOS. Therefore, users are required to manually update the file again on the FortiGate using the Update file option. This option is located under Resource Type and is only visible if the file was previously uploaded via the File Upload option.

The sequence of steps for configuring EDM is consistent with other DLP configurations. See Basic DLP settings on page 2606 for more information.

Data external feed

A data external feed is a dynamic list that contains data. The data is patterns for DLP data types. The dynamic list is stored in a text (TXT) or comma-separated value (CSV) file format on an external server and periodically updated. After FortiGate can access the file, the patterns can be used with an EDM template.

```
config system external-resource
    edit <name>
        set type data
        set resource <string>
        set refresh-rate <integer>
```

```
    next
end
```

Command	Description
`type {category | domain | malware | address | mac-address | data}`	Specify the type of user resource. • `data`: Specify a data file as the user source.
`resource <string>`	Specify the URL of the external resource.
`refresh-rate <integer>`	Time interval to refresh the external resource (minutes).

The size of the Data External Feed file varies depending on the device model. The maximum file size limit for each model is as follows:

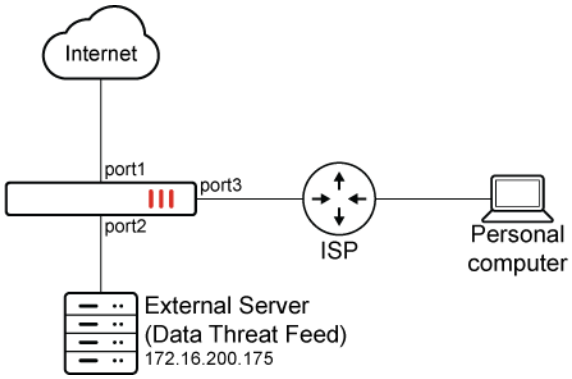
- 128 MB for High-End (Data Center) models
- 64 MB for Mid-Range (Campus) models
- 32 MB for Entry-Level (Branch) models.

Example

This configuration will block HTTPS upload traffic that matches the DLP profile.

When utilizing commonly-used SSL-encrypted protocols such as HTTPS, SMTPS, POP3S, IMAPS, and FTPS, SSL inspection must be set to Deep Inspection. See Deep inspection for more information.

Additionally, the client machine must have the corresponding deep inspection Certificate Authority (CA) certificate installed.



In this example, an EDM template named Customer SSN EDM is created on the FortiGate. During this process, a CSV file (customer_data.csv) located on an external server is imported using the data thread feed.

Sample CSV file:

SSN	Last Name	First Name	Address	City	State	ZIP	Phone	Email	CCN
172-32-1176	Doe	John	10932 Big Rd	Malibu	CA	94025	408-497-7223	jdoe@domain.com	5270-4267-6450-5516
514-14-8905	Bard	Ashley	4469 Sher St	Golf	KS	66428	785-939-6046	abard@domain.com	5370-4638-8881-302

In this example, the EDM template specifies:

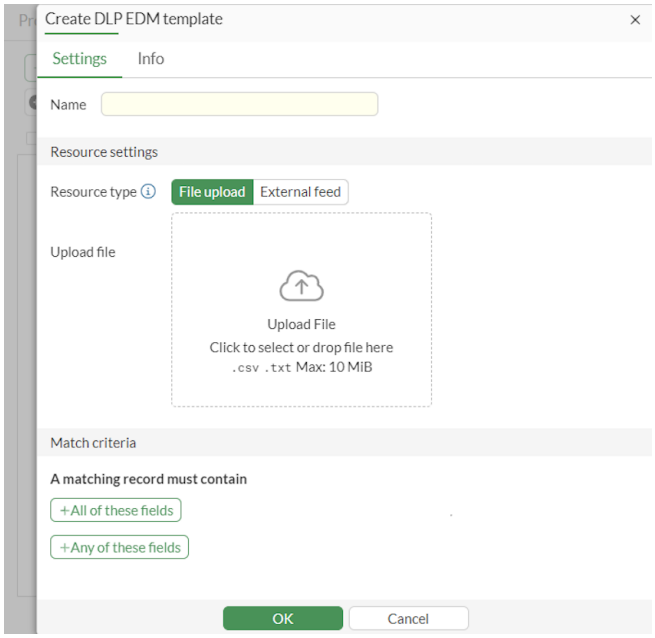
- Column index 1 in the external data feed file contains patterns for the `ssn-us` data type.
- Column index 3 and 9 contain patterns for the `edm-keyword` data type.
- The patterns from column index 1 must match for FortiGate to take an action.
- The pattern from either column index 3 or 9 must match for FortiGate to take an action.

Based on the aforementioned template, the DLP profile will match any traffic containing data that corresponds to the SSN in column 1, and either the First Name in column 3 or the Email in column 9. For instance, if the HTTPS upload traffic sent from personal computer contains '172-32-1176' AND 'John', or '172-32-1176' AND 'jdoe@domain.com', the traffic will be blocked and a DLP log is generated. See Sample log for a log sample.

To configure EDM for DLP in the GUI:

1. Ensure that *Data Loss Prevention* is enabled.
   a. Go to *System* > *Feature Visibility*.
   b. Under *Security Features*, enable *Data Loss Prevention*, and click *Apply*.
2. Create an EDM template with matching criteria:

a. Go to *Security Profiles* > *Data Loss Prevention* > *EDM Template*, and click *Create New*.



b. Specify a name for the template, such as *Customer SSN EDM*.

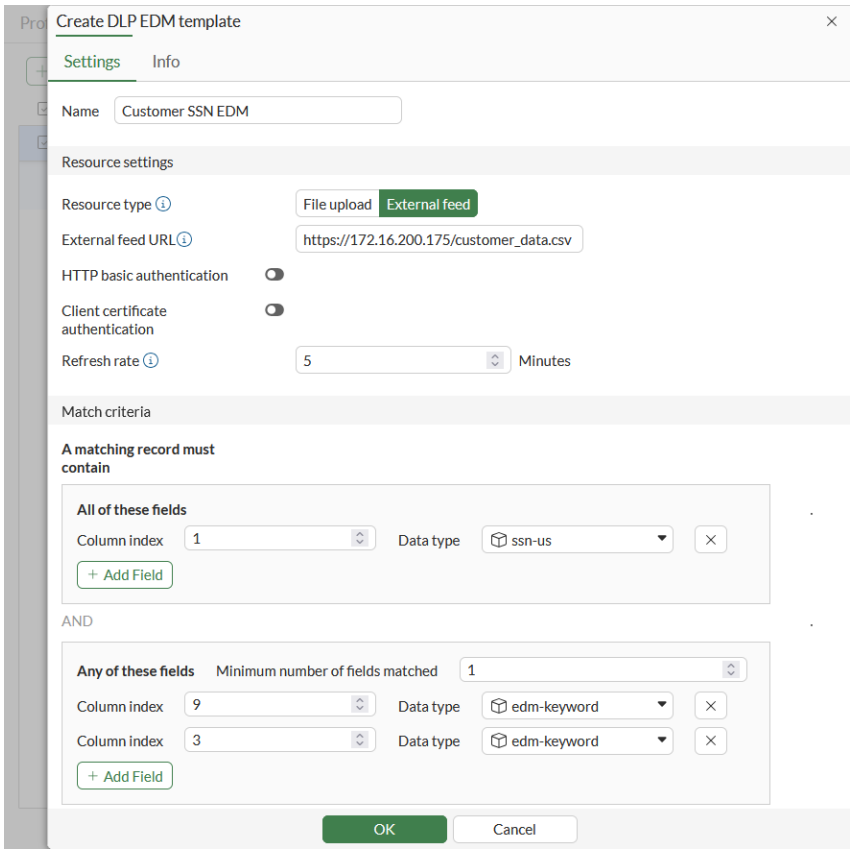
c. Set *Resource type* to *External feed*, and set *External feed URL* to the location of the file on the external server, such as *https://172.16.200.175/customer_data.csv*.

d. Click *+All of these fields* to pair the column index of patterns with a DLP data type. All of the specified data in this section must match for FortiGate to take an action.

In this example, column 1 in the external resource file contains the patterns for the *ssn-us* data type.

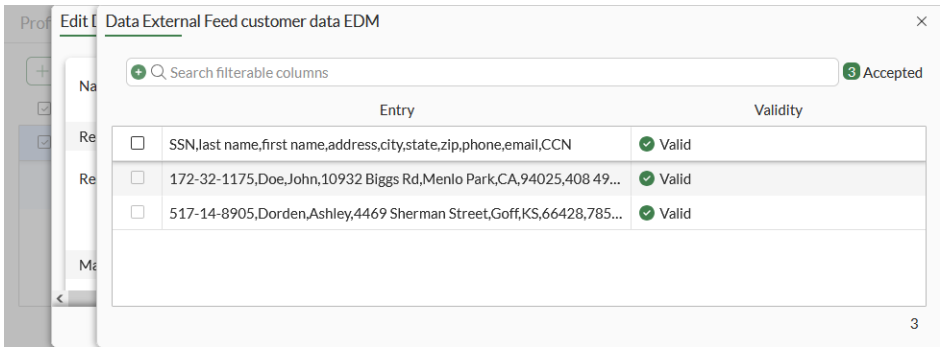
e. Click *+Any of these fields* to pair the column index of patterns with a DLP data type, and to specify how many of these pairs must match for FortiGate to take an action.

In this example, columns 3 and 9 in the external resource file contains the patterns for the *edm-keyword* data type. Only one pattern from the two columns must match.



f. Click *OK*.

g. Edit the DLP EDM template and click *View Entries* to view the data entries in the field.



When viewing EDM entries, the GUI currently does not validate the entries and displays all entries as Valid. However, a Valid entry must have all values matching the data-type of the specific column. If one value does not match, the entry is invalid and will not be used for pattern matching.

3. Configure a DLP sensor for the EDM template.

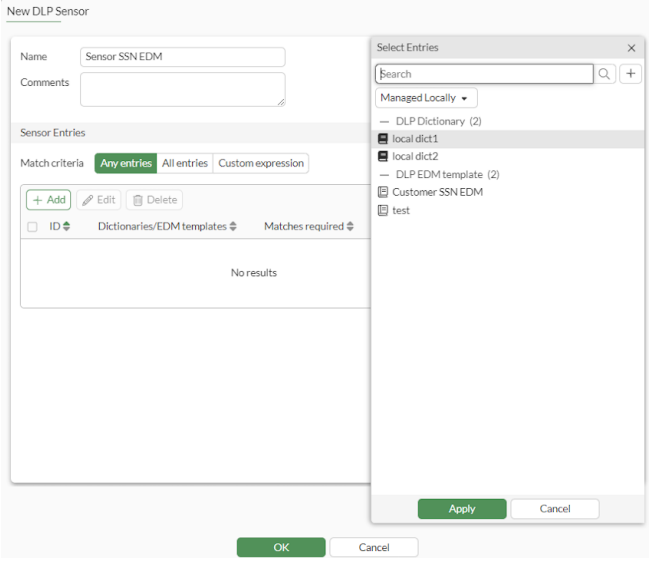
a. In *Security Profiles* > *Data Loss Prevention*, click *Sensor* > *Create New*.

b. Specify a name for the DLP sensor, such as *Sensor SSN EDM*.

c. Click *Add*. The *Select Entries* pane is displayed.

d. Select *Managed Locally* from the dropdown menu.

e. From the list, select *Customer SSN EDM*, and click *Apply*.



f. Click *OK*.

4. Create a DLP profile and select the DLP sensor for the EDM template.
   a. In *Security Profiles* > *Data Loss Prevention*, click *Profile* > *Create New*.
   b. Specify a name for the DLP profile, such as *Profile SSN EDM*.
   c. Click *Create New*. The *New Rule* pane is displayed.
   d. Specify a name for the rule, such as *Rule SSN EDM*.
   e. Set *Data source type* to *Sensor*, and select *Sensor SSN EDM*.
   f. Set *Action* to *Block*.
   g. Set *Match type* to *Message*.
   h. Select *HTTP-POST* protocol.
   i. Click *OK*. The *New DLP Profile* pane is displayed.
   j. Click *OK* to save the profile.
5. Add the DLP profile to a firewall policy:
   a. Go to *Policy & Objects* > *Firewall Policy*.
   b. Click *Create New*.
   c. Set the *Inspection Mode* to *Proxy-based*.
   d. In the *Security Profiles* section, enable *DLP Profile* and select *Profile SSN EDM*.
   e. Set *SSL Inspection* to *deep-inspection*.
   f. Configure the other settings as needed.
   g. Click *OK*.

To configure EDM for DLP in the CLI:

1. Add the URL for the data external feed file to FortiGate.

   In this example, an external resource named `customer data EDM` is created, and it defines the location of the data external feed file in CSV format on an external server.

```
config system external-resource
    edit "customer data EDM"
        set type data
        set resource "https://172.16.200.175/customer_data.csv"
    next
end
```

2. Configure the EDM template.

   In this example, an exact data-match template named `Customer SSN EDM` is created for the external resource named `customer data EDM`. The matching record must contain the pattern for the data type from column index 1 (`ssn-us`) and at least one pattern for the data type from column index 3 (`edm-keyword`) or 9 (`edm-keyword`).

```
config dlp exact-data-match
    edit "Customer SSN EDM"
        set optional 1
        set data "customer data EDM"
        config columns
            edit 1
                set type "ssn-us"
            next
            edit 3
                set type "edm-keyword"
                set optional enable
            next
            edit 9
                set type "edm-keyword"
                set optional enable
            next
        end
    next
end
```

3. Add the EDM template to a DLP sensor.

```
config dlp sensor
    edit "Sensor SSN EDM"
        config entries
            edit 1
                set dictionary "Customer SSN EDM"
            next
        end
    next
end
```

4. Configure a DLP profile to use the DLP sensor.

```
config dlp profile
    edit "Profile SSN EDM"
        set feature-set proxy
        config rule
            edit 1
```

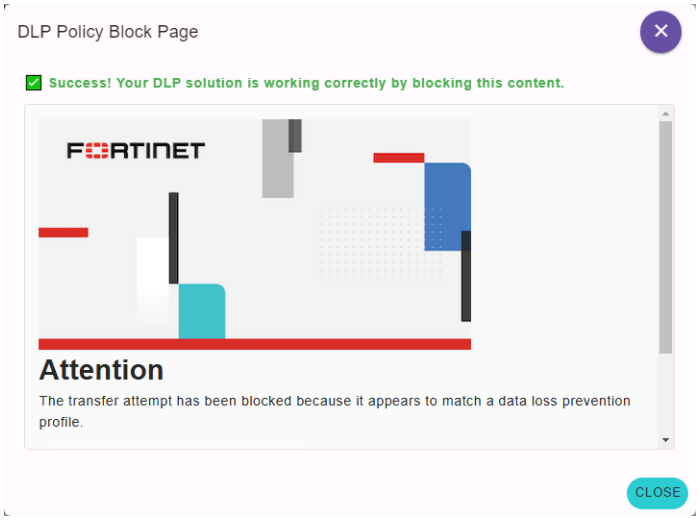
```
                set name "Rule SSN EDM"
                set type message
                set proto http-post
                set filter-by sensor
                set sensor "Sensor SSN EDM"
                set action block
            next
        end
    next
end
```

5. Add the DLP profile to a firewall policy.

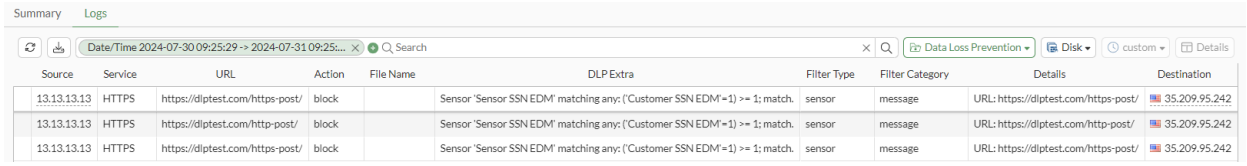
```
config firewall policy
    edit 1
        set name "Internet"
        set srcintf "port3"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "deep-inspection"
        set dlp-profile "Profile SSN EDM"
    next
end
```

To verify:

1. A user attempts to post a sensitive message through HTTPS to dlptest.ai, and the message content matches the EDM template.
2. FortiGate blocks the user's attempt and displays a replacement message:



3. FortiGate generates a DLP log:




```
1: date=2024-07-31 time=09:24:44 eventtime=1722443083953870602 tz="-0700" logid="0954024576"
type="utm" subtype="dlp" eventtype="dlp" level="warning" vd="root" ruleid=1 rulename="Rule SSN
EDM" dlpextra="Sensor 'Sensor SSN EDM' matching any: ('Customer SSN EDM'=1) >= 1; match."
filtertype="sensor" filtercat="message" severity="medium" policyid=1 poluuid="1696f98a-3413-
51ef-ed16-01791b5b8127" policytype="policy" sessionid=18087 transid=1 epoch=959760595
eventid=1 srcip=13.13.13.13 srcport=62595 srccountry="United States" srcintf="port3"
srcintfrole="undefined" srcuuid="93edbd62-33d5-51ef-2497-b193994b4d2e" dstip=35.209.95.242
dstport=443 dstcountry="United States" dstintf="port1" dstintfrole="undefined"
dstuuid="93edbd62-33d5-51ef-2497-b193994b4d2e" proto=6 service="HTTPS" filetype="N/A"
direction="outgoing" action="block" hostname="dlptest.ai" url="https://dlptest.ai/https-post/"
agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/126.0.0.0 Safari/537.36" httpmethod="POST" referralurl="https://dlptest.ai/https-post/"
profile="Profile SSN EDM"
```

DLP examples

The following topics provide examples of DLP:

- Block HTTPS upload traffic that includes credit card information on page 2640
- Log FTP upload traffic with a specific pattern on page 2644
- Block HTTPS downloads of EXE files and log HTTPS downloads of files larger than 500 KB on page 2648
- Block HTTPS upload traffic that includes Visa or Mastercard information using evaluation through logical expression on page 2650
- Proximity search on page 2654
- Assign confidence levels in FortiGuard managed DLP dictionaries on page 2658
- FortiData labels on page 2662
- Prevent sensitive token leakage in HTTPS uploads using EDM on page 2670

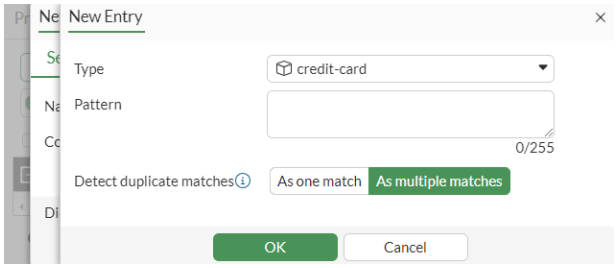
Block HTTPS upload traffic that includes credit card information

This configuration will block HTTPS upload traffic that includes credit card information. The predefined data type for credit card is used in the dictionary.

To block HTTPS upload traffic that includes credit card information in the GUI:

1. Configure the DLP dictionary:
   a. Go to *Security Profiles* > *Data Loss Prevention*, select the *Dictionary* tab, and click *Create New*.
   b. Enter a name (*dic-case1*).
   c. In the *Dictionary Entries* section, click *Create New*.

d. Set the *Type* to *credit-card* and click *OK*.



e. Click *OK*.

f. Click *OK* to save the dictionary.

2. Configure the DLP sensor:

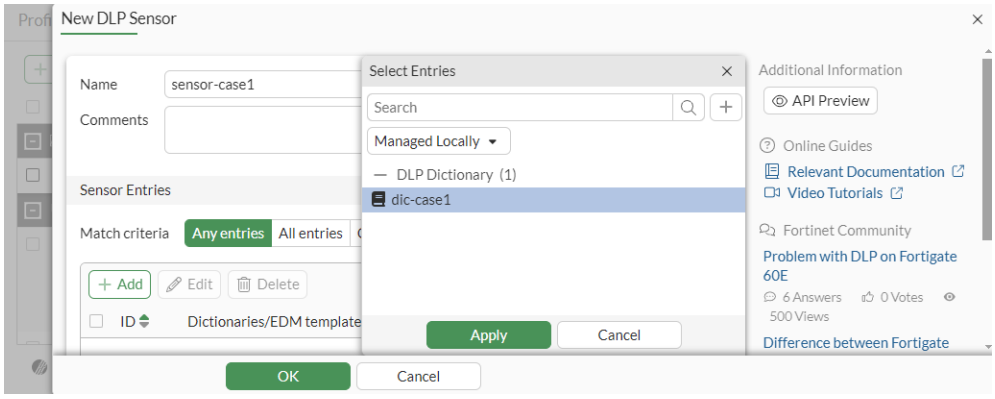
a. Go to *Security Profiles* > *Data Loss Prevention*, select the *Sensor* tab, and click *Create New*.

b. Enter a name (*sensor-case1*).

c. In the *Sensor Entries* section, click *Add*. The *Select Entries* pane is displayed.

d. Select *Managed Locally* from the dropdown menu.

e. Select *dic-case1* and click *Apply*.



f. Click *Apply*.

g. Click *OK* to save the sensor.

3. Configure the DLP profile:

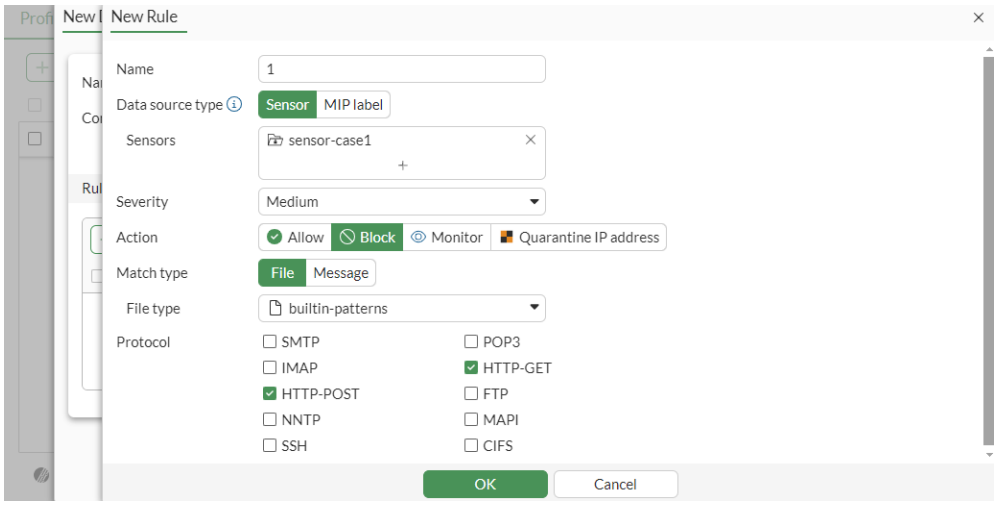
a. Go to *Security Profiles* > *Data Loss Prevention*, select the *Profile* tab, and click *Create New*.

b. Enter a name (*profile-case1*).

c. In the *Rules* section, click *Create New*.

d. Configure the following settings:

Name	*1*
Sensors	*sensor-case1*
Severity	*Medium*
Action	*Block*
Type	*File*
File type	*builtin-patterns*
Protocol	*HTTP-POST*, *HTTP-GET*



e. Click *OK*.

f. Click *OK* to save the profile.

4. Add the DLP profile to a firewall policy:

a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.

b. Set the *Inspection Mode* to *Proxy-based*.

c. In the *Security Profiles* section, enable *DLP Profile* and select *profile-case1*.

d. Configure the other settings as needed.

e. Click *OK*.

When a credit card is included in HTTP POST traffic, the file is blocked and a DLP log is generated.

To block HTTPS upload traffic that includes credit card information in the CLI:

1. Configure the DLP dictionary:

```
config dlp dictionary
    edit "dic-case1"
        config entries
            edit 1
                set type "credit-card"
            next
        end
    next
end
```

2. Configure the DLP sensor:

```
config dlp sensor
    edit "sensor-case1"
        config entries
            edit 1
                set dictionary "dic-case1"
            next
        end
```

```
    next
end
```

3. Configure the DLP profile:

```
config dlp profile
    edit "profile-case1"
        set feature-set proxy
        config rule
            edit 1
                set name "1"
                set proto http-get http-post
                set filter-by sensor
                set file-type 1
                set sensor "sensor-case1"
                set action block
            next
        end
    next
end
```

4. Add the DLP profile to a firewall policy:

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set profile-protocol-options "protocol"
        set ssl-ssh-profile "protocols"
        set dlp-profile "profile-case1"
        set nat enable
    next
end
```

When a credit card is included in HTTP POST traffic, a replacement message appears because it is blocked. A DLP log is generated.

Sample log

From Windows, the following command below can be used to generate a sample log, using the cURL tool to post data, which contains a sample credit card number. See sample-data for sample credit card numbers.

```
# curl –k -d 4024007149133315 https://172.16.200.55/card.doc -o?
```

```
1: date=2022-10-26 time=11:25:01 eventtime=1666808700281057923 tz="-0700" logid="0954024576"
type="utm" subtype="dlp" eventtype="dlp" level="warning" vd="root" ruleid=1 rulename="1"
dlpextra="builtin-patterns;sensor-case1" filtertype="sensor" filtercat="file" severity="medium"
policyid=1 poluuid="891a526a-51cd-51ed-577a-6505bec88af9" policytype="policy" sessionid=3905
epoch=2143297701 eventid=0 srcip=10.1.100.11 srcport=40370 srccountry="Reserved" srcintf="port2"
srcintfrole="undefined" srcuuid="502d2c8e-51cd-51ed-a24e-a091f4ff6fed" dstip=172.16.200.55
dstport=443 dstcountry="Reserved" dstintf="port1" dstintfrole="undefined" dstuuid="502d2c8e-51cd-
51ed-a24e-a091f4ff6fed" proto=6 service="HTTPS" filetype="msoffice" direction="outgoing"
action="block" hostname="172.16.200.55" url="https://172.16.200.55/card.doc" agent="curl/7.83.1"
httpmethod="POST" filename="card.doc" filesize=108 profile="profile-case1"
```

Log FTP upload traffic with a specific pattern

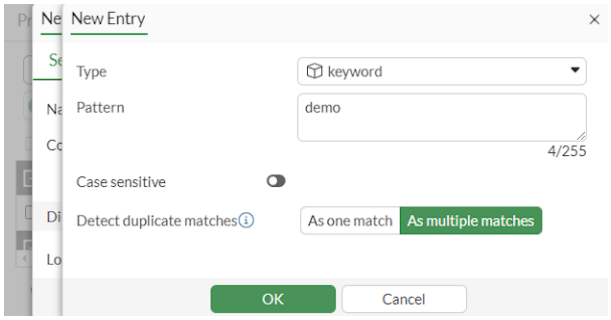
This configuration will log FTP upload traffic with the following patterns:

- keyword = demo
- regex = demo(regex){1,5}
- hex = e6b58be8af95

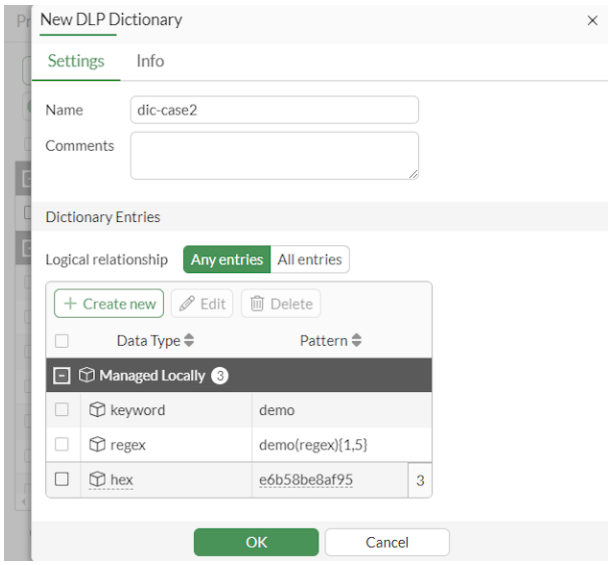
The dictionary entries have repeat match enabled. The DLP sensor is set so this is repeated five times.

To log FTP upload traffic that has specific keyword, regex, and hex patterns repeated for five times in the GUI:

1. Configure the DLP dictionary with three entries:
   a. Go to *Security Profiles* > *Data Loss Prevention*, select the *Dictionary* tab, and click *Create New*.
   b. Enter a name (*dic-case2*).
   c. In the *Dictionary Entries* section, click *Create New*.
   d. Set the *Type* to *keyword* and the *Pattern* to *demo*.
   e. Set *Detect duplicate matches* to *As multiple matches*.

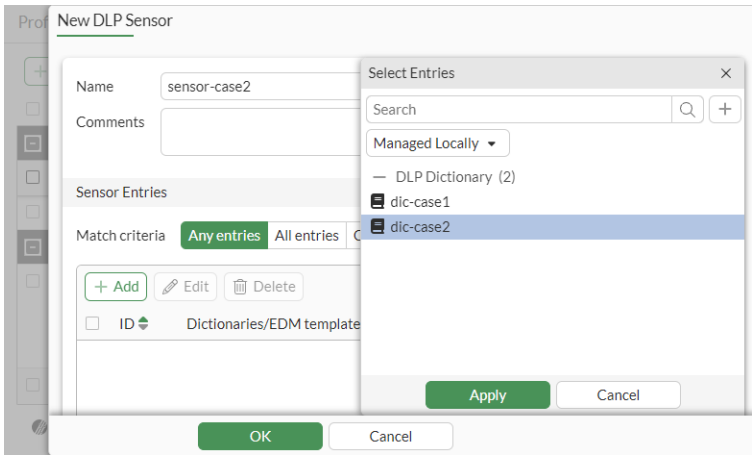


   f. Click *OK*.
   g. Repeat these steps to add dictionary entries for the following (with *Repeats* enabled):
      i. Set the *Type* to *regex* and the *Pattern* to *demo(regex){1,5}*.
      ii. Set the *Type* to *hex* and the *Pattern* to *e6b58be8af95*.



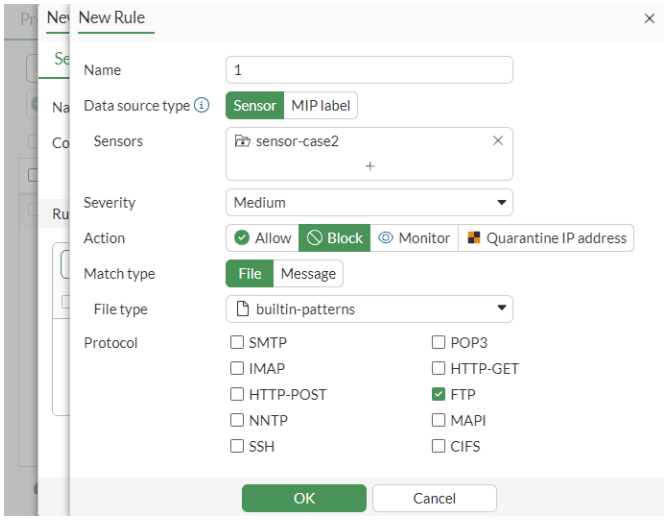
h. Click *OK* to save the dictionary.

2. Configure the DLP sensor:
   a. Go to *Security Profiles* > *Data Loss Prevention*, select the *Sensor* tab, and click *Create New*.
   b. Enter a name (*sensor-case2*).
   c. In the *Sensor Entries* section, click *Add*. The *Select Entries* pane is displayed.
   d. Select *Managed Locally* from the dropdown menu.
   e. Select *dic-case2* and click *Apply*.



   f. Edit the new entry and set the *Number of matches required* to *5*.
   g. Click *OK*.
   h. Click *OK* to save the sensor.
3. Configure the DLP profile:
   a. Go to *Security Profiles* > *Data Loss Prevention*, select the *Profile* tab, and click *Create New*.
   b. Enter a name (*profile-case2*).
   c. In the *Rules* section, click *Create New*.
   d. Configure the following settings:

Name	*1*
Sensors	*sensor-case2*
Severity	*Medium*
Action	*Block*
Type	*File*
File type	*builtin-patterns*
Protocol	*FTP*



e. Click *OK*.
f. Click *OK* to save the profile.

4. Add the DLP profile to a firewall policy:
   a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
   b. Set the *Inspection Mode* to *Proxy-based*.
   c. In the *Security Profiles* section, enable *DLP Profile* and select *profile-case2*.
   d. Configure the other settings as needed.
   e. Click *OK*.
5. Upload a Word document that contains "demo, demo, demo, demoregexregex," using FTP.
   A DLP log is generated after the FTP traffic passes.

To log FTP upload traffic that has specific keyword, regex, and hex patterns repeated for five times in the CLI:

1. Configure the DLP dictionary:

```
config dlp dictionary
    edit "dic-case2"
        config entries
            edit 1
                set type "keyword"
```

```
                set pattern "demo"
                set repeat enable
            next
            edit 2
                set type "regex"
                set pattern "demo(regex){1,5}"
                set repeat enable
            next
            edit 3
                set type "hex"
                set pattern "e6b58be8af95"
                set repeat enable
            next
        end
    next
end
```

2. Configure the DLP sensor:

```
config dlp sensor
    edit "sensor-case2"
        config entries
            edit 1
                set dictionary "dic-case2"
                set count 5
            next
        end
    next
end
```

3. Configure the DLP profile:

```
config dlp profile
    edit "profile-case2"
        set feature-set proxy
        config rule
            edit 1
                set proto ftp
                set filter-by sensor
                set file-type 1
                set sensor "sensor-case2"
                set action block
            next
        end
    next
end
```

4. Add the DLP profile to a firewall policy:

```
config firewall policy
    edit 1
        set srcintf "port2"
```

```
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set profile-protocol-options "protocol"
        set ssl-ssh-profile "protocols"
        set dlp-profile "profile-case2"
        set nat enable
    next
end
```

5. Upload a Word document that contains "demo, demo, demo, demoregexregex," using FTP.
   A DLP log is generated after the FTP traffic passes.

Sample log

```
1: date=2022-10-26 time=12:37:57 eventtime=1666813077679725858 tz="-0700" logid="0954024576"
type="utm" subtype="dlp" eventtype="dlp" level="warning" vd="root" ruleid=1 rulename="1"
dlpextra="builtin-patterns;sensor-case2" filtertype="sensor" filtercat="file" severity="medium"
policyid=1 poluuid="891a526a-51cd-51ed-577a-6505bec88af9" policytype="policy" sessionid=6267
epoch=909159520 eventid=0 srcip=10.1.100.11 srcport=52858 srccountry="Reserved" srcintf="port2"
srcintfrole="undefined" srcuuid="502d2c8e-51cd-51ed-a24e-a091f4ff6fed" dstip=172.16.200.55
dstport=43411 dstcountry="Reserved" dstintf="port1" dstintfrole="undefined" dstuuid="502d2c8e-
51cd-51ed-a24e-a091f4ff6fed" proto=6 service="FTP" filetype="msoffice" direction="outgoing"
action="block" filename="realizedDoc.doc" filesize=26624 profile="profile-case2"
```

Block HTTPS downloads of EXE files and log HTTPS downloads of files larger than 500 KB

FortiGate can detect any file larger than the configured limit. See File size filtering on page 2605 for more information.

This configuration will block HTTPS downloads of EXE files and log HTTPS downloads of files larger than 500 KB.

To block HTTPS download of EXE files and log downloads larger than 500 KB:

1. Configure the DLP file pattern:

```
config dlp filepattern
    edit 3
        set name "case3-exe"
        config entries
```

```
            edit "exe"
                set filter-type type
                set file-type exe
            next
        end
    next
end
```

2. Configure the DLP profile:

```
config dlp profile
    edit "profile-case3-type-size"
        config rule
            edit 1
                set proto http-get
                set filter-by none
                set file-type 3
                set action block
            next
            edit 2
                set proto http-get
                set filter-by none
                set file-size 500
                set action log-only
            next
        end
    next
end
```

3. Add the DLP profile to a firewall policy:

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "custom-deep-inspection"
        set dlp-profile "profile-case3-type-size"
        set logtraffic all
        set nat enable
    next
end
```

4. Download an EXE file using HTTPS. The download is blocked, a replacement message appears, and a DLP log is generated.

Sample log

```
1: date=2022-02-15 time=11:54:29 eventtime=1644954869682887856 tz="-0800" logid="0954024577"
type="utm" subtype="dlp" eventtype="dlp" level="notice" vd="root" ruleid=2 dlpextra="500 kB"
filtertype="none" filtercat="file" severity="medium" policyid=1 poluuid="905fb604-7ed4-51ec-0853-
79e498591bf8" policytype="policy" sessionid=12082 epoch=901683674 eventid=0 srcip=10.1.100.18
srcport=59520 srccountry="Reserved" srcintf="port2" srcintfrole="undefined" srcuuid="358d0f56-
7ed4-51ec-50f7-a5e4525a641d" dstip=51.81.186.201 dstport=443 dstcountry="United States"
dstintf="port1" dstintfrole="undefined" dstuuid="358d0f56-7ed4-51ec-50f7-a5e4525a641d" proto=6
service="HTTPS" direction="incoming" action="log-only" hostname="2.na.dl.wireshark.org"
url="https://2.na.dl.wireshark.org/win64/Wireshark-win64-3.6.2.exe" agent="curl/7.61.1"
filename="Wireshark-win64-3.6.2.exe" filesize=10502090 profile="profile-case3-type-size"
```

Block HTTPS upload traffic that includes Visa or Mastercard information using evaluation through logical expression

This example will allow users to create a subset of the existing DLP data type, *credit-card*. It can be very beneficial for an organization that wants to prevent only certain types of credit cards and not all.

This configuration will block HTTPS traffic that includes Visa or Mastercard information. Two dictionary entries with DLP data-type regex are created with custom patterns to match Visa and Mastercard numbers respectively, and a third dictionary entry is created with predefined data type *credit-card*. All three entries are used in the sensor using evaluation via logical expression to further supplement the detection.

In the CLI, evaluation via logical expression can be defined using the command `match-eval`. It is a tool used to combine multiple entries to define an accurate DLP sensor.

Sample topology

In this example, a Microsoft Office document with Visa credit card information is sent securely to the receiver using the HTTP POST method. The FortiGate intercepts the traffic using deep inspection and blocks the traffic as it matches the DLP profile configured on this FortiGate.

To block HTTPS upload traffic that includes Visa or Mastercard credit card information in the GUI:

1. Configure the DLP dictionary:
   a. Go to *Security Profiles* > *Data Loss Prevention*, select the *Dictionary* tab, and click *Create New*.
   b. Create an entry for Visa:
      i. Enter a name (*Finance_Credit_Card_Visa*).
      ii. In the *Dictionary Entries* section, click *Create New*.
      iii. Set the *Type* to *regex*.
      iv. Set *Pattern* to *4[0-9]{12}(?:[0-9]{3})* and click *OK*.
      v. Click *OK* to save the dictionary.
   c. Create an entry for Mastercard:

i. Enter a name (*Finance_Credit_Card_Mastercard*).
ii. In the *Dictionary Entries* section, click *Create New*.
iii. Set the *Type* to *regex*.
iv. Set *Pattern* to *(?:5[1-5][0-9]{2}|222[1-9]|22[3-9][0-9]|2[3-6][0-9]{2}|27[01][0-9]|2720)[0-9]{12}* and click *OK*.
v. Click *OK* to save the dictionary.

d. Create an entry for Credit Card:
i. Enter a name (*CC_Number*).
ii. In the *Dictionary Entries* section, click *Create New*.
iii. Set the *Type* to *credit-card* and click *OK*.
iv. Click *OK* to save the dictionary.

2. Configure the DLP sensor:
a. Go to *Security Profiles* > *Data Loss Prevention*, select the *Sensor* tab, and click *Create New*.
b. Enter a name (*Finance_Credit_Card_High*).
c. In the *Sensor Entries* section, click *Add*. The *Select Entries* pane is displayed.
d. Select *Managed Locally* from the dropdown menu.
e. Select *Finance_Credit_Card_Visa* and click *Apply*.
f. Repeat the previous step twice to add *Finance_Credit_Card_Mastercard* and *CC_Number* in this order.
g. Click *OK* to save the sensor.
h. Edit the newly created sensor.
i. Set the *Match criteria* to *Custom expression*.
j. In the *Evaluated by* field, enter *(dict(1) > 0 && dict(3) > 0) || (dict(2) > 0 && dict(3) > 0)*.
k. Click *OK* to save the sensor.

For the DLP sensor with the *Logical relationship* set to *Evaluate*, *Count* and *Status* of any sensor entry will be ignored.

3. Configure the DLP profile:
a. Go to *Security Profiles* > *Data Loss Prevention*, select the *Profile* tab, and click *Create New*.
b. Enter a name (*cc-block*).
c. In the *Rules* section, click *Create New*.
d. Configure the following settings:

Name	*1*
Sensors	*Finance Credit Card High*
Severity	*Critical*
Action	*Block*
Type	*File*
File type	*builtin-patterns*
Protocol	*HTTP-POST*, *HTTP-GET*

e. Click *OK*.

f. Click *OK* to save the profile.

4. Add the DLP profile to a firewall policy:
   a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
   b. Set the *Inspection Mode* to *Proxy-based*.
   c. In the *Security Profiles* section, enable *DLP Profile* and select *cc-block*.
   d. Set *SSL Inspection* to *deep-inspection* to inspect HTTPS traffic.
   e. Configure the other settings as needed.
   f. Click *OK*.

When a Visa or Mastercard credit card is included in HTTP GET or POST traffic, the file is blocked and a DLP log is generated. See the Sample log on page 2654 for details on how to test this configuration.

To block HTTPS upload traffic that includes Visa or Mastercard credit card information in the CLI:

1. Configure the DLP dictionary:

```
config dlp dictionary
    edit "Finance_Credit_Card_Visa"
        config entries
            edit 1
                set type "regex"
                set pattern "4[0-9]{12}(?:[0-9]{3})"
                set repeat enable
                set comment "Visa"
            next
        end
    next
    edit "Finance_Credit_Card_Mastercard"
        config entries
            edit 1
                set type "regex"
                set pattern "(?:5[1-5][0-9]{2}|222[1-9]|22[3-9][0-9]|2[3-6][0-9]{2}|27[01][0-
9]|2720)[0-9]{12}"
                set repeat enable
                set comment "Mastercard"
            next
        end
    next
    edit "CC_Number"
        config entries
            edit 1
                set type "credit-card"
            next
        end
    next
end
```

To use "?" in a regex pattern, see CLI basics on page 76. This method only supports direct console connection and SSH. It does not support the CLI console in the GUI.

2. Configure the DLP sensor:

```
config dlp sensor
    edit "Finance_Credit_Card_High"
        config entries
            edit 1
                   set dictionary "Finance_Credit_Card_Visa"
            next
            edit 2
                set dictionary "Finance_Credit_Card_Mastercard"
            next
            edit 3
                set dictionary "CC_Number"
            next
        end
        set match-type match-eval
        set eval "(dict(1) > 0 && dict(3) > 0) || (dict(2) > 0 && dict(3) > 0)"
    next
end
```

3. Configure the DLP profile:

```
config dlp profile
    edit "cc_block"
        set feature-set proxy
        config rule
            edit 1
                set name "1"
                set severity critical
                set proto http-get http-post
                set filter-by sensor
                set file-type 1
                set sensor "Finance_Credit_Card_High"
                set action block
            next
        end
    next
end
```

4. Add the DLP profile to a firewall policy:

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "port3"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "deep-inspection"
        set dlp-profile "cc_block"
```

```
        set nat enable
    next
end
```

When a Visa or Mastercard credit card is included in HTTP GET or POST traffic, a replacement message appears because it is blocked. A DLP log is generated. See the Sample log on page 2654 for details on how to test this configuration.

Sample log

From Windows, the following command below can be used to generate a sample log via HTTP POST traffic, using the cURL tool to post data, which contains a sample Visa credit card number. See sample-data for sample credit card numbers.

```
# curl –k -d 4024007149133315 https://192.168.10.13/cc.doc -o?

1: date=2023-03-17 time=15:37:41 eventtime=1679092660998869199 tz="-0700" logid="0954024576"
type="utm" subtype="dlp" eventtype="dlp" level="warning" vd="root" ruleid=1 rulename="1"
dlpextra="builtin-patterns;Finance_Credit_Card_High" filtertype="sensor" filtercat="file"
severity="critical" policyid=1 poluuid="26540ed0-ae54-51ed-80eb-89af8af4d53f" policytype="policy"
sessionid=14854 epoch=570215534 eventid=0 srcip=172.20.120.13 srcport=58012 srccountry="Reserved"
srcintf="port2" srcintfrole="undefined" srcuuid="3342cb44-9140-51ed-5dbe-8e0787bedeec"
dstip=192.168.10.13 dstport=443 dstcountry="Reserved" dstintf="port3" dstintfrole="wan"
dstuuid="3342cb44-9140-51ed-5dbe-8e0787bedeec" proto=6 service="HTTPS" filetype="msoffice"
direction="incoming" action="block" hostname="192.168.10.13" url="https://192.168.10.13/cc.doc"
agent="curl/7.83.1" httpmethod="POST" filename="cc.doc" filesize=12288 profile="cc-block"
```

Proximity search

In this example, any HTTPS upload traffic containing both a keyword and a username within a specified proximity can be identified. The FortiGate intercepts this traffic through deep inspection. When HTTPS upload traffic includes both a keyword and a username, which match the criteria defined on the data loss prevention (DLP) profile configured on the FortiGate, the traffic is blocked. This action subsequently generates a DLP log. See Verification on page 2658 for a log sample.

To block HTTPS upload traffic that match the DLP profile in the GUI:

1. Configure the DLP dictionary:
   a. Go to *Security Profiles* > *Data Loss Prevention* and select the *Dictionary* tab.
   b. Select *Create New*.
   c. Set *Name* to: *matcharound*.
   d. In the *Dictionary Entries* table, click *Create New*:
      i. Set *Type* to *keyword*
      ii. Set *Pattern* to *user*
      iii. Enable *Case sensitive*
      iv. Click *OK*.
   e. Click *OK*.
   f. Select the newly created dictionary and click *Edit*.

g. Select *Edit in CLI* and enter the following command:
```
#set match-around enable
end
```
h. Close the CLI Console and click *Cancel*.

> DLP data types can only be configured in the CLI. See step 2 of To block HTTPS upload traffic that match the DLP profile in the CLI: on page 2656.

2. Configure the DLP dictionary:
   a. Go to *Security Profiles* > *Data Loss Prevention* and select the *Dictionary* tab.
   b. Click *Create New*.
   c. Set *Name* to *username*.
   d. In the *Dictionary Entries* table, click *Create New*:
      i. Set *Type* to *user*
      ii. Click *OK*.
   e. Click *OK*.
3. Configure the DLP sensor:
   a. Go to *Security Profiles* > *Data Loss Prevention* and select the *Sensor* tab.
   b. Click *Create New*.
   c. Enter a name *(user)*.
   d. In the *Sensor Entries* section, click *Add*. The *Select Entries* pane is displayed.
   e. Select *Managed Locally* from the dropdown menu.
   f. Select *username* and click *Apply*.
   g. Click *OK* to save the sensor.
4. Configure the DLP profile:
   a. Go to *Security Profiles* > *Data Loss Prevention* and select the *Profile* tab.
   b. Click *Create New*.
   c. Enter a name *(keyword)*.
   d. In the *Rules* section, click *Create New*.
   e. Configure the following settings:

Name	keyword
Data source type	Sensor
Sensors	user
Severity	Critical
Action	Block
Match Type	Message
Protocol	HTTP-POST

   f. Click *OK*.
   g. Click *OK* to save the profile.
5. Add the DLP profile to a firewall policy:

a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
b. Set the *Inspection Mode* to *Proxy-based*.
c. In the *Security Profiles* section, enable *DLP Profile* and select *keyword*.
d. Set *SSL Inspection* to *deep-inspection* to inspect HTTPS traffic.
e. Configure the other settings, as needed.
f. Click *OK*.

To block HTTPS upload traffic that match the DLP profile in the CLI:

1. Configure the DLP dictionary:

```
config dlp dictionary
    edit "matcharound"
        set match-around enable
        config entries
            edit 1
                set type "keyword"
                set pattern "user"
                set ignore-case enable
            next
        end
    next
end
```

2. Configure the DLP data type:

```
config dlp data-type
    edit "user"
        set pattern "\\b[a-zA-Z]{6,12}\\b"
        set verify "(?<=@)\\w+"
        set match-around "matcharound"
        set look-back 13
        set match-back 15
    next
end
```

The pattern specified using the `set pattern` command, is designed to match any word, irrespective of case sensitivity, that contains between 6 and 12 characters. On the other hand, the pattern specified using the `set verify` command employs a positive lookbehind assertion. This assertion checks for the presence of the @ symbol preceding the word, without including it in the match. This is a feature supported in PCRE but not in Hyperscan. For a match to occur, the content must satisfy all parameters defined in the DLP Data type.

3. Configure the DLP dictionary:

```
config dlp dictionary
    edit "username"
        config entries
            edit 1
```

```
                set type "user"
            next
        end
    next
end
```

4. Configure the DLP sensor:

```
config dlp sensor
    edit "user"
        config entries
            edit 1
                set dictionary "username"
            next
        end
    next
end
```

5. Configure the DLP profile:

```
config dlp profile
    edit "keyword"
        set feature-set proxy
        config rule
            edit 1
                set name "keyword"
                set severity critical
                set type message
                set proto http-post
                set filter-by sensor
                set sensor "user"
                set action block
            next
        end
    next
end
```

6. Add the DLP profile to a firewall policy:

```
config firewall policy
    edit 1
        set name "keyword"
        set srcintf "port2"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
```

```
        set inspection-mode proxy
        set ssl-ssh-profile "deep-inspection"
        set dlp-profile "keyword"
        set logtraffic all
        set nat enable
    next
end
```

Verification

1. Visit https://dlptest.ai/http-post/com.
2. Enter any phrase that will match the DLP profile configured on the FortiGate and click *Submit*.
   Example:
   user: @kikinaynay
3. Verify that the replacement message indicates the transfer attempt has been blocked.
4. Review the log that was generated when the attempt was made to send an HTTP POST request containing sensitive keywords:

```
1: date=2024-05-17 time=13:12:33 eventtime=1715908352455559762 tz="+1200"
logid="0954024576" type="utm" subtype="dlp" eventtype="dlp" level="warning" vd="root"
ruleid=1 rulename="keyword" dlpextra="Sensor 'user' matching any: ('username'=1) >= 1;
match." filtertype="sensor" filtercat="message" severity="critical" policyid=1
poluuid="8abe7a4e-08ae-51ef-edb0-45c05b514641" policytype="policy" sessionid=18462
epoch=1293108816 eventid=1 srcip=13.13.13.13 srcport=64341 srccountry="United States"
srcintf="port2" srcintfrole="undefined" srcuuid="6e01eac6-a97d-51ed-5220-dac5db63d2ca"
dstip=35.209.95.242 dstport=443 dstcountry="United States" dstintf="port1"
dstintfrole="wan" dstuuid="6e01eac6-a97d-51ed-5220-dac5db63d2ca" proto=6 service="HTTPS"
filetype="N/A" direction="outgoing" action="block" hostname="dlptest.ai"
url="https://dlptest.ai/http-post/" agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/124.0.0.0 Safari/537.36" httpmethod="POST"
referralurl="https://dlptest.ai/http-post/" profile="keyword"
```

Assign confidence levels in FortiGuard managed DLP dictionaries

Users can select a FortiGuard dictionary with varying confidence levels based on their specific requirements.

- The high level provides maximum precision to minimize false positives.
- The medium level balances match quantity and precision.
- The low level captures the most matches, but may result in more false positives.

> A valid DLP license is required to obtain the latest package.
>
> When applying a FortiGuard built-in dictionary to a custom sensor, the dictionary with the highest confidence level is selected by default.

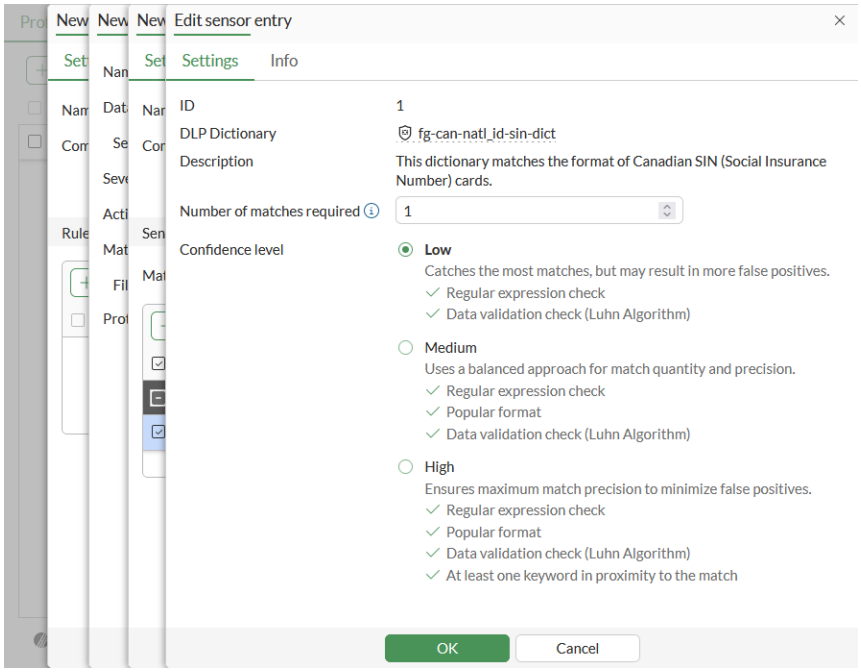
Use case examples

In these use case examples, various Canadian Social Insurance Number (SIN) formats are tested at different confidence levels using different protocols.

SIN format	Low Confidence Matching criteria: regular expression, data validation	Medium Confidence Matching criteria: regular expression, data validation SIN format validation	High Confidence Matching criteria: regular expression, data validation, SIN format validation, Match-around data
815489034	match	does not match	does not match
193849270	match	match	does not match
sin# 193849270	match	match	match

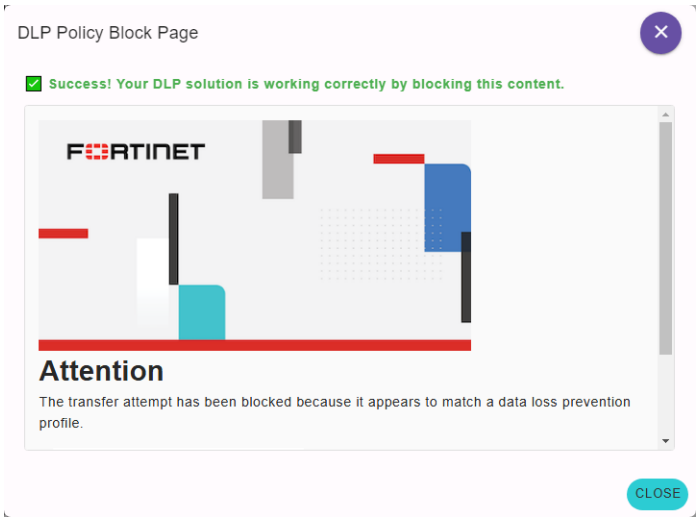
To verify that a FortiGuard dictionary with the low confidence level will block matching message through an HTTPS post:

1. Configure a DLP profile with a DLP sensor that uses the Canadian SIN card dictionary (`fg-can-natl_id-sin-dict`) DLP dictionary with the *Confidence level* set to *Low* and then use the profile in a policy.



2. Test that an HTTPS message containing a SIN is blocked. DLP Test > HTTPS Post can be used to send a test message.

   The message is blocked:

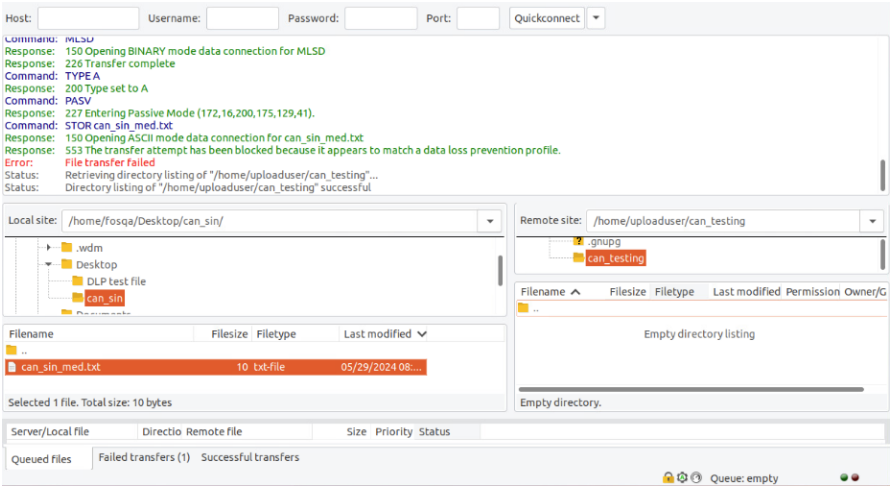


3. Go to *Log & Report > Security Events* and view the *Data Loss Prevention* logs matching the *can-natl_id-sin-dict-low* dictionary.
4. Check the raw logs:

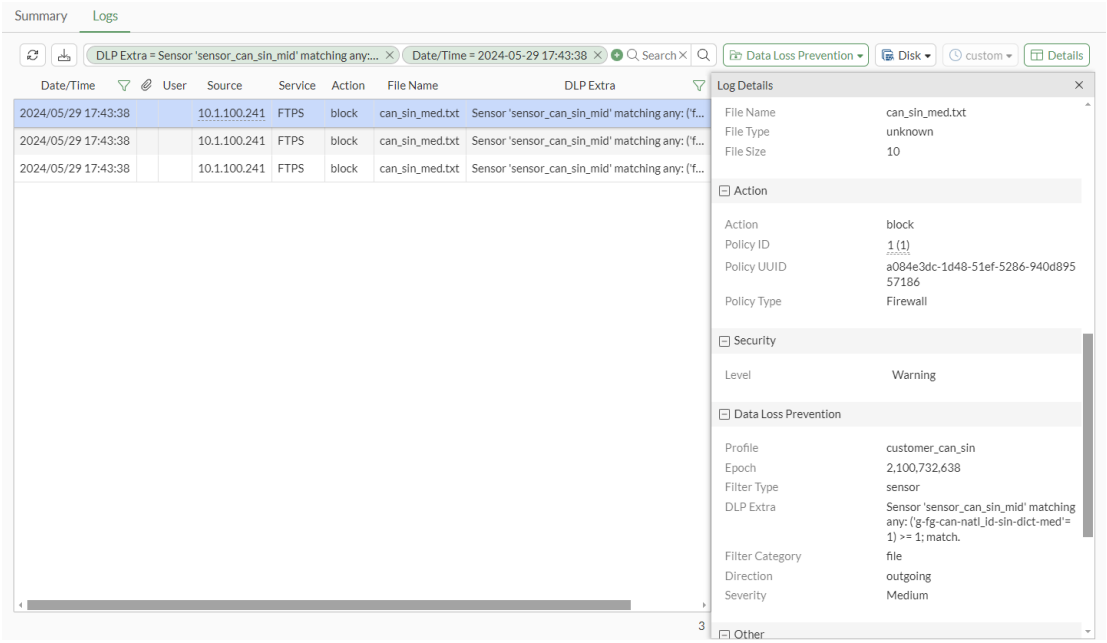
```
1: date=2024-05-29 time=16:55:27 eventtime=1717026926501493215 tz="-0700" logid="0954024576"
type="utm" subtype="dlp" eventtype="dlp" level="warning" vd="vdom1" ruleid=1 rulename="sensor_
can_sin_low" dlpextra="Sensor 'sensor_can_sin_low' matching any: ('g-fg-can-natl_id-sin-dict-
low'=1) >= 1; match." filtertype="sensor" filtercat="message" severity="medium" policyid=1
poluuid="a084e3dc-1d48-51ef-5286-940d89557186" policytype="policy" sessionid=64304
epoch=2100732550 eventid=1 srcip=10.1.100.241 srcport=34184 srccountry="Reserved"
srcintf="port2" srcintfrole="undefined" srcuuid="5a9d01f6-1d48-51ef-1c5f-c5a49f106988"
dstip=35.209.95.242 dstport=443 dstcountry="United States" dstintf="port1"
dstintfrole="undefined" dstuuid="5a9d01f6-1d48-51ef-1c5f-c5a49f106988" proto=6 service="HTTPS"
filetype="N/A" direction="outgoing" action="block" hostname="dlptest.ai"
url="https://dlptest.ai/https-post/" agent="Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/122.0.0.0 Safari/537.36" httpmethod="POST"
referralurl="https://dlptest.ai/https-post/" profile="customer_can_sin"
```

To verify that a FortiGuard dictionary with medium confidence level will block matching message through a FTPS post:

1. Configure a DLP profile with a DLP sensor that uses the Canadian SIN card dictionary (`fg-can-natl_id-sin-dict`) DLP dictionary with the *Confidence level* set to *Medium* and then use the profile in a policy.
2. Test that posting a file that contains `193849270` is blocked.



3. Go to *Log & Report > Security Events* and view the *Data Loss Prevention* logs matching the *can-natl_id-sin-dict-med* dictionary.



4. Check the raw logs:

```
1: date=2024-05-29 time=17:43:38 eventtime=1717029818309788622 tz="-0700" logid="0954024576"
type="utm" subtype="dlp" eventtype="dlp" level="warning" vd="vdom1" ruleid=2 rulename="sensor_
can_sin_med" dlpextra="Sensor 'sensor_can_sin_mid' matching any: ('g-fg-can-natl_id-sin-dict-
med'=1) >= 1; match." filtertype="sensor" filtercat="file" severity="medium" policyid=1
poluuid="a084e3dc-1d48-51ef-5286-940d89557186" policytype="policy" sessionid=65893
epoch=2100732638 eventid=0 srcip=10.1.100.241 srcport=37561 srccountry="Reserved"
srcintf="port2" srcintfrole="undefined" srcuuid="5a9d01f6-1d48-51ef-1c5f-c5a49f106988"
dstip=172.16.200.175 dstport=33065 dstcountry="Reserved" dstintf="port1"
dstintfrole="undefined" dstuuid="5a9d01f6-1d48-51ef-1c5f-c5a49f106988" proto=6 service="FTPS"
filetype="unknown" direction="outgoing" action="block" filename="can_sin_med.txt" filesize=10
profile="customer_can_sin"
```

To verify that the FortiGuard dictionary with a high confidence level will block matching message through an SMTP post:

1. Configure a DLP profile with a DLP sensor that uses the Canadian SIN card dictionary (`fg-can-natl_id-sin-dict`) DLP dictionary with the *Confidence level* set to *High* and then use the profile in a policy.
2. Test that sending email with an attached file that contains `sin# 193849270` is blocked.
3. Go to *Log & Report > Security Events* and view the *Data Loss Prevention* logs matching the *can-natl_id-sin-dict-high* dictionary.
4. Check the raw logs:

```
1: date=2024-05-30 time=11:37:18 eventtime=1717094238851929893 tz="-0700" logid="0954024576"
type="utm" subtype="dlp" eventtype="dlp" level="warning" vd="vdom1" ruleid=3 rulename="sensor_
can_sin_high" dlpextra="Sensor 'sensor_can_sin_high' matching any: ('g-fg-can-natl_id-sin-
dict-high'=1) >= 1; match." filtertype="sensor" filtercat="file" severity="medium" policyid=1
poluuid="a084e3dc-1d48-51ef-5286-940d89557186" policytype="policy" sessionid=96838
epoch=1455065196 eventid=2 srcip=10.1.100.171 srcport=51141 srccountry="Reserved"
srcintf="port2" srcintfrole="undefined" srcuuid="5a9d01f6-1d48-51ef-1c5f-c5a49f106988"
dstip=172.16.200.175 dstport=25 dstcountry="Reserved" dstintf="port1" dstintfrole="undefined"
dstuuid="5a9d01f6-1d48-51ef-1c5f-c5a49f106988" proto=6 service="SMTP" filetype="unknown"
direction="outgoing" action="block" from="johndoe@example.com" to="smithwhite@example.com"
sender="emailuser1@qa.fortinet.com" recipient="emailuser2@qa.fortinet.com" subject="Canadian
SIN" attachment="yes" filename="sin.txt" filesize=70 profile="customer_can_sin"
```

FortiData labels

When FortiData and FortiGate are integrated, you can use FortiData labels for DLP processing in FortiGate proxy-based firewall policies. For information about integrating FortiData with FortiGate, see FortiData on page 133.

FortiData is used to create labels by discovering, classifying, and labeling files with sensitive in your file storage system. See the *FortiData Administration Guide* for information.

Once FortiData labels are created, you can configure DLP labels in FortiOS, reference the FortiData labels in a DLP profile, and then apply the DLP profile to a proxy-based firewall policy.

To use FortiData labels in FortiOS:

1. Configure a DLP label of the FortiData type:

```
config dlp label
    edit "fortidata"
        set type fortidata
        config entries
            edit 1
                set fortidata-label-name "Critical"
            next
        end
    next
end
```

2. Reference the FortiData DLP label in your DLP profile:

```
config dlp profile
    edit "fortidata"
        set feature-set proxy
        config rule
            edit 1
                set proto
                set filter-by label
                set name fortidata
                set action block
            next
        end
    next
end
```

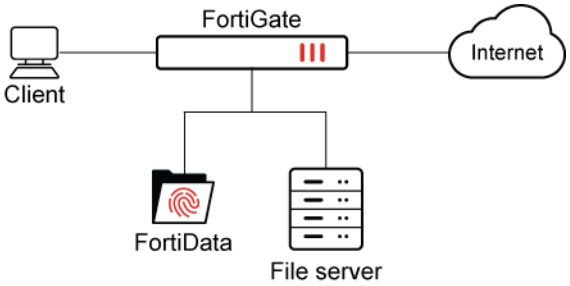
3. Apply your DLP profile into a proxy-based firewall policy on your FortiGate to scan your traffic and detect any data loss based on the label of your document.

```
config firewall policy
    edit 1
        set inspection-mode proxy
        set dlp-profile fortidata
    next
end
```

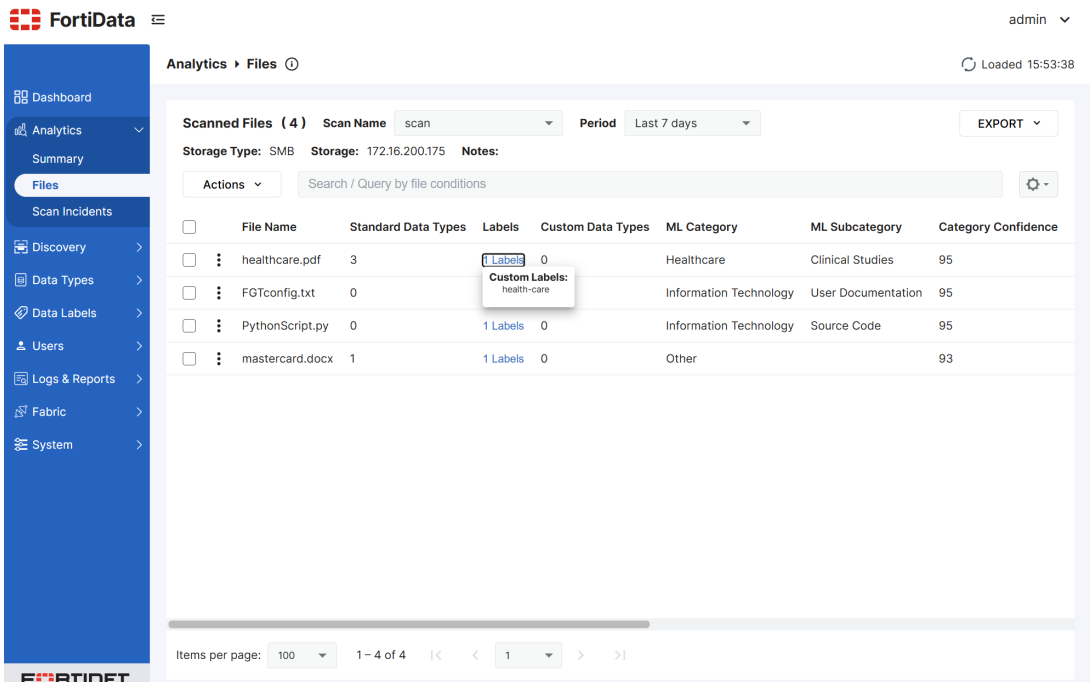
Example

This example describes how to use following FortiData labels in FortiOS DLP processing: health-care, credit-card, and python-source-code.

A file server is configured, and labels are selected in FortiData. FortiData scans the file server and applies labels to matching files.



Following is an example FortiData file source with the labels health-care, credit-card, and python-source-code. Only the *health-care* label is visible in the screen shot:



The next step is to configure FortiGate to use the FortiData labels. See Configuring FortiGate to use FortiData labels on page 2664.

With FortiData and FortiGate configured, the labels can be used for DLP processing in FortiGate policies. The results of DLP processing are described in the following example scenarios:

- Scenario 1: FortiData error action on page 2666
- Scenario 2: DLP action quarantine on page 2667
- Scenario 3: DLP action monitor on page 2669

Configuring FortiGate to use FortiData labels

Configure the following FortiData labels in FortiOS: health-care, credit-card, and python-source-code.

To configure FortiGate to use FortiData labels:

1. In FortiOS, configure DLP FortiData labels:

```
config dlp label
    edit "fortidata-label"
        set type fortidata
        config entries
            edit 1
                set fortidata-label-name "python-source-code"
            next
            edit 2
                set fortidata-label-name "credit-card"
            next
```

```
            edit 3
                set fortidata-label-name "health-care"
            next
        end
    next
end
```

2. Configure a DLP profile:
   In this example, `action` is set to `block`, and `fortidata-error-action` is set to `block`:

```
config dlp profile
    edit "fortidata-sensitive"
        set feature-set proxy
        config rule
            edit 1
                set type file
                set proto http-get http-post ftp
                set filter-by label
                set label "fortidata-label"
                set action block
            next
        end
        set fortidata-error-action block
    next
end
```

3. Create a firewall policy:

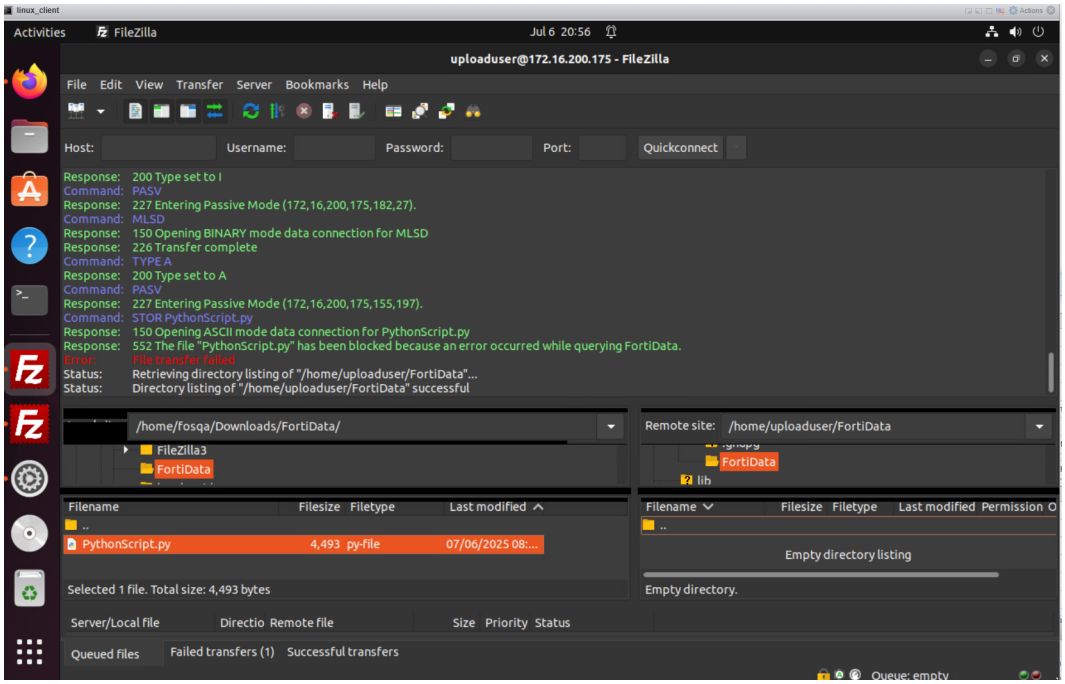
```
config firewall policy
    edit 1
        set uuid 19e776e0-5c63-51f0-9d68-3053988cbc21
        set srcintf "port2"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set profile-protocol-options "protocol"
        set ssl-ssh-profile "protocols"
        set dlp-profile "fortidata-sensitive"
        set nat enable
    next
end
```

Scenario 1: FortiData error action

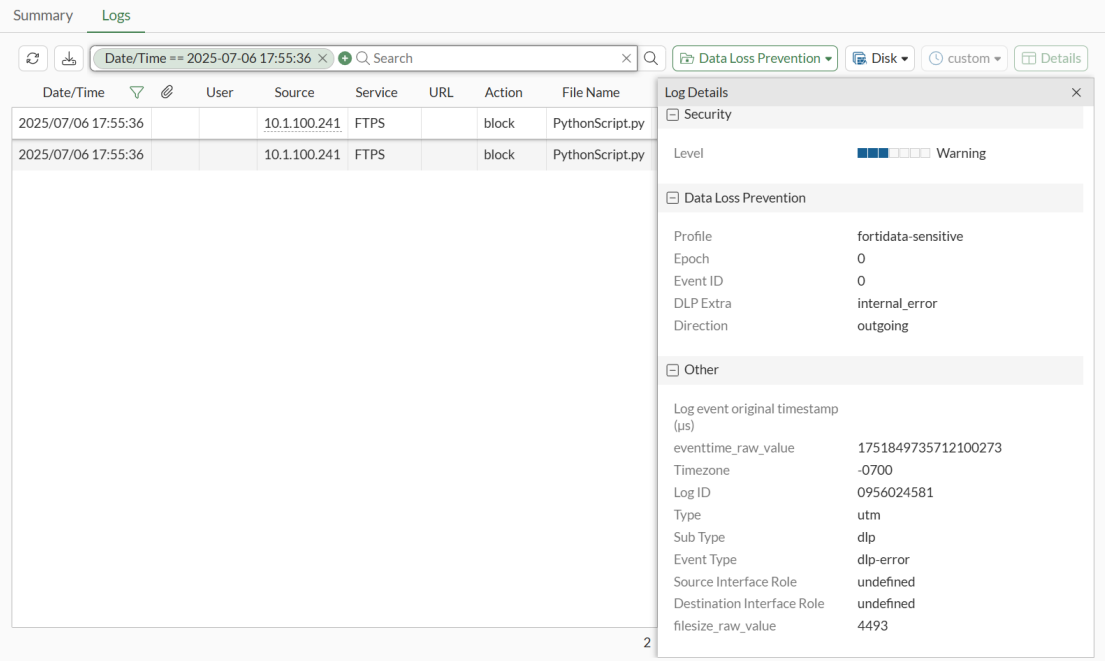
This scenario demonstrates what happens when the FortiGate DLP profile has `fortidata-error-action` set to `block`, and the FortiData query fails.

When the user attempts to post a `PythonScript.py` file through FTPS and the FortiData query fails, FortiGate blocks the attempt with an error message.

The following screen shot shows the `PythonScript.py` file has been blocked:



A DLP log with block action is generated on FortiGate, and you can view the log in the GUI on the *Log & Report > Security Events > Data Loss Prevention* pane:



You can also view the log in the CLI:

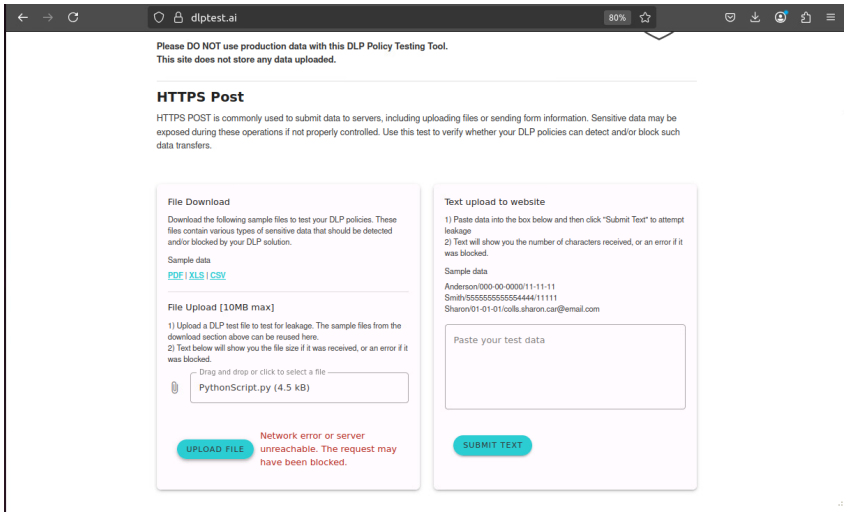
```
30: date=2025-07-06 time=17:55:24 eventtime=1751849724254438363 tz="-0700" logid="0956024581"
type="utm" subtype="dlp" eventtype="dlp-error" level="warning" vd="vdom1" error="fortidata"
dlpextra="internal_error" policyid=1 poluuid="8a20ddd4-5abf-51f0-3fcd-80a21a2eac18"
policytype="policy" sessionid=3142 epoch=0 eventid=0 srcip=10.1.100.241 srcport=58837
srccountry="Reserved" srcintf="port2" srcintfrole="undefined" srcuuid="5146e274-5abf-51f0-48e4-
5a4728500b83" dstip=172.16.200.175 dstport=40325 dstcountry="Reserved" dstintf="port1"
dstintfrole="undefined" dstuuid="5146e274-5abf-51f0-48e4-5a4728500b83" proto=6 service="FTPS"
filetype="python" direction="outgoing" action="block" filename="PythonScript.py" filesize=4493
profile="fortidata-sensitive"
```

Scenario 2: DLP action quarantine

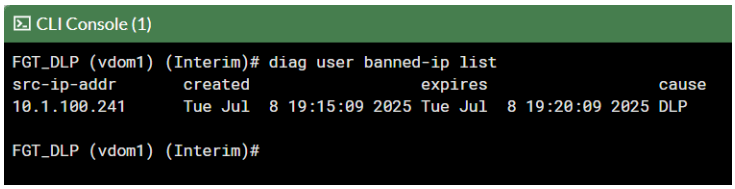
This scenario demonstrates what happens when the FortiGate DLP profile has `action` set to `quarantine-ip`:

```
config dlp profile
    edit "fortidata-sensitive"
        set feature-set proxy
        config rule
            edit 1
                set proto http-get http-post ftp ssh
                set filter-by label
                set label "fortidata-label"
                set action quarantine-ip
            next
        end
    next
end
```

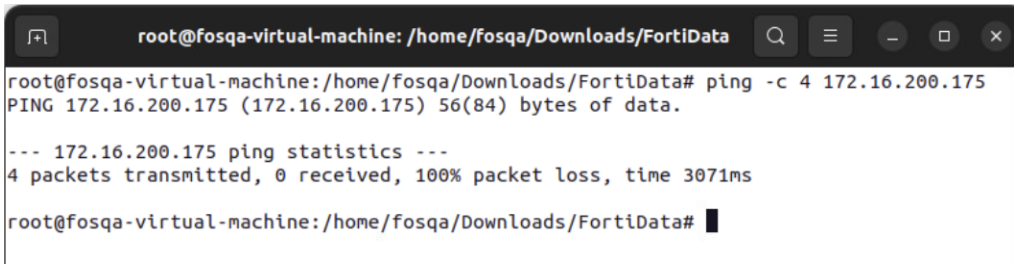
When the user attempts to post a `PythonScript.py` file through dlptest.ai that matches `python-source-code` FortiData label, FortiGate blocks the file and quarantines the IP:



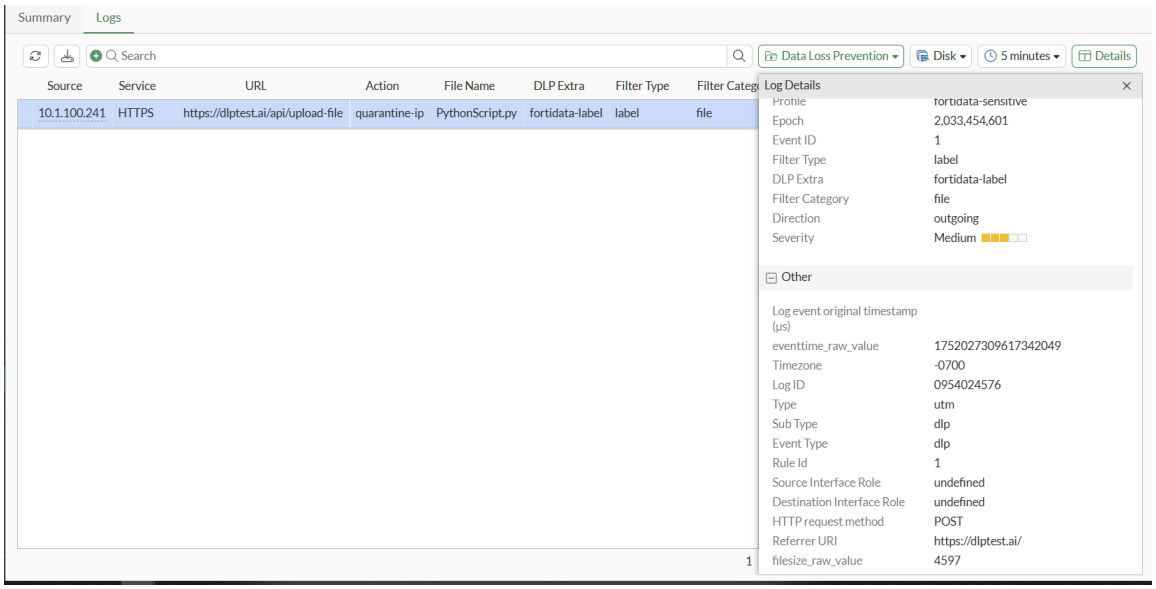
In FortiGate, the client is banned:



The client cannot ping the server through FortiGate:



A DLP log with quarantine-ip action is generated on FortiGate, and you can view the log in the GUI on the *Log & Report > Security Events > Data Loss Prevention* pane:



You can view the log in the CLI:

```
1: date=2025-07-08 time=19:15:09 eventtime=1752027309617342049 tz="-0700" logid="0954024576"
type="utm" subtype="dlp" eventtype="dlp" level="warning" vd="vdom1" ruleid=1 dlpextra="fortidata-
label" filtertype="label" filtercat="file" severity="medium" policyid=1 poluuid="19e776e0-5c63-
51f0-9d68-3053988cbc21" policytype="policy" sessionid=3465 epoch=2033454601 eventid=1
srcip=10.1.100.241 srcport=37258 srccountry="Reserved" srcintf="port2" srcintfrole="undefined"
srcuuid="dea915de-5c62-51f0-b993-090ad249da82" dstip=206.47.132.132 dstport=443
dstcountry="Canada" dstintf="port1" dstintfrole="undefined" dstuuid="dea915de-5c62-51f0-b993-
090ad249da82" proto=6 service="HTTPS" filetype="python" direction="outgoing" action="quarantine-
ip" hostname="dlptest.ai" url="https://dlptest.ai/api/upload-file" agent="Mozilla/5.0 (X11;
Ubuntu; Linux x86_64; rv:139.0) Gecko/20100101 Firefox/139.0" httpmethod="POST"
referralurl="https://dlptest.ai/" filename="PythonScript.py" filesize=4597 profile="fortidata-
sensitive"
```

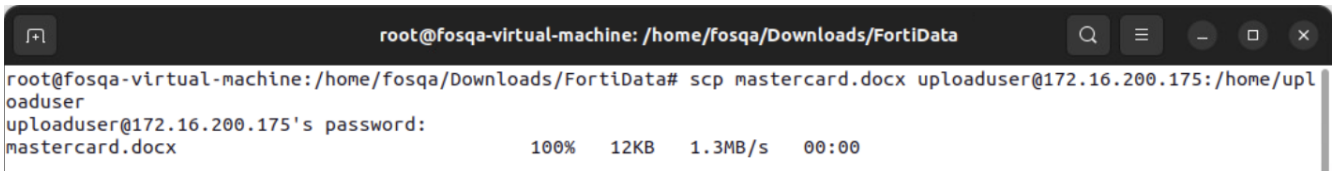
Scenario 3: DLP action monitor

This scenario demonstrates what happens when the FortiGate DLP profile has `action` set to `monitor`:

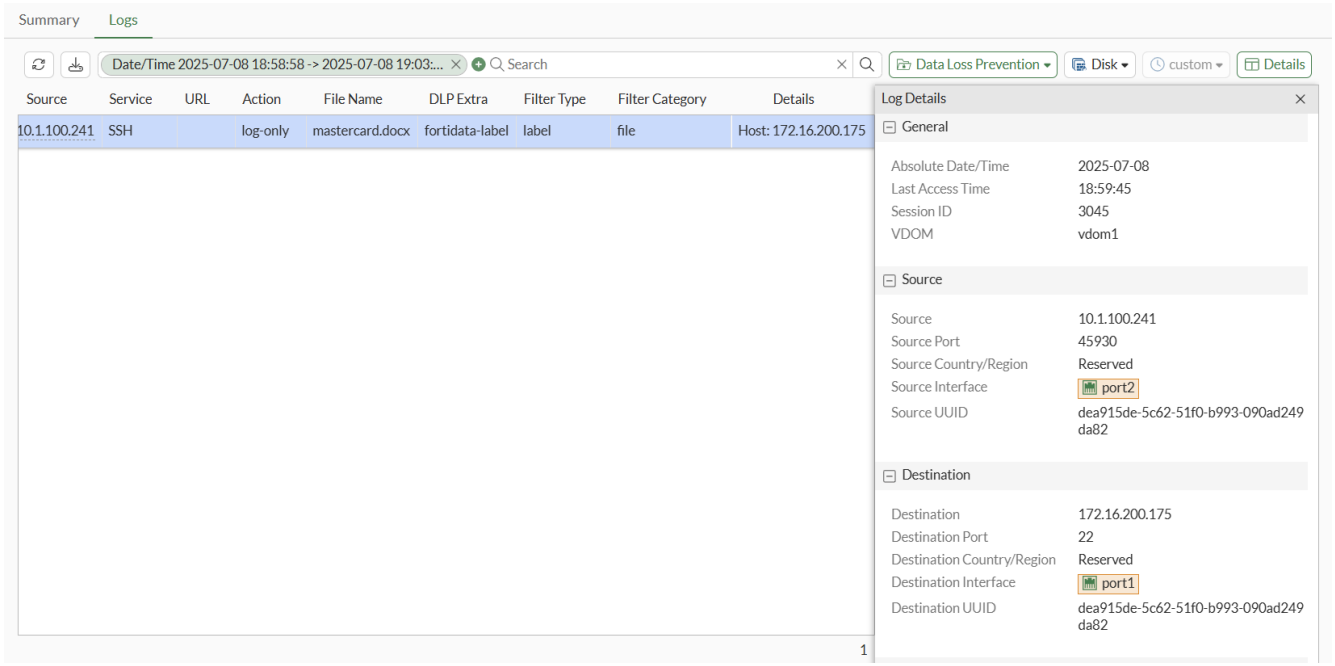
```
config dlp profile
    edit "fortidata-sensitive"
        set feature-set proxy
        config rule
            edit 1
                set proto http-get http-post ftp ssh
                set filter-by label
                set label "fortidata-label"
                set action monitor
            next
        end
    next
end
```

When the user attempts to post a file (`mastercard.docx`) that matches a FortiData label, FortiGate passes the file through.

User secure copy to upload the `mastercard.docx` file to the server:



A DLP log with log-only action is generated on FortiGate, and you can view the log in the GUI on the *Log & Report > Security Events > Data Loss Prevention* pane



You can view the log in the CLI:

```
1: date=2025-07-08 time=18:59:45 eventtime=1752026384529707336 tz="-0700" logid="0954024577"
type="utm" subtype="dlp" eventtype="dlp" level="notice" vd="vdom1" ruleid=1 dlpextra="fortidata-
label" filtertype="label" filtercat="file" severity="medium" policyid=1 poluuid="19e776e0-5c63-
51f0-9d68-3053988cbc21" policytype="policy" sessionid=3045 epoch=2033454538 eventid=0
srcip=10.1.100.241 srcport=45930 srccountry="Reserved" srcintf="port2" srcintfrole="undefined"
srcuuid="dea915de-5c62-51f0-b993-090ad249da82" dstip=172.16.200.175 dstport=22
dstcountry="Reserved" dstintf="port1" dstintfrole="undefined" dstuuid="dea915de-5c62-51f0-b993-
090ad249da82" proto=6 service="SSH" subservice="SCP" filetype="msofficex" direction="outgoing"
action="log-only" filename="mastercard.docx" filesize=12604 profile="fortidata-sensitive"
```

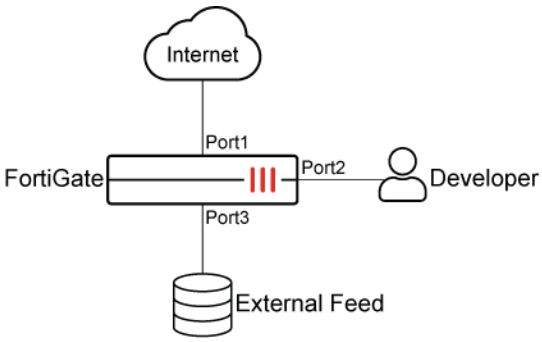
Prevent sensitive token leakage in HTTPS uploads using EDM

Exact Data Matching (EDM) enables precise matching of patterns or keywords from a predefined dataset, ensuring high accuracy. FortiGate can be configured to reference external resources containing lists of sensitive tokens, such as AWS Access Keys, JWTs (JSON Web Tokens), OAuth tokens, or other application secrets, that

are used in your organization. These resources are dynamically synchronized and periodically updated, so any changes are automatically imported by FortiOS.

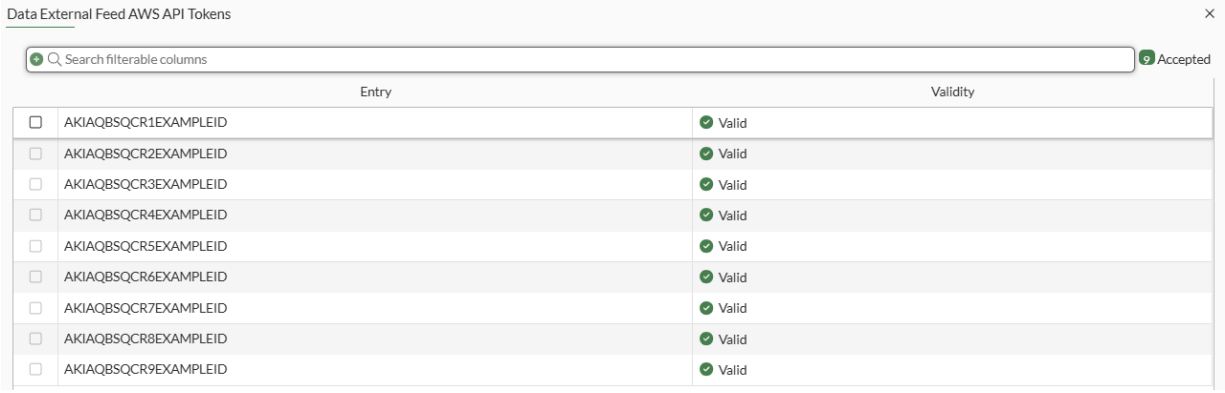
This configuration helps prevent accidental or intentional leakage of API keys, authentication tokens, or other secrets by blocking any code snippet containing these values when users attempt to upload files to external platforms such as GitHub, GitLab, or cloud storage services.

In this example, a file containing an AWS access token is being uploaded to an external GitHub repository using the HTTP POST method. FortiGate intercepts the traffic through deep inspection and blocks the upload because it matches the DLP profile configured on the device. This ensures that sensitive credentials, such as AWS Access Tokens, are not accidentally or intentionally leaked outside the organization.



To block HTTPS upload traffic that includes AWS access tokens in the GUI:

1. 1. Ensure that *Data Loss Prevention* is enabled:
   a. Go to *System* > *Feature Visibility*.
   b. Under *Security Features*, enable *Data Loss Prevention*.
   c. Click Apply.
2. Create an EDM template with matching criteria:
   a. Go to *Security Profiles* > *Data Loss Prevention*, select the *EDM Template* tab, and click *Create New*.
   b. Enter a name for the template, such as *AWS API Tokens*.
   c. Set *Resource type* to *External feed*, and set *External feed URL* to the location of the file on the external server, such as *https://13.13.13.13/aws_api_tokens.csv*.
   d. Click *+All of these fields* to pair the column index of patterns with a DLP data type.
      In this example, column 1 in the external resource file contains the patterns for the edm-keyword data type.
   e. Click *OK*.
   f. Edit the DLP EDM template and click *View Entries* to view the data entries in the field.



3. Configure a DLP sensor for the EDM template:
   a. Go to *Security Profiles* > *Data Loss Prevention*, select the *Sensor* tab, and click *Create New*.
   b. Enter a name for the DLP sensor, such as *AWS API Sensor*.
   c. In the *Sensor Entries* table, click *Add*.
   d. In the drop-down list, select *Managed Locally*.
   e. Select *AWS API Tokens* then click *Apply*.
   f. Click *OK*.
4. Create a DLP profile and select the DLP sensor for the EDM template:
   a. Go to *Security Profiles* > *Data Loss Prevention*, select the *Profile* tab, and click *Create New*.
   b. Enter a name for the DLP profile, such as *AWS API Profile*.
   c. In the *Rules* tables, click *Create New*.
   d. Enter a name for the rule, such as *AWS API Rule*.
   e. Configure the following, then click *OK*.

Field	Value
Type	Sensor
Sensors	AWS API Sensor
Severity	Critical
Action	Block
Match type	File
File type	(leave blank)
Protocol	HTTP-POST

   f. Click *OK*.
5. Add the DLP profile to a firewall policy:
   a. Go to *Policy & Objects* > *Firewall Policy*. and click *Create New*.
   b. Set *Inspection mode* to *Proxy-based*.
   c. Under *Security Profiles*, enable *DLP Profile* and select *AWS API Profile*.
   d. Set *SSL inspection* to *deep-inspection*.
   e. Configure the other settings as needed.
   f. Click *OK*.

To block HTTPS upload traffic that includes AWS access tokens in the CLI:

1. Create an EDM template with matching criteria:

```
config dlp exact-data-match
    edit "AWS API Tokens"
        set data "AWS API Tokens"
        config columns
            edit 1
                set type "edm-keyword"
            next
        end
    next
end
```

2. Configure a DLP sensor for the EDM template:

```
config dlp sensor
    edit "AWS API Sensor"
        config entries
            edit 1
                set dictionary "AWS API Tokens"
            next
        end
    next
end
```

3. Create a DLP profile and select the DLP sensor for the EDM template:

```
config dlp profile
    edit "AWS API Profile"
        set feature-set proxy
        config rule
            edit 1
                set name "AWS API Profile"
                set severity critical
                set proto http-post
                set filter-by sensor
                set sensor "AWS API Sensor"
                set action block
            next
        end
    next
end
```

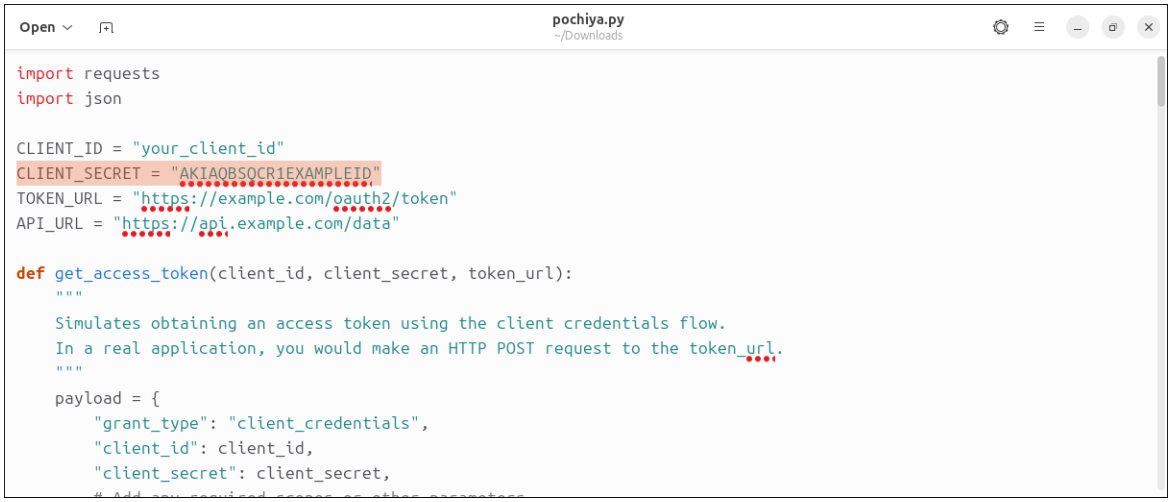
4. Add the DLP profile to a firewall policy:

```
config firewall policy
    edit 1
        set name "AWS_Token_Block"
        set srcintf "port2"
        set dstintf "port1"
        set action accept
```

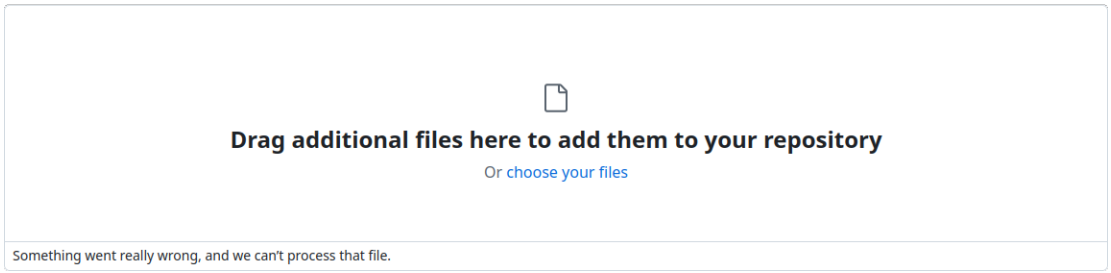
```
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "deep-inspection"
        set dlp-profile "AWS API Profile"
        set logtraffic all
        set nat enable
    next
end
```

To verify the configuration:

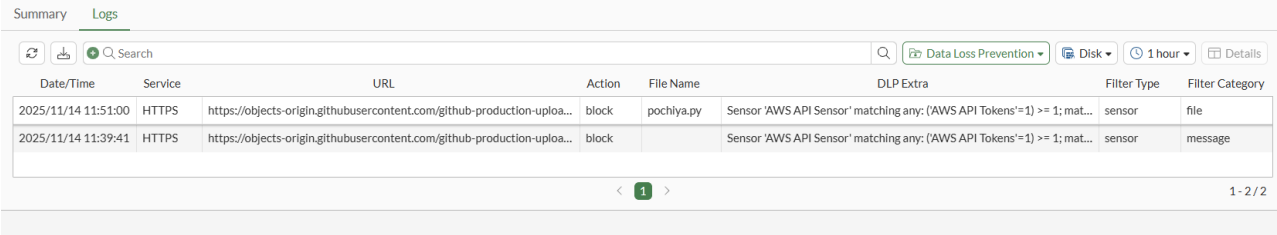
1. A developer attempts to upload a file containing AWS access tokens through HTTPS to github.com, and the file content matches the EDM template.



2. FortiGate blocks the user's attempt and displays an error message:



3. FortiGate generates a DLP log:



1: date=2025-11-14 time=11:51:00 eventtime=1763077860331286861 tz="+1200" logid="0954024576" type="utm" subtype="dlp" eventtype="dlp" level="warning" vd="root" ruleid=1 **rulename="AWS API Profile" dlpextra="Sensor 'AWS API Sensor' matching any: ('AWS API Tokens'=1) >= 1; match."** filtertype="sensor" filtercat="file" severity="critical" policyid=1 poluuid="3c57da8c-c0de-51f0-6f4a-95b898d269b3" policytype="policy" sessionid=28834 epoch=834063557 eventid=1 srcip=10.10.10.10 srcport=53912 srccountry="United States" srcintf="port2" srcintfrole="undefined" srcuuid="d2f06fda-15e7-51ee-0d22-faaf5170dad2" dstip=140.82.113.21 dstport=443 dstcountry="United States" dstintf="port1" dstintfrole="undefined" dstuuid="d2f06fda-15e7-51ee-0d22-faaf5170dad2" proto=6 **service="HTTPS" filetype="python"** direction="outgoing" **action="block"** hostname="objects-origin.githubusercontent.com" url="https://objects-origin.githubusercontent.com/github-production-upload-manifest-file-7fdce7" agent="Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:135.0) Gecko/20100101 Firefox/135.0" **httpmethod="POST" referralurl="https://github.com/" filename="pochiya.py"** filesize=2338 **profile="AWS API Profile"**

Virtual patching

Virtual patching is a method for mitigating vulnerability exploits against OT and IoT devices by applying patches virtually on the FortiGate. This is done in several steps:

1. A FortiGate uses the OT Detection Signatures and Service to collect device information from OT and IoT devices that are connected to an interface.
2. The device information is used to perform a vulnerability lookup by querying FortiGuard for device-specific vulnerabilities and mitigation rules.

   FortiGuard returns OT virtual patching signatures and IPS signatures.
3. The FortiGate caches the applicable signatures and mitigation rules that apply to each device. The signatures and rules are mapped to the MAC address of the device.
4. When a virtual patching profile is applied to a firewall policy, traffic that enters the firewall policy is subject to signature matching on a per-device basis.
   a. The IPS engine uses the MAC address of the device to match any mitigation rules that should apply.
   b. If the MAC address is in the exempted list, then patching is exempted or skipped.
   c. If the signature rule is in the exempted list, then patching is also exempted or skipped for that signature.
   d. Otherwise, all applicable rules for the device will be applied.

OT and IoT device detection

When device detection is enabled on a LAN interface, FortiGate detects OT and IoT devices that are connected to the interface by using the OT Detection Signatures and Service.

Device detection can be enabled in the GUI and CLI. In the GUI, go to *Network* > *Interfaces*, edit a LAN interface, enable *Device detection*, and click *OK*. In the CLI:

```
config system interface
    edit <name>
        set device-identification enable
    next
end
```

You can further refine OT and IoT device detection by excluding OT and/or IoT application control signatures in the CLI.

```
config system interface
    edit <name>
        set device-identification enable
        set exclude-signatures {iot ot}
    next
end
```

Command	Description
device-identification {enable | disable}	Enable/disable passively gathering of device identity information about the devices on the network connected to this interface (default = disable).
exclude-signatures {iot ot}	Exclude OT and/or IoT application control signatures. This option is hidden when device-identification is disabled.

For example, when an IoT application control signature is excluded from device detection (`exclude-signatures iot`), IoT devices are not detected.

When OT and IoT signatures are included in the policy interfaces, FortiGate automatically creates and applies a built-in application list to ensure relative IoT and OT device categories are active.

A firewall policy with `utm-status` enabled must be configured for the LAN interface for OT and IoT device detection to occur.

Virtual patching profiles

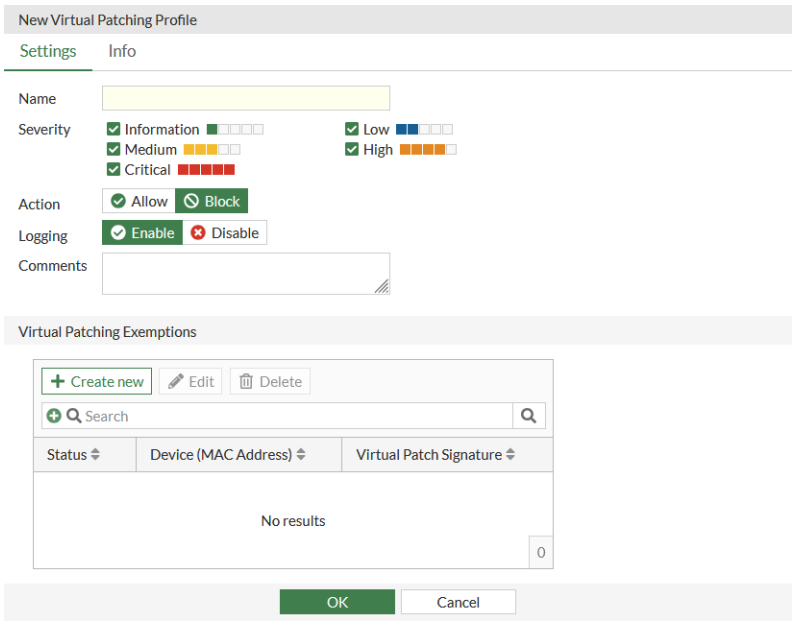
A virtual patching profile can be applied to firewall policies in any direction, protecting traffic from or to the vulnerable OT and IoT devices. Virtual patching profiles can also be combined with virtual patching on NAC policies, so that vulnerable OT and IoT devices are first assigned to a protected VLAN, and then firewall policies associated with the VLAN will apply the virtual patching profile. See OT and IoT virtual patching on NAC policies on page 2687 for more information.

The following are requirements for the virtual patching feature:

- Purchase the appropriate OT-related license (virtual patching only applies to OT devices). See License and entitlement information for more details.
- Enable device detection on the LAN interface.
- Configure a firewall policy with `utm-status` enabled in order for device detection to occur.

The following options can be configured in a virtual patching profile (see also OT virtual patching basic examples on page 2681):

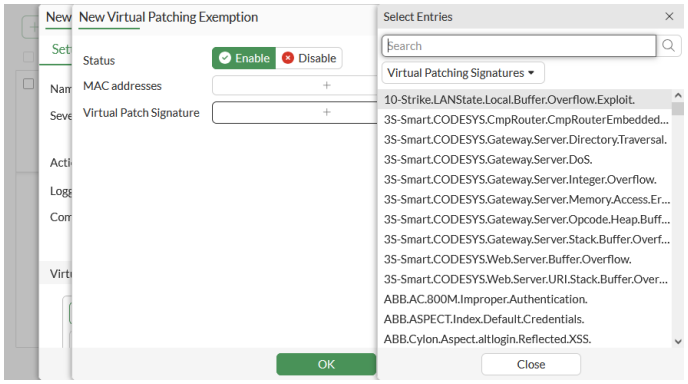
GUI option	CLI option	Description
Basic profile settings		
Name	`name <string>`	Enter a unique name for the profile.
Severity	`severity {info low medium high critical}`	Relative severity of the signature.
Action	`action {pass | block}`	Set the action to take for a matched device: • *Pass*/`pass`: allow sessions that match the profile. • *Block*/`block`: block sessions that match the profile (default).
Logging	`log {enable | disable}`	Enable/disable detection logging. This setting is enabled by default.
Comments	`comment <var-string>`	Enter a comment (optional).
Virtual patching exemptions settings		
Status	`status {enable | disable}`	Enable/disable the exemption.
MAC addresses	`device <mac_address1>, <mac_address2>, ...`	Select the device MAC addresses to exempt.
Virtual Patch Signature	`rule <id1>, <id2>, ...`	Select the signatures to exempt. See Virtual patching exemptions for more details.



To configure virtual patching in the GUI, ensure that *Virtual Patching* is enabled on the *System > Feature Visibility* page.

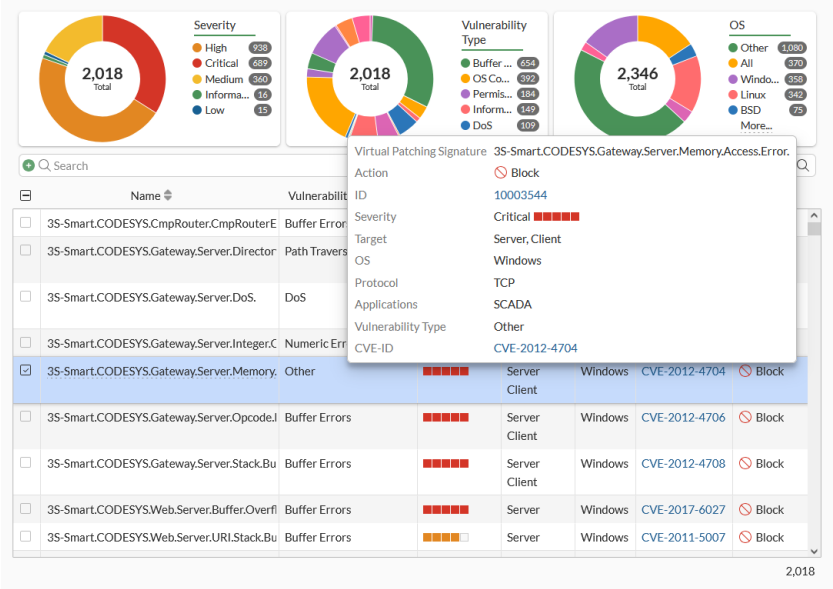
Virtual patching exemptions

Set the device MAC addresses and virtual patch signatures to exempt.

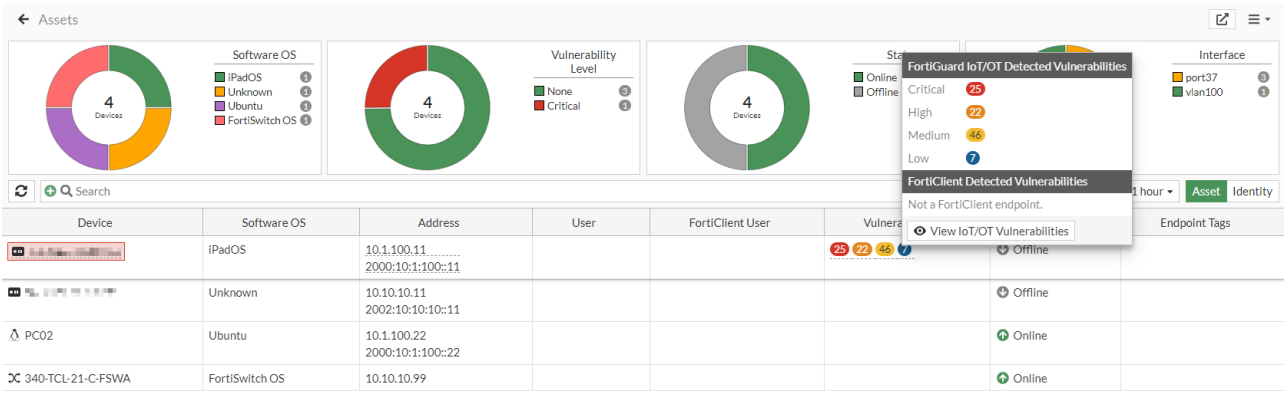


Virtual patching signatures

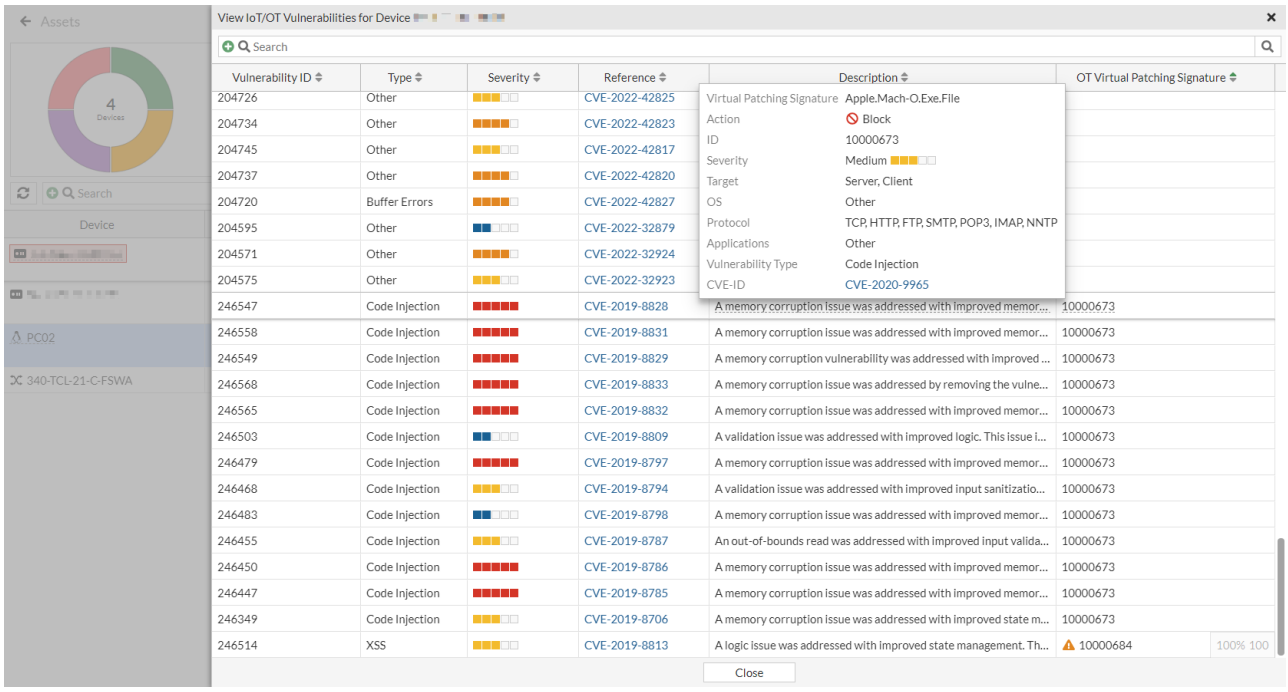
The *Security Profiles > Virtual Patching Signatures* page displays all OT virtual patching signatures. When using multi-VDOM mode, the OT virtual patching signatures are displayed per VDOM.



The *Dashboard* > *Assets & Identities* > *Assets* widget displays a tooltip for detected IoT and OT vulnerabilities when hovering over the *Vulnerabilities* column.



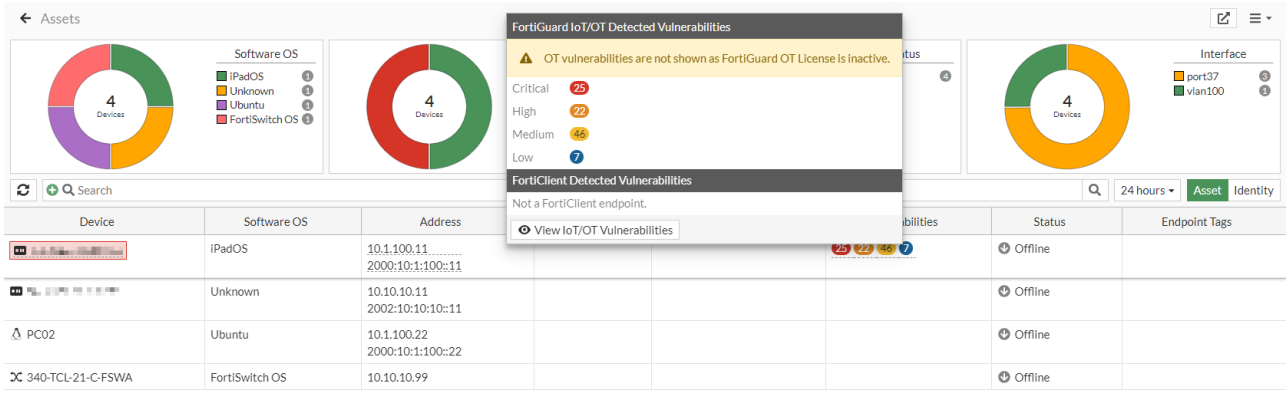
Clicking *View IoT/OT Vulnerabilities* in the tooltip displays a list of vulnerabilities retrieved from the FortiGuard API server for the device. The *OT Virtual Patching Signature* column includes the virtual patch signature ID that is mapped to the *Vulnerability ID*.



License and entitlement information

If a FortiGate does not have a valid OT license, a warning message is included in top of the IoT and OT vulnerabilities tooltip (*Assets* widget), indicating that OT vulnerabilities will not be detected.

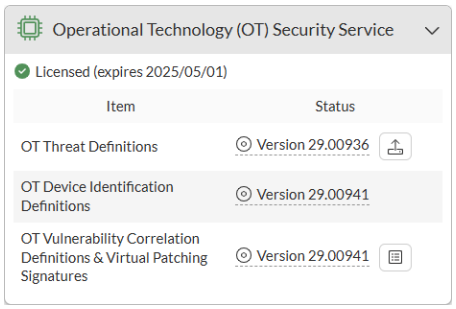
In a Security Fabric, each device must have a license.



The right-side gutter of virtual patching profile pages includes information about the following:

- *Operational Technology (OT) Security Service* entitlement status
- *OT Detection Definitions Package* version
- *OT Virtual Patching Signatures Package* version

The *System > FortiGuard > Subscriptions* tab also includes the list of signatures under the *Operational Technology (OT) Security Service* entitlement.

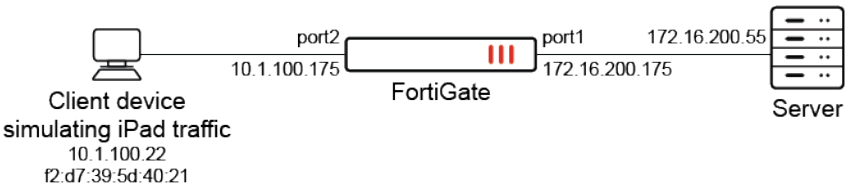


OT virtual patching basic examples

This topic contains two OT virtual patching examples: a basic configuration, and configuration that uses a NAC policy.

Example 1: basic configuration

This example demonstrates the flow for OT virtual patching from start to finish. First, a device (10.1.100.22) goes through device detection, which matches an OT detection signature downloaded on the FortiGate. Next, known vulnerabilities and OT patch signatures for this device are mapped to its MAC address. When traffic is generated by this device, IPS scans the traffic to identify any traffic patterns that match known OT patch signatures for this device. If a match is found, traffic is blocked by the FortiGate.



For demonstrative purposes, the simulated vulnerable OT device is a PC simulating web traffic from an iPad. An OT detection signature is specially crafted to match this Apple iPad traffic to the OT device category. To simulate vulnerable traffic, a test OT patch signature is used to match a generic cross-site scripting (XSS) attack over HTTP.

To verify the status of the OT related definitions:

1. Verify the current contracts licensed to the FortiGate:

```
# diagnose test update info
…
OTDT,Mon Sep 24 17:00:00 2029
OTVP,Mon Sep 24 17:00:00 2029
…
```

2. Verify the versions and status of the OT definitions:

```
# diagnose autoupdate versions
…
```

```
OT Detect Definitions
---------
Version: 23.00545 signed
Contract Expiry Date: Sun Sep 23 2029
Last Updated using manual update on Thu Jul 20 09:40:03 2023
Last Update Attempt: n/a
Result: Updates Installed
--
OT Patch Definitions
---------
Version: 23.00505 signed
Contract Expiry Date: Sun Sep 23 2029
Last Updated using manual update on Thu Jul 20 09:39:50 2023
Last Update Attempt: n/a
Result: Updates Installed
…
```

3. View the OT detection rules downloaded on the FortiGate. In this example, the OT detection rule ID 10000870 is a specially crafted signature to match Apple iPad traffic to the OT category:

```
# get rule otdt status
app-name: "Apple.iPad"
id: 10000870
category: "OT"
cat-id: 34
popularity: 5.low
risk: 1.medium
weight: 10
shaping: 0
protocol: 1.TCP, 9.HTTP
vendor: 7.Apple
technology: 0.Network-Protocol
behavior:
dev_cat: Other
```

4. View the OT patch rules downloaded on the FortiGate. In this example, the OT patch rule is a specially crafted signature to match a generic XSS attack to a vulnerability:

```
# get rule otvp status
rule-name: "WAP.Generic.XSS"
rule-id: 10000684
rev: 20.321
date: 1653379200
action: pass
status: enable
log: disable
log-packet: disable
severity: 2.medium
service: TCP, HTTP
location: server
os: Other
application: Other
```

```
rate-count: 0
rate-duration: 0
rate-track: none
rate-mode: continuous
vuln_type: XSS
cve: 20198625
```

To configure virtual patching in the GUI:

1. Enable device detection on port2 :
   a. Go to *Network* > *Interfaces* and edit port2.
   b. In the *Network* section, enable *Device detection*.
   c. Click *OK*.
2. Configure the virtual patching profile:
   a. Go to *Security Profiles* > *Virtual Patching* and click *Create New*.
   b. Configure the following settings:

Name	*test*
Severity	Select *Info*, *Low*, *Medium*, *High*, and *Critical*
Action	*Block*
Logging	*Enable*

   c. Click *OK*.
3. Apply the virtual patching profile to a firewall policy for traffic from port2 to port1:
   a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
   b. In the *Security Profiles* section, enable *Virtual Patching* and select the virtual patch profile (*test*).
   c. Set *SSL Inspection* to a profile that uses deep inspection profile in order to scan SSL encrypted traffic.
   d. Configure the other settings as needed.
   e. Click *OK*.

To configure virtual patching in the CLI:

1. Enable device detection on port2:

```
config system interface
    edit "port2"
        set device-identification enable
    next
end
```

2. Configure the virtual patching profile:

```
config virtual-patch profile
    edit "test"
        set comment ''
        set severity info low medium high critical
        set action block
```

```
        set log enable
    next
end
```

3. Apply the virtual patching profile to a firewall policy:

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "custom-deep-inspection"
        set virtual-patch-profile "default"
        set nat enable
    next
end
```

To test the virtual patching:

1. On the PC, generate traffic that simulates web traffic from an iPad. This traffic is generated in order for the FortiGate to perform device detection on port2. The OT detection signature 10000870 will be triggered, which considers this traffic from an OT device in this simulated scenario:

```
# curl 172.16.200.55 -H "User-Agent: Mozilla/5.0 (iPad; CPU OS 12_5_5 like Mac OS X)
AppleWebKit/605.1.15 (KHTML, like Gecko) Version/10.1.2 Mobile/15E148 Safari/604.1"
```

   A log is generated, indicating the traffic that triggered the match:

```
3: date=2023-07-24 time=15:31:26 eventtime=1690237885960202460 tz="-0700" logid="1059028704"
type="utm" subtype="app-ctrl" eventtype="signature" level="information" vd="root"
appid=10000870 srcip=10.1.100.22 srccountry="Reserved" dstip=172.16.200.55
dstcountry="Reserved" srcport=51548 dstport=80 srcintf="port2" srcintfrole="undefined"
dstintf="port1" dstintfrole="undefined" proto=6 service="HTTP" direction="outgoing" policyid=1
poluuid="a3424268-1ffc-51ed-3ba9-f3a60e2271cf" policytype="policy" sessionid=7284
applist="default" action="pass" appcat="OT" app="Apple.iPad" hostname="172.16.200.55"
incidentserialno=18882457 url="/" agent="Mozilla/5.0 (iPad; CPU OS 12_5_5 like Mac OS X)
AppleWebKit/605.1.15 (KHTML, like Gecko) Version/10.1.2 Mobile/15E148 Safari/604.1"
httpmethod="GET" msg="OT: Apple.iPad" clouddevice="Vendor=Apple, Product=ipados,
Version=12.5.5, Firmware=IOS" apprisk="low"
```

   The FortiGate queries the FortiGuard OT query service with information about the OT device vendor and product. The service responds with the vulnerabilities and patch_sign_id applicable to this device. IPS caches this information in its device vulnerability database.

2. Verify the vulnerability by device MAC and IP address:

```
# diagnose user-device-store device memory vulnerability-query f2:d7:39:5d:40:21 10.1.100.22
Got 28 vulnerabilities, response size:1792
```

```
[Vulnerability-0]
        'vulnerability_id' = '110977'
        'severity' = '2'
        'signature' = '10000684'
```

3. Verify the virtual patch signatures stored and enabled on the FortiGate:

```
# diagnose ips share list otvp_cfgcache
f2:d7:39:5d:40:21 1 10000684
```

4. Using the vulnerable device 10.1.100.22, generate vulnerable traffic to the destination server 172.16.200.55. The traffic from this IP and MAC address triggers OT patch signature 1000684 to match and is subsequently blocked by the firewall policy:

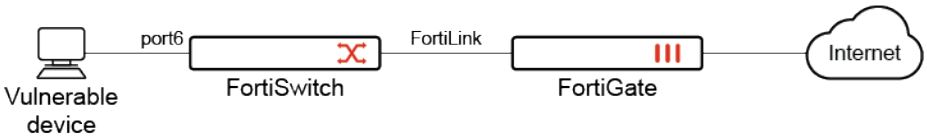
```
# curl -X POST http://172.16.200.55/'index.html?<javascript>'
```

5. Verify the UTM virtual patch log that was recorded with information about the vulnerability that was virtually patched:

```
# execute log filter category 24
# execute log display
2 logs found.
2 logs returned.

1: date=2023-07-20 time=16:03:00 eventtime=1689894179977743851 tz="-0700" logid="2400064600"
type="utm" subtype="virtual-patch" eventtype="virtual-patch" level="warning" vd="root"
count=medium srcip=10.1.100.22 profiletype="Reserved" dstip=172.16.200.55 direction="Reserved"
srcintfrole="port2" dstintf="undefined" dstintfrole="port1" sessionid=undefined
eventtype="12514" action="dropped" proto=6 service="HTTP" policyid=1 poluuid="a3424268-1ffc-
51ed-3ba9-f3a60e2271cf" policytype="policy" attack="WAP.Generic.XSS" srcport=47830 dstport=80
hostname="172.16.200.55" url="/index.html?<javascript>" agent="curl/7.61.1" httpmethod="POST"
direction="outgoing" attackid=10000684
```

Example 2: NAC policy

In this example, a NAC policy is pre-configured to detect devices with information or higher vulnerabilities, as demonstrated in OT and IoT virtual patching on NAC policies on page 2687. The NAC policy assigns the devices to vlan300.



A virtual patching profile is created to block any vulnerabilities with low, medium, high, or critical severity. The profile is applied to a firewall policy for outbound traffic.

To configure virtual patching in the GUI:

1. Enable device detection on vlan300:
   a. Go to *Network* > *Interfaces* and edit vlan300.
   b. In the *Network* section, enable *Device detection*.
   c. Click *OK*.
2. Configure the virtual patching profile:
   a. Go to *Security Profiles* > *Virtual Patching* and click *Create New*, or edit an existing profile.
   b. Configure the following settings:

Name	*OT_check*
Severity	Select *Info*, *Low*, *Medium*, *High*, and *Critical*
Action	*Block*
Logging	*Enable*

   c. Click *OK*.
3. Apply the virtual patching profile to a firewall policy:
   a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*, or edit an existing policy.
   b. In the *Security Profiles* section, enable *Virtual Patching* and select the virtual patch profile (*OT_check*).
   c. Configure the other settings as needed.
   d. Click *OK*.

To configure virtual patching in the CLI:

1. Enable device detection on vlan300:

```
config system interface
    edit "vlan300"
        set device-identification enable
    next
end
```

2. Configure the virtual patching profile:

```
config virtual-patch profile
    edit "OT_check"
        set severity info low medium high critical
    next
end
```

3. Apply the virtual patching profile to a firewall policy:

```
config firewall policy
    edit 1
        set name "virtualpatch-policy"
        set srcintf "vlan300"
        set dstintf "port1"
        set srcaddr "all"
```

```
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set virtual-patch-profile "OT_check"
        set logtraffic all
    next
end
```

4. Verify the logs:

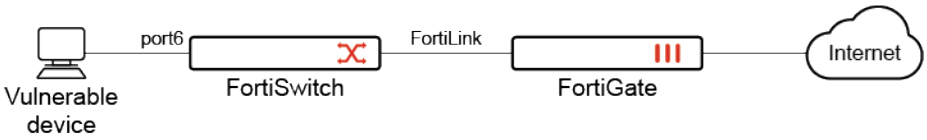
```
# execute log filter category utm-virtual-patch
# execute log display
...
1: date=2023-06-20 time=16:21:00 eventtime=1686180059982988434 tz="-0700" logid="2400064600"
type="utm" subtype="virtual-patch" eventtype="virtual-patch" level="warning" vd="root"
severity="medium" srcip=10.1.100.11 srccountry="Reserved" dstip=172.16.200.55
dstcountry="Reserved" srcintf="vlan300" srcintfrole="undefined" dstintf="port1"
dstintfrole="undefined" sessionid=1445 action="dropped" proto=6 service="HTTP" policyid=1
poluuid="ce6b724c-0558-51ee-e9d3-f0b8ef1c115f" policytype="policy" attack="WAP.Generic.XSS"
srcport=37062 dstport=80 hostname="172.16.200.55" url="/index.html?<javascript>"
agent="curl/7.61.1" httpmethod="POST" direction="outgoing" attackid=10000684
ref="http://www.fortinet.com/ids/VID10000684" incidentserialno=214959182 msg="vPatch:
WAP.Generic.XSS" crscore=10 craction=16384 crlevel="medium"
```

OT and IoT virtual patching on NAC policies

OT and IoT virtual patching can be applied to a NAC policy by setting the category to *Vulnerability* and configuring the *Match* criteria based on severity. Devices that match the criteria can be assigned and isolated to a NAC VLAN.

Example

In this example, a device with a certain vulnerability severity is detected by the NAC policy on the FortiGate. Subsequently, the FortiSwitch port in which it is connected to is moved to vlan300 where traffic can be controlled for vulnerable devices. For more information about NAC policies, see Defining a FortiSwitch NAC policy in the FortiLink Administration Guide. This example assumes the vlan300 has already been configured.



The following settings are required for IoT device detection:

- A valid Attack Surface Security Rating service license to download the IoT signature package.
- Enable device detection on the LAN interface used by IoT devices.

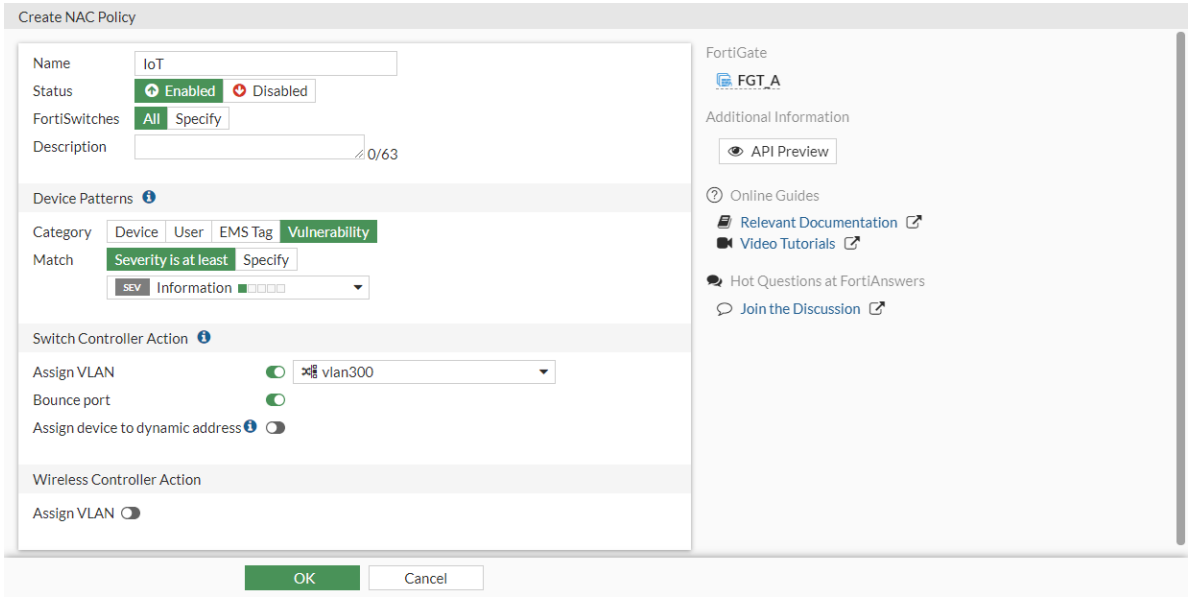
- In the GUI, go to *Network* > *Interfaces*, edit a LAN interface, enable *Device detection*, and click *OK*.
- In the CLI, enter:

```
config system interface
    edit <name>
        set device-identification enable
    next
end
```

- Configure a firewall policy with `utm-status` enabled in order for device detection to occur.

To configure virtual patching on NAC policies

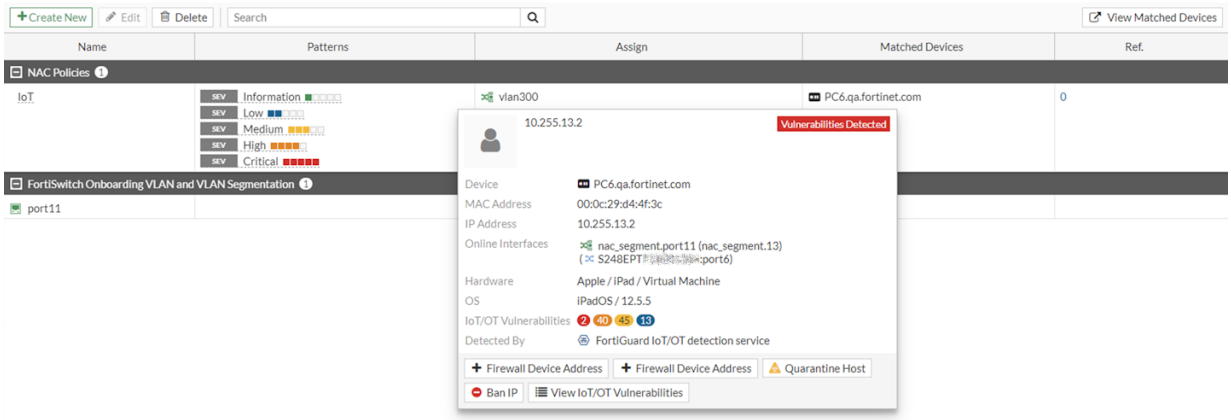
1. Configure the NAC policy:
   a. Go to *WiFi & Switch Controller* > *NAC Policies* and click *Create New*, or edit an existing policy.
   b. In the *Device Patterns* section, set *Category* to *Vulnerability*.
   c. Set *Match* to *Severity is at least* and select a severity level (*Information* is used in this example).
   d. In the *Switch Controller Action* section, enable *Assign VLAN* and select *vlan300*.



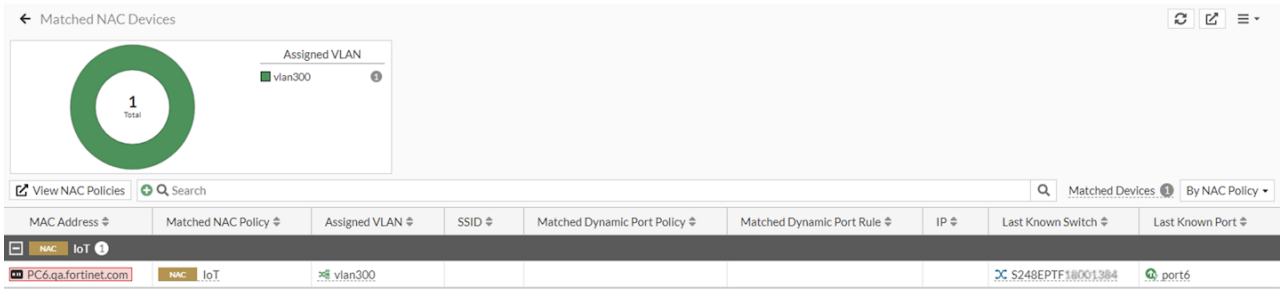
   e. Configure the other settings as needed.
   f. Click *OK*.
2. Enable NAC mode on the desired FortiSwitch ports (port6 in this example):
   a. Go to *WiFi & Switch Controller* > *FortiSwitch Ports*.
   b. Select *port6*, then right-click and set the *Mode* to *NAC*.
3. Enable utm-status on the firewall policy that is used to control outbound internet access for vulnerable devices (vlan300 to port1).

```
config firewall policy
    edit <name>
        set utm-status enable
    next
end
```

4. Generate traffic on the vulnerable client device.
5. Once the NAC policy is matched, go to *WiFi & Switch Controller > NAC Policies* to view the device matched to the policy.



The vulnerable device is also shown on *Dashboards > Assets & Identities* in the *Matched NAC Devices* widget.



To configure virtual patching on NAC policies in the CLI:

1. Configure the VLAN in the MAC policy:

```
config switch-controller mac-policy
    edit "IoT"
        set fortilink "fortilink"
        set vlan "vlan300"
    next
end
```

2. Configure the NAC policy:

```
config user nac-policy
    edit "IoT"
        set category vulnerability
        set severity 0 1 2 3 4
        set switch-fortilink "fortilink"
        set switch-mac-policy "IoT"
    next
end
```

3. Enable NAC mode on the desired FortiSwitch ports:

```
config switch-controller managed-switch
    edit "S248E**********"
        config ports
            edit "port6"
                set access-mode nac
            next
        end
    next
end
```

4. Configure a firewall policy with `utm-status` enabled to limit access for devices in this VLAN (vlan300).

SSL & SSH Inspection

Secure Sockets Layer (SSL) content scanning and inspection allows you to apply antivirus scanning, web filtering, and email filtering to encrypted traffic. You can apply SSL inspection profiles to firewall policies.

FortiOS includes four preloaded SSL/SSH inspection profiles, three of which are read-only and can be cloned:

- *certificate-inspection*
- *deep-inspection*
- *no-inspection*

The *custom-deep-inspection* profile can be edited, or you can create your own SSL/SSH inspection profiles.

Deep inspection (also known as SSL/SSH inspection) is typically applied to outbound policies where destinations are unknown. Depending on your policy requirements, you can configure the following:

- Which CA certificate will be used to decrypt the SSL encrypted traffic
- Which SSL protocols will be inspected
- Which ports will be associated with which SSL protocols for inspection
- Whether or not to allow invalid SSL certificates
- Whether or not SSH traffic will be inspected
- Which addresses or web category allowlists can bypass SSL inspection

The following topics provide information about SSL & SSH Inspection:

- Configuring an SSL/SSH inspection profile on page 2691
- Certificate inspection on page 2694
- Deep inspection on page 2697
- Protecting an SSL server on page 2700
- Handling SSL offloaded traffic from an external decryption device on page 2701
- SSH traffic file scanning on page 2703
- Redirect to WAD after handshake completion on page 2706
- HTTP/2 support in proxy mode SSL inspection on page 2706
- Define multiple certificates in an SSL profile in replace mode on page 2708
- Disabling the FortiGuard IP address rating on page 2710
- Block or allow ECH TLS connections on page 2711

- Configuring certificate probe failure option on page 2720
- Hybrid PQC support in proxy mode SSL deep inspection on page 2721

Configuring an SSL/SSH inspection profile

The *custom-deep-inspection* profile can be edited or new SSL/SSH inspection profiles can be configured to be used in firewall policies.

To configure an SSL/SSH inspection profile in the GUI:

1. Go to *Security Profiles* > *SSL/SSH Inspection* and click *Create New*.
2. Configure the following settings:

Name		Enter a unique name for the profile.
Comments		Enter a comment (optional).
SSL Inspection Options		
	Enable SSL Inspection of	Enable SSL inspection of: • *Multiple Clients Connecting to Multiple Servers*: Use this option for generic policies where the destination is unknown. This is normally used when inspecting outbound internet traffic. Other *SSL Inspection Options* become available to configure if this option is selected. • *Protecting SSL Server*: Use this option when setting up a profile customized for a specific SSL server with a specific certificate. Define the certificate using the *Server certificate* field. See Protecting an SSL server on page 2700 for more information.
	Inspection method	Define the inspection method: • *SSL Certificate Inspection*: Only inspects the certificate, by way of the headers up to the SSL/TLS layer, and not the contents of the traffic. Allows Encrypted Client Hello (ECH) to be allowed or blocked. • *Full SSL Inspection*: Inspects the SSL/TLS encrypted traffic payload. See Deep inspection on page 2697.
	CA certificate	Use the dropdown menu to select one of the installed certificates for the inspection of the packets. Click *Download* to save the certificate.
	Blocked certificates	Block or allow potentially malicious certificates. Select *View Blocked Certificates* for a detailed list of blocked certificates, including the listing reason and date.
	Untrusted SSL certificates	Configure the action to take when a server certificate is not issued by a trusted CA. • *Allow*: Allow the untrusted server certificate. This is the default value. • *Block*: Block the session.

		• *Ignore*: This option is for *Full SSL inspection* only. It re-signs the server certificate as trusted. When configured in the GUI for certificate inspection it has no effect and the setting is not saved. Click *View Trusted CAs List* to see a list of the factory bundled and user imported CAs that are trusted by the FortiGate.
	Server certificate SNI check	Check the SNI in the hello message with the CN or SAN field in the returned server certificate: • *Enable*: If it is mismatched, use the CN in the server certificate for URL filtering. • *Strict*: If it is mismatched, close the connection. • *Disable*: Server certificate SNI check is disabled.
	Enforce SSL cipher compliance	Enable/disable SSL cipher compliance. This option is for *Full SSL inspection* only.
	Enforce SSL negotiation compliance	Enable/disable SSL negotiation compliance. This option is for *Full SSL inspection* only.
	RPC over HTTPS	Enable/disable inspection of Remote Procedure Calls (RPC) over HTTPS traffic. This option is for *Full SSL inspection* only.
	MAPI over HTTPS	Enable/disable inspection of Messaging Application Programming Interface (MAPI) over HTTPS traffic.
Protocol Port Mapping		
	Inspect all ports	Inspect all ports with the IPS engine by enabling *Inspect all ports*. Choose whether to disable *DNS over TLS*.
	HTTPS, SMTPS, POP3S, IMAPS, FTPS	If *Inspect all ports* is disabled, specify the port through which traffic will be inspected in the field next to the listed protocols. See Inspecting all ports on page 2694 for details.
	DNS over TLS	Enable/disable inspection of DNS over TLS.
	Encrypted Client Hello	Allow or block TLS connections that use Encrypted Client Hello (ECH).
	HTTP/3	When *Inspection method* is set to *SSL Certificate Inspection* or *HTTPS* is disabled, *HTTP/3* is set to *Bypass* and cannot be changed. When *Inspection method* is set to *Full SSL Inspection*, choose whether to *Inspect*, *Bypass*, or *Block* HTTP/3.
	DNS over QUIC	Available when *Inspection method* is set to *Full SSL Inspection*. Choose whether to *Inspect*, *Bypass*, or *Block* DNS over QUIC.
Exempt from SSL Inspection		These options are for *Full SSL inspection* only. Use the menus in this section to specify any reputable websites, FortiGuard Web Categories, or addresses that will be exempt from SSL inspection: • *Reputable Websites*: Enable this option to exempt any websites identified by FortiGuard as reputable.

	• *Web Categories*: The categories of *Finance and Banking*, *Health and Wellness*, and *Personal Privacy* have been added by default. These categories are the most likely to have applications that will require a specific certificate. • *Addresses*: These can be any of the address objects that have an interface of *any*. • *Log SSL exemptions*: Enable this option to log all SSL exemptions. See Exempt web sites from deep inspection on page 2699 for more information.
SSH Inspection Options	
SSH deep scan	Enable/disable SSH protocol packet deep scanning capabilities. *SSH port* will become available if *SSH deep scan* is enabled.
SSH port	Define what ports will search for SSH protocol packets: • *Any*: Select this option to search all traffic regardless of service or TCP/IP port for packets that conform to the SSH protocol. • *Specify*: Select this option and enter the port number to restrict the search for SSH protocol packets to the TCP/IP port number specified. This is not as comprehensive but it is easier on the performance of the firewall.
Common Options	
Invalid SSL certificates	Allow or block the passing of traffic in invalid certificates. Additional common options that provide more granularity with actions for different types of invalid SSL certificates will become available if *Invalid SSL certificates* is set to *Custom*: • *Expired certificates*: Action to take when the server certificate is expired. The default action is block. • *Revoked certificates*: Action to take when the server certificate is revoked. The default action is block. • *Validation timed-out certificates*: Action to take when the server certificate validation times out. For certificate inspection, the default action is allow. For deep inspection, the default action is Keep Untrusted & Allow. • *Validation failed certificates*: Action to take when the server certificate validation fails. The default action is block. For deep inspection, the above options have the following actions: • *Keep Untrusted & Allow*: Allow the server certificate and keep it untrusted. • *Block*: Block the certificate. • *Trust & Allow*: Allow the server certificate and re-sign it as trusted.
Log SSL anomalies	Enable this feature to record and log traffic sessions containing invalid certificates. By default, SSL anomalies logging is enabled. Logs are generated in the UTM log type under the SSL subtype when invalid certificates are detected.

3. Click *OK*.

Inspecting all ports

The behavior of inspecting all ports can be different between flow and proxy mode inspection when the inspection mode is configured in a firewall policy.

In proxy mode inspection, when deep inspection is enabled:

- If *Inspect all ports* is disabled, only the ports specified in the *Protocol Port Mapping* section will be scanned.
- If *Inspect all ports* is enabled, all ports will be scanned.

In flow mode inspection, when deep-inspection is enabled:

- All ports will be scanned, regardless of whether or not *Inspect all ports* is enabled or disabled.

When performing certificate inspection instead of deep inspection, flow and proxy mode inspection behave the same:

- If *Inspect all ports* is disabled, only the ports specified in the *Protocol Port Mapping* section will be scanned.
- If *Inspect all ports* is enabled, all ports will be scanned.

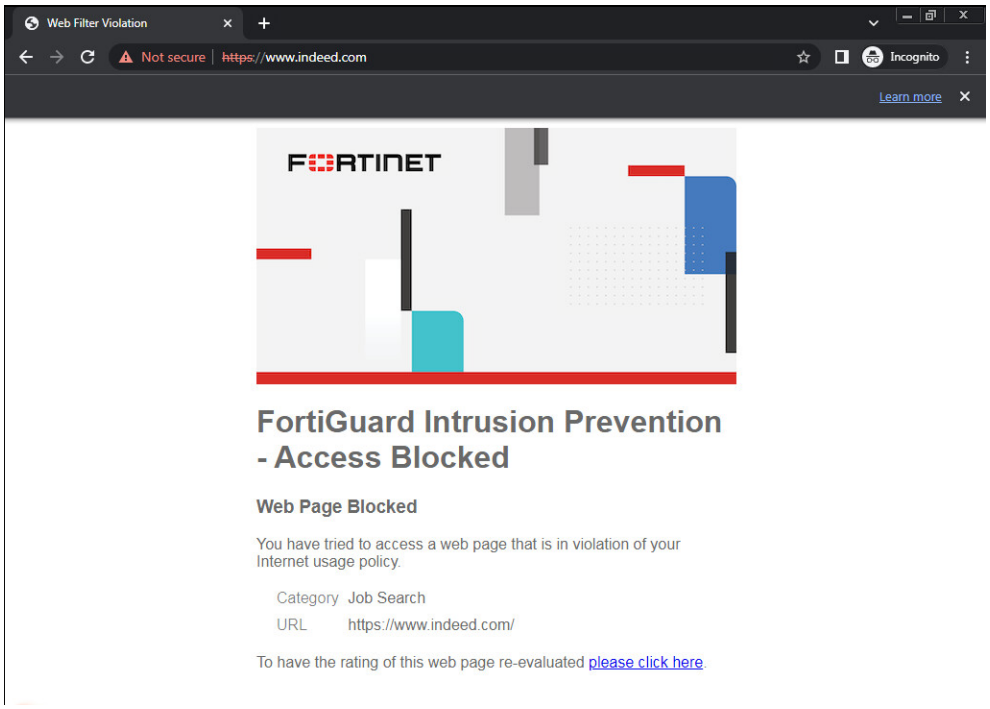
Certificate inspection

FortiGate supports certificate inspection. The default configuration has a built-in *certificate-inspection* profile which you can use directly. When you use certificate inspection, the FortiGate only inspects the headers up to the SSL/TLS layer.

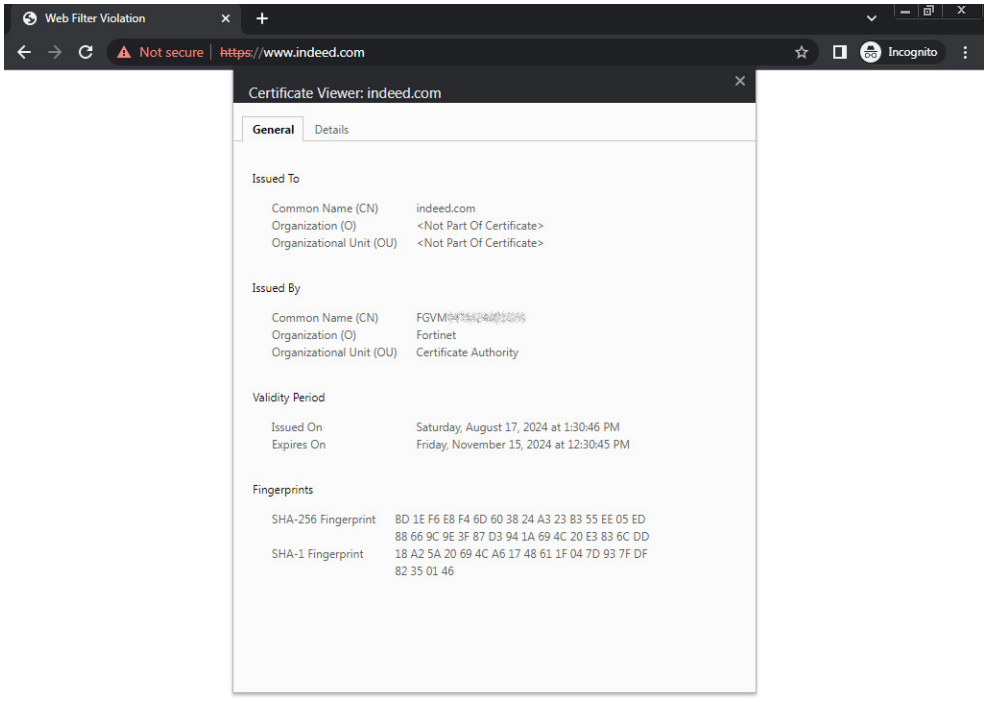
If you do not want to deep scan for privacy reasons but you want to control web site access, you can use *certificate-inspection*.

Replacement messages for HTTPS connections

When using certificate inspection in a firewall policy, a user will encounter a block page served in HTTPS when accessing an HTTPS web page that triggers a replacement message.



The replacement message can display for many reasons, such as a page being blocked by a web filter category, a page displaying a warning for a web filter category, a page requiring authentication, and so on. If you examine the certificate, it shows that the replacement page is signed by the FortiGate CA.



For your clients to no longer see a warning message for an invalid certificate, install the respective CA certificate in the client's certificate store.

To download the CA certificate:

1. Go to *Security Profiles* > *SSL/SSH Inspection*.
2. Edit the SSL/SSH inspection profile that is being used in the firewall policy.
3. Beside the *CA Certificate* field, click *Download*.
4. Share and install this certificate on the client endpoints devices.

By default, the SSL/SSH inspection profile uses the *Fortinet_CA_SSL* certificate. You can customize this certificate by changing the selection in the *CA Certificate* field to another certificate in the FortiGate's certificate store.

Inspect non-standard HTTPS ports

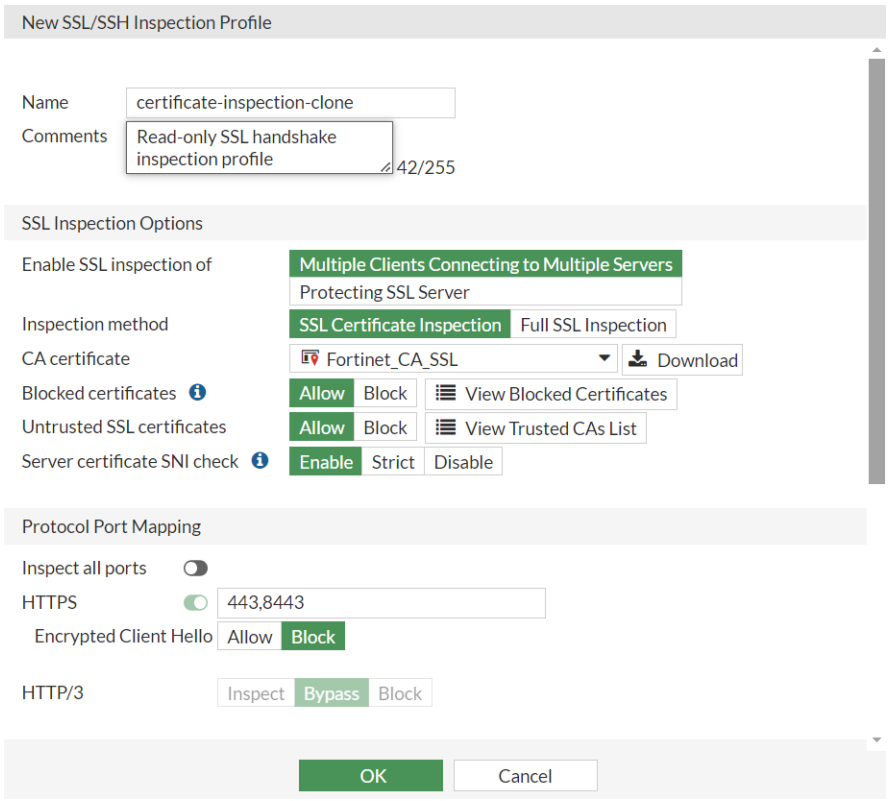
The built-in *certificate-inspection* profile is read-only and only listens on port 443. If you want to make changes, you must create a new certificate inspection profile.

If you know the non-standard port that the web server uses, such as port 8443, you can add this port to the *HTTPS* field.

To add a port to the inspection profile in the GUI:

1. Go to *Security Profiles* > *SSL/SSH Inspection*.
2. Create a new profile, or clone the default profile.
3. If you do no know what port is used in the HTTPS web server, under *Protocol Port Mapping*enable *Inspect All Ports*.

   If you know the port, such as port 8443, then set *HTTPS* to *443,8443*.



4. Configure the remaining setting as needed.
5. Click *OK*.

Common options

Invalid SSL certificates can be blocked, allowed, or a different actions can be configured for the different invalid certificates types. See Configuring an SSL/SSH inspection profile on page 2691.

Deep inspection

You can configure address and web category allowlists to bypass SSL deep inspection.

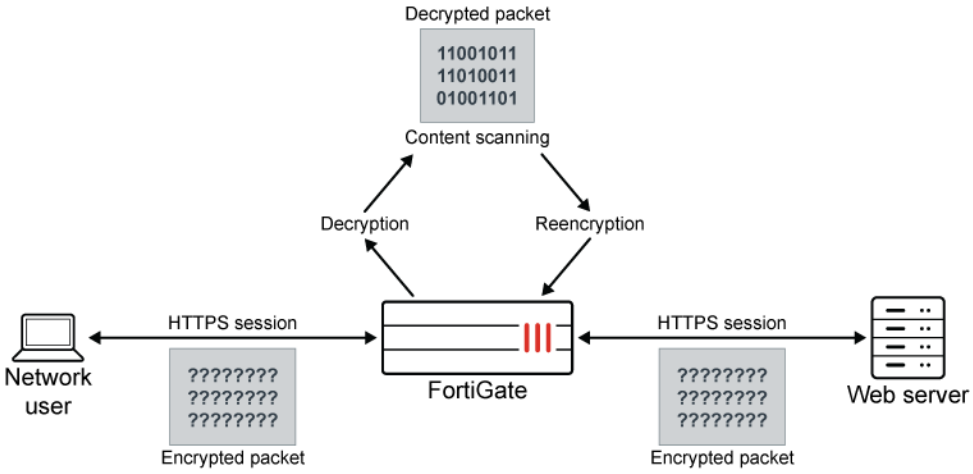
Reasons for using deep inspection

While Hypertext Transfer Protocol Secure (HTTPS) offers protection on the Internet by applying Secure Sockets Layer (SSL) encryption to web traffic, encrypted traffic can be used to get around your network's normal defenses.

For example, you might download a file containing a virus during an e-commerce session, or you might receive a phishing email containing a seemingly harmless download that, when launched, creates an encrypted session to a command and control (C&C) server and downloads malware onto your computer. Because the sessions in these attacks are encrypted, they might get past your network's security measures.

When you use deep inspection, the FortiGate impersonates the recipient of the originating SSL session, then decrypts and inspects the content to find threats and block them. It then re-encrypts the content and sends it to the real recipient.

Deep inspection not only protects you from attacks that use HTTPS, it also protects you from other commonly-used SSL-encrypted protocols such as SMTPS, POP3S, IMAPS, and FTPS.



Protocol port mapping

To optimize the FortiGate's resources, the mapping and inspection of the following protocols can be enabled or disabled:

- HTTPS
- SMTPS
- POP3S
- IMAPS
- FTPS
- DNS over TLS

Each protocol has a default TCP port. The ports can be modified to inspect any port with flowing traffic. The packet headers indicate which protocol generated the packet.

Protocol port mapping only works with proxy-based inspection. Flow-based inspection inspects all ports regardless of the protocol port mapping configuration.

Browser messages when using deep inspection

When the FortiGate re-encrypts the content, it uses a stored certificate, such as *Fortinet_CA_SSL*, *Fortinet_CA_Untrusted*, or your own CA certificate that you uploaded.

Because there is no *Fortinet_CA_SSL* in the browser trusted CA list, the browser displays an untrusted certificate warning when it receives a FortiGate re-signed server certificate. To stop the warning messages, trust the FortiGate-trusted CA *Fortinet_CA_SSL* and import it into your browser.

If you still get messages about untrusted certificates after importing *Fortinet_CA_SSL* into your browser, it is due to *Fortinet_CA_Untrusted*. Never import the *Fortinet_CA_Untrusted* certificate into your browser.

To import *Fortinet_CA_SSL* into your browser:

1. On the FortiGate, go to *Security Profiles* > *SSL/SSH Inspection* and edit the *deep-inspection* profile.
   The default *CA Certificate* is *Fortinet_CA_SSL*.
2. Click *Download* and save the certificate to the management computer.
3. On the client PC, use the *Certificate Import Wizard* to install the certificate into the *Trusted Root Certificate Authorities* store.
   If a security warning appears, select *Yes* to install the certificate.

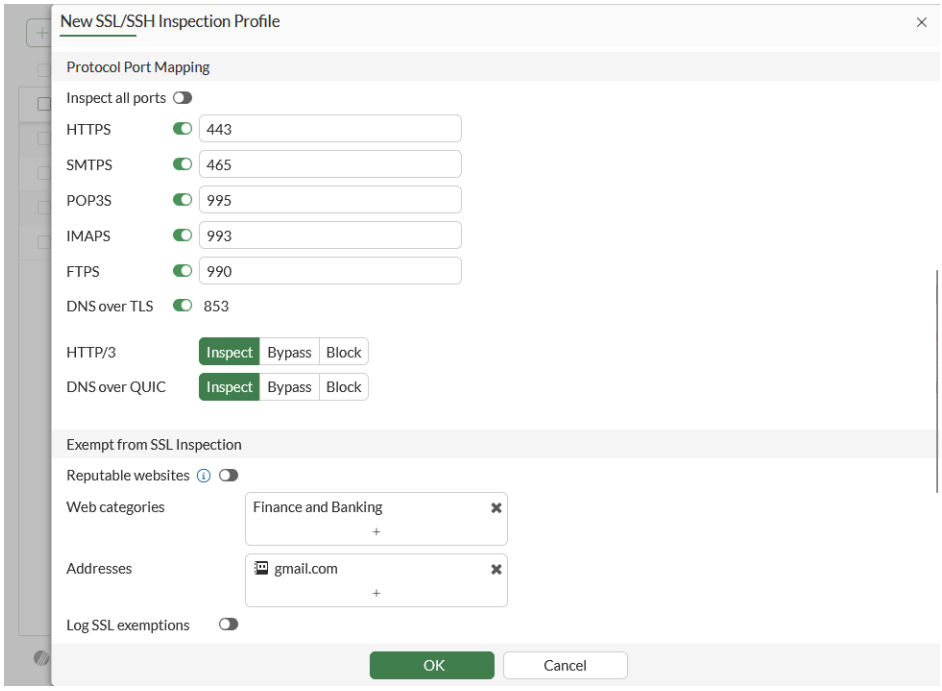
> You can upload and distribute CA certificates using a group policy on multiple Windows devices that are part of an Active Directory. See Distribute certificates to Windows devices by using Group Policy.

Exempt web sites from deep inspection

If you do not want to apply deep inspection for privacy or other reasons, you can exempt the session by address, category, or allowlist.

If you know the address of the server you want to exempt, you can exempt that address. You can exempt specific address type including IP address, IP address range, IP subnet, FQDN, wildcard-FQDN, and geography.

If you want to exempt all bank web sites, an easy way is to exempt the *Finance and Banking* category, which includes all finance and bank web sites identified in FortiGuard. For information about creating and using custom local and remote categories, see Web rating override on page 2736 and External feeds on page 4695.



If you want to exempt commonly trusted web sites, you can bypass the SSL allowlist in the SSL/SSH profile by enabling *Reputable websites*. The allowlist includes common web sites trusted by FortiGuard.

SSL version support

There are two ways to limit which SSL versions deep inspection is applied to.

- In the global attributes:

```
config system global
    set strong-crypto enable
end
```

- In the protocol configuration of a deep inspection profile:

```
config firewall ssl-ssh-profile
    edit <name>
        config {ssl | https | ftps}
            set min-allowed-ssl-version {ssl-3.0 | tls-1.0 | tls-1.1 | tls-1.2 | tls-1.3}
        end
    next
end
```

Enabling `strong-crypto` in the global attributes sets the `min-allowed-ssl-version` to `tls-1.1` by default.

When a session is attempted using an SSL version below the minimum allowed version, the session can be blocked (default) or allowed.

The setting `min-allowed-ssl-version` is not available when the inspection status in the profile is `certificate-inspection`. When performing certificate inspection, the proxy is unable to negotiate with the server on the version of SSL/TLS to use.

To configure the action based on the SSL version used being unsupported:

```
config firewall ssl-ssh-profile
    edit <name>
        config {ssl | https | ftps | imaps | pop3s | smtps | dot}
            set unsupported-ssl-version {allow | block}
        end
    next
end
```

HTTPS and DOT options can enable QUIC inspection using the `set quic inspect` command. See DNS over QUIC and DNS over HTTP3 for transparent and local-in DNS modes on page 2416 for more information.

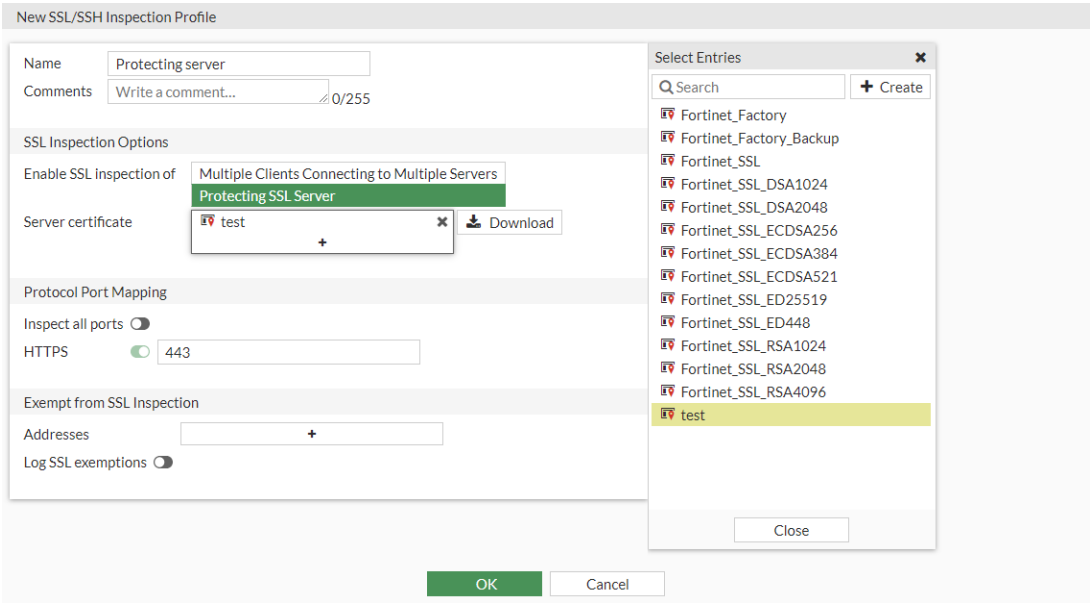
Protecting an SSL server

The *Protecting SSL Server* option of the SSL/SSH Inspection profile is typically applied to an inbound firewall policy for clients on the internet that access a server behind the FortiGate. FortiGate uses the server certificate of the protected server to simulate the real server, which enables FortiGate to decrypt and inspect traffic

destined to the real server. Therefore, a valid server certificate must be installed on the FortiGate to enable traffic inspection.

To upload a server certificate into FortiGate and use that certificate in the SSL/SSH inspection profile:

1. Go to *System* > *Certificates*.
2. Select *Import* > *Local Certificate* and upload the certificate.
3. Go to *Security Profiles* > *SSL/SSH Inspection* and edit or create a new profile.
4. For *Enable SSL Inspection of*, select *Protecting SSL Server*.
5. For *Server Certificate*, click the *+* and select the local certificate you imported.



6. Click *OK*.

When you apply the *Protecting SSL Server* profile in a policy, the FortiGate will send the server certificate to the client as your server does.

Handling SSL offloaded traffic from an external decryption device

In scenarios where the FortiGate is sandwiched between load-balancers and SSL processing is offloaded on the external load-balancers, the FortiGate can perform scanning on the unencrypted traffic by specifying the `ssl-offloaded` option in firewall `profile-protocol-options`. This option is supported in proxy and flow mode (previous versions only supported proxy mode).

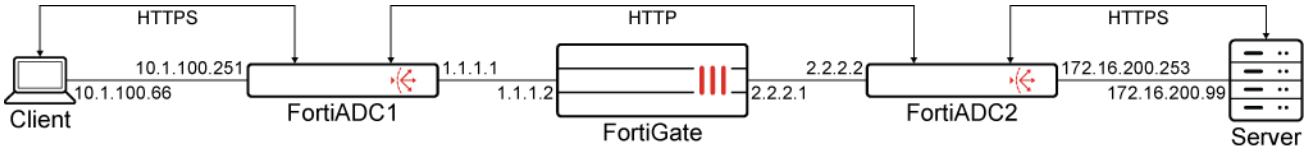
If the FortiGate receives an AUTH TLS, PBSZ, or PROT command before receiving plain text traffic from a decrypted device, by default, it will expect encrypted traffic, determine that the traffic belongs to an abnormal protocol, and bypass the traffic.

When the `ssl-offloaded` command is enabled, the AUTH TLS command is ignored, and the traffic is treated as plain text rather than encrypted data. SSL decryption and encryption are performed by the external device.

Sample topology

In this example, the FortiGate is between two FortiADCs and in SSL offload sandwich mode. The FortiGate receives plain text from ADC1 and forwards plain text to ADC2. There is no encrypted traffic passing through the FortiGate.

The client sends HTTPS traffic to ADC1, which then decrypts the traffic and sends HTTP to the FortiGate. The FortiGate forwards HTTP to ADC2, and the ADC2 re-encrypts the traffic to HTTPS.

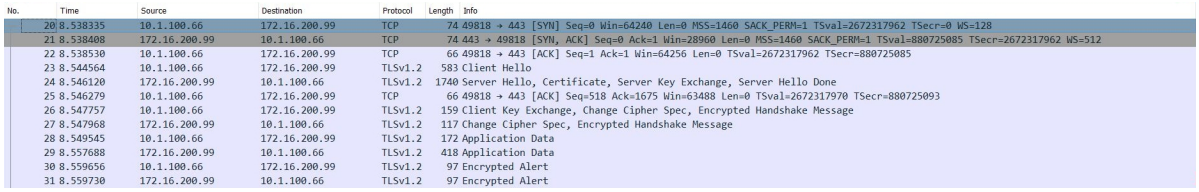


To configure SSL offloading:

```
config firewall profile-protocol-options
    edit "default-clone"
        config http
            set ports 80
            unset options
            unset post-lang
            set ssl-offloaded yes
        end
        config ftp
            set ports 21
            set options splice
            set ssl-offloaded yes
        end
        config imap
            set ports 143
            set options fragmail
            set ssl-offloaded yes
        end
        config pop3
            set ports 110
            set options fragmail
            set ssl-offloaded yes
        end
        config smtp
            set ports 25
            set options fragmail splice
            set ssl-offloaded yes
        end
    next
end
```

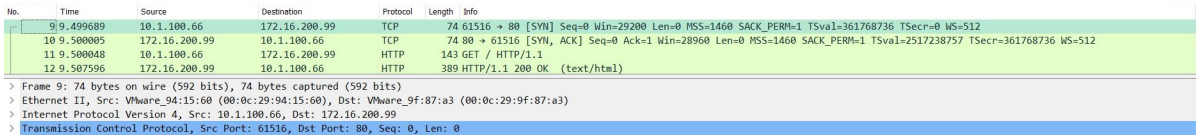
Verifying the packet captures

The ADC1 incoming port capture shows that ADC1 receives HTTPS traffic:



No.	Time	Source	Destination	Protocol	Length	Info
20	8.538335	10.1.100.66	172.16.200.99	TCP	74	49818 → 443 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=2672317962 TSecr=0 WS=128
21	8.538408	172.16.200.99	10.1.100.66	TCP	74	443 → 49818 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=880725085 TSecr=2672317962 WS=512
22	8.538530	10.1.100.66	172.16.200.99	TCP	66	49818 → 443 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=2672317962 TSecr=880725085
23	8.544564	10.1.100.66	172.16.200.99	TLSv1.2	583	Client Hello
24	8.546120	172.16.200.99	10.1.100.66	TLSv1.2	1740	Server Hello, Certificate, Server Key Exchange, Server Hello Done
25	8.546279	10.1.100.66	172.16.200.99	TCP	66	49818 → 443 [ACK] Seq=518 Ack=1675 Win=63488 Len=0 TSval=2672317970 TSecr=880725093
26	8.547757	10.1.100.66	172.16.200.99	TLSv1.2	159	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
27	8.547968	172.16.200.99	10.1.100.66	TLSv1.2	117	Change Cipher Spec, Encrypted Handshake Message
28	8.549545	10.1.100.66	172.16.200.99	TLSv1.2	172	Application Data
29	8.557688	172.16.200.99	10.1.100.66	TLSv1.2	418	Application Data
30	8.559656	10.1.100.66	172.16.200.99	TLSv1.2	97	Encrypted Alert
31	8.559730	172.16.200.99	10.1.100.66	TLSv1.2	97	Encrypted Alert

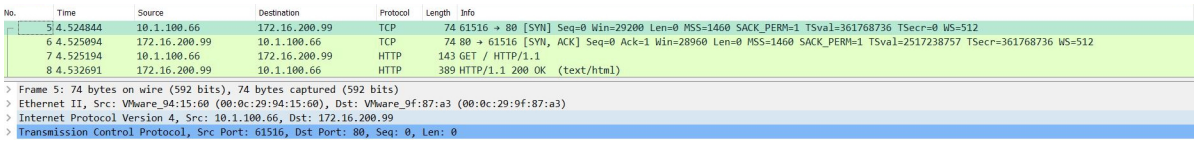
The ADC1 outgoing port capture shows that ADC1 decrypts traffic and forwards HTTP traffic to the FortiGate:



No.	Time	Source	Destination	Protocol	Length	Info
9	9.499689	10.1.100.66	172.16.200.99	TCP	74	61516 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=361768736 TSecr=0 WS=512
10	9.500005	172.16.200.99	10.1.100.66	TCP	74	80 → 61516 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=2517238757 TSecr=361768736 WS=512
11	9.500048	10.1.100.66	172.16.200.99	HTTP	143	GET / HTTP/1.1
12	9.507596	172.16.200.99	10.1.100.66	HTTP	389	HTTP/1.1 200 OK (text/html)

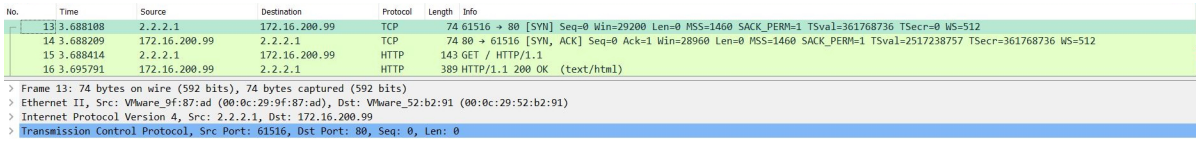
```
> Frame 9: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
> Ethernet II, Src: VMware_94:15:60 (00:0c:29:94:15:60), Dst: VMware_9f:87:a3 (00:0c:29:9f:87:a3)
> Internet Protocol Version 4, Src: 10.1.100.66, Dst: 172.16.200.99
> Transmission Control Protocol, Src Port: 61516, Dst Port: 80, Seq: 0, Len: 0
```

The FortiGate's incoming and outgoing port captures show that HTTP traffic passes through the FortiGate:



No.	Time	Source	Destination	Protocol	Length	Info
5	4.524844	10.1.100.66	172.16.200.99	TCP	74	61516 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=361768736 TSecr=0 WS=512
6	4.525094	172.16.200.99	10.1.100.66	TCP	74	80 → 61516 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=2517238757 TSecr=361768736 WS=512
7	4.525194	10.1.100.66	172.16.200.99	HTTP	143	GET / HTTP/1.1
8	4.532691	172.16.200.99	10.1.100.66	HTTP	389	HTTP/1.1 200 OK (text/html)

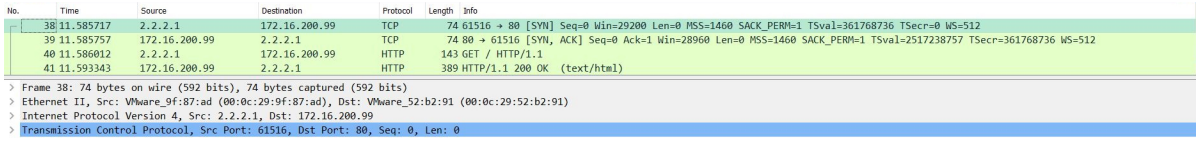
```
> Frame 5: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
> Ethernet II, Src: VMware_94:15:60 (00:0c:29:94:15:60), Dst: VMware_9f:87:a3 (00:0c:29:9f:87:a3)
> Internet Protocol Version 4, Src: 10.1.100.66, Dst: 172.16.200.99
> Transmission Control Protocol, Src Port: 61516, Dst Port: 80, Seq: 0, Len: 0
```



No.	Time	Source	Destination	Protocol	Length	Info
13	3.688108	2.2.2.1	172.16.200.99	TCP	74	61516 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=361768736 TSecr=0 WS=512
14	3.688209	172.16.200.99	2.2.2.1	TCP	74	80 → 61516 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=2517238757 TSecr=361768736 WS=512
15	3.688414	2.2.2.1	172.16.200.99	HTTP	143	GET / HTTP/1.1
16	3.695791	172.16.200.99	2.2.2.1	HTTP	389	HTTP/1.1 200 OK (text/html)

```
> Frame 13: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
> Ethernet II, Src: VMware_9f:87:ad (00:0c:29:9f:87:ad), Dst: VMware_52:b2:91 (00:0c:29:52:b2:91)
> Internet Protocol Version 4, Src: 2.2.2.1, Dst: 172.16.200.99
> Transmission Control Protocol, Src Port: 61516, Dst Port: 80, Seq: 0, Len: 0
```

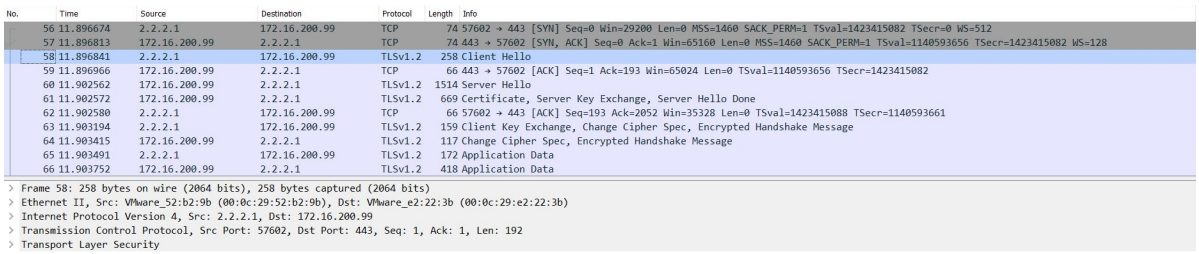
The ADC2 incoming port capture shows that the ADC2 receives HTTP traffic:



No.	Time	Source	Destination	Protocol	Length	Info
38	11.585717	2.2.2.1	172.16.200.99	TCP	74	61516 → 80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=361768736 TSecr=0 WS=512
39	11.585757	172.16.200.99	2.2.2.1	TCP	74	80 → 61516 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=2517238757 TSecr=361768736 WS=512
40	11.586012	2.2.2.1	172.16.200.99	HTTP	143	GET / HTTP/1.1
41	11.593343	172.16.200.99	2.2.2.1	HTTP	389	HTTP/1.1 200 OK (text/html)

```
> Frame 38: 74 bytes on wire (592 bits), 74 bytes captured (592 bits)
> Ethernet II, Src: VMware_9f:87:ad (00:0c:29:9f:87:ad), Dst: VMware_52:b2:91 (00:0c:29:52:b2:91)
> Internet Protocol Version 4, Src: 2.2.2.1, Dst: 172.16.200.99
> Transmission Control Protocol, Src Port: 61516, Dst Port: 80, Seq: 0, Len: 0
```

The ADC2 outgoing port capture shows that ADC2 forwards HTTPS traffic to the server:



No.	Time	Source	Destination	Protocol	Length	Info
56	11.896674	2.2.2.1	172.16.200.99	TCP	74	57602 → 443 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1423415082 TSecr=0 WS=512
57	11.896813	172.16.200.99	2.2.2.1	TCP	74	443 → 57602 [SYN, ACK] Seq=0 Ack=1 Win=65160 Len=0 MSS=1460 SACK_PERM=1 TSval=1140593656 TSecr=1423415082 WS=128
58	11.896841	2.2.2.1	172.16.200.99	TLSv1.2	258	Client Hello
59	11.896966	172.16.200.99	2.2.2.1	TCP	66	443 → 57602 [ACK] Seq=1 Ack=193 Win=65024 Len=0 TSval=1140593656 TSecr=1423415082
60	11.902562	172.16.200.99	2.2.2.1	TLSv1.2	1514	Server Hello
61	11.902572	172.16.200.99	2.2.2.1	TLSv1.2	669	Certificate, Server Key Exchange, Server Hello Done
62	11.902580	2.2.2.1	172.16.200.99	TCP	66	57602 → 443 [ACK] Seq=193 Ack=2052 Win=35328 Len=0 TSval=1423415088 TSecr=1140593661
63	11.903194	2.2.2.1	172.16.200.99	TLSv1.2	159	Client Key Exchange, Change Cipher Spec, Encrypted Handshake Message
64	11.903415	172.16.200.99	2.2.2.1	TLSv1.2	117	Change Cipher Spec, Encrypted Handshake Message
65	11.903491	2.2.2.1	172.16.200.99	TLSv1.2	172	Application Data
66	11.903752	172.16.200.99	2.2.2.1	TLSv1.2	418	Application Data

```
> Frame 58: 258 bytes on wire (2064 bits), 258 bytes captured (2064 bits)
> Ethernet II, Src: VMware_52:b2:9b (00:0c:29:52:b2:9b), Dst: VMware_e2:22:3b (00:0c:29:e2:22:3b)
> Internet Protocol Version 4, Src: 2.2.2.1, Dst: 172.16.200.99
> Transmission Control Protocol, Src Port: 57602, Dst Port: 443, Seq: 1, Ack: 1, Len: 192
> Transport Layer Security
```

SSH traffic file scanning

FortiGates can buffer, scan, log, or block files sent over SSH traffic (SCP and SFTP) depending on the file size, type, or contents (such as viruses or sensitive content).

This feature is supported in proxy-based inspection mode. It is currently not supported in flow-based inspection mode.

This feature is not supported on FortiGate models with 2 GB RAM or less. See Proxy-related features not supported on FortiGate 2 GB RAM models on page 126 for more information.

You can configure the following SSH traffic settings in the CLI:

- Protocol options
- DLP profile
- Antivirus (profile and quarantine options)

To configure SSH protocol options:

```
config firewall profile-protocol-options
    edit <name>
        config ssh
           set options {oversize clientcomfort servercomfort}
           set comfort-interval <1 - 900>
           set comfort-amount <1 - 65535>
           set oversize-limit <1 - 798>
           set uncompressed-oversize-limit <0 - 798>
           set uncompressed-nest-limit <2 - 100>
           set scan-bzip2 {enable | disable}
        end
    next
end
```

To configure SCP block and log options:

```
config ssh-filter profile
    edit <name>
        set block scp
        set log scp
    next
end
```

To configure the DLP profile:

```
config dlp profile
    edit <name>
        set full-archive-proto ssh
        set summary-proto ssh
        config filter
            edit 1
                set proto ssh
            next
        end
    next
end
```

To configure the antivirus profile options:

```
config antivirus profile
    edit <name>
        config ssh
            set av-scan {disable | block | monitor}
            set outbreak-prevention {disable | block | monitor}
            set external-blocklist {disable | block | monitor}
            set fortindr {disable | block | monitor}
            set quarantine {enable | disable}
            set archive-block {encrypted corrupted partiallycorrupted multipart nested mailbomb
timeout unhandled}
            set archive-log {encrypted corrupted partiallycorrupted multipart nested mailbomb
timeout unhandled}
            set emulator {enable | disable}
        end
    next
end
```

To configure the antivirus quarantine options:

```
config antivirus quarantine
    set drop-infected ssh
    set store-infected ssh
    set drop-machine-learning ssh
    set store-machine-learning ssh
end
```

To configure SCP block and log options:

```
config ssh-filter profile
    edit <name>
        set block scp
        set log scp
    next
end
```

To apply the ssh-filter to a policy:

```
config firewall policy
    edit <id>
        set utm-status enable
        set inspection-mode proxy
        set ssh-filter-profile <ssh-filter profile>
    next
end
```

Redirect to WAD after handshake completion

In a proxy-based policy, the TCP connection is proxied by the FortiGate. A TCP three-way handshake can be established with the client even though the server did not complete the handshake.

This option uses IPS to handle the initial TCP three-way handshake. It rebuilds the sockets and redirects the session back to proxy only when the handshake with the server is established.

To enable proxy after a TCP handshake in an SSL/SSH profile:

```
config firewall ssl-ssh-profile
    edit "test"
        config https
            set ports 443
            set status certificate-inspection
            set proxy-after-tcp-handshake enable
        end
   next
end
```

To enable proxy after a TCP handshake in protocol options:

```
config firewall profile-protocol-options
    edit "test"
        config http
            set ports 80
            set proxy-after-tcp-handshake enable
            unset options
            unset post-lang
        end
   next
end
```

HTTP/2 support in proxy mode SSL inspection

Security profiles in proxy mode can perform SSL inspection on HTTP/2 traffic that is secured by TLS 1.2 or 1.3 using the Application-Layer Protocol Negotiation (ALPN) extension.

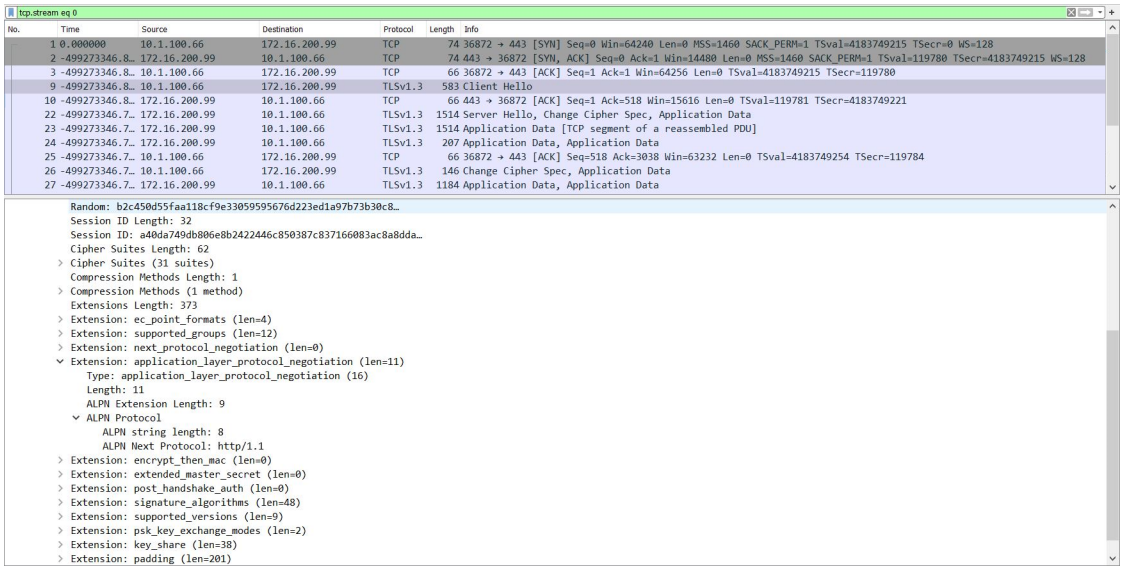
To set the ALPN support:

```
config firewall ssl-ssh-profile
    edit <profile>
        set supported-alpn {all | http1-1 | http2 | none}
    next
end
```

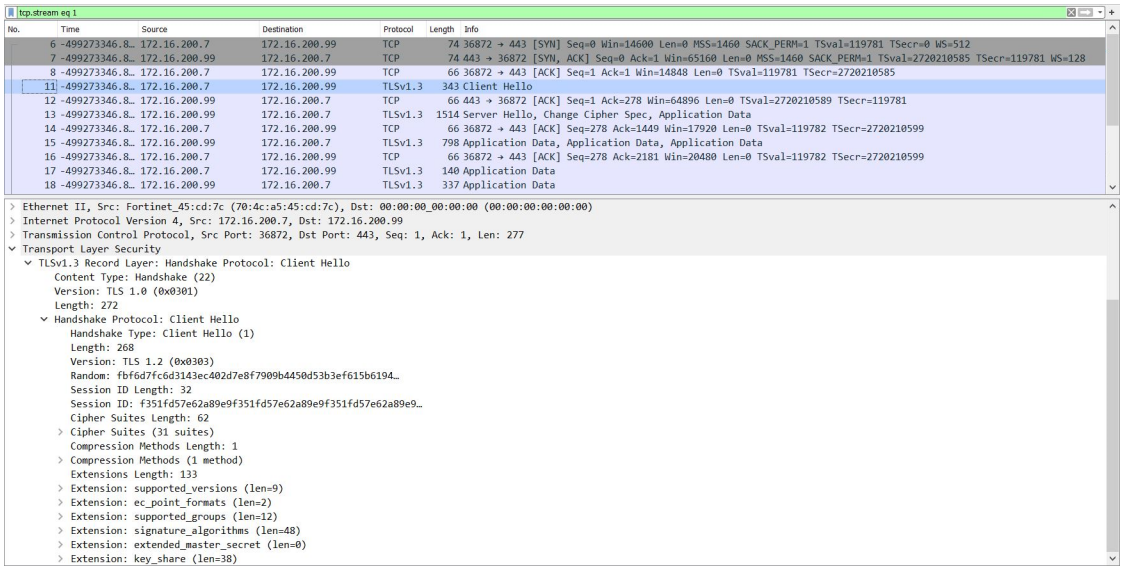
`all`	The FortiGate forwards all ALPN extensions, except SPDY. This is the default value.
`http1-1`	The FortiGate only forwards ALPN extensions that use HTTP/1.1. If the ALPN extension uses HTTP/2 or SPDY, then the FortiGate strips the ALPN header from the Client Hello.
`http2`	The FortiGate only forwards ALPN extensions that use HTTP/2. If the ALPN extension uses HTTP/1.1 or SPDY, then the FortiGate strips the ALPN header from the Client Hello.
`none`	The FortiGate always strips the ALPN header from the Client Hello when forwarding.

For example, if `supported-alpn` is set to `http2`, but the extension uses HTTP/1.1 or SPDY, the ALPN header is stripped from the Client Hello:

- Incoming packet capture:



- Outgoing packet capture:

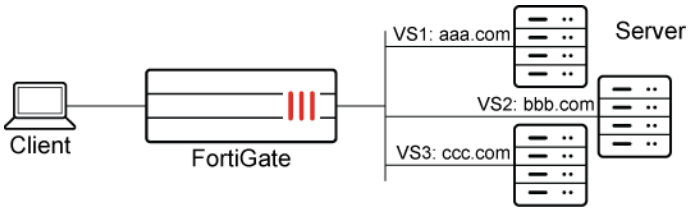


Define multiple certificates in an SSL profile in replace mode

Multiple certificates can be defined in an SSL inspection profile in replace mode (*Protecting SSL Server*). This allows multiple sites to be deployed on the same protected server IP address, and inspection based on matching the SNI in the certificate.

When the FortiGate receives the client and server hello messages, it will compare the server name identification (SNI) and the common name (CN) with the certificate list in the SSL profile, and use the matched certificate as a replacement. If there is no matched server certificate in the list, then the first server certificate in the list is used as a replacement.

Example



To configure an SSL profile in replace mode with multiple certificates:

```
config firewall ssl-ssh-profile
    edit "multi-cert"
        set server-cert-mode replace
        set server-cert "bbb" "aaa"
    next
end
```

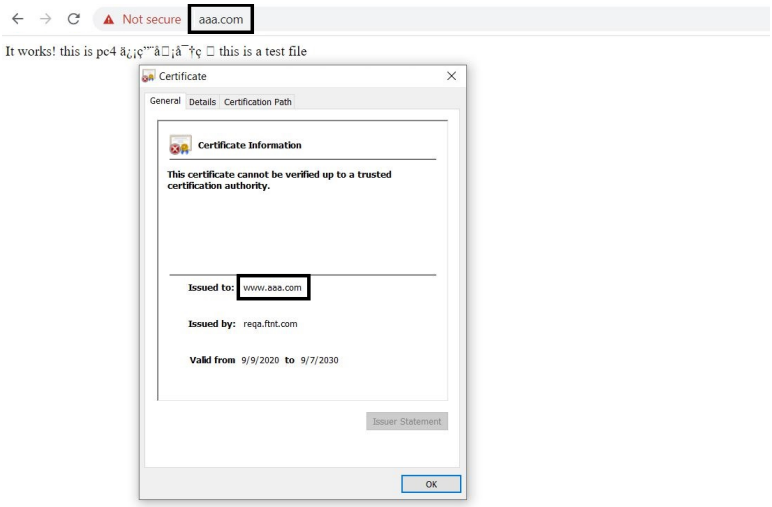
To configure a policy that uses the SSL profile:

```
config firewall policy
    edit 1
        set name "multi-cert"
        set srcintf "port6"
        set dstintf "port11"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "multi-cert"
        set av-profile "default"
        set webfilter-profile "default"
        set logtraffic all
```

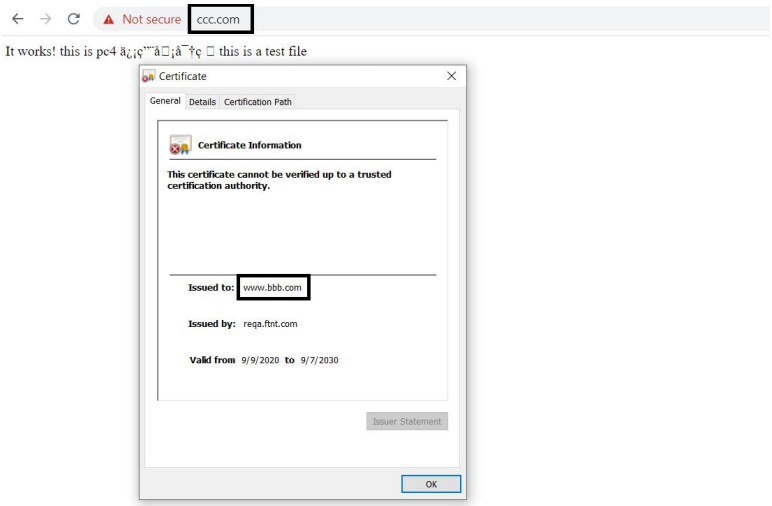
```
        set nat enable
    next
end
```

Results

If the SNI matches the CN in the certificate list in the SSL profile, then the FortiGate uses the matched server certificate. In this example, when the client accesses *www.aaa.com*, the FortiGate will use the *aaa* certificate as a replacement.



If the SNI does not match the CN in the certificate list in the SSL profile, then the FortiGate uses the first server certificate in the list. In this example, when the client accesses *www.ccc.com*, because there is no certificate for *www.ccc.com*, the FortiGate will use the *bbb* certificate as a replacement.



Disabling the FortiGuard IP address rating

The FortiGuard IP address rating for SSL exemptions and proxy addresses can be disabled using the `ssl-exemption-ip-rating` and `address-ip-rating` options.

To disable using the FortiGuard IP address rating for SSL exemptions:

```
config firewall ssl-ssh-profile
    edit <name>
        set ssl-exemption-ip-rating {enable | disable}
    next
end
```

To disable using the FortiGuard IP address rating for proxy addresses:

```
config firewall profile-protocol-options
    edit <name>
        config http
            set address-ip-rating {enable | disable}
        end
    next
end
```

The `ssl-exemption-ip-rating` and `address-ip-rating` options are enabled by default, so when both a website domain and its IP address return different categories after being rated by FortiGuard, the IP address category takes precedence when evaluating SSL exemptions associated with the SSL inspection profile and proxy addresses associated with the proxy protocol options profile. SSL exemptions and the `ssl-exemption-ip-rating` option work in both inspection modes (proxy and flow).

When the categories associated with the website domain and IP address are different, disabling the FortiGuard IP rating ensures that the FortiGuard domain category takes precedence when evaluating the preceding objects. For most websites, the domain category is valid when its IP address is unrated by FortiGuard. Since being unrated is considered as not having a category, the FortiGate uses the domain category as the website category.

A website might have an IP category that differs from its domain category. If they are different, the FortiGate uses the rating weight of the IP address or domain name to determine the rating result and decision. The rating weight is hard-coded in the FortiGate and depending on the relative category weights, the FortiGate may use the IP category instead of the website category. If the `ssl-exemption-ip-rating` option is disabled in the SSL inspection profile, then the FortiGate uses the domain category as the website category, which ensures SSL exemption operation as intended.

The `address-ip-rating` option in a proxy protocol options profile functions the same way as the `ssl-exemption-ip-rating` option. If the `address-ip-rating` option is disabled in a profile that is used in an explicit proxy policy that also uses a web filter profile, for HTTP or HTTPS traffic to a website that has different IP and domain categories and that matches the policy, the FortiGate will use the domain category when it evaluates categories for the web filter.

Block or allow ECH TLS connections

Encrypted Client Hello (ECH) is an extension to TLS that allows TLS to effectively hide information that is exposed in the unencrypted TLS ClientHello message. The ClientHello is one of the first messages sent in a TLS handshake, containing information inside the Server Name Indication (SNI) field about the destination host. By encrypting the ClientHello, this information is not exposed in plaintext.

Using the ECH extension, the client queries DNS for the DNS record of the destination host containing the ECH configuration and public key for encrypting the ClientHello message. To prevent the identity from being exposed within the DNS query, clients usually use DoH or DoT to encrypt the DNS packets.

With the public key returned from the DNS server, the client can encrypt the ClientHello message, now called the inner ClientHello. An outer, unencrypted ClientHello must still be present to route to the server, often a CDN, that can unpack and reroute the traffic to the final destination.

support for inspecting TLS connections that utilize ECH is supported in both proxy and flow mode.

Impact on the firewall

When a FortiGate does certificate inspection, for example for web category filtering, the FortiGate relies on the SNI field in the ClientHello to accurately determine the hostname of the server it is connecting to, and then performs category filtering based on this hostname. When ECH is used, this is equivalent to looking up the encrypted SNI field in the inner ClientHello. This means that applying ECH can break certificate inspection.

If the FortiGate is performing deep inspection, it always strips the ECH extension from an ECH, effectively forcing the client browser to use a non-ECH TLS connection.

Prevent ECH from affecting certificate inspection

By preventing the use of ECH, the destination hostname is not encrypted and certificate inspection can be performed accurately on the SNI on the unencrypted ClientHello. There are two ways to do this:

1. Blocking the ClientHello that uses ECH, allowing TLS to fallback to using a plaintext ClientHello.
2. Stripping the ECH response that is returned from the DNS server, preventing TLS from receiving the ECH encryption key to encrypt the ClientHello.

This feature is supported in both proxy and flow inspection mode.

Configuration

To configure blocking/allowing ECH and filtering on the SNI string in the TLS connection in the CLI:

```
config firewall ssl-ssh-profile
    edit <name>
        config https
```

```
            set status certificate-inspection
            set encrypted-client-hello {block | allow}
        end
        config ech-outer-sni
            edit <name>
                set sni <string>
            next
        end
    next
end
```

`encrypted-client-hello {block | allow}`	Block/allow session based on existence of encrypted-client-hello.
`ech-outer-sni`	Filtering on the ClientHelloOuter Server Name Indications (SNIs) to be blocked.
`sni <string>`	ClientHelloOuter SNI to be blocked.

To configure stripping ECH information from DNS responses in the CLI:

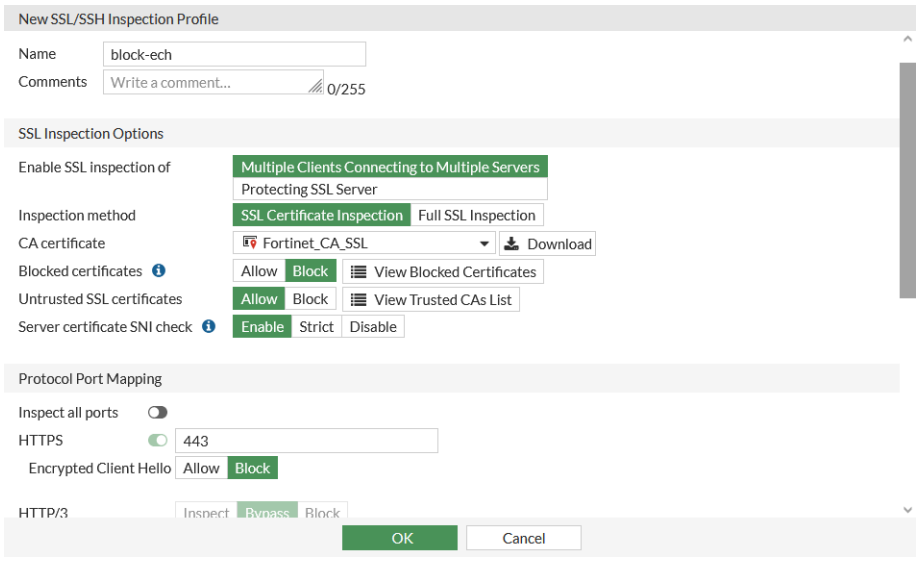
```
config dnsfilter profile
    edit <name>
        set strip-ech {enable | disable}
    next
end
```

`strip-ech {enable | disable}`	Enable/disable removal of the ECH service parameter from supporting DNS RRs.

To configure blocking/allowing ECH and filtering on the SNI string in the TLS connection GUI:

Blocking/allowing ECH is only configurable in an SSL/SSH inspection profile with *Inspection method* set to *SSL Certificate Inspection*.

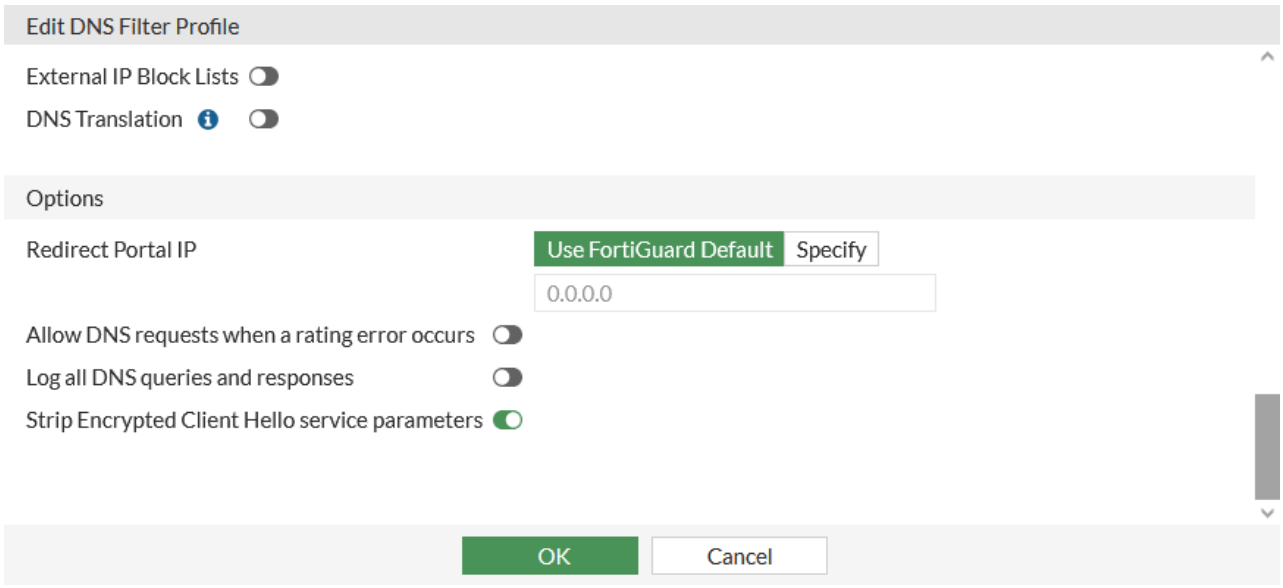
1. Go to *Security Profiles* > *SSL/SSH Inspection* and edit an existing profile or click *Create New*.
2. Set *Inspection method* to *SSL Certificate Inspection*.
3. Set *Encrypted Client Hello* to *Block*.



4. Click *OK*.
   SNIs cannot be configured in the GUI.

To configure stripping ECH information from DNS responses in the GUI:

1. Go to *Security Profiles* > *DNS Filter* and edit an existing profile or click *Create New*.
2. Enable *Strip Encrypted Client Hello service parameters*.



3. Click *OK*.

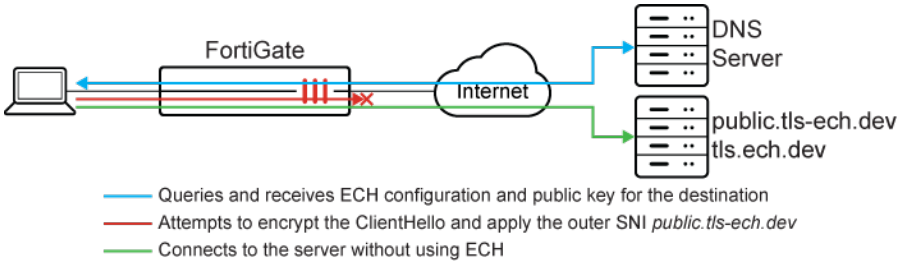
Examples

- Blocking TLS connections with certificate inspection when ECH is used in the TLS handshake through the FortiGate on page 2714

- Allowing TLS connections with certificate inspection when ECH is used in the TLS handshake through the FortiGate on page 2716
- Stripping ECH information from DoH responses on page 2718

Blocking TLS connections with certificate inspection when ECH is used in the TLS handshake through the FortiGate

In this example, an SSL/SSH inspection profile is configured to block TLS connections from some SNIs when ECH is used in the TLS handshake. Client messages with the outer SNI `public.tls-ech.dev` are blocked.



A webfilter block message will be shown when trying to connect directly to public.tls-ech.dev. Accessing the webpage tls-ech.dev, which uses public.tls-ech.dev in its outer SNI, will show that the client is not using ECH.

To configure blocking a TLS connection that uses ECH:

1. Configure an SSL/SSH inspection profile to block ECH and set the SNIs to match the outer SNI in the ECH message during the TLS handshake:

```
config firewall ssl-ssh-profile
    edit "block-ech"
        config https
            set status certificate-inspection
            set encrypted-client-hello block
        end
        config ech-outer-sni
            edit "cloudflare"
                set sni "cloudflare-ech.com"
            next
            edit "tls-ech"
                set sni "public.tls-ech.dev"
            next
            edit "defo.ie"
                set sni "cover.defo.ie"
            next
        end
    next
end
```

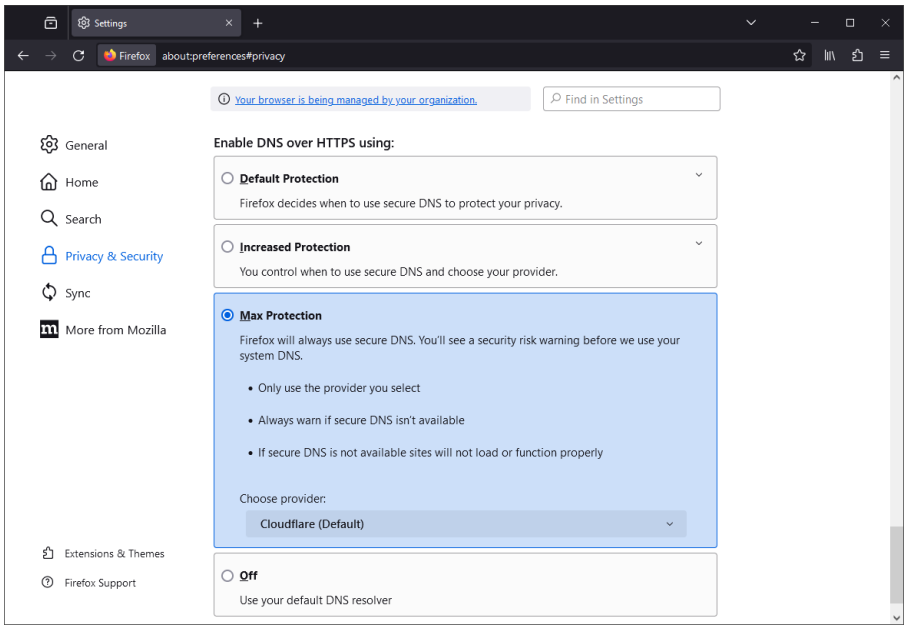
2. Apply the profile in a firewall policy:

```
config firewall policy
    edit 1
        set ssl-ssh-profile "block-ech"
```

```
    next
end
```

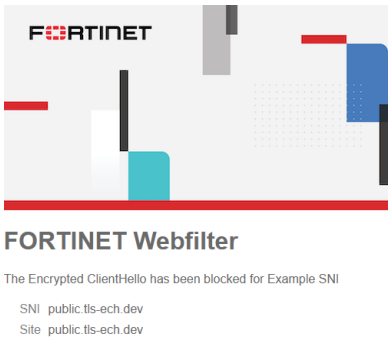
To check the results:

1. In a browser, enable DNS over HTTPS. For example, in Firefox go to *Settings* > *Privacy & Security* and under *DNS over HTTPS* enable *Max Protection*.



2. Visit a website, such as *https://public.tls-ech.dev*.

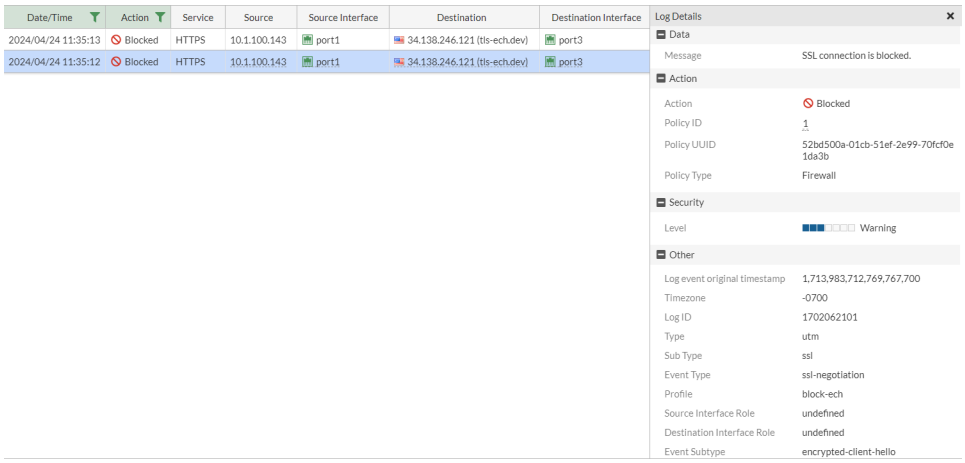
   Because ECH is blocked and the outer SNI matches one of the configured SNIs, the ECH initiated by the browser is blocked. In this case, the browser shows a replacement message:



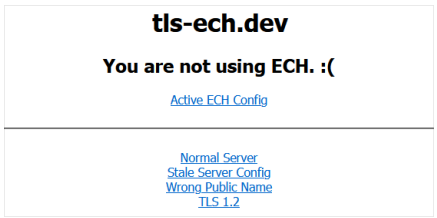
3. An SSL log shows the blocked connection:

```
2: date=2024-04-24 time=11:35:12 eventtime=1713983712769767807 tz="-0700" logid="1702062101"
type="utm" subtype="ssl" eventtype="ssl-negotiation" level="warning" vd="vdom1"
action="blocked" policyid=1 poluuid="52bd500a-01cb-51ef-2e99-70fcf0e1da3b" policytype="policy"
sessionid=120849 service="HTTPS" profile="block-ech" srcip=10.1.100.143 srcport=63253
srccountry="Reserved" dstip=34.138.246.121 dstport=443 dstcountry="United States"
srcintf="port1" srcintfrole="undefined" dstintf="port3" dstintfrole="undefined"
srcuuid="fbd0def6-01ca-51ef-496f-08c90cb6b123" dstuuid="fbd0def6-01ca-51ef-496f-08c90cb6b123"
```

```
proto=6 eventsubtype="encrypted-client-hello" hostname="public.tls-ech.dev" msg="SSL
connection is blocked."
```



4. Try to visit a different website, such as *tls-ech.dev*, that does not match a default SNI to be blocked. The browser will initially try to establish an ECH-enabled TLS connection to *public.tls-ech.dev*, but that will be blocked, forcing the browser to connect to the actual website without ECH. In this case, the browser will load the actual website and not the replacement message:

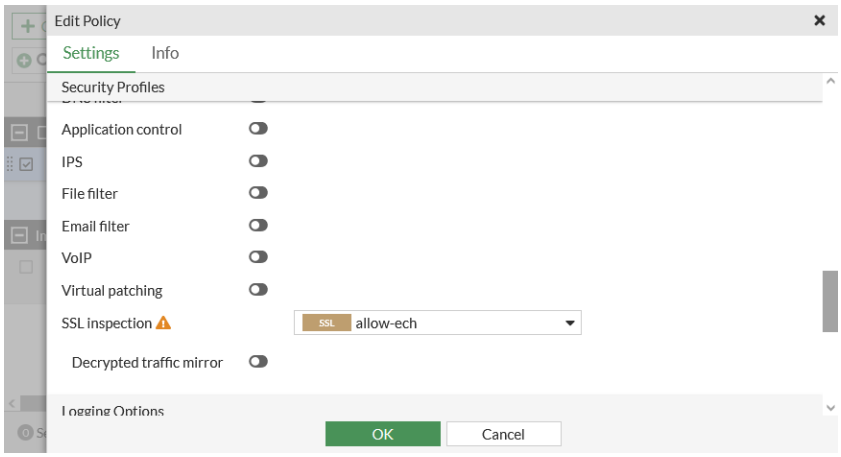


Allowing TLS connections with certificate inspection when ECH is used in the TLS handshake through the FortiGate

In this example, an SSL/SSH inspection profile is configured to allow TLS connections when ECH is used in the TLS handshake.

To configure allowing a TLS connection that uses ECH in the GUI:

1. Go to *Security Profiles* > *SSL/SSH Inspection* and edit an existing profile or click *Create New*.
2. Set *Inspection method* to *SSL Certificate Inspection*.
3. Set *Encrypted Client Hello* to *Allow*.
4. Click *OK*.
5. Go to *Policy & Objects* > *Firewall Policy* and edit an existing policy or click *Create New*.
6. Set *SSL inspection* to the SSL/SSH Inspection profile.



7. Click *OK*.

To configure allowing a TLS connection that uses ECH in the CLI:

1. Configure an SSL/SSH inspection profile to block ECH and set the SNIs to match the outer SNI in the ECH message during the TLS handshake:

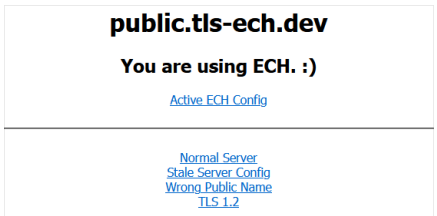
```
config firewall ssl-ssh-profile
    edit "allow-ech"
        config https
            set status certificate-inspection
            set encrypted-client-hello allow
        end
    next
end
```

2. Apply the profile in a firewall policy:

```
config firewall policy
    edit 1
        set ssl-ssh-profile "allow-ech"
    next
end
```

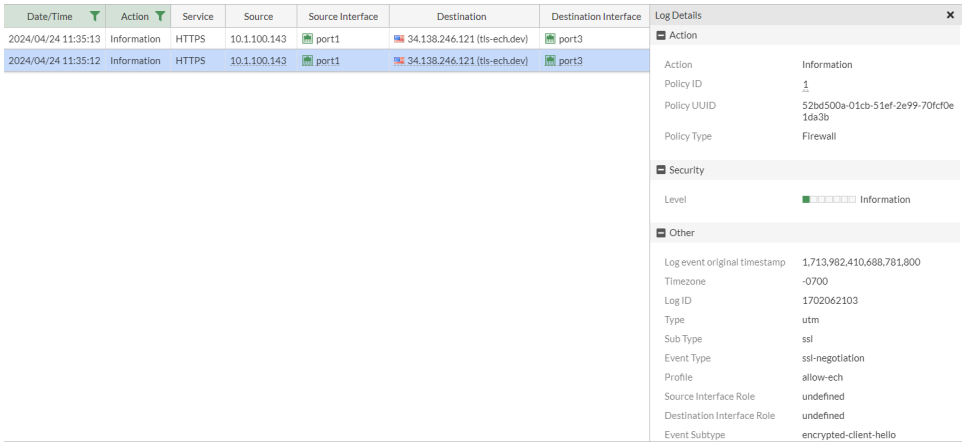
To check the results:

1. In a browser, enable DNS over HTTPS. For example, in Firefox go to *Settings* > *Privacy & Security* and under *DNS over HTTPS* enable *Max Protection*.
2. Visit a website, such as *https://public.tls-ech.dev*.
   The ECH-enabled TLS connection is allowed through the firewall policy and the website opens:



3. An SSL log shows the allowed connection:

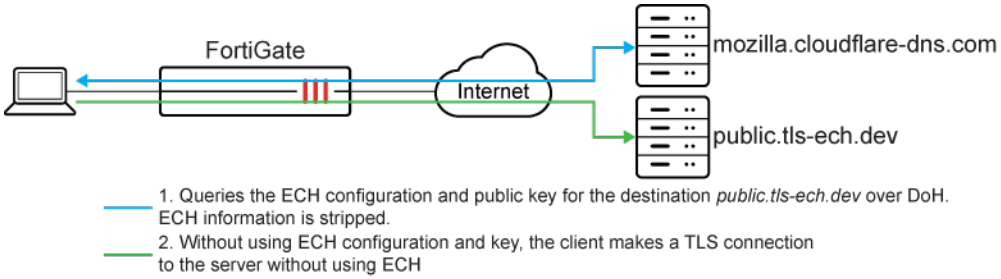
```
2: date=2024-04-24 time=11:13:31 eventtime=1713982410688781760 tz="-0700" logid="1702062103"
type="utm" subtype="ssl" eventtype="ssl-negotiation" level="information" vd="vdom1"
action="info" policyid=1 poluuid="52bd500a-01cb-51ef-2e99-70fcf0e1da3b" policytype="policy"
sessionid=118528 service="HTTPS" profile="allow-ech" srcip=10.1.100.143 srcport=63232
srccountry="Reserved" dstip=34.138.246.121 dstport=443 dstcountry="United States"
srcintf="port1" srcintfrole="undefined" dstintf="port3" dstintfrole="undefined"
srcuuid="fbd0def6-01ca-51ef-496f-08c90cb6b123" dstuuid="fbd0def6-01ca-51ef-496f-08c90cb6b123"
proto=6 eventsubtype="encrypted-client-hello" hostname="public.tls-ech.dev"
```



Stripping ECH information from DoH responses

DNS filters are used to strip ECH information from DNS responses, and force the browser to not use ECH for TLS connections. The browser relies on the ECH information from DNS over HTTPS (DoH) for ECH-enabled TLS connections.

In this example, a client sends a DoH query to a Cloudflare DNS server. The ECH information in the DNS response is stripped.



1. Policy 3 performs deep inspection on DoH traffic between the browser and the DoH server (mozilla.cloudflare-dns.com). A DNS filter profile is applied that strips the ECH information from the DoH response, forcing the browser to use a non-ECH TLS connection.
2. The browser then establishes a TLS connection through policy 1. Although policy 1 has an SSL/SSH inspection profile applied that allows ECH, ECH is not used because the ECH information from the DoH response was stripped by policy 3.

To configure and test stripping ECH information from DoH responses:

1. Configure the DNS filter profile that removes the ECH information:

```
config dnsfilter profile
    edit "strip-ech-enable"
        set strip-ech enable
    next
end
```

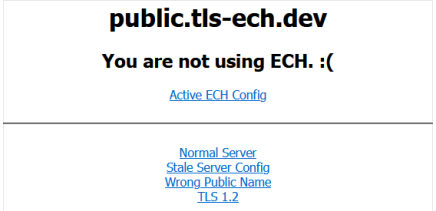
2. Configure a firewall address:

```
config firewall address
    edit "mozilla.cloudflare-dns.com"
        set type fqdn
        set fqdn "mozilla.cloudflare-dns.com."
    next
end
```

3. Configure the policies:

```
config firewall policy
    edit 3
        set srcintf "port1"
        set dstintf "port3"
        set action accept
        set srcaddr "all"
        set dstaddr "mozilla.cloudflare-dns.com"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "deep-inspection"
        set dnsfilter-profile "strip-ech-enable"
        set logtraffic all
        set nat enable
    next
    edit 1
        set srcintf "port1"
        set dstintf "port3"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "allow-ech"
        set logtraffic all
        set nat enable
    next
end
```

4. In a browser, go to *https://public.tls-ech.dev* to see that ECH is not being used:



5. The debug log will show that the ECH service parameter was removed:

```
# diagnose debug application dnsproxy -1
...
[worker 0] dns_secure_filter_ech()-2166: Found ECH key=5
...
[worker 0] dns_secure_filter_ech()-2203: Removed ECH service parameter
```

6. If the DNS filter profile is changed to disable stripping the ECH information, then the website will show that ECH is being used:

```
config dnsfilter profile
    edit "strip-ech-enable"
        set strip-ech disable
    next
end
```

Configuring certificate probe failure option

The `cert-probe-failure` command is not available for FortiGate models with 2 GB RAM or less, including FortiGate/FortiWiFi 40F and 60F series of devices and their variants, and FortiGate-Rugged 60F (2 GB versions only). See Proxy-related features not supported on FortiGate 2 GB RAM models on page 126.

With an SSL inspection profile configured for either certificate or deep inspection, the FortiGate performs certificate probing where it checks a server certificate before a client-server HTTPS connection is established. Certificate probe failures can occur due to issues like TCP or TLS handshake failures, misrouted traffic, or untrusted root or intermediate CA certificates. If a certificate is invalid, untrusted, or mismatched, then the FortiGate flags a certificate probe failure in the logs.

Certificate probe failure can be allowed or blocked for HTTPS and SSL:

```
config firewall ssl-ssh-profile
    edit <name>
        config https
```

```
            set cert-probe-failure {allow | block}
        end
        config ssl
            set cert-probe-failure {allow | block}
        end
    next
end
```

Option	Description
allow	Bypass the session when unable to retrieve server's certificate for inspection.
block	Block the session when unable to retrieve server's certificate for inspection.

For cases that have increased security requirements or stringent organization security policy, certificate probe failure may need to be configured to block. `cert-probe-failure` is available for custom SSL deep inspection profiles. This option applies to flow mode policies and is available when inspecting all ports is disabled (`set inspect-all disable`).

Hybrid PQC support in proxy mode SSL deep inspection

FortiGate supports TLS 1.3 hybrid Post-Quantum Cryptography (PQC) key exchanges in SSL deep inspection (proxy mode), enabling secure traffic inspection. This ensures compatibility with modern browsers and PQC-enabled servers that use hybrid PQC algorithms, such as X25519MLKEM768.

Now, both pure PQC and hybrid PQC are supported in SSL deep inspection:

- Hybrid PQC algorithms:
  - X25519MLKEM768 (0x11EC)
  - SecP256r1MLKEM768 (0x11EB)
  - SecP384r1MLKEM1024 (0x11ED)
- Pure PQC algorithms:
  - MLKEM512 (0x1302)
  - MLKEM768 (0x1302)
  - MLKEM1024 (0x1302)

FortiGate can perform SSL deep inspection and the client can access PQC algorithm web servers in proxy mode.

These changes also add support for DNS over TLS, HTTPS (1.0, 2.0, 3.0), POP3, IMAPS, and SMTPS with deep inspection to negotiate hybrid+ML-KEM. Hybrid+ML-KEM and ML-KEM are also supported when these protocols run in a virtual-server or explicit-proxy modes.

Although SIP over TLS and the urlfilter TLS connection to FortiGuard use the same TLS implementation as HTTPS (1.0, 2.0, 3.0), POP3, IMAPS, and SMTPS, they do not yet support hybrid+ML-KEM.

Example

This example demonstrates the following scenarios:

- How clients can connect to a web server that uses PQC encryption
- How clients can access through a DNS connection to Cloudflare a web server using hybrid PQC key exchange

First configure the FortiGate for testing, and use the same configuration to test both scenarios.

To configure FortiGate for testing:

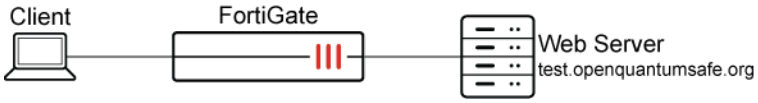
1. On FortiGate, clone a deep-inspection profile and add port 6000-6005 to HTTPS:
   a. Go to *Security Profiles* > *SSL/SSH Inspection*.
   b. Select a deep-inspection profile, and click *Clone*.
   c. Leave *Name* set to *Clone of deep-inspection*.
   d. Under *Protocol Port Mapping*, enable *HTTPS*, and enter *443,6000,6001,6002,6003,6004,6005*.
   e. Click *OK* to save the changes to *Clone of deep-inspection*.
2. Add Port Range 6000–6005 to the HTTPS Service:
   a. Go to *Policy & Objects* > *Service*.
   b. Locate the service named HTTPS and click *Edit*.
   c. In the *Edit Service* window, scroll to the *Protocol Options* section.
   d. Under *Destination Port*, click + to add a new port entry.
   e. Set the second entry to:
      - *Protocol*: TCP
      - *Low*: 6000
      - *High*: 6005
   f. Confirm the configuration and click *OK* to save the service.
3. On FortiGate, configure a policy and select the *Clone of deep-inspection* profile.

```
config firewall policy
   edit 1
       set srcintf "port2"
       set dstintf "port1"
       set action accept
       set srcaddr "all"
       set dstaddr "all"
       set schedule "always"
       set service "ALL"
       set utm-status enable
       set inspection-mode proxy
       set ssl-ssh-profile "Clone of deep-inspection"
       set av-profile "av"
       set logtraffic all
       set nat enable
  next
end
```

4. Test the following scenarios:

- Connect to a web server that uses PQC encryption on page 2723
- Send DNS to a Cloudflare DNS server that uses PQC encryption on page 2723

Connect to a web server that uses PQC encryption



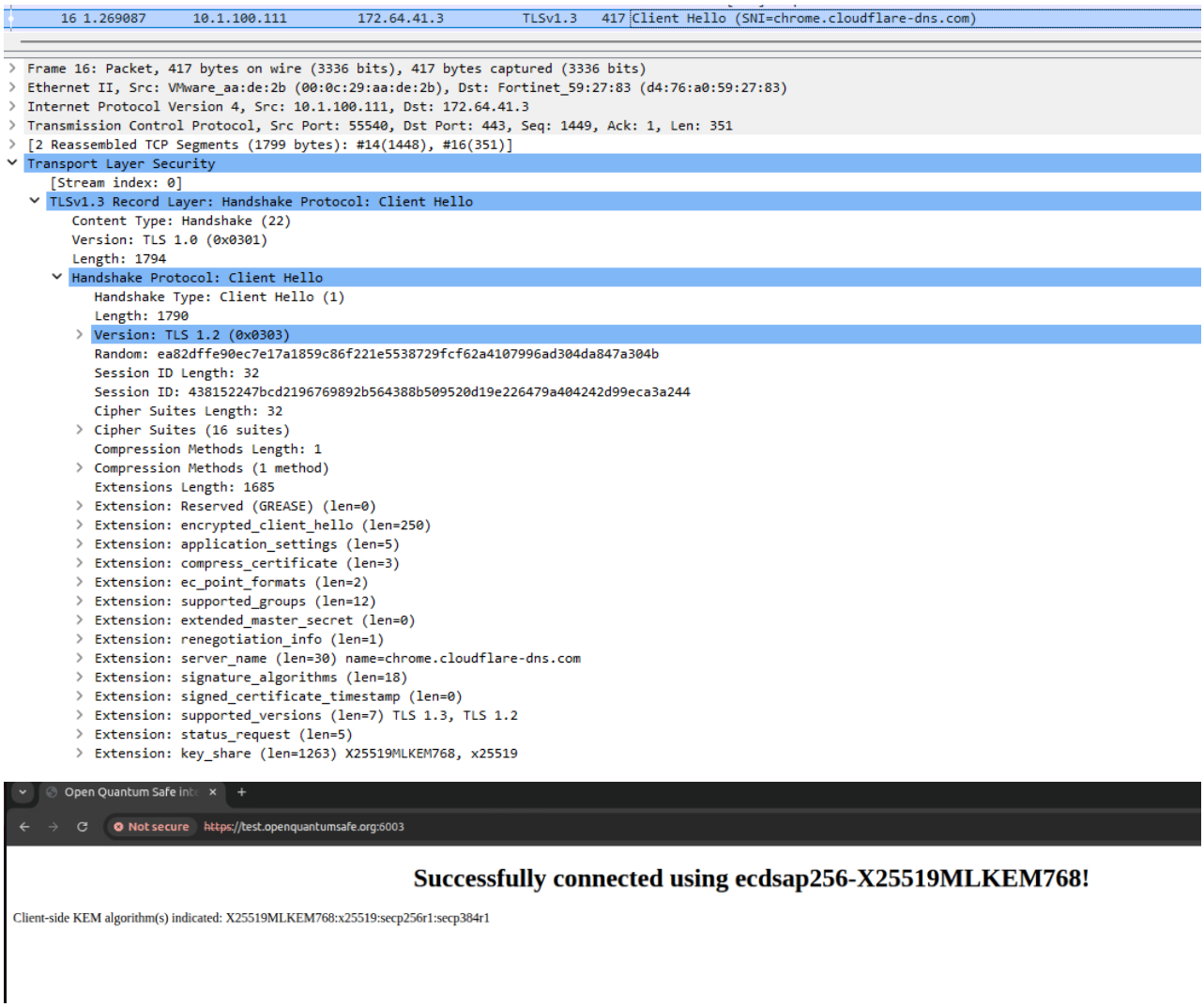
`test.openquantumsafe.org:6000` supports the X25519MLKEM768 key exchange algorithm. Previously, hybrid PQC algorithm was downgraded to non-ML-KEM encryption, and the site would be unreachable. Now, the client can access the web server using hybrid PQC key exchange.

Send DNS to a Cloudflare DNS server that uses PQC encryption

To test the example:

1. In a browser, enable secure DNS connection with Cloudflare (1.1.1.1) to force DNS over TLS.

2. Packet capture shows that the Client Hello sent to the Cloudflare DNS server uses X25519MLKEM768.



Custom signatures

You can create the following custom signatures and apply them to firewall policies:

- Application group
- Application signature
- IPS signature

The following topic provides information about custom signatures:

- Configuring custom signatures on page 2725
- Blocking applications with custom signatures on page 2726
- Filters for application control groups on page 2730
- Application groups in traffic shaping policies on page 2733

Configuring custom signatures

IPS signatures are the basis of signature-based intrusion prevention. Every attack can be reduced to a particular string of commands or a sequence of commands and variables. See Intrusion prevention on page 2478 for more information.

An IPS signature identifies characteristics of a packet that are unique to an attack, such as the protocol type, an option/value pair within the payload, other special aspects of the payload, or specific application options. Custom IPS signatures can be created to block, monitor, or quarantine specific traffic that is not covered by the IPS definitions list. To view the IPS definitions list:

- Go to *Security Profiles* > *IPS Signatures*.
- Go to *Security Profiles* > *Intrusion Prevention*, edit an existing IPS sensor, and click *View IPS Signatures* in the right-hand pane.
- Go to *System* > *FortiGuard*, in the *Subscriptions* > *Licensed* tab, expand *Intrusion Prevention System (IPS)*, and in the *IPS Signatures* row click *Actions* > *View List*.

An application signature identifies characteristics of a packet that is unique to an application. Custom application signatures can be used in application control profiles to block traffic from specific applications that are not covered by the application control signatures list. To view the application control signatures list:

- Go to *Security Profiles* > *Application Signatures* and select the *Signature* view.
- Go to *Security Profiles* > *Application Control*, edit an existing application sensor, and click *View Application Signatures* in the right-hand pane.
- Go to *System* > *FortiGuard*, in the *Subscriptions* > *Licensed* tab, expand *Firmware, VM, and base subscriptions*, and in the *Application Control Signatures* row click *Actions* > *View List*.

Application groups can be created by selecting individual application, or by filtering by application category. The groups can then be used in firewall policies.

For information about the syntax for building IPS and application control signatures, see the Custom IPS and Application Control Signature Syntax Guide.

To make the application signatures settings visible in the GUI:

1. Go to *System* > *Feature Visibility*
2. In the *Security Features* section, enable *Application Control*.
3. Click *Apply*.

To configure custom signatures:

1. Custom application and IPS signatures can be configured:
   - To configure custom application signatures, go to *Security Profiles* > *Application Signatures* and click *Create New* > *Custom Application Signature*. See Blocking applications with custom signatures on page 2726 for an example.
   - To configure custom IPS signatures, go to *Security Profiles* > *IPS Signatures* and click *Create New*.
2. Configure the following settings:

Name	Enter a unique name for the signature.

Comments	Enter a comment (optional).
Signature	Enter the signature.

3. Click *OK*.

To configure application groups:

1. Go to *Security Profiles* > *Application Signatures* and click *Create New* > *Application Group*.
2. Configure the following settings:

Group Name	Enter a unique name for the signature group.
Type	Set the application group type, either application ID or application filter. See Filters for application control groups on page 2730 for information about the available filters.
Members	Select the applications or filter to include in the group.
Comments	Enter a comment (optional).

3. Click *OK*.
   See Application groups in traffic shaping policies on page 2733 for more information.

Blocking applications with custom signatures

Custom signatures can be used in application control profiles to block web traffic from specific applications, such as out of support operating systems.

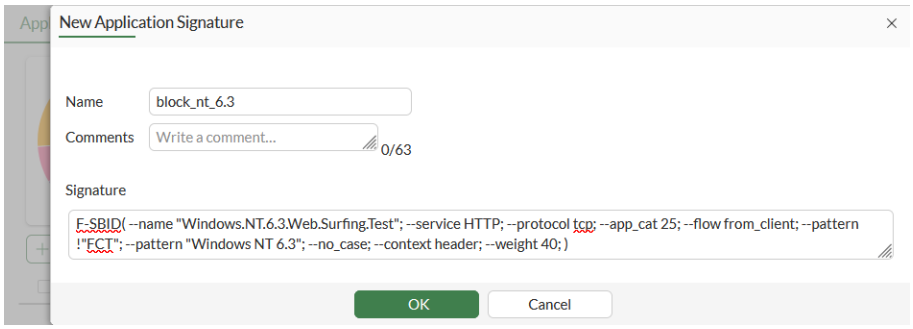
In this example, a custom signature is created to detect PCs running Windows NT 6.3 operating systems, including Windows 8.1. The signature is added to an application control profile and the action is set to block. The profile is then used in a firewall policy so that web traffic matching the signature is blocked. The logs generated by this example can be used to help identify other computers that need to be blocked.

To create the custom application signature:

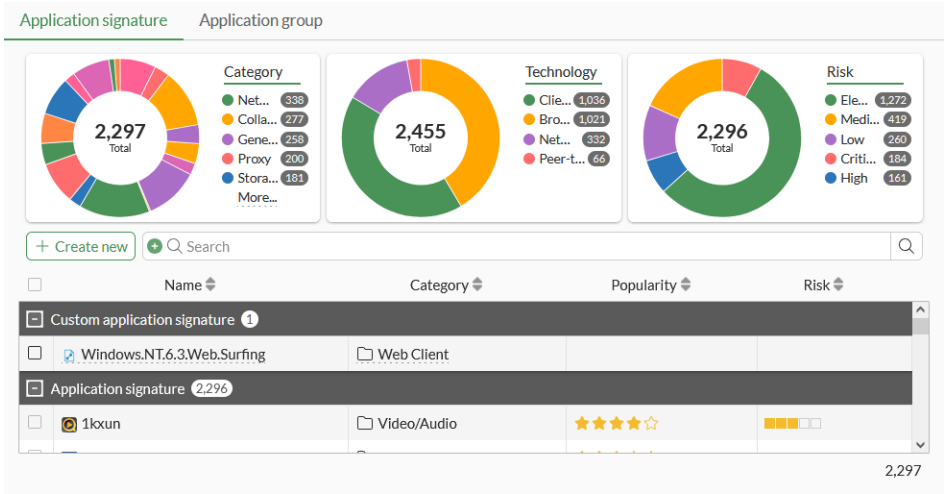
1. Go to *Security Profiles* > *Application Signatures* and click *Create New* > *Custom Application Signature*.
2. Enter a name for the custom signature, such as *block_nt_6.3*.
3. Enter the *Signature*. In this example:

```
F-SBID( --name "Windows.NT.6.3.Web.Surfing.Test"; --service HTTP; --protocol tcp; --app_cat
25; --flow from_client; --pattern !"FCT"; --pattern "Windows NT 6.3"; --no_case; --context
header; --weight 40; )
```

This signature scans HTTP and HTTPS traffic that matches the pattern *Windows NT 6.3* in its header. For blocking older versions of Windows, such as Windows XP, you would use the pattern *Windows NT 5.1*. An attack ID is automatically generated when the signature is created.

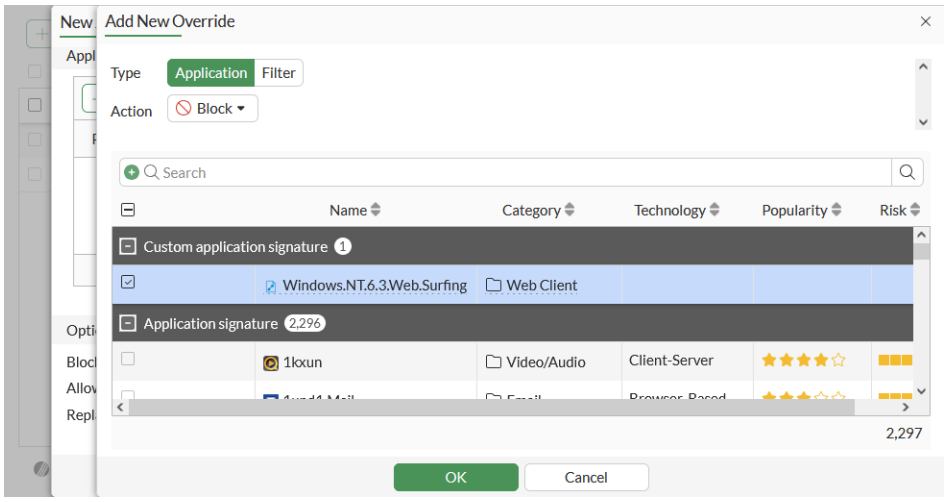


4. Click *OK*.
   The signature is included in the *Custom application signature* section of the signature list.



To use the signature in an application control profile:

1. Go to *Security Profiles* > *Application Control*.
2. Create a new profile, or edit an existing one.
3. In the *Application and Filter Overrides* table, click *Create New*.
4. Set *Type* to *Application* and *Action* to *Block*.
5. Select the custom signature from the list, using the search feature if required.



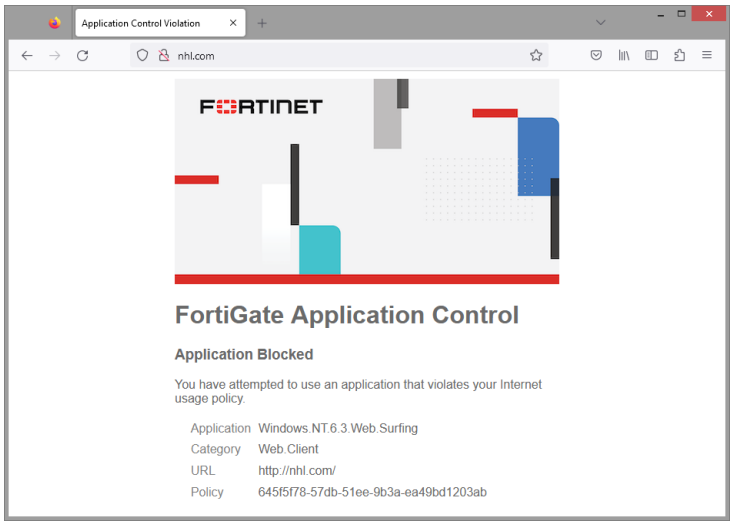
6. Click *OK*.
   The signature is added to the table.
7. Click *OK*.

To add the application control profile to a firewall policy:

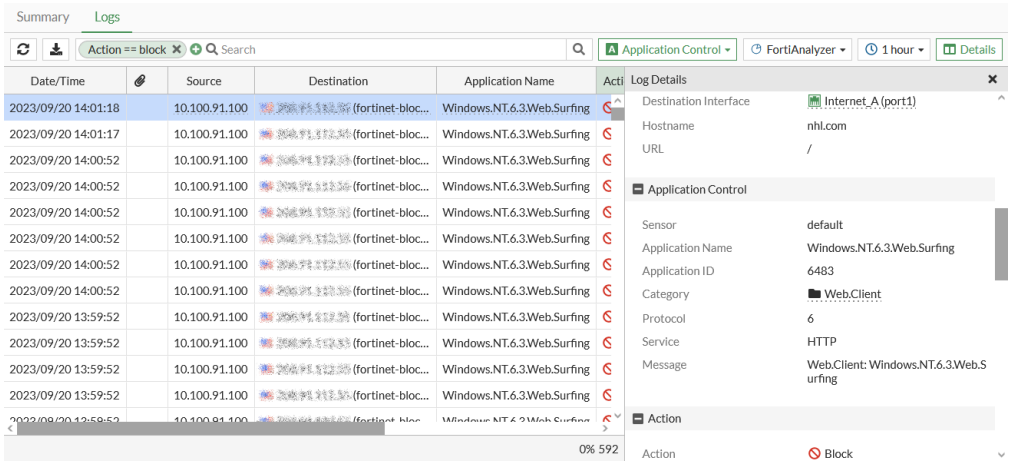
1. Go to *Policy & Objects* > *Firewall Policy*.
2. Edit the policy that currently allows a connection from the internal network to the internet.
3. In the *Security Profiles* section, enable *Application Control* and select the profile.
   If deep inspection is not enabled, then only HTTP traffic will be scanned. To scan HTTPS traffic, set SSL Inspection to a profile that includes deep inspection. See SSL & SSH Inspection on page 2690 for more information.
4. Click *OK*.

Results

When a PC running one of the affected operating systems tries to connect to the internet using a web browser, a replacement message is shown. For information on customizing replacement messages, see Replacement messages on page 3899.



Go to *Log & Report* > *Security Events* to view the web traffic that is logged for the PC that is blocked by the application signature in the *Application Control* card.



The raw logs can also be downloaded:

```
date=2023-09-20 time=14:01:18 id=7281016160055722045 itime="2023-09-20 14:01:19" euid=3 epid=1045
dsteuid=3 dstepid=101 logver=704012463 sfsid=7281016183678284934 type="utm" subtype="app-ctrl"
level="warning" action="block" sessionid=139787 policyid=13 srcip=10.100.91.100
dstip=XXX.XXX.XXX.XXX srcport=7436 dstport=80 proto=6 logid=1059028705 service="HTTP"
eventtime=1695243678286416491 incidentserialno=153203213 direction="outgoing" appid=6483
srcintfrole="lan" dstintfrole="wan" applist="default" appcat="Web.Client"
app="Windows.NT.6.3.Web.Surfing" hostname="nhl.com" url="/" eventtype="signature" srcintf="port3"
dstintf="port1" msg="Web.Client: Windows.NT.6.3.Web.Surfing" tz="-0700" policytype="policy"
srccountry="Reserved" dstcountry="United States" poluuid="645f5f78-57db-51ee-9b3a-ea49bd1203ab"
agent="Mozilla/5.0 (Windows NT 6.3; Win64; x64; rv:109.0) Gecko/20100101 Firefox/115.0"
httpmethod="GET" devid="FGVM02XXXXXXXXXX" vd="root" csf="fabric" dtime="2023-09-20 14:01:18"
itime_t=1695243679 devname="Enterprise_Core"
```

Filters for application control groups

When defining application groups in NGFW policy or profile mode, the following group filters are available: protocols, risk, vendor, technology, behavior, popularity, and category.

```
config application group
    edit <name>
        set type filter
        set protocols <integer>
        set risk <integer>
        set vendor <id>
        set technology <id>
        set behavior <id>
        set popularity <integer>
        set category <id>
    next
end
```

Command	Description
`protocols <integer>`	Application protocol filter (0 - 47, or all).
`risk <integer>`	Risk or impact of allowing traffic from this application to occur (1 - 5; low (1), elevated (2), medium (3), high (4), and critical (5)).
`vendor <id>`	Application vendor filter (0 - 25, or all).
`technology <id>`	Application technology filter: • all • 0 (network-protocol) • 1 (browser-based) • 2 (client-server) • 4 (peer-to-peer)
`behavior <id>`	Application behavior filter: • all • 2 (botnet) • 3 (evasive) • 5 (excessive bandwidth) • 6 (tunneling) • 9 (cloud)
`popularity <integer>`	Application popularity filter (1 - 5, from least to most popular).
`category <id>`	Application category filter: • 2 (P2P) • 3 (VoIP) • 5 (video/audio) • 6 (proxy) • 7 (remote access)

Command	Description
	• 8 (game) • 12 (general interest) • 15 (network service) • 17 (update) • 21 (email) • 22 (storage backup) • 23 (social media) • 25 (web client) • 26 (industrial) • 28 (collaboration) • 29 (business) • 30 (cloud IT) • 31 (mobile) • 32 (unknown applications) • 36 (generative AI)

Sample configurations

In this example, a single filter (risk level 1) is configured in the application group in NGFW policy mode, so only signatures matching this filter will match the security policy.

To configure the application group:

```
config application group
    edit "risk_1"
        set type filter
        set risk 1
    next
end
```

To configure the security policy:

```
config firewall security-policy
    edit 1
        set srcintf "port2"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set status enable
        set schedule "always"
        set enforce-default-app-port disable
        set service "ALL"
        set app-group risk_1
        set logtraffic all
```

```
   next
end
```

In this example, the application group is configured so that only signatures matching both filters, category 5 (video/audio) and technology 1 (browser-based), will match the security policy. The application group can also be configured in a traffic shaping policy.

To configure the application group:

```
config application group
    edit "two"
        set type filter
        set category 5
        set technology 1
    next
 end
```

To configure the security policy:

```
config firewall security-policy
    edit 1
        set srcintf "port2"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set status enable
        set schedule "always"
        set enforce-default-app-port disable
        set service "ALL"
        set app-group two
        set logtraffic all
    next
end
```

To configure the traffic shaping policy:

```
config firewall shaping-policy
    edit 1
        set ip-version 4
        set service "ALL"
        set app-group two
        set dstintf port1
        set traffic-shaper "max-100"
        set traffic-shaper-reverse "max-100"
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

Application groups in traffic shaping policies

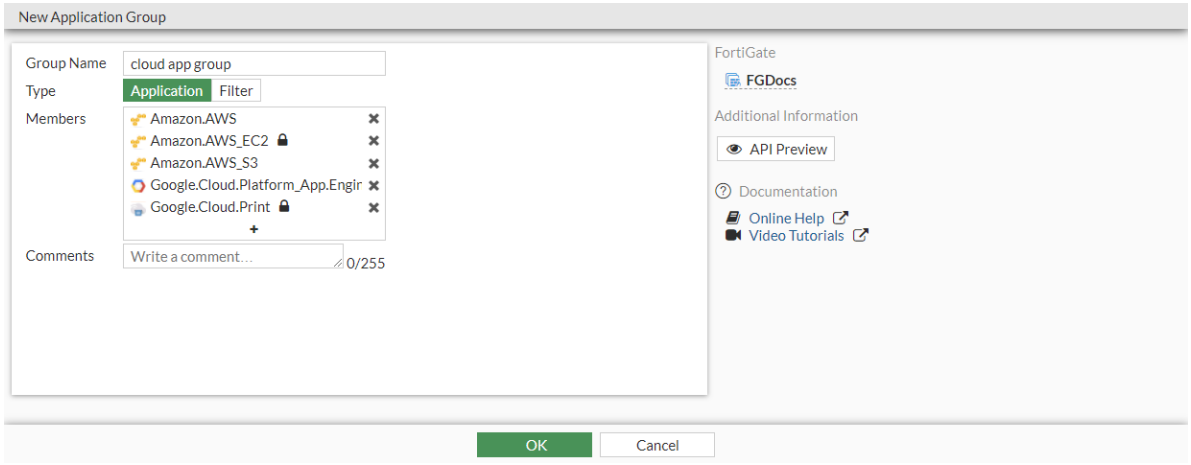
Application groups can be configured in traffic shaping policies. In this example, there are two traffic shaping policies:

- Policy 1 is for traffic related to cloud applications and has high priority.
- Policy 2 is for other traffic and has low priority.

At least one firewall policy must have application control enabled for the applications to match any policy traffic.

To configure a traffic shaping policy to use an application group in the GUI:

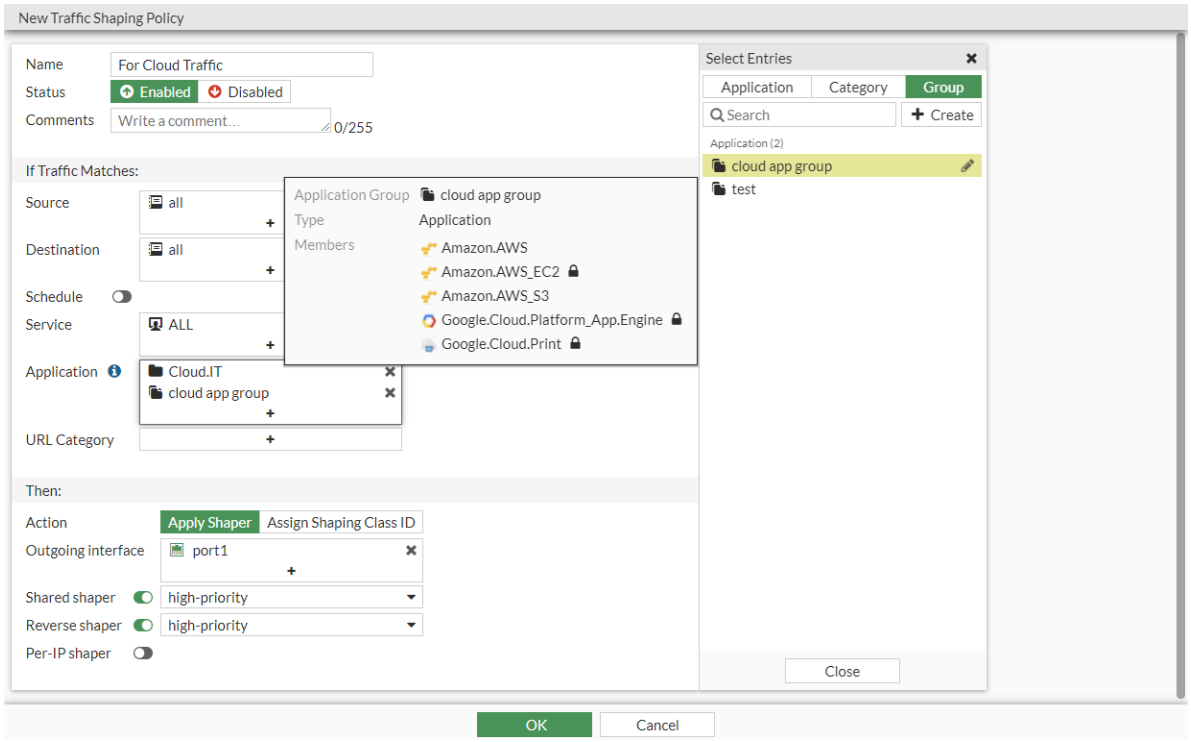
1. Configure an application group for cloud applications:
   a. Go to *Security Profiles* > *Application Signatures*.
   b. Click *Create New* > *Application Group*. The *New Application Group* page opens.
   c. Enter a name for the group, and for *Type*, select *Application*.
   d. Click the + to add the group the members.



   e. Click *OK*.
2. Create the shaping policy for the high priority cloud application traffic:
   a. Go to *Policy & Objects* > *Traffic Shaping*, select the *Traffic Shaping Policy* tab, and click *Create New*.
   b. Enter the following:

Name	For Cloud Traffic
Source	All
Destination	All
Service	All
Application	Add the *Cloud.IT* category and the *cloud app group* application group.
Outgoing interface	port1

Shared shaper	high-priority
Reverse shaper	high-priority

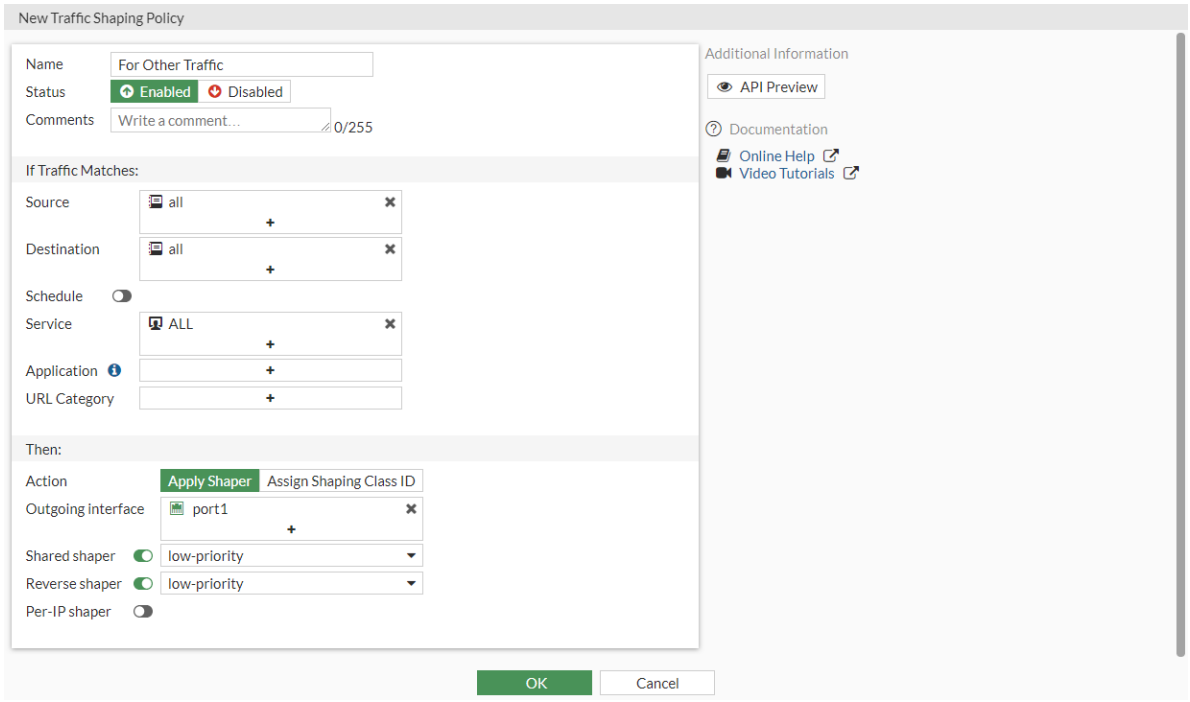


c. Click *OK*.

3. Create the shaping policy for the low priority other traffic:

a. Click *Create New* and enter the following:

Name	For Other Traffic
Source	All
Destination	All
Service	All
Outgoing interface	port1
Shared shaper	low-priority
Reverse shaper	low-priority



b. Click *OK*.

To configure a traffic shaping policy to use an application group in the CLI:

1. Configure an application group for cloud applications:

```
config application group
    edit "cloud app group"
        set application 27210 36740 35944 43296 33048
    next
end
```

2. Create the shaping policies for the high priority cloud application traffic and low priority other traffic:

```
config firewall shaping-policy
    edit 1
        set name "For Cloud Traffic"
```

```
        set service "ALL"
        set app-category 30
        set app-group "cloud app group"
        set dstintf "port1"
        set traffic-shaper "high-priority"
        set traffic-shaper-reverse "high-priority"
        set srcaddr "all"
        set dstaddr "all"
    next
    edit 2
        set name "For Other Traffic"
        set service "ALL"
        set dstintf "port1"
        set traffic-shaper "low-priority"
        set traffic-shaper-reverse "low-priority"
        set srcaddr "all"
        set dstaddr "all"
    next
end
```

Overrides

Web filter configuration can be separated into profile configuration and profile overrides.

You can also override web filter behavior based on the FortiGuard website categorization:

- Use alternate categories (web rating overrides): this method manually assigns a specific website to a different Fortinet category or a locally-created category.
- Use alternate profiles: configured users or IP addresses can use an alternative web filter profile when attempting to access blocked websites.

Some features of this functionality require a subscription to FortiGuard Web Filtering.

The following topics provide information about web overrides:

- Web rating override on page 2736
- Using local and remote categories on page 2745
- Web profile override on page 2747

Web rating override

Web rating overrides allow you to apply a category override to a URL. This overrides the original FortiGuard category for the URL with either a different FortiGuard category, a custom local category, or an external feed remote category.

If a URL is in multiple active categories, the order of precedence is local categories, then remote categories, and then FortiGuard categories.

Web rating override requires a FortiGuard license.

This section includes information about override configurations and examples:

- Configuring the category override rule on page 2737
- Sub-category actions on page 2739
- Category override examples on page 2740

Configuring the category override rule

This topic includes information about configuring following category types:

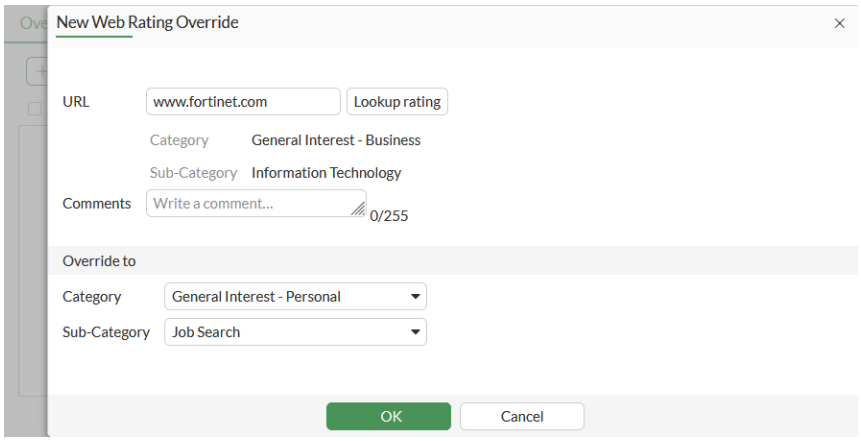
- FortiGuard category override
- Local category override
- Remote category override

Once you have configured the category override rules, ensure that the override category is active in a web filter profile. See Sub-category actions on page 2739 for more information.

To create a FortiGuard category override:

1. Go to *Security Profiles* > *Web Rating Overrides* and click *Create New*.
2. Configure the following settings:

URL	Enter the URL to override.
Lookup rating	Select to view any current *Category* and *Sub-Category* ratings.
Comments	Enter a comment (optional).
Override to	
Category	Select a FortiGuard category, external feed remote category, or a *Custom Category*.
Sub-Category	Select a sub-category to further define the rating. If *Custom Category* was selected for the *Category*, you can select from a list of categories you created.



3. Click *OK*.

See Example 1: Override a FortiGuard category with another FortiGuard category on page 2740 for a sample configuration.

To create a custom local category override:

1. Create a custom category :
   a. Go to *Security Profiles* > *Web Rating Overrides*.
   b. Click *Custom Categories*, then click *Create New*.
   c. Configure the following settings:

Name	Enter a unique name for the category.
Status	Enable/disable the status of the category.

   d. Click *OK*.
2. Create a web rating override:
   a. Go to *Security Profiles* > *Web Rating Overrides* and click *Create New*.
   b. Configure the following settings:

URL	Enter the URL to override.
Lookup rating	Select to view any current *Category* and *Sub-Category* ratings.
Comments	Enter a comment (optional).
Override to	
Category	Select *Custom Category*.
Sub-Category	Select the custom category that was just created.

   c. Click *OK*.

   See Example 3: Override a FortiGuard category with a custom local category on page 2744 for a sample configuration.

To create an external feed remote category override:

1. Go to *Security Fabric > External Connectors* and click *Create New*.
2. In the *External Feeds* section, click *FortiGuard Category*.
3. Enter a name for the external feed. This will also be the name of the remote category.
4. Enter the *URL of external resource* that contains the list of URLs that will be overridden in this remote category.
5. Configure the remaining settings as needed, then click *OK*.

See Example 2: Override a FortiGuard category with a remote category on page 2742for a sample configuration.

Sub-category actions

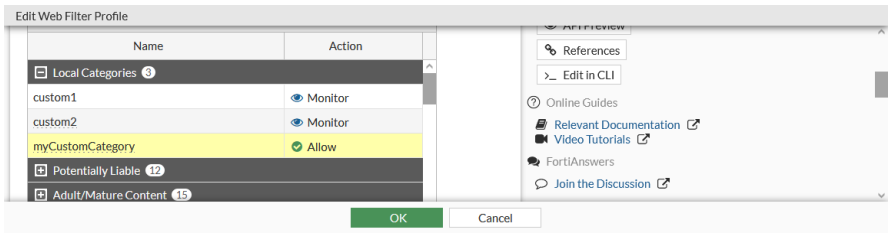
After configuring category override rules, an override category must be active in a web filter profile for it to take effect. Whether a category is active or not depends on the override method and action:

Override method	Active category actions	Inactive category actions
FortiGuard categories	Monitor, Block, Warning, or Authenticate	Allow
Local categories	Allow, Monitor, Block, Warning, or Authenticate	Disable*
Remote categories	Allow, Monitor, Block, Warning, or Authenticate	Disable*

*The *Disable* action is only available for local and remote categories by right clicking on the sub-category.

The *Allow* action in the GUI is different for FortiGuard categories compared to local and remote categories.

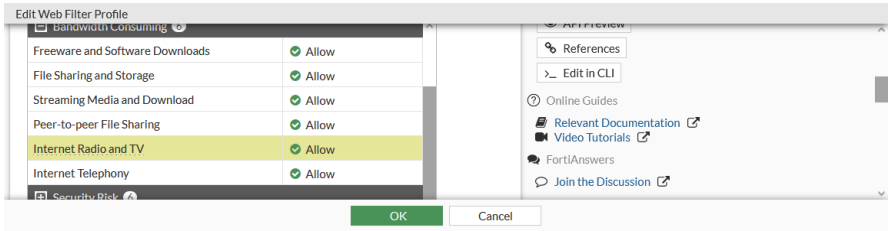
For local and remote categories, the *Allow* action in the GUI corresponds to the monitor action with logging disabled in the CLI:




```
config webfilter profile
    edit <profile>
        config ftgd-wf
            config filters
                edit 142
                    set category 142
                    set action monitor
                    set log disable
                next
            end
```

```
        end
    next
end
```

For FortiGuard categories, the *Allow* action in the GUI corresponds to no entry in the CLI:



The *Internet Radio and TV* sub-category has ID number 75.

```
config webfilter profile
    edit <profile>
        config ftgd-wf
            config filters
            end
        end
    next
end
```

This means that a FortiGuard category with the *Allow* action applied is effectively inactive, as there is no actual action specified in the CLI.

See Category override examples on page 2740 for sample configurations.

Category override examples

This topic includes examples that overrides the original FortiGuard category:

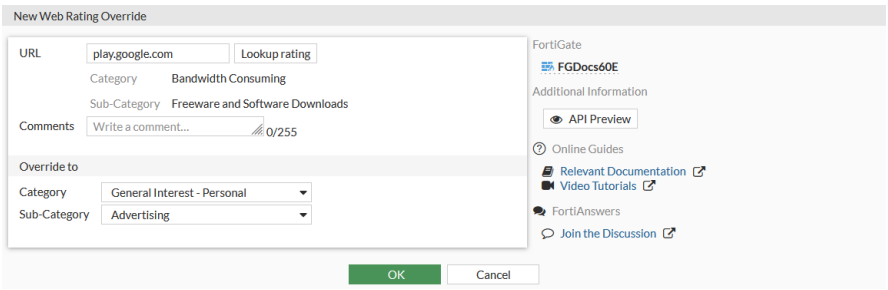
- Example 1: Override a FortiGuard category with another FortiGuard category on page 2740
- Example 2: Override a FortiGuard category with a remote category on page 2742
- Example 3: Override a FortiGuard category with a custom local category on page 2744

Example 1: Override a FortiGuard category with another FortiGuard category

In this example, play.google.com is overridden from its original category, Freeware and Software Download (19), to the Advertising category (17). In the web filter profile, the Advertising category is set to *Block* and the Freeware and Software Download category is set to *Allow*.

To configure a FortiGuard web rating override:

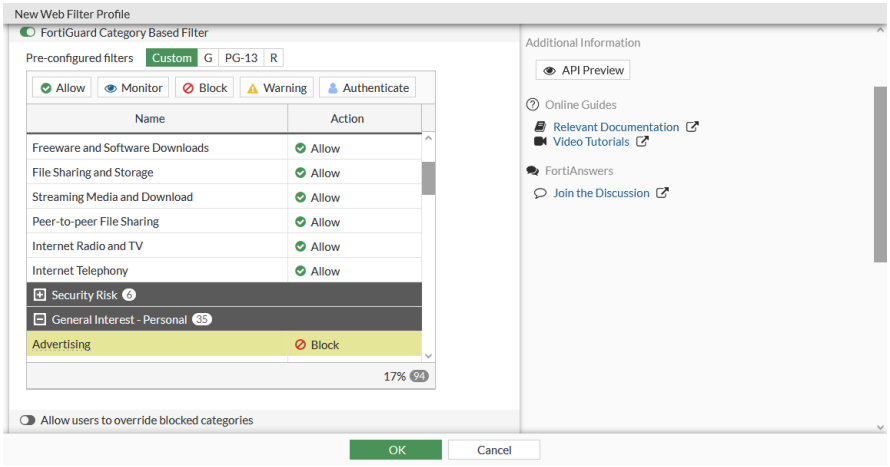
1. Go to *Security Profiles* > *Web Rating Overrides* and click *Create New*.
2. Enter the URL: play.google.com.
3. Optionally, click *Lookup rating* to see what its current rating is.
4. Set the *Category* and *Sub-Category* to an existing category that is different from the original category.



5. Click *OK*.

To apply the category in a web filter profile:

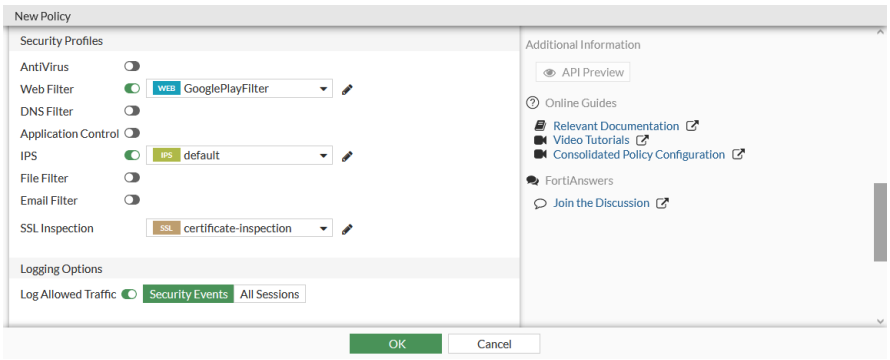
1. Go to *Security Profiles* > *Web Filter* and create or edit a web filter profile. See FortiGuard filter on page 2315 for more information.
2. Enable *FortiGuard category based filter*.
3. Set the action for the *Advertising* category in the *General Interest - Personal* group to *Block*.
4. Set the action for the *Freeware and Software Download* category in the *Bandwidth Consuming* group to *Allow*.



5. Configure the remaining settings required, then click *OK*.

To apply the category in firewall policy:

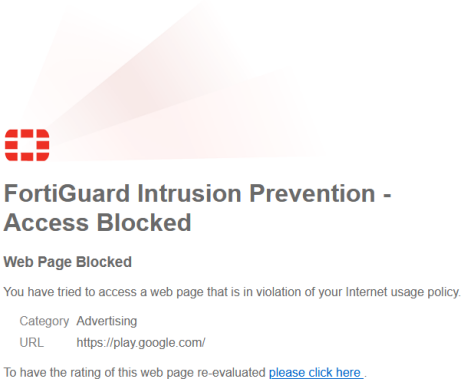
1. Go to *Policy & Objects* > *Firewall Policy* and create or edit a policy.
2. Configure the policy fields as required.
3. Under *Security Profiles*, enable *Web Filter* and select the profile that you just created.
4. Set *SSL Inspection* to *certificate-inspection* or *deep-inspection*.



5. Enable *Log Allowed Traffic*.
6. Click *OK*.

To test the filter:

1. From a Workstation behind the firewall, open a browser and browse to play.google.com. The page will be blocked by the category override.



2. Go to *Log & Report* > *Security Events* and select *Web Filter*.
3. View the log details in the GUI, or download the log file:

```
date=2022-09-21 time=16:43:31 eventtime=1663803811966781540 tz="-0700" logid="0316013056"
type="utm" subtype="webfilter" eventtype="ftgd_blk" level="warning" vd="root" policyid=2
sessionid=891040 srcip=192.168.2.8 srcport=50318 srcintf="port2" srcintfrole="undefined"
dstip=142.251.211.238 dstport=443 dstintf="port1" dstintfrole="undefined" proto=6
service="HTTPS" hostname="play.google.com" profile="FGD-Override-FGD-Flow" action="blocked"
reqtype="direct" url="https://play.google.com/" sentbyte=517 rcvdbyte=0 direction="outgoing"
msg="URL belongs to a denied category in policy" method="domain" cat=17 catdesc="Advertising"
```

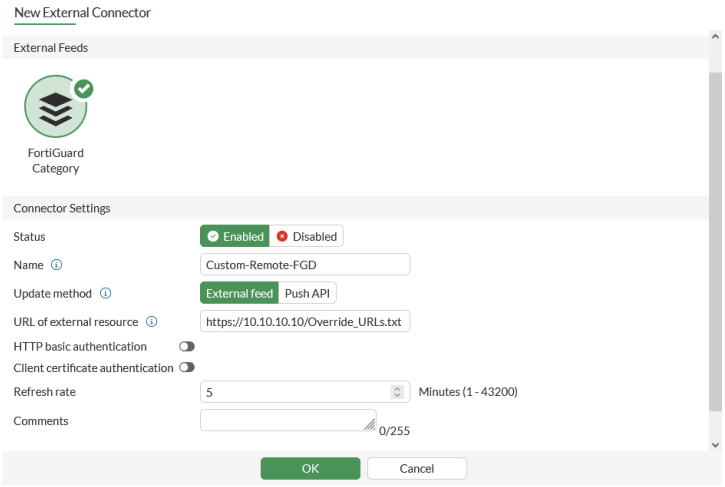
Example 2: Override a FortiGuard category with a remote category

In this example, play.google.com is added to an external URL category list and applied to an external feed. In the web filter profile, the remote category is set to *Allow*, and the original FortiGuard category (Freeware and Software Download) is set to *Block*. Remote categories take precedence over FortiGuard categories, so the override action for the remote category will apply.

Delete the web rating override entry from example 1 for play.google.com before configuring this example.

To configure a FortiGuard external feed for remote category override:

1. Go to *Security Fabric* > *External Connectors* and click *Create New*.
2. In the *External Feeds* section, click *FortiGuard Category*.
3. Enter a name for the external feed, such as *Custom-Remote-FGD*. This will be the name of the remote category.
4. Enter the *URL of external resource* that contains the list of URLs that will be overridden to this remote category. This list will contain one entry for play.google.com.



5. Configure the remaining settings as needed, then click *OK*.

To apply the category in a web filter profile:

1. Go to *Security Profiles* > *Web Filter* and create or edit a web filter profile. See FortiGuard filter on page 2315 for more information.
2. Enable *FortiGuard category based filter*.
3. Set the action for the *Custom-Remote-FGD* category in the *Remote Categories* group to *Allow*.
4. Set the action for the *Freeware and Software Download* category in the *Bandwidth Consuming* group to *Block*.
5. Configure the remaining settings are required, then click *OK*.

To apply the category in firewall policy:

1. Go to *Policy & Objects* > *Firewall Policy* and create or edit a policy.
2. Configure the policy fields as required.
3. Under *Security Profiles*, enable *Web Filter* and select the profile that you just created.
4. Set *SSL Inspection* to *certificate-inspection* or *deep-inspection*.
5. Enable *Log Allowed Traffic*.
6. Click *OK*.

To test the filter:

1. From a Workstation behind the firewall, open a browser and browse to play.google.com. The page will be allowed by the remote category override.

2. No logs are recorded because the *Allow* action is selected.

Example 3: Override a FortiGuard category with a custom local category

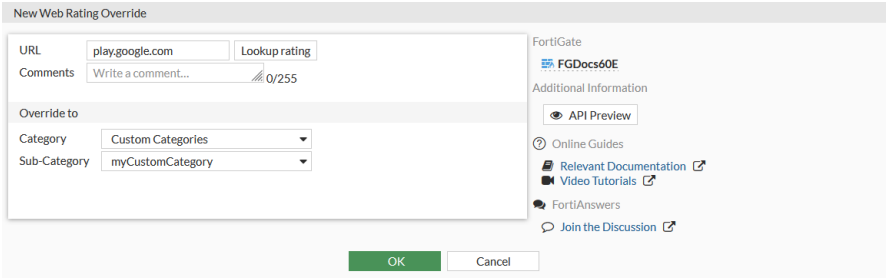
In this example, play.google.com is added to a custom local category. that is set to Monitor in the web filter profile. Local custom categories take precedence over both remote and FortiGuard categories, so the override action for the local category will apply.

To create a custom local category override:

1. Go to *Security Profiles* > *Web Rating Overrides*.
2. Click *Custom Categories*, then click *Create New*.
3. Enter a name for the category, such as *myCustomCategory*, and ensure the *Status* is set to *Enable*.
4. Click *OK*.

To create a web rating override for the custom local category:

1. Go to *Security Profiles* > *Web Rating Overrides* and click *Create New*.
2. Enter the URL to override.
3. For *Category*, select *Custom Categories* and for *Sub-Category* select *myCustomCategory*.



4. Click *OK*.

To apply the category in a web filter profile:

1. Go to *Security Profiles* > *Web Filter* and create or edit a web filter profile. See FortiGuard filter on page 2315 for more information.
2. Enable *FortiGuard category based filter*.
3. Set the action for the *myCustomCategory* category in the *LocalCategories* group to *Monitor*.
4. The other actions can be left as they were at the end of example 2, *Custom-Remote-FGD* set to *Allow* and *Freeware and Software Download* set to *Block*.
5. Configure the remaining settings are required, then click *OK*.

To apply the category in firewall policy:

1. Go to *Policy & Objects* > *Firewall Policy* and create or edit a policy.
2. Configure the policy fields as required.
3. Under *Security Profiles*, enable *Web Filter* and select the profile that you just created.
4. Set *SSL Inspection* to *certificate-inspection* or *deep-inspection*.
5. Enable *Log Allowed Traffic*.

6. Click *OK*.

To test the filter:

1. From a Workstation behind the firewall, open a browser and browse to play.google.com. The page will be allowed by the local category override.
2. Go to *Log & Report* > *Security Events* and select *Web Filter*.
3. View the log details in the GUI, or download the log file:

```
date=2022-09-21 time=17:17:00 eventtime=1663805820486294353 tz="-0700" logid="0317013312"
type="utm" subtype="webfilter" eventtype="ftgd_allow" level="notice" vd="root" policyid=2
sessionid=893147 srcip=192.168.2.8 srcport=50417 srcintf="port2" srcintfrole="undefined"
dstip=142.251.211.238 dstport=443 dstintf="port1" dstintfrole="undefined" proto=6
service="HTTPS" hostname="play.google.com" profile="FGD-Override-FGD-Flow"
action="passthrough" reqtype="direct" url="https://play.google.com/" sentbyte=517 rcvdbyte=0
direction="outgoing" msg="URL belongs to an allowed category in policy" method="domain"
cat=142 catdesc="myCustomCategory"
```

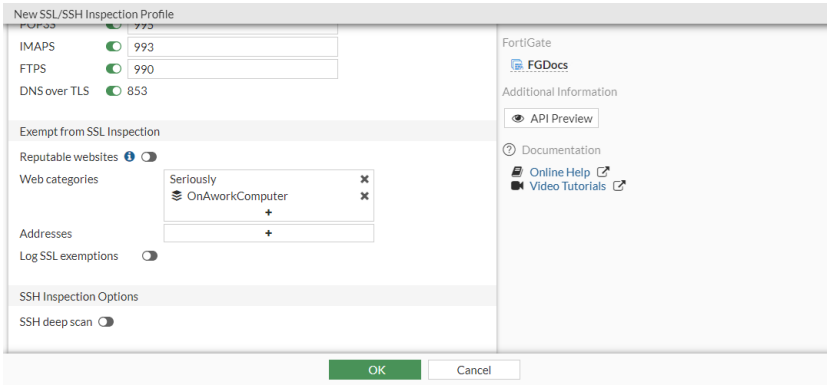
Using local and remote categories

For some functions, local and remote FortiGuard categories must be explicitly selected to apply. In SSL/SSH inspection profiles, custom categories must be explicitly selected to be exempt from SSL inspection. In Proxy addresses, custom categories must be explicitly selected as URL categories for them to apply. In both settings, if a URL is in multiple selected categories, the order of precedence is local categories, then remote categories, and then FortiGuard categories.

SSL/SSH inspection profiles

To use local and remote categories in an SSL/SSH inspection profile to exempt them from SSL inspection in the GUI:

1. Go to *Security Profiles* > *SSL/SSH Inspection*.
2. Create a new profile or edit an existing one.
3. Ensure that *Inspection method* is *Full SSL Inspection*.
4. In the *Exempt from SSL Inspection* section, add the local and remote categories to the *Web categories* list .



5. Configure the remaining settings as required, then click *OK*.

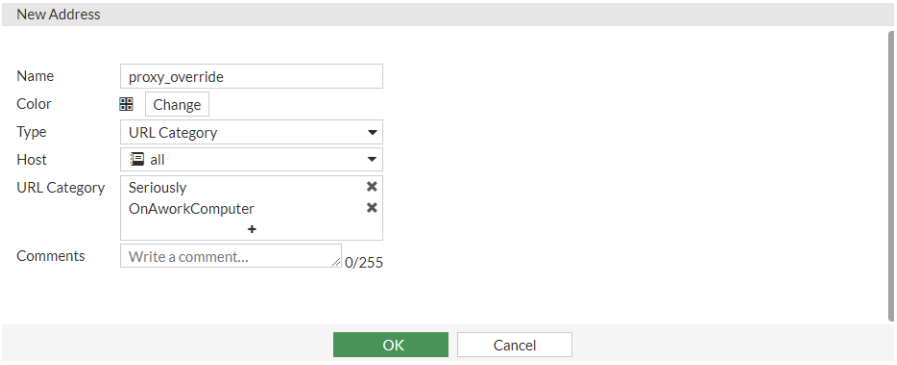
To use local and remote categories in an SSL/SSH inspection profile to exempt them from SSL inspection in the CLI:

```
config vdom
    edit root
        config firewall ssl-ssh-profile
            edit "SSL_Inspection"
                config https
                    set ports 443
                    set status deep-inspection
                end
                ...
                config ssl-exempt
                    edit 1
                        set fortiguard-category 140
                    next
                    edit 2
                        set fortiguard-category 192
                    next
                end
            next
        end
    next
end
```

Proxy addresses

To use local and remote categories in a proxy address in the GUI:

1. Go to *Policy & Objects* > *Addresses* and select *Proxy* > *Address*.
2. Click *Create new* or edit an existing proxy address.
3. Set *Type* to *URL Category*.
4. In the *URL Category*, add the local and remote categories.



5. Configure the remaining settings as required, then click *OK*.

To use local and remote categories in a proxy address in the CLI:

```
config vdom
    edit root
        config firewall proxy-address
            edit "proxy_override"
                set type category
                set host "all"
                set category 140 192
                set color 23
            next
        end
    next
end
```

Web profile override

The following profile override methods are available:

- Administrative override
- Allow users to override blocked categories

Administrative override

Administrators can grant temporary access to sites that are otherwise blocked by a web filter profile. You can grant temporary access to a user, user group, or source IP address. You can set the time limit by selecting a date and time. The default is 15 minutes.

When the administrative web profile override is enabled, a blocked access page or replacement message does not appear, and authentication is not required.

Scope range

You can choose one of the following scope ranges:

- User: authentication for permission to override is based on whether or not the user is using a specific user account.
- User group: authentication for permission to override is based on whether or not the user account supplied as a credential is a member of the specified user group.
- Source IP: authentication for permission to override is based on the IP address of the computer that was used to authenticate. This would be used for computers that have multiple users. For example, if a user logs on to the computer, engages the override by using their credentials, and then logs off, anyone who logs on with an account on that computer would be using the alternate override web filter profile.

When you enter an IP address in the administrative override method, only individual IP addresses are allowed.

Differences between IP and identity-based scope

Using the IP scope does not require using an identity-based policy.

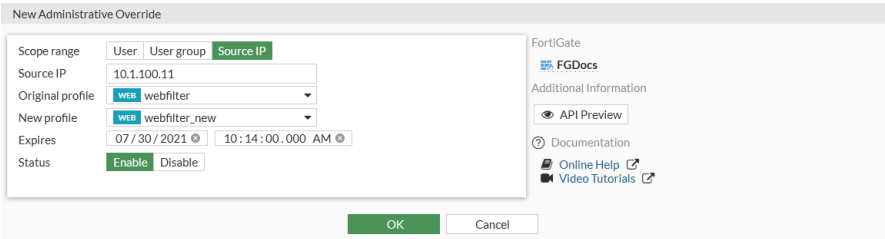
When using the administrative override method and IP scope, you might not see a warning message when you change from using the original web filter profile to using the alternate profile. There is no requirement for credentials from the user so, if allowed, the page will just appear in the browser.

Configuring a web profile administrative override

This example describes how to override the *webfilter* profile with the *webfilter_new* profile.

To configure web profile administrative override using the GUI:

1. Go to *Security Profiles* > *Web Profile Overrides* and click *Create New*.
2. Configure the administrative override:
   a. For *Scope Range*, click *Source IP*.
   b. In the *Source IP* field, enter the IP address for the client computer (*10.1.100.11* in this example).
   c. In the *Original profile* dropdown, select *webfilter*.
   d. In the *New profile* dropdown, select *webfilter_new*.
   In the *Expires* field, the default 15 minutes appears, which is the desired duration for this example.



3. Click *OK*.

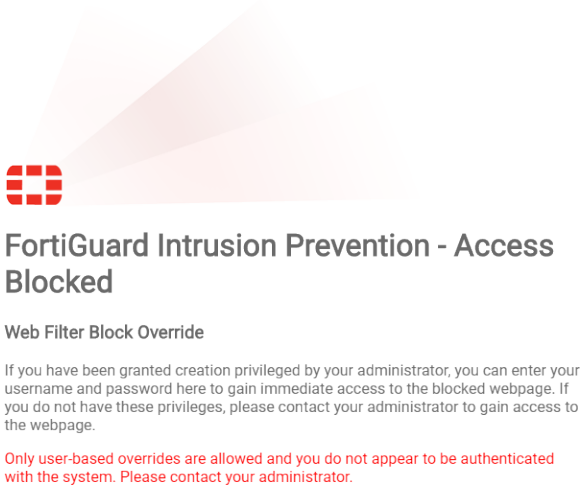
To configure web profile administrative override using the CLI:

```
config webfilter override
    edit 1
        set status enable
        set scope ip
        set old-profile "webfilter"
        set new-profile "webfilter_new"
        set expires 2021/07/30 10:14:00
        set initiator "admin"
        set ip 10.1.100.11
    next
end
```

Allow users to override blocked categories

For both override methods, the scope ranges (for specified users, user groups, or IP addresses) allow sites blocked by web filtering profiles to be overridden for a specified length of time.

But there is a difference between the override methods when the users or user group scope ranges are selected. In both cases, you would need to apply the user or user group as source in the firewall policy. With administrative override, if you do not apply the source in the firewall policy, the traffic will not match the override and will be blocked by the original profile. With the *Allow users to override blocked categories* setting, the traffic will also be blocked, but instead of displaying a blocking page, the following message appears:



When you choose the user group scope, once one user overrides, it will affect the other users in the group when they attempt to override. For example, user1 and user2 both belong to the local_user group. Once user1 successfully overrides, this will generate an override entry for the local_user group instead of one specific user. This means that if user2 logs in from another PC, they can override transparently.

Other features

Besides the scope, there are some other features in *Allow users to override blocked categories*.

Apply to user groups

Individual users can not be selected. You can select one or more of the user groups recognized by the FortiGate. They can be local to the system or from a third party authentication device, such as an AD server through FSSO.

Switch duration

Administrative override sets a specified time frame that is always used for that override. The available options are:

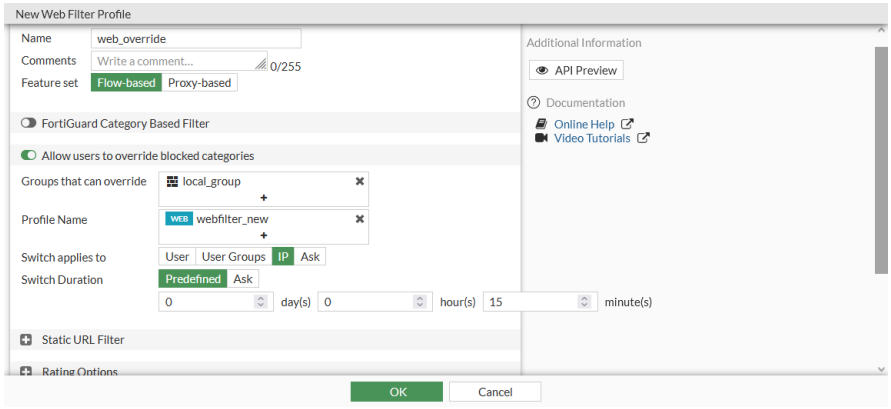
- *Predefined*: the value entered is the set duration (length of time in days, hours, or minutes) that the override will be in effect. If the duration variable is set to 15 minutes, the length of the override will always be 15 minutes. The option will be visible in the override message page, but the setting will be grayed out.
- *Ask*: the user has the option to set the override duration once it is engaged. The user can set the duration in terms of days, hours, or minutes.

Creating a web profile users override

This example describes how to allow users in the *local_group* to override the *webfilter_new* profile.

To allow users to override blocked categories using the GUI:

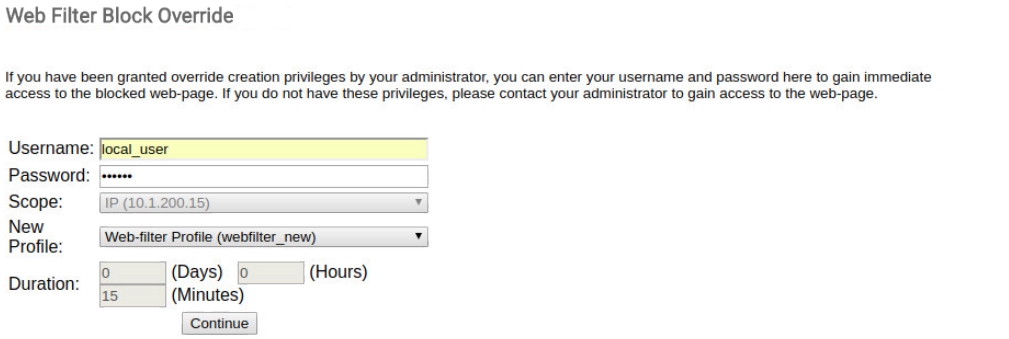
1. Go to *Security Profiles* > *Web Filter* and click *Create New*.
2. Enter a name for the profile.
3. Enable *Allow users to override blocked categories*.
4. Configure the web filter profile:
   a. Click the *Groups that can override* field, and select a group (*local_group* in this example).
   b. Click the *Profile Name* field, and select the *webfilter_new* profile.
   c. For the *Switch applies to* field, click *IP*.
   d. For the *Switch Duration* field, click *Predefined*. The default *15* minutes appears, which is the desired duration for this example.
   e. Configure the rest of the profile as needed.



5. Click *OK*.

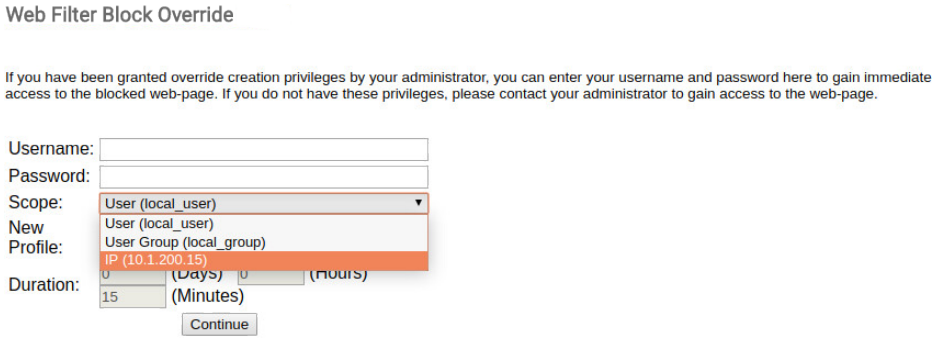
Using the ask feature

This option is only available in *Allow users to override blocked categories* is enabled. It configures the message page to have the user choose which scope they want to use. Normally on the message page, the scope options are grayed out and not editable. In the following example, the *Scope* is predefined with *IP*.



When the ask option is enabled (through the *Switch applies to* field in the GUI), the *Scope* dropdown is editable. Users can choose one of the following:

- User
- User group
- IP



User and *User Group* are only available when there is a user group in the firewall policy. You must specify a user group as a source in the firewall policy so the scope includes *User* and *User Group*; otherwise, only the IP option will be available.

IP ban

The FortiGate IP ban feature is a powerful tool for network security. It allows the system to block traffic originating from specific IP addresses that are deemed potentially harmful by the system administrator.

When an IP address is banned, any active connections originating from the banned IP address are immediately terminated. Any subsequent connection attempts are rejected by the Kernel's packet filter, further fortifying the network's security.

Checks for IP bans are carried out only if there is a corresponding firewall policy with an ACCEPT action. If a match is found, the action is then altered to DENY. In scenarios where there is no matching policy, the connection is refused due to the implicit deny rule that is in effect.

Several methods can be used to ban IP addresses:

- FortiView Source: This method allows you to ban an IP address directly from the *FortiView Sources* monitor. See To ban an IP address for more information.
- IP ban: Administrators can configure an automation stitch with the *IP Ban* action, using a trigger such as a *Compromised Host* or an *Incoming Webhook*. When the automation is triggered, the client PC is quarantined. See Actions on page 4530 and Incoming Webhook Quarantine stitch on page 4503 for more information. The Automation Stitch feature can also be used to configure IP bans from other fabric devices.
- Command line interface (CLI): For those who prefer using command line, IP ban can be added with the CLI. See IP ban using the CLI on page 2752 for more information.
- Security profiles: Most security profiles include a mechanism to ban a source IP address. See IP ban using security profiles on page 2753 for more information.

- DoS policy: A Denial of Service (DoS) policy can be used to block any further traffic from a source IP address that is considered a malicious actor. See DoS policy on page 1901 for more information.

Additionally, administrators can control whether the banned IP list remains intact through a power cycle. See Configuring the persistency for a banned IP list on page 2755 for more information.

IP ban using the CLI

Administrators can use the following command to manage the banned IP address list:

```
# diagnose user banned-ip {list | add | delete | clear | stat}
```

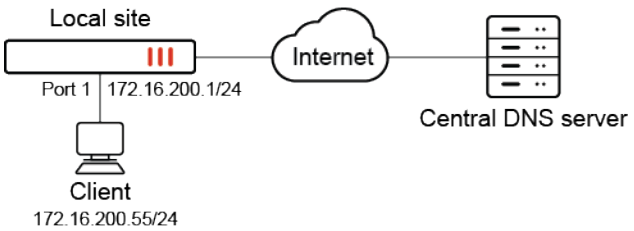
Option	Description
`list`	List banned IPs.
`add`	Add banned IP address.
`delete`	Delete banned IP address.
`clear`	Clear all banned IP addresses.
`stat`	Statistics

To add an IP address to the ban list:

```
# diagnose user banned-ip add <src4 | src6> <src-ipv4 | src-ipv6> <expiry> <admin | dlp | ips | av
| dos | app>
```

Setting the expiry time to `0` results in an indefinite expiry time. If this is combined with the `banned-ip-persistency` (either `permanent-only` or `all`), the ban becomes permanent.

Example



In this example, a client PC is configured with the IP address 172.16.200.55, and an administrator adds the IP address to the IP ban list.

To add an IP address to the ban list:

```
# diagnose user banned-ip add src4 172.16.200.55 2 admin
```

To view the banned IP list:

```
# diagnose user banned-ip list
src-ip-addr       created                  expires                  cause
172.16.200.55       Tue Jan 16 14:46:00 2024 Tue Jan 16 14:56:00 2024 Administrative
```

To verify that the banned IP list is working:

1. From the client with the banned IP address of 172.16.200.55, send a DNS query for a domain that is configured on the Central DNS server.
2. Go to *Log & Report > Forward Traffic*, and search for IP address *172.16.200.55*.
3. View the log details in the GUI, or download the log file.
   In the following log file example, action is deny for source IP 172.16.200.55.

```
date=2024-01-16 time=14:48:43 eventtime=1705445143824107713 tz="+1200" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="root" srcip=172.16.200.55
srcname="DESKTOP-644U0A1" srcport=55619 srcintf="port2" srcintfrole="undefined" dstip=8.8.8.8
dstport=53 dstintf="port1" dstintfrole="undefined" srccountry="United States"
dstcountry="United States" sessionid=259700 proto=17 action="deny" policyid=1
policytype="policy" poluuid="f4fe48a4-938c-51ee-8856-3e84e3b24af4" policyname="client_yt_v4"
service="DNS" trandisp="noop" duration=0 sentbyte=0 rcvdbyte=0 sentpkt=0 rcvdpkt=0
appcat="unscanned" crscore=30 craction=131072 crlevel="high" srchwvendor="VMware"
devtype="Server" srcfamily="Virtual Machine" osname="Windows" srcswversion="10"
mastersrcmac="00:0c:29:a6:5a:94" srcmac="00:0c:29:a6:5a:94" srcserver=0
```

IP ban using security profiles

Configure one or all of the security profiles to quarantine all traffic originating from the infected host's IP address for a configurable duration. The IP address of the attacker is also incorporated into the list of banned users. The address of the target remains unaffected.

The following types of security profiles can be used to ban IP addresses:

- Antivirus on page 2753
- Application control on page 2754
- DLP on page 2754
- IPS on page 2755

Antivirus

To ban IP addresses using an antivirus profile:

1. Configure an antivirus profile:

```
config antivirus profile
    edit <name>
    config nac-quar
```

```
    set infected quar-src-ip
    set expiry <duration>
 end
```

2. View the banned IP address:

```
# diagnose user banned-ip list
src-ip-addr       created                  expires                  cause
172.16.200.55       Wed Jan 17 13:06:05 2024 Wed Jan 17 13:08:05 2024 AV
```

Application control

Quarantine is available as one of the action types when the application matches this application control profile. See Basic category filters and overrides on page 2432 for more information.

To ban IP addresses using an application control profile:

1. Configure the application control profile:

```
config application list
    edit <name>
        config entries
            edit <id>
                set quarantine attacker
                set quarantine-expiry <duration>
            next
        end
    next
end
```

2. View the banned IP address:

```
# diagnose user banned-ip list
src-ip-addr       created                  expires                  cause     172.16.200.55
   Thu Jan 18 07:17:13 2024 Thu Jan 18 07:22:13 2024 APP
```

DLP

Quarantine is available as one of the action types when the content matches this DLP profile. See Basic DLP settings on page 2606 for more information

To ban IP addresses using a DLP profile:

1. Configure the DLP profile:

```
config dlp profile
    edit <name>
        config rule
```

```
            edit <id>
                set proto <protocols>
                set action quarantine-ip
                set expiry <duration>
            next
        end
    next
end
```

2. View the banned IP address:

```
# diagnose user banned-ip list
src-ip-addr       created                  expires                  cause     172.16.200.55
   Thu Jan 18 07:03:03 2024 Thu Jan 18 07:05:03 2024 DLP
```

IPS

Quarantine is available as one of the action types when the signature matches this IPS profile. A protocol must also be set. See Configuring an IPS sensor on page 2483 for more information.

To ban IP addresses using an IPS profile:

1. Configure the IPS profile:

```
config ips sensor
    edit <name>
        config entries
            edit <id>
                set quarantine attacker
                set quarantine-expiry <duration>
            next
        end
end
```

2. View the banned IP address:

```
# diagnose user banned-ip list
src-ip-addr       created                  expires                  cause
172.16.200.55      Thu Jan 18 06:42:06 2024 Thu Jan 18 06:44:06 2024 IPS
```

Configuring the persistency for a banned IP list

The `banned-ip-persistency` option configures whether the banned IP list persists through a power cycle.

```
config firewall global
    set banned-ip-persistency {disabled | permanent-only | all}
end
```

`banned-ip-persistency {disabled | permanent-only | all}`	Set the persistency of banned IPs across power cycling: • `disabled`: no entries are kept across power cycling (default). • `permanent-only`: only permanent IP bans are kept across power cycling. • `all`: all IP bans are kept across power cycling.

The banned IP list is created from quarantining. For example, when quarantining is enabled for IPS, application control, and DDoS. Permanent quarantining can be added manually using `diagnose user banned-ip add src4`.

The `diagnose user quarantine <parameter>` command has changed to `diagnose user banned-ip <parameter>`.

Example 1: keep all banned IPs across power cycling

When `banned-ip-persistency` is set to `all`, all the banned IPs are saved after a reboot. In this example, an application control security profile with quarantining is already configured. After traffic is generated that triggers the quarantine rule, a quarantine list is generated.

To view the list of banned IPs:

```
# diagnose user banned-ip list
src-ip-addr       created                  expires                  cause
10.1.100.12       Tue Jul  5 18:01:05 2022 Tue Jul  5 18:21:05 2022 APP
```

After a reboot, the banned IP list is the same:

```
# diagnose user banned-ip list
src-ip-addr       created                  expires                  cause
10.1.100.12       Tue Jul  5 18:01:05 2022 Tue Jul  5 18:21:05 2022 APP
```

Example 2: keep only permanent banned IPs across power cycling

When `banned-ip-persistency` is set to `permanent-only`, only banned IPs with an indefinite expiry time are saved after a reboot. The permanent IP ban was already configured for 10.1.100.11 using `diagnose user banned-ip add src4 10.1.100.11 0 ips`.

To view the list of banned IPs:

```
# diagnose user banned-ip list
src-ip-addr       created                  expires                  cause
10.1.100.12       Tue Jul  5 18:01:05 2022 Tue Jul  5 18:21:05 2022 APP
10.1.100.11       Tue Jul  5 18:06:35 2022 indefinite               IPS
```

After a reboot, only 10.1.100.11 remains in the banned IP list:

```
# diagnose user banned-ip list
src-ip-addr       created                  expires                  cause
10.1.100.11       Tue Jul  5 18:06:35 2022 indefinite               IPS
```

Profile groups

Security profiles can be organized into groups. They are useful when there are multiple policies that use the same security profiles, helping save time and preventing missing profiles when configuring policies. When changes need to be made, only the group has to be changed and not the individual policies.

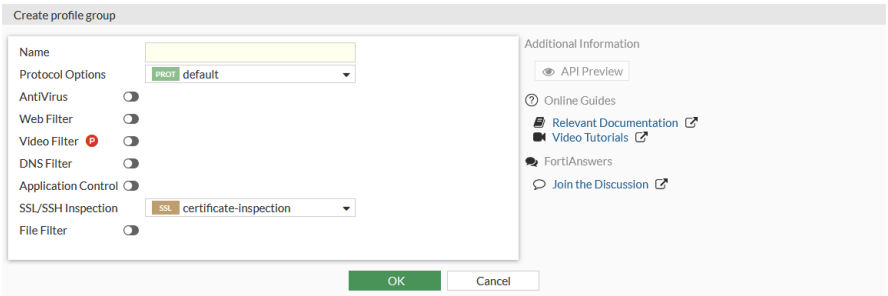
By default, *Security Profiles* > *Profile Groups* is not visible in the GUI. It can only be enabled using the CLI.

To show profile groups in the GUI:

```
config system settings
    set gui-security-profile-group enable
end
```

To configure a profile group in the GUI:

1. Go to *Security Profiles* > *Profile Groups* and click *Create New*.



2. Enter a name for the group.
3. Enable the required profile types and select the profile that will be included in the group.
   A *Protocol Option* must be selected.
4. Click *OK*.

To configure a profile group in the CLI:

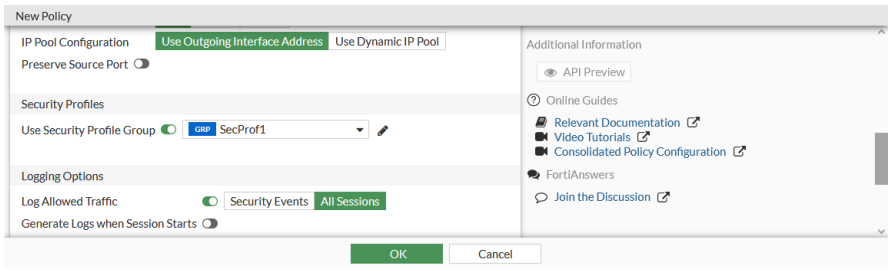
```
config firewall profile-group
    edit <name>
        set application-list <string>
        set av-profile <string>
        set casb-profile <string>
        set cifs-profile <string>
        set diameter-filter-profile <string>
        set dlp-profile <string>
        set dnsfilter-profile <string>
        set emailfilter-profile <string>
        set file-filter-profile <string>
        set icap-profile <string>
        set ips-sensor <string>
        set profile-protocol-options <string>
```

```
        set sctp-filter-profile <string>
        set ssh-filter-profile <string>
        set ssl-ssh-profile <string>
        set videofilter-profile <string>
        set virtual-patch-profile <string>
        set voip-profile <string>
        set waf-profile <string>
        set webfilter-profile <string>
    next
end
```

`application-list <string>`	Name of an existing application list.
`av-profile <string>`	Name of an existing antivirus profile.
`casb-profile <string>`	Name of an existing CASB profile.
`cifs-profile <string>`	Name of an existing CIFS profile.
`diameter-filter-profile <string>`	Name of an existing Diameter filter profile.
`dlp-profile <string>`	Name of an existing DLP profile.
`dnsfilter-profile <string>`	Name of an existing DNS filter profile.
`emailfilter-profile <string>`	Name of an existing email filter profile.
`file-filter-profile <string>`	Name of an existing file-filter profile.
`icap-profile <string>`	Name of an existing ICAP profile.
`ips-sensor <string>`	Name of an existing IPS sensor profile.
`profile-protocol-options <string>`	Name of an existing protocol options profile (default = `default`).
`sctp-filter-profile <string>`	Name of an existing SCTP filter profile.
`ssh-filter-profile <string>`	Name of an existing SSH filter profile.
`ssl-ssh-profile <string>`	Name of an existing SSL SSH profile (default = `certificate-inspection`).
`videofilter-profile <string>`	Name of an existing video filter profile.
`virtual-patch-profile <string>`	Name of an existing virtual-patch profile.
`voip-profile <string>`	Name of an existing VOIP profile.
`waf-profile <string>`	Name of an existing WAF profile.
`webfilter-profile <string>`	Name of an existing web filter profile.

To use the profile group in a policy in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy* and edit an existing policy or create a new one.
2. In the *Security Profiles* section, enable *Use Security Profile Group* and select a group.
   No individual profiles can be selected if using a profile group.



3. Click *OK*.

To use the profile group in a policy in the CLI:

```
config firewall policy
    edit <policyid>
        set name <string>
        set srcintf <interface(s)>
        set dstintf <interface(s)>
        set action {accept | deny | ipsec}
        set srcaddr <address(es)>
        set dstaddr <address(es)>
        set schedule <schedule>
        set service <service(s)>
        set utm-status enable
        set profile-type group
        set profile-group <group>
    next
end
```

IPsec VPN

Virtual Private Network (VPN) technology lets remote users connect to private computer networks to gain access to their resources in a secure way. For example, an employee traveling or working at home can use a VPN to securely access the office network through the internet.

Instead of remotely logging into a private network using an unencrypted and unsecured internet connection, using a VPN ensures that unauthorized parties cannot access the office network and cannot intercept information going between the employee and the office. Another common use of a VPN is to connect the private networks of multiple offices.

IPsec VPN uses the Internet Protocol Security (IPsec) protocol to create encrypted tunnels on the internet. The IPsec protocol operates at the network layer of the OS model and runs on top of the IP protocol, which routes packets. All transmitted data is protected by the IPsec tunnel.

The following sections provide instructions on configuring IPsec VPN connections in FortiOS 8.0.0.

- General IPsec VPN configuration on page 2760
- Site-to-site VPN on page 2857
- Remote access on page 2961
- Aggregate and redundant VPN on page 3186
- ADVPN on page 3235
- Fabric Overlay Orchestrator on page 3277
- Other VPN topics on page 3302
- VPN IPsec troubleshooting on page 3335

General IPsec VPN configuration

The following sections provide instructions on general IPsec VPN configurations:

- Network topologies on page 2761
- Phase 1 configuration on page 2761
- Phase 2 configuration on page 2795
- Post-Quantum readiness on page 2820
- VPN security policies on page 2840
- Blocking unwanted IKE negotiations and ESP packets with a local-in policy on page 2852
- Configurable UDP port for IKE on page 2802
- Renaming IPsec tunnels on page 2854

Network topologies

The topology of your network will determine how remote peers and clients connect to the VPN and how VPN traffic is routed.

Topology	Description
Site-to-Site	Standard one-to-one VPN between two FortiGates. See Site-to-site VPN on page 2857.
Hub and spoke/ADVPN	One central FortiGate (hub) has multiple VPNs to other remote FortiGates (spokes). In ADVPN, shortcuts can be created between spokes for direct communication. See ADVPN on page 3235.
FortiClient dialup	Typically remote FortiClient dialup clients use dynamic IP addresses through NAT devices. The FortiGate acts as a dialup server allowing dialup VPN connections from multiple sources. See FortiClient as dialup client on page 2973.
FortiGate dialup	Similar to site-to-site except one end is a dialup server and the other end is a dialup client. This facilitates scenarios in which the remote dialup end has a dynamic address, or does not have a public IP, possibly because it is behind NAT. See FortiGate as dialup client on page 2962.
Aggregate VPN	Natively support aggregating multiple VPN tunnels to increase performance and provide redundancy over multiple links. See Packet distribution and redundancy for aggregate IPsec tunnels on page 3205.
Redundant VPN	Options for supporting redundant and partially redundant IPsec VPNs, using route-based approaches. See Redundant hub and spoke VPN on page 3228.
L2TP over IPsec	Configure VPN for Microsoft Windows dialup clients using the built in L2TP software. Users do not have to install any Fortinet software. See Native VPN as dialup client - Windows New on page 3043.
GRE over IPsec	Legacy support for routers requiring point-to-point GRE over IPsec for tunneling. See GRE over IPsec on page 2911.

Phase 1 configuration

Phase 1 configuration primarily defines the parameters used in IKE (Internet Key Exchange) negotiation between the ends of the IPsec tunnel. The local end can be an endpoint client or a FortiGate interface that initiates the IKE negotiations. The remote end is the remote gateway that responds and exchanges messages with the initiator. Hence, they are sometimes referred to as the initiator and responder. The purpose of phase 1 is to secure a tunnel with one bi-directional IKE SA (security association) for negotiating IKE phase 2 parameters.

The `auto-negotiate` and `negotiation-timeout` commands control how the IKE negotiation is processed when there is no traffic, and the length of time that the FortiGate waits for negotiations to occur.

IPsec tunnels can be configured in the GUI using the *VPN Wizard*. Go to *VPN* > *VPN Wizard*. The wizard includes several templates (site-to-site, hub and spoke, remote access, and FortiClient Secure Internet Access), but a custom tunnel can be configured with the following settings.

> The IPsec phase 1 interface type cannot be changed after it is configured. This is due to the tunnel ID parameter (`tun_id`), which is used to match routes to IPsec tunnels to forward traffic. If the IPsec phase 1 interface type needs to be changed, a new interface must be configured.

Name		Phase 1 definition name. The maximum length is 15 characters for an interface mode VPN. FortiGate also uses the name for the virtual IPsec interface that it creates automatically.
Network		
	IP Version	Protocol, either IPv4 or IPv6. > FortiClient 7.4.4 does not support IPv6. Use FortiClient 7.4.6 or later.
	Remote Gateway	Category of the remote connection: • *Static IP Address*: the remote peer has a static IP address. • *Dialup User*: one or more FortiClient, FortiGate, or remote dialup clients will connect to the FortiGate. • *Dynamic DNS*: a remote peer that has a domain name and subscribes to a dynamic DNS service will connect to the FortiGate.
	IP Address	The IP address of the remote peer. This option is only available when the *Remote Gateway* is *Static IP Address*.
	Dynamic DNS	The domain name of the remote peer. This option is only available when the *Remote Gateway* is *Dynamic DNS*.
	Interface	The interface through which remote peers or dialup clients connect to the FortiGate. This option is only available in NAT mode.
	Local Gateway	IP address for the local end of the VPN tunnel. When this setting is disabled, 0.0.0.0 is used. • *Primary IP*: defaults to the IP address of the interface chosen as the *Remote Gateway* interface. • *Secondary IP*: secondary address of the interface selected in the *Interface* field. • *Specify*: manually enter an address. Interface mode cannot be configured in a transparent mode VDOM.
	Mode Config	Enable in Dialup VPNs to select the IP stack (IPv4 and/or IPv6), select the method in which client IPs are assigned and configure the client IP address range, subnet mask/prefix length, DNS server, and split tunnel capability to automate remote client addressing.

Transport	By default, IPsec uses encapsulating security payload (ESP) for encrypting data in the VPN session. However, because it uses IP protocol 50 without port numbers, when traversing over a NAT device the packets cannot be demultiplexed. NAT traversal encapsulates the ESP packets inside a UDP packet, thereby adding a unique source port to the packet, allowing the NAT device to map the packet to the correction session. Furthermore, FortiGate also allows ESP and IKE packets to be encapsulated in TCP and encrypted in TLS. See Encapsulate ESP packets within TCP headers on page 2809. Transport mode is configurable only when the *Remote gateway* type is *Static*. In *Static* mode, the Transport is defaulted to UDP. Go to *VPN > VPN Tunnels > Settings* to enable VPN over TCP to allow setting the Transport mode to *Auto*. When Transport mode is in *Auto*, UDP will be attempted first. If the transport threshold has been reached, then connection over TCP will be attempted. When the *Remote gateway* type is *Dialup* and VPN over TCP is allowed, it will behave as if Transport mode is *Auto*.
NAT Traversal	• *Enable*: a NAT device exists between the local FortiGate and the VPN peer or client. Outbound encrypted packets are wrapped inside a UDP IP header that contains a port number. The local FortiGate and the VPN peer or client must have the same NAT traversal setting (both selected or both cleared) to connect reliably. When in doubt, enable NAT traversal. • *Disable*: disable the NAT traversal setting. • *Forced*: the FortiGate will use a port value of zero when constructing the NAT discovery hash for the peer. This causes the peer to think it is behind a NAT device, and it will use UDP encapsulation for IPsec, even if no NAT is present. This approach maintains interoperability with any IPsec implementation that supports the NAT-T RFC.
Keepalive Frequency	Keepalive frequency setting. This option is only available when *NAT Traversal* is set to *Enable* or *Forced*. The NAT device between the VPN peers may remove the session when the VPN connection remains idle for too long. The value represents an interval in seconds where the connection will be maintained with periodic keepalive packets. The keepalive interval must be smaller than the session lifetime value used by the NAT device. The keepalive packet is a 138-byte ISAKMP exchange.
Dead Peer Detection	Reestablishes VPN tunnels on idle connections and cleans up dead IKE peers if required. This feature minimizes the traffic required to check if a VPN peer is available or unavailable (dead). The available options are: • *Disable*: disable dead peer detection (DPD). • *On Idle*: triggers DPD when IPsec is idle. • *On Demand*: Passively sends DPD to reduce load on the firewall. Only triggers DPD when IPsec outbound packets are sent, but no reply is received from the peer. When there is no traffic and the last DPD-ACK

		has been received, IKE will not send DPDs periodically. Notifications are received whenever a tunnel goes up or down, or to keep the tunnel connection open when no traffic is being generated inside the tunnel. For example, in scenarios where a dialup client or dynamic DNS peer connects from an IP address that changes periodically, traffic may be suspended while the IP address changes. When *Dead Peer Detection* is selected, optionally specify a retry count and a retry interval using `dpd-retrycount` and `dpd-retryinterval`. See Dead peer detection on page 2772.
	Forward Error Correction	Enable on both ends of the tunnel to correct errors in data transmission by sending redundant data across the VPN.
	EMS SN verification	Ensures that only licensed FortiClient endpoints can establish an IPsec VPN connection with FortiGate. Both the FortiGate and FortiClient endpoints must be connected to the same FortiClient EMS
Advanced network settings		
	Add route	In a dialup VPN, add a route to a peer destination selector when a dynamic tunnel is negotiated. See Dynamic IPsec route control on page 2773.
	Auto discovery sender	In ADVPN, enable or disable sending of auto-discovery short-cut messages
	Auto discovery receiver	In ADVPN, enable or disable accepting auto-discovery short-cut messages
	Auto discovery forwarder	In ADVPN, enable or disable forwarding auto-discovery short-cut messages
	Exchange interface IP	A Fortinet-specific setting to allow two FortiGates to exchange their overlay tunnel IP. This allows a dialup server to learn the tunnel IP of each client, or in ADVPN, it allow spokes to learn others' tunnel IP during shortcut negotiation.
	Device creation	When enabled, a dynamic interface (network device) is created for each dialup tunnel.
	Aggregate member	When enabled, the tunnel can be used as an aggregate member candidate.
Authentication		
	Method	Either *Pre-shared Key* or *Signature*.
	Pre-shared Key	The pre-shared key that the FortiGate will use to authenticate itself to the remote peer or dialup client during phase 1 negotiations. The same key must be defined at the remote peer or client. See Pre-shared key.
	Certificate Name	When Authentication *Method* is *Signature*, select the server certificate that the FortiGate will use to authenticate itself to the remote peer or dialup client during phase 1 negotiations. See Digital certificates.

IKE	Either *Version 1* or *Version 2*. See Choosing IKE version 1 and 2 on page 2774. Note: When using FortiClient for remote access, IKEv1 is not supported on FortiClient 7.4.4 and later.
Mode	This option is only available when IKEv1 is selected. The two available options are: • *Aggressive*: the phase 1 parameters are exchanged in a single message with unencrypted authentication information. • *Main (ID protection)*: the phase 1 parameters are exchanged in multiple rounds with encrypted authentication information. When the remote VPN peer has a dynamic IP address and is authenticated by a pre-shared key, you must select *Aggressive* mode if there is more than one dialup phase 1 configuration for the interface IP address. When the remote VPN peer has a dynamic IP address and is authenticated by a certificate, you must select *Aggressive* mode if there is more than one phase 1 configuration for the interface IP address and these phase 1 configurations use different proposals.
Accepted Peer ID	Options to authenticate VPN peers or clients depending on the *Remote Gateway* and *Authentication Method* settings.
Any peer ID	Accepts the local ID of any remote VPN peer or client. The FortiGate does not check identifiers (local IDs). *Mode* can be set to *Aggressive* or *Main* in IKEv1. This option can be used with digital certificate authentication, but for higher security, use *Peer certificate*.
Specific peer ID	This option is only applicable when *Aggressive Mode* is enabled in IKEv1. Enter the identifier that is used to authenticate the remote peer. The identifier must match the local ID configured by the remote peer's administrator. For more details, see Customizing IPsec tunnel settings. If the remote peer is a FortiGate, the identifier is specified in the *Local ID* field of the *Phase 1 Proposal* settings. If the remote peer is a FortiClient user, the identifier is specified in the *Local ID* field. In circumstances where multiple remote dialup VPN tunnels exist, each tunnel must have a peer ID set. In IKEv2, use *Network ID*.
Peer ID from dialup group	Authenticate multiple FortiGate or FortiClient dialup clients that use unique identifiers and unique pre-shared keys (or unique pre-shared keys only) through the same VPN tunnel. You must create a dialup user group for authentication purposes. Select the group from the list next to the *Peer ID from dialup group* option. You must set *Mode* to *Aggressive* in IKEv1 when the dialup clients use unique identifiers and unique pre-shared keys. If the dialup clients use unique pre-shared keys only, you can set *Mode* to *Main* if there is only one dialup Phase 1 configuration for this interface IP address.

Peer certificate	Define the CA certificate used to authenticate the remote peer when the authentication mode is *Signature*. If the FortiGate will act as a VPN client, and you are using security certificates for authentication, set the *Local ID* to the distinguished name (DN) of the local server certificate that the FortiGate unit will use for authentication purposes.
Peer certificate group	Select a group from the User Peer Group.

Remote gateway matching	FortiOS supports remote gateway matching in a dial-up IPsec tunnel connection based on the public source IP, geolocation or Security Posture of dial-up clients. When a dial-up client first makes an IPsec connection to the FortiGate VPN gateway, the FortiGate will use the source IP to match the IPsec tunnel based on the IP subnet, address range, or country/region defined for that IPsec tunnel. IPv4 and IPv6 are supported. This feature requires the dynamic (dial-up) tunnel to be defined in IKEv2.
	For geolocation-based matching, see Matching IPsec tunnel gateway based on address parameters.
	
	To configure remote gateway matching:
	
	`config vpn ipsec phase1-interface`
	`edit <name>`
	`set type dynamic`
	`set ike-version 2`
	`set remote-gw-match {any | ipmask | iprange | geography | ztna}`
	`next`
	`end`

Options	Description
`any`	Match any connecting clients.
`ipmask`	Use `src-ip` of the connecting client to match IP subnet of remote VPN gateway. You can then define the IP subnet by setting `remote-gw-subnet`.
`iprange`	Use `src-ip` of the connecting client to match IP range of remote VPN gateway. You can then define the IP address range by setting `remote-gw-start-ip` and `remote-gw-end-ip`.

Options	Description
`geography`	Use `src-ip` of the connecting client to match the specified country/region of the VPN gateway. You can then define the country by setting `remote-gw-country`.
`ztna`	Use `ztna` of the connecting client to match against ZTNA security posture tags. You can then define the tags by setting `remote-gw-ztna-tags`. Alternatively, select the security posture tag(s) directly in the GUI.

Network Overlay

This option is only available when IKEv2 is selected.

In a SD-WAN configuration where a Hub has multiple dialup VPN overlays across different interfaces, enable Network overlay and assign a Network ID to the VPN tunnel to identify the tunnels.

This can also be used to differentiate multiple dial-up VPNs terminating on the same interface and local IP address. The Network ID is used to VPN gateway lookup since it is included in the first packet from the initiator.

Finally, in ADVPN, use the auto-discovery-crossover setting to enable or disable shortcuts to be setup between different network IDs.

Network ID

When *Network Overlay* is enabled, specify the *Network ID*.

The network ID is a Fortinet-proprietary attribute that is used to select the correct phase 1 between IPsec peers, so that multiple IKEv2 tunnels can be established between the same local/remote gateway pairs.

In a dial-up VPN, `network-id` is in the first initiator message of an IKEv2 phase 1 negotiation. The responder (Hub) uses the `network-id` to match a phase 1 configuration with a matching `network-id`. The Hub can then differentiate multiple dial-up phase 1s that are bound to the same underlay interface and IP address. Without a `network-id`, the Hub cannot have multiple phase 1 dialup tunnels on the same interface.

In static phase 1 configurations, `network-id` is used with the pair of gateway IPs to negotiate the correct tunnel with a matching `network-id`. This allows IPsec peers to use the same pair of underlay IPs to establish multiple IPsec tunnels. Without it, only a single tunnel can be established over the same pair of underlay IPs.

XAUTH

This option supports the authentication of dialup clients. It is only available for IKE version 1.

- *Disable*: do not use XAuth.
- *XAuth client*: available only if the *Remote Gateway* is set to *Static IP Address* or *Dynamic DNS*. If the FortiGate is a dialup client, enter the user name and password for the FortiGate to authenticate itself to the remote XAuth server.
- *PAP Server*, *CHAP Server*, *Auto Server*: available only if *Remote Gateway* is set to *Dialup User*. Dialup clients authenticate as members

	of a dialup user group. A user group must be created first for the dialup clients that need access to the network behind the FortiGate. The FortiGate must be configured to forward authentication requests to an external RADIUS or LDAP authentication server. Select the server type based on the encryption method used between the FortiGate, the XAuth client, and the external authentication server. Then select the user group (*Inherit from policy* or *Specify*). See XAuth user authentication on page 2779.

EAP	In IKEv2, use EAP to configure user authentication.
	• IKEv2 IDi payload: Client uses the Identification – Initiator (IDi) payload in the IKE_AUTH exchange to identify itself using unique IDs like IP address, email address, or FQDN.
	• EAP identity request: Request the client to send their identity when authentication is initiated. Use this method when using EAP-MSCHAPv2, EAP-TLS with RADIUS, and LDAP.
	IPsec IKEv2 VPNs support certificate authentication and EAP authentication at the same time from a dialup FortiClient. With the eap-cert-auth setting enabled, FortiGate will validate the X.509 certificate sent from FortiClient when establishing an IKEv2 tunnel. After it is authenticated, FortiGate will continue to perform EAP authentication.
	`config vpn ipsec phase1-interface`
	`edit <tunnel>`
	`set eap enable`
	`set eap-cert-auth {*disable | enable}`
	`set peer <user peer>`
	`next`
	`end`
	See VPN 2FA with EAP and certificate authentication on page 3019.

User Group	When EAP is chosen, you can specify a single user group directly in the VPN phase 1 configurations, or inherit the user groups from the firewall policy.
Phase 1 Proposal	The encryption and authentication algorithms used to generate keys for the IKE SA. There must be a minimum of one combination. The remote peer or client must be configured to use at least one of the proposals that you define.
Encryption	The following symmetric-key encryption algorithms are available: • *DES*: Digital Encryption Standard, a 64-bit block algorithm that uses a 56-bit key. Not Recommended. • *3DES*: triple-DES; plain text is encrypted three times by three keys. Not Recommended. • *AES128*: Advanced Encryption Standard, a 128-bit block algorithm that uses a 128-bit key. • *AES128GCM*: AES in Galois/Counter Mode, a 128-bit block algorithm that uses a 128-bit key. Only available for IKEv2.

	• *AES192*: a 128-bit block algorithm that uses a 192-bit key. • *AES256*: a 128-bit block algorithm that uses a 256-bit key. • *AES256GCM*: AES in Galois/Counter Mode, a 128-bit block algorithm that uses a 256-bit key. Only available for IKEv2. • *CHACHA20POLY1305*: a 128-bit block algorithm that uses a 128-bit key and a symmetric cipher. Only available for IKEv2. See also HMAC settings. • *SM4*: a 128-bit block cipher that uses a 128-bit key and a symmetric cipher. See SM3/SM4 cryptographic algorithms for IKEv1/ IKEv2 NEW on page 2924 for model limitations. *DES* and *3DES* are considered insecure by Fortinet's Security Rating Insights. If a security rating warning appears, disable the use of the Encryption algorithms.
Authentication	The following message digests that check the message authenticity during an encrypted session are available: • *MD5*: message digest 5. Not Recommended. • *SHA1*: secure hash algorithm 1; a 160-bit message digest. Not Recommended. • *SHA256*: a 256-bit message digest. • *SHA384*: a 384-bit message digest. • *SHA512*: a 512-bit message digest. • *SM3*: a cryptographic hash function that produces a 256 bit digest. See SM3/SM4 cryptographic algorithms for IKEv1/ IKEv2 NEW on page 2924 for model limitations. In IKEv2, encryption algorithms include authentication, but a PRF (pseudo random function) is still required (*PRFSHA1*, *PRFSHA256*, *PRFSHA384*, *PRFSHA512*). See also HMAC settings. MD5 and SHA1 are considered insecure by Fortinet's Security Rating Insights. If a security rating warning appears, disable the use of the Authentication algorithms.
Diffie-Hellman Groups	Asymmetric key algorithms used for public key cryptography. Select one or more from groups 1, 2, 5, and 14 through 32. At least one of the *Diffie-Hellman Groups* (DH) settings on the remote peer or client must match one the selections on the FortiGate. Failure to match one or more DH groups will result in failed negotiations. DH groups 1, 2, 5, 14, 22-27 are considered insecure by Fortinet's Security Rating Insights. If a security rating warning appears, disable the use of these DH groups. By default, FortiGate uses DH group 20 and 21 when configuring a VPN from the GUI or CLI.
Post Quantum Cryptography Additional Key Exchanges	Enable quantum-resistant encryption by applying hybrid key exchange per RFC 9370 and RFC 9242. When creating a new PQC Additional KE group: • *Transform Type*: Each Transform type represents an additional round of KE exchange. Choose up to seven rounds or groups

	• *KE Groups*: Choose up to three Key Exchange groups from these Key Exchange Mechanisms (KEMs): *ML KEM (Kyber)*, *BIKE*, *HQC*, and *FRODO*. For more information, see Post-Quantum Cryptography for IPsec key exchange on page 2822.
Key Lifetime	The time (in seconds) that must pass before the IKE encryption key expires. When the key expires, a new key is generated without interrupting service. The keylife can be from 120 to 172 800 seconds.
Local ID	Optional setting. This value must match the peer ID value given for the remote VPN peer's *Peer Options*. • If the FortiGate will act as a VPN client and you are using peer IDs for authentication purposes, enter the identifier that the FortiGate will supply to the VPN server during the phase 1 exchange. • If the FortiGate will act as a VPN client and you are using security certificates for authentication, select the distinguished name (DN) of the local server certificate that the FortiGate will use for authentication purposes.

Additional CLI configurations

The following phase 1 settings can be configured in the CLI:

GRE over IPsec	Packets with a GRE header are encapsulated within IPsec tunnel mode.
	To configure GRE over IPsec:
	`config vpn ipsec {phase1-interface | phase1}`
	`edit ipsec`
	`set interface <name>`
	`set encapsulation gre`
	`set encapsulation-address ike/ipv4/ipv6`
	`set encap-local-gw4 xxx.xxx.xxx.xxx`
	`set encap-remote-gw xxx.xxx.xxx.xxx`
	`next`
	`end`

IPsec tunnel idle timer	Define an idle timer for IPsec tunnels. When no traffic has passed through the tunnel for the configured `idle-timeout` value, the IPsec tunnel will be flushed.
	To configure IPsec tunnel idle timeout:
	`config vpn ipsec phase1-interface`
	`edit p1`
	`set idle-timeout [enable | disable]`
	`set idle-timeoutinterval <integer>` *`IPsec tunnel idle timeout`*

	in minutes (10 - 43200). next end
Monitor tunnel for failover	Monitor a site-to-site tunnel to guarantee operational continuity if the primary tunnel fails. Configure the secondary phase 1 interface to monitor the primary interface. **To configure the monitor:** config vpn ipsec phase1-interface edit <secondary phase1-interface> set monitor <primary phase1-interface> next end
Passive mode	Passive mode turns one side of the tunnel to be a responder only. It does not initiate VPN tunnels either by auto-negotiation, rekey, or traffic initiated behind the FortiGate. **To configure passive mode:** config vpn ipsec phase1-interface edit <example> set rekey {enable | disable} set passive-mode {enable | disable} set passive-tunnel-interface {enable | disable} next end

Show only recommended IKE proposals by default

FortiGate by default only shows recommended configurations for IKE proposals in the CLI. This helps administrators more easily define secured and recommended cryptographic algorithms in the VPN configurations that adheres to security best practices.

The following DH groups are available in the CLI by default:

- 15, 16, 17, 18, 19, 20, 21, 28, 29, 30, 31, 32 (1,2,5,14,27 are hidden)

The following encryption algorithms are available in the CLI by default:

- AES128_CBC
- AES128_GCM
- AES192_CBC
- AES256_CBC
- AES256_GCM
- CHACHA20POLY1305

The following authentication algorithms are available in the CLI by default:

- SHA256
- SHA384
- SHA512

To set your preference for displaying IKE proposals:

```
config system settings
    set ike-proposal-visibility {recommended | all}
end
```

This setting is set to `recommended` by default.

Dead peer detection

By default, dead peer detection (DPD) sends probe messages every five seconds. If you are experiencing high network traffic, you can experiment with increasing the ping interval. However, longer intervals will require more traffic to detect dead peers, which will result in more traffic.

In a dynamic (dialup) connection, the *On Idle* option encourages dialup server configurations to more proactively delete tunnels if the peer is unavailable.

In the GUI, the dead peer detection option can be configured when defining phase 1 options. The following CLI commands support additional options for specifying a retry count and a retry interval.

For example, enter the following to configure DPD on the existing IPsec phase 1 configuration to use 15-second intervals and to wait for three missed attempts before declaring the peer dead and taking action.

To configure DPD:

```
config vpn ipsec phase1-interface
    edit <value>
        set dpd [disable | on-idle | on-demand]
        set dpd-retryinveral 15
        set dpd-retrycount 3
    next
end
```

DPD scalability

On a dialup server, if many VPN connections are idle, the increased DPD exchange could negatively impact the performance/load of the daemon. The `on-demand` option in the CLI triggers DPD when IPsec traffic is sent, but no reply is received from the peer.

When there is no traffic and the last DPD-ACK had been received, IKE will not send DPDs periodically. IKE will only send out DPDs if there are outgoing packets to send, but no inbound packets have since been received.

HMAC settings

The FortiGate uses the HMAC based on the authentication proposal that is chosen in phase 1 or phase 2 of the IPsec configuration. Each proposal consists of the encryption-hash pair (such as `3des-sha256`). The FortiGate matches the most secure proposal to negotiate with the peer.

To view the chosen proposal and the HMAC hash used:

```
# diagnose vpn ike gateway list

vd: root/0
name: MPLS
version: 1
interface: port1 3
addr: 192.168.2.5:500 -> 10.10.10.1:500
tun_id: 10.10.10.1
virtual-interface-addr: 172.31.0.2 -> 172.31.0.1
created: 1015820s ago
IKE SA: created 1/13 established 1/13 time 10/1626/21010 ms
IPsec SA: created 1/24 established 1/24 time 0/11/30 ms

  id/spi: 124 43b087dae99f7733/6a8473e58cd8990a
  direction: responder
  status: established 68693-68693s ago = 10ms
  proposal: 3des-sha256
  key: e0fa6ab8dc509b33-aa2cc549999b1823-c3cb9c337432646e
  lifetime/rekey: 86400/17436
  DPD sent/recv: 000001e1/00000000
```

Dynamic IPsec route control

The `add-route` option adds a route to peer destination selector into the FortiGate routing information base when the dynamic tunnel is negotiated. You can use the distance and priority options to set the distance and priority of this route. If this results in a route with the lowest distance, it is added to the FortiGate forwarding information base.

You can also enable `add-route` in phase 2 configuration that is associated with a dynamic (dialup) phase 1. In phase 2, `add-route` can be enabled, disabled, or set to use the same route as phase 1.

The `add-route` option is enabled by default.

To configure add-route in phase 1:

```
config vpn ipsec phase1
    edit <name>
        set type dynamic
        set add-route {enable | disable}
    next
end
```

To configure add-route in phase 2:

```
config vpn ipsec {phase2 | phase2-interface}
    edit <name>
        set add-route {phase1 | enable | disable}
    next
end
```

Blocking IPsec SA negotiation

IPsec SA negotiation blocking can only be removed if the peer offers a wildcard selector. If a wildcard selector is offered, then the wildcard route will be added to the routing table with the distance/priority value configured in phase 1. If that is the route with the lowest distance, it will be installed into the forwarding information base.

In this scenario, it is important to ensure that the distance value configured for phase 1 is set appropriately.

Choosing IKE version 1 and 2

FortiGate supports IKEv1 and IKEv2, and both are configured similarly. The underlying protocol for IKEv2 defined in RFC 7296 is more streamlined, requiring fewer message exchanges to negotiate the SAs compared to IKEv1. The major difference is IKEv1 uses XAuth (Extended Authentication) for user authentication, and IKEv2 uses EAP (Extensible Authentication Protocol). IKEv2 also supports using TCP as transport enabling IPsec to negotiate over TCP, encapsulate ESP packets within TCP and operate on custom TCP port such as port 443.

IKEv1 is generally well used and well understood, with a more rigid protocol that is simpler to troubleshoot. Whereas IKEv2 offers more flexibility, resulting in more variations when troubleshooting.

However, starting with FortiClient 7.4.4, IKEv1 is no longer supported on the client. Therefore, plan accordingly when choosing your IKE version. Use IKEv2 if you plan on deploying FortiClient 7.4.4 and later for remote access. New tunnel configurations on the FortiGate also default to using IKEv2.

Repeated authentication in IKEv2

This feature provides the option to control whether a device requires its peer to re-authenticate or whether re-key is sufficient. It does not influence the re-authentication or re-key behavior of the device itself, which is controlled by the peer (the default being to re-key). This solution is in response to RFC 4478. As described by the IETF, "the purpose of this is to limit the time that security associations (SAs) can be used by a third party who has gained control of the IPsec peer".

To configure IKE SA re-authentication:

```
config vpn ipsec phase1-interface
    edit p1
        set reauth [enable | disable]
    next
end
```

IKEv2 quick crash detection

There is support for IKEv2 quick crash detection (QCD) as described in RFC 6290.

RFC 6290 describes a method in which an IKE peer can quickly detect that the gateway peer it has and established an IKE session with has rebooted, crashed, or otherwise lost IKE state. When the gateway receives IKE messages or ESP packets with unknown IKE or IPsec SPIs, the IKEv2 protocol allows the gateway to send the peer an unprotected IKE message containing INVALID_IKE_SPI or INVALID_SPI notification payloads.

RFC 6290 introduces the concept of a QCD token, which is generated from the IKE SPIs and a private QCD secret, and exchanged between peers during the protected IKE AUTH exchange.

To configure QCD:

```
config system settings
    set ike-quick-crash-detect [enable | disable]
end
```

IKEv1 quick crash detection

Based on the IKEv2 QCD feature previously described, IKEv1 QCD is implemented using a new IKE vendor ID (Fortinet Quick Crash Detection) so both endpoints must be FortiGates. The QCD token is sent in the phase 1 exchange and must be encrypted, so this is only implemented for IKEv1 in main mode (aggressive mode is not supported as there is no available AUTH message to include the token). Otherwise, the feature works the same as in IKEv2 (RFC 6290).

IKEv1 fragmentation

UDP fragmentation can cause issues in IPsec when either the ISP or perimeter firewall(s) cannot pass or fragment the oversized UDP packets that occur when using a very large public security key (PSK). The result is that IPsec tunnels do not come up. The solution is IKE fragmentation.

For most configurations, enabling IKE fragmentation allows connections to automatically establish when they otherwise might have failed due to intermediate nodes dropping IKE messages containing large certificates, which typically push the packet size over 1500 bytes.

FortiOS will fragment a packet on sending if only all the following are true:

- Phase 1 contains `set fragmentation enable`.
- The packet is larger than the minimum MTU (576 for IPv4, 1280 for IPv6).
- The packet is being re-transmitted.

By default, IKE fragmentation is enabled.

To configure IKEv1 fragmentation:

```
config vpn ipsec phase1-interface
    edit 1
        set fragmentation [enable | disable]
    next
end
```

IKEv2 fragmentation

RFC 7383 requires each fragment to be individually encrypted and authenticated. With IKEv2, a copy of the unencrypted payloads around for each outgoing packet would need to be kept in case the original single packet was never answered and would retry with fragments. With the following implementation, if the IKE payloads are greater than a configured threshold, the IKE packets are preemptively fragmented and encrypted.

To configure IKEv2 fragmentation:

```
config vpn ipsec phase1-interface
    edit ike
        set ike-version 2
        set fragmentation [enable|disable]
        set fragmentation-mtu <500-16000>
    next
end
```

IPsec global IKE embryonic limit

When trying to establish thousands of tunnels simultaneously, a situation can arise where new negotiations starve other SAs from progressing to an established state in IKEv2. The IKE daemon can prioritize established SAs, offload groups 20 and 21 to CP9, and optimize the default embryonic limits for mid- and high-end platforms. The IKE embryonic limit can be configured in the CLI.

```
config system ike
    set embryonic-limit <integer>
end
```

`embryonic-limit <integer>`	Set the maximum number of IPsec tunnels to negotiate simultaneously (50 - 20000, default = 1000).

To configure an IKE embryonic limit of 50:

```
config system ike
    set embryonic-limit 50
end
```

Authentication method

For site-to-site IPsec tunnels between two FortiGates or a FortiGate and another VPN gateway, authentication is typically done using pre-shared keys or certificates. For remote access dialup VPN, users can authenticate using pre-shared keys or certificates and through Extended Authentication (XAuth) in IKEv1 tunnels or EAP in IKEv2 tunnels. Authentication can be any of the following methods supported by the FortiGate:

Authentication Methods	IKE Version	Requirement
• Pre-shared key • PKI (Signature)	IKEv1 and IKEv2	Required to configure one of these authentication methods

Authentication Methods	IKE Version	Requirement
• LDAP	IKEv1 and IKEv2 (requires EAP-TTLS)	Optional user authentication methods. (IPsec IKEv1 uses XAUTH, and IPsec IKEv2 uses EAP for user authentication.)
• Local • RADIUS	IKEv1 and IKEv2	
• SAML	IKEv2	
• Two-factor authentication	IKEv1 and IKEv2	Optional

Pre-shared key and PKI authentication can be paired with any of the other user authentication methods. Two-factor authentication using FortiToken is also supported and can work in combination with Local, LDAP, RADIUS, or SAML authentication. Two-factor authentication with client certificate is also supported.

Pre-shared key

Using a pre-shared key is less secure than using certificates, especially if it is used alone, without requiring peer IDs or extended user authentication like XAuth or EAP. There also needs to be a secure way to distribute the pre-shared key to the peers.

If you use pre-shared key authentication alone, all remote peers and dialup clients must be configured with the same pre-shared key. Optionally, you can configure remote peers and dialup clients with unique pre-shared keys. On the FortiGate, these are configured in user accounts, not in the phase 1 settings.

The pre-shared key must contain at least six printable characters and should be known by network administrators. For optimum protection against currently known attacks, the key must consist of a minimum of 16 randomly chosen alphanumeric characters. The limit is 128 characters.

If you use a pre-shared key as the authentication method, you cannot also use client certificate as another authentication method.

To authenticate the FortiGate using a pre-shared key:

1. Go to *VPN > VPN Tunnels* and create a new tunnel, or edit an existing one.
2. Configure or edit the *Network* section as needed.
3. Configure or edit the *Authentication* settings as follows:

Method	*Pre-shared Key*
Pre-shared Key	<string>
IKE	*Version 1* or *Version 2*
Mode	In IKEv1, select *Aggressive* or *Main* mode.
Accepted Peer ID	Select an *Accept Type* and the corresponding peer. Options vary based on the *Remote Gateway* and *Authentication Method* settings in the *Network* section. *Peer Options* are only available in IKEv1 *Aggressive* mode. For IKEv2, use Network overlay.

Network Overlay	In IKEv2, enable to require peers to configure a *Network ID*. When multiple tunnels are configured on the same interface, *Network ID* is used to match the tunnel.

4. For the *Phase 1 Proposal* section, keep the default settings unless changes are needed to meet your requirements.
5. Optionally, for authentication parameters for a dialup user group, define *XAUTH* parameters.
6. Click *OK*.

Digital certificates

To authenticate the FortiGate using digital certificates, you must have the required certificates installed on the remote peer and on the FortiGate. The signed server certificate on one peer is validated by the presence of the root certificate installed on the other peer. If you use certificates to authenticate the FortiGate, you can also require the remote peers or dialup clients to authenticate using certificates. See Basic site-to-site VPN with digital certificate on page 2867 for a detailed example.

To authenticate the FortiGate using a digital certificate:

1. Go to *VPN > VPN Tunnels* and create a new tunnel, or edit an existing one.
2. Configure or edit the *Network* section as needed.
3. Configure or edit the *Authentication* settings as follows:

Method	*Signature*
Certificate Name	Select the certificate used to identify this FortiGate. If there are no imported certificates, consider first to generate a certificate from a Certificate Authority and import it into the FortiGate. See Import a certificate on page 3962. If that is not possible, consider using the *Fortinet_Factory* certificate.
IKE	*Version 1* or *Version 2*
Mode	In IKEv1, *Aggressive* is recommended.
Accepted Peer ID	For *Accept Type*, select *Peer certificate* and select the peer and the CA certificate used to authenticate the peer. You can define more granular parameters for the peer when you create or edit a PKI user. See Configuring a PKI user on page 3662. If the other end is using the *Fortinet_Factory* certificate, then use the *Fortinet_CA* certificate here.

4. For the *Phase 1 Proposal* section, keep the default settings unless changes are needed to meet your requirements.
5. Optionally, for authentication parameters for a dialup user group, define XAUTH or EAP parameters.
6. Click *OK*.

When using signature as an authentication method in IKEv2, FortiClient supports submitting a certificate to FortiOS IKEv2 for validation, then continuing to the configured EAP method (EAP-MSCHAPv2 or EAP-TTLS) for authentication.

Certificate authentication as an additional step is enabled using the following configuration in FortiOS.

```
config vpn ipsec phase1-interface
    edit <tunnel>
        set authmethod signature
        set peertype peer
        set eap enable
        set eap-cert-auth {*disable | enable}
        set peer <user peer>
    next
end
```

For an example, see VPN 2FA with EAP and certificate authentication on page 3019.

XAuth user authentication

Extended authentication (XAuth) increases security for IKEv1 by requiring remote dialup client users to authenticate in a separate exchange at the end of phase 1. XAuth draws on existing FortiGate user group definitions and uses established authentication mechanisms such as PAP, CHAP, RADIUS, and LDAP to authenticate dialup clients. You can configure a FortiGate to function either as an XAuth server or client. If the server or client is attempting a connection using XAuth and the other end is not using XAuth, the failed connection attempts that are logged will not specify XAuth as the reason.

XAuth server

A FortiGate can act as an XAuth server for dialup clients. When the phase 1 negotiation completes, the FortiGate challenges the user for a user name and password. It then forwards the user's credentials to an external RADIUS or LDAP server for verification.

> If the user records on the RADIUS server have suitably configured Framed-IP-Address fields, you can assign client virtual IP addresses by XAuth instead of from a DHCP address range.

The authentication protocol you use for XAuth depends on the capabilities of the authentication server and the XAuth client:

- Select *CHAP Server* whenever possible for higher challenge-response based security.
- Select *PAP Server* for all implementations of LDAP and some implementations of Microsoft RADIUS that do not support CHAP.
- Select *Auto Server* when the authentication server supports *CHAP Server* but the XAuth client does not. The FortiGate will use PAP to communicate with the XAuth client and CHAP to communicate with the authentication server. You can also use *Auto Server* to allow multiple source interfaces to be defined in an IPsec/IKE policy.

Before you begin, create user accounts and user groups to identify the dialup clients that need to access the network behind the FortiGate dialup server. If password protection will be provided through an external RADIUS or LDAP server, you must configure the FortiGate dialup server to forward authentication requests to the authentication server.

To configure XAuth to authenticate a dialup user group:

1. On the FortiGate dialup server, go to *VPN > IPsec Tunnels* and create a new tunnel, or edit an existing one.
2. Set *Remote gateway* to *Dialup user*.
3. Set *IKE* to *Version 1*.
4. Enable *XAUTH* and select the method to use between the XAuth client, the FortiGate, and the authentication server.
5. For *User Group*:
   a. Select *Inherit from policy* for multiple user groups defined in the IPsec/IKE policy, or
   b. Select *Specify* and, in the dropdown, select the user group that needs to access the private network behind the FortiGate.

   Only one user group may be defined for *Auto Server*.
6. Configure the remaining *Network*, *Authentication*, and *Phase 1 Proposal* options as needed.
7. Click *OK*.
8. Create as many policies as needed, specifying the source user(s) and destination address.

XAuth client

If the FortiGate acts as a dialup client, the remote peer, acting as an XAuth server, might require a username and password. You can configure the FortiGate as an XAuth client with its own username and password, which it provides when challenged.

To configure the FortiGate dialup client as an XAuth client:

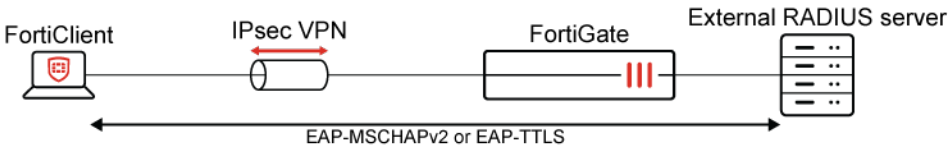
1. On the FortiGate dialup client, go to *VPN > IPsec Tunnels* and create a new tunnel, or edit an existing one.
2. Set *Remote gateway* to either *Static IP address* or *Dynamic DNS*.
3. Set *IKE* to *Version 1*.
4. Enable *XAUTH*, *XAuth client* is selected as the only option.
5. Enter the *Username* and *Password* that will be used to authenticate this FortiGate.
6. Configure the remaining *Network*, *Authentication*, and *Phase 1 Proposal* options as needed.
7. Click *OK*.

EAP user authentication

EAP is a native authentication mechanism of IKEv2 that occurs during the IKE_AUTH exchange. The IKE SA is only completed when EAP authentication completes successfully. EAP allows the VPN gateway to authenticate the initiator using a variety of methods, including EAP-MSCHAPv2, EAP-TLS, and EAP-TTLS.

However, the EAP method must be supported by the client endpoint as well as the external authentication servers. FortiClient IKEv2 only supports initiating EAP-MSCHAPv2 and EAP-TTLS authentication.

EAP-MSCHAPv2 or EAP-TTLS with External RADIUS server



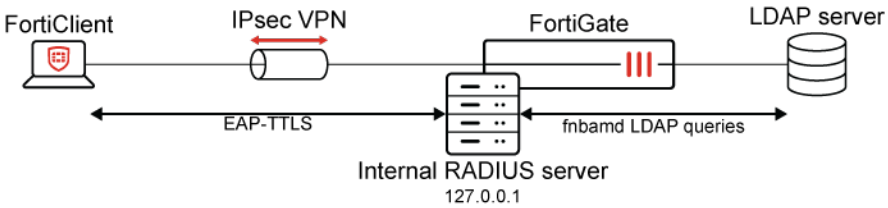
If an external RADIUS server is configured for the VPN, EAP-MSCHAPv2 or EAP-TTLS can be used for end to end communication.

FortiOS does not terminate the EAP-TTLS tunnel on its internal eap_proxy RADIUS server. Instead, it passes the EAP-TTLS connection attempt through to the external RADIUS server. EAP-TTLS with PAP as an inner method must be supported and properly configured on the external RADIUS server, or the authentication attempt will fail.

The EAP method attempted is configured on the client, and the default is EAP-MSCHAPv2. If EAP-TTLS/PAP is in use, the EAP server certificate is configured on the remote RADIUS server. Unlike EAP-TLS, EAP-TTLS does not require the use of client certificates.

Multiple RADIUS servers associated with the same dial-up gateway are not supported.

EAP-TTLS with LDAP server



If EAP-TTLS is used and no external RADIUS server is configured for the VPN (like when LDAP is used), the EAP-TTLS tunnel terminates on the FortiGate's own internal RADIUS server.

Because the firewall can view the user's original plaintext credentials in this case, FortiOS attempts to authenticate them against the set of local users and remote LDAP servers configured for the tunnel.

To support EAP-TTLS on the client side, FortiClient 7.4.3 and later should be used.

FortiGate cannot authenticate Active Directory users against an LDAP server directly using EAP-MSCHAPv2. See IKEv2 dial up VPN with LDAP authentication.

The TLS certificate that FortiOS uses for EAP-TTLS is *Fortinet_Wifi* by default and is signed by a public CA. The certificate is configured in global settings with the `wifi-certificate` command. If the certificate is not trusted by the client, VPN connections will fail with certificate errors showing in debug messages. See EAP-TTLS connection failing with FortiClient and IPsec IKEv2 connection due to certificate error.

Two-factor authentication with FortiToken and FortiAuthenticator

When the user's FortiToken is hosted on a FortiAuthenticator, EAP-MSCHAPv2 must be used for the user to see a token prompt or get a push notification. This can be used to trigger token prompts for on-premises Active

Directory users when the FortiAuthenticator is joined to the Active Directory domain. See Authenticating users using MSCHAP2 PEAP.

When authenticating against a remote RADIUS server that performs 2FA, extending the RADIUS timeout is usually required.

```
config user radius
    edit <server name>
        set timeout 60
    next
end
```

When EAP-TTLS must be used, FortiAuthenticator does not support sending the prompt, but the user may append the token code to their password as a workaround. See Multi-Factor Authentication support for Windows FortiClient with LDAP (EAP-TTLS).

The default IPsec negotiation timeout is 30 seconds. If OTP entry takes longer, the IPsec connection may expire, preventing access even after successful authentication. Consider increasing the IPsec negotiation timeout to allow enough time for users to enter their one-time password (OTP) or accept the token prompt.

```
config vpn ipsec phase1-interface
    edit <tunnel name>
        set negotiate-timeout 60
    next
end
```

If using FortiTokens provisioned on FortiGate, EAP-TTLS authentication requires FortiOS 7.4.9 or later and FortiClient Windows 7.4.4. See Multi-Factor Authentication support for Windows FortiClient with LDAP (EAP-TTLS) and FortiOS 7.4.9 Resolved issue 1031789.

SAML authentication

SAML authentication is not an EAP method. When SAML authentication is in use, FortiClient initiates an HTTPS connection to the FortiGate as the SAML Service Provider (SP) first. After SAML redirect and authentication with the SAML IDP, FortiClient presents the SAML cookie to the FortiGate, which caches a user authentication entry internally.

FortiClient attempts to connect to the IKEv2 dial-up gateway. FortiClient authenticates FortiGate using pre-shared key or signature authentication, then proves ownership of the cached user authentication by authenticating itself against the FortiGate eap_proxy using EAP-MSCHAPv2 or EAP-TTLS.

For this reason, it is still required to enable EAP in the FortiGate phase1-interface configuration, even though SAML is not an EAP method.

When configuring a single-sign-on tunnel on FortiClient, it is recommended leave eap_method in its default value. FortiClient 7.2.4 and later support using SAML authentication with IKEv2. See IPsec VPN SAML-based authentication.

FQDN support for remote gateways

FortiGate supports FQDN when defining an IPsec remote gateway with a dynamically assigned IPv6 address. When FortiGate attempts to connect to the IPv6 device, FQDN will resolve the IPv6 address even when the address changes.

Using FQDN to configure the remote gateway is useful when the remote end has a dynamic IPv6 address assigned by their ISP or DHCPv6 server.

To set the VPN to DDNS and configure FQDN:

```
config vpn ipsec phase1-interface
    edit "ddns6"
        set type ddns
        set interface "agg1"
        set ip-version 6
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set dpd on-idle
        set remotegw-ddns "rgwa61.vpnlab.org"
        set psksecret **********
    next
end
```

```
config vpn ipsec phase2-interface
    edit "ddns6"
        set phase1name "ddns6"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
        set src-addr-type subnet6
        set dst-addr-type subnet6
        set src-subnet6 2003:1:1:1::/64
    next
end
```

FQDN resolves the IPv6 address

```
# diagnose test application dnsproxy 7

vfid=0, name=rgwa61.vpnlab.org, ttl=3600:3547:1747
         2003:33:1:1::22 (ttl=3600)
```

FortiGate uses FQDN to connect to the IPv6 device

```
# diagnose vpn tunnel list name ddns6
list ipsec tunnel by names in vd 0
------------------------------------------------------
```

```
name=ddns6 ver=2 serial=2 2003:33:1:1::1:0->2003:33:1:1::22:0 dst_mtu=1500
bound_if=32 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/520 options[0208]=npu frag-rfc  run_
state=0 accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=10 ilast=9 olast=9 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=1 idle=20000ms retry=3 count=0 seqno=72340
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=ddns6 proto=0 sa=1 ref=2 serial=1
  src: 0:2003:1:1:1::/64:0
  dst: 0:::/0:0
  SA:  ref=3 options=10226 type=00 soft=0 mtu=1422 expire=42680/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 itn=0 qat=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42901/43200
  dec: spi=ac7a5718 esp=aes key=16 9976b66280cc49f500d8edca093e03fb
       ah=sha1 key=20 4d94d76fc18df5a180c52e0a6cd5f430fde48fe8
  enc: spi=7ab888ec esp=aes key=16 841a95d3ee5ea5108a2ba269b74998d1
       ah=sha1 key=20 ed0b52d27776e30149ee36af4fd4626681c2a3a1
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=2003:33:1:1::22 npu_lgwy=2003:33:1:1::1 npu_selid=0 dec_npuid=0 enc_npuid=0
run_tally=1
```

The tunnel can still connect to the FQDN address when the IPv6 address changes

```
# diagnose debug application ike -1
# diagnose debug enable
ike 0:ddns6: set oper down
ike 0:ddns6: carrier down
ike shrank heap by 159744 bytes
ike 0: cache rebuild start
ike 0:ddns6: sending DNS request for remote peer rgwa61.vpnlab.org
ike 0: send IPv6 DNS query : rgwa61.vpnlab.org
ike 0: cache rebuild done
ike 0:ddns6: remote IPv6 DDNS gateway is empty, retry to resolve it
ike 0: DNS response received for remote gateway rgwa61.vpnlab.org
ike 0: DNS rgwa61.vpnlab.org -> 2003:33:1:1::33
ike 2:test:46932: could not send IKE Packet(P1_RETRANSMIT):50.1.1.1:500->50.1.1.2:500, len=716:
error 101:Network is unreachable
ike 0:ddns6: remote IPv6 DDNS gateway is empty, retry to resolve it
ike 0:ddns6: 'rgwa61.vpnlab.org' resolved to 2003:33:1:1::33
ike 0: cache rebuild start
ike 0:ddns6: local:2003:33:1:1::1, remote:2003:33:1:1::33
ike 0:ddns6: cached as static-ddns.
ike 0: cache rebuild done
ike 0:ddns6: auto-negotiate connection
ike 0:ddns6: created connection: 0x155aa510 32 2003:33:1:1::1->2003:33:1:1::33:500.

.....................................................................................
..................
ike 0:ddns6:46933:ddn6:47779: add IPsec SA: SPIs=ac7a5719/7ab888ed
ike 0:ddns6:46933:ddn6:47779: IPsec SA dec spi ac7a5719 key 16:0F27F1D1D02496F90D15A30E2C032678
auth 20:46564E0E86A054374B31E58F95E4458340121BCE
```

```
ike 0:ddns6:46933:ddn6:47779: IPsec SA enc spi 7ab888ed key 16:926B12908EE670E1A5DDA6AD8E96607B
auth 20:42BF438DC90867B837B0490EAB08E329AB62CBE3
ike 0:ddns6:46933:ddn6:47779: added IPsec SA: SPIs=ac7a5719/7ab888ed
ike 0:ddns6:46933:ddn6:47779: sending SNMP tunnel UP trap
ike 0:ddns6: carrier up
```

Resuming sessions for IPsec tunnel IKE version 2

FortiOS supports session resumptions for IPsec tunnel IKE version 2. This feature enhances the user experience by maintaining the tunnel in an idle state, which allows for uninterrupted usage even after a client resumes from sleep or when connectivity is restored after a disruption. Furthermore, it removes the necessity for re-authentication when reconnecting, making the process more efficient.

```
config vpn ipsec phase1-interface
    edit <phase 1 name>
        set client-resume enable
        set client-resume-interval {integer length of idle time}
    next
end
```

Example

In the following example, the client FortiGate will be configured to enable session resumption after returning from an idle state. The resume interval will be set as 120 seconds and the interface status will be tested when the client resumes within and past this interval.

This example uses a pre-shared key for authentication, although signature authentication can also be used.

To enable session resumption for IPsec tunnel IKE version 2:

1. Configure IPsec VPN for the dialup client FortiGate:
   a. Configure the IPsec phase 1 interface:

```
config vpn ipsec phase1-interface
    edit "toServer"
        set interface "port9"
        set ike-version 2
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set dpd on-idle
        set remote-gw 173.1.1.1
        set psksecret ****************
        set dpd-retryinterval 60
    next
end
```

b. Configure the IPsec phase 2 interface:

```
config vpn ipsec phase2-interface
    edit "toServer"
        set phase1name "toServer"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
    next
end
```

c. Configure the firewall policy:

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "toServer"
        set action accept
        set srcaddr "10.1.100.0"
        set dstaddr "192.168.5.0"
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set srcintf "toServer"
        set dstintf "port2"
        set action accept
        set srcaddr "192.168.5.0"
        set dstaddr "10.1.100.0"
        set schedule "always"
        set service "ALL"
   next
end
```

2. Configure IPsec VPN for the dialup server FortiGate:

a. Configure the IPsec phase 1 interface:

```
config vpn ipsec phase1-interface
    edit "toClient"
        set type dynamic
        set interface "port8"
        set ike-version 2
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set dpd on-idle
        set client-resume enable
        set client-resume-interval 120
        set psksecret ****************
        set dpd-retryinterval 60
    next
end
```

b. Configure the IPsec phase 2 interface:

```
config vpn ipsec phase2-interface
    edit "toClient"
        set phase1name "toClient"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
    next
end
```

c. Configure the firewall policy:

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "toClient"
        set action accept
        set srcaddr "192.168.5.0"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set srcintf "toClient"
        set dstintf "port2"
        set action accept
        set srcaddr "all"
        set dstaddr "192.168.5.0"
        set schedule "always"
        set service "ALL"
    next
end
```

3. Check the IPsec phase 1 and phase 2 interface status and the client resume messages:

a. In the following scenario, the client becomes idle or has connectivity issues but resumes within the set 120 second interval:

```
# diagnose debug application ike -1
ike V=root:0:toClient_0: starting client-resume sleep period 120 sec (1)
ike V=root:0: comes 11.101.1.1:500->173.1.1.1:500,ifindex=16,vrf=0,len=632....
ike V=root:0: IKEv2 exchange=SA_INIT id=dfa40e0473d89550/0000000000000000 len=632
ike V=root:0:toClient_0: client has resumed (1)
ike 0:toClient_0:10: out
6749C602DDF25B141E24AC649641D2242E2025000000000200000050000000343F9A56C6D16E93E2F5D7BC7F66
DC7CBDB3E44EA75F0A87A3238DCC08EE0BF478817EC93DF72EB2B3E027D695FACECF4E
ike V=root:0:toClient_0:11:toClient:30: sending SNMP tunnel UP trap
ike V=root:0:toClient_0: tunnel up event
```

b. In the following scenario, the client becomes idle or has connectivity issues but the set 120 second interval expires before it resumes. Therefore, the tunnel is not maintained:

```
# diagnose debug application ike -1
ike V=root:0:toClient_0:9: sent IKE msg (RETRANSMIT_INFORMATIONAL): 173.1.1.1:500-
```

```
>11.101.1.1:500, len=80, vrf=0, id=e50861aebc1e5b1a/6457ea1e8512148c, oif=16
ike V=root:0:toClient_0: link is idle 16 173.1.1.1->11.101.1.1:0 dpd=1 seqno=2 rr=0
ike V=root:0:toClient_0:9: send IKEv2 DPD probe, seqno 2
ike V=root:0:toClient_0:9: e50861aebc1e5b1a/6457ea1e8512148c retransmission timeout
ike V=root:0:toClient_0: starting client-resume sleep period 120 sec (1)
ike V=root:0:toClient_0: client-resume sleep period has expired (1)
ike V=root:0:toClient_0: going to be deleted
ike V=root:0:toClient_0: flushing
ike V=root:0:toClient_0: deleting IPsec SA with SPI 30c477cd
ike V=root:0:toClient_0:toClient: deleted IPsec SA with SPI 30c477cd, SA count: 0
ike V=toClient_0:0:toClient_0:23: del route 0.0.0.0/0.0.0.0 tunnel 11.101.1.1 oif
toClient_0(45) metric 15 priority 1
ike V=root:0:toClient_0: sending SNMP tunnel DOWN trap for toClient
ike V=root:0:toClient_0: last dialup SA expired while client sleeping
ike V=root:0:toClient_0: flushed
```

Fragmenting IP packets before IPsec encapsulation

The `ip-fragmentation` command controls packet fragmentation before IPsec encapsulation, which can benefit packet loss in some environments.

For example, when a packet is encapsulated and exceeds the MTU size but it is NP-offloaded, the NPU cannot generate an ICMP unreachable, need to frag message. This message can only be generated when NPU offloading is disabled. This can lead to packet loss because the sender cannot be notified and the IP fragments cannot be transmitted.

By enabling pre-encapsulation, the TCP packet is fragmented before it is encapsulated, so all fragments are encapsulated and transmitted as ESP packets.

The following options are available for the `ip-fragmentation` variable.

Option	Description
`pre-encapsulation`	Fragment before IPsec encapsulation.
`post-encapsulation`	Fragment after IPsec encapsulation (RFC compliant) (default value).

To configure packet fragmentation using the CLI:

```
config vpn ipsec phase1-interface
    edit "demo"
        set interface "port1"
        set authmethod signature
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set ip-fragmentation pre-encapsulation
        set remote-gw 172.16.200.4
        set certificate "Fortinet_Factory"
```

```
    next
end
```

In 7.6.1 and later, FortiGate does not reduce the MSS size. It only uses the ICMP_FRAG_NEEDED message to notify the sender of the MTU size so that the sender can adjust the TCP MSS and packet size. In earlier versions, when TCP traffic goes through an IPsec tunnel, FortiGate would reduce the TCP MSS size if the packet was larger than the tunnel's MTU. FortiGate would then use the ICMP_FRAG_NEEDED message to notify the sender of the MTU size so that the sender can adjust the TCP MSS and packet size.

Cross-validation for IPsec VPN

FortiOS supports a cross-validation mechanism for IPsec VPN, aimed at bolstering security and user authentication by mitigating the risk of unauthorized access and identity spoofing. This mechanism functions by cross-checking whether the username provided by the client matches the identity field specified in the peer certificate. The identity field, which could be an Othername, RFC 822 Name, or CN, serves as a unique identifier for the client.

```
config vpn ipsec phase1-interface
    edit <name>
        set cert-peer-username-validation {none | othername | rfc822name | cn}
        set cert-peer-username-strip {enable | disable}
    next
end
```

Validation can be enabled for IPsec VPN when the peer sends certificates for authentication and XAUTH, EAP, SAML, or `azure-ad-autoconnect` is used. Name matching is not case sensitive.

To configure cross-validation for IPsec VPN:

1. Configure IPsec VPN phase1:

```
config vpn ipsec phase1-interface
    edit "toclient"
        set type dynamic
        set interface "port8"
        set authmethod signature
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256
        set dpd on-idle
        set dhgrp 5
        set xauthtype auto
        set authusrgrp "vpnad01- ldap-group"
        set cert-peer-username-validation othername
        set cert-peer-username-strip enable
        set certificate "FTG-D"
```

```
    next
end
```

2. Configure IPsec VPN phase2:

```
config vpn ipsec phase2-interface
    edit "toclient"
        set phase1name "toclient"
        set proposal aes128-sha256 aes256-sha256
        set dhcp-ipsec enable
    next
end
```

3. In FortiClient, input a username that corresponds with the identity in the certificate, such as *tester2*.
4. Verify that the cross-validation succeeded and the tunnel is established:

```
# diagnose vpn ike gateway list

vd: root/0
name: toclient_0
version: 1
interface: port8 16
addr: 173.1.1.1:4500 -> 173.1.1.2:64917
tun_id: 173.1.1.2/::10.0.0.12
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 17s ago
xauth-user: tester2
2FA: no
peer-id: DC=com, DC=vpnfosqa, CN=Users, CN=tester2
peer-id-auth: yes
FortiClient UID: 43B7079E9D244365A91CB0F139EA470F
nat: peer
pending-queue: 0
IKE SA: created 1/1  established 1/1  time 20/20/20 ms
IPsec SA: created 1/2  established 1/2  time 0/95/190 ms

  id/spi: 26 df4fd267a4ba2093/a9161305effacd5a
  direction: responder
  status: established 17-17s ago = 20ms
  proposal: aes128-sha256
  key: 3cbdc74916c1b2aa-e00b39fda1b2e324
  QKD: no
  lifetime/rekey: 86400/86112
  DPD sent/recv: 00000000/00000002
  peer-id: DC=com, DC=vpnfosqa, CN=Users, CN=tester2
```

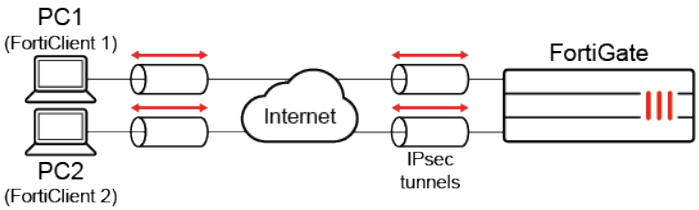
5. In FortiClient, input a username that does not correspond with the identity in the certificate, such as *tester1*.
6. Verify that the cross-validation failed:

```
# diagnose debug application fnbamd -1
# diagnose debug enable

ike V=root:0:toclient:23: received p1 notify type INITIAL-CONTACT
ike V=root:0:toclient:23: received peer identifier DER_ASN1_DN 'DC=com, DC=vpnfosqa, CN=Users,
CN=tester2'
ike V=root:0:toclient:23: re-validate gw ID
ike V=root:0:toclient:23: gw validation OK
ike V=root:0:toclient:23: Validating X.509 certificate

[984] __ldap_next_state-State: Primary Group Query -> Done
[1982] ldap_copy_grp_list-copied CN=Domain Admins,CN=Users,DC=vpnfosqa,DC=com
[1982] ldap_copy_grp_list-copied CN=Administrators,CN=Builtin,DC=vpnfosqa,DC=com
[1982] ldap_copy_grp_list-copied CN=Domain Users,CN=Users,DC=vpnfosqa,DC=com
[626] fnbam_user_auth_group_match-req id: 2207758716938, server: vpnad01, local auth: 0, dn
match: 1
[580] __group_match-Check if vpnad01 is a group member
[586] __group_match-Group 'vpnad01- ldap-group' passed group matching
[589] __group_match-Add matched group 'vpnad01- ldap-group'(17)
[202] find_matched_usr_grps-Passed group matching
[2423] fnbamd_ldap_result-Result for ldap svr vpnad01 is SUCCESS
[626] fnbam_user_auth_group_match-req id: 2207758716938, server: vpnad01, local auth: 0, dn
match: 1
[580] __group_match-Check if vpnad01 is a group member
[586] __group_match-Group 'vpnad01- ldap-group' passed group matching
[589] __group_match-Add matched group 'vpnad01- ldap-group'(17)
[2431] fnbamd_ldap_result-Passed group matching
[239] fnbamd_comm_send_result-Sending result 0 (nid 0) for req 0x20208AC900A, len=2750
ike V=root:0:toclient_0:23: XAUTH 2207758716938 result FNBAM_SUCCESS
ike V=root:0:toclient_0: XAUTH succeeded for user "tester1" group "vpnad01- ldap-group" 2FA=no
[1254] fnbamd_rads_destroy-
ike V=root:0:toclient_0: client cert username validation failed. Input username [tester1] does
not match name [tester2@vpnfosqa.com] in client cert.
[1830] fnbamd_ldaps_destroy-
ike V=root:0:toclient_0: connection expiring due to client cert username validation failure
[442] fnbamd_ldap_auth_ctx_free-Freeing 'vpnad01' ctx
ike V=root:0:toclient_0: going to be deleted
```

IPsec VPN IP address assignments

When a user disconnects from a VPN tunnel, it is not always desirable for the released IP address to be used immediately. In IPsec VPN, IP addresses can held for the specified delay interval before being released back into the pool for assignment. The first-available address assignment method is still used.

Example



In this example, two PCs connect to the VPN. The IP address reuse delay interval is used to prevent a released address from being reused for at least four minutes. After the interval elapses, the IP address becomes available to clients again. Dual stack address assignment (both IPv4 and IPv6) is used.

To configure IPsec VPN with an IP address reuse delay interval:

1. Configure the IPsec phase1 interface, setting the IP address reuse delay interval to 240 seconds:

```
config vpn ipsec phase1-interface
    edit "FCT"
        set type dynamic
        set interface "port27"
        set mode aggressive
        set peertype any
        set net-device disable
        set mode-cfg enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set wizard-type dialup-forticlient
        set xauthtype auto
        set authusrgrp "local-group"
        set ipv4-start-ip 10.20.1.1
        set ipv4-end-ip 10.20.1.100
        set dns-mode auto
        set ipv4-split-include "FCT_split"
        set ipv6-start-ip 2001::1
        set ipv6-end-ip 2001::2
        set ip-delay-interval 240
        set save-password enable
        set psksecret **********
    next
end
```

2. Configure the IPsec phase2 interface:

```
config vpn ipsec phase2-interface
    edit "FCT"
        set phase1name "FCT"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
    next
    edit "FCT6"
        set phase1name "FCT"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
```

```
chacha20poly1305
        set src-addr-type subnet6
        set dst-addr-type subnet6
    next
end
```

To test the results:

1. Connect to the VPN with FortiClient 1 on PC1 then check the assigned IP address:

```
# diagnose vpn ike gateway list

vd: root/0
name: FCT_0
version: 1
interface: port27 17
addr: 173.1.1.1:4500 -> 173.1.1.2:60417
tun_id: 173.1.1.2
remote_location: 0.0.0.0
virtual-interface-addr: 169.254.1.1 -> 169.254.1.1
created: 14s ago
xauth-user: userc
2FA: no
FortiClient UID: 7C0897D80C8E4B6DAC775DD6B0F93BAA
assigned IPv4 address: 10.20.1.1/255.255.255.255
assigned IPv6 address: 2001::1/128
nat: peer
IKE SA: created 1/1  established 1/1  time 100/100/100 ms
IPsec SA: created 2/2  established 2/2  time 0/5/10 ms

  id/spi: 2 66140ba3e38b9b07/b64668f110ca4a48
  direction: responder
  status: established 14-14s ago = 100ms
  proposal: aes256-sha256
  key: 356637ee6e9a9cb5-fade432c09efb8aa-54be307fc1eeeab5-6e4b9ef19f98d5fa
  lifetime/rekey: 86400/86115
  DPD sent/recv: 00000000/00000394
```

2. Disconnect FortiClient 1 and connect with FortiClient 2. The IP address assigned to FortiClient 1 is not released to the pool, and a different IP address is assigned to FortiClient 2:

```
# diagnose vpn ike gateway list

vd: root/0
name: FCT_0
version: 1
interface: port27 17
addr: 173.1.1.1:4500 -> 173.1.1.2:64916
tun_id: 173.1.1.2
remote_location: 0.0.0.0
virtual-interface-addr: 169.254.1.1 -> 169.254.1.1
created: 6s ago
```

```
xauth-user: usera
2FA: no
FortiClient UID: EAF90E297393456AB546A041066C0720
assigned IPv4 address: 10.20.1.2/255.255.255.255
assigned IPv6 address: 2001::2/128
nat: peer
IKE SA: created 1/1  established 1/1  time 110/110/110 ms
IPsec SA: created 2/2  established 2/2  time 0/5/10 ms

  id/spi: 3 b25141d5a915e67e/b32decdb8cf98318
  direction: responder
  status: established 6-6s ago = 110ms
  proposal: aes256-sha256
  key: 374ab753f3207ea0-83496b5cb24b5a8d-c51da1fd505cf3a4-727884839897808a
  lifetime/rekey: 86400/86123
  DPD sent/recv: 00000000/00000453
```

3. Wait for 240 seconds, then disconnect and reconnect FortiClient 2. The IP address previously assigned to FortiClient 1 has been released back to the pool, and is assigned to FortiClient 2:

```
# diagnose vpn ike gateway list

vd: root/0
name: FCT_0
version: 1
interface: port27 17
addr: 173.1.1.1:4500 -> 173.1.1.2:64916
tun_id: 173.1.1.2
remote_location: 0.0.0.0
virtual-interface-addr: 169.254.1.1 -> 169.254.1.1
created: 20s ago
xauth-user: usera
2FA: no
FortiClient UID: EAF90E297393456AB546A041066C0720
assigned IPv4 address: 10.20.1.1/255.255.255.255
assigned IPv6 address: 2001::1/128
nat: peer
IKE SA: created 1/1  established 1/1  time 100/100/100 ms
IPsec SA: created 2/2  established 2/2  time 0/0/0 ms

  id/spi: 4 fb1fbad0c12f5476/aa06a2de76964f63
  direction: responder
  status: established 20-20s ago = 100ms
  proposal: aes256-sha256
  key: af43f1bb876dc79c-16448592fe608dc3-f251746d71b2c35d-c848e8c03bf738e9
  lifetime/rekey: 86400/86109
  DPD sent/recv: 00000000/000000a9
```

Instead of waiting for 240 seconds, you can instead use the `diagnose vpn ike gateway flush` command to release the previously used IP addresses back into the pool.

Phase 2 configuration

After phase 1 negotiations end successfully, phase 2 begins. In Phase 2, the VPN peer or client and the FortiGate exchange keys again to establish a secure communication channel. The phase 2 proposal parameters select the encryption and authentication algorithms needed to generate keys for protecting the implementation details of security associations (SAs). The keys are generated automatically using a Diffie-Hellman algorithm. Post-Quantum cryptographic algorithms can also be used in additional Key Exchange just like in phase 1.

The basic phase 2 settings associate IPsec phase 2 parameters with the phase 1 configuration that specifies the remote end point of the VPN tunnel. In most cases, you need to configure only basic Phase 2 settings.

Some settings can be configured in the CLI. The following options can be configured when you create a custom VPN tunnel by clicking *Create New* in the *Phase 2 selectors* table.

Command	Description
Name	Phase 2 definition name.
Encapsulation	Define whether you encapsulate the entire IP packet or leave the original IP headers intact. • *Tunnel Mode*: The original IP header and payload are encrypted and encapsulated in ESP with a new IP header. Used for site-to-site and remote access VPNs where the gateway is the endpoint for the outer IP header. • *Transport Mode*: Only the payload is encrypted and encapsulated in ESP. The original header is used for the outer IP header. Used for VPNs between two hosts. For NAT traversal and IKEv2 over TCP, the ESP packet is further encapsulated in UDP or TCP.
IP version	*Tunnel Mode* only. Choose the IP version for the packet
Named address	*Tunnel Mode* only. Instead of defining the local and remote selector address directly in Phase 2 settings, select address objects instead.
Local Address	*Tunnel Mode* only. A value of `0.0.0.0/0` means all IP addresses behind the local VPN peer. Add a specific address or range to allow traffic from and to only this local address. See Quick mode selectors on page 2798.
Remote Address	*Tunnel Mode* only. Enter the destination IP address that corresponds to the recipients or network behind the remote VPN peer. A value of `0.0.0.0/0` means all IP addresses behind the remote VPN peer. See Quick mode selectors on page 2798.
Advanced	Select the encryption and authentication algorithms that will be proposed to the remote VPN peer. To establish a VPN connection, at least one of the proposals specified must match the configuration on the remote peer.
Encryption	The following symmetric-key encryption algorithms are available: • *NULL*: do not use an encryption algorithm. **Not recommended**.

Command	Description
	• *DES*: Digital Encryption Standard, a 64-bit block algorithm that uses a 56-bit key. **Not recommended**. • *3DES*: triple-DES; plain text is encrypted three times by three keys. **Not recommended**. • *AES128*: Advanced Encryption Standard, a 128-bit block algorithm that uses a 128-bit key. • *AES128GCM*: AES in Galois/Counter Mode, a 128-bit block algorithm that uses a 128-bit key. Only available for IKEv2. • *AES192*: a 128-bit block algorithm that uses a 192-bit key. • *AES256*: a 128-bit block algorithm that uses a 256-bit key. • *AES256GCM*: AES in Galois/Counter Mode, a 128-bit block algorithm that uses a 256-bit key. Only available for IKEv2. • *CHACHA20POLY1305*: a 128-bit block algorithm that uses a 128-bit key and a symmetric cipher. Only available for IKEv2. See ChaCha20 and Poly1305 AEAD cipher on page 2800, AES-GCM for IKEv2 phase 1 on page 2800, and HMAC settings.
Authentication	The following message digests that check the message authenticity during an encrypted session are available: • *NULL*: do not use a message digest. **Not recommended**. • *MD5*: message digest 5. **Not recommended**. • *SHA1*: secure hash algorithm 1; a 160-bit message digest. **Not recommended**. • *SHA256*: a 256-bit message digest. • *SHA384*: a 384-bit message digest. • *SHA512*: a 512-bit message digest. See also HMAC settings.
L2TP	Enable or disable L2TP over IPsec. Typically used by native VPN clients such as Windows VPN client. See Native VPN as dialup client - Windows New on page 3043.
Replay Detection	Replay attacks occur when an unauthorized party intercepts a series of IPsec packets and replays them back into the tunnel. Replay detection allows the FortiGate to check all IPsec packets to see if they have been received before. If any encrypted packets arrive out of order, the FortiGate discards them. Note that 64-bit extended sequence numbers (as described in RFC 4303, RFC 4304 as an addition to IKEv1, and RFC 5996 for IKEv2) are supported for IPsec when replay detection is enabled.
Perfect Forward Secrecy (PFS)	Perfect forward secrecy (PFS) improves security by forcing a new Diffie-Hellman exchange whenever keylife expires.
Diffie-Hellman Group	Asymmetric key algorithms used for public key cryptography.

Command	Description
	Select one or more from groups 1, 2, 5, and 14 through 32. At least one of the *Diffie-Hellman Groups* (DH) settings on the remote peer or client must match one the selections on the FortiGate. Failure to match one or more DH groups will result in failed negotiations. Fortinet recommends choosing DH groups in the range 15-21 and 28-32 for security best practices.
Post Quantum Cryptography Additional Key Exchanges	Enable quantum-resistant encryption by applying hybrid key exchange per RFC 9370 and RFC 9242. When creating a new PQC Additional KE group: • *Transform Type*: Each Transform type represents an additional round of KE exchange. Choose up to seven rounds or groups • *KE Groups*: Choose up to three Key Exchange groups from these Key Exchange Mechanisms (KEMs): *ML KEM (Kyber)*, *BIKE*, *HQC*, and *FRODO*. For more information, see Post-Quantum Cryptography for IPsec key exchange on page 2822.
Local Port	*Tunnel Mode* only. Enter the port number that the local VPN peer uses to transport traffic related to the specified service (protocol number). The range is from 0 to 65535. To specify all ports, select *All*, or enter `0`.
Remote Port	*Tunnel Mode* only. Enter the port number that the remote VPN peer uses to transport traffic related to the specified service (protocol number). To specify all ports, select *All*, or enter `0`.
Protocol	*Tunnel Mode* only. Enter the IP protocol number of the service. To specify all services, select *All*, or enter `0`.
Auto-negotiate	Select this option for the tunnel to be automatically renegotiated when the it expires. See Auto-negotiate on page 2799. This option is only available when the *Remote Gateway* mode in phase 1 is not a dialup user.
Autokey Keep Alive	Select this option for the tunnel to remain active when no data is being processed.
Key Lifetime	Select the method for determining when the phase 2 key expires: • *Seconds* • *Kilobytes* • *Both* Enter a corresponding value for *Seconds* and/or *Kilobytes* in the text boxes. If *Both* is selected, the key expires when either the time has passed or the number of kilobytes have been processed.

Quick mode selectors

Quick mode selectors determine which IP addresses can perform IKE negotiations to establish a tunnel. By only allowing authorized IP addresses access to the VPN tunnel, the network is more secure.

The default settings are as broad as possible: any IP address or configured address object using any protocol on any port.

> ⚠ While the dropdown menus for specifying an address also show address groups, the use of address groups may not be supported on a remote endpoint device that is not a FortiGate.

When configuring a quick mode selector for *Local Address* and *Remote Address*, valid options include IPv4 and IPv6 single addresses, subnets, or ranges.

There are some configurations that require specific selectors:

- The VPN peer is a third-party device that uses specific phase2 selectors.
- The FortiGate connects as a dialup client to another FortiGate, in which case (usually) you must specify a local IP address, IP address range, or subnet. However, this is not required if you are using dynamic routing and `mode-cfg`.

With FortiOS VPNs, your network has multiple layers of security, with quick mode selectors being an important line of defense:

- Routes guide traffic from one IP address to another.
- Phase 1 and phase 2 connection settings ensure there is a valid remote end point for the VPN tunnel that agrees on the encryption and parameters.
- Quick mode selectors allow IKE negotiations only for allowed peers.
- Security policies control which IP addresses can connect to the VPN.
- Security policies also control what protocols are allowed over the VPN along with any bandwidth limiting.

If you are editing an existing phase 2 configuration, the local address and remote address fields are unavailable if the tunnel has been configured to use firewall addresses as selectors. This option exists only in the CLI.

Using the add-route option

Consider using the `add-route` option to add a route to a peer destination selector in phase 2 to automatically match the settings in phase 1.

To configure add-route:

```
config vpn ipsec {phase2 | phase2-interface}
    edit <name>
        set add-route {phase1 | enable | disable}
    next
end
```

Auto-negotiate

By default, the phase 2 security association (SA) is not negotiated until a peer attempts to send data. The triggering packet and some subsequent packets are dropped until the SA is established. Applications normally resend this data, so there is no loss, but there might be a noticeable delay in response to the user.

If the tunnel goes down, the auto-negotiate feature (when enabled) attempts to re-establish the tunnel. Auto-negotiate initiates the phase 2 SA negotiation automatically, repeating every five seconds until the SA is established.

Automatically establishing the SA can be important for a dialup peer. It ensures that the VPN tunnel is available for peers at the server end to initiate traffic to the dialup peer. Otherwise, the VPN tunnel does not exist until the dialup peer initiates traffic.

To configure auto-negotiate:

```
config vpn ipsec phase2
    edit <phase2_name>
        set auto-negotiate enable
    next
end
```

Installing dynamic selectors via auto-negotiate

The IPsec SA connect message generated is used to install dynamic selectors. These selectors can be installed via the auto-negotiate mechanism. When phase 2 has `auto-negotiate` enabled, and phase 1 has `mesh-selector-type` set to `subnet`, a new dynamic selector will be installed for each combination of source and destination subnets. Each dynamic selector will inherit the auto-negotiate option from the template selector and begin SA negotiation. Phase 2 selector sources from dialup clients will all establish SAs without traffic being initiated from the client subnets to the hub.

DHCP

The `dhcp-ipsec` option lets the FortiGate assign IP addresses to FortiClient dialup clients through a DHCP server or relay in IKEv1. This option is only available if the remote gateway in the phase 1 configuration is set to dialup user and `mode-cfg` is disabled.

With `dhcp-ipsec`, the FortiGate dialup server acts as a proxy for FortiClient dialup clients that have VIP addresses on the subnet of the private network behind the FortiGate. In this case, the FortiGate dialup server acts as a proxy on the local private network for the FortiClient dialup client. A host on the network behind the dialup server issues an ARP request, corresponding to the device MAC address of the FortiClient host (when a remote server sends an ARP to the local FortiClient dialup client). The FortiGate then answers the ARP request on behalf of the FortiClient host, and then forwards the associated traffic to the FortiClient host through the tunnel.

Acting as a proxy prevents the VIP address assigned to the FortiClient dialup client from causing possible ARP broadcast problems—the normal and VIP addresses can confuse some network switches when two addresses have the same MAC address.

Bypassing the FortiGate and using DHCP is not recommended. Using `mode-cfg` or IPAM is preferred.

ChaCha20 and Poly1305 AEAD cipher

In IKEv2 to support RFC 7634, the ChaCha20 and Poly1305 crypto algorithms can be used together as a combined mode AEAD cipher (like AES-GCM) in the `crypto_ftnt` cipher in `cipher_chacha20poly1305.c`:

```
config vpn ipsec phase2-interface
    edit <name>
        set phase1name <name>
        set proposal chacha20poly1305
    next
end
```

AES-GCM for IKEv2 phase 1

In IKEv2 to support RFC 5282, the AEAD algorithm AES-GCM supports 128- and 256-bit variants:

```
config vpn ipsec phase2-interface
    edit <name>
        set phase1name <name>
        set proposal [aes128gcm | aes256gcm]
    next
end
```

Tunneling protocol and encapsulation

Fortinet's IPsec VPN offers the following options for tunneling and encapsulation:

- Native ESP
- UDP encapsulation
- TCP encapsulation with Fortinet proprietary extension
- TCP encapsulation using RFC 8229
- TCP encapsulation using RFC 8229 with TLS 1.3

When ESP is used without encapsulation, it connects directly over IP Protocol 50. When ESP is encapsulated within UDP, it uses UDP/500 and UDP/4500 for NAT traversal, which are the options for dialup IPsec VPN. Furthermore the port can be modified on a per-VDOM basis.

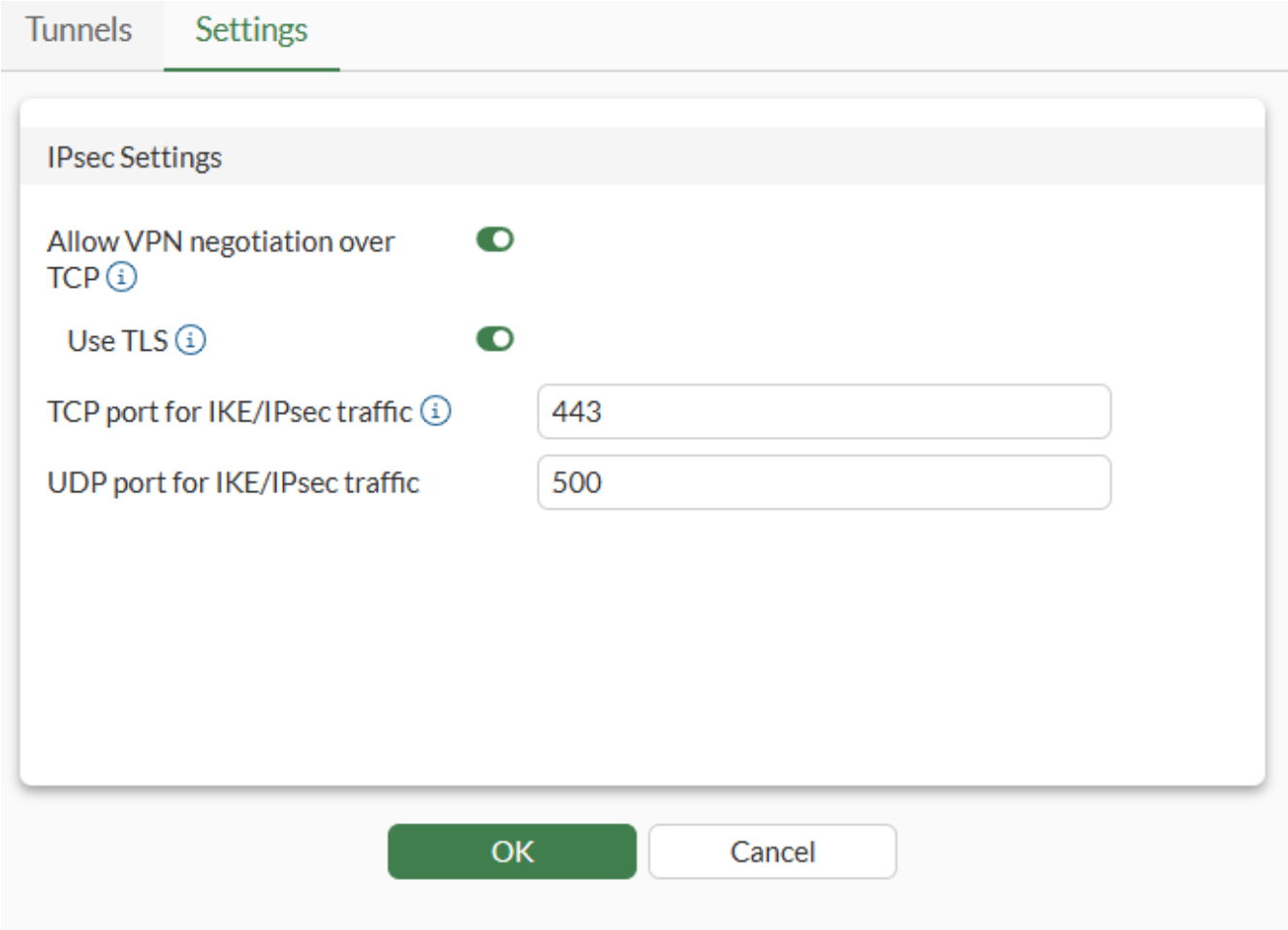
FortiGate can also operate in an Auto-transport mode where UDP will be attempted first before failing over to TCP or TCP with TLS. By default, TCP encapsulation and Auto-transport is disabled per VDOM. However, once enabled, tunnels can be changed to use Auto-transport mode.

This allows IPsec to encapsulate ESP packets within TCP and operate on TCP port 443 (default), enabling ESP packets to traverse carrier networks where direct IPsec traffic is blocked or impeded by carrier-grade NAT.

This feature requires using IKE version 2. For remote access over dialup tunnels, it also requires FortiClient 7.4.1 or later.

Tunnel and encapsulation settings

In the GUI, go to *VPN* > *VPN Tunnels* and select the *Settings* tab:



In the CLI:

```
config system settings
    set ike-tcp-service {enable | disable}
    set ike-tls-service {enable | disable}
end
```

Both settings are per-vdom settings to control whether IKE over TCP and TLS is used in VPN tunnels. These settings are disabled by default.

When `ike-tcp-service` is enabled, dialup VPNs configured in the VDOM will behave as the *Auto* transport option - transport over UDP will be preferred, but when the IKE connection cannot be made over UDP, communication will be attempted in TCP or TLS. For site-to-site VPNs, you must set the transport option to *Auto* to use TCP or TLS.

When disabled (default), dialup VPNs will not attempt to connect over TCP.

Upgrading from 7.6 and lower:

1. If a dynamic-type dialup VPN with auto or TCP transport mode exists, the `ike-tcp-service` setting is enabled for the VDOM.
2. If a static VPN with TCP transport mode exists, the `ike-tcp-service` setting is enabled for the VDOM.
3. If a static VPN with auto transport mode exists, that VPN will be changed to use transport UDP.

For more information:

- Configurable UDP port for IKE on page 2802
- Encapsulate ESP packets within TCP headers on page 2809
- GUI warnings for IKE-TCP port conflicts on page 2811
- TLS 1.3 based VPN over TCP on page 2815

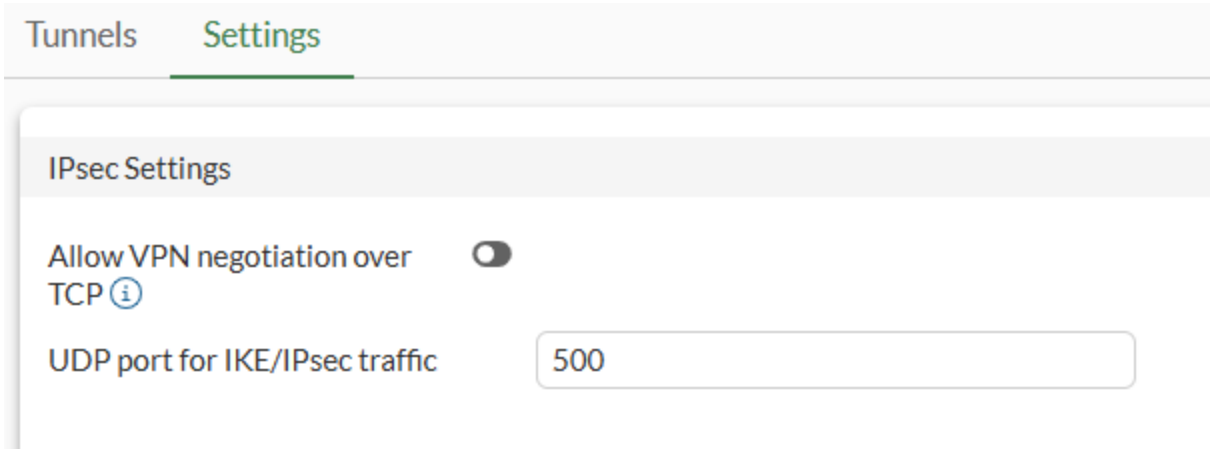
Configurable UDP port for IKE

Some ISPs block UDP port 500 or UDP port 4500, preventing an IPsec VPN from being negotiated and established. FortiOS allows the IKE port to be changed to an alternate UDP port to work around this restriction. Additionally, up to three extra UDP ports can be configured to receive IKE negotiations simultaneously. These ports are automatically added to the local-in policy so that IKE traffic can pass through the firewall without manual policy configuration.

Even if the IKE port is changed, FortiGate will continue to listen on UDP port 50 and 4500, per RFC requirements.

To set the IKE port in the GUI:

1. Go to *VPN > VPN Tunnels* and select the *Settings* tab.
2. Edit the *UDP port for IKE/IPsec traffic* field.



3. Click *OK*.

To set the IKE port in the CLI:

```
config system settings
    set ike-port <integer>
end
```

Command	Description
`ike-port <integer>`	UDP port for IKE/IPsec traffic (1024 - 65535, default = 500, special = 500 and 443).

To set the IKE extra port in the CLI:

```
config system settings
    set ike-extra-ports <value> [<value>] [<value>]
end
```

Command	Description
`ike-extra-ports <value> [<value>] [<value>]`	Additional UDP ports to listen on for IKE negotiations, beyond ports 500 and 4500 (1024 - 65535, special = 443). Cannot be set to 500 or 4500. Cannot overlap with `ike-port`.

Example 1: site-to-site VPN without NAT

In this example, the IKE port is set to 6000 on the two site-to-site VPN gateways. There is no NAT between the VPN gateways, but the ISP has blocked UDP port 500. A site-to-site VPN is established using the defined IKE port.

To set the IKE port:

```
config system settings
    set ike-port 6000
end
```

To configure and check the site-to-site VPN:

1. Configure the phase1 and phase2 interfaces:

```
config vpn ipsec phase1-interface
    edit "s2s"
        set interface "port27"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set wizard-type static-fortigate
        set remote-gw 11.101.1.1
```

```
        set psksecret **********
    next
end
```

```
config vpn ipsec phase2-interface
    edit "s2s"
        set phase1name "s2s"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
        set src-addr-type name
        set dst-addr-type name
        set src-name "s2s_local"
        set dst-name "s2s_remote"
    next
end
```

2. Check the IKE gateway list and confirm that the specified port is used:

```
# diagnose vpn ike gateway list

vd: root/0
name: s2s
version: 2
interface: port27 17
addr: 173.1.1.1:6000 -> 11.101.1.1:6000
tun_id: 11.101.1.1
remote_location: 0.0.0.0
created: 194s ago
PPK: no
IKE SA: created 1/2  established 1/2  time 0/4500/9000 ms
IPsec SA: created 1/2  established 1/2  time 0/4500/9000 ms
...
```

3. Check the VPN tunnel list:

```
# diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=s2s ver=2 serial=1 173.1.1.1:6000->11.101.1.1:6000 tun_id=11.101.1.1 dst_mtu=1500 dpd-
link=on remote_location=0.0.0.0 weight=1
bound_if=17 lgwy=static/1 tun=tunnel/15 mode=auto/1 encap=none/520 options[0208]=npu frag-rfc
run_state=0 accept_traffic=1 overlay_id=0
...
```

Example 2: dialup VPN with NAT

In this example, the IKE port is set to 5000 on the VPN gateway and the dialup peer. The dialup peer is behind NAT, so NAT traversal (NAT-T) is used. The ISP blocks both UDP port 500 and UDP port 4500. The VPN connection is initiated on UDP port 5000 from the dialup VPN client and remains on port 5000 since NAT-T floating to 4500 is only required when the IKE port is 500.

To set the IKE port:

```
config system settings
    set ike-port 5000
end
```

To configure and check the dialup VPN with NAT:

1. Configure the phase1 and phase2 interfaces:

```
config vpn ipsec phase1-interface
    edit "server"
        set type dynamic
        set interface "port27"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set dpd on-idle
        set wizard-type static-fortigate
        set psksecret **********
        set dpd-retryinterval 60
    next
end
```

```
config vpn ipsec phase2-interface
    edit "server"
        set phase1name "server"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
        set src-addr-type name
        set dst-addr-type name
        set src-name "server_local"
        set dst-name "server_remote"
    next
end
```

2. Check the IKE gateway list and confirm that the specified port is used:

```
# diagnose vpn ike gateway list

vd: root/0
name: server_0
version: 2
interface: port27 17
addr: 173.1.1.1:5000 -> 173.1.1.2:65416
tun_id: 173.1.1.2
remote_location: 0.0.0.0
created: 90s ago
nat: peer
PPK: no
```

```
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms
...
```

3. Check the VPN tunnel list:

```
# diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=server_0 ver=2 serial=a 173.1.1.1:5000->173.1.1.2:65416 tun_id=173.1.1.2 dst_mtu=1500
dpd-link=on remote_location=0.0.0.0 weight=1
bound_if=17 lgwy=static/1 tun=tunnel/15 mode=dial_inst/3 encap=none/904 options[0388]=npu
rgwy-chg rport-chg frag-rfc  run_state=0 accept_traffic=1 overlay_id=0
...
```

Example 3: UDP port 443 for dialup IPsec VPN

This example demonstrates a simple VPN configuration utilizing IKEv2 over UDP/443 between a Dialup VPN Gateway (FGTD) and a VPN Client (FGTA) where both ends utilize UDP/443 for IKEv2 negotiations.

To configure the hub FortiGate, FGTD:

1. Configure the IKE port:

```
config system settings
    set ike-port 443
end
```

2. Configure an IPSec VPN tunnel:

```
config vpn ipsec phase1-interface
    edit "tofgta"
        set type dynamic
        set interface "port13"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set dhgrp 20 21
        set transport auto
        set psksecret <password>
    next
end
```

```
config vpn ipsec phase2-interface
    edit "tofgta"
```

```
        set phase1name "tofgta"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
        set dhgrp 20 21
    next
end
```

3. Configure a firewall policy:

```
config firewall policy
    edit 1
        set srcintf "any"
        set dstintf "any"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
    next
end
```

To configure the spoke FortiGate, FGTA:

1. Configure the IKE port:

```
config system settings
    set ike-port 443
end
```

2. Configure an IPSec VPN tunnel:

```
config vpn ipsec phase1-interface
    edit "tofgtd"
        set interface "port9"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set dhgrp 20 21
        set transport auto
        set remote-gw 173.1.1.1
        set psksecret <password>
    next
end
```

```
config vpn ipsec phase2-interface
    edit "tofgtd"
        set phase1name "tofgtd"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
```

```
chacha20poly1305
        set dhgrp 20 21
    next
end
```

3. Configure a firewall policy:

```
config firewall policy
    edit 1
        set srcintf "any"
        set dstintf "any"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
    next
end
```

To verify the IPsec VPN tunnel status:

Note that `addr` shows the UDP custom port value and `transport` shows `UDP`.

```
# diagnose vpn ike gateway list
vd: root/0
name: tofgtd
version: 2
interface: port9 15
addr: 11.101.1.1:443-> 173.1.1.1:443
tun_id: 173.1.1.1/::173.1.1.1
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 642s ago
peer-id: 173.1.1.1
peer-id-auth: no
nat: me
pending-queue: 0
PPK: no
IKE SA: created 1/1  established 1/1  time 10/10/10 ms
IPsec SA: created 1/1  established 1/1  time 10/10/10 ms

  id/spi: 5 ea6ed73ac96d8151/df3decebc465a65f
  direction: initiator
  status: established 642-642s ago = 10ms
  proposal: aes128-sha256
  child: no
  SK_ei: d09da40c2a7b9989-cf5538d9462a81e4
  SK_er: 6a9d32b1b60a201b-aec078baabffa9c7
  SK_ai: 248682bcff1866ea-444582d14090b9ac-de08d11d6a5f58a7-5e3fdb29f53d25c2
```

```
SK_ar: caa2d26a92aac4ad-bcd92aa0018aa398-e3088448964778cb-499e8b05acffb64b
PPK: no
message-id sent/recv: 2/0
QKD: no
PQC-KEM (IKE): no
PQC-KEM (all IPsec): no
lifetime/rekey: 86400/85457
DPD sent/recv: 00000000/00000000
peer-id: 173.1.1.1
```

Encapsulate ESP packets within TCP headers

FortiOS supports encapsulation of IKE and ESP packets within Transmission Control Protocol (TCP) headers, in accordance with RFC 8229. This allows IKE & ESP packets to be assigned a TCP port number that enables them to traverse over carrier networks where direct IPsec traffic is blocked or impeded by carrier-grade NAT. This standards-based TCP encapsulation method is also supported across multiple vendors, ensuring that you can maintain a secure and efficient network, while also having the flexibility to choose the hardware that aligns best with your requirements.

This feature only works with IKE version 2, and it does not support ADVPN or NPU offloading.

You can choose between a standards-based (RFC 8229) or Fortinet-proprietary method to encapsulate IKE and ESP traffic within TCP headers. This table compares the encapsulation methods. Using the standards-based method is recommended.

Encapsulation method	Dialup IPsec VPN using FortiClient	IPsec VPN between FortiGate to FortiGate	IPsec VPN between FortiGate and 3rd party device	Support TLS 1.3
RFC 8229 compliant (`fortinet-esp` disabled)	Supported. For example, see Dialup IPsec VPN using custom TCP port on page 3159.	Supported.	Supported.	Yes. See TLS 1.3 based VPN over TCP on page 2815.
Fortinet propriety (`fortinet-esp` enabled)	Not supported	Supported. See VPN over TCP with Fortinet proprietary encapsulation on page 2905.	Not supported	No

To configure TCP encapsulation for IPsec VPN:

1. Go to *VPN > VPN Tunnels* and select the *Settings* tab.
2. Enable *Allow VPN negotiation over TCP*.
3. Optionally, enable *TCP*.
4. Optionally, change the *TCP port*.

5. Click *OK*.

To configure TCP encapsulation for IPsec VPN in the CLI:

```
config system settings
    set ike-tcp-service {enable | disable}
    set ike-tls-service {enable | disable}
    set ike-tcp-port <port>
end
```

Command	Description
`ike-tcp-service {enable | disable}`	Enable/disable the use of TCP transport for IKE and ESP. Disabled by default. This is a per-VDOM setting
`ike-tls-service {enable | disable}`	When ike-tcp-service is enabled, enable/disable the use of TLS 1.3
`ike-tcp-port <port>`	Set the TCP port for IKE/IPsec traffic (1 - 65535, default = 443). By default, IKE listens on port 4500 when ike-tcp-port is configured as 4500. If ike-tcp-port is configured on any other port, the IKE daemon listens on port 4500 and port 11443 while traffic enters on the ike-tcp-port and redirects to port 11443

```
config vpn ipsec phase1-interface
    edit <name>
        set ike-version 2
        set transport {auto | udp}
        set fortinet-esp {enable | disable}
    next
end
```

Command	Description
`transport {auto | udp}`	This setting is only available in a site-to-site VPN configuration (type is either static or ddns). Set the IKE transport protocol: • `auto`: use UDP transport for IKE and ESP, with automatic fallback to TCP transport if UDP attempt fails for timeout threshold. • `udp`: use UDP transport for IKE and ESP only.
`fortinet-esp {enable | disable}`	The Fortinet propriety feature is designed to offload IPsec VPN traffic to Fortinet's NP (Network Processor) ASICs to improve performance. This command enables or disables encapsulation of ESP (Encapsulating Security Payload) packets within non-standard TCP headers. • `disable`: Encapsulates ESP packets using standard TCP headers. This is the default option. • `enable`: Encapsulates ESP packets using non-standard TCP headers.

Command	Description
	This feature is not supported in the following scenarios: • When FortiGate is configured as Dialup IPsec VPN server for remote access when using FortiClient as dialup client. • In multi-vendor environments that use TCP encapsulation for ESP packets. Make sure that this setting is disabled for these two scenarios to ensure uninterrupted ESP packet flow encapsulated within standard TCP headers.

When using TCP port 443 for IKE/IPsec traffic, GUI access can be affected for interfaces that are bound to an IPsec tunnel when the GUI admin port is also using port 443. To ensure continued functionality, change either the IKE/IPsec port or the administrative access port. See GUI warnings for IKE-TCP port conflicts on page 2811.

To change the administrative access port:

```
config system global
    set admin-sport <port>
end
```

`admin-sport <port>`	Set the administrative access port for HTTPS (1 - 65535, default = 443).

For port conflicts with ZTNA and Agentless VPN, ZTNA and Agentless VPN will take precedence. To avoid any port conflicts with other services, review the FortiOS Ports guide for other incoming ports used on the FortiGate.

GUI warnings for IKE-TCP port conflicts

TCP port 443 is used by default when TCP over VPN is enabled to encapsulate ESP packets within TCP headers.

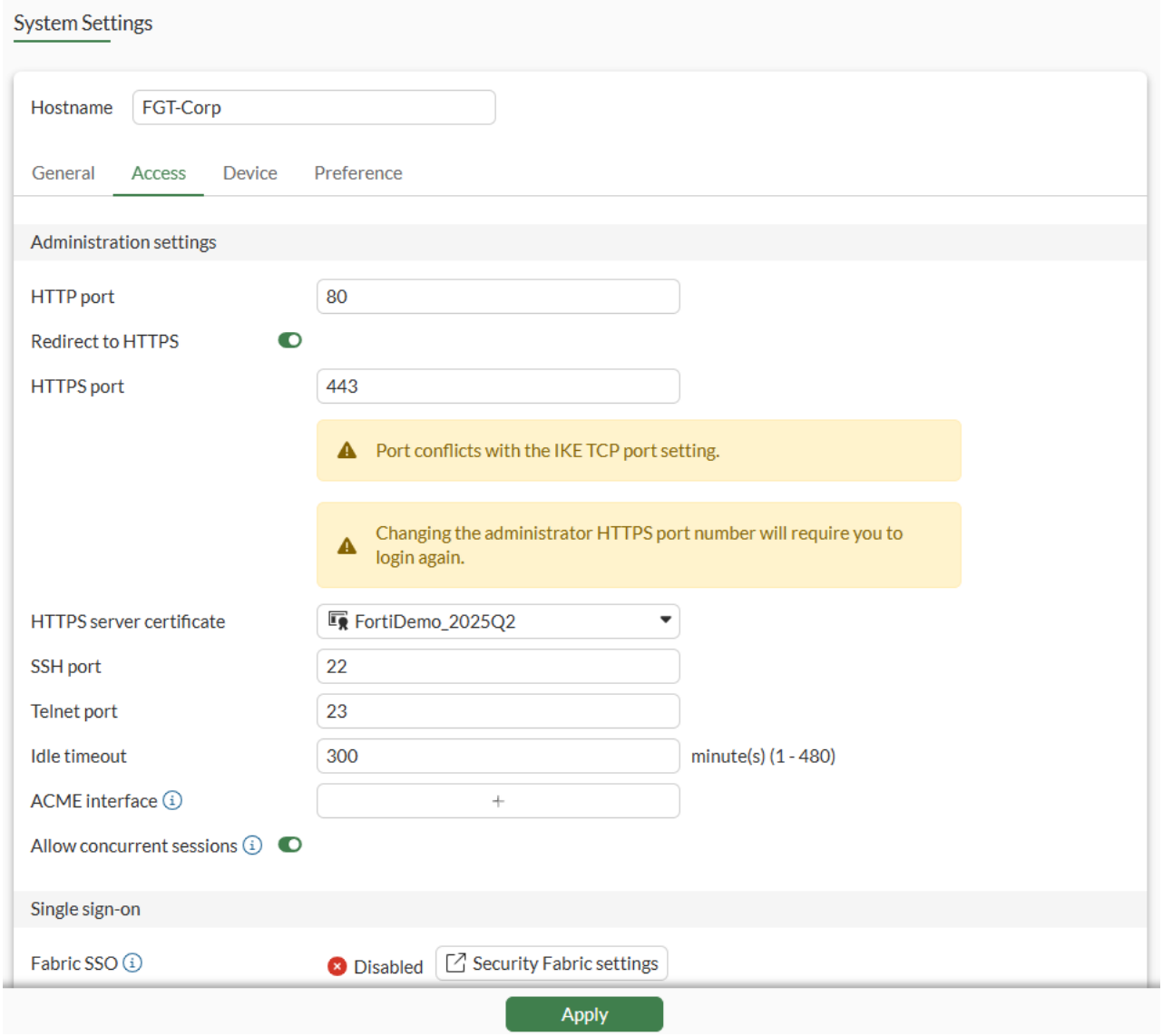
If administrators assign port 443 for HTTPS administrative access on an interface that is also bound to an IPsec tunnel, FortiOS will display a warning indicating that HTTPS access on that port will no longer be available. This is because port 443 is also used for IKE over TCP, and in such cases, IKE takes precedence over HTTPS, resulting in the loss of GUI access on that interface.

The default IKE-TCP value of port 443 is only applicable to new FortiGate configurations with FortiOS 7.6.1 and above. If FortiOS is upgraded to 7.6.1 and above, the `ike-tcp-port` value from before the upgrade is retained.

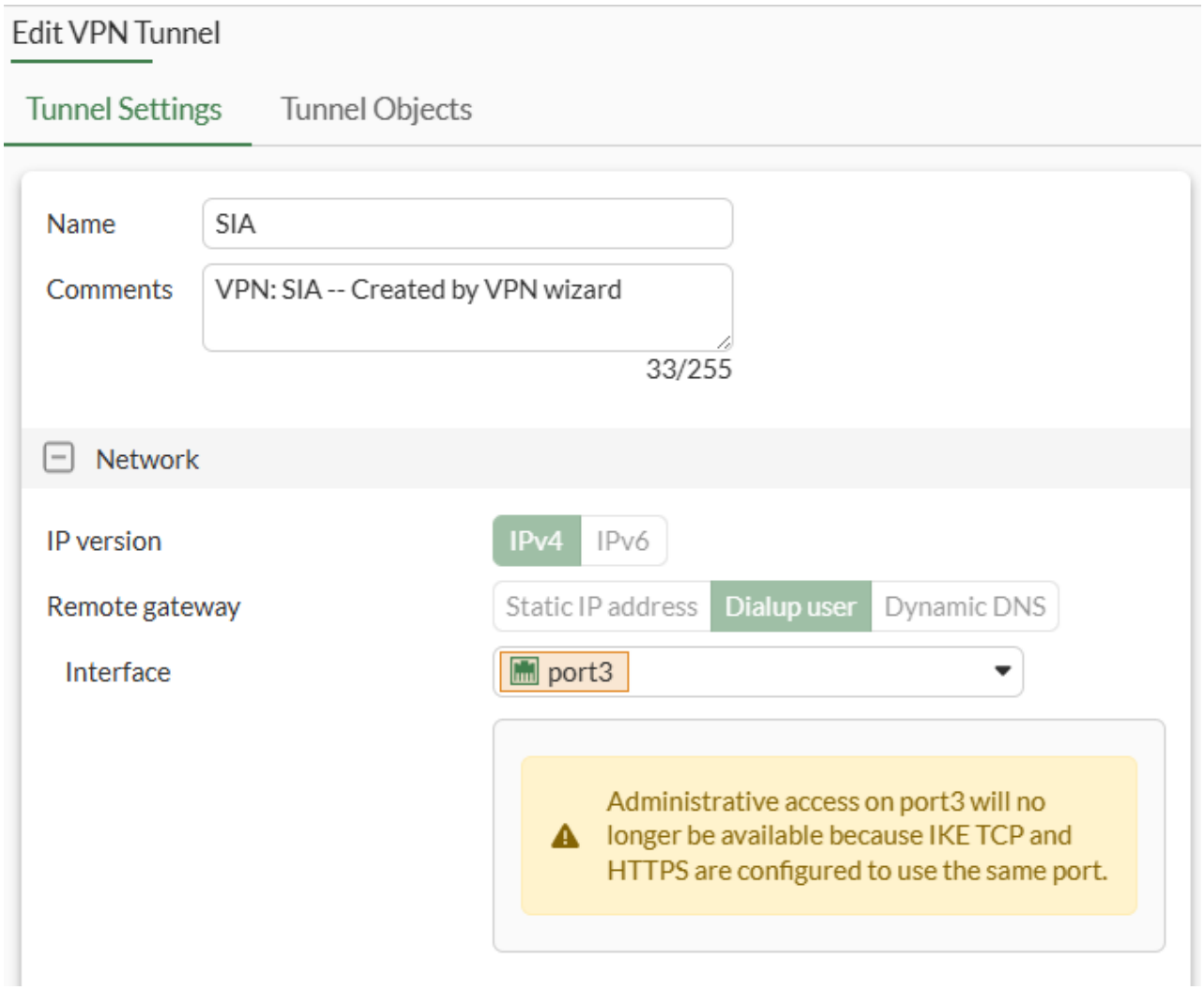
Furthermore, by default IKE listens on port 4500 when `ike-tcp-port` is configured as 4500. If `ike-tcp-port` is configured on any other port, the IKE daemon listens on port 4500 and port 11443 while traffic enters on the `ike-tcp-port` and redirects to port 11443. Therefore, when the `ike-tcp-port` is set to 443 by default, do not use port 443, 11443, or 4500 for HTTPS GUI access.

Warnings that may appear include the following:

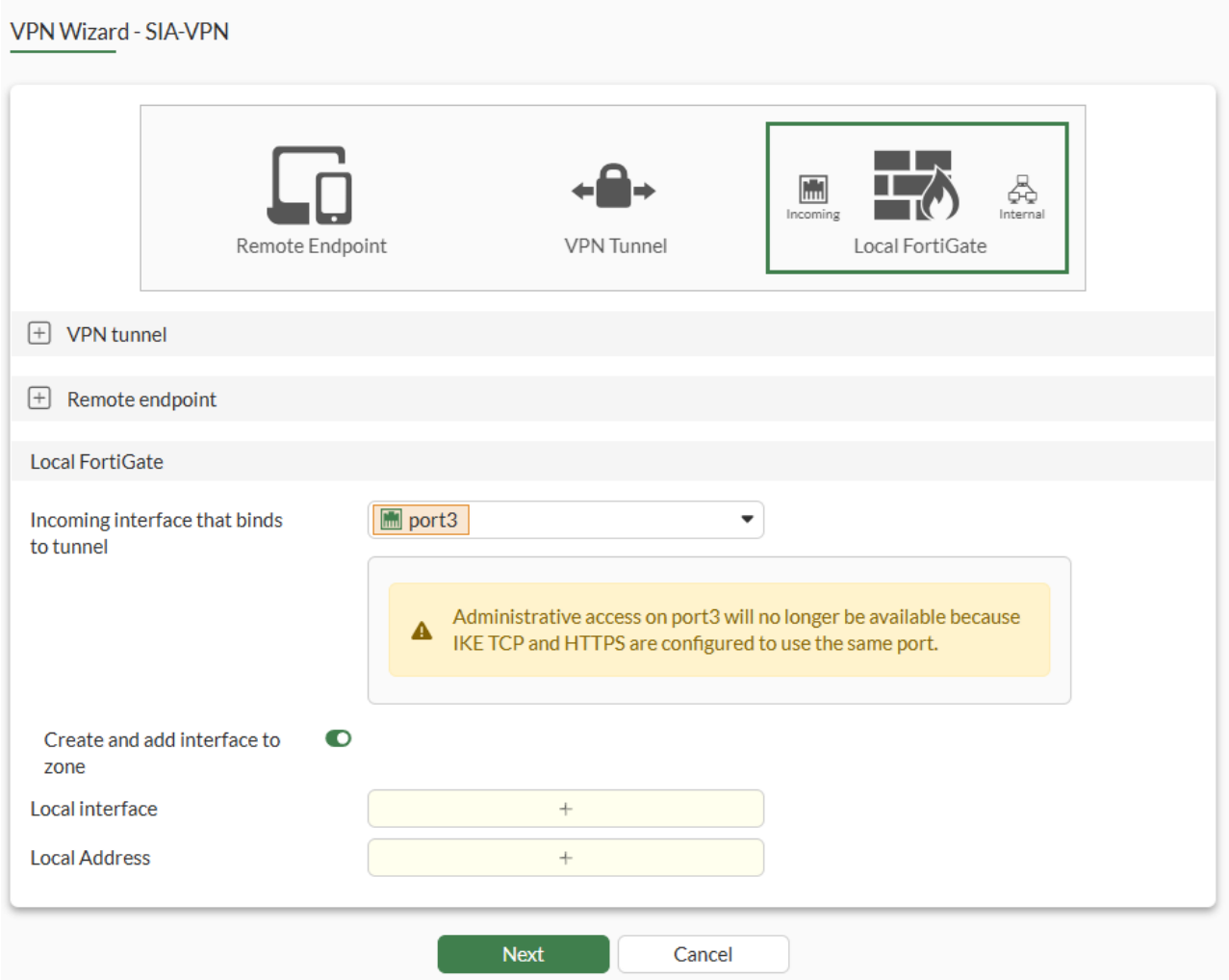
- In *System > Settings > Access*, a warning is displayed for the *Administrator Settings > HTTPS port* field when the entered HTTPS port conflicts with the IKE-TCP port. This happens even if *Allow VPN negotiation over TCP* is disabled.



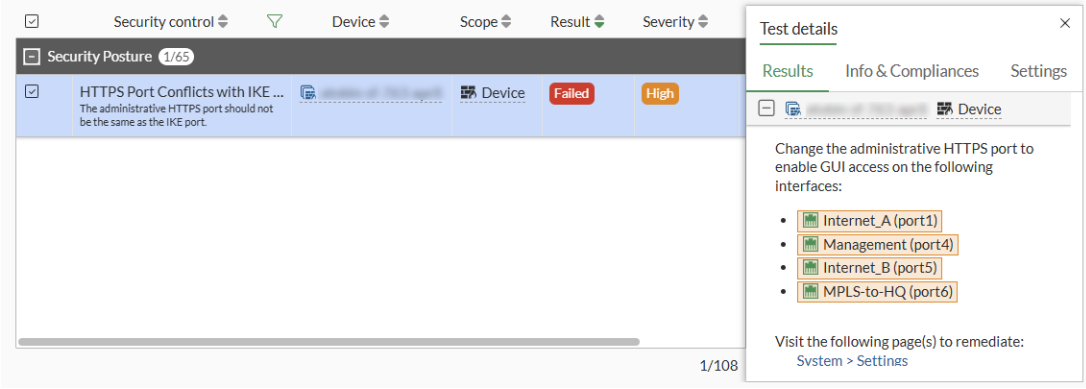
- In *VPN > VPN Tunnels*, a warning is displayed for the *Network > Interface* field when the HTTPS port conflicts with IKE-TCP port and the selected port has HTTPS allow access.



- In *VPN > VPN Wizard*, a warning is displayed in the *Local FortiGate* configuration page where *Incoming interface that binds to tunnel* field has a port conflicts with IKE-TCP port and the selected port has HTTPS allow access.



- In *Security Fabric > Security Rating*, a *Security Posture* failure is flagged for the *HTTPS Port Conflict with IKE Port* check, indicating a configuration issue. Likewise, a *Security Ratings Insights* recommendation is listed. This occurs when the HTTPS port conflicts with the IKE port when a tunnel is configured and TCP over VPN is enabled.



TLS 1.3 based VPN over TCP

IKE based VPN tunnels can be configured to be ESP, UDP or TCP based. When TCP is used as the transport protocol, IKE and ESP packets are encapsulated in TCP in the port that you specified.

In this enhancement, TLS 1.3 is added as an option for VPN over TCP. The TLS option can be selected when *Allow VPN negotiation over TCP* is enabled in the *VPN > VPN Tunnels > Settings* page. With TLS enabled, IKE and ESP traffic are transported over TLS.

This option is supported for Site-to-Site and Dial-up VPN between FortiGates. FortiClient endpoints do not currently support TLS based VPN over TCP.

For information about configurations and usage, see Tunnel and encapsulation settings on page 2801.

Example

An IKEv2 over TLS 1.3 VPN tunnel is formed between a VPN Client (FGTA) and a dialup VPN Gateway (FGTD).

To configure the hub FortiGate, FGTD, in the GUI:

1. Go to *VPN > VPN Tunnels* and select the *Settings* tab.
2. Enable *Allow VPN negotiation over TCP*.
3. Enable *Use TLS*.
4. Click *OK*.
5. Go to *VPN > VPN Tunnels*, select the *Tunnels* tab, and click *Create new > Custom IPsec tunnel*.
6. Configure the following settings:

Field	Value
Name	tofgta
Remote gateway	Dialup user
Interface	port13
Authentication Method	Signature
Certificate name	Select your certificate
IKE	Version 2 (default)

7. In the *Phase 2 selectors* table, click *Create New* and configure the following:

Field	Value
Name	tofgta
Encryption – authentication	Remove all SHA1 based entries

8. Click *OK* to save the phase 2 selector.
9. Click *OK* to save the tunnel.
10. Create firewall policies to allow traffic from *tofgta* to a protected network, and to allow traffic from a protected network to *tofgta*.

To configure the hub FortiGate, FGTD, in the CLI:

```
config system settings
    set ike-tcp-service enable
    set ike-tls-service enable
end
```

```
config vpn ipsec phase1-interface
    edit "tofgta"
        set type dynamic
        set interface "port13"
        set ike-version 2
        set authmethod signature
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set dhgrp 20 21
        set certificate "test1"
    next
end
```

```
config vpn ipsec phase2-interface
    edit "tofgta"
        set phase1name "tofgta"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
    next
end
```

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "tofgta"
        set action accept
        set srcaddr "192.168.5.0"
        set dstaddr "10.1.100.0"
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set srcintf "tofgta"
        set dstintf "port2"
        set action accept
        set srcaddr "10.1.100.0"
        set dstaddr "192.168.5.0"
```

```
        set schedule "always"
        set service "ALL"
    next
end
```

To configure the dialup client FortiGate, FGTA, on the GUI:

1. Go to *VPN* > *VPN Tunnels* and select the *Settings* tab.
2. Enable *Allow VPN negotiation over TCP*.
3. Enable *Use TLS*.
4. Click *OK*.
5. Go to *VPN* > *VPN Tunnels*, select the *Tunnels* tab, and click *Create new* > *Custom IPsec tunnel*.
6. Configure the following settings:

Field	Value
Name	tofgtd
Remote gateway	Static IP address
IP address	173.1.1.1
Interface	port9
Transport	Auto
Authentication Method	Signature
Certificate name	Select your certificate
IKE	Version 2 (default)

7. In the *Phase 2 selectors* table, click *Create New* and configure the following:

Field	Value
Name	tofgtd
Encryption – authentication	Remove all SHA1 based entries

8. Click *OK* to save the phase 2 selector.
9. Click *OK* to save the tunnel.
10. Create firewall policies to allow traffic from *tofgtd* to a protected network, and to allow traffic from a protected network to *tofgtd*.

To configure the dialup client FortiGate, FGTA, on the CLI:

```
config system settings
    set ike-tcp-service enable
    set ike-tls-service enable
end
```

```
config vpn ipsec phase1-interface
    edit "tofgtd"
        set interface "port9"
        set ike-version 2
        set authmethod signature
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set dhgrp 20 21
        set transport auto
        set remote-gw 173.1.1.1
        set certificate "test3"
    next
end
```

```
config vpn ipsec phase2-interface
    edit "tofgtd"
        set phase1name "tofgtd"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
    next
end
```

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "tofgtd"
        set action accept
        set srcaddr "10.1.100.0"
        set dstaddr "192.168.5.0"
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set srcintf "tofgtd"
        set dstintf "port2"
        set action accept
        set srcaddr "192.168.5.0"
        set dstaddr "10.1.100.0"
        set schedule "always"
        set service "ALL"
    next
end
```

Verification

When the tunnel is established, run the following debugs on FGTA to verify:

```
# diagnose vpn ike gateway list
```

```
vd: root/0
name: tofgtd
version: 2
interface: port9 15
addr: 11.101.1.1:443 -> 173.1.1.1:7179
tun_id: 173.1.1.1/::173.1.1.1
remote_location: 0.0.0.0
network-id: 0
transport: TLS
tcp-tx-1m-rate : 0 kbps
tcp-tx-5m-rate : 0 kbps
tcp-tx-drop    : 0
tcp-rx-1m-rate : 0 kbps
tcp-rx-5m-rate : 0 kbps
tcp-rx-drop    : 0
created: 4968s ago
peer-id: C = CA, ST = BC, L = Burnaby, O = Fortinet, OU = QA, CN = test3
peer-id-auth: yes
pending-queue: 0
PPK: no
IKE SA: created 1/2  established 0/1  time 0/0/0 ms
IPsec SA: created 1/2  established 1/1  time 0/0/0 ms

  id/spi: 3 823c08a1ddd29111/9e2f8c8339d5ba43
  direction: responder
  status: established 4948-4948s ago = 0ms
  proposal: aes128-sha256
  child: no
  SK_ei: bf126e0a84efa7ff-8d6e8c56694a1850
  SK_er: 78c57cbadd9d949c-ff88391ff06083ee
  SK_ai: 451eec850591848d-10d9deae1e58df5a-741e1cadf1922d98-a81086392bcfd0e2
  SK_ar: c83ecb5fb337286a-908b54e1454e859f-8e5ad8c4fd5184c3-48ea79a494ad6bef
  PPK: no
  message-id sent/recv: 0/2
  QKD: no
  PQC-KEM (IKE): no
  PQC-KEM (all IPsec): no
  lifetime/rekey: 86400/81181
  DPD sent/recv: 00000000/00000000
  peer-id: C = CA, ST = BC, L = Burnaby, O = Fortinet, OU = QA, CN = test3
```

To verify the TLS connection in IPsec VPN tunnel real-time debugs:

```
# diagnose debug application ike 127

ike V=root:0:tofgtd: going to be deleted
ike V=root:0:tofgtd: reset TCP ports and transport mode
...
ike V=root:tcp deletes tcp-transport(vd=0, vrf=0, intf=15:15, 11.101.1.1:443->173.1.1.1:7179
sock=43 refcnt=2 ph1=0x55c2b67379e0) (1).
ike V=root:tcp destroys tcp-transport(vd=0, vrf=0, intf=15:15, 11.101.1.1:443->173.1.1.1:7179
sock=43 refcnt=0 ph1=(nil)) (0).
```

```
ike V=root:0:tofgtd: set oper down
ike V=root:0:tofgtd: schedule auto-negotiate
ike V=root:0:tofgtd: auto-negotiate connection
ike V=root:0:tofgtd:tofgtd: created connection: 0x55c2b673f5d0 15 11.101.1.1->173.1.1.1:443.
ike V=root:0:tofgtd:tofgtd: chosen to populate IKE_SA traffic-selectors
ike V=root:0:tofgtd: no suitable IKE_SA, queuing CHILD_SA request and initiating IKE_SA
negotiation
ike V=root:tls connection is inprogress tcp-transport(vd=0, vrf=0, intf=15:15, 11.101.1.1:7007-
>173.1.1.1:443 sock=41 refcnt=2 ph1=0x55c2b6730a10) (1).
ike V=root:0:tofgtd:4: generate DH public value request pending
ike V=root:tcp accepts ike tcp-transport(vd=0, vrf=0, intf=0:15, 11.101.1.1:443->173.1.1.1:4205
sock=43 refcnt=2 ph1=(nil)) (2).
ike V=root:tls connecting, ssl_want(2)
ike V=root:tls accepting.
ike V=root:tls connected, IKE tcp prefix sent
ike V=root:0:tofgtd:4: create NAT-D hash local 11.101.1.1/7007 remote 173.1.1.1/443
ike V=root:tls accepted tcp-transport(vd=0, vrf=0, intf=0:15, 11.101.1.1:443->173.1.1.1:4205
sock=43 refcnt=2 ph1=(nil)).
ike V=root:tls verified connection tcp-transport(vd=0, vrf=0, intf=0:15, 11.101.1.1:443-
>173.1.1.1:4205 sock=43 refcnt=2 ph1=(nil)).
ike V=root:0: tcp comes 173.1.1.1:443->11.101.1.1:7007,ifindex=15,vrf=0,len=264....
...
```

Post-Quantum readiness

Post-quantum readiness is the ability of a system to remain secure and trustworthy in the presence of large scale quantum computers. A strategy for post-quantum readiness should build on top of existing frameworks without requiring disruptive design changes when quantum attacks become practical.

Practically speaking, systems must ensure:

- Confidentiality is protected against Harvest Now, Decrypt Later (HNDL) attacks
- Authentication and trust do not solely depend on traditional methods that can be broken, such as RSA and ECC
- Multiple cryptographic paths can inter-operate together

Harvest Now, Decrypt Later (HNDL) attacks can capture encrypted VPN traffic today, and decrypt it later once quantum computers become viable and available for breaking current public-key cryptography. This could enable attackers to recover session secrets by breaking key exchange mechanisms and retroactively decrypt stored traffic. It may also allow the recovery of private keys used for authentication.

In IKEv2-based VPN, post-quantum readiness requires adhering to new standards around key establishments and authentication. Traditionally, IKEv2 relies on Diffie-Hellman (DH) for key exchange and RSA or ECDSA for authentication. Post-quantum readiness requires that DH be replaced or augmented with Post Quantum Cryptography (PQC) or Quantum Key Distribution (QKD). Additionally, traditional signature authentication algorithms are replaced with PQC-based signatures.

In terms of RFCs, the following enables post-quantum readiness in IKEv2:

RFC	Security Property	Effect
RFC 9370	Quantum-safe secrecy	Introduce Key Encapsulation Mechanisms (KEMs) to IKEv2, allowing post-quantum algorithms such as ML-KEM to participate directly in the IKE_SA_INIT exchange. This effectively prevents future decryption of captured traffic. This aligns with NIST FIPS 203 (ML-KEM).
RFC 9242	Hybrid Key Exchange	Migrate to defense-in-depth by allowing multiple key exchange methods to be combined in a single IKEv2 exchange. This is typically applied using classical DH and post-quantum KEM together, effectively ensuring interoperability during migration but protecting against weaknesses of a single algorithm.
RFC 8784	Out-of-band entropy	This allows IKEv2 to mix an external preshared key (PPK) into the key derivation process, adding entropy and enabling integration of offline secrets or QKD-derived keys.
RFC 7427	Quantum-safe authentication	Modernize IKEv2 signature authentication making it algorithm-agnostic rather than based on RSA or ECDSA. This effectively allows certificates with PQC public keys to be used, and post-quantum signature algorithms like ML-DSA (FIPS 204) and SLH-DSA (FIPS 205) to be used in IKE_AUTH.

FortiGate's post-quantum strategy for IPsec VPNs is a layered and modular approach based on the aforementioned standards. It ensures that both key establishment and authentication are protected against future quantum exploits. Furthermore, it preserves current IKEv2 interoperability, allowing companies to continue using IKEv2-based VPNs while adopting new emerging standards.

Administrators can consider a progressive adoption of the features in phases:

Phase	What administrators deploy	Standards anchor
Near term	IKEv2 + Hybrid PQC key exchange	RFC 9242, RFC 9370, FIPS 203
Authentication upgrade	PQC-based VPN authentication	RFC 7427, FIPS 204/205
Advanced environments	QKD-backed IPsec keys	ETSI GS QKD 014
Target state	Full hybrid (DH + PQC + QKD + PQ signatures)	Combined

For more information:

Post-Quantum Cryptography for IPsec key exchange on page 2822	Explore applying PQC with classical DH for a hybrid key exchange.

Signature authentication for VPNs using Post Quantum Cryptography	Modernize IKEv2 authentication by decoupling from RSA/ECDSA and enabling post-quantum signature algorithms for VPN authentication.
IPsec key retrieval with a QKD system on page 2829	Retrieve keying material from an external QKD system, bypassing computational key exchange entirely for IPsec SAs.
Applying hybrid QKD and PQC in IPsec key establishment on page 2834	Apply both PQC in hybrid key exchange and QKD together.

Post-Quantum Cryptography for IPsec key exchange

IPsec key exchange supports Post-Quantum Cryptography (PQC) to enhance security with algorithms that protect against quantum computer attacks. This update ensures future-proof encryption and addresses vulnerabilities in traditional methods, aligning with upcoming security standards.

PQC in IPsec applies Hybrid Key Exchange per RFC 9370, which extends IKEv2 to perform multiple key exchanges in a single IKE SA establishment. It uses the IKE_INTERMEDIATE exchange from RFC 9242 to carry extra Key Exchanges (KEs) during initial SA setup. And it uses IKE_FOLLOWUP_KE for additional key exchanges when rekeying the IKE SA or creating additional child SAs in IKEv2 Phase 2.

In addition to the classic DH key exchange, one or more post quantum Key Exchange Mechanisms (KEMs), such as ML KEM (Kyber), BIKE, HQC and FRODO are supported. When configuring Phase1 VPN settings, choose up to three KE groups in each round of additional key exchanges. Choose up to seven rounds of additional key exchanges (for example, ADDKE1, ADDKE2, and so on).

While FortiOS allows users to specify various KE groups and mechanisms, only the following KE groups are standardized by NIST and are FIPS 203 compliant:

- ML-KEM-512
- ML-KEM-768
- ML-KEM-1024

FIPS 203, also known as the Module-Lattice-Based Key-Encapsulation Mechanism (ML-KEM) Standard, is a set of guidelines established by the National Institute of Standards and Technology (NIST). These guidelines specify the use of lattice-based cryptographic algorithms for key encapsulation mechanisms, which are crucial for secure communication in various applications.

The three parameter sets offer different levels of security and performance:

ML-KEM-512	Provides a balance between security and efficiency, suitable for environments where moderate security is sufficient.
ML-KEM-768	Offers a higher level of security compared to ML-KEM-512, making it suitable for more sensitive applications.
ML-KEM-1024	Delivers the highest level of security among the three, ideal for highly sensitive data and critical applications.

See FIPS203 and Module-Lattice-Based Key-Encapsulation Mechanism Standard for more information.

CLI configuration

The following commands can be used to enable and configure PQC:

```
config vpn ipsec phase1-interface
    edit <name>
        set addke1 <option1>, <option2>, <option3>
        set addke2 <option1>, <option2>, <option3>
        set addke3 <option1>, <option2>, <option3>
        set addke4 <option1>, <option2>, <option3>
        set addke5 <option1>, <option2>, <option3>
        set addke6 <option1>, <option2>, <option3>
        set addke7 <option1>, <option2>, <option3>
        set childless-ike enable
    next
end
config vpn ipsec phase2-interface
    edit <name>
        set addke1 <option1>, <option2>, <option3>
        set addke2 <option1>, <option2>, <option3>
        set addke3 <option1>, <option2>, <option3>
        set addke4 <option1>, <option2>, <option3>
        set addke5 <option1>, <option2>, <option3>
        set addke6 <option1>, <option2>, <option3>
        set addke7 <option1>, <option2>, <option3>
    next
end
```

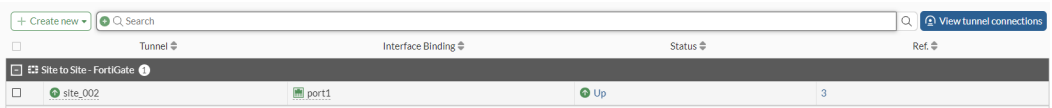
Example

A financial institution uses IPsec VPN to move sensitive customer data, such as account numbers, social insurance numbers, and credit card information. The current encryption used is based on traditional algorithms, which could be vulnerable to attacks from quantum computers in the future. By implementing Post-Quantum Cryptography, the financial institution can ensure that their data remains secure even as technology advances, protecting themselves and their customers from potential breaches due to advancements in computing power. This ensures compliance with regulatory requirements and maintains customer trust.

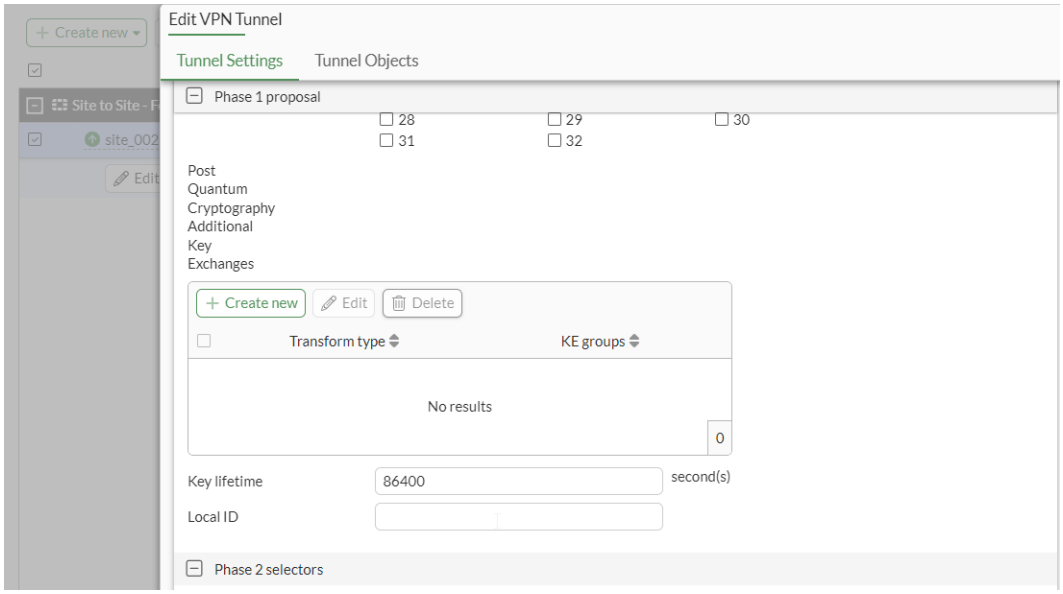
To enable PQC in the GUI:

This is a site-to-site VPN setup. Only the new configuration is being demonstrated in the GUI for this example. For more information, see Basic site-to-site VPN with pre-shared key on page 2857.

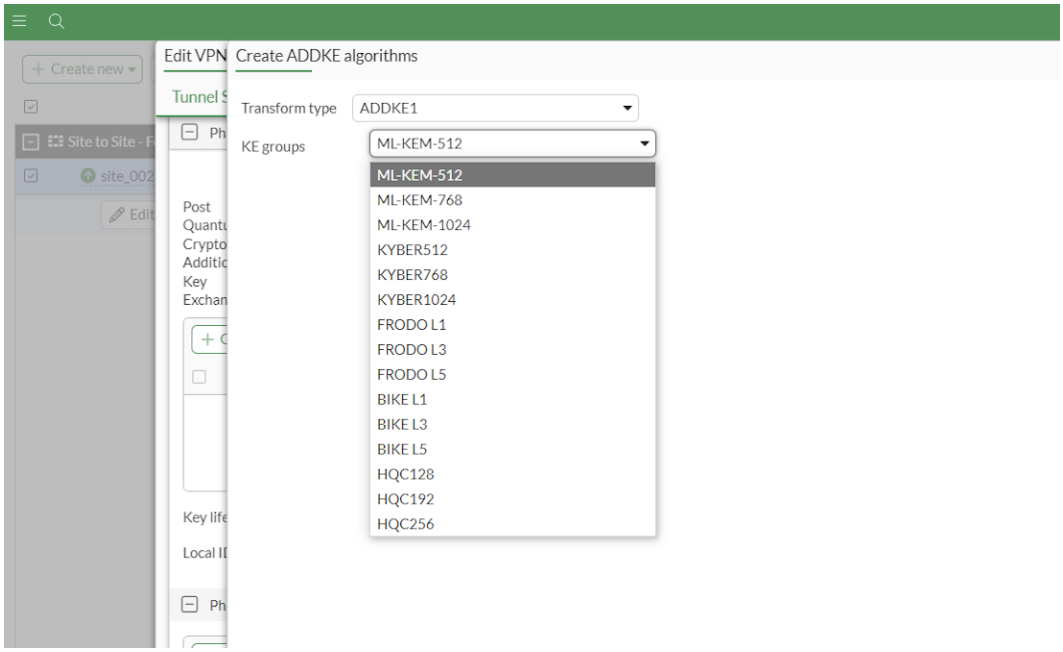
1. Go to *VPN* > *VPN Tunnels*.



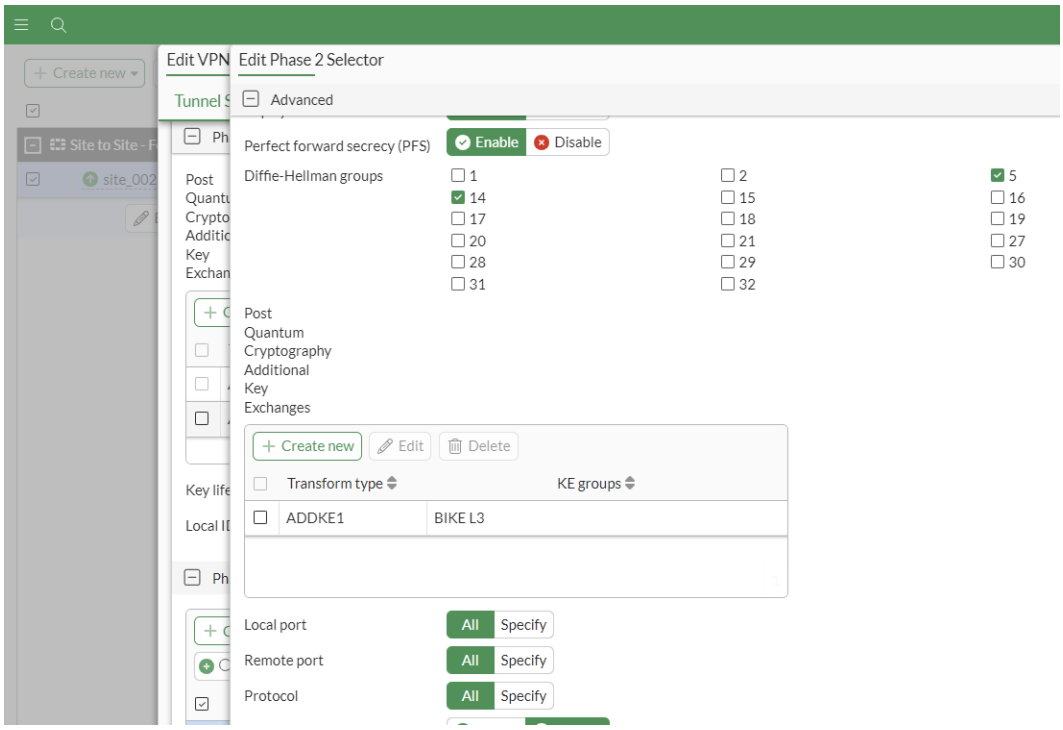
2. Double-click the VPN Tunnel to open it for editing.
3. Scroll down to *Post Quantum Cryptography Additional Key Exchanges*, and click *Create new*.



4. Set *Transform type*. Each transform type represents an additional round of KE exchange in addition to the DH group Key Exchange. A total of seven transform types can be created.
5. For each *Transform type*, select up to three *KE groups*, and click *OK*.



6. In *Phase 2 selectors*, click *Create new* and repeat the steps above.



7. Click *OK* to save the tunnel.

To enable PQC key exchange for an IPsec tunnel in the CLI:

1. Configure FGT-C:

```
config vpn ipsec phase1-interface
    edit "site_002"
        set interface "port1"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set addke1 35 36 37
        set addke2 1083
        set childless-ike enable
        set transport auto
        set remote-gw 172.16.200.9
        set psksecret XXXXXX
    next
end
config vpn ipsec phase2-interface
    edit "site_002"
        set phase1name "site_002"
        set addke1 1090
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
```

```
   next
end
```

2. Configure FGT-D:

```
config vpn ipsec phase1-interface
    edit "site_001"
        set interface "port1"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set addke1 35 36 37
        set addke2 1083
        set childless-ike enable
        set transport auto
        set remote-gw 172.16.200.8
        set psksecret XXXXXX
    next
end
config vpn ipsec phase2-interface
    edit "site_001"
        set phase1name "site_001"
        set addke1 1090
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
    next
end
```

3. Verify the IPsec VPN tunnel state on FGT-D:

```
#diagnose vpn ike gateway list

vd: root/0
name: site_001PPPPPP1
version: 2
interface: port1 9
addr: 172.16.200.9:500 -> 172.16.200.8:500
tun_id: 172.16.200.8/::172.16.200.8
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 88s ago
peer-id: 172.16.200.8
peer-id-auth: no
pending-queue: 0
PPK: no
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 111 7316ebc639a46c4b/4be2fd15d7304333
```

```
  direction: responder
  status: established 88-88s ago = 0ms
  proposal: aes128-sha256
  child: no
  SK_ei: 5a81d7880641e298-3c6cd9753e62c482
  SK_er: b52ad6f1590ca132-86245f3df596eb32
  SK_ai: 00d051d432556218-5708374c8496c37e-aef06d7d0457bebd-506bd9034bdd1c30
  SK_ar: 15dac6f288163198-2c34b87dda107af7-6db98a53b29cf757-0d6d5b8020cc8d47
  PPK: no
  message-id sent/recv: 0/6
  QKD: no
  PQC-KEM (IKE): yes
  PQC-KEM (all IPsec): yes
  lifetime/rekey: 86400/86041
  DPD sent/recv: 00000000/00000000
  peer-id: 172.16.200.8
```

4. Verify the IKE debug output:

```
#diagnose debug enable
#diagnose debug application ike -1

ike V=root:0:site_002PPPPPP1:14: initiator received SA_INIT response
ike V=root:0:site_002PPPPPP1:14: processing notify type NAT_DETECTION_SOURCE_IP
ike V=root:0:site_002PPPPPP1:14: processing NAT-D payload
ike V=root:0:site_002PPPPPP1:14: NAT not detected
ike V=root:0:site_002PPPPPP1:14: process NAT-D
ike V=root:0:site_002PPPPPP1:14: processing notify type NAT_DETECTION_DESTINATION_IP
ike V=root:0:site_002PPPPPP1:14: processing NAT-D payload
ike V=root:0:site_002PPPPPP1:14: NAT not detected
ike V=root:0:site_002PPPPPP1:14: process NAT-D
ike V=root:0:site_002PPPPPP1:14: processing notify type FRAGMENTATION_SUPPORTED
ike V=root:0:site_002PPPPPP1:14: processing notify type CHILDLESS_IKEV2_SUPPORTED
ike V=root:0:site_002PPPPPP1:14: processing notify type INTERMEDIATE_EXCHANGE_SUPPORTED
ike V=root:0:site_002PPPPPP1:14: incoming proposal:
ike V=root:0:site_002PPPPPP1:14: proposal id = 1:
ike V=root:0:site_002PPPPPP1:14:   protocol = IKEv2:
ike V=root:0:site_002PPPPPP1:14:      encapsulation = IKEv2/none
ike V=root:0:site_002PPPPPP1:14:         type=ENCR, val=AES_CBC (key_len = 128)
ike V=root:0:site_002PPPPPP1:14:         type=INTEGR, val=AUTH_HMAC_SHA2_256_128
ike V=root:0:site_002PPPPPP1:14:         type=PRF, val=PRF_HMAC_SHA2_256
ike V=root:0:site_002PPPPPP1:14:         type=DH_GROUP, val=MODP2048.
ike V=root:0:site_002PPPPPP1:14:         type=ADDKE1, val=ML-KEM-512.
ike V=root:0:site_002PPPPPP1:14:         type=ADDKE2, val=FRODO L1.
ike V=root:0:site_002PPPPPP1:14: matched proposal id 1
ike V=root:0:site_002PPPPPP1:14: proposal id = 1:
ike V=root:0:site_002PPPPPP1:14:   protocol = IKEv2:
ike V=root:0:site_002PPPPPP1:14:      encapsulation = IKEv2/none
ike V=root:0:site_002PPPPPP1:14:         type=ENCR, val=AES_CBC (key_len = 128)
ike V=root:0:site_002PPPPPP1:14:         type=INTEGR, val=AUTH_HMAC_SHA2_256_128
ike V=root:0:site_002PPPPPP1:14:         type=PRF, val=PRF_HMAC_SHA2_256
ike V=root:0:site_002PPPPPP1:14:         type=DH_GROUP, val=MODP2048.
ike V=root:0:site_002PPPPPP1:14:         type=ADDKE1, val=ML-KEM-512.
```

```
ike V=root:0:site_002PPPPPP1:14:         type=ADDKE2, val=FRODO L1.
ike V=root:0:site_002PPPPPP1:14: lifetime=86400
ike V=root:0:site_002PPPPPP1:14: compute DH shared secret request pending
ike 0:site_002PPPPPP1:14: IKE SA 1d7b30eb8916377b/bcc3ac86051fd55c SK_ei
16:636BFFEC63FB7B02AC3C55CBC2A615ED
ike 0:site_002PPPPPP1:14: IKE SA 1d7b30eb8916377b/bcc3ac86051fd55c SK_er
16:EBA51E65F2E101A6922C728D8507B1D6
ike 0:site_002PPPPPP1:14: IKE SA 1d7b30eb8916377b/bcc3ac86051fd55c SK_ai
32:146F21B8D780175B7CF3C13475754B5296D11E0658695F77E7688C33681AA8A6
ike 0:site_002PPPPPP1:14: IKE SA 1d7b30eb8916377b/bcc3ac86051fd55c SK_ar
32:ADA3A9275B35CD2309B73FCE7A17AD0A75DC5F98A4D4FF008D7503CD6A69EA94
ike V=root:0:site_002PPPPPP1:14: initiator preparing INTERMEDIATE msg
ike V=root:0:site_002PPPPPP1:14: generating KE for group 35
ike 0:site_002PPPPPP1:14: enc

ike V=root:0:site_002PPPPPP1:14: initiator received INTEREMDIATE response
ike V=root:0:site_002PPPPPP1:14: processing KE group 35
ike V=root:0:site_002PPPPPP1:14: KEM decapsulate okay
ike 0:site_002PPPPPP1:14: IKE SA 1d7b30eb8916377b/bcc3ac86051fd55c SK_ei
16:7D54A38F71768F91B32EA828521E56D7
ike 0:site_002PPPPPP1:14: IKE SA 1d7b30eb8916377b/bcc3ac86051fd55c SK_er
16:AC66EF35E670D558A5704519E348BFD2
ike 0:site_002PPPPPP1:14: IKE SA 1d7b30eb8916377b/bcc3ac86051fd55c SK_ai
32:39DED1A07EF0D0A37D2FA51AF478EC981E17D4092FA1DE3CF3A46EC8E28DD9E9
ike 0:site_002PPPPPP1:14: IKE SA 1d7b30eb8916377b/bcc3ac86051fd55c SK_ar
32:3E8BB857F90943DD289DDCDE742AC18DFBF0DA7590F9CEA5E413B02463BE4303
ike V=root:0:site_002PPPPPP1:14: initiator preparing INTERMEDIATE msg
ike V=root:0:site_002PPPPPP1:14: generating KE for group 1083
ike V=root:0:site_002PPPPPP1:14: splitting payload len=9624 into 10 fragments

ike V=root:0:site_002PPPPPP1:14:site_002:17: phase2 matched by subset
ike V=root:0:site_002PPPPPP1:14:site_002:17: accepted proposal:
ike V=root:0:site_002PPPPPP1:14:site_002:17: TSr_0 0:10.1.100.0-10.1.100.255:0
ike V=root:0:site_002PPPPPP1:14:site_002:17: TSi_0 0:10.1.200.0-10.1.200.255:0
ike V=root:0:site_002PPPPPP1:14:site_002:17: autokey
ike V=root:0:site_002PPPPPP1:14:site_002:17: incoming child SA proposal:
ike V=root:0:site_002PPPPPP1:14:site_002:17: proposal id = 1:
ike V=root:0:site_002PPPPPP1:14:site_002:17:   protocol = ESP:
ike V=root:0:site_002PPPPPP1:14:site_002:17:      encapsulation = TUNNEL
ike V=root:0:site_002PPPPPP1:14:site_002:17:         type=ENCR, val=AES_CBC (key_len = 128)
ike V=root:0:site_002PPPPPP1:14:site_002:17:         type=INTEGR, val=SHA
ike V=root:0:site_002PPPPPP1:14:site_002:17:         type=DH_GROUP, val=MODP2048
ike V=root:0:site_002PPPPPP1:14:site_002:17:         type=ESN, val=NO
ike V=root:0:site_002PPPPPP1:14:site_002:17:         type=ADDKE1, val=BIKE L3.
ike V=root:0:site_002PPPPPP1:14:site_002:17:         type=ADDKE1, val=BIKE L5.
ike V=root:0:site_002PPPPPP1:14:site_002:17:         type=ADDKE1, val=HQC128.
ike V=root:0:site_002PPPPPP1:14:site_002:17:         type=ADDKE1, val=HQC192.
ike V=root:0:site_002PPPPPP1:14:site_002:17:         type=ADDKE1, val=HQC256.
ike V=root:0:site_002PPPPPP1:14:site_002:17:         type=ADDKE1, val=1090.
ike V=root:0:site_002PPPPPP1:14:site_002:17: matched proposal id 1
ike V=root:0:site_002PPPPPP1:14:site_002:17: proposal id = 1:
ike V=root:0:site_002PPPPPP1:14:site_002:17:   protocol = ESP:
```

```
ike V=root:0:site_002PPPPPP1:14:site_002:17:      encapsulation = TUNNEL
ike V=root:0:site_002PPPPPP1:14:site_002:17:         type=ENCR, val=AES_CBC (key_len = 128)
ike V=root:0:site_002PPPPPP1:14:site_002:17:         type=INTEGR, val=SHA
ike V=root:0:site_002PPPPPP1:14:site_002:17:         type=DH_GROUP, val=MODP2048
ike V=root:0:site_002PPPPPP1:14:site_002:17:         type=ESN, val=NO
ike V=root:0:site_002PPPPPP1:14:site_002:17:         type=ADDKE1, val=BIKE L3.
ike V=root:0:site_002PPPPPP1:14:site_002:17:         type=ADDKE1, val=BIKE L5.
ike V=root:0:site_002PPPPPP1:14:site_002:17:         type=ADDKE1, val=HQC128.
ike V=root:0:site_002PPPPPP1:14:site_002:17:         type=ADDKE1, val=HQC192.
ike V=root:0:site_002PPPPPP1:14:site_002:17:         type=ADDKE1, val=HQC256.
ike V=root:0:site_002PPPPPP1:14:site_002:17:         type=ADDKE1, val=1090.
ike V=root:0:site_002PPPPPP1:14:site_002:17: lifetime=43200
ike V=root:0:site_002PPPPPP1:14:site_002:17: ADDKE negotiated
ike V=root:0:site_002PPPPPP1:17: initiator preparing FOLLOWUP_KE message (CHILD_SA)
ike V=root:0:site_002PPPPPP1:17: construct KE grp (1090) payload
ike V=root:0:site_002PPPPPP1:14: splitting payload len=3103 into 3 fragments
```

IPsec key retrieval with a QKD system

FortiGates support IPsec key retrieval with a quantum key distribution (QKD) system using the ETSI standardized API. A QKD server, which is also referred to as Key Management Entity (KME) or Key Management Server (KMS) is responsible for delivering quantum-generated cryptographic keys for FortiGate to use in IKE SAs.

This eliminates the Key Exchange and negotiation process by delivering keys directly for use for the IPSec SA (Security Association). This has the advantage of being quantum-safe, no longer depending on the DH key exchange, higher entropy and more resilience in case IKE negotiations is compromised. This also enhances efficiency in IPsec key management.

```
config vpn qkd
    edit <name>
        set server <string>
        set port <integer>
        set id <string>
        set peer <string>
        set certificate <certificate_name>
    next
end
```

`server <string>`	Enter the IPv4, IPv6, or DNS address of the key management entity (KME).
`port <integer>`	Enter the port to connect to on the KME, 1 - 65535.
`id <string>`	Enter the quantum key distribution ID assigned by the KME.
`peer <string>`	Enter the peer or peer group to authenticate with the quantum key device's certificate.
`certificate <certificate_name>`	Enter the name of up to four certificates to offer to the KME.

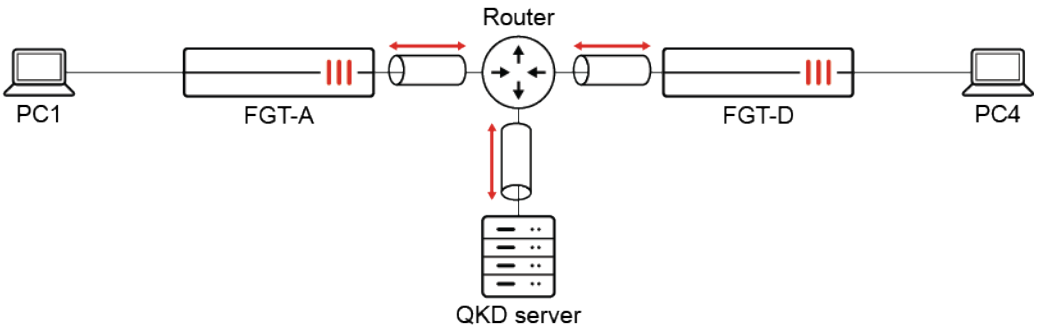
```
config ipsec phase1-interface
    edit <name>
        set qkd {disable | allow | require}
        set qkd-profile <profile>
    next
end
```

`qkd {disable | allow | require}`	• `disable`: Do not use QKD server. • `allow`: Allowed but does not require the use of QKD server. • `require`: Requires the use of QKD server.
`qkd-profile <profile>`	Name of the profile created from `config vpn qkd`.

When QKD is enabled, PFS (Perfect Forward Secrecy) is disabled, and DH Group cannot be configured in Phase 2.

Example

In this example, a quantum key distribution (QKD) system is deployed to perform central IPsec key management. The FortiGates installed as security gateways will terminate large amount of IPsec tunnels.



To configure IPsec key retrieval with a QKD system:

1. Configure FGT-A:
   a. Configure the QKD profile:

```
config vpn qkd
    edit "qkd_1"
        set server "172.16.200.83"
        set port 8989
        set id "FGT-A"
        set peer "qkd"
        set certificate "FGT_qkd1"
    next
end
```

   b. Configure the IPsec phase 1 interface settings:

```
config vpn ipsec phase1-interface
    edit "site1"
        set interface "wan1"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set qkd allow
        set qkd-profile "qkd_1"
        set remote-gw 173.1.1.1
        set psksecret **********
    next
end
```

c. Configure the IPsec phase 2 interface settings:

```
config vpn ipsec phase2-interface
    edit "site1"
        set phase1name "site1"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
    next
end
```

2. Configure FGT-D:

a. Configure the QKD profile:

```
config vpn qkd
    edit "qkd_1"
        set server "172.16.200.83"
        set port 8989
        set id "FGT-D"
        set peer "qkd"
        set certificate "FGT_qkd3"
    next
end
```

b. Configure the IPsec phase 1 interface settings:

```
config vpn ipsec phase1-interface
    edit "site2"
        set interface "port25"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set qkd require
        set qkd-profile "qkd_1"
        set remote-gw 11.101.1.1
        set psksecret **********
    next
end
```

c. Configure the IPsec phase 2 interface settings:

```
config vpn ipsec phase2-interface
    edit "site2"
        set phase1name "site2"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
    next
end
```

To verify the configuration:

1. Generate traffic between PC1 and PC4.
2. Run diagnostics on FGT-A:
   a. Verify the IPsec phase 1 interface status:

```
# diagnose vpn ike gateway list

vd: root/0
name: site1
version: 1
interface: wan1 17
addr: 11.101.1.1:500 -> 173.1.1.1:500
tun_id: 172.16.200.4/::172.16.200.4
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 3s ago
peer-id: 173.1.1.1
peer-id-auth: no
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/1  established 1/1  time 30/30/30 ms

  id/spi: 21 ad7d995677250c7e/053f958ea7be66c8
  direction: initiator
  status: established 3-3s ago = 0ms
  proposal: aes128-sha256
  key: 5b198e1a431c20fb-c08135cf0c007704
  QKD: yes
  lifetime/rekey: 86400/86096
  DPD sent/recv: 00000000/00000000
  peer-id: 173.1.1.1
```

   b. Verify the IPsec phase 2 tunnel status:

```
# diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=site1 ver=1 serial=2 11.101.1.1:0->173.1.1.1:0 tun_id=172.16.200.4 tun_
id6=::172.16.200.4 dst_mtu=1500 dpd-link=on weight=1
bound_if=17 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc
run_state=0 role=primary accept_traffic=1 overlay_id=0
```

```
proxyid_num=1 child_num=0 refcnt=4 ilast=12 olast=11 ad=/0
stat: rxp=1 txp=2 rxb=84 txb=168
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=site1 proto=0 sa=1 ref=3 serial=2
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=6 options=10226 type=00 soft=0 mtu=1438 expire=42883/0B replaywin=2048
       seqno=3 esn=0 replaywin_lastseq=00000002 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42897/43200
  dec: spi=b2af532f esp=aes key=16 c1d5d17e6bdecd5b145f672a5054cde1
       ah=sha1 key=20 084f1c0fee48994f59a125606f9c757838dc2421
  enc: spi=3d14392a esp=aes key=16 66277c8cf2bdbd2d12a9d829dde356ad
       ah=sha1 key=20 fdbaa42cca5c3a9bffb1cf0fc74ff29a643a2b9f
  dec:pkts/bytes=1/84, enc:pkts/bytes=2/304
  npu_flag=03 npu_rgwy=173.1.1.1 npu_lgwy=11.101.1.1 npu_selid=4 dec_npuid=2 enc_npuid=2
```

The IPsec tunnel is up and traffic passes through.

c. Verify the IKE debug messages:

```
# diagnose debug application ike -1
...
ike V=root:0:site1:site1: IPsec SA connect 17 11.101.1.1->173.1.1.1:0
ike V=root:0:site1:site1: using existing connection
ike V=root:0:site1:site1: config found
ike V=root:0:site1:site1: IPsec SA connect 17 11.101.1.1->173.1.1.1:500 negotiating
ike 0:site1:20:site1:22: QKD initiator request
ike 0:site1:20:site1:22: QKD initiator key-id '4e0592fe-9568-11ee-97b8-5fb93000b0c2'
...
ike V=root:0:site1:20:site1:22: add IPsec SA: SPIs=b2af532d/3d143928
ike 0:site1:20:site1:22: IPsec SA dec spi b2af532d key 16:958EE561ABD2B6F0F4C6E042202F451E
auth 20:4D694E6951ADB425A2A1C3261140957C9469A4DC
ike 0:site1:20:site1:22: IPsec SA enc spi 3d143928 key 16:6016E26398B70E55A17EF73611B30028
auth 20:357880E885F3ED23092233737B9FD0573DCB0D08
ike V=root:0:site1:20:site1:22: added IPsec SA: SPIs=b2af532d/3d143928
ike V=root:0:site1:20:site1:22: sending SNMP tunnel UP trap
```

d. Verify the statistics for qkd_1:

```
# diagnose vpn ike qkd qkd_1
client.count.fd: now 0  max 1  total 3
client.count.fp: now 0  max 1  total 3
client.count.mmap: now 2  max 2  total 9
client.event: 4
client.retry: 0
client.cmd.request.initiator: 4
client.cmd.request.responder: 0
client.cmd.reply.initiator: 4
client.cmd.reply.responder: 0
server.boot.count: 3
server.boot.last.time: 4295388395
```

```
server.boot.last.ago: 247
server.stop.budget: 0
server.stop.error: 0
server.stop.auth.count: 0
server.cmd.reading: 7
server.cmd.read: 4
server.cmd.request.initiator: 4
server.cmd.request.responder: 0
server.cmd.reply.initiator: 4
server.cmd.reply.responder: 0
server.auth.request.sending.count: 4
server.auth.request.sending.last.time: 4295389413
server.auth.request.sending.last.ago: 237
server.auth.request.sent.count: 4
server.auth.request.sent.last.time: 4295389413
server.auth.request.sent.last.ago: 237
server.auth.reply.reading.count: 4
server.auth.reply.reading.last.time: 4295389413
server.auth.reply.reading.last.ago: 237
server.auth.reply.read.count: 4
server.auth.reply.read.last.time: 4295389413
server.auth.reply.read.last.ago: 237
server.dns.addrs:
server.curl.get.count: 4
server.curl.get.last.time: 4295389413
server.curl.get.last.ago: 237
server.curl.json.parse: 4
server.curl.json.parsed: 4
```

Applying hybrid QKD and PQC in IPsec key establishment

Independently, Quantum Key Distribution (IPsec key retrieval with a QKD system) and Post-Quantum Cryptography (Post-Quantum Cryptography for IPsec key exchange) can be used to secure the IPsec tunnel against future quantum threats. Together, FortiGate also supports the hybrid approach of applying QKD and PQC and Diffie-Hellman Exchange (DHE) together.

This allows you to mix keys from QKD, PQC, and traditional Diffie-Hellman (DH) key exchange in establishing the IKE SA and IPsec SA, ensuring robust security and defense-in-depth. By combining different types of keys, users can achieve maximum resilience against potential threats.

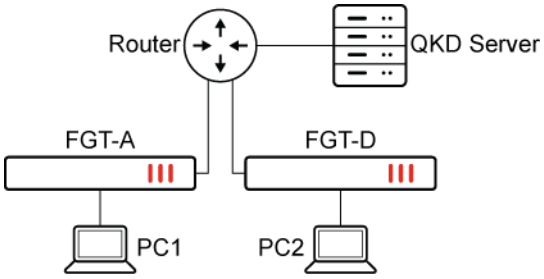
This feature can be used in environments handling highly sensitive data, necessitating the highest level of security. For example, a financial institution might use this feature in its network infrastructure to secure communications between different system components. This ensures that even if one key exchange method is compromised, the other methods will still provide secure communication.

```
config ipsec phase1-interface
    edit <name>
        set qkd {disable | allow | require}
        set qkd-profile <profile>
        set qkd-hybrid {disable | allow | require}
```

```
    next
end
```

`qkd-hybrid {disable | allow | require}`	• `disable`: Do not use QKD hybrid keys. • `allow`: Allowed but do not require the use of QKD hybrid keys. • `require`: Require the use of QKD hybrid keys.

Example



To configure IPsec key establishment with a QKD and PQC keys together by CLI:

1. Configure FGT-A
   a. Configure the QKD profile:

```
config vpn qkd
    edit "qkdtest"
        set server "10.1.100.9"
        set port 8989
        set id "123456"
        set peer "qkdtest"
    next
end
```

   b. Configure the IPsec phase1 interface:

```
config vpn ipsec phase1-interface
    edit "test_qkd"
        set interface "port9"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set dhgrp 21
        set addke1 1080
        set qkd require
        set qkd-hybrid require
        set qkd-profile "qkdtest"
        set remote-gw 173.1.1.1
        set psksecret **********
```

```
   next
end
```

c. Configure the IPsec phase2 interface :

```
config vpn ipsec phase2-interface
    edit "test_qkd"
        set phase1name "test_qkd"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set dhgrp 20 21
        set addke1 35
    next
end
```

2. Configure FGT-D

a. Configure the QKD profile:

```
config vpn qkd
    edit "qkdtest"
        set server "10.1.100.9"
        set port 8989
        set id "123456"
        set peer "qkdtest"
    next
end
```

b. Configure the IPsec phase1 interface:

```
config vpn ipsec phase1-interface
    edit "test_qkd"
        set interface "test_qkd"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set dhgrp 21
        set addke1 1080
        set qkd require
        set qkd-hybrid require
        set qkd-profile "qkdtest"
        set remote-gw 11.101.1.1
        set psksecret **********
    next
end
```

c. Configure the IPsec phase2 interface :

```
config vpn ipsec phase2-interface
    edit "test_qkd"
        set phase1name "test_qkd"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
```

```
aes256gcm chacha20poly1305
        set dhgrp 20 21
        set addke1 35
    next
end
```

To verify the configuration on FGT-A:

1. Generate traffic between PC1 and PC4.
2. Verify the IPsec phase1 interface status:

```
# diagnose vpn ike gateway list
vd: root/0
name: test_qkd
version: 2
interface: port9 15
addr: 11.101.1.1:500 -> 173.1.1.1:500
tun_id: 173.1.1.1/::173.1.1.1
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 1557s ago
peer-id: 173.1.1.1
peer-id-auth: no
pending-queue: 0
PPK: no
IKE SA: created 1/18  established 1/18  time 0/2/10 ms
IPsec SA: created 1/33  established 1/18  time 0/16/20 ms

  id/spi: 67 bc882e536cbc7f1d/ede854f1e0dc71bb
  direction: responder
  status: established 17-17s ago = 0ms
  proposal: aes128-sha256
  child: yes
  SK_ei: 4e415c84e086e980-059f1be89239ec30
  SK_er: fb14db5b3718dad4-e7f5158308ba00c0
  SK_ai: 8894937bd4e66304-a5c64941dc08d544-c7d725408247cfc4-489a292a3fb44b51
  SK_ar: 0bbf85dcd9daaa1b-1f1e04318b69aaae-befc871e40f9ab4c-0d005f0f980a3d60
  message-id sent/recv: 0/1
  QKD: yes
  PQC-KEM (IKE): yes
  PQC-KEM (all IPsec): yes
  lifetime/rekey: 86400/86112
  DPD sent/recv: 00000000/00000000
  peer-id: 173.1.1.1
```

3. Verify the IPsec phase2 interface status:

```
# diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=test_qkd ver=2 serial=1 11.101.1.1:0->173.1.1.1:0 nexthop=11.101.1.2 tun_id=173.1.1.1
```

```
tun_id6=::173.1.1.1 status=up dst_mtu=1500 weight=1
bound_if=15 real_if=15 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu
frag-rfc  run_state=0 role=primary accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=3 ilast=1270 olast=1270 ad=/0
stat: rxp=69 txp=70 rxb=10420 txb=5880
dpd: mode=off
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=test_qkd proto=0 sa=1 ref=2 serial=2 auto-negotiate
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=6 options=18227 type=00 soft=0 mtu=1438 expire=3297/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=3329/3600
  dec: spi=450287b6 esp=aes key=16 42551cf97c77fb110f44574e0b33b36a
       ah=sha1 key=20 640b627fb80f4342a488349bdd35437db381640c
  enc: spi=a8a6afc3 esp=aes key=16 238c43ac7ee569c20ba5861fb8336c9a
       ah=sha1 key=20 d1226a16bfecb186148aafbce42a5da34e246afb
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=03 npu_rgwy=173.1.1.1 npu_lgwy=11.101.1.1 npu_selid=1 dec_npuid=1 enc_npuid=1
```

4. Verify IKE debug messages on:

```
# diagnose debug application ike -1
...
ike V=root:0:64555dc0381b820e/0000000000000000:70: received notify type USE_QKD
ike V=root:0:64555dc0381b820e/0000000000000000:70: received notify type INTERMEDIATE_EXCHANGE_
SUPPORTED
ike V=root:0:64555dc0381b820e/0000000000000000:70: incoming proposal:
ike V=root:0:64555dc0381b820e/0000000000000000:70: proposal id = 1:
ike V=root:0:64555dc0381b820e/0000000000000000:70:   protocol = IKEv2:
ike V=root:0:64555dc0381b820e/0000000000000000:70:      encapsulation = IKEv2/none
ike V=root:0:64555dc0381b820e/0000000000000000:70:         type=ENCR, val=AES_CBC (key_len =
128)
ike V=root:0:64555dc0381b820e/0000000000000000:70:         type=INTEGR, val=AUTH_HMAC_SHA2_
256_128
ike V=root:0:64555dc0381b820e/0000000000000000:70:         type=PRF, val=PRF_HMAC_SHA2_256
ike V=root:0:64555dc0381b820e/0000000000000000:70:         type=DH_GROUP, val=ECP521.
ike V=root:0:64555dc0381b820e/0000000000000000:70:         type=ADDKE1, val=KYBER512.
...
ike V=root:0:test_qkd:70: process NAT-D
ike V=root:0:test_qkd:70: processing notify type FRAGMENTATION_SUPPORTED
ike V=root:0:test_qkd:70: processing notify type USE_QKD
ike V=root:0:test_qkd:70: responder preparing SA_INIT msg
...
ike V=root:0:test_qkd:70: responder received INTEREMDIATE msg
ike V=root:0:test_qkd:70: processing KE group 1080
ike V=root:0:test_qkd:70: KEM encapsulate okay
ike V=root:0:test_qkd:70: responder INTERMEDIATE send
ike V=root:0:test_qkd:70: construct KE (1080) payload
...
```

```
ike V=root:0:test_qkd:73: received followup-ke request
ike V=root:0:test_qkd:73: responder received FOLLOWUP_KE msg
ike V=root:0:test_qkd:73: processing notify type ADDITIONAL_KEY_EXCHANGE
ike V=root:0:test_qkd:73: received ADDKE notify link data size (len=4)
ike V=root:0:test_qkd:73: FOLLOWUP_KE continuation for IKE SA
1d5770c3f3044565/b2cb648b2f8e064d
ike V=root:0:test_qkd:74: KEM encapsulate okay
ike V=root:0:test_qkd:74: responder preparing IKE SA FOLLOWUP_KE message
ike V=root:0:test_qkd:74: construct KE grp (1080) payload
...
ike V=root:0:test_qkd:74::80: QKD responder request '49d2bff4-0e61-11f0-8e8b-6bdf091987ab'
ike V=root:0:test_qkd:74::80: QKD responder key-id '49d2bff4-0e61-11f0-8e8b-6bdf091987ab'
ike V=root:0:test_qkd:74:80: peer proposal:
ike V=root:0:test_qkd:74:80: TSi_0 0:0.0.0.0-255.255.255.255:0
ike V=root:0:test_qkd:74:80: TSr_0 0:0.0.0.0-255.255.255.255:0
ike V=root:0:test_qkd:74:test_qkd:80: comparing selectors
ike V=root:0:test_qkd:74:test_qkd:80: matched by rfc-rule-2
ike V=root:0:test_qkd:74:test_qkd:80: phase2 matched by subset
ike V=root:0:test_qkd:74:test_qkd:80: accepted proposal:
ike V=root:0:test_qkd:74:test_qkd:80: TSi_0 0:0.0.0.0-255.255.255.255:0
ike V=root:0:test_qkd:74:test_qkd:80: TSr_0 0:0.0.0.0-255.255.255.255:0
ike V=root:0:test_qkd:74:test_qkd:80: autokey
ike V=root:0:test_qkd:74:test_qkd:80: incoming child SA proposal:
ike V=root:0:test_qkd:74:test_qkd:80: proposal id = 1:
ike V=root:0:test_qkd:74:test_qkd:80:   protocol = ESP:
ike V=root:0:test_qkd:74:test_qkd:80:      encapsulation = TUNNEL
ike V=root:0:test_qkd:74:test_qkd:80:         type=ENCR, val=AES_CBC (key_len = 128)
ike V=root:0:test_qkd:74:test_qkd:80:         type=INTEGR, val=SHA
ike V=root:0:test_qkd:74:test_qkd:80:         type=DH_GROUP, val=ECP384
ike V=root:0:test_qkd:74:test_qkd:80:         type=DH_GROUP, val=ECP521
ike V=root:0:test_qkd:74:test_qkd:80:         type=ESN, val=NO
ike V=root:0:test_qkd:74:test_qkd:80:         type=ADDKE1, val=ML-KEM-512.
...
ike V=root:0:test_qkd:74:test_qkd:80: ADDKE negotiated
...
ike V=root:0:test_qkd:74: responder received FOLLOWUP_KE msg
ike V=root:0:test_qkd:74: processing notify type ADDITIONAL_KEY_EXCHANGE
ike V=root:0:test_qkd:74: received ADDKE notify link data size (len=4)
ike V=root:0:test_qkd:74: FOLLOWUP_KE continuation for IPsec SA bc870245/c9afa6a8
ike V=root:0:test_qkd:80: KEM encapsulate okay
ike V=root:0:test_qkd:80: responder preparing CHILD SA FOLLOWUP_KE message
ike V=root:0:test_qkd:80: construct KE grp (35) payload
```

5. Verify the statistics of QKD profile:

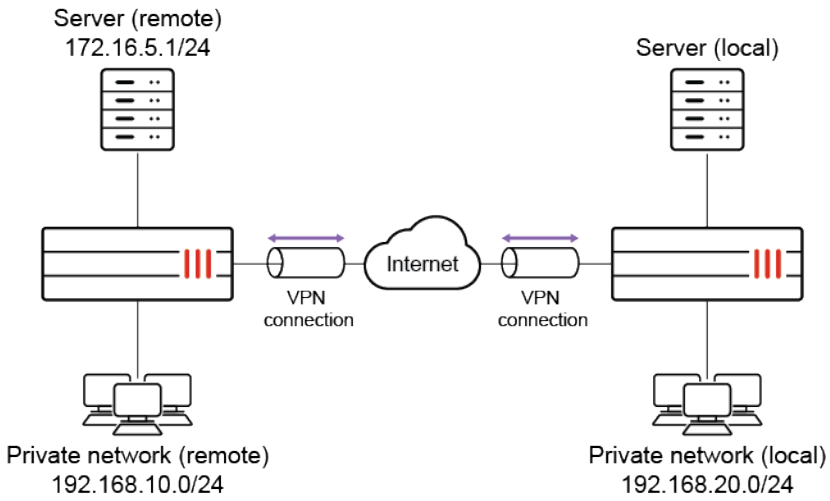
```
# diagnose vpn ike qkd qkdtest
ike.qkd.server.dns.addrs:
ike.qkd.server.curl.initiator.error.request.send.count: 10
ike.qkd.server.curl.initiator.error.request.send.last.ticks: 4295033761
ike.qkd.server.curl.initiator.error.request.send.last.ago: 1986
ike.qkd.server.curl.initiator.error.request.send.last.local: 2025-04-01 06:12:40 +1200
ike.qkd.server.curl.initiator.error.request.send.last.utc: 2025-03-31 18:12:40
```

```
ike.qkd.server.curl.initiator.error.request.send.value: 7
ike.qkd.server.curl.inflight: now 0  max 2  total 48
ike.qkd.server.curl.ssl.verify.count: 3952
ike.qkd.server.curl.ssl.verify.last.ticks: 4295231030
ike.qkd.server.curl.ssl.verify.last.ago: 13
ike.qkd.server.curl.ssl.verify.last.local: 2025-04-01 06:45:33 +1200
ike.qkd.server.curl.ssl.verify.last.utc: 2025-03-31 18:45:33
```

VPN security policies

This section explains how to specify the source and destination IP addresses of traffic transmitted through an IPsec VPN, and how to define appropriate security policies.

Topology



Defining policy addresses

In a gateway-to-gateway, hub-and-spoke, dynamic DNS, redundant tunnel, or transparent configuration, you need to define a policy address for the private IP address of the network behind the remote VPN peer (for example, `192.168.10.0/255.255.255.0` or `192.168.10.0/24`).

In a peer-to-peer configuration, you need to define a policy address for the private IP address of a server or host behind the remote VPN peer (for example, `172.16.5.1/255.255.255.255`, `172.16.5.1/32`, or `172.16.5.1`).

For a FortiGate dialup server in a dialup-client or internet-browsing configuration, the source IP should reflect the IP addresses of the dialup clients:

Defining security policies

Policy-based and route-based VPNs require different security policies.

- A policy-based VPN requires an IPsec policy. You specify the interface to the private network, the interface to the remote peer and the VPN tunnel. A single policy can enable traffic inbound, outbound, or in both directions.
- A route-based VPN requires an accept policy for each direction. For the source and destination interfaces, you specify the interface to the private network and the virtual IPsec interface (phase 1 configuration) of the VPN. The IPsec interface is the destination interface for the outbound policy and the source interface for the inbound policy. One security policy must be configured for each direction of each VPN interface.

If the policy that grants the VPN connection is limited to certain services, DHCP must be included, otherwise the client will not be able to retrieve a lease from the FortiGate's (IPsec) DHCP server because the DHCP request (coming out of the tunnel) will be blocked.

Policy-based VPN

An IPsec policy enables the transmission and reception of encrypted packets, specifies the permitted direction of VPN traffic, and selects the VPN tunnel. In most cases, a single policy is needed to control both inbound and outbound IP traffic through a VPN tunnel. For a detailed example, see Policy-based IPsec tunnel on page 2917. Be aware of the following before creating an IPsec policy.

Allow traffic to be initiated from the remote site

Policies specify which IP addresses can initiate a tunnel. By default, traffic from the local private network initiates the tunnel. When the *Allow traffic to be initiated from the remote site* option is selected, traffic from a dialup client, or a computer on a remote network, initiates the tunnel. Both can be enabled at the same time for bi-directional initiation of the tunnel.

Outbound and inbound NAT

When a FortiGate operates in NAT mode, you can enable inbound or outbound NAT. Outbound NAT may be performed on outbound encrypted packets or IP packets in order to change their source address before they are sent through the tunnel. Inbound NAT is performed to intercept and decrypt emerging IP packets from the tunnel.

By default, these options are not selected in security policies and can only be set through the CLI.

Defining multiple IPsec policies for the same tunnel

You must define at least one IPsec policy for each VPN tunnel. If the same remote server or client requires access to more than one network behind a local FortiGate, the FortiGate must be configured with an IPsec policy for each network. Multiple policies may be required to configure redundant connections to a remote destination or control access to different services at different times.

To ensure a secure connection, the FortiGate must evaluate policies with *Action* set to *IPsec* before *ACCEPT* and *DENY*. Because the FortiGate unit reads policies starting at the top of the list, you must move all IPsec policies to the top of the list, and be sure to reorder your multiple IPsec policies that apply to the tunnel so that specific constraints can be evaluated before general constraints. If you create two equivalent IPsec policies for two different tunnels, the system will select the correct policy based on the specified source and destination addresses.

Adding multiple IPsec policies for the same VPN tunnel can cause conflicts if the policies specify similar source and destination addresses, but have different settings for the same service. When policies overlap in this manner, the system may apply the wrong IPsec policy or the tunnel may fail.

Route-based VPN

When you define a route-based VPN, you create a virtual IPsec interface on the physical interface that connects to the remote peer. You create ordinary accept policies to enable traffic between the IPsec interface and the interface that connects to the private network. This makes configuration simpler than for policy-based VPNs.

To configure policies for a route-based VPN:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Click *Create New* and define an *ACCEPT* policy to permit communication between the local private network and the private network behind the remote peer and enter these settings in particular:

Name	Enter a name for the security policy.
Incoming Interface	Select the interface that connects to the private network behind this FortiGate.
Outgoing Interface	Select the IPsec interface you configured.
Source	Select the address name you defined for the private network behind this FortiGate.
Destination	Select the address name you defined for the private network behind the remote peer.
Action	Select *ACCEPT*.
NAT	Disable *NAT*.

3. Click *OK*.
   To permit the remote client to initiate communication, you need to define a security policy for communication in that direction.
4. Click *Create New* and enter these settings in particular:

Name	Enter a name for the security policy.
Incoming Interface	Select the IPsec interface you configured.
Outgoing Interface	Select the interface that connects to the private network behind this FortiGate.
Source	Select the address name you defined for the private network behind the remote peer.
Destination	Select the address name you defined for the private network behind this FortiGate.
Action	Select *ACCEPT*.
NAT	Disable *NAT*.

5. Click *OK*.

Encryption algorithms

This topic provides a brief introduction to IPsec phase 1 and phase 2 encryption algorithms and includes the following sections:

- IKEv2 phase 1 encryption algorithm
- IKEv2 phase 2 encryption algorithm
- IKEv1 phase 1 encryption algorithm
- IKEv1 phase 2 encryption algorithm
- HMAC settings

FortiGate by default only shows recommended configurations for IKE proposals in the CLI.

To set your preference for displaying IKE proposals:

```
config system settings
    set ike-proposal-visibility {recommended | all}
end
```

This setting is set to recommended by default. See Show only recommended IKE proposals by default on page 2771.

IKEv2 phase 1 encryption algorithm

The default encryption algorithm is:

`aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384 chacha20poly1305-prfsha256`

While MD5 and SHA1 may be configurable, they are not recommended.

DES is a symmetric-key algorithm, which means the same key is used for encrypting and decrypting data. FortiOS supports the following; however, they are not recommended:

- des-md5
- des-sha1
- des-sha256
- des-sha384
- des-sha512

3DES applies the DES algorithm three times to each data. FortiOS supports the following; however, they are not recommended.

- 3des-md5
- 3des-sha1
- 3des-sha256
- 3des-sha384
- 3des-sha512

AES is a symmetric-key algorithm with different key lengths (128, 192, and 256 bits). FortiOS supports:

- aes128-md5*
- aes128-sha1*
- aes128-sha256
- aes128-sha384
- aes128-sha512
- aes128gcm-prfsha1
- aes128gcm-prfsha256
- aes128gcm-prfsha384
- aes128gcm-prfsha512
- aes192-md5*
- aes192-sha1*
- aes192-sha256
- aes192-sha384
- aes192-sha512
- aes256-md5*
- aes256-sha1*
- aes256-sha256
- aes256-sha384
- aes256-sha512
- aes256gcm-prfsha1
- aes256gcm-prfsha256
- aes256gcm-prfsha384
- aes256gcm-prfsha512

*Not recommended

The ARIA algorithm is based on AES with different key lengths (128, 192, and 256 bits). FortiOS supports:

- aria128-md5*
- aria128-sha1*
- aria128-sha256
- aria128-sha384
- aria128-sha512
- aria192-md5*
- aria192-sha1*
- aria192-sha256
- aria192-sha384
- aria192-sha512
- aria256-md5*
- aria256-sha1*
- aria256-sha256
- aria256-sha384
- aria256-sha512

*Not recommended

With the chacha20poly1305 encryption algorithm, FortiOS supports:

- chacha20poly1305-prfsha1
- chacha20poly1305-prfsha256
- chacha20poly1305-prfsha384
- chacha20poly1305-prfsha512

SEED is a symmetric-key algorithm. FortiOS supports:

- seed128-md5*
- seed128-sha1*
- seed128-sha256
- seed128-sha384
- seed128-sha512

*Not recommended

Suite-B is a set of AES encryption with ICV in GCM mode. IPsec traffic can be offloaded on NP6XLite and NP7 platforms. They cannot be offloaded on other NP6 processors and below. CP9 supports Suite-B offloading, otherwise packets are encrypted and decrypted by software. FortiOS supports:

- suite-b-gcm-128
- suite-b-gcm-256

See Network processors (NP7, NP7Lite, NP6, NP6XLite, and NP6Lite) and Content processors (CP10, CP9, CP9XLite, CP9Lite) in the Hardware Acceleration guide for more information.

ShangMi SM4 is a 128-bit block cipher that uses a 128-bit key and a symmetric cipher. SM3 is a cryptographic hash function that produces a 256 bit digest. See SM3/SM4 cryptographic algorithms for IKEv1/ IKEv2 NEW on page 2924 for model limitations. FortiOS supports:

- SM4-SM3

SM4-SM3 does not support hardware acceleration and offloading. It is only available to physical or FortiGate-VMs using x86_64 processors.

IKEv2 phase 2 encryption algorithm

The default encryption algorithm is:

```
aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
```

With null encryption, IPsec traffic can offload NPU/CP. FortiOS supports the following, however, they are not recommended:

- null-md5
- null-sha1
- null-sha256
- null-sha384
- null-sha512

With the DES encryption algorithm, IPsec traffic can offload NPU/CP. FortiOS supports the following, however, they are not recommended:

- des-null
- des-md5
- des-sha1
- des-sha256
- des-sha384
- des-sha512

With the 3DES encryption algorithm, IPsec traffic can offload NPU/CP. FortiOS supports the following, however, they are not recommended:

- 3des-null
- 3des-md5
- 3des-sha1
- 3des-sha256
- 3des-sha384
- 3des-sha512

With the AES encryption algorithm, IPsec traffic can offload NPU/CP. FortiOS supports:

- aes128-null*
- aes128-md5*
- aes128-sha1*
- aes128-sha256
- aes128-sha384
- aes128-sha512
- aes192-null*
- aes192-md5*
- aes192-sha1*
- aes192-sha256
- aes192-sha384
- aes192-sha512
- aes256-null*
- aes256-md5*
- aes256-sha1*
- aes256-sha256
- aes256-sha384
- aes256-sha512

*Not recommended

NP7 and NP6XLite can offload the AESGCM encryption algorithm for IPsec traffic. CP9 supports AESGCM offloading. FortiOS supports:

- aes128gcm
- aes256gcm

With the chacha20poly1305 encryption algorithm, IPsec traffic **cannot** offload NPU/CP. FortiOS supports:

- chacha20poly1305

With the ARIA encryption algorithm, IPsec traffic **cannot** offload NPU/CP. FortiOS supports:

- aria128-null*
- aria128-md5*
- aria128-sha1*
- aria128-sha256
- aria128-sha384
- aria128-sha512
- aria192-null*
- aria192-md5*
- aria192-sha1*
- aria192-sha256
- aria192-sha384
- aria192-sha512
- aria256-null*
- aria256-md5*
- aria256-sha1*
- aria256-sha256
- aria256-sha384
- aria256-sha512

*Not recommended

With the SEED encryption algorithm, IPsec traffic **cannot** offload NPU/CP. FortiOS supports:

- seed-null*
- seed-md5*
- seed-sha1*
- seed-sha256
- seed-sha384
- seed-sha512

*Not recommended

ShangMi SM4 is a 128-bit block cipher that uses a 128-bit key and a symmetric cipher. SM3 is a cryptographic hash function that produces a 256 bit digest. See SM3/SM4 cryptographic algorithms for IKEv1/ IKEv2 NEW on page 2924 for model limitations. FortiOS supports:

- SM4-SM3

SM4-SM3 does not support hardware acceleration and offloading. It is only available to physical or FortiGate-VMs using x86_64 processors.

Post-Quantum Cryptography (PQC) algorithm for IKEv2

The following PQC key exchange mechanisms and cryptographic algorithms are supported:

ml-kem-512	ML-KEM-512. Module-Lattice-Based Key-Encapsulation Mechanism.
ml-kem-768	ML-KEM-768.
ml-kem-1024	ML-KEM-1024.
frodo-l1	FRODO L1. Frodo-KEM Level 1
frodo-l3	FRODO L3.
frodo-l5	FRODO L5.
bike-l1	BIKE L1. Bit Flipping Key Encapsulation Level 1
bike-l3	BIKE L3.
bike-l5	BIKE L5.

IKEv1 phase 1 encryption algorithm

The default encryption algorithm is:

```
aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
```

While MD5 and SHA1 may be configurable, they are not recommended.

DES is a symmetric-key algorithm, which means the same key is used for encrypting and decrypting data. FortiOS supports the following, however, they are not recommended:

- des-md5
- des-sha1
- des-sha256
- des-sha384
- des-sha512

3DES applies the DES algorithm three times to each data. FortiOS supports the following, however, they are not recommended:

- 3des-md5
- 3des-sha1
- 3des-sha256
- 3des-sha384
- 3des-sha512

AES is a symmetric-key algorithm with different key lengths (128, 192, and 256 bits). FortiOS supports:

- aes128-md5*
- aes128-sha1*
- aes128-sha256
- aes128-sha384
- aes128-sha512
- aes192-md5*
- aes192-sha1

- aes192-sha256
- aes192-sha384
- aes192-sha512
- aes256-md5*
- aes256-sha1*
- aes256-sha256
- aes256-sha384
- aes256-sha512

*Not recommended

The ARIA algorithm is based on AES with different key lengths (128, 192, and 256 bits). FortiOS supports the following from the CLI only:

- aria128-md5*
- aria128-sha1*
- aria128-sha256
- aria128-sha384
- aria128-sha512
- aria192-md5*
- aria192-sha1*
- aria192-sha256
- aria192-sha384
- aria192-sha512
- aria256-md5*
- aria256-sha1*
- aria256-sha256
- aria256-sha384
- aria256-sha512

*Not recommended

SEED is a symmetric-key algorithm. FortiOS supports:

- seed128-md5*
- seed128-sha1*
- seed128-sha256
- seed128-sha384
- seed128-sha512

*Not recommended

Suite-B is a set of AES encryption with ICV in GCM mode. IPsec traffic can be offloaded on NP6XLite and NP7 platforms. They cannot be offloaded on other NP6 processors and below. CP9 supports Suite-B offloading, otherwise packets are encrypted and decrypted by software. FortiOS supports:

- suite-b-gcm-128
- suite-b-gcm-256

See Network processors (NP7, NP7Lite, NP6, NP6XLite, and NP6Lite) and Content processors (CP10, CP9, CP9XLite, CP9Lite) in the Hardware Acceleration guide for more information.

IKEv1 phase 2 encryption algorithm

The default encryption algorithm is:

```
aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
```

With null encryption, IPsec traffic can offload NPU/CP. FortiOS supports the following, however, they are not recommended:

- null-md5
- null-sha1
- null-sha256
- null-sha384
- null-sha512

With the DES encryption algorithm, IPsec traffic can offload NPU/CP. FortiOS supports the following, however, they are not recommended:

- des-null
- des-md5
- des-sha1
- des-sha256
- des-sha384
- des-sha512

With the 3DES encryption algorithm, IPsec traffic can offload NPU/CP. FortiOS supports the following, however, they are not recommended:

- 3des-null
- 3des-md5
- 3des-sha1
- 3des-sha256
- 3des-sha384
- 3des-sha512

With the AES encryption algorithm, IPsec traffic can offload NPU/CP. FortiOS supports:

- aes128-null*
- aes128-md5*
- aes128-sha1*
- aes128-sha256
- aes128-sha384
- aes128-sha512
- aes192-null*
- aes192-md5*
- aes192-sha1*
- aes192-sha256
- aes192-sha384
- aes192-sha512
- aes256-null*

- aes256-md5*
- aes256-sha1*
- aes256-sha256
- aes256-sha384
- aes256-sha512

*Not recommended

With the AESGCM encryption algorithm, IPsec traffic **cannot** offload NPU/CP. FortiOS supports:

- aes128gcm
- aes256gcm

With the chacha20poly1305 encryption algorithm, IPsec traffic **cannot** offload NPU/CP. FortiOS supports:

- chacha20poly1305

With the ARIA encryption algorithm, IPsec traffic **cannot** offload NPU/CP. FortiOS supports:

- aria128-null*
- aria128-md5*
- aria128-sha1*
- aria128-sha256
- aria128-sha384
- aria128-sha512
- aria192-null*
- aria192-md5*
- aria192-sha1*
- aria192-sha256
- aria192-sha384
- aria192-sha512
- aria256-null*
- aria256-md5*
- aria256-sha1*
- aria256-sha256
- aria256-sha384
- aria256-sha512

*Not recommended

With the SEED encryption algorithm, IPsec traffic **cannot** offload NPU/CP. FortiOS supports:

- seed-null*
- seed-md5*
- seed-sha1*
- seed-sha256
- seed-sha384
- seed-sha512

*Not recommended

HMAC settings

The FortiGate uses the HMAC based on the authentication proposal that is chosen in phase 1 or phase 2 of the IPsec configuration. Each proposal consists of the encryption-hash pair (such as 3des-sha256). The FortiGate matches the most secure proposal to negotiate with the peer.

To view the chosen proposal and the HMAC hash used:

```
# diagnose vpn ike gateway list

vd: root/0
name: MPLS
version: 1
interface: port1 3
addr: 192.168.2.5:500 -> 10.10.10.1:500
tun_id: 10.10.10.1
virtual-interface-addr: 172.31.0.2 -> 172.31.0.1
created: 1015820s ago
IKE SA: created 1/13 established 1/13 time 10/1626/21010 ms
IPsec SA: created 1/24 established 1/24 time 0/11/30 ms

  id/spi: 124 43b087dae99f7733/6a8473e58cd8990a
  direction: responder
  status: established 68693-68693s ago = 10ms
  proposal: 3des-sha256
  key: e0fa6ab8dc509b33-aa2cc549999b1823-c3cb9c337432646e
  lifetime/rekey: 86400/17436
  DPD sent/recv: 000001e1/00000000
```

Blocking unwanted IKE negotiations and ESP packets with a local-in policy

It is not unusual to receive IPsec connection attempts or malicious IKE packets from all over the internet. Malicious parties use these probes to try to establish an IPsec tunnel in order to gain access to your private network. A good way to prevent this is to use local-in policies to deny such traffic.

Sometimes there are malicious attempts using crafted invalid ESP packets. These invalid attempts are automatically blocked by the FOS IPsec local-in handler when it checks the SPI value against the SAs of existing tunnels. The IPsec local-in handler processes the packet instead of the firewall's local-in handler. So when these attempts are blocked, you will notice an `unknown SPI` message in your VPN logs instead of being silently blocked by your local-in policy. These log messages are rate limited.

Sample log and alert email

```
Message meets Alert condition
```

```
date=2020-08-11 time=09:28:40 devname=toSite1 devid=FGT60Fxxxxxxxxxx logid="0101037131"
type="event" subtype="vpn" level="error" vd="root" eventtime=1597163320747963100 tz="-0700"
logdesc="IPsec ESP" msg="IPsec ESP" action="error" remip=131.62.25.102 locip=192.157.116.88
remport=40601 locport=500 outintf="wan1" cookies="N/A" user="N/A" group="N/A" xauthuser="N/A"
xauthgroup="N/A" assignip=N/A vpntunnel="N/A" status="esp_error" error_num="Received ESP packet
with unknown SPI." spi="f6c9e2x1" seq="02000400"
```

The ESP packet handling process has the detection of unknown ESP packets enabled by default. You can disable the detection of unknown ESP packets using the `detect-unknown-esp` command.

To configure unknown ESP packet detection:

```
config system settings
    set detect-unknown-esp {enable | disable}
end
```

Note that invalid SPIs may not always indicate malicious activity. For example, the SPI may not match during rekey, or when one unit flushes its tunnel SAs. Administrators should collect as much information as possible before making a conclusion.

To block undesirable IPsec connection attempts and IKE packets using a local-in policy:

1. Configure an address group that excludes legitimate IPs:

```
config firewall addrgrp
    edit "All_exceptions"
        set member "all"
        set exclude enable
        set exclude-member "remote-vpn"
    next
end
```

2. Create a local-in policy that blocks IKE traffic from the address group:

```
config firewall local-in-policy
    edit 1
        set intf "wan1"
        set srcaddr "All_exceptions"
        set dstaddr "all"
        set service "IKE"
        set schedule "always"
    next
end
```

The default action is deny.

3. Verify the traffic blocked by the local-in policy:

```
# diagnose debug flow filter dport 500
# diagnose debug flow trace start 10
# diagnose debug enable
```

```
id=20085 trace_id=290 func=print_pkt_detail line=5588 msg="vd-root:0 received a packet
(proto=17, 10.10.10.13:500->10.10.10.1:500) from wan1. "
id=20085 trace_id=290 func=init_ip_session_common line=5760 msg="allocate a new session-
003442e7"
id=20085 trace_id=290 func=vf_ip_route_input_common line=2598 msg="find a route: flag=84000000
gw-10.10.10.1 via root"
id=20085 trace_id=290 func=fw_local_in_handler line=430 msg="iprope_in_check() check failed on
policy 1, drop"
```

Renaming IPsec tunnels

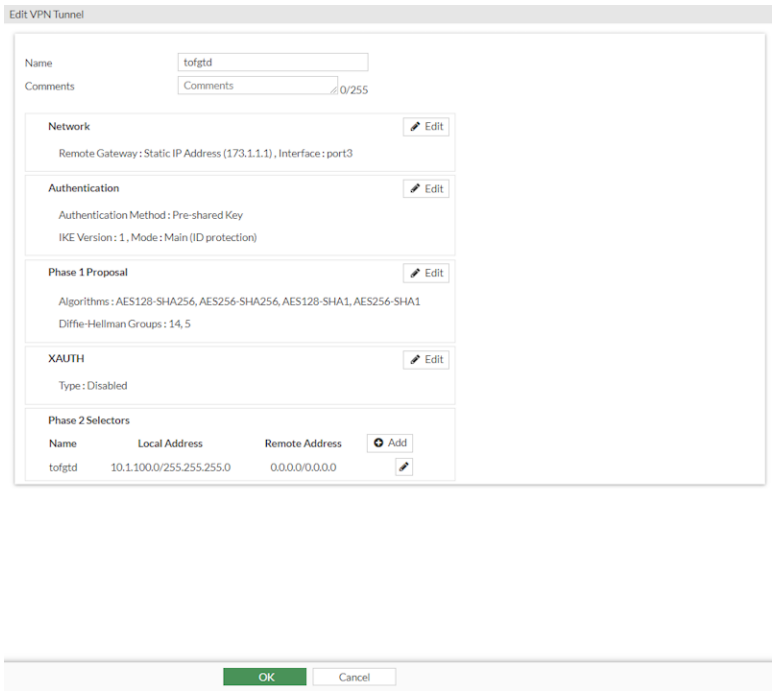
IPsec tunnels can be renamed. When you rename an IPsec tunnel, all references to the tunnel, such as routing and policies, are automatically updated to reflect the new name.

```
config vpn ipsec phase1-interface
    rename <string> to <string>
end
```

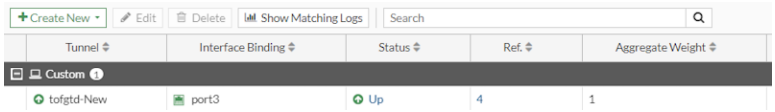
In this example, the IPsec tunnel named *tofgtd* is renamed to *tofgtd-New*, and all associated references are updated.

To rename an IPsec tunnel in the GUI:

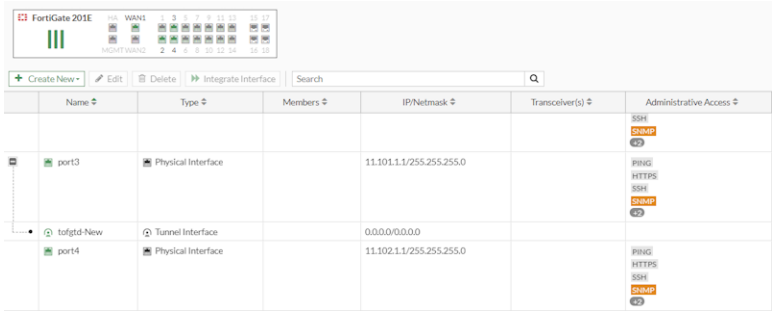
1. Go to *VPN* > *IPsec Tunnels* and double-click an IPsec tunnel to open it for editing.
   In this example, the IPsec tunnel name is *tofgtd*.



2. In the *Name* box, type a new name, and click *OK*. The IPsec tunnel is renamed, and all associated references are updated.
   In this example, the IPsec tunnel is renamed to *tofgtd-New*.

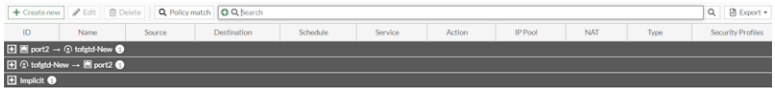


3. Check the associated references:
   In this example, all associated references show the new IPsec tunnel name of *tofgtd-New*.
   - Go to *Network* > *Interfaces* to see that the interface references the new IPsec tunnel name.



   - Go to *Network* > *Static Routes* to see that the static route references the new IPsec tunnel name.

   - Go to *Policy & Objects* > *Firewall Policy* to see that the policy references the new IPsec tunnel name



To rename an IPsec tunnel in the CLI:

1. Rename the IPsec tunnel.
   In this example, the IPsec tunnel named *tofgtd* is renamed to *tofgtd-New*:

```
config vpn ipsec phase1-interface
    rename tofgtd to tofgtd-New
end
```

2. Show the configuration to confirm that the IPsec tunnel was renamed.
   In this example, the IPsec tunnel was renamed to *tofgtd-New*:

```
show
config vpn ipsec phase1-interface
    edit "tofgtd-New"
        set interface "port3"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set dpd disable
        set remote-gw 173.1.1.1
```

```
        ...
    next
end
```

3. Check the associated references.
   In this example, all associated references show the new IPsec tunnel name of *tofgtd-New*.
   - Confirm that the interfaces reference the new IPsec tunnel name:

```
config router static
show
config router static
    edit 3
        set dst 192.168.5.0 255.255.255.0
        set device "tofgtd-New"
    next
end
```

   - Confirm that the static route references the new IPsec tunnel name:

```
config system interface
show
    edit "tofgtd-New"
    ....
end
```

   - Confirm that the policies references the new IPsec tunnel name:

```
config firewall policy
show
config firewall policy
    edit 1
        set uuid 802c6c2e-8368-51ee-bf40-6c3c32da1024
        set srcintf "port2"
        set dstintf "tofgtd-New"
        set action accept
        ...
    next
    edit 2
        set uuid 80d136aa-8368-51ee-cc52-b0b06306fb80
        set srcintf "tofgtd-New"
        set dstintf "port2"
        set action accept
        ...
    next
end
```

Site-to-site VPN

A site-to-site VPN connection lets branch offices use the Internet to access the main office's intranet. A site-to-site VPN allows offices in multiple, fixed locations to establish secure connections with each other over a public network such as the Internet.

The following sections provide instructions for configuring site-to-site VPNs:

- FortiGate-to-FortiGate on page 2857
- FortiGate-to-third-party on page 2935

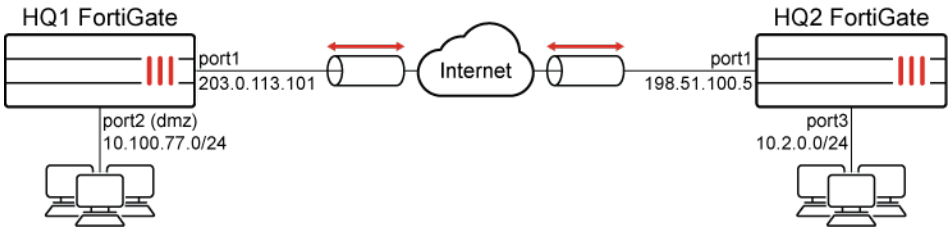
FortiGate-to-FortiGate

This section contains the following topics about FortiGate-to-FortiGate VPN configurations:

- Basic site-to-site VPN with pre-shared key on page 2857
- Basic site-to-site VPN with digital certificate on page 2867
- Site-to-site VPN with overlapping subnets on page 2879
- Tunneled Internet browsing on page 2891
- Geography-based tunneled internet browsing New on page 2900
- VPN over TCP with Fortinet proprietary encapsulation on page 2905
- GRE over IPsec on page 2911
- Policy-based IPsec tunnel on page 2917
- SM3/SM4 cryptographic algorithms for IKEv1/ IKEv2 NEW on page 2924
- Securely exchange serial numbers between FortiGates connected with IPsec VPN on page 2930

Basic site-to-site VPN with pre-shared key

This example shows how to configure a basic site-to-site IPsec tunnel between two FortiGates. This IKEv2 tunnel will use a pre-shared key to authenticate. Only traffic from the local subnets are permitted through the tunnel.



Individual options may vary. However, in order for the tunnel to come up, options configured on one FortiGate should be identical to the other.

For more information on IPsec settings, see Phase 1 configuration on page 2761 and Phase 2 configuration on page 2795.

For optimal security, it is advised to use digital certificates for authentication rather than pre-shared keys. See Basic site-to-site VPN with digital certificate on page 2867.

To configure an IPsec VPN using the VPN Wizard in the GUI:

1. Configure the HQ1 FortiGate.
   a. Go to *VPN > VPN Wizard* and configure the following settings for VPN Setup:
      i. Enter a VPN name in the *Tunnel name* field. In this example, *HQ2B2*.
      ii. From the *Select a template* options, select *Site to Site*.
      iii. Click *Begin*.
   b. Configure the VPN tunnel:
      i. For *Authentication Method*, select *Pre-shared Key*.
      ii. In the *Pre-shared Key* field, enter your key.
      iii. For *IKE*, select *Version 2*.
      iv. For *Transport*, if the option is grayed out and set to *UDP*, this means that *Allow VPN negotiation over TCP* has not been enabled. If the *Transport* can be selected, select *Auto*. Leave the *Use Fortinet encapsulation* option toggled off.

      When *Auto* is selected, UDP transport will be attempted first. If this fails, VPN negotiations over TCP will be attempted.

      v. For *NAT traversal*, select *Enable*.
      vi. For *Keepalive frequency*, enter *10*.
      vii. Click *Next*.
   c. Configure the *Remote Site*:
      i. For *Remote site device type*, select *FortiGate*.
      ii. For *Remote site device*, select *Accessible and static*.
      iii. For the *IP address*, enter *198.51.100.5*.
      iv. For *Remote site subnets that can access VPN*, enter 10.2.0.0/24.
      v. Click *Next*.
   d. Configure the *Local Site*:
      i. For *Outgoing interface that binds to tunnel*, select the WAN interface. In this example, *port1*. Disable the *Create and add interface to zone* toggle, if you do not want a zone created.
      ii. For *Local Interface*, select an interface from the *Select Entries* slide-out panel. In this example, *port2*.
      iii. The *Local subnets that can access VPN* field should automatically fill with the IP address related to the selected local interface (*10.100.77.0/24*).
      iv. Leave the *Allow remote site's internet traffic through this device* toggle disabled.
      v. Click *Next*.
   e. Review the configuration and click *Submit*.
2. Configure the HQ2 FortiGate

a. Go to *VPN > VPN Wizard* and configure the following settings for VPN Setup:
   i. Enter a VPN name in the *Tunnel name* field. In this example, *B2toHQ*.
   ii. From the *Select a template* options, select *Site to Site*.
   iii. Click *Begin*.
b. Configure the *VPN tunnel*:
   i. For *Authentication Method*, select *Pre-shared Key*.
   ii. In the *Pre-shared Key* field, enter your key.
   iii. For *IKE*, select *Version 2*.
   iv. If *Transport* can be selected, select *Auto*. Leave the *Use Fortinet encapsulation* option toggled off.
   v. For *NAT traversal*, select *Enable*.
   vi. For *Keepalive frequency*, enter *10*.
   vii. Click *Next*.
c. Configure the *Remote Site*:
   i. For *Remote site device type*, select *FortiGate*.
   ii. For *Remote site device*, select *Accessible and static*.
   iii. For the *IP address*, enter *203.0.113.101*.
   iv. For *Remote site subnets that can access VPN*, enter *10.100.77.0/24*.
   v. Click *Next*.
d. Configure the *Local Site*:
   i. For *Outgoing interface that binds to tunnel*, select the WAN interface. In this example, *port1*. Leave the *Create and add interface to zone* toggle enabled.
   ii. For *Local Interface*, select an interface from the *Select Entries* slide-out panel. In this example, *port3*.
   iii. The *Local subnets that can access VPN* field should automatically fill with the IP address related to the selected local interface (*10.2.0.0/24*).
   iv. Leave the *Allow remote site's internet traffic through this device* toggle disabled.
   v. Click *Next*.

Verify the VPN tunnel configurations created in the VPN Wizard

Verify the VPN tunnel configurations for HQ1 and HQ2 created in the *VPN Wizard* using the CLI.

Verify the HQ1 configuration

```
config firewall address
    edit "HQ2B2_local_subnet_0"
        set allow-routing enable
        set subnet 10.100.77.0 255.255.255.0
    next
    edit "HQ2B2_remote_subnet_0"
        set allow-routing enable
        set subnet 10.2.0.0 255.255.255.0
    next
end
```

```
config firewall addrgrp
    edit "HQ2B2_local"
        set allow-routing enable
        set member "HQ2B2_local_subnet_0"
        set comment "VPN: HQ2B2 -- Created by VPN wizard"
    next
    edit "HQ2B2_remote"
        set allow-routing enable
        set member "HQ2B2_remote_subnet_0"
        set comment "VPN: HQ2B2 -- Created by VPN wizard"
    next
end
```

```
config vpn ipsec phase1-interface
    edit "HQ2B2"
        set interface "port1"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set comments "VPN: HQ2B2 -- Created by VPN wizard"
        set dhgrp 20 21
        set wizard-type static-fortigate
        set remote-gw 198.51.100.5
        set psksecret ENC <encrypted key>
    next
end
```

```
config vpn ipsec phase2-interface
    edit "HQ2B2"
        set phase1name "HQ2B2"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
        set comments "VPN: HQ2B2 -- Created by VPN wizard"
        set src-addr-type name
        set dst-addr-type name
        set src-name "HQ2B2_local"
        set dst-name "HQ2B2_remote"
    next
end
```

```
config router static
    edit 14
        set device "HQ2B2"
        set comment "VPN: HQ2B2 -- Created by VPN wizard"
        set dstaddr "HQ2B2_remote"
    next
    edit 16
        set distance 254
        set comment "VPN: HQ2B2 -- Created by VPN wizard"
        set blackhole enable
```

```
        set dstaddr "HQ2B2_remote"
        set vrf 0
    next
end
```

```
config firewall policy
    edit 29
        set name "vpn_HQ2B2_local"
        set srcintf "port2"
        set dstintf "HQ2B2"
        set action accept
        set srcaddr "HQ2B2_local"
        set dstaddr "HQ2B2_remote"
        set schedule "always"
        set service "ALL"
        set comments "VPN: HQ2B2 -- Created by VPN wizard"
    next
    edit 32
        set name "vpn_HQ2B2_remote"
        set srcintf "HQ2B2"
        set dstintf "port2"
        set action accept
        set srcaddr "HQ2B2_remote"
        set dstaddr "HQ2B2_local"
        set schedule "always"
        set service "ALL"
        set comments "VPN: HQ2B2 -- Created by VPN wizard"
    next
end
```

Verify the HQ2 configuration

```
config firewall address
    edit "B2toHQ_local_subnet_0"
        set allow-routing enable
        set subnet 10.2.0.0 255.255.255.0
    next
    edit "B2toHQ_remote_subnet_0"
        set allow-routing enable
        set subnet 10.100.77.0 255.255.255.0
    next
end
```

```
config firewall addrgrp
    edit "B2toHQ_local"
        set allow-routing enable
        set member "B2toHQ_local_subnet_0"
        set comment "VPN: B2toHQ -- Created by VPN wizard"
    next
    edit "B2toHQ_remote"
        set allow-routing enable
```

```
        set member "B2toHQ_remote_subnet_0"
        set comment "VPN: B2toHQ -- Created by VPN wizard"
    next
end
```

```
config vpn ipsec phase1-interface
    edit "B2toHQ"
        set interface "port1"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set comments "VPN: B2toHQ -- Created by VPN wizard"
        set dhgrp 20 21
        set wizard-type static-fortigate
        set remote-gw 203.0.113.101
        set psksecret ENC <encrypted key>
    next
end
```

```
config vpn ipsec phase2-interface
    edit "B2toHQ"
        set phase1name "B2toHQ"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
        set comments "VPN: B2toHQ -- Created by VPN wizard"
        set src-addr-type name
        set dst-addr-type name
        set src-name "B2toHQ_local"
        set dst-name "B2toHQ_remote"
    next
end
```

```
config firewall policy
    edit 5
        set name "vpn_B2toHQ_local"
        set uuid dbc31212-9326-51ef-fea3-ea8cb4e2a064
        set srcintf "port3"
        set dstintf "B2toHQ"
        set action accept
        set srcaddr "B2toHQ_local"
        set dstaddr "B2toHQ_remote"
        set schedule "always"
        set service "ALL"
        set comments "VPN: B2toHQ -- Created by VPN wizard"
    next
    edit 6
        set name "vpn_B2toHQ_remote"
        set uuid dbc58e3e-9326-51ef-1896-d5d2f3c1932d
        set srcintf "B2toHQ"
        set dstintf "port3"
```

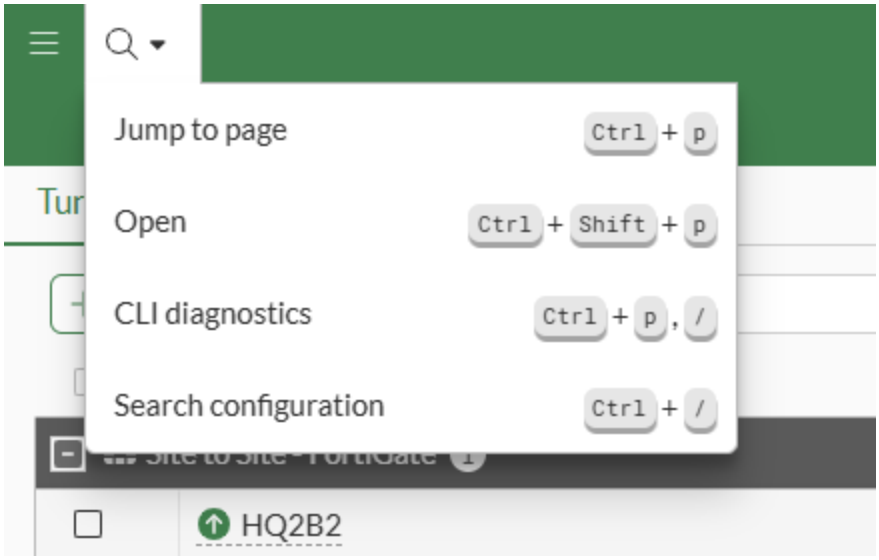
```
        set action accept
        set srcaddr "B2toHQ_remote"
        set dstaddr "B2toHQ_local"
        set schedule "always"
        set service "ALL"
        set comments "VPN: B2toHQ -- Created by VPN wizard"
    next
end
```

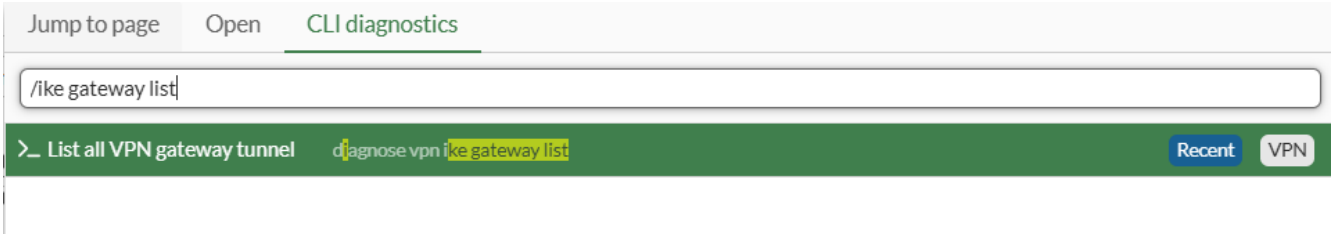
```
config router static
    edit 7
        set device "B2toHQ"
        set comment "VPN: B2toHQ -- Created by VPN wizard"
        set dstaddr "B2toHQ_remote"
    next
    edit 9
        set distance 254
        set comment "VPN: B2toHQ -- Created by VPN wizard"
        set blackhole enable
        set dstaddr "B2toHQ_remote"
        set vrf 0
    next
end
```

Testing the tunnel

To verify that a tunnel is established, goo to *Dashboard* > *Network Monitor* > *VPN*, or use the `diagnose vpn ike gateway list name <tunnel>` CLI command.

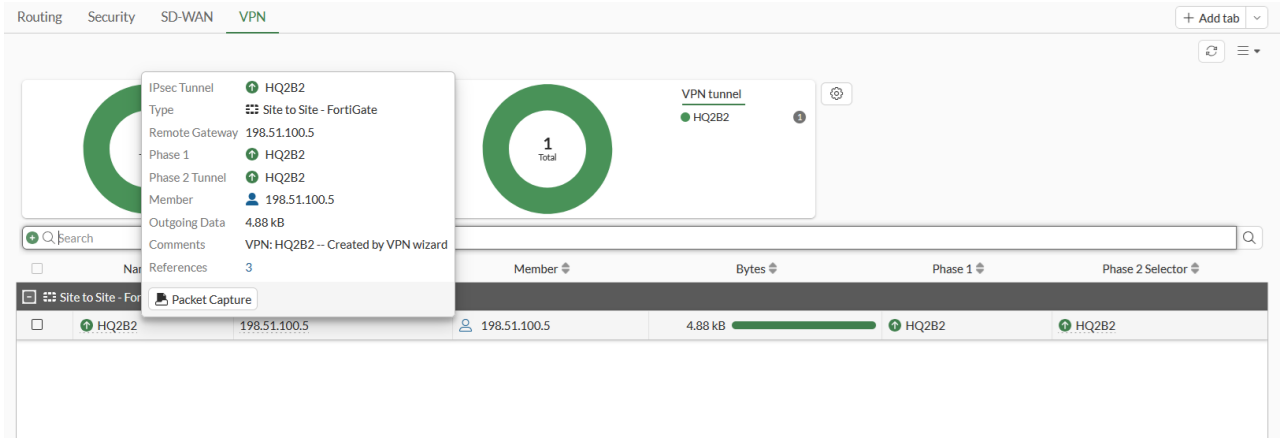
You can also use the command palette for CLI diagnostics. Open the command palette from the Find icon at the top of the page and then search for the command to run:





To verify the HQ1 tunnel in the GUI:

1. Go to *Dashboard* > *Network Monitor* and select the *VPN* tab.
2. Look for the newly created tunnel, in this example, *HQ2B2*, and hover the cursor over the tunnel name and observe additional details.



To verify the HQ1 tunnel in the CLI:

```
# diagnose vpn ike gateway list name HQ2B2
vd: root/0
name: HQ2B2
version: 2
interface: port1 3
addr: 203.0.113.101:500 -> 198.51.100.5:500
tun_id: 198.51.100.5/::198.51.100.5
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 852804s ago
peer-id: 198.51.100.5
peer-id-auth: no
pending-queue: 0
PPK: no
IKE SA: created 1/11  established 1/11  time 0/820/9010 ms
IPsec SA: created 1/21  established 1/21  time 0/433/9010 ms

id/spi: 8895 45e69522bd784183/69be2940ae314677
direction: responder
status: established 77898-77898s ago = 0ms
```

```
proposal: aes128-sha256
child: yes
SK_ei: 07446ef38bd406fa-18e45bec886a2bd3
SK_er: b940f4338b27f369-7beaa308b7241a1b
SK_ai: 61eeb720d062c70b-e606340beab6f9d5-3a4da2966373ab15-e5766540cf6c7dd1
SK_ar: bbdd888b98dbda7b-669da7a726ce2148-3f08940f802f580c-52b6ef93f48672aa
message-id sent/recv: 0/1783
QKD: no
lifetime/rekey: 86400/8231
DPD sent/recv: 00000000/00000000
peer-id: 198.51.100.5
```

HQ1 acted as the responder, and established the tunnel with remote peer *198.51.100.5* over UDP/500.

To verify the HQ2 tunnel in the GUI:

1. Go to *Dashboard > Network Monitor > VPN*.
2. Look for the newly created tunnel, in this example, *B2toHQ*, and hover the cursor over the tunnel name and observe additional details.

Verify the HQ2 tunnel

1. Go to *Dashboard > Network Monitor > VPN*.
2. Look for the newly created tunnel, in this example, *B2toHQ*, and hover the cursor over the tunnel name and observe additional details.

To verify the HQ2 tunnel in the CLI:

```
# diagnose vpn ike gateway list name B2toHQ
vd: root/0
name: B2toHQ
version: 2
interface: port1 3
addr: 198.51.100.5:500 -> 203.0.113.101:500
tun_id: 203.0.113.101/::203.0.113.101
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 853181s ago
peer-id: 203.0.113.101
peer-id-auth: no
pending-queue: 0
PPK: no
IKE SA: created 1/11  established 1/11  time 0/5/10 ms
IPsec SA: created 1/21  established 1/21  time 0/0/10 ms

id/spi: 5788 45e69522bd784183/69be2940ae314677
direction: initiator
status: established 78281-78281s ago = 10ms
proposal: aes128-sha256
child: yes
```

```
SK_ei: 07446ef38bd406fa-18e45bec886a2bd3
SK_er: b940f4338b27f369-7beaa308b7241a1b
SK_ai: 61eeb720d062c70b-e606340beab6f9d5-3a4da2966373ab15-e5766540cf6c7dd1
SK_ar: bbdd888b98dbda7b-669da7a726ce2148-3f08940f802f580c-52b6ef93f48672aa
message-id sent/recv: 1792/0
QKD: no
lifetime/rekey: 86400/7818
DPD sent/recv: 00001ed7/00001ed7
peer-id: 203.0.113.101
```

HQ2 acted as the initiator, and established the tunnel with remote peer *203.0.113.101* over UDP/500.

Verifying traffic flow

Send traffic from *PC1 10.100.77.101* to the *10.2.0.0/24* network and verify traffic goes through.

To verify traffic on PC1 (10.100.77.101):

```
root@dmz-public-server:~# ping 10.2.0.7
PING 10.2.0.7 (10.2.0.7) 56(84) bytes of data.
64 bytes from 10.2.0.7: icmp_seq=1 ttl=62 time=9.84 ms
64 bytes from 10.2.0.7: icmp_seq=2 ttl=62 time=0.808 ms
```

To verify traffic on HQ1:

In the CLI, run `diagnose sniffer packet any 'host 10.2.0.7' 4 0 l` and observe the output:

```
HQ1 # diagnose sniffer packet any 'host 10.2.0.7' 4 0 l
Using Original Sniffing Mode
interfaces=[any]
filters=[host 10.2.0.7]
2024-11-05 12:06:05.509027 port2 in 10.100.77.101 -> 10.2.0.7: icmp: echo request
2024-11-05 12:06:05.509074 HQ2B2 out 10.100.77.101 -> 10.2.0.7: icmp: echo request
2024-11-05 12:06:05.518607 HQ2B2 in 10.2.0.7 -> 10.100.77.101: icmp: echo reply
2024-11-05 12:06:05.518632 port2 out 10.2.0.7 -> 10.100.77.101: icmp: echo reply
```

The ICMP request is received on port2 and routed to the HQ2B2 tunnel. The ICMP reply is received on the HQ2B2 tunnel and routed back out port2.

To verify traffic on HQ2:

In the CLI, run `diagnose sniffer packet any 'host 10.100.77.101' 4 0 l` and observe the output:

```
HQ2 # diagnose sniffer packet any 'host 10.100.77.101' 4 0 l
Using Original Sniffing Mode
interfaces=[any]
filters=[host 10.100.77.101]
2024-11-05 12:06:05.509711 B2toHQ in 10.100.77.101 -> 10.2.0.7: icmp: echo request
2024-11-05 12:06:05.509791 port3 out 10.100.77.101 -> 10.2.0.7: icmp: echo request
2024-11-05 12:06:05.518809 port3 in 10.2.0.7 -> 10.100.77.101: icmp: echo reply
2024-11-05 12:06:05.518899 B2toHQ out 10.2.0.7 -> 10.100.77.101: icmp: echo reply
```

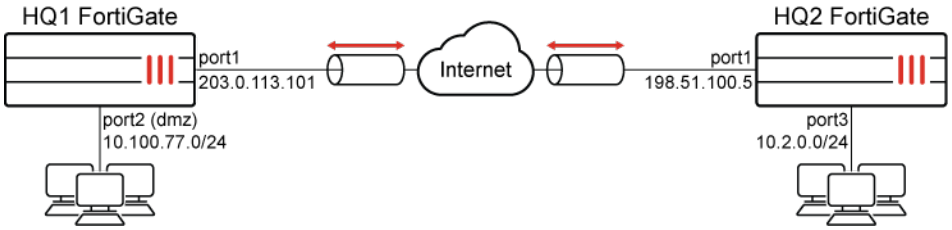
The ICMP request is received on the B2toHQ tunnel and routed to port3. The ICMP reply is received on port3 and routed back out the B2toHQ tunnel.

Basic site-to-site VPN with digital certificate

This is a sample configuration of IPsec VPN authenticating a remote FortiGate peer with a certificate. The certificate on one peer is validated by the presence of the CA certificate that is installed on the other peer.

The VPN Wizard is used to configure a site-to-site VPN tunnel.

It is assumed that the certificates have been configured and imported to the FortiGates. For information about configuring certificates using third party tools, such as XCA, see Creating certificates with XCA on page 3982. For information about importing a certificate, see Import a certificate on page 3962.



Individual settings can vary, but in order for the tunnel to come up, the configuration must be the same on both FortiGates.

For more information on IPsec settings, see Phase 1 configuration on page 2761 and Phase 2 configuration on page 2795.

To configure an IPsec VPN using the VPN Wizard on the HQ1 FortiGate:

1. Go to *VPN > VPN Wizard*.
2. Enter a name for the VPN in the *Tunnel name* field. In this example, *HQ2B2*.
3. Set *Select a template* to *Site to Site*.
4. Click *Begin*.
5. Configure the *VPN tunnel* settings:

Field	Value
Authentication method	*Signature*
Certificate name	Select a certificate.
Peer certificate CA	Select a peer CA certificate.
IKE	*Version 2*
Transport	If the option is grayed out and set to *UDP*, this means that *Allow VPN negotiation over TCP* has not been enabled. If the *Transport* can be selected, select *Auto*. Leave *Use Fortinet encapsulation* toggled off.
NAT traversal	*Enable* Set *Keepalive frequency* to *10*.

6. Click *Next*.
7. Configure the *Remote site* settings:

Field	Value
Remote site device type	*FortiGate*
Remote site device	*Accessible and static* Set *IP/FQDN* to *198.51.100.5*.
Route this device's internet traffic through the remote site.	Disabled
Remote site subnets that can access VPN	*10.2.0.0/24*

8. Click *Next*.
9. Configure the *Local site* settings:

Field	Value
Outgoing interface that binds to tunnel	*port1* (the WAN interface in this example) Disable *Create and add interface to zone* if you do not need a zone to be created.
Local interface	*port2* (in this example)
Local subnets that can access VPN	This field automatically uses the IP address related to the selected local interface. In this example, 10.100.77.0/24
Allow remote site's internet traffic through this device	Disabled

10. Click *Next*.
11. Review the configuration and, if everything is correct, click *Submit*.

To configure an IPsec VPN using the VPN Wizard on the HQ2 FortiGate:

1. Go to *VPN > VPN Wizard*.
2. Enter a name for the VPN in the *Tunnel name* field. In this example, *B2toHQ*.
3. Set *Select a template* to *Site to Site*.
4. Click *Begin*.
5. Configure the *VPN tunnel* settings:

Field	Value
Authentication method	*Signature*
Certificate name	Select a certificate.
Peer certificate CA	Select a peer CA certificate.

Field	Value
IKE	*Version 2*
Transport	If the option is grayed out and set to *UDP*, this means that *Allow VPN negotiation over TCP* has not been enabled. If the *Transport* can be selected, select *Auto*. Leave *Use Fortinet encapsulation* toggled off.
NAT traversal	*Enable* Set *Keepalive frequency* to *10*.

6. Click *Next*.
7. Configure the *Remote site* settings:

Field	Value
Remote site device type	*FortiGate*
Remote site device	*Accessible and static* Set *IP/FQDN* to *203.0.113.101*.
Route this device's internet traffic through the remote site.	Disabled
Remote site subnets that can access VPN	*10.100.77.0/24*

8. Click *Next*.
9. Configure the *Local site* settings:

Field	Value
Outgoing interface that binds to tunnel	*port1* (the WAN interface in this example) Enable *Create and add interface to zone*.
Local interface	*port3* (in this example)
Local subnets that can access VPN	This field automatically uses the IP address related to the selected local interface. In this example, 10.2.0.0/24
Allow remote site's internet traffic through this device	Disabled

10. Click *Next*.
11. Review the configuration and, if everything is correct, click *Submit*.

To configure HQ1 for IPsec VPN authenticating a remote FortiGate peer with a digital certificate using the CLI:

1. Configure the WAN interface and default route.
   The WAN interface is the interface connected to the ISP. The IPsec tunnel is established over the WAN interface.

```
config system interface
    edit "port1"
        set vdom "root"
        set ip 203.0.113.101 255.255.255.0
    next
end
config router static
    edit 1
        set gateway 203.0.113.102
        set device "port1"
    next
end
```

2. Configure address objects for internal protected networks.

```
config firewall address
    edit "10.2.0.0"
        set subnet 10.2.0.0 255.255.255.0
    next
    edit "10.100.77.0"
        set subnet 10.100.77.0 255.255.255.0
    next
end
```

3. Configure the internal (protected subnet) interface.

   The internal interface connects to the corporate internal network. Traffic from this interface routes out the IPsec VPN tunnel.

```
config system interface
    edit "port2"
        set vdom "root"
        set ip 10.100.77.0 255.255.255.0
    next
end
```

4. Configure the import certificate and its CA certificate information.

   The certificate and its CA certificate must be imported on the remote peer FortiGate and on the primary FortiGate before configuring IPsec VPN tunnels. If the built-in *Fortinet_Factory* certificate and the *Fortinet_CA* CA certificate are used for authentication, you can skip this step.

```
config vpn certificate local
    edit "test1"
        ...
        set range global
    next
end
config vpn certificate ca
    edit "CA_Cert_1"
        ...
        set range global
    next
end
```

5. Configure the peer user.

   The peer user is used in the IPsec VPN tunnel peer setting to authenticate the remote peer FortiGate.

   - If not using the built-in *Fortinet_Factory* certificate and *Fortinet_CA* CA certificate, use the certificate the you imported (CA_Cert_1):

     ```
     config user peer
         edit "peer1"
             set ca "CA_Cert_1"
         next
     end
     ```

   - If the built-in *Fortinet_Factory* certificate and *Fortinet_CA* CA certificate are used for authentication, the peer user must be configured based on *Fortinet_CA*:

     ```
     config user peer
         edit "peer1"
             set ca "Fortinet_CA"
         next
     end
     ```

6. Configure the IPsec phase1-interface.

   ```
   config vpn ipsec phase1-interface
       edit "HQ2B2"
           set interface "port1"
           set ike-version 2
           set authmethod signature
           set net-device enable
           set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
           set dhgrp 20 21
           set remote-gw 198.51.100.5
           set certificate "test1"
           set peer "peer1"
       next
   end
   ```

7. Configure the IPsec phase2-interface.

   ```
   config vpn ipsec phase2-interface
       edit "HQ2B2"
           set phase1name "HQ2B2 "
           set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
           set dhgrp 20 21
           set auto-negotiate enable
           set src-addr-type name
           set dst-addr-type name
           set src-name "10.100.77.0"
           set dst-name "10.2.0.0"
       next
   end
   ```

8. Configure the static routes.

Two static routes are added to reach the remote protected subnet. The blackhole route is important to ensure that IPsec traffic does not match the default route when the IPsec tunnel is down.

```
config router static
    edit 2
        set dst 10.20.0.0 255.255.255.0
        set device "HQ2B2"
    next
    edit 3
        set dst 10.20.0.0 255.255.255.0
        set blackhole enable
        set distance 254
    next
end
```

9. Configure two firewall policies to allow bidirectional IPsec traffic flow over the IPsec VPN tunnel.

```
config firewall policy
    edit 1
        set name "inbound"
        set srcintf "HQ2B2"
        set dstintf "port2"
        set srcaddr "10.2.0.0"
        set dstaddr "10.100.77.0"
        set action accept
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set name "outbound"
        set srcintf "port2"
        set dstintf "HQ2B2"
        set srcaddr "10.100.77.0"
        set dstaddr "10.2.0.0"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end
```

To configure HQ2 for IPsec VPN authenticating a remote FortiGate peer with a digital certificate using the CLI:

1. Configure the WAN interface and default route.

```
config system interface
    edit "port1"
        set vdom "root"
        set ip 198.51.100.5 255.255.255.0
    next
end
config router static
```

```
    edit 1
        set gateway 198.51.100.2
        set device "port1"
    next
end
```

2. Configure address objects for internal protected networks.

```
config firewall address
    edit "10.2.0.0"
        set subnet 10.2.0.0 255.255.255.0
    next
end
config firewall address
    edit "10.100.77.0"
        set subnet 10.100.77.0 255.255.255.0
    next
end
```

3. Configure the internal (protected subnet) interface.

```
config system interface
    edit "port3"
        set vdom "root"
        set ip 10.2.0.0 255.255.255.0
    next
end
```

4. Configure the import certificate and its CA certificate information.

```
config vpn certificate local
    edit "test2"
        ...
        set range global
    next
end
config vpn certificate ca
    edit "CA_Cert_1"
        ...
        set range global
    next
end
```

5. Configure the peer user.
   - If not using the built-in *Fortinet_Factory* certificate and *Fortinet_CA* CA certificate, use the certificate the you imported (CA_Cert_1):

```
config user peer
    edit "peer2"
        set ca "CA_Cert_1"
    next
end
```

- If the built-in *Fortinet_Factory* certificate and *Fortinet_CA* CA certificate are used for authentication, the peer user must be configured based on *Fortinet_CA*:

```
config user peer
    edit "peer2"
        set ca "Fortinet_CA"
    next
end
```

6. Configure the IPsec phase1-interface.

```
config vpn ipsec phase1-interface
    edit "HQ2B2"
        set interface "port1"
        set ike-version 2
        set authmethod signature
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set dhgrp 20 21
        set remote-gw 198.51.100.5
        set certificate "test1"
        set peer "peer1"
    next
end
```

7. Configure the IPsec phase2-interface.

```
config vpn ipsec phase2-interface
    edit "B2toHQ"
        set phase1name "B2toHQ"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
        set auto-negotiate enable
        set src-addr-type name
        set dst-addr-type name
        set src-name "10.2.0.0"
        set dst-name "10.100.77.0"
    next
end
```

8. Configure the static routes.

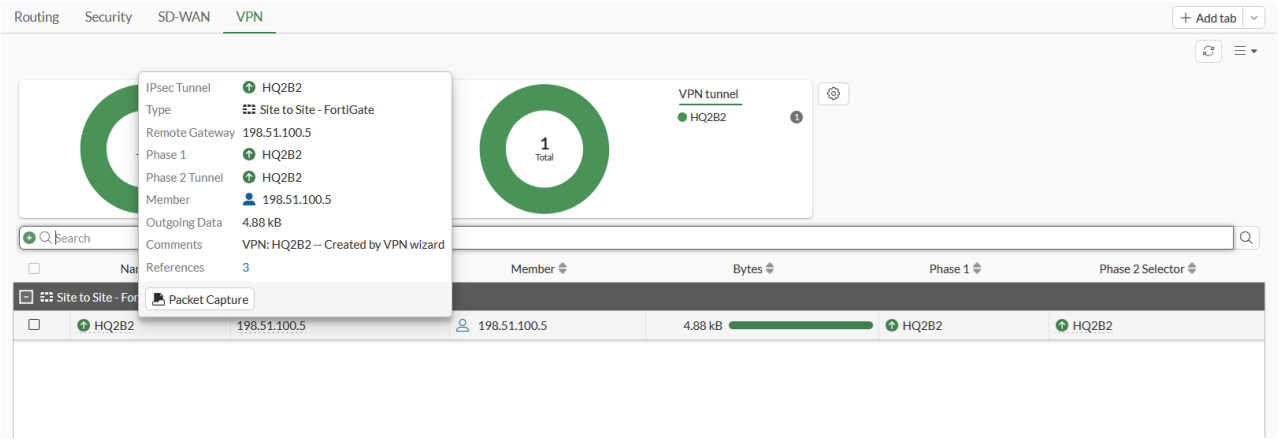
```
config router static
    edit 2
        set dst 10.100.77.0 255.255.255.0
        set device "B2toHQ"
    next
    edit 3
        set dst 10.100.77.0 255.255.255.0
        set blackhole enable
        set distance 254
    next
end
```

9. Configure two firewall policies to allow bidirectional IPsec traffic flow over the IPsec VPN tunnel.

```
config firewall policy
    edit 1
        set name "inbound"
        set srcintf "B2toHQ"
        set dstintf "port3"
        set srcaddr "10.100.77.0"
        set dstaddr "10.2.0.0"
        set action accept
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set name "outbound"
        set srcintf "port3"
        set dstintf "B2toHQ"
        set srcaddr "10.2.0.0"
        set dstaddr "10.100.77.0"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end
```

To verify the tunnel status in the GUI:

1. On FortiGate HQ1, go to *Dashboard* > *Network Monitor* > *VPN*.
2. Find the new tunnel, *HQ2B2* in this example, and check that both *Phase 1* and *Phase 2 Tunnel* are up.



3. Verify the tunnel status on HQ2 in the same way.

To verify the tunnel status in the CLI:

1. On FortiGate HQ1, run the following command:

```
# diagnose vpn ike gateway list name HQ2B2
vd: root/0
```

```
name: HQ2B2
version: 2
interface: port1 3
addr: 203.0.113.101:500 -> 198.51.100.5:500
tun_id: 198.51.100.5/::198.51.100.5
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 852804s ago
peer-id: 198.51.100.5
peer-id-auth: no
pending-queue: 0
PPK: no
IKE SA: created 1/11  established 1/11  time 0/820/9010 ms
IPsec SA: created 1/21  established 1/21  time 0/433/9010 ms

id/spi: 8895 45e69522bd784183/69be2940ae314677
direction: responder
status: established 77898-77898s ago = 0ms
proposal: aes128-sha256
child: yes
SK_ei: 07446ef38bd406fa-18e45bec886a2bd3
SK_er: b940f4338b27f369-7beaa308b7241a1b
SK_ai: 61eeb720d062c70b-e606340beab6f9d5-3a4da2966373ab15-e5766540cf6c7dd1
SK_ar: bbdd888b98dbda7b-669da7a726ce2148-3f08940f802f580c-52b6ef93f48672aa
message-id sent/recv: 0/1783
QKD: no
lifetime/rekey: 86400/8231
DPD sent/recv: 00000000/00000000
peer-id: 198.51.100.5
```

Note that HQ1 acted as the *responder*, and established the tunnel with remote peer 198.51.100.5 over UDP/500.

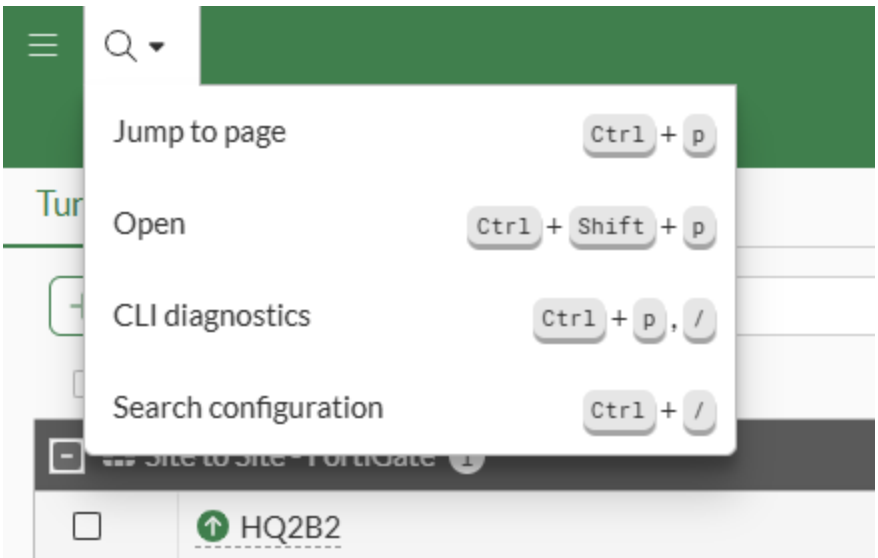
2. On FortiGate HQ2, run the following command:

```
# diagnose vpn ike gateway list name B2toHQ
vd: root/0
name: B2toHQ
version: 2
interface: port1 3
addr: 198.51.100.5:500 -> 203.0.113.101:500
tun_id: 203.0.113.101/::203.0.113.101
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 853181s ago
peer-id: 203.0.113.101
peer-id-auth: no
pending-queue: 0
PPK: no
IKE SA: created 1/11  established 1/11  time 0/5/10 ms
IPsec SA: created 1/21  established 1/21  time 0/0/10 ms
```

```
id/spi: 5788 45e69522bd784183/69be2940ae314677
direction: initiator
status: established 78281-78281s ago = 10ms
proposal: aes128-sha256
child: yes
SK_ei: 07446ef38bd406fa-18e45bec886a2bd3
SK_er: b940f4338b27f369-7beaa308b7241a1b
SK_ai: 61eeb720d062c70b-e606340beab6f9d5-3a4da2966373ab15-e5766540cf6c7dd1
SK_ar: bbdd888b98dbda7b-669da7a726ce2148-3f08940f802f580c-52b6ef93f48672aa
message-id sent/recv: 1792/0
QKD: no
lifetime/rekey: 86400/7818
DPD sent/recv: 00001ed7/00001ed7
peer-id: 203.0.113.101
```

Note that HQ2 acted as the *initiator*, and established the tunnel with remote peer 203.0.113.101 over UDP/500.

You can also use the command palette for CLI diagnostics. Open the command palette from the Find icon at the top of the page and then search for the command to run:



To verify traffic through the IPsec tunnel:

1. On a PC connected behind the port2 interface of HQ1, initiate ICMP traffic to the 10.2.0.0/24 network and verify that traffic goes through:

```
root@dmz-public-server:~# ping 10.2.0.7
PING 10.2.0.7 (10.2.0.7) 56(84) bytes of data.
64 bytes from 10.2.0.7: icmp_seq=1 ttl=62 time=9.84 ms
64 bytes from 10.2.0.7: icmp_seq=2 ttl=62 time=0.808 ms
```

2. Verify the traffic on HQ1:

```
# diagnose sniffer packet any 'host 10.2.0.7' 4 0 l
Using Original Sniffing Mode
interfaces=[any]
filters=[host 10.2.0.7]
2024-11-05 12:06:05.509027 port2 in 10.100.77.101 -> 10.2.0.7: icmp: echo request
2024-11-05 12:06:05.509074 HQ2B2 out 10.100.77.101 -> 10.2.0.7: icmp: echo request
2024-11-05 12:06:05.518607 HQ2B2 in 10.2.0.7 -> 10.100.77.101: icmp: echo reply
2024-11-05 12:06:05.518632 port2 out 10.2.0.7 -> 10.100.77.101: icmp: echo reply
```

   The ICMP request packets are received on port2 and routed to the HQ2B2 tunnel. ICMP reply packets are received on the HQ2B2 tunnel and routed back out port2.

3. Verify the traffic on HQ12:

```
# diagnose sniffer packet any 'host 10.100.77.101' 4 0 l
Using Original Sniffing Mode
interfaces=[any]
filters=[host 10.100.77.101]
2024-11-05 12:06:05.509711 B2toHQ in 10.100.77.101 -> 10.2.0.7: icmp: echo request
2024-11-05 12:06:05.509791 port3 out 10.100.77.101 -> 10.2.0.7: icmp: echo request
2024-11-05 12:06:05.518809 port3 in 10.2.0.7 -> 10.100.77.101: icmp: echo reply
2024-11-05 12:06:05.518899 B2toHQ out 10.2.0.7 -> 10.100.77.101: icmp: echo reply
```

   The ICMP request is received on the B2toHQ tunnel and routed to port3. The ICMP reply is received on port3 and routed back out the B2toHQ.

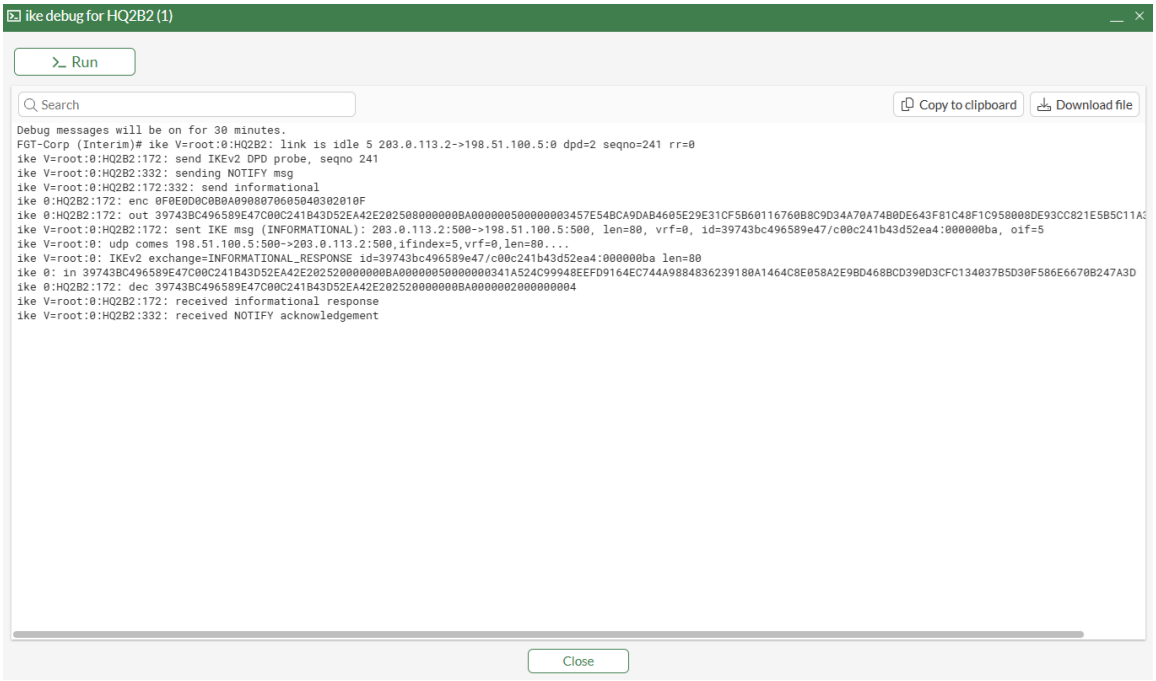
4. Optionally, run real time IKE debug using appropriate filters for troubleshooting if the IPsec tunnel does not come up:

```
# diagnose debug application ike -1
```

   If the remote FortiGate certificate cannot be validated, the following error is shown in the debug output:

```
ike 0: to_HQ2:15314: certificate validation failed
```

You can also quickly access the *CLI diagnostic* console to run the above command directly from the *VPN > VPN Tunnels > Tunnel* page. Select the Tunnel, click *More > CLI diagnostics*, and click the *Run* button to start collecting real-time debugs.



Site-to-site VPN with overlapping subnets

Overlapping subnets in IPsec occur when two or more networks involved in a VPN tunnel use the same or overlapping IP address ranges. This creates a conflict, as IPsec relies on unique network subnets to route traffic securely between them. Overlapping subnets often occur when two networks with the same IP ranges need to communicate, such as in mergers or acquisitions.

This example shows the configuration steps required to set up an IPsec VPN that enables communication between two overlapping networks located behind separate FortiGates using a route-based tunnel. Both source NAT (SNAT) and destination NAT (DNAT) are used to resolve IP conflicts.

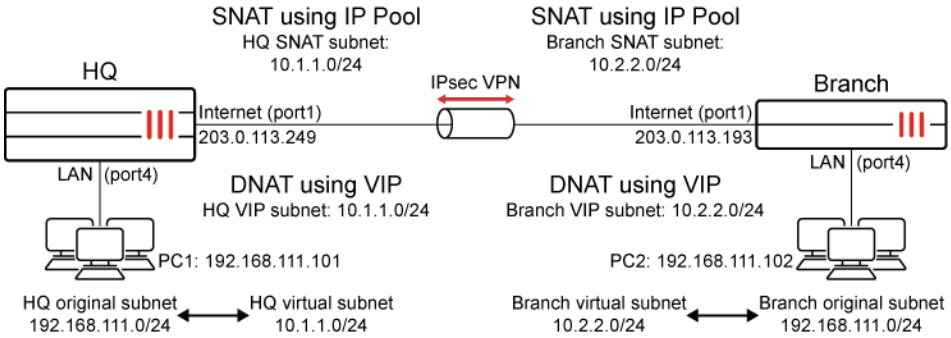
Both the HQ and Branch FortiGates use the same overlapping internal subnet, 192.168.111.0/24. To establish communication between these networks over the IPsec tunnel, address translation is required to distinguish between the identical subnets. Without SNAT and DNAT, routing traffic between these overlapping networks would lead to conflicts, as networking devices cannot differentiate between identical IP addresses.

To resolve this issue, a unique and non-overlapping subnet is used on both FortiGates for address translation during transit.

- On the HQ FortiGate, the IP addresses of the HQ original subnet are logically one-to-one mapped to IP address in the HQ virtual subnet of 10.1.1.0/24.
- On the Branch FortiGate, the IP addresses of the Branch original subnet are logically one-to-one mapped to IP address in the Branch virtual subnet of 10.2.2.0/24.

The virtual subnets must be the same size as the overlapping subnet to cover the entire IP address range in the subnet and avoid network issues. The virtual subnets are used for SNAT using IP Pools and DNAT using virtual

IP (VIP) configurations, ensuring that traffic originating from the overlapped subnets is translated according to the respective virtual subnets for communication.



After the configuration is complete, users that are in the HQ original subnet (192.168.111.0/24) are required to use the Branch virtual subnet (10.2.2.0/24) to communicate with users that are behind the Branch. Users that are in the Branch original subnet (192.168.111.0/24) are required to use the HQ virtual subnet (10.1.1.0/24) to communicate with users that are behind the HQ.

Source	Destination	Used subnet
HQ	Branch	Branch virtual subnet (10.2.2.0/24)
Branch	HQ	HQ virtual subnet (10.1.1.0/24)

This ensures seamless communication across the IPsec tunnel while resolving the challenges posed by overlapping subnets.

HQ FortiGate configuration

To configure IPsec VPN:

1. Go to *VPN* > *VPN Tunnels* and click on *Create New* > *Custom IPsec tunnel*.
2. Set the *Name*, such as *HQtoBranchVPN*.
3. Configure the *Network* settings:

Field	Value
IP version	*IPv4*
Remote gateway	*Static IP address*
IP address	*203.0.113.193*
Interface	*Internet (port1)*
Transport	If the option is grayed out and set to *UDP*, this means that *Allow VPN negotiation over TCP* has not been enabled. If the *Transport* can be selected, select *Auto*.

4. Leave the remaining *Network* settings as their default values.
5. Configure the *Authentication* settings:

Field	Value
Method	*Pre-shared Key*
Pre-shared key	Enter a suitable key. The same pre-shared key must be configured on the Branch FortiGate. For better security, use a digital certificate.
IKE	Version 2

6. Leave the remaining *Authentication* settings as their default values.
7. Leave the *Phase 1 proposal* settings as their default values.
8. Configure the *Phase 2 selectors* settings:
   a. In the table, click *Create New*.
   b. Configure the new phase 2 selector:

Field	Value
Name	*HQtoBranchVPN*
Local address	*Subnet Address* *10.1.1.0/24*
Remote address	*Subnet Address* *10.2.2.0/24*
Auto-negotiate	(Optional) *Enable*

Only traffic matching the subnets specified in the *Local address* and *Remote address* fields in the Phase 2 configuration can pass through the IPsec tunnel.

- Outgoing traffic exiting through the IPsec tunnel is first matched against a firewall policy, then Source NAT (if configured) is applied, and finally, is checked against the traffic selectors in the IPsec tunnel settings.
- Incoming traffic on IPsec tunnel is first evaluated against the traffic selectors, then matched to a firewall policy, and finally, DNAT (if configured) is applied.

   c. Leave the remaining settings as their default values and click *OK*.
9. Click *OK*.

To configure static routes:

1. Go to *Network* > *Static Routes* and click *Create New*.
2. Configure the static route for the Branch virtual subnet:

Field	Value
Destination	*Subnet* *10.2.2.0/24*
Interface	*HQtoBranchVPN* (the IPsec tunnel interface)

Field	Value
Administrative Distance	*10*
Status	*Enabled*

3. Click *OK*.
4. Create a second static route with the same *Destination* that uses the blackhole interface.
   This is a best practice for route-based IPsec VPN tunnels because it ensures that traffic for the remote FortiGate's subnet is not sent using the default route in the event that the IPsec tunnel goes down and is instead sent to the blackhole.

Field	Value
Destination	*Subnet* *10.2.2.0/24*
Interface	*Blackhole*
Administrative Distance	*254*
Status	*Enable*

To configure the address objects:

1. Go to *Policy & Objects* > *Addresses* and select *Address*.
2. Click *Create new* and configure an address object for HQ original subnet:

Field	Value
Name	*HQ original subnet*
Interface	*Any*
Type	*Subnet*
IP/Netmask	*192.168.111.0/24*
Routing configuration	Disable

3. Click *OK*.
4. Click *Create new* again and configure an address object for Branch virtual subnet:

Field	Value
Name	*Branch virtual subnet*
Interface	*Any*
Type	*Subnet*
IP/Netmask	*10.2.2.0/24*
Routing configuration	Disable

5. Click *OK*

To configure the IP pool for SNAT:

1. Go to *Policy & Objects* > *IP Pools* and click *Create new*.
2. Configure the dynamic IP pool:

Field	Value
Name	*HQ SNAT subnet*
Type	*Fixed Port Range*
External IP Range	*10.1.1.1-10.1.1.254*
Internal IP Range	*192.168.111.1-192.168.111.254*

3. Leave the remaining settings as their default values and click *OK*.

To configure the VIP for DNAT:

1. Go to *Policy & Objects* > *Virtual IPs* > *Virtual IP* and click *Create new*.
2. Configure the virtual IP:

Field	Value
Name	*HQ VIP subnet*
Type	*Static NAT*
External IP address/range	*10.1.1.1-10.1.1.254*
Map to IPv4 address/range	*192.168.111.1*

3. Click *OK*.

To configure the firewall policies:

1. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
2. Configure the policy to allow traffic to flow from the HQ original subnet to the Branch virtual subnet and to perform SNAT:

Field	Value
Name	*HQ to Branch*
Incoming interface	*LAN (port4)*
Outgoing interface	*HQtoBranchVPN*
Source	*HQ original subnet*
Destination	*Branch virtual subnet*
Service	*ALL*
NAT	Enable
IP pool configuration	*Use Dynamic IP Pool*

Field	Value
	HQ SNAT subnet IP

3. Configure the remaining settings as needed, then click *OK*.
4. Click *Create New* again and configure the policy to allow traffic to flow from the Branch virtual subnet to the HQ original subnet and to perform DNAT:

Field	Value
Name	*Branch to HQ*
Incoming interface	*HQtoBranchVPN*
Outgoing interface	*LAN (port4)*
Source	*Branch virtual subnet*
Destination	*HQ VIP subnet* (the virtual IP/server to perform DNAT)
Service	*ALL*
NAT	Disable

5. Configure the remaining settings as needed, then click *OK*.

Branch FortiGate configuration

To configure IPsec VPN:

1. Go to *VPN* > *VPN Tunnels* and click on *Create New* > *Custom IPsec tunnel*.
2. Set the *Name*, such as *BranchtoHQVPN*.
3. Configure the *Network* settings:

Field	Value
IP version	*IPv4*
Remote gateway	*Static IP address*
IP address	*203.0.113.249*
Interface	*Internet (port1)*
Transport	If the option is grayed out and set to *UDP*, this means that *Allow VPN negotiation over TCP* has not been enabled. If the *Transport* can be selected, select *Auto*.

4. Leave the remaining *Network* settings as their default values.
5. Configure the *Authentication* settings:

Field	Value
Method	*Pre-shared Key*
Pre-shared key	Enter a suitable key.

Field	Value
	The same pre-shared key must be configured on the HQ FortiGate. For better security, use a digital certificate.
IKE	Version 2

6. Leave the remaining *Authentication* settings as their default values.
7. Leave the *Phase 1 proposal* settings as their default values.
8. Configure the *Phase 2 selectors* settings:
   a. In the table, click *Create New*.
   b. Configure the new phase 2 selector:

Field	Value
Name	*BranchtoHQVPN*
Local address	*Subnet Address* *10.2.2.0/24*
Remote address	*Subnet Address* *10.1.1.0/24*
Auto-negotiate	(Optional) *Enable*

Only traffic matching the subnets specified in the *Local address* and *Remote address* fields in the Phase 2 configuration can pass through the IPsec tunnel.

- Outgoing traffic exiting through the IPsec tunnel is first matched against a firewall policy, then Source NAT (if configured) is applied, and finally, is checked against the traffic selectors in the IPsec tunnel settings.
- Incoming traffic on IPsec tunnel is first evaluated against the traffic selectors, then matched to a firewall policy, and finally, DNAT (if configured) is applied.

   c. Leave the remaining settings as their default values and click *OK*.
9. Click *OK*.

To configure static routes:

1. Go to *Network* > *Static Routes* and click *Create New*.
2. Configure the static route for the Branch virtual subnet:

Field	Value
Destination	*Subnet* *10.1.1.0/24*
Interface	*BranchtoHQVPN* (the IPsec tunnel interface)
Administrative Distance	*10*
Status	*Enabled*

3. Click *OK*.
4. Create a second static route with the same *Destination* that uses the blackhole interface.
   This is a best practice for route-based IPsec VPN tunnels because it ensures that traffic for the remote FortiGate's subnet is not sent using the default route in the event that the IPsec tunnel goes down and is instead sent to the blackhole.

Field	Value
Destination	*Subnet* *10.1.1.0/24*
Interface	*Blackhole*
Administrative Distance	*254*
Status	*Enable*

To configure the address objects:

1. Go to *Policy & Objects* > *Addresses* > *Address*.
2. Click *Create new* and configure an address object for HQ original subnet:

Field	Value
Name	*Branch original subnet*
Interface	*Any*
Type	*Subnet*
IP/Netmask	*192.168.111.0/24*
Routing configuration	Disable

3. Click *OK*.
4. Click *Create new* again and configure an address object for Branch virtual subnet:

Field	Value
Name	*HQ virtual subnet*
Interface	*Any*
Type	*Subnet*
IP/Netmask	*10.1.1.0/24*
Routing configuration	Disable

5. Click *OK*

To configure the IP pool for SNAT:

1. Go to *Policy & Objects* > *IP Pools* and click *Create new*.
2. Configure the dynamic IP pool:

Field	Value
Name	*Branch SNAT subnet*
Type	*Fixed Port Range*
External IP Range	*10.2.2.1-10.2.2.254*
Internal IP Range	*192.168.111.1-192.168.111.254*

3. Leave the remaining settings as their default values and click *OK*.

To configure the VIP for DNAT:

1. Go to *Policy & Objects* > *Virtual IPs* > *Virtual IP* and click *Create New*.
2. Configure the virtual IP:

Field	Value
Name	*Branch VIP subnet*
Type	*Static NAT*
External IP address/range	*10.2.2.1-10.2.2.254*
Map to IPv4 address/range	*192.168.111.1*

3. Click *OK*.

To configure the firewall policies:

1. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
2. Configure the policy to allow traffic to flow from the HQ original subnet to the Branch virtual subnet and to perform SNAT:

Field	Value
Name	*Branch to HQ*
Incoming interface	*LAN (port4)*
Outgoing interface	*BranchtoHQVPN*
Source	*Branch original subnet*
Destination	*HQ virtual subnet*
Service	*ALL*
NAT	Enable
IP pool configuration	*Use Dynamic IP Pool* *Branch SNAT subnet IP*

3. Configure the remaining settings as needed, then click *OK*.
4. Click *Create New* again and configure the policy to allow traffic to flow from the Branch virtual subnet to the HQ original subnet and to perform DNAT:

Field	Value
Name	*HQ to Branch*
Incoming interface	*BranchtoHQVPN*
Outgoing interface	*LAN (port4)*
Source	*HQ virtual subnet*
Destination	*Branch VIP subnet* (the virtual IP/server to perform DNAT)
Service	*ALL*
NAT	Disable

5. Configure the remaining settings as needed, then click *OK*.

Verification

To verify the status of IPsec tunnel in the GUI:

1. On either FortiGate, go to *Dashboard* > *Network Monitor* > *VPN*.
2. The tunnels should be up on both FortiGates. If you did not enable *Auto-negotiate* in the IPsec VPN settings, you may have to select the tunnel and click *Bring Up*.

To verify the status of IPsec tunnel in the CLI:

1. Verify the status of Phase 1:

```
Branch # diagnose vpn ike gateway list name BranchtoHQVPN
vd: root/0
name: BranchtoHQVPN
version: 2
interface: port1 3
addr: 203.0.113.193:4500 -> 203.0.113.249:4500
tun_id: 203.0.113.249/::203.0.113.249
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 58s ago
peer-id: 203.0.113.249
peer-id-auth: no
nat: me peer
pending-queue: 0
PPK: no
IKE SA: created 1/1  established 1/1  time 10/10/10 ms
IPsec SA: created 1/1  established 1/1  time 10/10/10 ms

  id/spi: 5 8d6f155741b90831/54940957144cdd82
  direction: initiator
  ...
  peer-id: 203.0.113.249
```

2. Verify the status of Phase 2:

```
Branch # diagnose vpn tunnel list name BranchtoHQVPN
list ipsec tunnel by names in vd 0
------------------------------------------------------
name=BranchtoHQVPN ver=2 serial=1 203.0.113.193:4500->203.0.113.249:4500 nexthop=203.0.113.1
tun_id=203.0.113.249 tun_id6=::203.0.113.249 status=up dst_mtu=1200 weight=1
bound_if=3 real_if=3 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-
rfc  run_state=0 role=primary accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=6 ilast=8103 olast=8103 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-demand on=1 status=ok idle=20000ms retry=3 count=0 seqno=0
natt: mode=keepalive draft=0 interval=10 remote_port=4500
fec: egress=0 ingress=0
proxyid=BranchtoHQVPN proto=0 sa=1 ref=2 serial=2 auto-negotiate
  src: 0:10.2.2.0-10.2.2.255:0
  dst: 0:10.1.1.0-10.1.1.255:0
  SA:  ref=3 options=38203 type=00 soft=0 mtu=1134 expire=42550/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42899/43200
  dec: spi=617afb8e esp=aes key=16 c930949dfb07586bf59621f7371bcf6d
       ah=sha1 key=20 53ef18c8d261ba13ca6faa130f1bcbb3eb9987c8
  enc: spi=03ec3c88 esp=aes key=16 88f4d37c477b5198c75ecfe19b3053c0
       ah=sha1 key=20 5bbaa833a7e099dc49557877f0023267db2f08ac
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=203.0.113.249 npu_lgwy=203.0.113.193 npu_selid=2 dec_npuid=0 enc_
npuid=0
```

Note that the protected subnet in the IPsec tunnel are the virtual subnets for Branch and HQ, respectively.

To verify the traffic flow from PC1 to PC2:

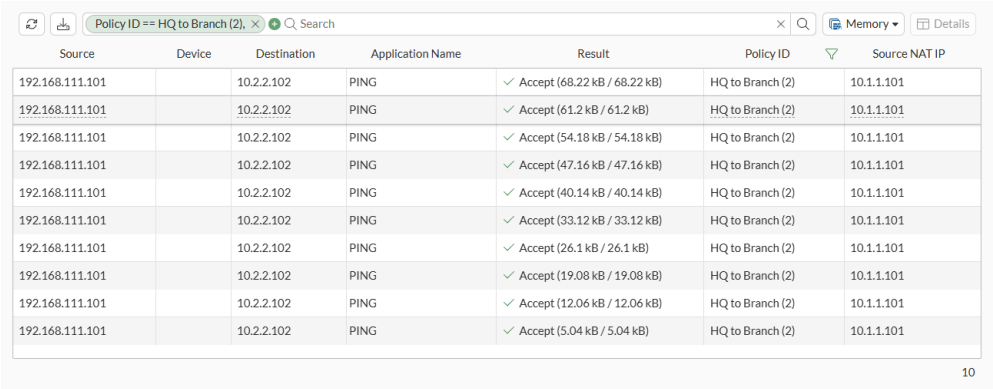
1. From a PC in the HQ network (PC1, 192.168.111.101), ping a PC in the Branch network (PC2, 192.168.111.102) using the virtual IP of PC2, 10.2.2.102.

   The ping should be successful.

```
C:\>ping 10.2.2.102

Pinging 10.2.2.102 with 32 bytes of data:
Reply from 10.2.2.102: bytes=32 times=10ms TTL=126
Reply from 10.2.2.102: bytes=32 times=3ms TTL=126
Reply from 10.2.2.102: bytes=32 times=3ms TTL=126
Reply from 10.2.2.102: bytes=32 times=3ms TTL=126

Ping statistics for 10.2.2.102:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 3ms, Maximum = 10ms, Average = 4ms
```

To verify the traffic flow from PC1 to PC2 in the GUI:

1. On either FortiGate, go to *Log & Report* > *Forward Traffic*.
2. Filter the content using the search bar and add the required columns to see the logs.



Source	Device	Destination	Application Name	Result	Policy ID	Source NAT IP
192.168.111.101		10.2.2.102	PING	Accept (68.22 kB / 68.22 kB)	HQ to Branch (2)	10.1.1.101
192.168.111.101		10.2.2.102	PING	Accept (61.2 kB / 61.2 kB)	HQ to Branch (2)	10.1.1.101
192.168.111.101		10.2.2.102	PING	Accept (54.18 kB / 54.18 kB)	HQ to Branch (2)	10.1.1.101
192.168.111.101		10.2.2.102	PING	Accept (47.16 kB / 47.16 kB)	HQ to Branch (2)	10.1.1.101
192.168.111.101		10.2.2.102	PING	Accept (40.14 kB / 40.14 kB)	HQ to Branch (2)	10.1.1.101
192.168.111.101		10.2.2.102	PING	Accept (33.12 kB / 33.12 kB)	HQ to Branch (2)	10.1.1.101
192.168.111.101		10.2.2.102	PING	Accept (26.1 kB / 26.1 kB)	HQ to Branch (2)	10.1.1.101
192.168.111.101		10.2.2.102	PING	Accept (19.08 kB / 19.08 kB)	HQ to Branch (2)	10.1.1.101
192.168.111.101		10.2.2.102	PING	Accept (12.06 kB / 12.06 kB)	HQ to Branch (2)	10.1.1.101
192.168.111.101		10.2.2.102	PING	Accept (5.04 kB / 5.04 kB)	HQ to Branch (2)	10.1.1.101

10

To make sure that the logs are collected and shown under *Forward Traffic*, set *Log Allowed Traffic* to *All Sessions* in the corresponding firewall policy configured under *Policy & Objects* > *Firewall Policy*.

To verify the traffic flow from PC1 to PC2 in the CLI:

1. Verify the SNAT operation for traffic flowing from PC1 to PC2 on HQ:
   a. Using sniffer trace:

```
HQ # diagnose sniffer packet any 'icmp' 4 10
Using Original Sniffing Mode
interfaces=[any]
filters=[icmp]
6.141932 port4 in 192.168.111.101 -> 10.2.2.102: icmp: echo request
6.144532 HQtoBranchVPN out 10.1.1.101 -> 10.2.2.102: icmp: echo request
6.149731 HQtoBranchVPN in 10.2.2.102 -> 10.1.1.101: icmp: echo reply
6.149770 port4 out 10.2.2.102 -> 192.168.111.101: icmp: echo reply
```

   b. Using session table:

```
HQ # diagnose sys session filter proto 1
HQ # diagnose sys session list
session info: proto=1 proto_state=00 duration=1 expire=58 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ tun_id=203.0.113.193/0.0.0.0 vlan_cos=0/255
state=may_dirty
statistic(bytes/packets/allow_err): org=60/1/1 reply=60/1/1 tuples=2
tx speed(Bps/kbps): 31/0 rx speed(Bps/kbps): 31/0
orgin->sink: org pre->post, reply pre->post dev=6->16/16->6 gwy=203.0.113.193/0.0.0.0
hook=post dir=org act=snat 192.168.111.101:1->10.2.2.102:8(10.1.1.101:5437)
hook=pre dir=reply act=dnat 10.2.2.102:5437->10.1.1.101:0(192.168.111.101:1)
```

```
misc=0 policy_id=2 pol_uuid_idx=15849 auth_info=0 chk_client_info=0 vd=0
serial=00008136 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000100
no_ofld_reason:  npu-flag-off
total session: 1
```

2. Verify the DNAT operation for traffic flowing from PC1 to PC2 on Branch:
   a. Using sniffer trace:

```
Branch # diagnose sniffer packet any 'icmp' 4 10
Using Original Sniffing Mode
interfaces=[any]
filters=[icmp]
5.738981 BranchtoHQVPN in 10.1.1.101 -> 10.2.2.102: icmp: echo request
5.739058 port4 out 10.1.1.101 -> 192.168.111.102: icmp: echo request
5.740026 port4 in 192.168.111.102 -> 10.1.1.101: icmp: echo reply
5.740068 BranchtoHQVPN out 10.2.2.102 -> 10.1.1.101: icmp: echo reply
```

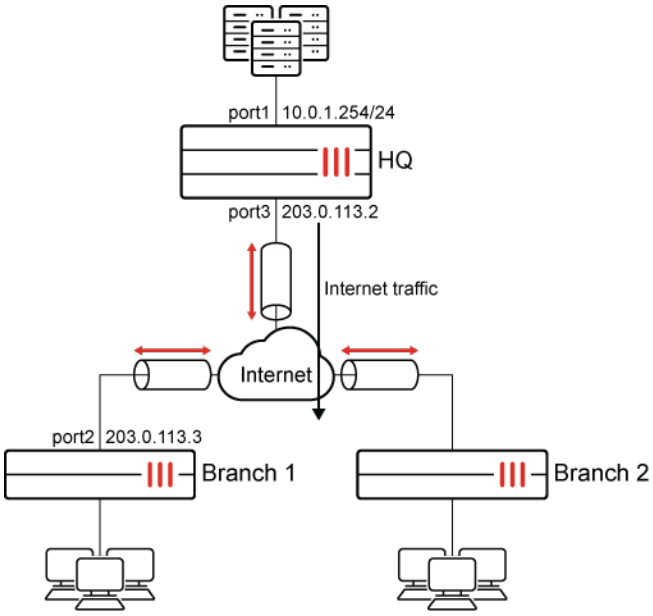
   b. Using session table:

```
Branch # diagnose sys session filter proto 1
Branch # diagnose sys session list
session info: proto=1 proto_state=00 duration=3 expire=56 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ tun_id=0.0.0.0/203.0.113.249 vlan_cos=0/255
state=may_dirty
statistic(bytes/packets/allow_err): org=60/1/1 reply=60/1/1 tuples=2
tx speed(Bps/kbps): 15/0 rx speed(Bps/kbps): 15/0
orgin->sink: org pre->post, reply pre->post dev=16->6/6->16 gwy=0.0.0.0/203.0.113.249
hook=pre dir=org act=dnat 10.1.1.101:5437->10.2.2.102:8(192.168.111.102:5437)
hook=post dir=reply act=snat 192.168.111.102:5437->10.1.1.101:0(10.2.2.102:5437)
misc=0 policy_id=3 pol_uuid_idx=15862 auth_info=0 chk_client_info=0 vd=0
serial=00005b94 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000100
no_ofld_reason:  npu-flag-off
total session: 1
```

For more information on running sniffer trace, see Performing a sniffer trace or packet capture on page 4966.
For more information about using session table, see Using a session table on page 4979.

Tunneled Internet browsing

This is a sample configuration of tunneled internet browsing using a dialup VPN. To centralize network management and control, all branch office traffic is tunneled to HQ, including Internet browsing.



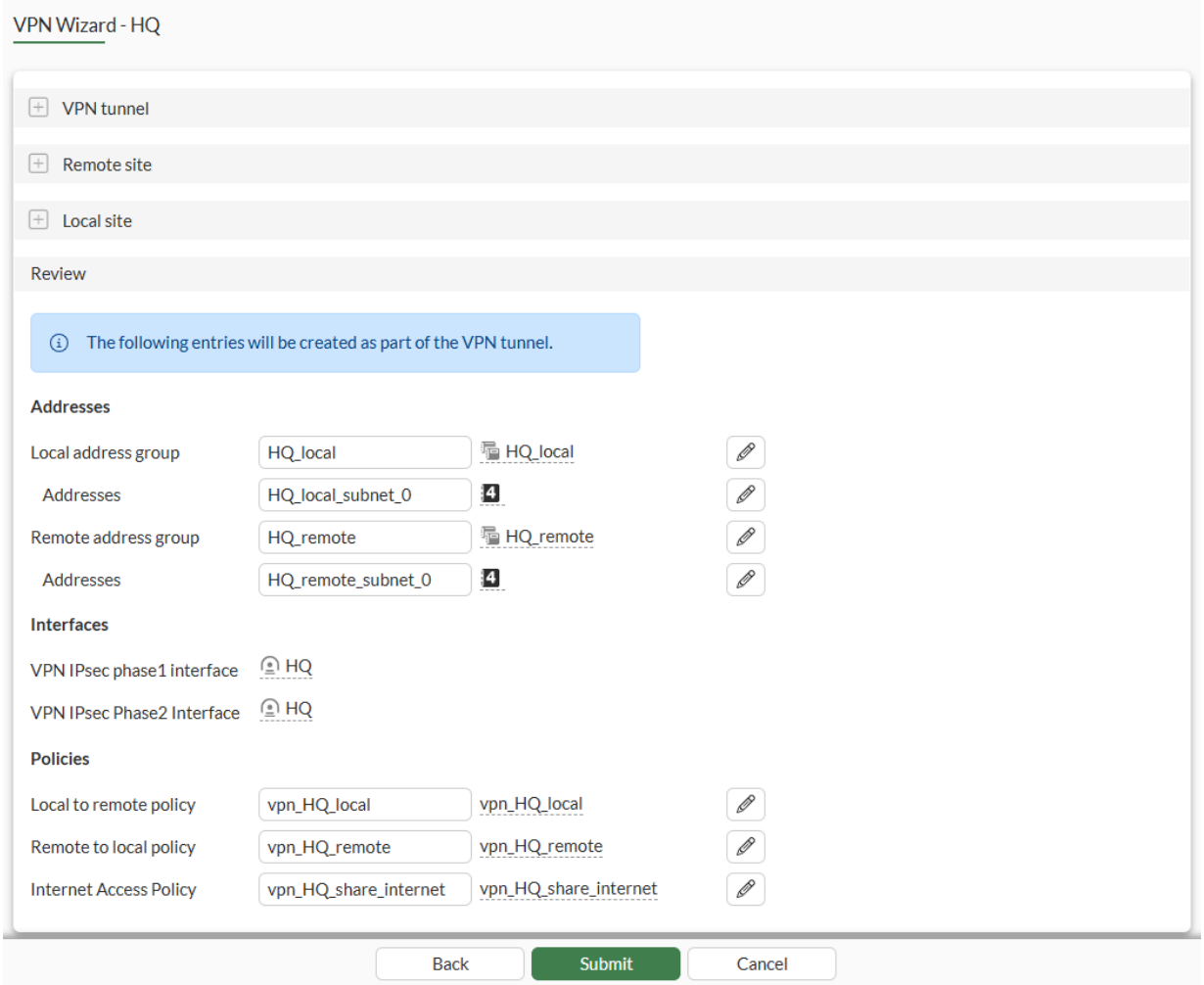
To configure a dialup VPN to tunnel Internet browsing using the GUI:

1. Configure the dialup VPN server FortiGate at HQ:
   a. Go to *VPN > VPN Wizard* and configure the following settings for *VPN Setup*:
      i. Enter a VPN name, in this example, *HQ*.
      ii. For *Template Type*, select *Site to Site*.
      iii. Click *Begin*.
   b. Configure the VPN tunnel:
      i. For *Authentication Method*, select *Pre-shared Key*.
      ii. In the *Pre-shared Key* field, enter your key.
      iii. For *IKE*, select *Version 2*.
      iv. Keep *Transport* set to *UDP*. If TCP is required, enable *Allow VPN negotiation over TCP* in *VPN > VPN Tunnels > Settings* before using the VPN wizard.
      v. Click *Next*.
   c. Configure the *Remote Site*:
      i. For *Remote site device type*, select *FortiGate*.
      ii. For *Remote site device*, select *Behind NAT or dynamic*.
      iii. For *Remote site subnets that can access VPN*, depending on whether you need to allow each branch individually, entire their subnets. Use 0.0.0.0/0 if you need to allow any subnets.
      iv. Click *Next*.
   d. Configure the *Local Site*:
      i. For *Outgoing interface that binds to tunnel*, select *port3*.
      ii. Disable *Create and add interface to zone*.
      iii. For *Local Interface*, select *port1*.
      iv. The *Local subnets that can access VPN* field should automatically fill with the local subnet on port1.
      v. Enable *Allow remote site's internet traffic through this device*.

vi. Set *Shared WAN* to *port3*. This is the interface that will provide internet access. If this is an interface other than the interface used for your VPN tunnel, specify it here.

vii. Click *Next*.

e. Review the configuration and click *Submit*.



This effectively creates a VPN dialup gateway on the HQ, which allows tunneled traffic to access the internet through the internet access policy *vpn_HQ_share_Internet*.

2. Configure the dialup VPN client FortiGate on Branch 1:

a. Go to *VPN* > *VPN Wizard* and configure the following settings for *VPN Setup*:

i. Enter a VPN name, in this example, *Branch1*.

ii. For *Template Type*, select *Site to Site*.

iii. Click *Begin*.

b. Configure the VPN tunnel:

i. For *Authentication Method*, select *Pre-shared Key*.

ii. In the *Pre-shared Key* field, enter *sample* as the key.

iii. For *IKE*, select *Version 2*.

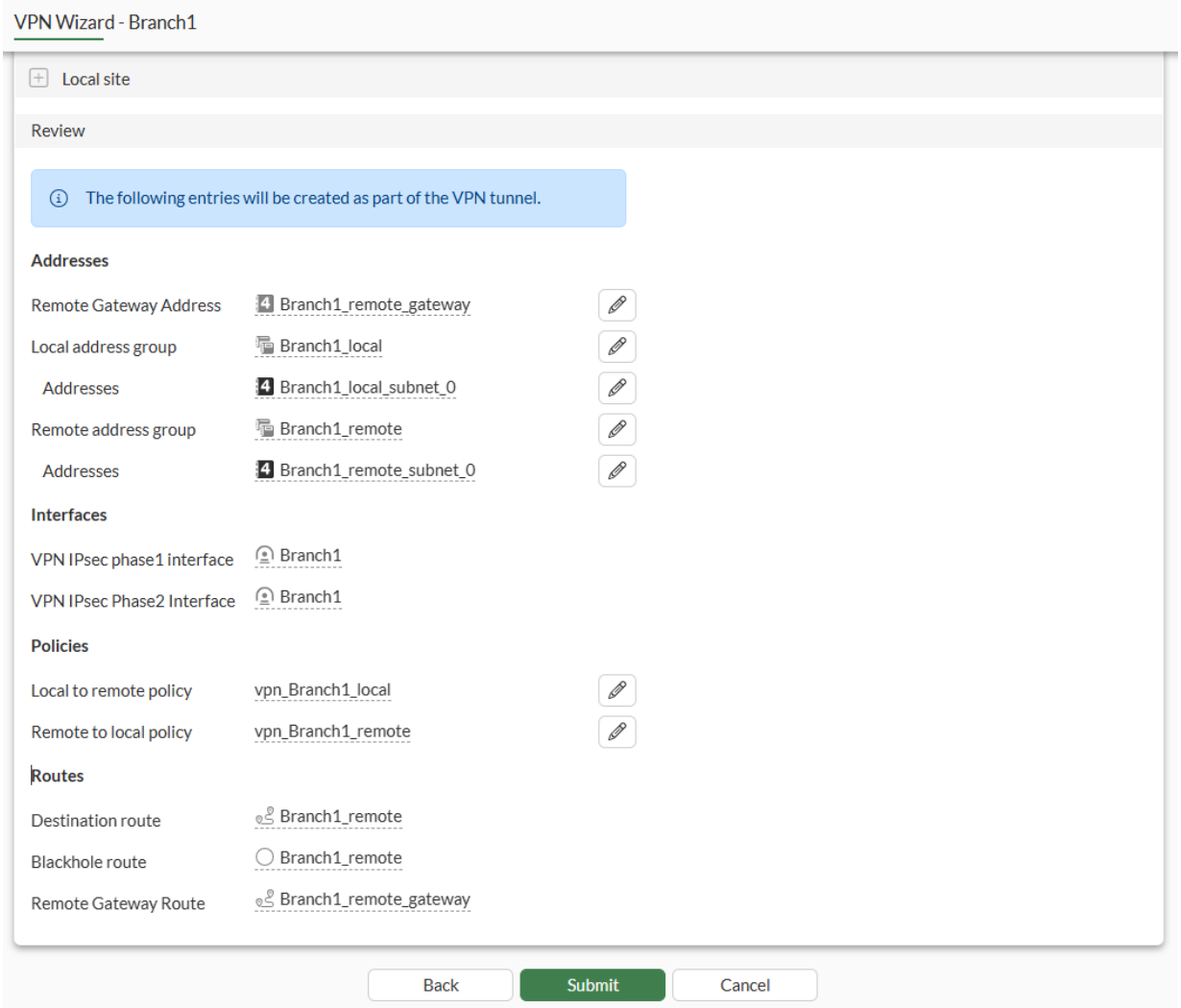
iv. Keep *Transport* set to *UDP*. If TCP is required, enable *Allow VPN negotiation over TCP* in *VPN* > *VPN Tunnels* > *Settings* before using the VPN wizard.

v. Click *Next*.

c. Configure the *Remote Site*:
   i. For *Remote site device type*, select *FortiGate*.
   ii. For *Remote site device*, select *Accessible and static*.
   iii. Set *IP* to *203.0.113.2*.
   iv. Enable *Route this device's internet traffic through the remote site*.
   v. Click *Next*.

d. Configure the *Local Site*:
   i. For *Outgoing interface that binds to tunnel*, select *port2*.
   ii. Disable *Create and add interface to zone*.
   iii. For *Local Interface*, select *port3*.
   iv. The *Local subnets that can access VPN* field should automatically fill with the local subnet on port1.
   v. For *Local Gateway*, specify the next hop of the outgoing interface.
   vi. Click *Next*.

e. Review the configuration and click *Submit*.



3. Configure the dialup VPN client FortiGate similarly on other branches.

To configure a dialup VPN to tunnel Internet browsing using the CLI:

1. Configure the WAN interface and static route on the FortiGate at HQ:

```
config system interface
    edit "port1"
        set ip 10.0.1.254 255.255.255.0
        set alias "Clients_LAN"
    next
    edit "port3"
        set ip 203.0.113.2 255.255.255.0
        set alias "WAN"
        set role wan
    next
end
config router static
    edit 1
        set gateway 203.0.113.254
        set device "port3"
    next
end
```

2. Configure IPsec phase1-interface and phase2-interface configuration at HQ:

```
config vpn ipsec phase1-interface
    edit "HQ"
        set type dynamic
        set interface "port3"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set dhgrp 20 21
        set psksecret <secret>
    next
end
config vpn ipsec phase2-interface
    edit "HQ"
        set phase1name "HQ"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
        set src-addr-type name
        set dst-addr-type name
        set src-name "all"
        set dst-name "HQ_remote"
    next
end
```

3. Configure the firewall policies at HQ:

```
config firewall policy
    edit 19
```

```
        set name "vpn_HQ_local"
        set srcintf "port1"
        set dstintf "HQ"
        set action accept
        set srcaddr "HQ_local"
        set dstaddr "HQ_remote"
        set schedule "always"
        set service "ALL"
    next
    edit 20
        set name "vpn_HQ_remote"
        set srcintf "HQ"
        set dstintf "port1"
        set action accept
        set srcaddr "HQ_remote"
        set dstaddr "HQ_local"
        set schedule "always"
        set service "ALL"
    next
    edit 21
        set name "vpn_HQ_share_internet"
        set srcintf "HQ"
        set dstintf "port3"
        set action accept
        set srcaddr "HQ_remote"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

4. Configure the WAN interface and static route on Branch 1:

```
config system interface
    edit "port2"
        set ip 203.0.113.3 255.255.255.0
    next
    edit "port3"
        set ip 172.16.7.8 255.255.255.0
    next
end
config router static
    edit 1
        set gateway 203.0.113.254
        set device "wan1"
    next
end
```

5. Configure the address objects on Branch 1:

```
config firewall address
    edit "Branch1_local_subnet_0"
        set allow-routing enable
        set subnet 172.16.7.0 255.255.255.0
    next
    edit "Branch1_remote_subnet_0"
        set allow-routing enable
        set subnet 0.0.0.0 0.0.0.0
    next
end
config firewall addrgrp
    edit "Branch1_local"
        set allow-routing enable
        set member "Branch1_local_subnet_0"
    next
    edit "Branch1_remote"
        set allow-routing enable
        set member "Branch1_remote_subnet_0"
    next
end
```

6. Configure IPsec phase1-interface and phase2-interface configuration on Branch 1:

```
config vpn ipsec phase1-interface
    edit "Branch1"
        set interface "port2"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set dhgrp 20 21
        set wizard-type static-fortigate
        set transport udp
        set remote-gw 203.0.113.2
        set psksecret <secret>
    next
end
config vpn ipsec phase2-interface
    edit "Branch1"
        set phase1name "Branch1"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
        set dhgrp 20 21
        set src-addr-type name
        set dst-addr-type name
        set src-name "Branch1_local"
        set dst-name "Branch1_remote"
    next
end
```

7. Configure the firewall policies on Branch 1:

```
config firewall policy
    edit 5
        set name "vpn_Branch1_local"
        set srcintf "port3"
        set dstintf "Branch1"
        set action accept
        set srcaddr "Branch1_local"
        set dstaddr "Branch1_remote"
        set schedule "always"
        set service "ALL"
    next
    edit 6
        set name "vpn_Branch1_remote"
        set srcintf "Branch1"
        set dstintf "port3"
        set action accept
        set srcaddr "Branch1_remote"
        set dstaddr "Branch1_local"
        set schedule "always"
        set service "ALL"
    next
end
```

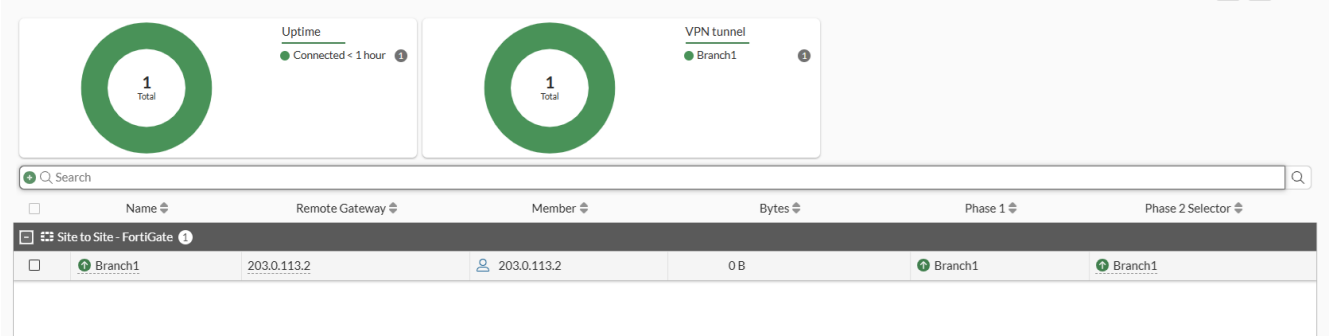
8. Configure the static routes to route traffic to the tunnel and the gateway on Branch 1:

```
config router static
    edit 4
        set distance 1
        set device "Branch1"
        set dstaddr "Branch1_remote"
    next
    edit 5
        set distance 254
        set blackhole enable
        set dstaddr "Branch1_remote"
        set vrf 0
    next
    edit 6
        set gateway 203.0.113.254
        set distance 1
        set device "port2"
        set dstaddr "Branch1_remote_gateway"
    next
end
```

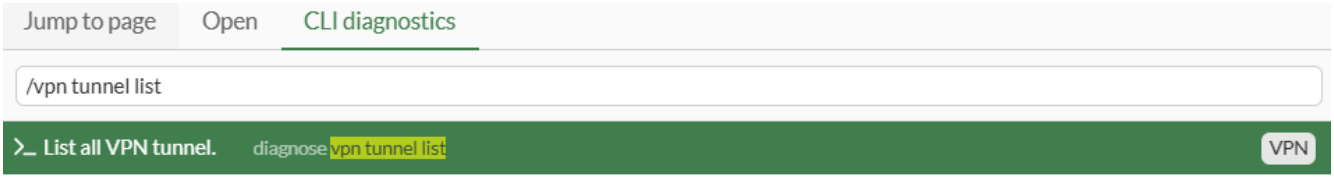
Verifying the tunnel connectivity

After the branch's tunnel has been created, it should come up automatically.

On the branch, go to the VPN monitor to see the tunnel status.



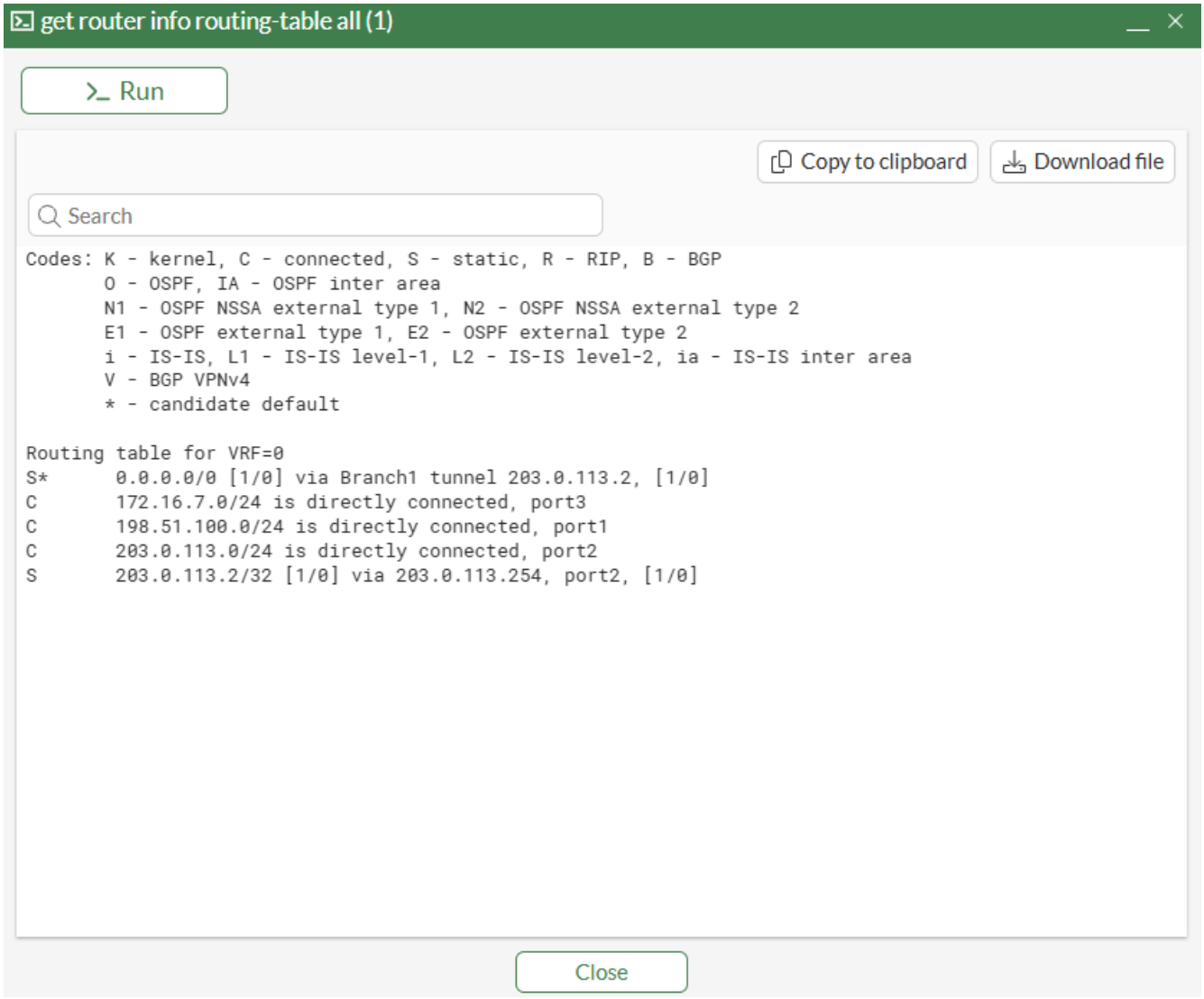
Optionally, view the VPN tunnel list on a branch with the `diagnose vpn tunnel list` command, or using the command palette (ctrl-p) > CLI diagnostics:




```
# diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=Branch1 ver=2 serial=2 203.0.113.3:0->203.0.113.2:0 nexthop=203.0.113.254 tun_id=203.0.113.2
tun_id6=::203.0.113.2 status=up dst_mtu=1500 weight=1
bound_if=4 real_if=4 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc
run_state=0 role=primary accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=4 ilast=42951134 olast=42951134 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-demand on=1 status=ok idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=Branch1 proto=0 sa=1 ref=2 serial=1
  src: 0:172.16.7.0-172.16.7.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=30202 type=00 soft=0 mtu=1438 expire=41683/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42898/43200
  dec: spi=aaed4265 esp=aes key=16 cc41eec111559e7214a9f27ab1c50162
       ah=sha256 key=32 e34e02d1b4232973422389751f4b63e88d34f36e5e03cd5bec2075441d4f0f80
  enc: spi=42379cc8 esp=aes key=16 2dd731af6c93b11b28d9e13245231c23
       ah=sha256 key=32 872e556acb840a19e56ff2af3caf8a2af6303e5b72c1ca386d72d48fedda24db
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=0.0.0.0:0 npu_lgwy=0.0.0.0:0 npu_selid=0
  dec_npuid=0 enc_npuid=0 dec_engid=-1 enc_engid=-1 dec_saidx=-1 enc_saidx=-1
```

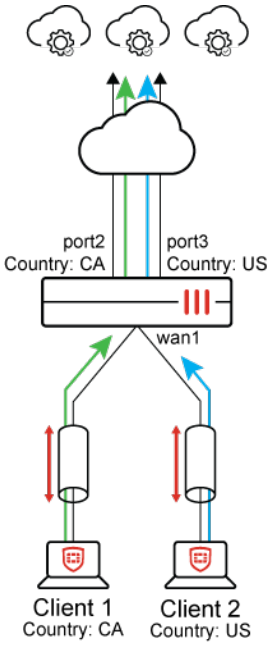
Optionally, view static routing table on a branch with the `get router info routing-table all` command, or from the command palette:

```
# get router info routing-table all
Routing table for VRF=0
S*      0.0.0.0/0 [5/0] is directly connected, branch1
S*      22.1.1.1/32 [1/0] via 15.1.1.1, wan1
```



Geography-based tunneled internet browsing NEW

A FortiGate VPN gateway can be configured to route egress traffic to the Internet through a specific interface (port2 or port3) based on the originating geolocation of the ingress client traffic (Canada or US). This can be achieved using FortiClient Secure Internet Access (SIA) configuration together with a policy route to route traffic to the proper egress interface.



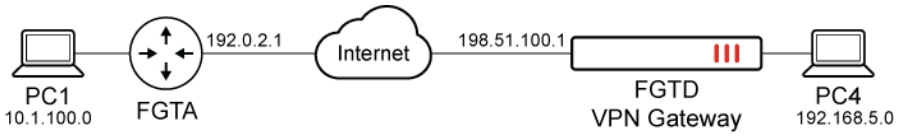
Previously, using the supported configuration of Matching IPsec tunnel gateway based on address parameters, IPsec clients with public source IP addresses originating from a specific geolocation (country/region) were matched by specifying the source geography within the IPsec tunnel configuration itself, namely, in the Phase 1 configuration. This meant that for every geolocation to be matched, a new IPsec tunnel, consisting of Phase 1 and Phase 2 settings, needed to be configured.

FortiOS has added support for specifying source geography addresses in firewall policies and policy routes on a FortiGate configured for dial-up IPsec remote access. With this enhancement, a single IPsec tunnel can be used while matching client source IP addresses based on geolocation can be achieved by configuring source geography addresses in the firewall policies and policy routes.

Example

In this example, the client PC1 behind FGTA with NAT enabled is located in Canada. FGTD is configured as the VPN Gateway with:

- Firewall source address (geo-CA) configured with CA country
- VPN IPsec tunnel
- Firewall policy with geo-CA selected as the source address



This example shows how a firewall policy and policy route involving a dialup IPsec tunnel can both be matched based on the source IP geolocation of incoming dialup clients. FortiOS internally keeps track of the original geolocation of incoming dialup traffic and integrates with firewall policies and with policy routing.

To configure FGTD:

1. On FGTD, configure a firewall address with a geolocation country/region:

```
config firewall address
    edit "geo-CA"
        set uuid 3c2d2398-9d91-51f0-4c8d-58f43ee65c8a
        set type geography
        set country "CA"
    next
end
```

2. Configure an IPsec VPN tunnel:

```
config vpn ipsec phase1-interface
    edit "Dialup"
        set type dynamic
        set interface "wan1"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set dhgrp 20 21
        set wizard-type dialup-fortigate
        set transport auto
        set psksecret ENC *
    next
end
config vpn ipsec phase2-interface
    edit "Dialup"
        set phase1name "Dialup"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
    next
end
```

3. Configure a firewall policy and select `geo-CA` as the source address:

```
config firewall policy
    edit 1
        set name "vpn_Dialup_local"
        set uuid 9378cc0e-9d8e-51f0-4e0d-a90274db10fb
        set srcintf "port1"
        set dstintf "Dialup"
        set action accept
        set srcaddr "Dialup_local"
        set dstaddr "Dialup_remote"
        set schedule "always"
        set service "ALL"
        set comments "VPN: Dialup -- Created by VPN wizard"
    next
    edit 2
        set name "vpn_Dialup_remote"
        set uuid 937abc9e-9d8e-51f0-c288-04ad4bb6f72f
```

```
        set srcintf "Dialup"
        set dstintf "port1"
        set action accept
        set srcaddr "geo-CA"
        set dstaddr "192.168.5.0"
        set schedule "always"
        set service "ALL"
        set comments "VPN: Dialup -- Created by VPN wizard"
    next
end
```

4. Configure a policy route to ensure client traffic from Canada is routed out of port2:

```
config router policy
    edit 1
        set input-device "Dialup"
        set srcaddr "geo-CA"
        set dstaddr "all"
        set gateway 198.51.100.2
        set output-device "port2"
    next
end
```

5. Verify using the VPN IKE gateway list and the VPN tunnel list:
   - VPN IKE gateway list:

```
# diagnose vpn ike gateway list

vd: root/0
name: Dialup_0
version: 2
interface: port13 19
addr: 198.51.100.1:500 -> 192.0.2.1:500
tun_id: 192.0.2.1/::10.0.0.7
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 263s ago
peer-id: 192.0.2.1
peer-id-auth: no
pending-queue: 0
PPK: no
IKE SA: created 1/1  established 1/1  time 10/10/10 ms
IPsec SA: created 1/2  established 1/2  time 0/0/0 ms

  id/spi: 5 7608e6dac7750e94/564ba227bdbd6dee
  direction: responder
  status: established 263-263s ago = 10ms
  proposal: aes128-sha256
  child: no
  SK_ei: 933120ad1055a39a-1c27f7ed4dfb258a
  SK_er: 32520d03666c4cbe-b79c65eff6b69c12
```

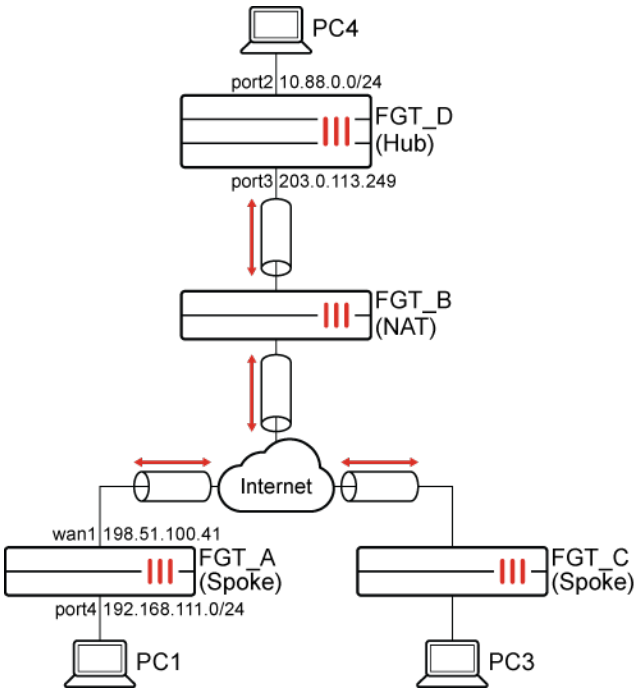
```
  SK_ai: 37da2c44ddb06cef-ab93d092d0c3ef7b-a182b40e35c98c60-2a6937a90a089707
  SK_ar: 4afc71586c84b9bd-e7e6a247bde21748-072dbdeb992b005c-c8466694bd177444
  PPK: no
  message-id sent/recv: 0/11
  QKD: no
  PQC-KEM (IKE): no
  PQC-KEM (all IPsec): no
  lifetime/rekey: 86400/85866
  DPD sent/recv: 00000000/00000000
  peer-id: 192.0.2.1
```

- VPN tunnel list:

```
# diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=Dialup_0 ver=2 serial=7 198.51.100.1:0->192.0.2.1:0 nexthop=198.51.100.2 tun_
id=192.0.2.1 tun_id6=::10.0.0.7 status=up dst_mtu=1500 weight=1 country=CA
bound_if=19 real_if=19 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none options
[0x22a8]=npu rgwy-chg frag-rfc  run_state=0 role=primary accept_traffic=1 overlay_id=0

parent=Dialup index=0
proxyid_num=1 child_num=0 refcnt=5 ilast=105 olast=106 ad=/0
stat: rxp=159 txp=4 rxb=24588 txb=336
dpd: mode=on-demand on=1 status=ok idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=Dialup proto=0 sa=1 ref=2 serial=2 add-route
  src: 0:192.168.5.0-192.168.5.255:0
  dst: 0:10.1.100.0-10.1.100.255:0
  SA:  ref=6 options=6a6 type=00 soft=0 mtu=1438 expire=43075/0B replaywin=2048
       seqno=405 esn=0 replaywin_lastseq=00000013 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43186/43200
  dec: spi=9dc42e9e esp=aes key=16 e3c82f72ce497d27736b120cda74f23a
       ah=sha256 key=32 b9702181e33b8b4986b4d3b847bfe0307231ee423dc8651085dcfe58d88771c8
  enc: spi=c8e99d60 esp=aes key=16 c40474714b560d0855653089b4cea604
       ah=sha256 key=32 45bf85337226f86b9a7774657170b206db1df5b40a9eb31a3ec2bb79949c9b32
  dec:pkts/bytes=2/168, enc:pkts/bytes=4/624
  npu_flag=03 npu_rgwy=192.0.2.1:0 npu_lgwy=198.51.100.1:0npu_selid=8
  dec_npuid=1 enc_npuid=1 dec_engid=-1 enc_engid=-1 dec_saidx=11 enc_saidx=8
------------------------------------------------------
name=Dialup ver=2 serial=1 198.51.100.1:0->0.0.0.0:0 nexthop=198.51.100.2 tun_id=10.0.0.1
tun_id6=::10.0.0.1 status=up dst_mtu=0 weight=1
bound_if=19 real_if=0 lgwy=static/1 tun=intf mode=dialup/2 encap=none options[0x228]=npu
frag-rfc  role=primary accept_traffic=1 overlay_id=0

proxyid_num=0 child_num=1 refcnt=3 ilast=42952401 olast=42952401 ad=/0
stat: rxp=734 txp=114 rxb=113640 txb=9576
dpd: mode=on-demand on=0 status=ok idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
```

VPN over TCP with Fortinet proprietary encapsulation

In this example, IPsec VPN crosses over a carrier network which blocks UDP and ESP packets. NAT is also applied between the Spokes and the Hub. Therefore, using Fortinet's proprietary encapsulation, IPsec VPN traffic is encapsulated in TCP on port 1443 to be transported over the carrier.



To configure the IKE TCP port settings in the GUI:

1. On each FortiGate, go to *VPN* > *VPN Tunnels* > *Settings*.
2. Enable *Allow VPN negotiations over TCP* and change the *TCP port* to *1443*.
3. Click *Apply*.

To configure the IKE TCP port settings in the CLI:

```
config system settings
    set ike-tcp-service enable
    set ike-tcp-port 1443
end
```

To disable anti-replay in the global settings on the FGT_B (NAT) FortiGate:

```
config system global
    set anti-replay disable
    set hostname "FGT-B"
end
```

See Verify the ESP packets sniffed on the NAT device for more details.

To configure the FGT_D (hub) FortiGate in the GUI:

1. Go to *VPN* > *VPN Wizard*.
2. Set *Tunnel name* to *Hub*.
3. Set *Select a template* to *Site to Site*.
4. Click *Begin*.
5. Configure the *VPN tunnel* settings:

Field	Value
Authentication method	*Pre-shared key* For better security, use *Signature*.
Pre-shared key	Enter an appropriate key
IKE	*Version 2*
Transport	*Auto*
Use Fortinet encapsulation	*Enable*

6. Click *Next*.
7. Configure the *Remote site* settings:

Field	Value
Remote site device type	*FortiGate*
Remote site device	*Behind NAT or dynamic*
Remote site subnets that can access VPN	*0.0.0.0/0* For better security, enter the remote subnets individually.

8. Click *Next*.
9. Configure the *Local site* settings:

Field	Value
Outgoing interface that binds to tunnel	*WAN (port3)*
Create and add interface to zone	Disable
Local interface	*DMZ (port2)*
Local subnets that can access VPN	*10.88.0.0/24*

10. Click *Next*.
11. Review the configuration and, if everything is correct, click *Submit*.

To configure the FGT_A (spoke) FortiGate in the GUI:

1. Go to *VPN* > *VPN Wizard*.
2. Set *Tunnel name* to *Spoke*.
3. Set *Select a template* to *Site to Site*.
4. Click *Begin*.
5. Configure the *VPN tunnel* settings:

Field	Value
Authentication method	*Pre-shared key* For better security, use *Signature*.
Pre-shared key	Enter an appropriate key
IKE	*Version 2*
Transport	*Auto*
Use Fortinet encapsulation	*Enable*

6. Click *Next*.
7. Configure the *Remote site* settings:

Field	Value
Remote site device type	*FortiGate*
Remote site device	*Accessible and static*
IP/FQDN	*203.0.113.249*
Route this device's internet traffic through the remote site	*Disable*
Remote site subnets that can access VPN	*0.0.0.0/0* For better security, enter the remote subnets individually.

8. Click *Next*.
9. Configure the *Local site* settings:

Field	Value
Outgoing interface that binds to tunnel	*wan1*
Create and add interface to zone	Disable
Local interface	*port4*
Local subnets that can access VPN	*192.168.111.0/24*

10. Click *Next*.

11. Review the configuration and, if everything is correct, click *Submit*.

Configure the FGT_C (spoke) FortiGate similar to FGT_A.

To configure the FGT_D (hub) FortiGate in the CLI:

1. Configure the IPsec phase 1 settings:

```
config vpn ipsec phase1-interface
    edit "Hub"
        set type dynamic
        set interface "port3"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set dhgrp 20 21
        set fortinet-esp enable
        set psksecret **********
    next
end
```

2. Configure the IPsec phase 2 settings:

```
config vpn ipsec phase2-interface
    edit "Hub"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
    next
end
```

To configure the FGT_A (spoke) FortiGate in the CLI:

1. Configure the IPsec phase 1 settings:

```
config vpn ipsec phase1-interface
    edit "spoke"
        set interface "wan1"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set dhgrp 20 21
        set transport auto
        set fortinet-esp enable
        set remote-gw 203.0.113.249
        set psksecret **********
    next
end
```

2. Configure the IPsec phase 2 settings:

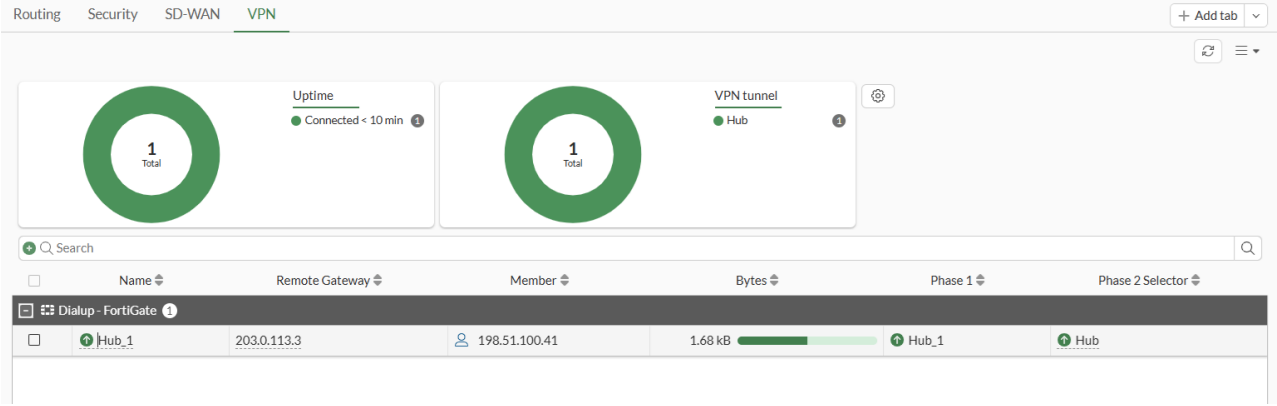
```
config vpn ipsec phase2-interface
    edit "spoke"
        set phase1name "spoke"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
    next
end
```

IKE and ESP will be encapsulated into TCP, and ESP packets encapsulated into a non-standard TCP header.

Configure the FGT_C (spoke) FortiGate similar to FGT_A.

To verify the tunnel:

1. On the Hub, go to *Dashboard* > *Network Monitor* > *VPN*.



The tunnel shows the remote member's IP address (198.51.100.41), and the NAT'd IP (203.0.113.3) as the Remote Gateway.

2. Verify the IPsec VPN tunnel state from the CLI or from the Command palette (CTRL+P) > CLI diagnostics:

```
# diagnose vpn ike gateway list

vd: root/0
name: Hub_1
version: 2
interface: port3 5
addr: 203.0.113.249:1443 -> 203.0.113.3:7782
tun_id: 10.0.0.7/::10.0.0.11
remote_location: 0.0.0.0
network-id: 0
transport: TCP
tcp-tx-1m-rate : 0 kbps
tcp-tx-5m-rate : 0 kbps
tcp-tx-drop    : 0
tcp-rx-1m-rate : 0 kbps
tcp-rx-5m-rate : 0 kbps
tcp-rx-drop    : 0
created: 485s ago
peer-id: 198.51.100.41
peer-id-auth: no
```

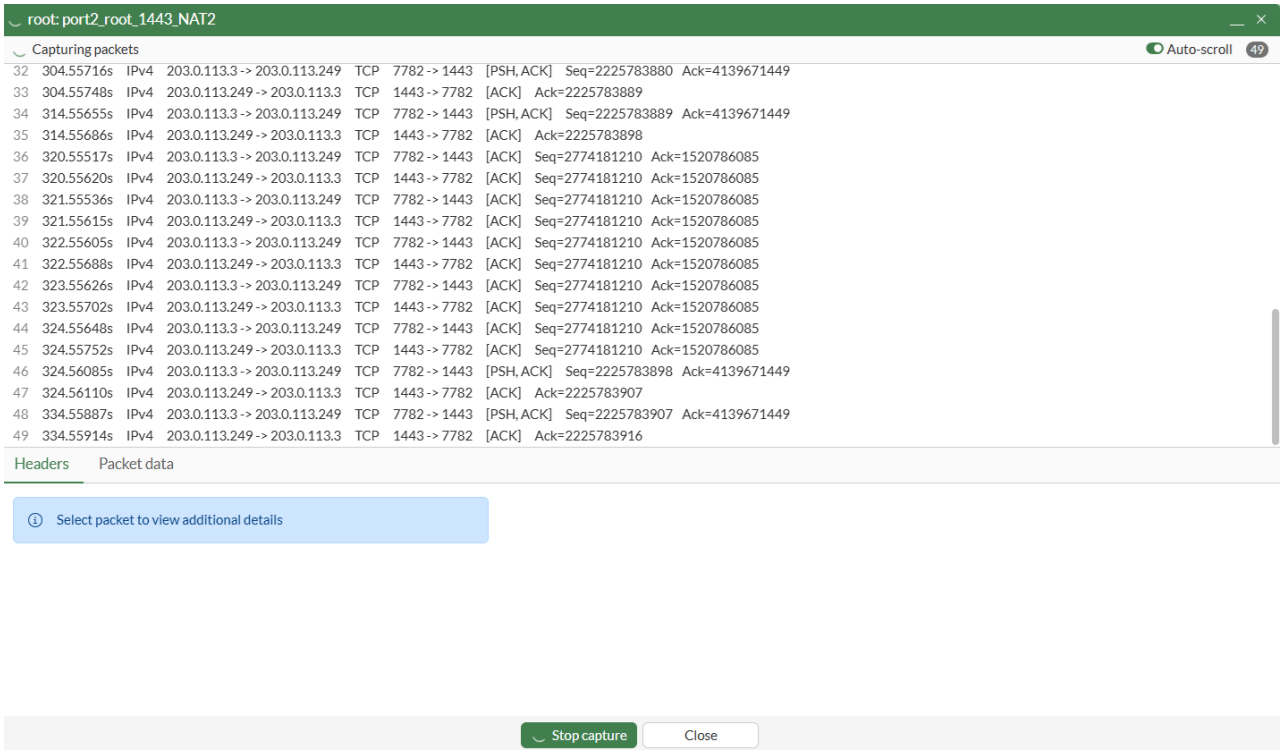
```
nat: peer
pending-queue: 0
PPK: no
IKE SA: created 1/1  established 1/1  time 150/150/150 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 2 089657727774ac21/4dde1b606f0602d6
  direction: responder
  status: established 485-485s ago = 150ms
  proposal: aes128-sha256
  child: no
  SK_ei: 25e18afd01da3dbf-74885d4e54ec223c
  SK_er: 35d8eb67940c92a8-32a7407626149f08
  SK_ai: 91cc3bf8b0c06216-5c32132ecac319c7-ff5d08f1bbbf5f1b-540cf4d98a403663
  SK_ar: 5a9c70e487a1709f-b5824d00ca25e6c0-9208ab6413c73258-72df047d0ee05736
  PPK: no
  message-id sent/recv: 0/2
  QKD: no
  PQC-KEM (IKE): no
  PQC-KEM (all IPsec): no
  lifetime/rekey: 86400/85644
  DPD sent/recv: 00000000/00000000
  peer-id: 198.51.100.41
```

3. Use the following command to see the tunnel information:

```
# diagnose vpn tunnel list

list all ipsec tunnel in vd 0
------------------------------------------------------
name=Hub ver=2 serial=e 203.0.113.249:0->0.0.0.0:0 nexthop= tun_id=10.0.0.6 tun_id6=::10.0.0.6
status=up dst_mtu=0 weight=1
bound_if=5 real_if=0 lgwy=static/1 tun=intf mode=dialup/2 encap=none options[0x20228]=npu
frag-rfc  role=primary accept_traffic=1 overlay_id=0

proxyid_num=0 child_num=1 refcnt=3 ilast=44162444 olast=44162444 ad=/0
stat: rxp=25 txp=25 rxb=2100 txb=2100
dpd: mode=on-demand on=0 status=ok idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
------------------------------------------------------
name=Hub_1 ver=2 serial=11 203.0.113.249:1443->203.0.113.3:7782 nexthop=0.0.0.0 tun_
id=10.0.0.7 tun_id6=::10.0.0.11 status=up dst_mtu=1500 weight=1 country=ZZ
bound_if=5 real_if=5 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none options[0x223a8]=npu
rgwy-chg rport-chg frag-rfc  run_state=0 role=primary accept_traffic=1 overlay_id=0

parent=Hub index=1
proxyid_num=1 child_num=0 refcnt=6 ilast=650 olast=650 ad=/0
stat: rxp=10 txp=10 rxb=840 txb=840
dpd: mode=on-demand on=1 status=ok idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=Hub proto=0 sa=1 ref=2 serial=1 add-route
```

```
src: 0:10.88.0.0-10.88.0.255:0
dst: 0:192.168.111.0-192.168.111.255:0
SA:  ref=3 options=20682 type=00 soft=0 mtu=1406 expire=42469/0B replaywin=2048
     seqno=b esn=0 replaywin_lastseq=0000000b qat=0 rekey=0 hash_search_len=1
life: type=01 bytes=0/0 timeout=43190/43200
dec: spi=81de5114 esp=aes key=16 20a323f2850fc89bd274f5192893c178
     ah=sha256 key=32 3f8af0b8c4352a1d169651abed35190921929c81eacc68c09c0986ec30ec04f4
enc: spi=ec013982 esp=aes key=16 e91b4810924fcf2d1b031abfd47d4b7f
     ah=sha256 key=32 5dca604e0171ec94ed2c997040a97fabf9eecf7dd91ad604e2e8e3d479316863
dec:pkts/bytes=10/840, enc:pkts/bytes=10/840
npu_flag=00 npu_rgwy=0.0.0.0:0 npu_lgwy=0.0.0.0:0 npu_selid=0
dec_npuid=0 enc_npuid=0 dec_engid=-1 enc_engid=-1 dec_saidx=-1 enc_saidx=-1
```

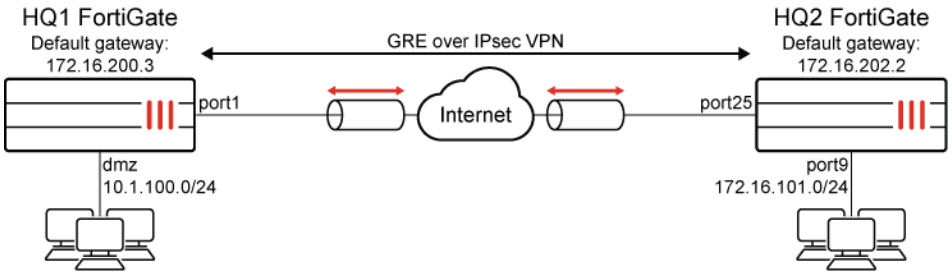
4. Verify the ESP packets sniffed on the NAT device:



In the packet capture, when data traffic passes through the tunnel, ESP packets are encapsulated into TCP ACK packets with the same sequence number. This is why anti-replay must be disabled on the NAT FortiGate.

GRE over IPsec

This is an example of GRE over an IPsec tunnel using a static route over GRE tunnel and `tunnel-mode` in the `phase2-interface` settings.



To configure GRE over an IPsec tunnel:

1. Enable subnet overlapping at both HQ1 and HQ2.

```
config system settings
    set allow-subnet-overlap enable
end
```

2. Configure the WAN interface and static route.
   a. HQ1.

```
config system interface
    edit "port1"
        set ip 172.16.200.1 255.255.255.0
    next
    edit "dmz"
        set ip 10.1.100.1 255.255.255.0
    next
end
config router static
    edit 1
        set gateway 172.16.200.3
        set device "port1"
    next
end
```

   b. HQ2.

```
config system interface
    edit "port25"
        set ip 172.16.202.1 255.255.255.0
    next
    edit "port9"
        set ip 172.16.101.1 255.255.255.0
    next
end
config router static
    edit 1
        set gateway 172.16.202.2
        set device "port25"
    next
end
```

3. Configure IPsec phase1-interface and phase2-interface.

a. HQ1.

```
config vpn ipsec phase1-interface
    edit "greipsec"
        set interface "port1"
        set peertype any
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set remote-gw 172.16.202.1
        set psksecret sample
        next
end
config vpn ipsec phase2-interface
    edit "greipsec"
        set phase1name "greipsec"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set protocol 47
    next
end
```

b. HQ2.

```
config vpn ipsec phase1-interface
    edit "greipsec"
        set interface "port25"
        set peertype any
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set remote-gw 172.16.200.1
        set psksecret sample
        next
end
config vpn ipsec phase2-interface
    edit "greipsec"
        set phase1name "greipsec"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set protocol 47
    next
end
```

4. Configure IPsec tunnel interface IP address.

a. HQ1.

```
config system interface
    edit "greipsec"
        set ip 10.10.10.1 255.255.255.255
        set remote-ip 10.10.10.2 255.255.255.255
    next
end
```

b. HQ2.

```
config system interface
    edit "greipsec"
        set ip 10.10.10.2 255.255.255.255
        set remote-ip 10.10.10.1 255.255.255.255
    next
end
```

5. Configure the GRE tunnel.
   a. HQ1.

```
config system gre-tunnel
    edit "gre_to_HQ2"
        set interface "greipsec"
        set remote-gw 10.10.10.2
        set local-gw 10.10.10.1
    next
end
```

   b. HQ2.

```
config system gre-tunnel
    edit "gre_to_HQ1"
        set interface "greipsec"
        set remote-gw 10.10.10.1
        set local-gw 10.10.10.2
    next
end
```

6. Configure the firewall policy.
   a. HQ1.

```
config firewall policy
    edit 1
        set srcintf "dmz"
        set dstintf "gre_to_HQ2"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set srcintf "gre_to_HQ2"
        set dstintf "dmz"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
    edit 3
        set srcintf "greipsec"
```

```
        set dstintf "greipsec"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end
```

b. HQ2.

```
config firewall policy
    edit 1
        set srcintf "port9"
        set dstintf "gre_to_HQ1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set srcintf "gre_to_HQ1"
        set dstintf "port9"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
    edit 3
        set srcintf "greipsec"
        set dstintf "greipsec"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end
```

7. Configure the static route.

a. HQ1.

```
config router static
    edit 2
        set dst 172.16.101.0 255.255.255.0
        set device "gre_to_HQ2"
    next
end
```

b. HQ2.

```
config router static
    edit 2
        set dst 10.1.100.0 255.255.255.0
        set device "gre_to_HQ1"
    next
end
```

To view the VPN tunnel list on HQ1:

```
diagnose vpn tunnel list
list all ipsec tunnel in vd 0
----
name=greipsec ver=1 serial=1 172.16.200.1:0->172.16.202.1:0 tun_id=172.16.202.1
bound_if=5 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/16 options[0010]=create_dev
proxyid_num=1 child_num=0 refcnt=12 ilast=19 olast=861 ad=/0
stat: rxp=347 txp=476 rxb=58296 txb=51408
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=8
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=greipsec proto=47 sa=1 ref=2 serial=2
  src: 47:0.0.0.0/0.0.0.0:0
  dst: 47:0.0.0.0/0.0.0.0:0
  SA:  ref=3 options=10226 type=00 soft=0 mtu=1438 expire=41689/0B replaywin=2048
       seqno=15c esn=0 replaywin_lastseq=0000015c itn=0
  life: type=01 bytes=0/0 timeout=42898/43200
  dec: spi=9897bd09 esp=aes key=16 5a60e67bf68379309715bd83931680bf
       ah=sha1 key=20 ff35a329056d0d506c0bfc17ef269978a4a57dd3
  enc: spi=e362f336 esp=aes key=16 5574acd8587c5751a88950e1bf8fbf57
       ah=sha1 key=20 d57ec76ac3c543ac89b2e4d0545518aa2d06669b
  dec:pkts/bytes=347/37476, enc:pkts/bytes=347/58296
```

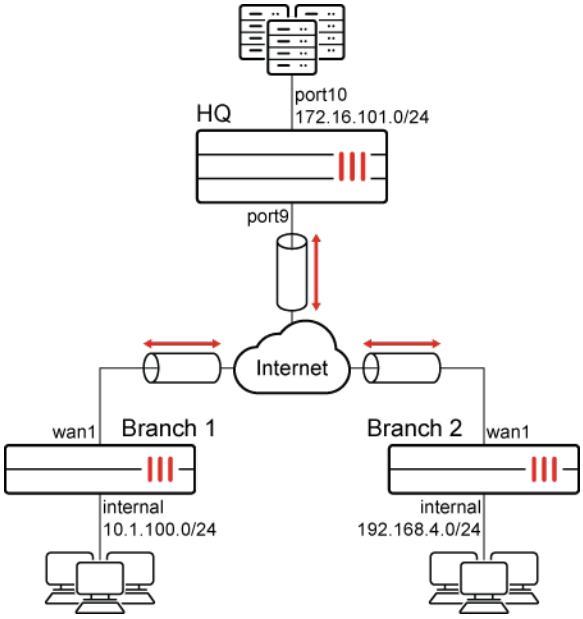
To view the static routing table on HQ1:

```
get router info routing-table static
Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 172.16.200.3, port1
S       172.16.101.0/24 [10/0] is directly connected, gre_to_HQ2
```

Policy-based IPsec tunnel

This is an example of policy-based IPsec tunnel using site-to-site VPN between branch and HQ. HQ is the IPsec concentrator.

Sample topology



Sample configuration

To configure a policy-based IPsec tunnel using the GUI:

- Configure the IPsec VPN at HQ.
- Configure the IPsec concentrator at HQ.
- Configure the firewall policy at HQ.
- Configure IPsec VPN at branch 1.
- Configure the firewall policy at branch 1.
- Configure IPsec VPN at branch 2.
- Configure the firewall policy at branch 2.

To configure the IPsec VPN at HQ:

1. Go to *VPN* > *IPsec Wizard* to set up branch 1.
   a. Enter a VPN *Name*. In this example, *to_branch1*.
   b. For *Template Type*, click *Custom*. Click *Next*.
   c. Uncheck *Enable IPsec Interface Mode*.
   d. For *Remote Gateway*, select *Static IP Address*.
   e. Enter IP address, in this example, *15.1.1.2*.
   f. For *Interface*, select *port9*.

g. In the *Authentication* section, for *Method*, select *Pre-shared Key* and enter the *Pre-shared Key*.
h. Click *OK*.

2. Go to *VPN > IPsec Wizard* to set up branch 2.
a. Enter a VPN *Name*. In this example, *to_branch2*.
b. For *Template Type*, click *Custom*. Click *Next*.
c. Uncheck *Enable IPsec Interface Mode*.
d. For *Remote Gateway*, select *Static IP Address*.
e. Enter IP address, in this example, *13.1.1.2*.
f. For *Interface*, select *port9*.
g. In the *Authentication* section, for *Method*, select *Pre-shared Key* and enter the *Pre-shared Key*.
h. Click *OK*.

To configure the IPsec concentrator at HQ:

1. Go to *VPN > IPsec Concentrator* and click *Create New*.
2. Enter a name. In this example, *branch*.
3. Add the *Membersto_branch1* and *to_branch2*.
4. Click *OK*.

To configure the firewall policy at HQ:

1. Go to *Policy & Objects > Firewall Policy* and click *Create New*.
2. Enter a policy *Name*.
3. For *Incoming Interface*, select *port10*.
4. For *Outgoing Interface*, select *port9*.
5. Select the *Source*, *Destination*, *Schedule*, *Service*, and set *Action* to *IPsec*.
6. Select the *VPN Tunnel*, in this example, *Branch1/Branch2*.
7. In this example, enable *Allow traffic to be initiated from the remote site*.
8. Click *OK*.

To configure IPsec VPN at branch 1:

1. Go to *VPN > IPsec Wizard* to set up branch 1.
2. Enter a VPN name. In this example, *to_HQ*.
3. For *Template Type*, click *Custom*. Click *Next*.
4. Uncheck *Enable IPsec Interface Mode*.
5. For *Remote Gateway*, select *Static IP Address*.
6. Enter IP address, in this example, *22.1.1.1*.
7. For *Interface*, select *wan1*.
8. In the *Authentication* section, for *Method*, select *Pre-shared Key* and enter the *Pre-shared Key*.
9. Click *OK*.

To configure the firewall policy at branch 1:

1. Go to *Policy & Objects > Firewall Policy* and click *Create New*.
2. Enter a policy *Name*.
3. Choose the *Incoming Interface*, in this example, *internal*.
4. Choose the *Outgoing Interface*, in this example, *wan1*.
5. Select the *Source*, *Destination*, *Schedule*, *Service*, and set *Action* to *IPsec*.
6. Select the *VPN Tunnel*, in this example, *Branch1/Branch2*.
7. In this example, enable *Allow traffic to be initiated from the remote site*.
8. Click *OK*.

To configure IPsec VPN at branch 2:

1. Go to *VPN > IPsec Wizard* to set up branch 1.
2. Enter a VPN name. In this example, *to_HQ*.
3. For *Template Type*, click *Custom*. Click *Next*.
4. Uncheck *Enable IPsec Interface Mode*.
5. For *Remote Gateway*, select *Static IP Address*.
6. Enter IP address, in this example, *22.1.1.1*.
7. For *Interface*, select *wan1*.
8. In the *Authentication* section, for *Method*, select *Pre-shared Key* and enter the *Pre-shared Key*.
9. Click *OK*.

To configure the firewall policy at branch 2:

1. Go to *Policy & Objects > Firewall Policy* and click *Create New*.
2. Enter a policy *Name*.
3. Choose the *Incoming Interface*, in this example, *internal*.
4. Choose the *Outgoing Interface*, in this example, *wan1*.
5. Select the *Source*, *Destination*, *Schedule*, *Service*, and set *Action* to *IPsec*.
6. Select the *VPN Tunnel*, in this example, *to_HQ*.
7. In this example, enable *Allow traffic to be initiated from the remote site*.
8. Click *OK*.

To configure a policy-based IPsec tunnel using the CLI:

1. Configure the HQ WAN interface and static route.

```
config system interface
    edit "port9"
        set alias "WAN"
        set ip 22.1.1.1 255.255.255.0
    next
    edit "port10"
        set alias "Internal"
        set ip 172.16.101.1 255.255.255.0
```

```
    next
end
config router static
    edit 1
        set gateway 22.1.1.2
        set device "port9"
    next
end
```

2. Configure the HQ IPsec phase1 and phase2.

```
config vpn ipsec phase1
    edit "to_branch1"
        set interface "port9"
        set peertype any
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set remote-gw 15.1.1.2
        set psksecret sample
        next
    edit "to_branch2"
        set interface "port9"
        set peertype any
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set remote-gw 13.1.1.2
        set psksecret sample
        next
end
config vpn ipsec phase2
    edit "to_branch1"
        set phase1name "to_branch1"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
    next
    edit "to_branch2"
        set phase1name "to_branch2"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
    next
end
```

3. Configure the firewall policy at HQ.

```
config firewall policy
    edit 1
        set srcintf "port10"
        set dstintf "port9"
        set srcaddr "all"
        set dstaddr "10.1.100.0"
        set action ipsec
        set schedule "always"
        set service "ALL"
        set inbound enable
```

```
        set vpntunnel "to_branch1"
    next
    edit 2
        set srcintf "port10"
        set dstintf "port9"
        set srcaddr "all"
        set dstaddr "192.168.4.0"
        set action ipsec
        set schedule "always"
        set service "ALL"
        set inbound enable
        set vpntunnel "to_branch2"
    next
end
```

4. Configure the IPsec concentrator at HQ.

```
config vpn ipsec concentrator
    edit 1
        set name "branch"
        set member "to_branch1" "to_branch2"
    next
end
```

5. Configure the branch WAN interface and static route.
   a. For branch 1.

```
config system interface
    edit "wan1"
        set alias "primary_WAN"
        set ip 15.1.1.2 255.255.255.0
    next
    edit "internal"
        set ip 10.1.100.1 255.255.255.0
    next
end
config router static
    edit 1
        set gateway 15.1.1.1
        set device "wan1"
    next
end
```

   b. For branch 2.

```
config system interface
    edit "wan1"
        set alias "primary_WAN"
        set ip 13.1.1.2 255.255.255.0
    next
    edit "internal"
        set ip 192.168.4.1 255.255.255.0
```

```
    next
end
config router static
    edit 1
        set gateway 13.1.1.1
        set device "wan1"
    next
end
```

6. Configure the branch IPsec phase1 and phase2.
   a. For branch 1.

```
config vpn ipsec phase1
    edit "to_HQ"
        set interface "wan1"
        set peertype any
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set remote-gw 22.1.1.1
        set psksecret sample
        next
end
config vpn ipsec phase2
    edit "to_HQ"
        set phase1name "to_HQ"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
    next
end
```

   b. For branch 2.

```
config vpn ipsec phase1
    edit "to_HQ"
        set interface "wan1"
        set peertype any
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set remote-gw 22.1.1.1
        set psksecret sample
        next
end
config vpn ipsec phase2
    edit "to_HQ"
        set phase1name "to_HQ"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
    next
end
```

7. Configure the branch firewall policy.

a. For branch 1.

```
config firewall policy
    edit 1
        set srcintf "internal"
        set dstintf "wan1"
        set srcaddr "10.1.100.0"
        set dstaddr "all"
        set action ipsec
        set schedule "always"
        set service "ALL"
        set inbound enable
        set vpntunnel "to_HQ"
    next
end
```

b. For branch 2.

```
config firewall policy
    edit 1
        set srcintf "internal"
        set dstintf "wan1"
        set srcaddr "192.168.4.0"
        set dstaddr "all"
        set action ipsec
        set schedule "always"
        set service "ALL"
        set inbound enable
        set vpntunnel "to_HQ"
    next
end
```

To view the IPsec VPN tunnel list at HQ:

```
# diagnose vpn tunnel list
```

```
list all ipsec tunnel in vd 0
----
name=to_branch1 ver=1 serial=4 22.1.1.1:0->15.1.1.2:0 tun_id=15.1.1.2
bound_if=42 lgwy=static/1 tun=tunnel/1 mode=auto/1 encap=none/8 options[0008]=npu
proxyid_num=1 child_num=0 refcnt=8 ilast=0 olast=0 ad=/0
stat: rxp=305409 txp=41985 rxb=47218630 txb=2130108
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=to_branch1 proto=0 sa=1 ref=3 serial=1
  src: 0:0.0.0.0/0.0.0.0:0
  dst: 0:0.0.0.0/0.0.0.0:0
  SA:  ref=6 options=10226 type=00 soft=0 mtu=1438 expire=42604/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000680 itn=0
  life: type=01 bytes=0/0 timeout=42932/43200
  dec: spi=ca646442 esp=aes key=16 58c91d4463968dddccc4fd97de90a4b8
       ah=sha1 key=20 c9176fe2fbc82ef7e726be9ad4af83eb1b55580a
```

```
  enc: spi=747c10c4 esp=aes key=16 7cf0f75b784f697bc7f6d8b4bb8a83c1
       ah=sha1 key=20 cdddc376a86f5ca0149346604a59af07a33b11c5
  dec:pkts/bytes=1664/16310, enc:pkts/bytes=0/16354
  npu_flag=03 npu_rgwy=15.1.1.2 npu_lgwy=22.1.1.1 npu_selid=3 dec_npuid=2 enc_npuid=2
----
name=to_branch2 ver=1 serial=5 22.1.1.1:0->13.1.1.2:0 tun_id=13.1.1.2
bound_if=42 lgwy=static/1 tun=tunnel/1 mode=auto/1 encap=none/8 options[0008]=npu
proxyid_num=1 child_num=0 refcnt=7 ilast=2 olast=43228 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=to_branch2 proto=0 sa=1 ref=2 serial=1
  src: 0:0.0.0.0/0.0.0.0:0
  dst: 0:0.0.0.0/0.0.0.0:0
  SA:  ref=3 options=10226 type=00 soft=0 mtu=1280 expire=40489/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 itn=0
  life: type=01 bytes=0/0 timeout=42931/43200
  dec: spi=ca646441 esp=aes key=16 57ab680d29d4aad4e373579fb50e9909
       ah=sha1 key=20 12a2bc703d2615d917ff544eaff75a6d2c17f1fe
  enc: spi=f9cffb61 esp=aes key=16 3d64da9feb893874e007babce0229259
       ah=sha1 key=20 f92a3ad5e56cb8e89c47af4dac10bf4b4bebff16
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=13.1.1.2 npu_lgwy=22.1.1.1 npu_selid=4 dec_npuid=0 enc_npuid=0
```

To view the IPsec VPN concentrator at HQ:

```
# diagnose vpn concentrator list
```

```
list all ipsec concentrator in vd 0
name=branch              ref=3          tuns=2 flags=0
```

SM3/SM4 cryptographic algorithms for IKEv1/ IKEv2 NEW

The ShangMi SM4 encryption algorithm and SM3 hash algorithm are supported proposals for IKEv1 and IKEv2 site-to-site VPN configurations. The SM4-SM3 pair can be selected for Phase1 IKE proposal and Phase2 ESP tunnel proposal.

These encryption algorithms are often used in regulatory requirements in China. They are incorporated into ISO/IEC standards and recognized by IETF in the Internet-Draft for Using ShangMi in the Internet Key Exchange Protocol Version 2 (IKEv2).

For IKEv2:

FortiGate supports the following transform types and names for IKEv2:

Transform Type	Type	Name
1	Encryption Algorithm Transform	ENCR_SM4_CBC

Transform Type	Type	Name
2	Pseudorandom Function Transform	PRF_HMAC_SM3
3	Integrity Algorithm Transform	AUTH_HMAC_SM3

In IKEv2, when SM4-SM3 proposal is chosen:

- HMAC-SM3 is used as the PRF and integrity algorithm.
- SM4-CBC is used for IKE encryption.
- 128-bit IV length defined by SM4-CBC block size.
- For IPsec:
  - SM4-CBC is used for payload encryption. IV length: 128 bits.
  - HMAC-SM3 is used for ESP integrity. ICV uses the full 256-bit hash output without truncation.

For IKEv1:

Similarly, IKEv1 uses the following transform identifiers:

- OAKLEY_ATTR_ENC_ALG_SM4
- OAKLEY_ATTR_HASH_ALG_SM3
- IPSECDOI_ESP_SM4_CBC
- IPSECDOI_ATTR_AUTH_HMAC_SM3_256

Supported FortiGate models:

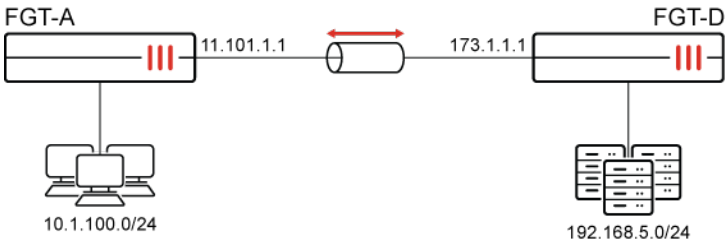
- 200G series
- 400F
- 600F
- 900G
- All high-end F-series from 1000F and above
- 7000F
- FG-VM64

Limitations

- Any tunnels using SM4-SM3 proposal do not support hardware acceleration.
- It is only available to physical or FortiGate-VMs using x86_64 processors.
- FGT using 3.2 Linux kernel are not supported

Example

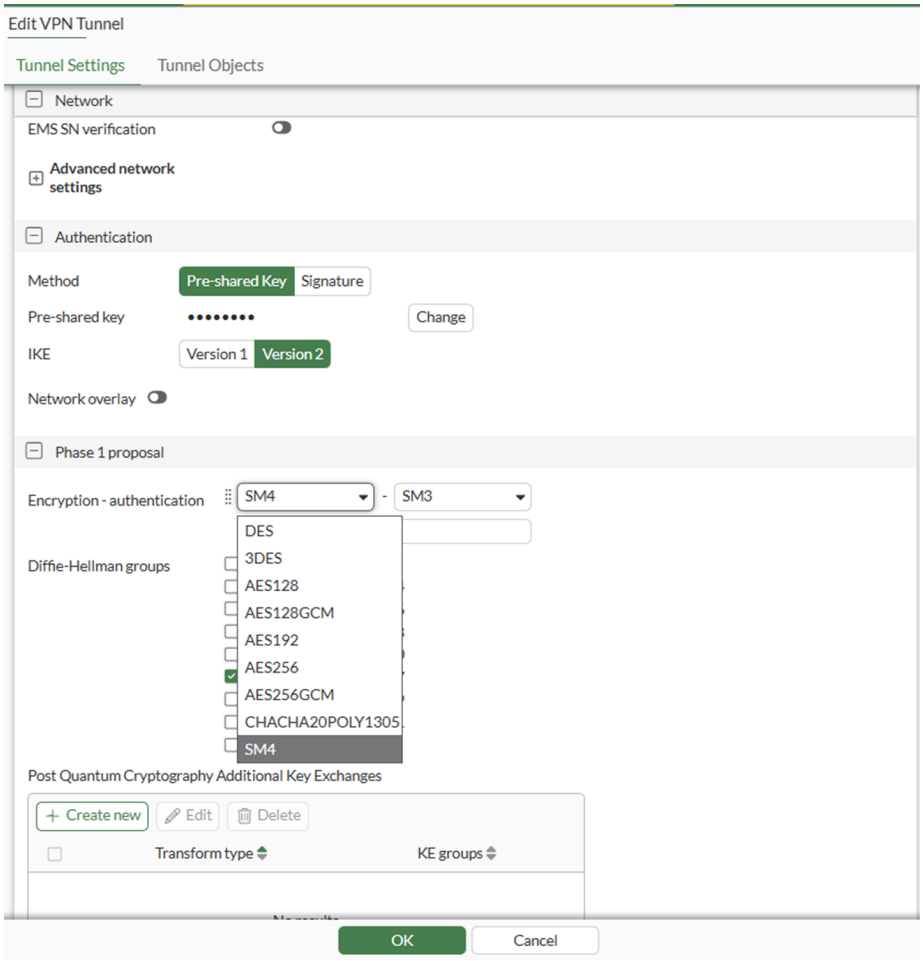
SM4-SM3 is used in the IKE proposal in IKEv2 to meet regulatory requirement for a site-to-site VPN.



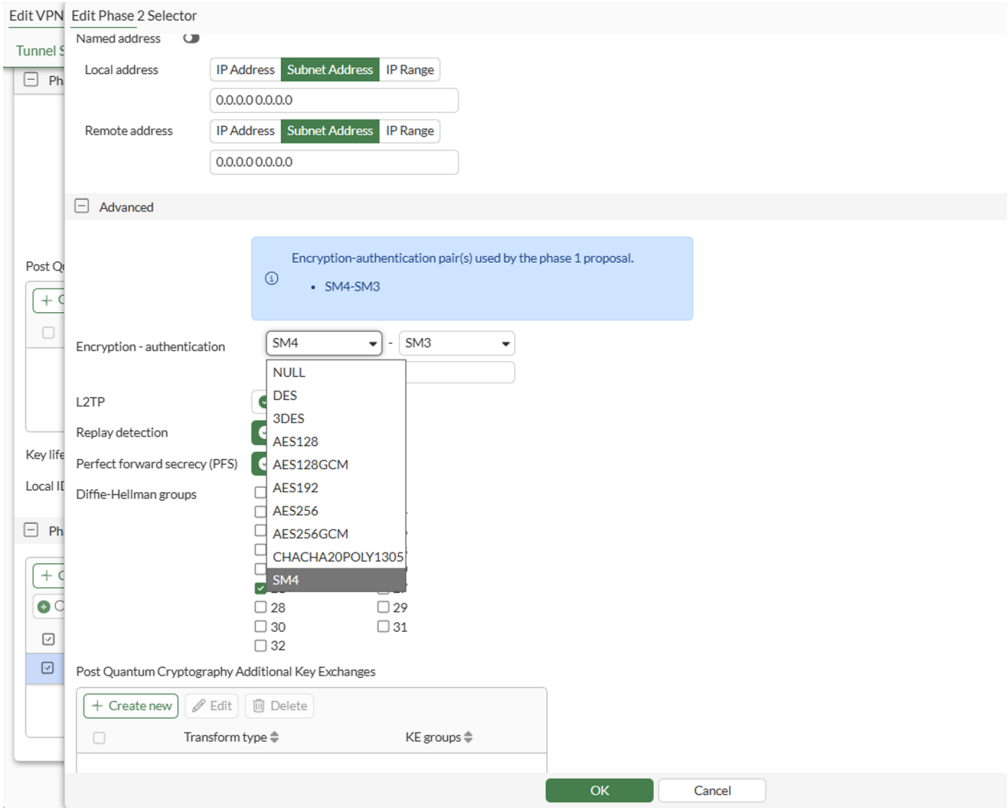
To configure in the GUI:

Exact steps are not provided in detail. See CLI configuration for exact configurations.

1. On FGT-A, go to *VPN > VPN Tunnels* and create a new tunnel.
2. Set *Remote gateway* to *Static IP address*, and configure the *Network* and *Authentication* settings as needed.
3. For Phase 1 proposal:
   a. Remove all default proposals.
   b. Select *SM4* for the *Encryption* algorithm. SM3 will be selected automatically for the *authentication* algorithm.



4. For Phase 2, create a new selector:
   a. Under *Advanced*, remove all default proposals.
   b. Select *SM4* for the *Encryption* algorithm. SM3 will be selected automatically for the *authentication* algorithm.
   c. Click *OK*.



5. Click *OK*.
6. Similarly configure SM4-SM3 proposals in the VPN configurations on your remote FortiGate FGT-D.

To configure in the CLI:

```
config system settings
    set ike-proposal-visibility all
end
```

```
config vpn ipsec phase1-interface
    edit "HUB"
        set type dynamic
        set interface "port13"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal sm4-sm3
        set dhgrp 20 21
        set psksecret <password>
```

```
   next
end
```

```
config vpn ipsec phase2-interface
    edit "HUB"
        set phase1name "HUB"
        set proposal sm4-sm3
        set dhgrp 20 21
    next
end
```

```
config firewall policy
    edit 1
        set srcintf "any"
        set dstintf "any"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
    next
end
```

This policy is for example purposes only. Do not configure an any-to-any allow policy in production.

Verification

Use the following CLI commands to view the real-time debugs when establishing the VPN tunnel. Note the proposals that were offered and matched in the three transform types (ENCRYPTION, INTEGRITY, PRF) for IKE (P1) and for ESP (P2).

```
# diagnose debug application ike -1
# diagnose debug enable

ike V=root:0:toHUB:376: incoming proposal:
ike V=root:0:toHUB:376: proposal id = 1:
ike V=root:0:toHUB:376:   protocol = IKEv2:
ike V=root:0:toHUB:376:      encapsulation = IKEv2/none
ike V=root:0:toHUB:376:         type=ENCR, val=SM4_CBC (key_len = 128)
ike V=root:0:toHUB:376:         type=INTEGR, val=AUTH_HMAC_SM3
ike V=root:0:toHUB:376:         type=PRF, val=PRF_HMAC_SM3
ike V=root:0:toHUB:376:         type=DH_GROUP, val=ECP384.
ike V=root:0:toHUB:376: matched proposal id 1
ike V=root:0:toHUB:376: proposal id = 1:
ike V=root:0:toHUB:376:   protocol = IKEv2:
ike V=root:0:toHUB:376:      encapsulation = IKEv2/none
ike V=root:0:toHUB:376:         type=ENCR, val=SM4_CBC (key_len = 128)
ike V=root:0:toHUB:376:         type=INTEGR, val=AUTH_HMAC_SM3
```

```
ike V=root:0:toHUB:376:         type=PRF, val=PRF_HMAC_SM3
ike V=root:0:toHUB:376:         type=DH_GROUP, val=ECP384.
...
ike V=root:0:toHUB:376:toHUB:495: incoming child SA proposal:
ike V=root:0:toHUB:376:toHUB:495: proposal id = 1:
ike V=root:0:toHUB:376:toHUB:495:   protocol = ESP:
ike V=root:0:toHUB:376:toHUB:495:      encapsulation = TUNNEL
ike V=root:0:toHUB:376:toHUB:495:         type=ENCR, val=SM4_CBC (key_len = 128)
ike V=root:0:toHUB:376:toHUB:495:         type=INTEGR, val=SM3
ike V=root:0:toHUB:376:toHUB:495:         type=ESN, val=NO
ike V=root:0:toHUB:376:toHUB:495:         PFS is disabled
ike V=root:0:toHUB:376:toHUB:495: matched proposal id 1
ike V=root:0:toHUB:376:toHUB:495: proposal id = 1:
ike V=root:0:toHUB:376:toHUB:495:   protocol = ESP:
ike V=root:0:toHUB:376:toHUB:495:      encapsulation = TUNNEL
ike V=root:0:toHUB:376:toHUB:495:         type=ENCR, val=SM4_CBC (key_len = 128)
ike V=root:0:toHUB:376:toHUB:495:         type=INTEGR, val=SM3
ike V=root:0:toHUB:376:toHUB:495:         type=ESN, val=NO
ike V=root:0:toHUB:376:toHUB:495:         PFS is disabled
...
ike V=root:0:toHUB:376:toHUB:495: sending SNMP tunnel UP trap
ike V=root:0:toHUB: static tunnel up event 0.0.0.0 (dev=53)
ike V=root:0:toHUB: static tunnel up event :: (dev=53)
```

Use the following CLI commands to verify the tunnel is established with the SM4-SM3 proposal:

```
# diagnose vpn ike gateway list name <tunnel>

vd: root/0
name: toHUB
version: 2
interface: port9 15
addr: 11.101.1.1:500 -> 173.1.1.1:500
tun_id: 173.1.1.1/::173.1.1.1
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 146s ago
peer-id: 173.1.1.1
peer-id-auth: no
pending-queue: 0
PPK: no
IKE SA: created 1/1  established 1/1  time 10/10/10 ms
IPsec SA: created 1/1  established 1/1  time 10/10/10 ms

  id/spi: 375 b276017ea711477c/e963d7bfba066129
  direction: initiator
  status: established 146-146s ago = 10ms
  proposal: sm4_128-sm3
  child: no
  SK_ei: 152ebe7d097da0e6-459a04571cc85cc0
  SK_er: 8613e0506e88b1df-2129cd5e00349db8
  SK_ai: d3a2a1a84a6581c1-1a97633532fa0d51-829a21602f0276f7-b8e79b2a888e2204
```

```
  SK_ar: fa08e32590ed6d7c-f27a4e6dc8159c12-be0f8cf0b1b56483-a955315d5caf2fa2
  PPK: no
  message-id sent/recv: 9/1
  QKD: no
  PQC-KEM (IKE): no
  PQC-KEM (all IPsec): no
  lifetime/rekey: 86400/85953
  DPD sent/recv: 00000033/00000033
  peer-id: 173.1.1.1
```

Similarly, use the following CLI commands to show the encryption and authentication algorithms used in the IPsec SAs.

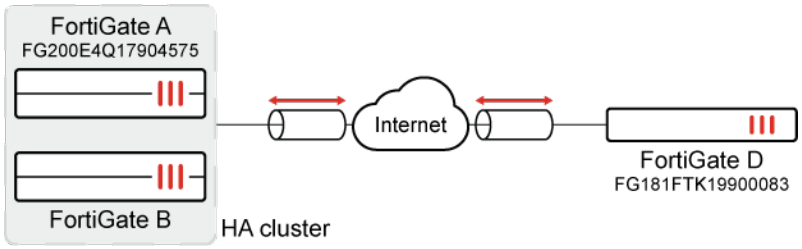
```
# diagnose vpn tunnel list name <tunnel>

list all ipsec tunnel in vd 0
------------------------------------------------------
name=toHUB ver=2 serial=3 11.101.1.1:0->173.1.1.1:0 nexthop=11.101.1.2 tun_id=173.1.1.1 tun_
id6=::173.1.1.1 status=up dst_mtu=1500 weight=1 country=US
bound_if=15 real_if=15 lgwy=static/1 tun=intf mode=auto/1 encap=none options[0x228]=npu frag-rfc
run_state=0 role=primary accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=4 ilast=15 olast=0 ad=/0
stat: rxp=0 txp=146 rxb=0 txb=12264
dpd: mode=on-demand on=1 status=ok idle=20000ms retry=3 count=0 seqno=51
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=toHUB proto=0 sa=1 ref=3 serial=4
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=10226 type=00 soft=0 mtu=1422 expire=42749/0B replaywin=2048
       seqno=93 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42900/43200
  dec: spi=c7c6a1b5 esp=sm4 key=16 869f98b2bd18bdfd7211c0db9488e850
       ah=sm3 key=32 bfaff448ea0b23bca803420429d871a8d349ff12089bf277f89258d2567b3f3d
  enc: spi=d5e84636 esp=sm4 key=16 83ac4c93abfb1b57e6d06d3e4da595bf
       ah=sm3 key=32 832da674f1beb4fd9a8a01c3a992a55cff12792fb72e7bd38f0d21ca17cc586a
  dec:pkts/bytes=0/0, enc:pkts/bytes=146/12264
  npu_flag=20 npu_rgwy=0.0.0.0:0 npu_lgwy=0.0.0.0:0 npu_selid=6
  dec_npuid=0 enc_npuid=0 dec_engid=-1 enc_engid=-1 dec_saidx=-1 enc_saidx=-1
```

Securely exchange serial numbers between FortiGates connected with IPsec VPN

Serial numbers can be securely exchanged between FortiGates connected with IPsec VPN. This feature is supported in IKEv2, IKEv1 main mode, and IKEv1 aggressive mode. The exchange is only performed with participating FortiGates that have enabled the `exchange-fgt-device-id` setting under `config vpn ipsec phase1-interface`.

Example

In this example, FortiGates A and B are in an HA cluster, so the serial numbers will not exchange after failover. The cluster is connected to FortiGate D through IPsec VPN.



To securely exchange serial numbers between the FortiGates:

1. Configure the IPsec settings on FortiGate A.
   a. Configure the phase 1 interface settings:

```
config vpn ipsec phase1-interface
    edit "to_FGTD"
        set interface "port1"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set exchange-fgt-device-id enable
        set remote-gw 172.16.200.4
        set psksecret **********
    next
end
```

   b. Configure the phase 2 interface settings:

```
config vpn ipsec phase2-interface
    edit "to_FGTD"
        set phase1name "to_FGTD"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set src-addr-type name
        set dst-addr-type name
        set src-name "to_FGTD_local"
        set dst-name "to_FGTD_remote"
    next
end
```

2. Configure the IPsec settings on FortiGate D.
   a. Configure the phase 1 interface settings:

```
config vpn ipsec phase1-interface
    edit "to_FGTA"
        set interface "port2"
        set peertype any
```

```
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set exchange-fgt-device-id enable
        set remote-gw 172.16.200.1
        set psksecret **********
    next
end
```

b. Configure the phase 2 interface settings:

```
config vpn ipsec phase2-interface
    edit "to_FGTA"
        set phase1name "to_FGTA"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set src-addr-type name
        set dst-addr-type name
        set src-name "to_FGTA_local"
        set dst-name "to_FGTA_remote"
    next
end
```

3. Verify the peer serial numbers.

a. On FortiGate A:

```
# diagnose vpn ike gateway list

vd: root/0
name: to_FGTD
version: 1
interface: port1 19
addr: 172.16.200.1:500 -> 172.16.200.4:500
tun_id: 172.16.200.4/::172.16.200.4
remote_location: 0.0.0.0
network-id: 0
created: 783s ago
peer-id: 172.16.200.4
peer-id-auth: no
peer-SN: FG181FTK19900083
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 2 a8b2df203ef134e8/955fafbd10a04fa0
  direction: initiator
  status: established 783-783s ago = 0ms
  proposal: aes128-sha256
  key: 644db099e1178d1f-119fee3141f1e2a6
  lifetime/rekey: 86400/85316
  DPD sent/recv: 00000000/00000000
  peer-id: 172.16.200.4
```

b. On FortiGate D:

```
# diagnose vpn ike gateway list

vd: root/0
name: to_FGTA
version: 1
interface: port2 10
addr: 172.16.200.4:500 -> 172.16.200.1:500
tun_id: 172.16.200.1/::172.16.200.1
remote_location: 0.0.0.0
network-id: 0
created: 723s ago
peer-id: 172.16.200.1
peer-id-auth: no
peer-SN: FG200E4Q17904575
IKE SA: created 1/1  established 1/1  time 10/10/10 ms
IPsec SA: created 0/0

  id/spi: 7 a8b2df203ef134e8/955fafbd10a04fa0
  direction: responder
  status: established 723-723s ago = 10ms
  proposal: aes128-sha256
  key: 644db099e1178d1f-119fee3141f1e2a6
  lifetime/rekey: 86400/85406
  DPD sent/recv: 00000000/00000000
  peer-id: 172.16.200.1
```

4. After an HA failover, verify that the peer serial numbers have not changed.
   a. On FortiGate B:

```
# diagnose vpn ike gateway  list

vd: root/0
name: to_FGTD
version: 2
interface: port1 19
addr: 172.16.200.1:500 -> 172.16.200.4:500
tun_id: 172.16.200.4/::172.16.200.4
remote_location: 0.0.0.0
network-id: 0
created: 104s ago
peer-id: 172.16.200.4
peer-id-auth: no
peer-SN: FG181FTK19900083
PPK: no
IKE SA: created 1/2  established 1/2  time 0/0/0 ms
IPsec SA: created 1/2  established 1/2  time 0/0/0 ms

  id/spi: 8 3aab6778ea613bcd/e28dd0a1251a2eb1
  direction: responder
  status: established 101-101s ago = 0ms
  proposal: aes128-sha256
  child: no
```

```
  SK_ei: c05f59ac726e4c3c-0d273aa8bf5dde35
  SK_er: 5be947724fbbd85b-d1e090a757823e6a
  SK_ai: 11f85a5c896a897f-2d7a551a91d5c1e2-63394ec02414ddb2-33598a09e77c8207
  SK_ar: 4291445e00062982-f7c5a848c9ada403-6ce7e4394e3a4fd5-bf2dc03492576cfc
  PPK: no
  message-id sent/recv: 12/3
  lifetime/rekey: 86400/86028
  DPD sent/recv: 00000000/00000000
  peer-id: 172.16.200.4
```

b. On FortiGate D:

```
# diagnose vpn ike gateway list

vd: root/0
name: to_FGTA
version: 2
interface: port2 10
addr: 172.16.200.4:500 -> 172.16.200.1:500
tun_id: 172.16.200.1/::172.16.200.1
remote_location: 0.0.0.0
network-id: 0
created: 132s ago
peer-id: 172.16.200.1
peer-id-auth: no
peer-SN: FG200E4Q17904575
PPK: no
IKE SA: created 1/2  established 1/2  time 0/10500/21000 ms
IPsec SA: created 1/2  established 1/2  time 0/10500/21000 ms

  id/spi: 9 3aab6778ea613bcd/e28dd0a1251a2eb1
  direction: initiator
  status: established 132-111s ago = 21000ms
  proposal: aes128-sha256
  child: no
  SK_ei: c05f59ac726e4c3c-0d273aa8bf5dde35
  SK_er: 5be947724fbbd85b-d1e090a757823e6a
  SK_ai: 11f85a5c896a897f-2d7a551a91d5c1e2-63394ec02414ddb2-33598a09e77c8207
  SK_ar: 4291445e00062982-f7c5a848c9ada403-6ce7e4394e3a4fd5-bf2dc03492576cfc
  PPK: no
  message-id sent/recv: 3/12
  lifetime/rekey: 86400/85988
  DPD sent/recv: 00000000/00000000
  peer-id: 172.16.200.1
```

To retrieve the peer serial number in FortiManager:

1. Add and authorize FortiGate A (see Adding online devices using Discover mode for more details).
2. Go to *Device Manager > Device & Groups* and select the FortiGate A.
3. Add the *IPsec VPN* widget (see Customizing the dashboard for more details).
4. Open the developer tools in your browser and select the *Network* tab.
5. Refresh the *IPsec VPN* widget.

6. In the *Network* tab, there should be a JSON POST request that FortiManager will proxy request to the FortiGate for the IPsec API. The response should contain the peer serial number.

FortiGate-to-third-party

This section contains the following topics about FortiGate-to-third-party VPN configurations:

- IKEv2 IPsec site-to-site VPN to an AWS VPN gateway on page 2935
- IPsec VPN to Azure with virtual network gateway on page 2942
- IPsec VPN to an Azure with virtual WAN on page 2950
- IPSec VPN between a FortiGate and a Cisco ASA with multiple subnets on page 2954
- Cisco GRE-over-IPsec VPN on page 2955
- Cloud SDN Orchestration VPN wizard for AWS NEW on page 2961

IKEv2 IPsec site-to-site VPN to an AWS VPN gateway

This is a sample configuration of an IPsec site-to-site VPN connection between an on-premise FortiGate and an AWS virtual private cloud (VPC).

AWS uses unique identifiers to manipulate a VPN connection's configuration. Each VPN connection is assigned an identifier and is associated with two other identifiers: the customer gateway ID for the FortiGate and virtual private gateway ID.

This example includes the following IDs:

- VPN connection ID: vpn-07e988ccc1d46f749
- Customer gateway ID: cgw-0440c1aebed2f418a
- Virtual private gateway ID

This example assumes that you have configured VPC-related settings in the AWS management portal.

This example includes creating and configuring two tunnels. You must configure both tunnels on your FortiGate.

To configure IKEv2 IPsec site-to-site VPN to an AWS VPN gateway:

1. Configure the first VPN tunnel:
   a. Configure Internet Key Exchange (IKE).
   b. Configure IPsec.
   c. Configure the tunnel interface.
   d. Configure border gateway protocol (BGP).
   e. Configure firewall policies.
2. Configure the second VPN tunnel:
   a. Configure Internet Key Exchange (IKE).
   b. Configure IPsec.
   c. Configure the tunnel interface.
   d. Configure BGP.
   e. Configure firewall policies.

To configure IKE for the first VPN tunnel:

A policy is established for the supported ISAKMP encryption, authentication, Diffie-Hellman (DH), lifetime, and key parameters. These sample configurations fulfill the minimum requirements for AES128, SHA1, and DH Group 2. Category VPN connections in the GovCloud AWS region have a minimum requirement of AES128, SHA2, and DH Group 14. To take advantage of AES256, SHA256, or other DH groups such as 14-18, 22, 23, and 24, you must modify these sample configuration files. Higher parameters are only available for VPNs of category "VPN", not for "VPN-Classic".

Your FortiGate's external interface's address must be static. Your FortiGate may reside behind a device performing NAT. To ensure NAT traversal can function, you must adjust your firewall rules to unblock UDP port 4500. If not behind NAT, it is recommended to disable NAT traversal.

Begin configuration in the root VDOM. The interface name must be shorter than 15 characters. It is best if the name is shorter than 12 characters. IPsec dead peer detection (DPD) causes periodic messages to be sent to ensure a security association remains operational.

```
config vpn ipsec phase1-interface
    edit vpn-07e988ccc1d46f749-0
        set interface "wan1"
        set dpd enable
        set local-gw 35.170.66.108
        set dhgrp 2
        set proposal aes128-sha1
        set keylife 28800
        set remote-gw 3.214.239.164
        set psksecret iCelks0UOob8z4SYMRM6zlx.rU2C3jth
        set dpd-retryinterval 10
    next
end
```

To configure IPsec for the first VPN tunnel:

The IPsec transform set defines the encryption, authentication, and IPsec mode parameters.

```
config vpn ipsec phase2-interface
    edit "vpn-07e988ccc1d46f749-0"
        set phase1name "vpn-07e988ccc1d46f749-0"
        set proposal aes128-sha1
        set dhgrp 2
        set pfs enable
        set keylifeseconds 3600
    next
end
```

To configure the tunnel interface for the first VPN tunnel:

You must configure a tunnel interface as the logical interface associated with the tunnel. All traffic routed to the tunnel interface must be encrypted and transmitted to the VPC. Similarly, traffic from the VPC will be logically received on this interface.

You must configure the interface's address with your FortiGate's address. If the address changes, you must recreate the FortiGate and VPN connection with Amazon VPC.

The `tcp-mss` option causes the router to reduce the TCP packets' maximum segment size to prevent packet fragmentation.

```
config system interface
    edit "vpn-07e988ccc1d46f749-0"
        set vdom "root"
        set ip 169.254.45.90 255.255.255.255
        set allowaccess ping
        set type tunnel
        set tcp-mss 1379
        set remote-ip 169.254.45.89
        set mtu 1427
        set interface "wan1"
    next
end
```

To configure BGP for the first VPN tunnel:

BGP is used within the tunnel to exchange prefixes between the virtual private gateway and your FortiGate. The virtual private gateway announces the prefix according to your VPC.

The local BGP autonomous system number (ASN) (65000) is configured as part of your FortiGate. If you must change the ASN, you must recreate the FortiGate and VPN connection with AWS.

Your FortiGate may announce a default route (0.0.0.0/0) to AWS. This is done using a prefix list and route map in FortiOS.

```
config router bgp
    set as 65000
    config neighbor
        edit 169.254.45.89
            set remote-as 64512
        next
    end
end
config router bgp
    config neighbor
        edit 169.254.45.89
            set capability-default-originate enable
        next
    end
end
config router prefix-list
    edit "default_route"
        config rule
            edit 1
                set prefix 0.0.0.0 0.0.0.0
            next
        end
    next
end
config router route-map
    edit "routemap1"
```

```
        config rule
            edit 1
                set match-ip-address "default_route"
            next
        end
    next
end
```

To advertise additional prefixes to the Amazon VPC, add these prefixes to the network statement and identify the prefix you want to advertise. Ensure that the prefix is present in the routing table of the device with a valid next-hop. If you want to advertise 192.168.0.0/16 to Amazon, you would do the following:

```
config router bgp
    config network
        edit 1
            set prefix 192.168.0.0 255.255.0.0
        next
    end
end
```

To configure firewall policies for the first VPN tunnel:

Create a firewall policy permitting traffic from your local subnet to the VPC subnet, and vice-versa.

This example policy permits all traffic from the local subnet to the VPC. First, view all existing policies using the `show firewall policy` command. Then, create a new firewall policy starting with the next available policy ID. In this example, running `show firewall policy` displayed policies 1, 2, 3, and 4, so you would proceed to create policy 5.

```
config firewall policy
    edit 5
        set srcintf "vpn-07e988ccc1d46f749-0"
        set dstintf internal
        set srcaddr all
        set dstaddr all
        set action accept
        set schedule always
        set service ANY
    next
end
config firewall policy
    edit 5
        set srcintf internal
        set dstintf "vpn-07e988ccc1d46f749-0"
        set srcaddr all
        set dstaddr all
        set action accept
        set schedule always
        set service ANY
    next
end
```

To configure IKE for the second VPN tunnel:

A policy is established for the supported ISAKMP encryption, authentication, DH, lifetime, and key parameters. These sample configurations fulfill the minimum requirements for AES128, SHA1, and DH Group 2. Category VPN connections in the GovCloud AWS region have a minimum requirement of AES128, SHA2, and DH Group 14. To take advantage of AES256, SHA256, or other DH groups such as 14-18, 22, 23, and 24, you must modify these sample configuration files. Higher parameters are only available for VPNs of category "VPN", not for "VPN-Classic".

Your FortiGate's external interface's address must be static. Your FortiGate may reside behind a device performing NAT. To ensure NAT traversal can function, you must adjust your firewall rules to unblock UDP port 4500. If not behind NAT, it is recommended to disable NAT traversal.

Begin configuration in the root VDOM. The interface name must be shorter than 15 characters. It is best if the name is shorter than 12 characters. IPsec DPD causes periodic messages to be sent to ensure a security association remains operational.

```
config vpn ipsec phase1-interface
    edit vpn-07e988ccc1d46f749-1
        set interface "wan1"
        set dpd enable
        set local-gw 35.170.66.108
        set dhgrp 2
        set proposal aes128-sha1
        set keylife 28800
        set remote-gw 100.25.187.58
        set psksecret IjFzyDneUtDdAT4RNmQ85apUG3y4Akre
        set dpd-retryinterval 10
    next
end
```

To configure IPsec for the second VPN tunnel:

The IPsec transform set defines the encryption, authentication, and IPsec mode parameters.

```
config vpn ipsec phase2-interface
    edit "vpn-07e988ccc1d46f749-1"
        set phase1name "vpn-07e988ccc1d46f749-1"
        set proposal aes128-sha1
        set dhgrp 2
        set pfs enable
        set keylifeseconds 3600
    next
end
```

To configure the tunnel interface for the second VPN tunnel:

You must configure a tunnel interface as the logical interface associated with the tunnel. All traffic routed to the tunnel interface must be encrypted and transmitted to the VPC. Similarly, traffic from the VPC will be logically received on this interface.

You must configure the interface's address with your FortiGate's address. If the address changes, you must recreate the FortiGate and VPN connection with Amazon VPC.

The `tcp-mss` option causes the router to reduce the TCP packets' maximum segment size to prevent packet fragmentation.

```
config system interface
    edit "vpn-07e988ccc1d46f749-1"
        set vdom "root"
        set ip 169.254.44.162 255.255.255.255
        set allowaccess ping
        set type tunnel
        set tcp-mss 1379
        set remote-ip 169.254.44.161
        set mtu 1427
        set interface "wan1"
    next
end
```

To configure BGP for the second VPN tunnel:

BGP is used within the tunnel to exchange prefixes between the virtual private gateway and your FortiGate. The virtual private gateway announces the prefix according to your VPC.

The local BGP ASN (65000) is configured as part of your FortiGate. If you must change the ASN, you must recreate the FortiGate and VPN connection with AWS.

Your FortiGate may announce a default route (0.0.0.0/0) to AWS. This is done using a prefix list and route map in FortiOS.

```
config router bgp
    set as 65000
    config neighbor
        edit 169.254.44.161
            set remote-as 64512
        next
    end
end
config router bgp
    config neighbor
        edit 169.254.44.161
            set capability-default-originate enable
        next
    end
end
config router prefix-list
    edit "default_route"
        config rule
            edit 1
                set prefix 0.0.0.0 0.0.0.0
            next
        end
    next
end
config router route-map
    edit "routemap1"
```

```
        config rule
            edit 1
                set match-ip-address "default_route"
            next
        end
    next
end
```

To advertise additional prefixes to the Amazon VPC, add these prefixes to the network statement and identify the prefix you want to advertise. Ensure that the prefix is present in the routing table of the device with a valid next-hop. If you want to advertise 192.168.0.0/16 to Amazon, you would do the following:

```
config router bgp
    config network
        edit 1
            set prefix 192.168.0.0 255.255.0.0
        next
    end
end
```

To configure firewall policies for the second VPN tunnel:

Create a firewall policy permitting traffic from your local subnet to the VPC subnet, and vice-versa.

This example policy permits all traffic from the local subnet to the VPC. First, view all existing policies using the `show firewall policy` command. Then, create a new firewall policy starting with the next available policy ID. In this example, running `show firewall policy` displayed policies 1, 2, 3, 4, and 5, so you would proceed to create policy 6.

```
config firewall policy
    edit 6
        set srcintf "vpn-07e988ccc1d46f749-1"
        set dstintf internal
        set srcaddr all
        set dstaddr all
        set action accept
        set schedule always
        set service ANY
    next
end
config firewall policy
    edit 6
        set srcintf internal
        set dstintf "vpn-07e988ccc1d46f749-1"
        set srcaddr all
        set dstaddr all
        set action accept
        set schedule always
        set service ANY
    next
end
```

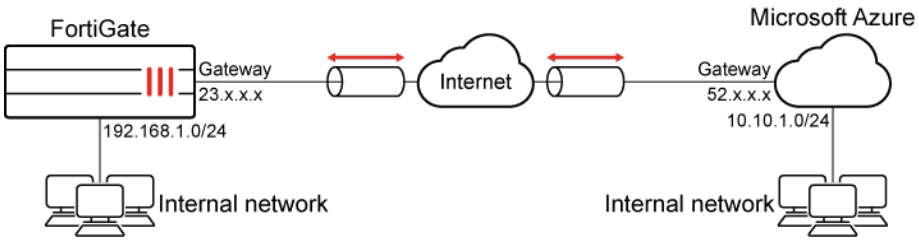
IPsec VPN to Azure with virtual network gateway

This example shows how to configure a site-to-site IPsec VPN tunnel to Microsoft Azure. It shows how to configure a tunnel between each site, avoiding overlapping subnets, so that a secure tunnel can be established.

Prerequisites

- A FortiGate with an Internet-facing IP address
- A valid Microsoft Azure account

Sample topology



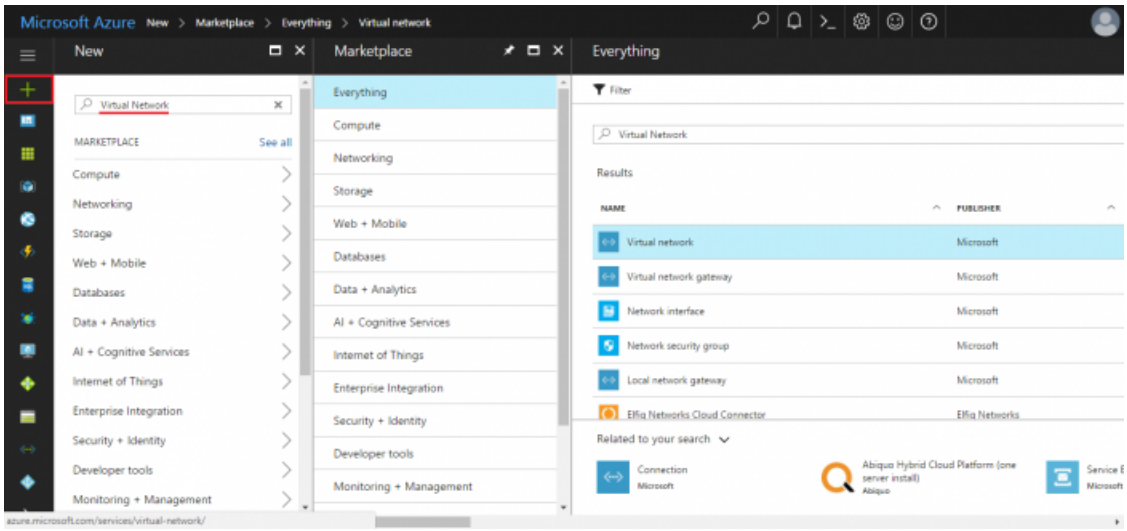
Sample configuration

This sample configuration shows how to:

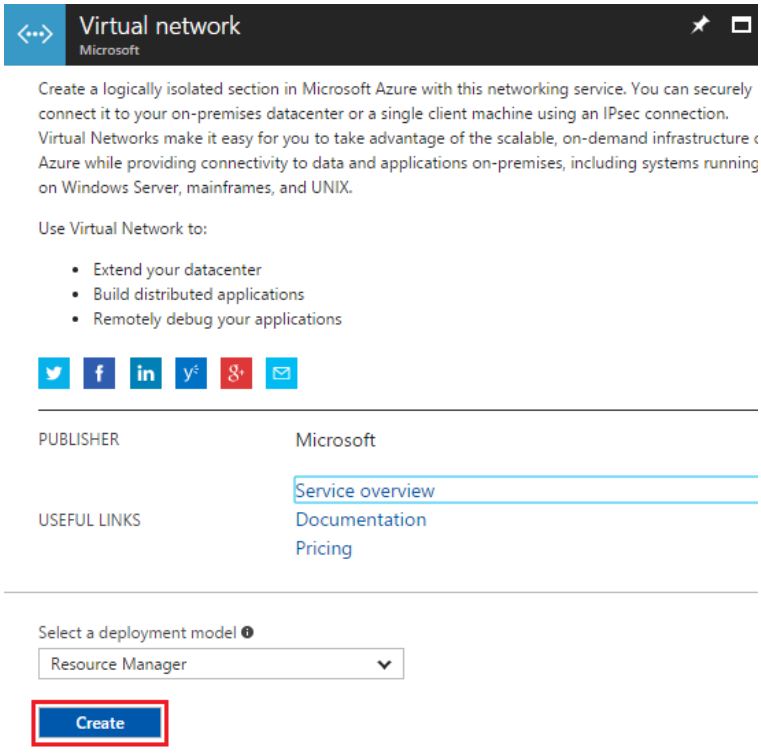
1. Configure an Azure virtual network
2. Specify the Azure DNS server
3. Configure the Azure virtual network gateway
4. Configure the Azure local network gateway
5. Configure the FortiGate tunnel
6. Create the Azure firewall object
7. Create the FortiGate firewall policies
8. Create the FortiGate static route
9. Create the Azure site-to-site VPN connection
10. Check the results

To configure an Azure virtual network:

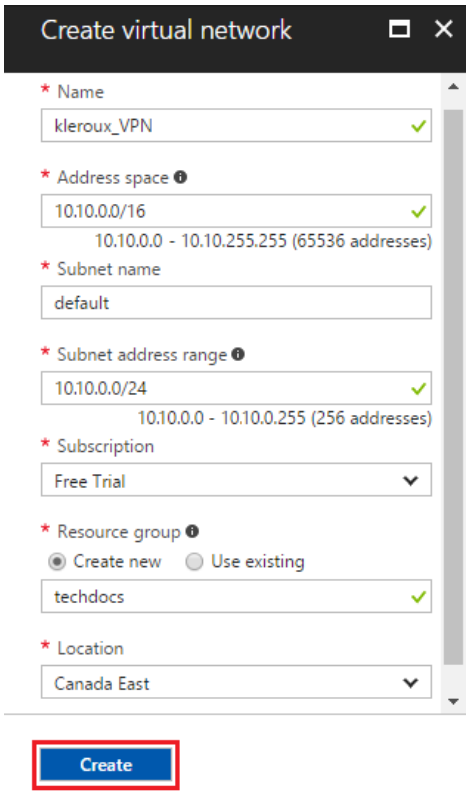
1. Log in to Azure and click *New*.
2. In *Search the Marketplace*, type *Virtual network*.
3. Click *Virtual network* to open the *Virtual network* pane.



4. At the bottom of the *Virtual network* pane, click the *Select a deployment model* dropdown list and select *Resource Manager*.
5. Click *Create*.

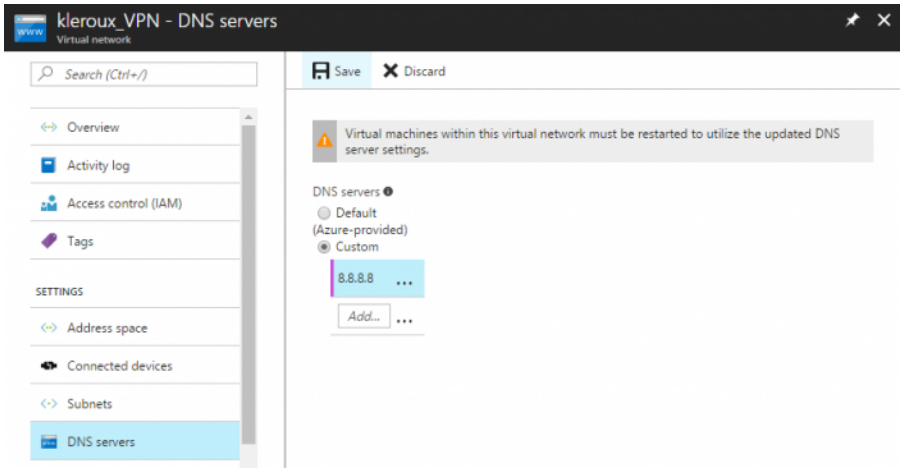


6. On the *Create virtual network* pane, enter you virtual network settings, and click *Create*.



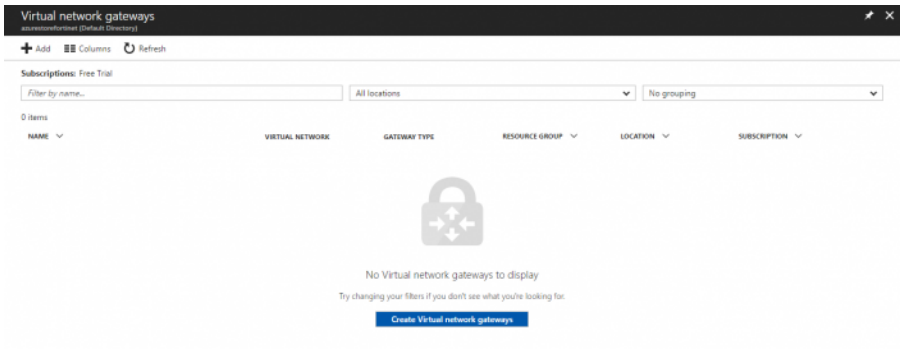
To specify the Azure DNS server:

1. Open the virtual network you just created.
2. Click *DNS servers* to open the *DNS servers* pane.
3. Enter the IP address of the DNS server and click *Save*.

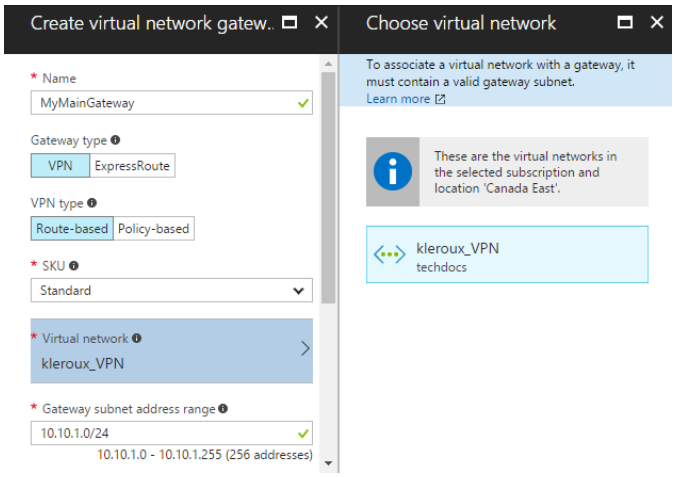


To configure the Azure virtual network gateway:

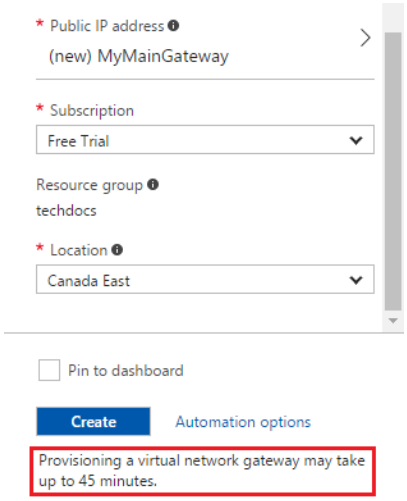
1. In the portal dashboard, go to *New*.
2. Search for *Virtual Network Gateway* and click it to open the *Virtual network gateway* pane.



3. Click *Create Virtual network gateways* and enter the settings for your virtual network gateway.

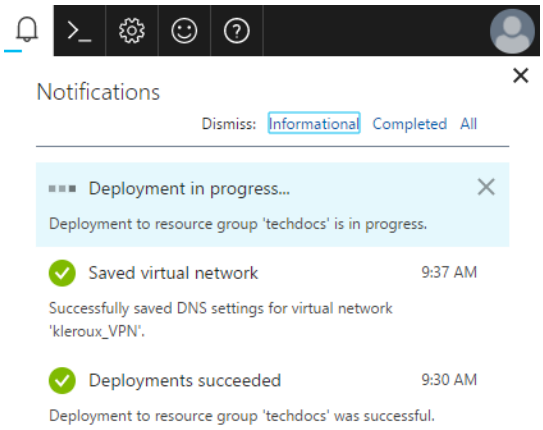


4. If needed, create a Public IP address.



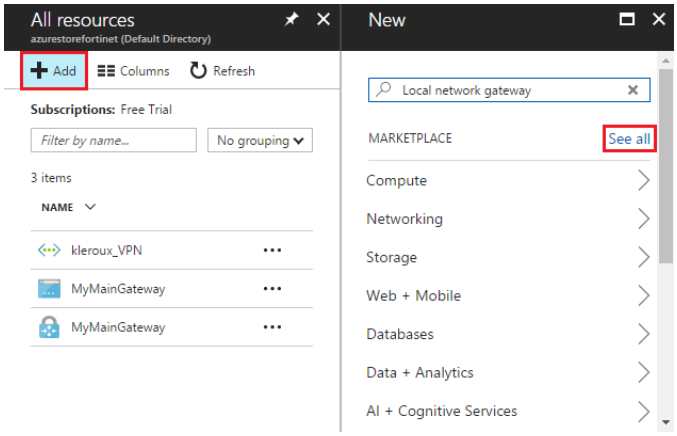
5. Click *Create*.

   Creating the virtual network gateway might take some time. When the provisioning is done, you'll receive a notification.

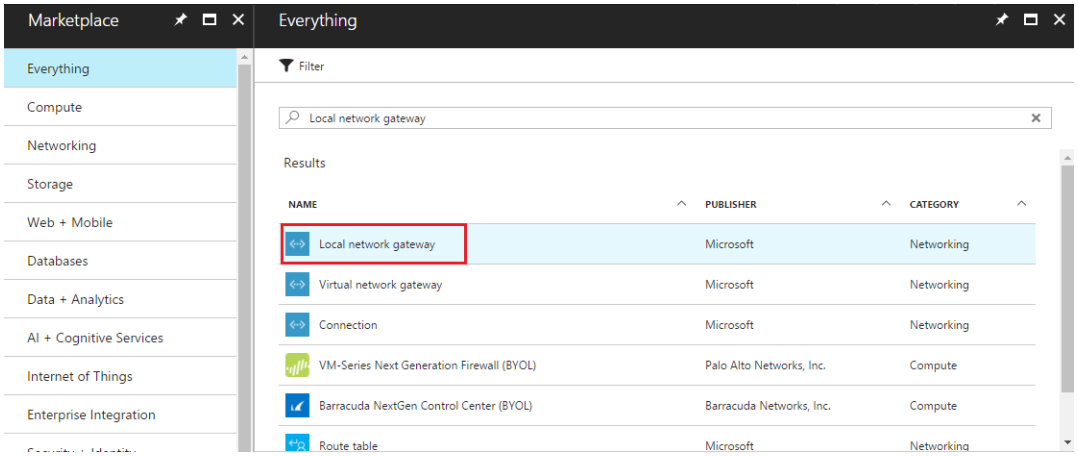


To configure the Azure local network gateway:

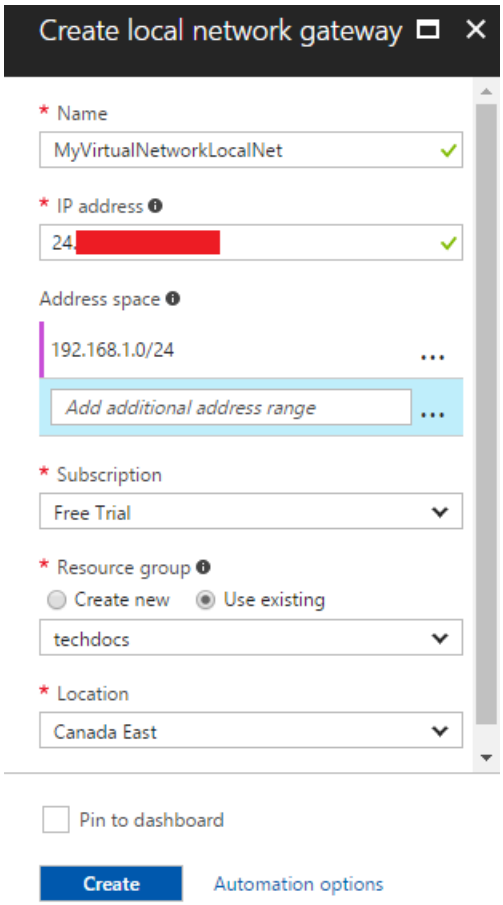
1. In the portal dashboard, click *All resources*.
2. Click *Add* and then click *See all*.



3. In the *Everything* pane, search for *Local network gateway* and then click *Create local network gateway*.



4. For the *IP address*, enter the local network gateway IP address, that is, the FortiGate's external IP address.



5. Set the remaining values for your local network gateway and click *Create*.

To configure the FortiGate tunnel:

1. In the FortiGate, go to *VPN > IP Wizard*.
2. Enter a *Name* for the tunnel, click *Custom*, and then click *Next*.
3. Configure the *Network* settings.
   a. For *Remote Gateway*, select *Static IP Address* and enter the IP address provided by Azure.
   b. For *Interface*, select *wan1*.
   c. For *NAT Traversal*, select *Disable*,
   d. For *Dead Peer Detection*, select *On Idle*.
   e. In the Authentication section, select
4. Configure the *Authentication* settings.
   a. For *Method*, select *Pre-shared Key* and enter the *Pre-shared Key*.
   b. For *IKE*, select *2*.
5. Configure the *Phase 1 Proposal* settings.
   a. Set the Encryption and Authentication combination to the three supported encryption algorithm combinations accepted by Azure.
      - AES256 and SHA1
      - 3DES and SHA1

    - AES256 and SHA256
  b. For *Diffie-Hellman Groups*, select *2*.
  c. Set *Key Lifetime (seconds)* to *28800*.
6. In *Phase 2 Selectors*, expand the *Advanced* section to configure the *Phase 2 Proposal* settings.
  a. Set the Encryption and Authentication combinations:
    - AES256 and SHA1
    - 3DES and SHA1
    - AES256 and SHA256
  b. Uncheck *Enable Perfect Forward Secrecy (PFS)*.
  c. Set *Key Lifetime (seconds)* to *27000*.
7. Click *OK*.

To create the Azure firewall object:

1. In the FortiGate, go to *Policy & Objects > Addresses*.
2. Create a firewall object for the Azure VPN tunnel.

To create the FortiGate firewall policies:

1. In the FortiGate, go to *Policy & Objects > Firewall Policy*.
2. Create a policy for the site-to-site connection that allows outgoing traffic.
  a. Set the *Source* address and *Destination* address using the firewall objects you just created.
  b. Disable *NAT*.
3. Create another policy that allows incoming traffic.
  a. For this policy, reverse the *Source* address and *Destination* address.
4. We recommend limiting the TCP maximum segment size (MSS) being sent and received so as to avoid packet drops and fragmentation.

  To do this, use the following CLI commands on both policies.

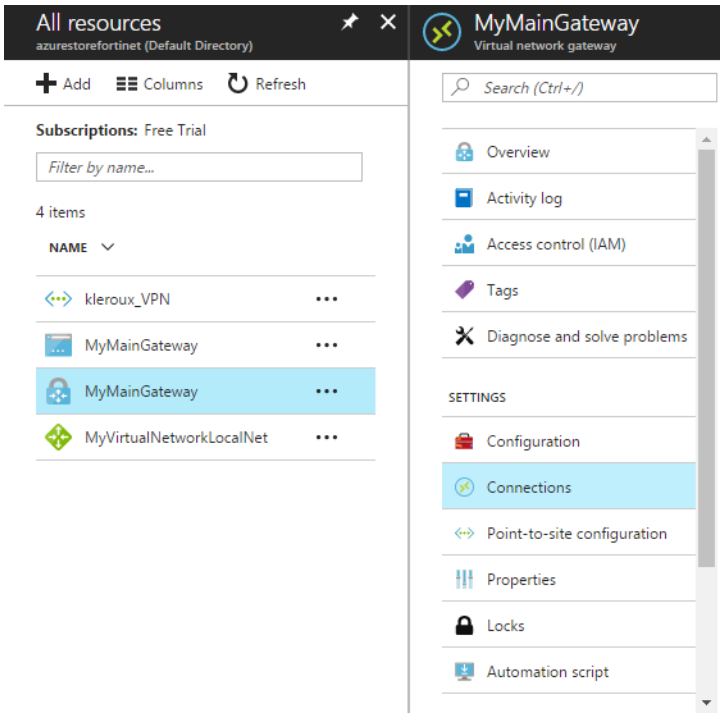
```
config firewall policy
   edit <policy-id>
      set tcp-mss-sender 1350
      set tcp-mss-receiver 1350
   next
end
```

To create the FortiGate static route:

1. In the FortiGate, go to *Network > Static Routes*.
2. Create an IPv4 Static Route that forces outgoing traffic going to Azure to go through the route-based tunnel.
3. Set the *Administrative Distance* to a value lower than the existing default route value.

To create the Azure site-to-site VPN connection:

1. In the Azure portal, locate and select your virtual network gateway.
2. In the *Settings* pane, click *Connections* and then click *Add*.

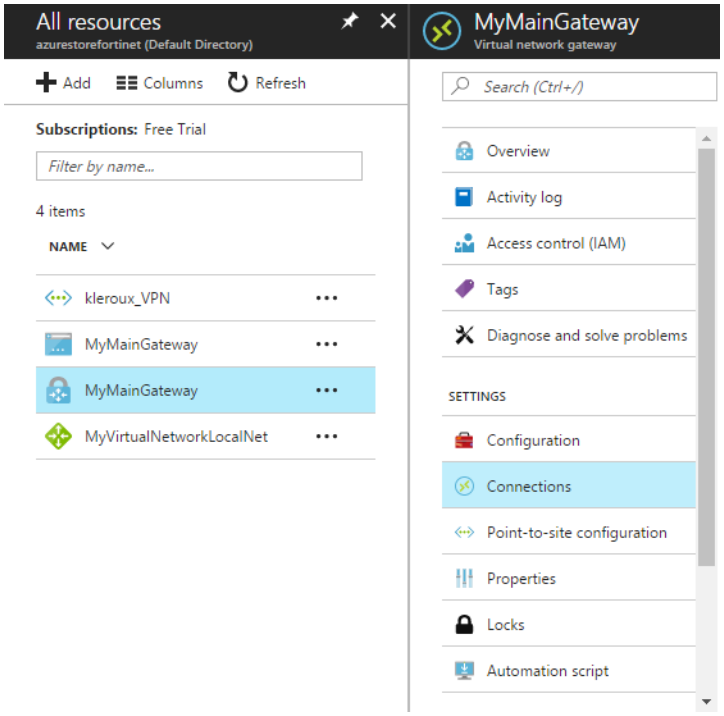


3. Enter the settings for your connection. Ensure the *Shared Key (PSK)* matches the *Pre-shared Key* for the FortiGate tunnel.

To check the results:

1. In the FortiGate, go to *Monitor* > *IPsec Monitor* and check that the tunnel is up. If the tunnel is down, right-click the tunnel and select *Bring Up*.
2. In the FortiGate, go to *Log & Report* > *System Events*.
   a. Select an event card to view more information and verify the connection.

3. In the Azure portal dashboard, click *All resources* and locate your virtual network gateway.
   a. In your virtual network gateway pane, click *Connections* to see the status of each connection.



   b. Click a connection to open the *Essentials* pane to view more information about that connection.
      - If the connection is successful, the *Status* shows *Connected*.
      - See the *ingress* and *egress* bytes to confirm traffic flowing through the tunnel.

IPsec VPN to an Azure with virtual WAN

This is a sample configuration of an IPsec site-to-site VPN connection between an on-premise FortiGate and an Azure virtual network (VNet). This example uses Azure virtual WAN (vWAN) to establish the VPN connection.

- Azure must use IPsec v2 for this configuration.
- Azure uses overlapped subnet IP addresses for the IPsec interfaces.

To configure IKEv2 IPsec site-to-site VPN to an Azure VPN gateway:

1. In the Azure management portal, configure vWAN-related settings as described in Tutorial: Create a Site-to-Site connection using Azure Virtual WAN.
   If a custom BGP IP address is configured on Azure's vWAN, such as 169.254.21.6 and 169.254.21.7, you must configure the FortiGate `remote-IP` to the corresponding *Custom BGP IP Address* value. If a custom BGP IP address is not configured, FortiGate `remote-IPs` should point to the *Default BGP IP Address* value.
2. Download the VPN configuration. The following shows an example VPN configuration:

```
[ {"configurationVersion":{"LastUpdatedTime":"2019-07-16T22:16:28.0409002Z","Version":"be5c5787-
     b903-43b1-a237-49eae1b373e4"},"vpnSiteConfiguration":
     {"Name":"toaws","IPAddress":"3.220.252.93","BgpSetting":
     {"Asn":7225,"BgpPeeringAddress":"169.254.24.25","PeerWeight":32768},"LinkName":"toaws"},"v
     pnSiteConnections":[{"hubConfiguration":{"AddressSpace":"10.1.0.0/16","Region":"West
     US","ConnectedSubnets":["10.2.0.0/16"]},"gatewayConfiguration":{"IpAddresses":
     {"Instance0":"52.180.90.47","Instance1":"52.180.89.94"},"BgpSetting":
     {"Asn":65515,"BgpPeeringAddresses":
     {"Instance0":"10.1.0.7","Instance1":"10.1.0.6"},"PeerWeight":0}},"connectionConfiguration":
     {"IsBgpEnabled":true,"PSK":"Fortinet123#","IPsecParameters":
     {"SADataSizeInKilobytes":102400000,"SALifeTimeInSeconds":3600}}}]} ]
```


3. Configure the following on the FortiGate. Note for `set proposal`, you can select from several proposals.

```
config vpn ipsec phase1-interface
    edit "toazure1"
        set interface "port1"
        set ike-version 2
        set keylife 28800
        set peertype any
        set proposal aes256-sha1
        set dhgrp 2
        set remote-gw 52.180.90.47
        set psksecret **********
    next
    edit "toazure2"
        set interface "port1"
        set ike-version 2
        set keylife 28800
        set peertype any
        set proposal aes256-sha1
        set dhgrp 2
        set remote-gw 52.180.89.94
        set psksecret **********
    next
end
config vpn ipsec phase2-interface
    edit "toazure1"
        set phase1name "toazure1"
        set proposal aes256-sha1
        set dhgrp 2
        set keylifeseconds 3600
    next
    edit "toazure2"
        set phase1name "toazure2"
        set proposal aes256-sha1
        set dhgrp 2
        set keylifeseconds 3600
    next
end
config system settings
    set allow-subnet-overlap enable
end
config system interface
    edit "toazure1"
```

```
        set vdom "root"
        set ip 169.254.24.25 255.255.255.255
        set type tunnel
        set remote-ip 10.1.0.7 255.255.255.255
        set snmp-index 4
        set interface "port1"
    next
    edit "toazure2"
        set vdom "root"
        set ip 169.254.24.25 255.255.255.255
        set type tunnel
        set remote-ip 10.1.0.6 255.255.255.255
        set snmp-index 5
        set interface "port1"
    next
end
config router bgp
    set as 7225
    set router-id 169.254.24.25
    config neighbor
        edit "10.1.0.7"
            set remote-as 65515
        next
        edit "10.1.0.6"
            set remote-as 65515
        next
    end
    config network
        edit 1
            set prefix 172.30.101.0 255.255.255.0
        next
    end
    config redistribute "connected"
        set status enable
    end
    config redistribute "rip"
    end
    config redistribute "ospf"
    end
    config redistribute "static"
    end
    config redistribute "isis"
    end
    config redistribute6 "connected"
    end
    config redistribute6 "rip"
    end
    config redistribute6 "ospf"
    end
    config redistribute6 "static"
    end
```

```
    config redistribute6 "isis"
    end
end
```

4. Run `diagnose vpn tunnel list`. If the configuration was successful, the output should resemble the following:

```
name=toazure1 ver=2 serial=3 172.30.1.83:4500->52.180.90.47:4500 tun_id=52.180.90.47
bound_if=3 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/0
proxyid_num=1 child_num=0 refcnt=15 ilast=16 olast=36 ad=/0
stat: rxp=41 txp=41 rxb=5104 txb=2209
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=1
natt: mode=keepalive draft=0 interval=10 remote_port=4500
proxyid=toazure1 proto=0 sa=1 ref=2 serial=4
  src: 0:0.0.0.0/0.0.0.0:0
  dst: 0:0.0.0.0/0.0.0.0:0
  SA:  ref=3 options=10226 type=00 soft=0 mtu=8926 expire=2463/0B replaywin=2048
       seqno=2a esn=0 replaywin_lastseq=00000029 itn=0
  life: type=01 bytes=0/0 timeout=3300/3600
  dec: spi=c13f7928 esp=aes key=32
009a86bb0d6f5fee66af7b8232c8c0f22e6ec5c61ba19c93569bd0cd115910a9
       ah=sha1 key=20 f05bfeb0060afa89d4afdfac35960a8a7a4d4856
  enc: spi=b40a6c70 esp=aes key=32
a1e361075267ba72b39924c5e6c766fd0b08e0548476de2792ee72057fe60d1d
       ah=sha1 key=20 b1d24bedb0eb8fbd26de3e7c0b0a3a799548f52f
  dec:pkts/bytes=41/2186, enc:pkts/bytes=41/5120
------------------------------------------------------
name=toazure2 ver=2 serial=4 172.30.1.83:4500->52.180.89.94:4500 tun_id=52.180.89.94
bound_if=3 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/0
proxyid_num=1 child_num=0 refcnt=16 ilast=16 olast=16 ad=/0
stat: rxp=40 txp=40 rxb=4928 txb=2135
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=1
natt: mode=keepalive draft=0 interval=10 remote_port=4500
proxyid=toazure2 proto=0 sa=1 ref=2 serial=4
  src: 0:0.0.0.0/0.0.0.0:0
  dst: 0:0.0.0.0/0.0.0.0:0
  SA:  ref=3 options=10626 type=00 soft=0 mtu=8926 expire=2427/0B replaywin=2048
       seqno=29 esn=0 replaywin_lastseq=00000028 itn=0
  life: type=01 bytes=0/0 timeout=3299/3600
  dec: spi=c13f791d esp=aes key=32
759898cbb7fafe448116b1fb0fb6d2f0eb99621ea6ed8dd4417ffdb901eb82be
       ah=sha1 key=20 533ec5dc8a1910221e7742b12f9de1b41205622c
  enc: spi=67934bfe esp=aes key=32
9b5710bfb4ba784722241ec371ba8066629febcd75da6f8471915bdeb874ca80
       ah=sha1 key=20 5099fed7edac2b960294094f1a8188ab42f34d7b
  dec:pkts/bytes=40/2087, enc:pkts/bytes=40/4976



Routing table for VRF=0
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
```

```
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       * - candidate default

S*      0.0.0.0/0 [5/0] via 172.30.1.1, port1
B       10.1.0.0/16 [20/0] via 10.1.0.6, toazure2, 00:15:01
C       10.1.0.6/32 is directly connected, toazure2
C       10.1.0.7/32 is directly connected, toazure1
B       10.2.0.0/16 [20/0] via 10.1.0.6, toazure2, 00:15:01
C       169.254.24.25/32 is directly connected, toazure1
                         is directly connected, toazure2
C       172.30.1.0/24 is directly connected, port1
C       172.30.101.0/24 is directly connected, port2
```

IPSec VPN between a FortiGate and a Cisco ASA with multiple subnets

When a Cisco ASA unit has multiple subnets configured, multiple phase 2 tunnels must be created on the FortiGate to allocate to each subnet (rather than having multiple subnets on one phase 2 tunnel).

The FortiGate uses the same SPI value to bring up the phase 2 negotiation for all of the subnets, while the Cisco ASA expects different SPI values for each of its configured subnets. Using multiple phase 2 tunnels on the FortiGate creates different SPI values for each subnet.

To configure multiple phase 2 interfaces in route-based mode:

```
config vpn ipsec phase2-interface
    edit "First subnet"
        set phase1name "VPN to Cisco"
        set src-subnet 192.168.227.253 255.255.255.255
        set dst-subnet 10.142.0.0 255.255.254.0
    next
    edit "Second subnet"
        set phase1name "VPN to Cisco"
        set src-subnet 192.168.227.253 255.255.255.255
        set dst-subnet 10.143.0.0 255.255.254.0
    next
end
```

To configure multiple phase 2 interfaces in policy-based mode:

```
config vpn ipsec phase2
    edit "First subnet"
        set phase1name "VPN to Cisco"
        set src-subnet 192.168.227.253 255.255.255.255
        set dst-subnet 10.142.0.0 255.255.254.0
    next
    edit "Second subnet"
```

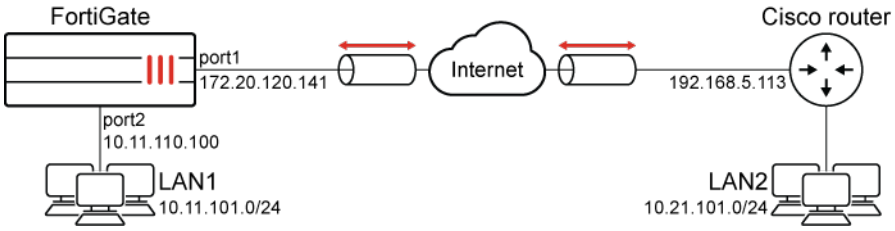
```
        set phase1name "VPN to Cisco"
        set src-subnet 192.168.227.253 255.255.255.255
        set dst-subnet 10.143.0.0 255.255.254.0
    next
end
```

Cisco GRE-over-IPsec VPN

This is a sample configuration of a FortiGate VPN that is compatible with Cisco-style VPNs that use GRE in an IPsec tunnel. Cisco products with VPN support often use the GRE protocol tunnel over IPsec encryption. Cisco VPNs can use either transport mode or tunnel mode IPsec.

Topology

In this example, LAN1 users are provided with access to LAN2.



Configuring the FortiGate

There are five steps to configure GRE-over-IPsec with a FortiGate and Cisco router:

1. Enable overlapping subnets.
2. Configure a route-based IPsec VPN on the external interface.
3. Configure a GRE tunnel on the virtual IPsec interface.
4. Configure security policies.
5. Configure the static route.

Enabling overlapping subnets

Overlapping subnets are required because the IPsec and GRE tunnels will use the same addresses. By default, each FortiGate network interface must be on a separate network. This configuration assigns an IPsec tunnel endpoint and the external interface to the same network.

To enable overlapping subnets:

```
config system settings
    set allow-subnet-overlap enable
end
```

Configuring a route-based IPsec VPN

A route-based VPN that use encryption and authentication algorithms compatible with the Cisco router is required. Pre-shared key authentication is used in this configuration.

To configure route-based IPsec in the GUI:

1. Go to *VPN* > *IPsec Wizard* and select the *Custom* template.
2. Enter the tunnel name (*tocisco*) and click *Next*.
3. Enter the following:

Remote Gateway	Static IP Address
IP Address	Cisco router public interface (192.168.5.113)
Interface	FortiGate public interface (172.20.120.141)
Authentication Method	Pre-shared Key
Pre-shared Key	Entry must match the pre-shared key on the Cisco router
Mode	Main (ID Protection)
Phase 1 Proposal	3DES-SHA1, AES128-SHA1 (at least one proposal must match the settings on the Cisco router)
Local Address	GRE local tunnel endpoint IP address (172.20.120.141)
Remote Address	GRE remote tunnel endpoint IP address (192.168.5.113)
Phase 2 Proposal	3DES-MD5 (at least one proposal must match the settings on the Cisco router)
Local Port	0
Remote Port	0
Protocol	47

4. Click *OK*.
5. If the Cisco router is configured to use transport mode IPsec, configure transport mode on the FortiGate:

```
config vpn phase2-interface
    edit tocisco_p2
        set encapsulation transport-mode
    next
end
```

To configure route-based IPsec in the CLI:

```
config vpn ipsec phase1-interface
    edit tocisco
        set interface port1
        set proposal 3des-sha1 aes128-sha1
        set remote-gw 192.168.5.113
```

```
        set psksecret xxxxxxxxxxxxxxx
    next
end
```

```
config vpn ipsec phase2-interface
    edit tocisco_p2
        set phase1name tocisco
        set proposal 3des-md5
        set encapsulation [tunnel-mode | transport-mode]
        set protocol 47
        set src-addr-type ip
        set dst-start-ip 192.168.5.113
        set src-start-ip 172.20.120.141
    next
end
```

To add the IPsec tunnel end addresses:

```
config system interface
    edit tocisco
        set ip 172.20.120.141 255.255.255.255
        set remote-ip 192.168.5.113
    next
end
```

Configuring the GRE tunnel

The local gateway and remote gateway addresses must match the local and remote gateways of the IPsec tunnel. The GRE tunnel runs between the virtual IPsec public interface on the FortiGate unit and the Cisco router.

To configure the GRE tunnel:

```
config system gre-tunnel
    edit gre1
        set interface tocisco
        set local-gw 172.20.120.141
        set remote-gw 192.168.5.113
        set keepalive-interval <integer>
        set keepalive-failtimes <integer>
    next
end
```

The Cisco router configuration requires an address for its end of the GRE tunnel, so you need to add the tunnel end addresses.

To add the tunnel end addresses:

```
config system interface
    edit gre1
        set ip 10.0.1.1 255.255.255.255
```

```
        set remote-ip 10.0.1.2
    next
end
```

Configuring the security policies

Two sets of security policies are required:

- Policies to allow traffic to pass in both directions between the GRE virtual interface and the IPsec virtual interface.
- Policies to allow traffic to pass in both directions between the protected network interface and the GRE virtual interface.

To configure security policies in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
2. Enter the following to allow traffic between the protected network and the GRE tunnel:

Name	LANtoGRE
Incoming Interface	Interface that connects to the private network behind the FortiGate (port2)
Outgoing Interface	GRE tunnel virtual interface (gre1)
Source	All
Destination	All
Action	ACCEPT
NAT	Disable

3. Click *OK*.
4. Create a new policy and enter the following to allow traffic between the GRE tunnel and the protected network:

Name	GREtoLAN
Incoming Interface	GRE tunnel virtual interface (gre1)
Outgoing Interface	Interface that connects to the private network behind the FortiGate (port2)
Source	All
Destination	All
Action	ACCEPT
NAT	Disable

5. Click *OK*.
6. Create a new policy and enter the following to allow traffic between the GRE virtual interface and the IPsec virtual interface:

Name	GREtoIPsec
Incoming Interface	GRE tunnel virtual interface (gre1)
Outgoing Interface	Virtual IPsec interface (tocisco)
Source	All
Destination	All
Action	ACCEPT
NAT	Disable

7. Click *OK*.
8. Create a new policy and enter the following to allow traffic between the IPsec virtual interface and the GRE virtual interface:

Name	IPsectoGRE
Incoming Interface	Virtual IPsec interface (tocisco)
Outgoing Interface	GRE tunnel virtual interface (gre1)
Source	All
Destination	All
Action	ACCEPT
NAT	Disable

9. Click *OK*.

To configure security policies in the CLI:

```
config firewall policy
    edit 1
        set name LANtoGRE
        set srcintf port2
        set dstintf gre1
        set srcaddr all
        set dstaddr all
        set action accept
        set schedule always
        set service ALL
    next
    edit 2
        set name GREtoLAN
        set srcintf gre1
        set dstintf port2
        set srcaddr all
        set dstaddr all
        set action accept
        set schedule always
        set service ALL
```

```
    next
    edit 3
        set name GREtoIPsec
        set srcintf gre1
        set dstintf tocisco
        set srcaddr all
        set dstaddr all
        set action accept
        set schedule always
        set service ALL
    next
    edit 4
        set name IPsectoGRE
        set srcintf tocisco
        set dstintf gre1
        set srcaddr all
        set dstaddr all
        set action accept
        set schedule always
        set service ALL
    next
end
```

Configuring routing

to direct traffic destined for the network behind the Cisco router into the GRE-over-IPsec tunnelTraffic destined for the network behind the Cisco router must be routed to the GRE tunnel. To do this, create a static route

To create the static route in the GUI:

1. Go to *Network* > *Static Routes* and click *Create New*.
2. Enter the following:

Destination	IP and netmask for the network behind the Cisco router (10.21.101.0 255.255.255.0)
Interface	GRE tunnel virtual interface (gre1)
Administrative Distance	Leave the default setting

3. Click *OK*.

To create the static route in the CLI:

```
config router static
    edit 0
        set device gre1
        set dst 10.21.101.0 255.255.255.0
    next
end
```

Configuring the Cisco router

For more information, refer to Configuring and verifying a GRE over IPsec tunnel in the Fortinet Knowledge Base.

In this type of deployment where you configure GRE over IPsec VPN, NP7 offloading is not natively supported, even though it is supported individually for GRE and for IPsec.

To workaround this, configure two separate VDOMs where one performs IPsec tunneling and the other performs GRE encapsulation.

For more information, see the following KB: Technical Tip: GRE + IPSec not supported for NP7 offloading.

Cloud SDN Orchestration VPN wizard for AWS NEW

The Cloud SDN Orchestration VPN wizard simplifies configuration of a VPN tunnel between a FortiGate and a virtual private gateway or transit gateway on AWS. When a FortiGate has an SDN connector with the proper permissions established with AWS, the VPN wizard creates the FortiGate VPN configurations and pushes the necessary customer gateway and VPN tunnel configurations to AWS under the configured VPC. This reduces the chance of mis-configurations and the number of steps needed to configure a VPN tunnel.

See FortiGate AWS cloud orchestration VPN TGW for details.

Remote access

Remote access allows users to connect to the office network over VPN for remote work from home scenarios. This could be for accessing servers and services in the office network, or routing the entire Internet access over the VPN gateway. The latter scenario, often called Secure Internet Access (SIA), allows remote workers to secure the Internet traffic through the corporate firewall which will perform standard NGFW inspection on the traffic.

The following topics provide instructions on configuring remote access:

- FortiGate as dialup client on page 2962
- FortiClient as dialup client on page 2973
- Add FortiToken multi-factor authentication on page 2985
- Add LDAP user authentication on page 2988
- Configure FortiClient SIA for IPsec VPN tunnels on page 2993
- Remote access VPNs with IPAM addressing New on page 3003
- LDAP authentication with TCP as transport on page 3007
- VPN 2FA with EAP and certificate authentication on page 3019
- Using single or multiple user groups for user authentication on page 3024
- VPN with FortiClient Standalone and FortiIdentity Cloud New on page 3026
- Native VPN as dialup client - Windows New on page 3043
- Native VPN as dialup client - iOS on page 3063
- Windows IKEv2 native VPN with user certificate on page 3070

- Dialup VPN with certificate authentication on page 3094
- SAML-based authentication for FortiClient remote access VPNs on page 3124
- Enhancing IPsec security using EMS SN verification on page 3146
- IKE Mode Config clients on page 3146
- IPsec DNS suffix on page 3153
- Defining gateway IP addresses in IPsec with mode-config and DHCP on page 3158
- Dialup IPsec VPN using custom TCP port on page 3159
- Matching IPsec tunnel gateway based on address parameters on page 3172
- Enforcing security posture verification before dial-up VPN connection on page 3174

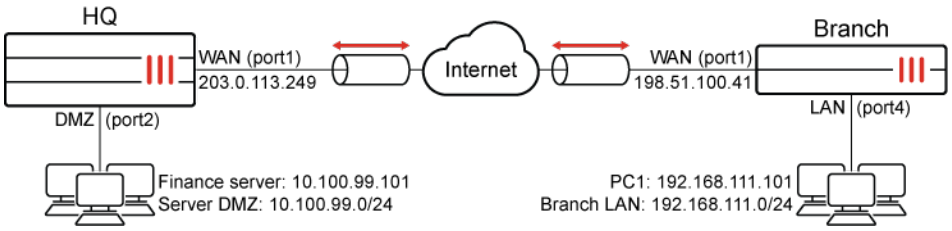
FortiGate as dialup client

A FortiGate configured as a dial-up client initiates an IPsec VPN connection to a remote IPsec VPN gateway while using a dynamically assigned WAN IP address, making it useful in networks with no static IP availability. It is also suitable for environments where the client FortiGate is located behind a NAT device.

The dial-up VPN significantly reduces configuration complexity and overhead when deploying new branches. Because the IPsec VPN server or IPsec VPN hub is preconfigured to accept dynamic IPsec connections, there is no need to reconfigure it for each new branch FortiGate deployment. This makes scaling easier, allowing organizations to add new sites without delays.

In this example, users in the Branch LAN subnet require connectivity with the HQ DMZ subnet to access servers over VPN. The IP address of the Branch's Internet (port1) interface is assigned dynamically using DHCP.

The Branch FortiGate is configured as a dialup VPN client to authenticate and connect to the HQ FortiGate. For mutual authentication, the default Fortinet_Factory certificates are used. This method is more secure than using a pre-shared key.



After the IPsec VPN tunnel is successfully established, PC1 (192.168.111.101) can ping the Finance server (10.100.99.101).

To configure the HQ FortiGate as an IPsec VPN dialup server using the VPN Wizard:

1. Go to *VPN > VPN Wizard*.
2. Enter a name for the VPN in the *Tunnel name* field. In this example, *vpn-server*
3. Set *Select a template* to *Site to Site*.
4. Click *Begin*.
5. Configure the *VPN tunnel* settings:

Field	Value
Authentication method	*Signature*
Certificate name	Fortinet_Factory
Peer certificate CA	Fortinet_CA
IKE	*Version 2*
Transport	Use default *UDP*. If IKE over TCP is required, enable *Allow VPN negotiation over TCP* in *VPN > VPN Tunnels > Settings* before using the VPN wizard, and then set *Transport* to *Auto* in the wizard.
NAT traversal	*Enable* Set *Keepalive frequency* to *10*.

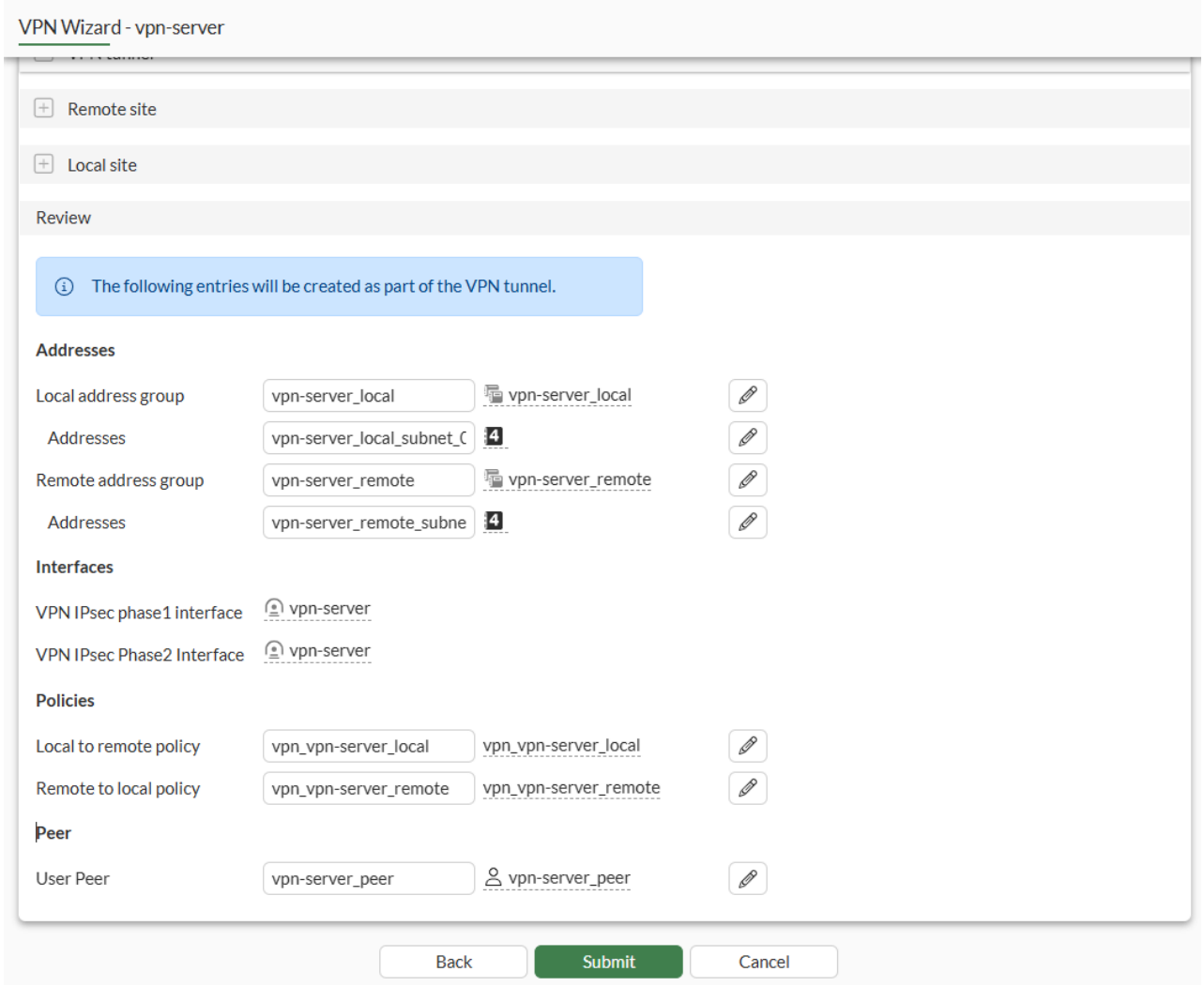
6. Click *Next*.
7. Configure the *Remote site* settings:

Field	Value
Remote site device type	*FortiGate*
Remote site device	*Behind NAT or dynamic*
Remote site subnets that can access VPN	*192.168.111.0/24* (The Branch's LAN subnet)

8. Click *Next*.
9. Configure the *Local site* settings:

Field	Value
Outgoing interface that binds to tunnel	*WAN (port1)* Disable *Create and add interface to zone*.
Local interface	*DMZ (port2)*
Local subnets that can access VPN	This field automatically uses the IP address related to the selected local interface. In this example, 10.100.99.0/24
Allow remote site's internet traffic through this device	Disabled

10. Click *Next*.
11. Review the configuration and, if everything is correct, click *Submit*.



The *vpn-server* IPsec tunnel is configured by the wizard and is visible in *VPN > VPN Tunnels* under *Dialup - FortiGate tunnels*.

To configure Branch FortiGate as IPsec VPN dialup client using the VPN Wizard:

1. Go to *VPN > VPN Wizard*.
2. Enter a name for the VPN in the *Tunnel name* field. In this example, *vpn-client*
3. Set *Select a template* to *Site to Site*.
4. Click *Begin*.
5. Configure the *VPN tunnel* settings:

Field	Value
Authentication method	*Signature*
Certificate name	Fortinet_Factory
Peer certificate CA	Fortinet_CA

Field	Value
IKE	*Version 2*
Transport	Use default *UDP*. If IKE over TCP is required, enable *Allow VPN negotiation over TCP* in *VPN > VPN Tunnels > Settings* before using the VPN wizard, and then set *Transport* to *Auto* in the wizard.
NAT traversal	*Enable* Set *Keepalive frequency* to *10*.

6. Click *Next*.
7. Configure the *Remote site* settings:

Field	Value
Remote site device type	*FortiGate*
Remote site device	*Accessible and static* Set *IP/FQDN* to *203.0.113.249*. The WAN interface IP address of the HQ FortiGate, or its FQDN (the FQDN must resolve to the WAN IP address).
Route this device's internet traffic through the remote site.	Disabled
Remote site subnets that can access VPN	*10.100.99.0/24* (The HQ's DMZ subnet).

8. Click *Next*.
9. Configure the *Local site* settings:

Field	Value
Outgoing interface that binds to tunnel	*WAN (port1)* Disable *Create and add interface to zone*.
Local interface	*LAN (port4)*
Local subnets that can access VPN	This field automatically uses the IP address related to the selected local interface. In this example, 192.168.111.0/24
Allow remote site's internet traffic through this device	Disabled

10. Click *Next*.
11. Review the configuration and, if everything is correct, click *Submit*.
    The *vpn-client* IPsec tunnel is configured by the wizard and is visible under *VPN > VPN Tunnels*.

To configure the HQ FortiGate as an IPsec VPN dialup server in the CLI:

1. Configure the WAN interface and default route.
   The WAN interface is the interface connected to the ISP and it is recommended to configure it with a static IP address to ensure that the IPsec VPN configuration on the Branch stays unchanged if the WAN IP changes on the HQ. The IPsec tunnel is established over the WAN interface.

```
config system interface
    edit "port1"
        set vdom "root"
        set ip 203.0.113.249 255.255.255.0
        set alias "WAN"
    next
end
config router static
    edit 1
        set gateway 203.0.113.250
        set device "port1"
    next
end
```

2. Configure address objects for the HQ server DMZ subnet (10.100.99.0/24) and Branch LAN subnet (192.168.111.0/24):

```
config firewall address
    edit "HQ Server DMZ"
        set subnet 10.100.99.0 255.255.255.0
    next
    edit "Branch LAN"
        set subnet 192.168.111.0 255.255.255.0
    next
end
```

3. Configure the DMZ interface:

```
config system interface
    edit "port2"
        set vdom "root"
        set alias "DMZ"
        set ip 10.100.99.1 255.255.255.0
    next
end
```

4. Create a peer user:

```
config user peer
    edit "vpn-server_peer"
        set ca "Fortinet_CA"
    next
end
```

5. Configure the IPsec phase1-interface:

```
config vpn ipsec phase1-interface
    edit "vpn-server"
        set type dynamic
        set interface "port1"
        set ike-version 2
        set authmethod signature
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set dhgrp 20 21
        set certificate "Fortinet_Factory"
        set peer "vpn-server_peer"
    next
end
```

6. Configure the IPsec phase2-interface:

```
config vpn ipsec phase2-interface
    edit "vpn-server"
        set phase1name "vpn-server"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
        set src-addr-type name
        set dst-addr-type name
        set src-name "HQ Server DMZ"
        set dst-name "Branch LAN"
    next
end
```

7. Configure two firewall policies to allow the Branch LAN users to access the HQ DMZ subnet over the IPsec tunnel and vice versa:

```
config firewall policy
    edit 1
        set name "HQ DMZ to Branch LAN"
        set srcintf "port2"
        set dstintf "vpn-server"
        set action accept
        set srcaddr "HQ Server DMZ"
        set dstaddr "Branch LAN"
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set name "Branch LAN to HQ DMZ"
        set srcintf "vpn-server"
        set dstintf "port2"
        set action accept
        set srcaddr "Branch LAN"
        set dstaddr "HQ Server DMZ"
        set schedule "always"
        set service "ALL"
```

```
    next
end
```

To configure Branch FortiGate as IPsec VPN dialup client in the CLI:

1. Configure the WAN interface and default route.

   The WAN interface is the interface connected to the ISP. It can work in static mode, DHCP (as in this example), or PPPoE. The IPsec tunnel is established over the WAN interface.

   ```
   config system interface
       edit "port1"
           set vdom "root"
           set mode dhcp
           set alias "WAN"
           set defaultgw enable
       next
   end
   ```

   Setting `defaultgw` to `enable` automatically injects a default route with an administrative distance (AD) of five, using the gateway IP address provided in DHCP Option 3 by the DHCP server. If the DHCP server does not supply this information, administrators will need to manually configure a default route.
2. Configure address objects for the HQ server DMZ subnet (10.100.99.0/24) and Branch LAN subnet (192.168.111.0/24):

   ```
   config firewall address
       edit "HQ Server DMZ"
           set allow-routing enable
           set subnet 10.100.99.0 255.255.255.0
       next
       edit "Branch LAN"
           set subnet 192.168.111.0 255.255.255.0
       next
   end
   ```

   Setting `allow-routing` to `enable` is optional. It allows address objects to be directly referenced when defining static routes. When this setting is disabled, administrators must manually specify subnets instead of using address objects for static route configuration. Enabling this option streamlines route management and ensures consistency by leveraging predefined address objects.
3. Configure the LAN interface:

   ```
   config system interface
       edit "port4"
           set vdom "root"
           set alias "LAN"
           set ip 192.168.111.0 255.255.255.0
       next
   end
   ```

4. Create a peer user:

```
config user peer
    edit "vpn-server_peer"
        set ca "Fortinet_CA"
    next
end
```

5. Configure the IPsec phase1-interface:

```
config vpn ipsec phase1-interface
    edit "vpn-client"
        set interface "port1"
        set ike-version 2
        set authmethod signature
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set dhgrp 20 21
        set remote-gw 203.0.113.249
        set certificate "Fortinet_Factory"
        set peer "vpn-client_peer"
    next
end
```

6. Configure the IPsec phase2-interface:

```
config vpn ipsec phase2-interface
    edit "vpn-client"
        set phase1name "vpn-client"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
        set src-addr-type name
        set dst-addr-type name
        set src-name "Branch LAN"
        set dst-name "HQ Server DMZ"
    next
end
```

7. Configure the static routes on the Branch to route traffic from the LAN subnet to the DMZ subnet.
   The blackhole route is important to ensure that IPsec traffic does not match the default route when the IPsec tunnel is down.

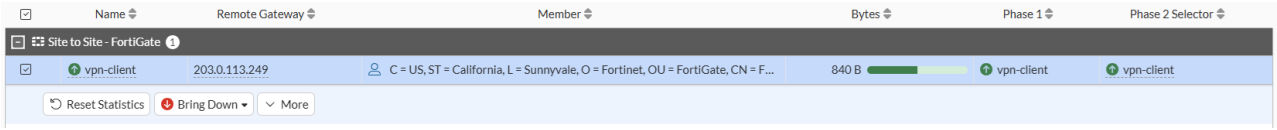
```
config router static
    edit 1
        set device "vpn-client"
        set dstaddr "HQ Server DMZ"
    next
    edit 2
        set distance 254
        set blackhole enable
        set dstaddr "HQ Server DMZ"
    next
end
```

8. Configure two firewall policies to allow the Branch LAN users to access the HQ DMZ subnet over the IPsec tunnel and vice versa:

```
config firewall policy
    edit 1
        set name "Branch LAN to HQ DMZ"
        set srcintf "port4"
        set dstintf "vpn-client"
        set action accept
        set srcaddr "Branch LAN"
        set dstaddr "HQ Server DMZ"
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set name "HQ DMZ to Branch LAN"
        set srcintf "vpn-client"
        set dstintf "port4"
        set action accept
        set srcaddr "HQ Server DMZ"
        set dstaddr "Branch LAN"
        set schedule "always"
        set service "ALL"
    next
end
```

To verify the tunnel status in the GUI:

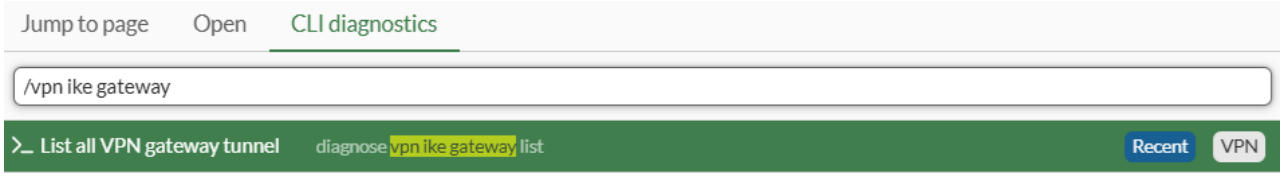
1. On the Branch FortiGate, go to *Dashboard > Network Monitor > VPN*.
2. Find the new tunnel and check the statuses of the *Phase 1* and *Phase 2* selectors.



3. If the Phase 2 selector is not up, click on the tunnel, and select *Bring Up > All Phase 2 Selectors*.

To verify the tunnel status in the CLI:

1. Verify the IPsec Phase 1 tunnel status on either the Branch or HQ using following diagnose command or the command palette (CTRL+P) > CLI diagnostics:




```
Branch # diagnose vpn ike gateway list name vpn-client
vd: root/0
name: vpn-client
version: 2
```

```
interface: port1 3
addr: 198.51.100.41:4500 -> 203.0.113.249:4500
tun_id: 203.0.113.249/::203.0.113.249
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 84005s ago
peer-id: C = US, ST = California, L = Sunnyvale, O = Fortinet, OU = FortiGate, CN =
FGVM02TM2500xxxx, emailAddress = support@fortinet.com
peer-id-auth: yes
nat: peer
pending-queue: 0
IKE SA: created 1/1  established 1/1  time 10/10/10 ms
IPsec SA: created 1/3  established 1/3  time 10/13/20 ms

  id/spi: 0 1b29fe469e07695f/bb46729f1ce4b621
  direction: initiator
  status: established 84005-84005s ago = 10ms
  proposal: aes128-sha256
  key: 03b3a13938d3691d-b0b74a7a6e480f74
  QKD: no
  PQC-KEM (IKE): no
  PQC-KEM (all IPsec): no
  lifetime/rekey: 86400/2094
  DPD sent/recv: 00000000/00000000
  peer-id: C = US, ST = California, L = Sunnyvale, O = Fortinet, OU = FortiGate, CN =
FGVM02TM2500xxxx, emailAddress = support@fortinet.com
```

Note that the IPsec tunnel is established using port 4500 due to NAT Traversal (NAT-T) detection, and the Branch FortiGate acts as the initiator, thus functioning as IPsec VPN dialup client.

2. Verify the IPsec Phase 2 tunnel status on either the Branch or HQ using following diagnose command or the command palette (CTRL+P) > CLI diagnostics:

```
Branch # diagnose vpn tunnel list name vpn-client
list all ipsec tunnel in vd 0
------------------------------------------------------
name=vpn-client ver=2 serial=1 198.51.100.41:4500->203.0.113.249:4500 nexthop=203.0.113.1 tun_
id=203.0.113.249 tun_id6=::203.0.113.249 status=up dst_mtu=1200 weight=1
bound_if=3 real_if=3 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-
rfc  run_state=0 role=primary accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=5 ilast=43039795 olast=43039795 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-demand on=1 status=ok idle=20000ms retry=3 count=0 seqno=0
natt: mode=keepalive draft=32 interval=10 remote_port=4500
fec: egress=0 ingress=0
proxyid=vpn-client proto=0 sa=1 ref=2 serial=1
  src: 0:192.168.111.0-192.168.111.255:0
  dst: 0:10.100.99.0-10.100.99.255:0
  SA:  ref=3 options=30202 type=00 soft=0 mtu=1134 expire=42272/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42903/43200
```

```
  dec: spi=4e2a9b08 esp=aes key=16 4413bc083820d4e3dc1a8a96fd72a784
       ah=sha256 key=32 bc1ffd76e9774f9149f6d7774255e2a4f2efe3f031043019683145465e75b154
  enc: spi=4237b164 esp=aes key=16 14529e0eb279f74acf5cea3ea0d14c53
       ah=sha256 key=32 0ce4b1ff3c7b3b107a4f8a66c5ea76761c2ca7c923855702be9d58b293cb0678
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=203.0.113.249 npu_lgwy=198.51.100.41 npu_selid=0 dec_npuid=0 enc_
npuid=0
```

Note the decryption, encryption, and authentication keys negotiated in Phase 2 to protect subnets under the `src` and `dst` fields.

To verify the traffic flow from PC1 to the Finance server:

1. From a PC1 (192.168.111.101), ping the Finance server (10.100.99.101).
   The ping should be successful.

```
C:\>ping 10.100.99.101

Pinging 10.100.99.101 with 32 bytes of data:
Reply from 10.100.99.101: bytes=32 time=5ms TTL=62
Reply from 10.100.99.101: bytes=32 time=2ms TTL=62
Reply from 10.100.99.101: bytes=32 time=9ms TTL=62
Reply from 10.100.99.101: bytes=32 time=2ms TTL=62

Ping statistics for 10.100.99.101:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 2ms, Maximum = 9ms, Average = 4ms
```

To verify the traffic flow on the Branch and HQ FortiGate in the GUI:

1. On either FortiGate, go to *Log & Report* > *Forward Traffic*.
2. Filter the content using the search bar based on the respective Policy ID.

To make sure that the logs are collected and shown under *Forward Traffic*, set *Log Allowed Traffic* to *All Sessions* in the corresponding firewall policy configured under *Policy & Objects* > *Firewall Policy*.

To verify the traffic flow on the Branch and HQ FortiGate in the CLI:

1. Verify traffic on the Branch:

```
Branch # diagnose sniffer packet any 'host 10.100.99.101 and icmp' 4 0 l
Using Original Sniffing Mode
interfaces=[any]
filters=[host 10.100.99.101 and icmp]
2025-01-16 08:37:22.515436 port4 in 192.168.111.101 -> 10.100.99.101: icmp: echo request
2025-01-16 08:37:22.515478 vpn-client out 192.168.111.101 -> 10.100.99.101: icmp: echo request
2025-01-16 08:37:22.520118 vpn-client in 10.100.99.101 -> 192.168.111.101: icmp: echo reply
2025-01-16 08:37:22.520141 port4 out 10.100.99.101 -> 192.168.111.101: icmp: echo reply
```

2. Verify traffic on the HQ:

```
HQ # diagnose sniffer packet any 'host 10.100.99.101 and icmp' 4 0 l
Using Original Sniffing Mode
interfaces=[any]
filters=[host 10.100.99.101 and icmp]
2025-01-16 08:37:22.518421 vpn-server in 192.168.111.101 -> 10.100.99.101: icmp: echo request
2025-01-16 08:37:22.518459 port2 out 192.168.111.101 -> 10.100.99.101: icmp: echo request
2025-01-16 08:37:22.518900 port2 in 10.100.99.101 -> 192.168.111.101: icmp: echo reply
2025-01-16 08:37:22.518914 vpn-server out 10.100.99.101 -> 192.168.111.101: icmp: echo reply
```

For more information, see Performing a sniffer trace or packet capture on page 4966.

3. Optionally, run real time IKE debug using appropriate filters for troubleshooting if the IPsec tunnel does not come up:

```
Branch # diagnose vpn ike log filter rem-addr4 203.0.113.249
Branch # diagnose debug application ike -1
Branch # diagnose debug enable
```

For more information, see VPN IPsec troubleshooting on page 3335.

FortiClient as dialup client

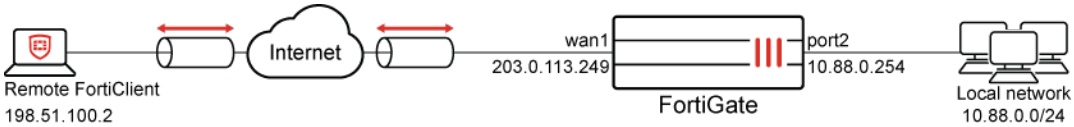
When FortiGate is configured as a dial-up VPN gateway, remote FortiClient users with dynamic IP addresses can establish secure connections over the Internet to access private resources protected by FortiGate.

This topic demonstrates using the *Remote Access* template on the VPN Wizard to configure FortiClient as a dialup client.

> The FortiClient *Secure Internet Access (SIA)* template for the *VPN Wizard* also creates a remote access VPN with the addition of routing all Internet traffic through the FortiGate IPsec VPN tunnel for security inspection. For more information, see Configure FortiClient SIA for IPsec VPN tunnels on page 2993.

Example

In this example, the FortiGate protects a local network (10.88.0.0/24) that a remote FortiClient user needs to securely access over the Internet using a VPN connection. The FortiGate is configured as a dialup VPN server on port3, and the FortiClient is the dialup IPsec VPN client.



This example uses IKE version 2 over UDP transport. For remote user authentication, a local user is configured to be a part of a local user group on the FortiGate.

Starting with FortiClient 7.4.4, IKEv1 is no longer supported on the client. IKEv1 is now considered a legacy configuration.

Also, FortiClient 7.4.4 does not support IPv6. Use FortiClient 7.4.6 or later.

For other authentication methods, see the following topics:

- Add FortiToken multi-factor authentication on page 2985
- Add LDAP user authentication on page 2988
- VPN 2FA with EAP and certificate authentication on page 3019
- SAML-based authentication for FortiClient remote access VPNs on page 3124

Some ISPs may block ESP and UDP ports. Therefore, consider using TCP as Transport.

Go to *VPN > VPN Tunnels > Settings* and enable *Allow VPN negotiations over TCP*. Once enabled, dialup VPNs will use *Auto* for *Transport* mode, allowing clients to attempt a UDP connection first before failing over to TCP.

For more details, see LDAP authentication with TCP as transport on page 3007.

To configure IPsec VPN with FortiClient as the dialup client in the GUI:

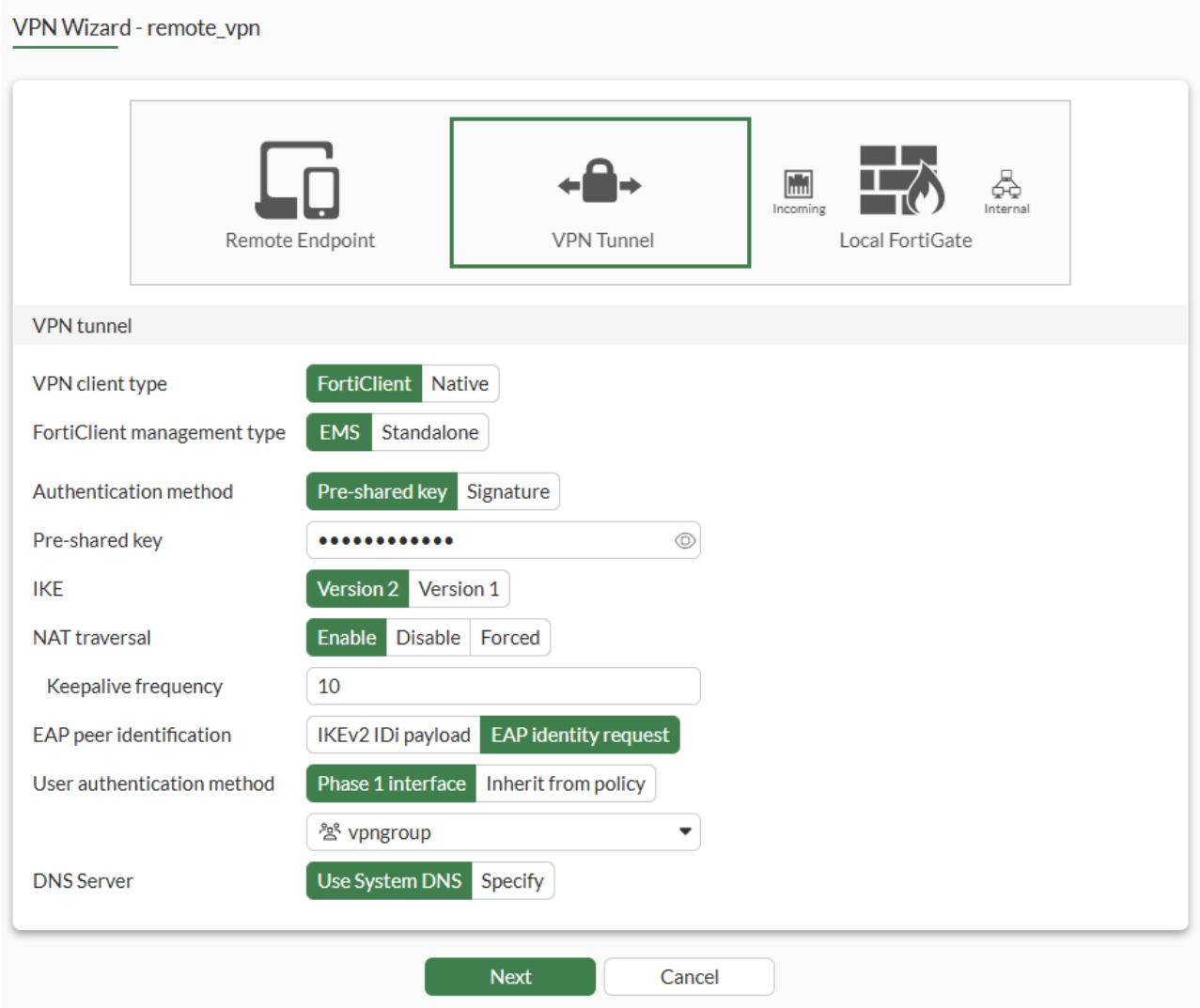
1. Configure a local user:
   a. Go to *User & Authentication > User Definition* and click *Create New*.
   b. Set the *User Type* to *Local User*, then click *Next*.
   c. Set the *Login Credentials*, then click *Next*:
      - *Username*: *vpnuser1*
      - *Password*: Enter a suitable password
   d. Disable *Two-factor Authentication*, then click *Next*.
   e. Set *User Account Status* to *Enabled* and disable *User Group*, then click *Submit*.
2. Configure a local user group:
   a. Go to *User & Authentication > User Groups* and click *Create New*.
   b. Configure the following:

Field	Value
Name	*vpngroup*
Type	*Firewall*
Members	*vpnuser1*

   c. Click *OK*.

   For other types of authentication, such as LDAP, PKI, or SAML, see User & Authentication on page 3484.
3. Go to *VPN > VPN Wizard*.
4. Enter a name for the VPN in the *Tunnel name* field. In this example, *remote_vpn*
5. Set *Select a template* to *Remote Access*.
6. Click *Begin*.
7. Configure the *VPN tunnel* settings:

Field	Value
VPN client type	*FortiClient* For configuring remote access for Native clients, see: • Native VPN as dialup client - Windows New on page 3043 • Native VPN as dialup client - iOS on page 3063
FortiClient management type	*EMS* (default) In most enterprise deployments, EMS is used to manage FortiClients and remote VPN settings. FortiClients managed this way are licensed through EMS. In 8.0 and later, standalone FortiClients can be purchased that are directly licensed through FortiCloud and FortiIdentity Cloud. See VPN with FortiClient Standalone and FortiIdentity Cloud New on page 3026.
Authentication method	*Pre-shared key*
Pre-shared key	Enter a suitable key. The same pre-shared key is used by all remote users on the FortiClient to connect to the dialup VPN. For better security, use a digital certificate instead.
IKE	*Version 2*
NAT traversal	*Enable* Set *Keepalive frequency* to *10*.
EAP peer identification	*EAP identity request*
User authentication method	*Phase 1 interface* Select *vpngroup* from the drop-down list. Optionally, if you have multiple user groups configured, select *Inherit from policy* and then specify the groups. See Using single or multiple user groups for user authentication on page 3024 for details.
DNS Server	*Use System DNS*



8. Click *Next*.
9. Configure the *Remote Endpoint* settings:

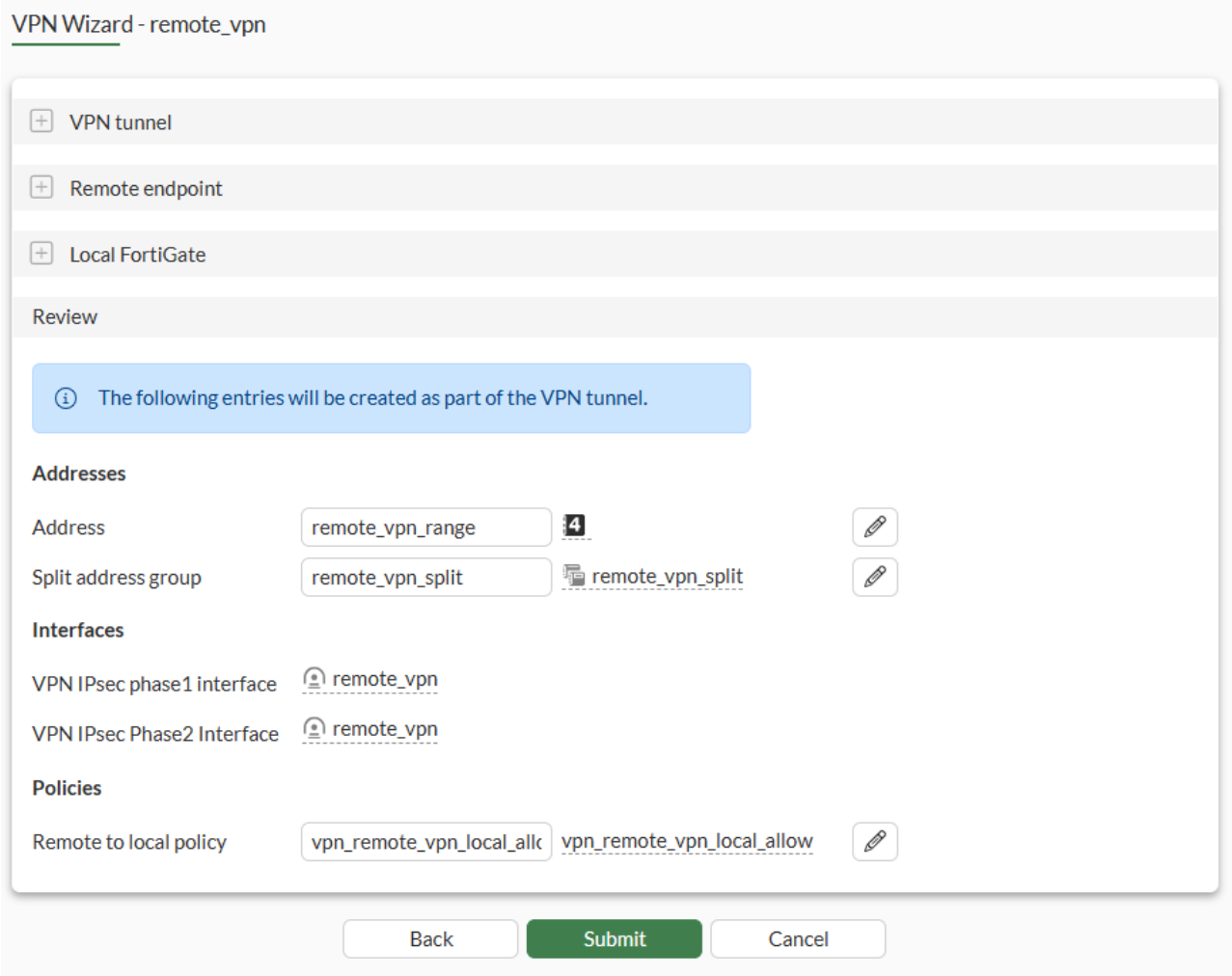
Field	Value
Addressing mode for connected endpoints	*Manual* IPAM is the preferred method. For more information, see Remote access VPNs with IPAM addressing New on page 3003.
Addresses to assign to connected endpoints	*10.10.2.1-10.10.2.200*
Subnet for connected endpoints	*255.255.255.255*
Security posture gateway matching	Disable
EMS SN verification	Disable

Field	Value
Save password	Enable
Auto Connect	Disable
Always up (keep alive)	Disable

10. Click *Next*.
11. Configure the *Local FortiGate* settings:

Field	Value
Incoming interface that binds to tunnel	*WAN (port3)* Disable *Create and add interface to zone*.
Local interface	*DMZ (port2)*
Local address	*DMZ_network* Click + to create a new address object with *Type* set to *Subnet* and *IP/Netmask* set to *10.88.0.0/24*, if not already configured.

12. Click *Next*.
13. Review the configuration and, if everything is correct, click *Submit*.



Because internet traffic is not routed through this firewall, the VPN Wizard automatically configures split tunneling. Only traffic destined for the Split address group remote_vpn_split will go through this tunnel.

To configure IPsec VPN with FortiClient as the dialup client in the CLI:

1. Configure a local user:

```
config user local
    edit "vpnuser1"
        set type password
        set passwd ******
    next
end
```

2. Configure a local user group:

```
config user group
    edit "vpngroup"
        set member "vpnuser1"
```

```
   next
end
```

3. Configure the DMZ (port2) interface that connects to the corporate internal network:

```
config system interface
    edit "port2"
        set vdom "root"
        set ip 10.88.0.254 255.255.255.0
    next
end
```

4. Configure the address group for split tunneling:

```
config firewall address
    edit "DMZ_Network"
        set subnet 10.88.0.0 255.255.255.0
    next
end
```

```
config firewall addrgrp
    edit "remote_vpn_split"
        set member "DMZ_Network"
    next
end
```

5. Configure the WAN interface.

   The WAN interface is the interface connected to the ISP. It can be configured in static mode (as shown here), DHCP, or PPPoE mode. It is recommended to configure it with a static IP address to ensure that the IPsec VPN configuration on the on the FortiClient stays unchanged if the WAN IP changes on the FortiGate. The IPsec tunnel is established over the WAN interface.

```
config system interface
    edit "port3"
        set vdom "root"
        set ip 203.0.113.249 255.255.255.0
    next
end
```

6. Configure the client address pool.

   You must create a firewall address to assign an IP address to a VPN client from the address pool.

```
config firewall address
    edit "remote_vpn_range"
        set type iprange
        set start-ip 10.10.2.1
        set end-ip 10.10.2.200
    next
end
```

7. Configure the IPsec phase1-interface.

   PSK is used as the authentication method in this example. Signature authentication is also an option.

```
config vpn ipsec phase1-interface
    edit "remote_vpn"
        set type dynamic
        set interface "port3"
        set ike-version 2
        set peertype any
        set net-device disable
        set mode-cfg enable
        set proposal es128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set dhgrp 20 21
        set eap enable
        set eap-identity send-request
        set wizard-type dialup-forticlient
        set authusrgrp "vpngroup"
        set assign-ip-from name
        set dns-mode auto
        set ipv4-split-include "remote_vpn_split"
        set ipv4-name "remote_vpn_range"
        set save-password enable
        set psksecret *****
    next
end
```

8. Configure the IPsec phase2-interface:

```
config vpn ipsec phase2-interface
    edit "remote_vpn"
        set phase1name "remote_vpn"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
    next
end
```

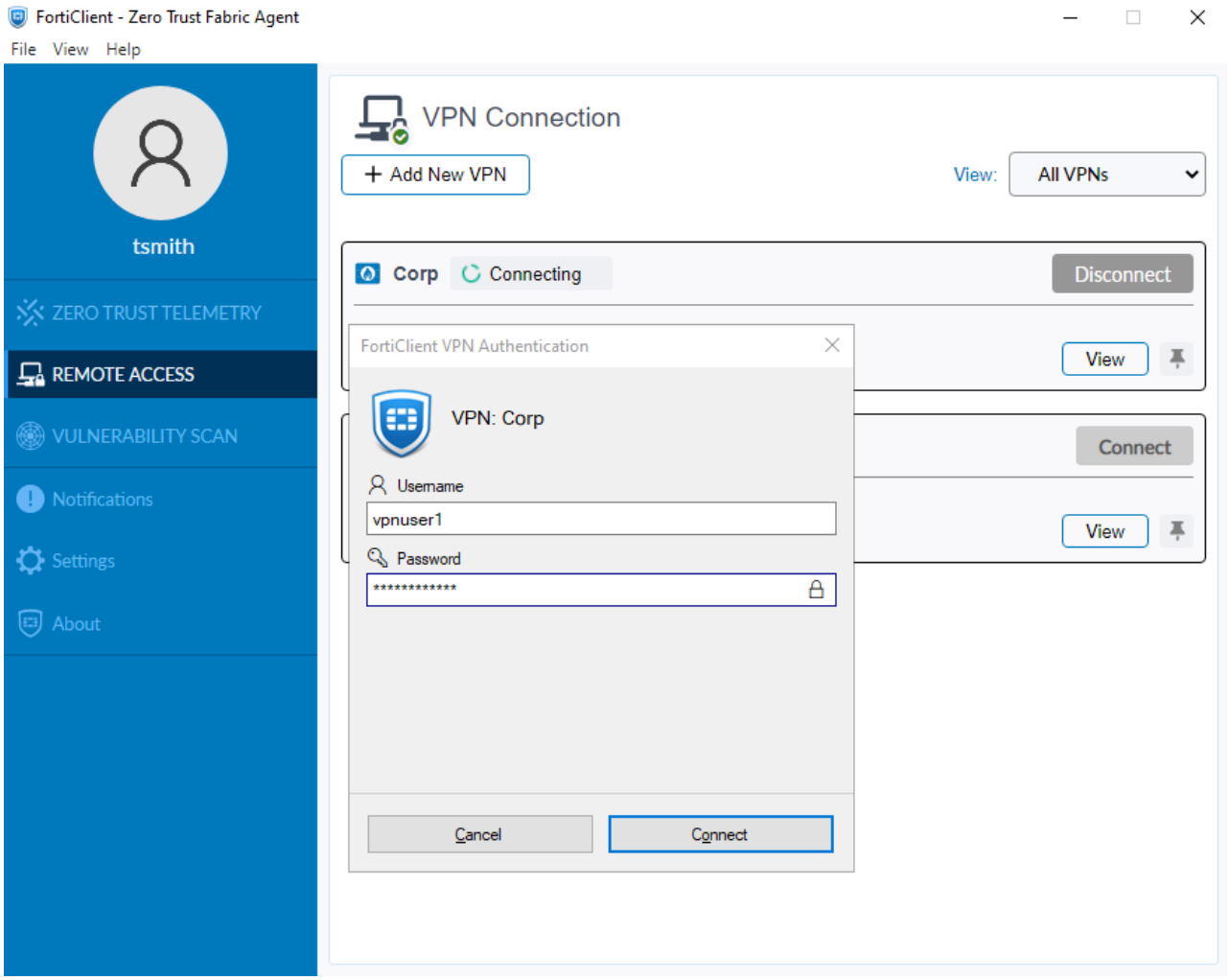
9. Configure a firewall policy to allow client traffic flow over the IPsec VPN tunnel to the FortiGate's lan interface:

```
config firewall policy
   edit 1
        set name "inbound"
        set srcintf "remote_vpn"
        set dstintf "port2"
        set action accept
        set srcaddr "remote_vpn_range"
        set dstaddr "DMZ_Network"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

One or more user groups for user authentication can be configured under a single IPsec VPN tunnel. See Using single or multiple user groups for user authentication on page 3024 for details.

To configure a VPN directly on the FortiClient:

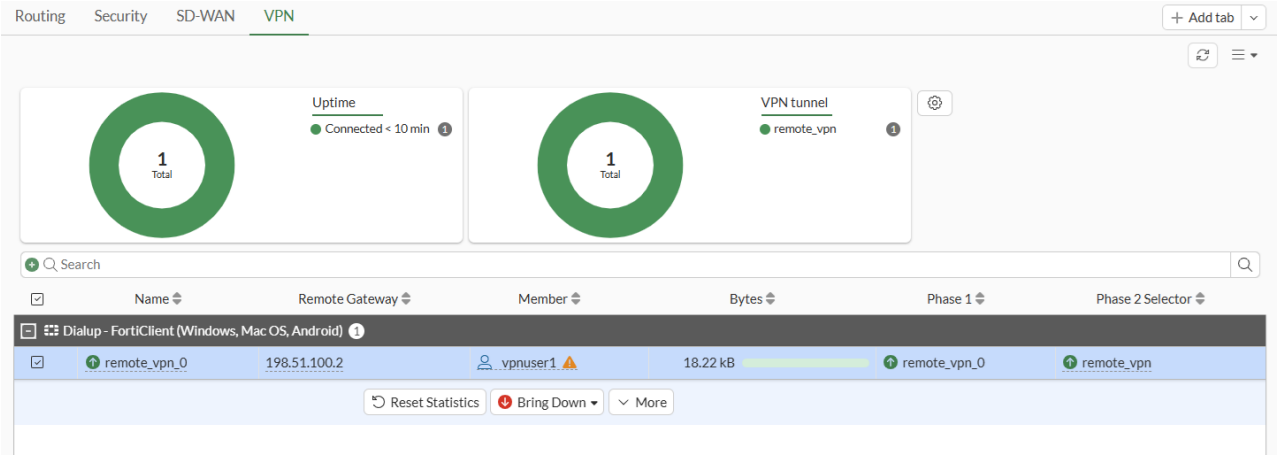
1. In FortiClient, go to *Remote Access* and click *Add a new connection*.
2. Set *VPN* to *IPsec VPN* and specify a *Connection Name*.
3. Set *Remote Gateway* to the FortiGate's WAN IP address (203.0.113.249).
4. Set *Authentication Method* to *Pre-Shared Key* and enter the key.
5. Configure the remaining settings as needed, then click *Save*.
6. Go to the *Remote Access* tab, select the VPN, and then click *Connect*.



To establish an IPsec tunnel, the Phase 1 and Phase 2 configurations of the dial-up IPsec VPN on FortiGate must precisely match the corresponding settings on FortiClient.

To verify the tunnel status in the GUI:

1. On the FortiGate, go to *Dashboard* > *Network Monitor* > *VPN*.
2. Find the new tunnel, *remote_vpn_0* in this example.

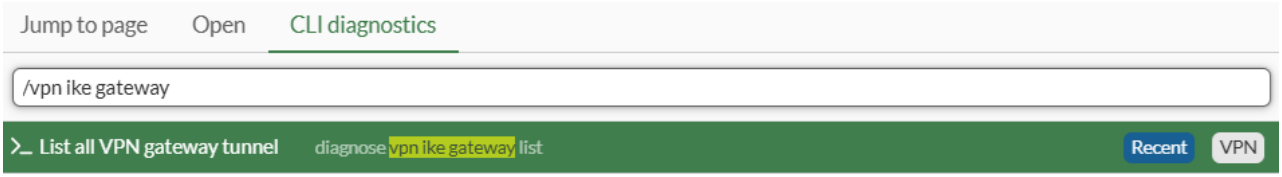


> Successfully established dialup IPsec VPN tunnels are displayed using a parent name followed by an index, creating a structured and organized association with the parent tunnel. For example, the parent name *remote_vpn* results in tunnel names such as *remote_vpn_0*, *remote_vpn_1*, and so on.

3. Hover the cursor over the tunnel name to see additional details.

To verify the tunnel status:

1. Verify the IPsec Phase 1 tunnel status on the FortiGate, either using the CLI or the Command palette (CTRL+P) > CLI diagnostics:




```
# diagnose vpn ike gateway list name remote_vpn_0

vd: root/0
name: remote_vpn_0
version: 2
interface: port3 5
addr: 203.0.113.249:4500 -> 198.51.100.2:61978
tun_id: 10.10.2.1/::10.0.0.3
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 496s ago
eap-user: vpnuser1
2FA: no
```

```
peer-id: 198.51.100.2
peer-id-auth: no
FortiClient UID: 9A016B5A6E914B42AD4168C066EB04CA
assigned IPv4 address: 10.10.2.1/255.255.255.255
nat: me peer
pending-queue: 0
PPK: no
IKE SA: created 1/1  established 1/1  time 7880/7880/7880 ms
IPsec SA: created 1/1  established 1/1  time 20/20/20 ms

  id/spi: 21718 0fd2b81ac558640c/66ea3061dc95301d
  direction: responder
  status: established 496-488s ago = 7880ms
  proposal: aes256-sha256
  child: no
  SK_ei: e88f1ac99983dd5f-08e807aeb605a06f-ddcbde523c652452-b7bf877a8ef9020e
  SK_er: f57903b04ab2e9a2-da36555aada93571-0d0343dd4bee6e89-f69d2394c415ae6e
  SK_ai: b2a1d56be7354fa1-2d761f2f3027c3ca-ae20e2f2a322d1f2-57d878b39febe294
  SK_ar: c2448f95a13ace5d-b6aa2f93f78f6f6c-6c3af0a16d221264-bebb43fa1897938f
  PPK: no
  message-id sent/recv: 0/6
  QKD: no
  PQC-KEM (IKE): no
  PQC-KEM (all IPsec): no
  lifetime/rekey: 86400/85641
  DPD sent/recv: 00000000/00000000
  peer-id: 198.51.100.2
```

Note that FortiGate acts as the responder, establishes the IKEv2 VPN tunnel with remote FortiClient user with IP 198.51.100.2 over UDP NAT-T port 4500 on its port3 interface, and dynamically assigns a VPN IP of 10.10.2.1 to FortiClient endpoint. The authenticated username is also displayed in the `eap-user` field.

2. Verify the IPsec Phase 2 tunnel status on the FortiGate:

```
# diagnose vpn tunnel list name remote_vpn_0

list all ipsec tunnel in vd 0
------------------------------------------------------
name=remote_vpn_0 ver=2 serial=8 203.0.113.249:4500->198.51.100.2:61978 nexthop=203.0.113.3
tun_id=10.10.2.1 tun_id6=::10.0.0.3 status=up dst_mtu=1500 weight=1 country=ZZ
bound_if=5 real_if=5 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none options[0x23a8]=npu
rgwy-chg rport-chg frag-rfc  run_state=0 role=primary accept_traffic=1 overlay_id=0

parent=remote_vpn index=0
proxyid_num=1 child_num=0 refcnt=6 ilast=1 olast=44059296 ad=/0
stat: rxp=369 txp=0 rxb=29687 txb=0
dpd: mode=on-demand on=1 status=ok idle=20000ms retry=3 count=0 seqno=0
natt: mode=keepalive draft=0 interval=10 remote_port=61978
fec: egress=0 ingress=0
proxyid=remote_vpn proto=0 sa=1 ref=2 serial=1 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.10.2.1-10.10.2.1:0
  SA:  ref=3 options=20682 type=00 soft=0 mtu=1422 expire=42480/0B replaywin=2048
```

```
      seqno=1 esn=0 replaywin_lastseq=00000171 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43189/43200
  dec: spi=4237d1be esp=aes key=32
28080aad5adb7d3c766c5be609c355e2a8be09d4f80793be7cd2aa0291d5a5a5
       ah=sha256 key=32 77e0869053c451223f138ba2f8b9f53b6a319cd13e1486de90f5b3b86be62715
  enc: spi=476588dc esp=aes key=32
87a2d4ff73c053e5f8c528a14ec12914fb5f7f5f3737f8b6db70f28f4aef7047
       ah=sha256 key=32 73cfdadc0fdf96b844d67d9a49f30944043134be7e14dd054fca7295cf32c194
  dec:pkts/bytes=369/29687, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=0.0.0.0:0 npu_lgwy=0.0.0.0:0 npu_selid=0
  dec_npuid=0 enc_npuid=0 dec_engid=-1 enc_engid=-1 dec_saidx=-1 enc_saidx=-1
```

Note the decryption, encryption, and authentication keys negotiated in Phase 2 to protect subnets under the src and dst fields. FortiOS also automatically adds the dynamically assigned VPN IP of 10.10.2.1 to its traffic selectors for dialup tunnel remote_vpn_0.

To verify traffic through the IPsec tunnel:

1. 1. On the remote FortiClient user, initiate ICMP traffic to a machine in the 10.88.0.0/24 network and verify that traffic goes through:

```
# ping 10.88.0.7
Pinging 10.88.0.7 with 32 bytes of data:
Reply from 10.88.0.7: bytes=32 time=2ms TTL=63
Reply from 10.88.0.7: bytes=32 time=2ms TTL=63
```

2. On FortiGate, run a sniffer trace for ICMP packets destined to 10.88.0.7:

```
# diagnose sniffer packet any 'host 10.88.0.7' 4 0 l
Using Original Sniffing Mode
interfaces=[any]
filters=[host 10.88.0.7]
2026-04-13 12:08:17.297621 remote_vpn in 10.10.2.1 -> 10.88.0.7: icmp: echo request
2026-04-13 12:08:17.297733 port2 out 10.88.0.254 -> 10.88.0.7: icmp: echo request
2026-04-13 12:08:17.298459 port2 in 10.88.0.7 -> 10.88.0.254: icmp: echo reply
2026-04-13 12:08:17.298487 remote_vpn out 10.88.0.7 -> 10.10.2.1: icmp: echo reply
```

The ICMP request packets are received on remote_vpn_0 tunnel interface and routed to the port2 interface. ICMP reply packets are received on the port2 interface and routed back out on remote_vpn_0 tunnel interface.

For more information, see Performing a sniffer trace or packet capture on page 4966.

3. Optionally, run real time IKE debug using appropriate filters for troubleshooting if the IPsec tunnel does not come up:

```
# diagnose vpn ike log filter rem-addr4 198.51.100.2
# diagnose debug application ike -1
# diagnose debug enable
```

For more information, see VPN IPsec troubleshooting on page 3335.

Add FortiToken multi-factor authentication

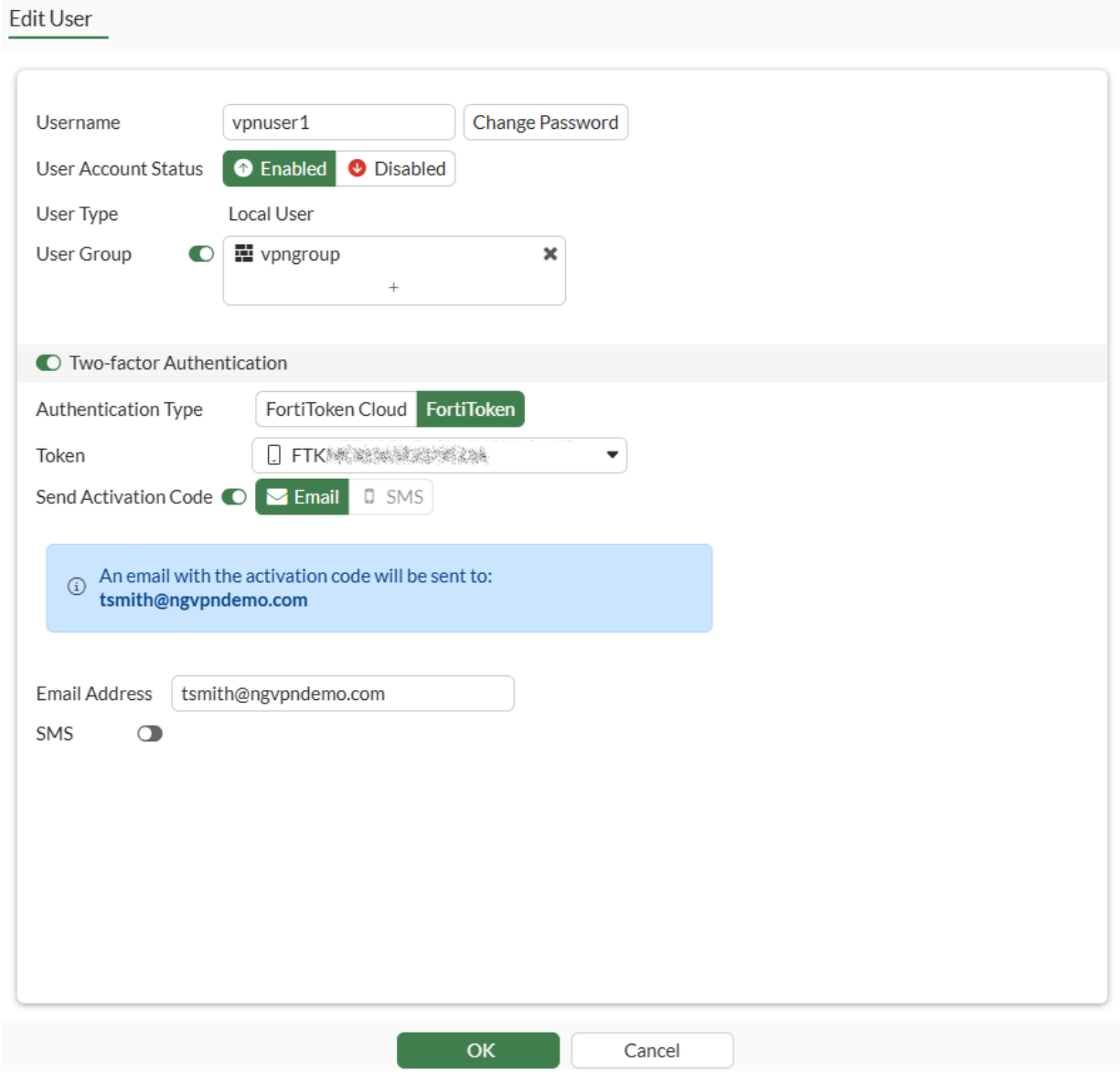
This configuration adds multi-factor authentication (MFA) to the FortiClient dialup VPN configuration (FortiClient as dialup client on page 2973). It uses one of the two free mobile FortiTokens that is already installed on the FortiGate.

For large number of users, it is recommended to use FortiAuthenticator for user and identity management. Also consider FortiIdentity Cloud for mass deployments of mobile tokens.

FortiGates with FortiCare Premium or FortiCare Elite has 3 free seats to FortiIdentity Cloud. See FortiCloud Subscriptions for details.

To configure MFA using the GUI:

1. Edit the user:
   a. Go to *User & Authentication* > *User Definition* and edit local user *vpnuser1*.
   b. Enable *Two-factor Authentication*.
   c. For *Authentication Type*, click *FortiToken* and select one mobile *Token* from the list.
   d. Enter the user's *Email Address*.
   e. Enable *Send Activation Code* and select *Email*.



f. Click *OK*.

2. Activate the mobile token.

   When a FortiToken is added to user *vpnuser1*, an email is sent to the user's email address. Follow the instructions to install your FortiToken mobile application on your device and activate your token.

To configure MFA using the CLI:

1. Edit the user and user group:

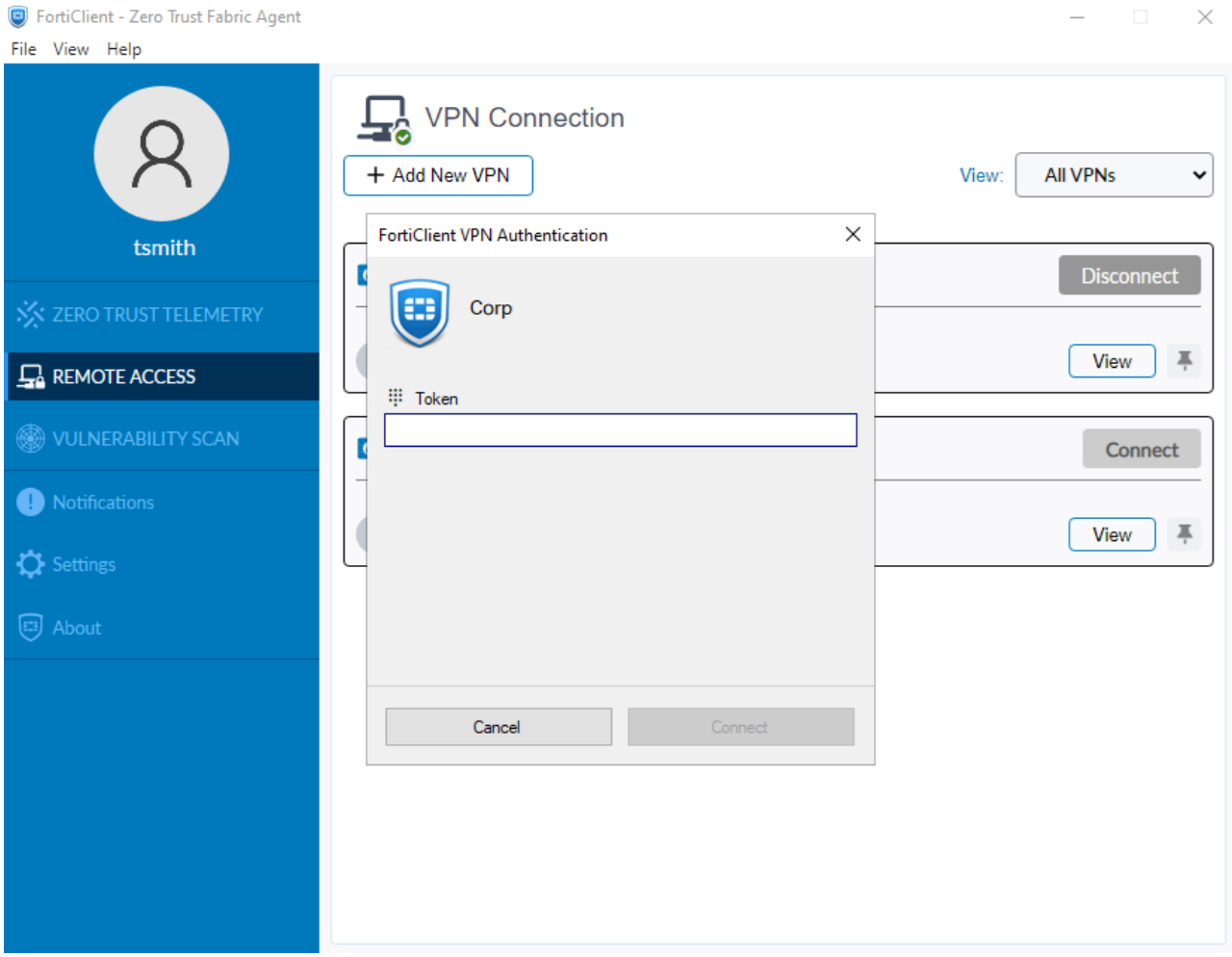
```
config user local
    edit "vpnuser1"
        set type password
        set two-factor fortitoken
        set fortitoken <select mobile token for the option list>
        set email-to <user's email address>
```

```
        set passwd <user's password>
    next
end
```

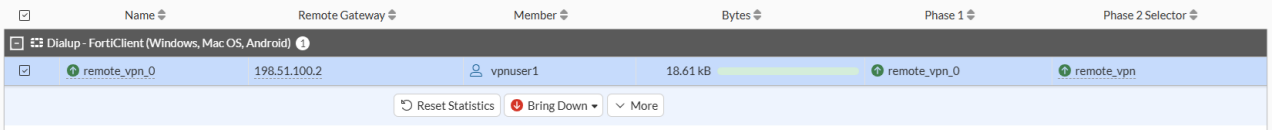
2. Activate the mobile token.
   When a FortiToken is added to user *vpnuser1*, an email is sent to the user's email address. Follow the instructions to install your FortiToken mobile application on your device and activate your token.

To verify the connection:

1. On the FortiClient, go to *Remote Access* and find the VPN tunnel to connect.
2. Click *Connect*.
3. When prompted, enter the username and password.
4. On the next screen, when prompted for your FortiToken code, locate the code on your FortiToken Mobile app and enter the code.



5. Click *Connect*.
6. Once connected, go to the *FortiGate > Dashboard > Network Monitor > VPN*.
   The user will be shown in the *Member* column. There will not be any warning icon indicating the need for 2FA.



7. In the debug command `diagnose vpn ike gateway list`, confirm that the `2FA` field displays `Yes`:

```
# diagnose vpn ike gateway list

vd: root/0
name: remote_vpn_0
version: 2
interface: port3 5
addr: 203.0.113.249:4500 -> 198.51.100.2:49720
tun_id: 10.10.2.1/::10.0.0.5
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 443s ago
eap-user: vpnuser1
2FA: yes
peer-id: 198.51.100.2
peer-id-auth: no
FortiClient UID: 9A016B5A6E914B42AD4168C066EB04CA
assigned IPv4 address: 10.10.2.1/255.255.255.255
```

Add LDAP user authentication

This configuration adds LDAP user authentication to the FortiClient dialup VPN configuration (FortiClient as dialup client on page 2973). You must have already generated and exported a CA certificate from your AD server.

Since IPsec uses IKEv2 by default, the FortiClient must utilize EAP-TTLS for authenticating against the FortiGate and the LDAPS authentication server. This requires FortiClient running 7.4.3 or later, and configurations synchronized from FortiClient EMS.

This example demonstrates the EMS configurations needed to configure the Remote Access profile.

For a complete end-to-end configuration with options to configure VPN over TCP, see LDAP authentication with TCP as transport on page 3007.

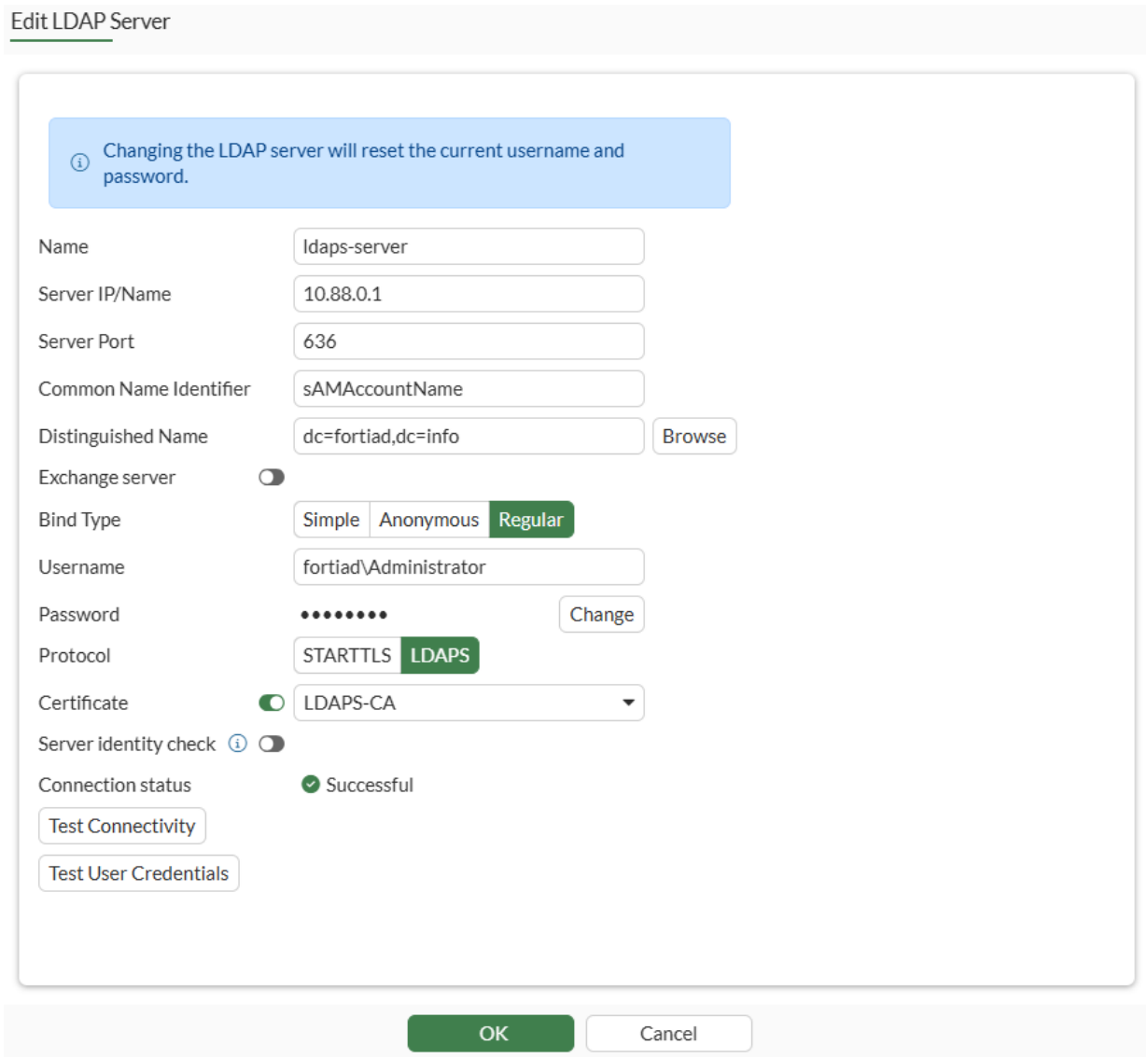
To configure LDAP user authentication using the GUI:

1. Import the CA certificate into FortiGate:
   a. Go to *System* > *Certificates*.
      If the *Certificates* option is not visible, enable it in *Feature Visibility*. See Feature visibility on page 3952 for details.
   b. Click *Import* > *CA Certificate*.
   c. Set *Type* to *File*.
   d. Click *Upload* then find and select the certificate file.

e. Click *OK*.
   The CA certificate now appears in the list of *External CA Certificates*. In this example, it is called *CA_Cert_1*.
f. Optionally, rename the system generated *CA_Cert_1* to something more descriptive:

```
config vpn certificate ca
    rename CA_Cert_1 to LDAPS-CA
end
```

2. Configure the LDAP user:
   a. Go to *User & Authentication > LDAP Servers* and click *Create New*.
   b. Set *Name* to *ldaps-server* and specify *Server IP/Name*.
   c. Specify *Common Name Identifier* and *Distinguished Name*.
   d. Set *Bind Type* to *Regular*.
   e. Specify *Username* and *Password*.
   f. Set *Protocol* to *LDAPS*.
   g. Enable *Certificate* and select *LDAPS-CA* from the list.
   h. Optionally, enable *Server identity check* if you need to verify the server domain name/IP address against the server certificate.
   i. Click *Test Connectivity*. The *Connection status* should change to *Successful*.



j. Click *OK*.

3. Add the LDAP user to the user group:
   a. Go to *User & Authentication* > *User Groups* and edit the *vpngroup* group.
   b. In *Remote Groups*, click *Add* to add the *ldaps-server* remote server.
   c. Click *OK*.

To configure LDAP user authentication using the CLI:

1. Import the CA certificate using the GUI.
2. Configure the LDAP user:

```
config user ldap
    edit "ldaps-server"
        set server "10.88.0.1"
        set server-identity-check disable
```

```
        set cnid "sAMAccountName"
        set dn "dc=fortiad,dc=info"
        set type regular
        set username "fortiad\\Administrator"
        set password ENC **********
        set secure ldaps
        set ca-cert "LDAPS-CA"
        set port 636
    next
end
```

3. Add the LDAP user to the user group:

```
config user group
    edit "vpngroup"
        append member "ldaps-server"
    next
end
```

To configure the Remote Access profile from EMS:

1. On FortiClient EMS, navigate to *Endpoint Profiles > Remote Access* and click *Edit* on the *Default* endpoint profile.
2. Under *VPN Tunnels*, click *Add Tunnel*.
3. Set *VPN type* as *Manual*.
4. Click *Next*.
5. Under *Basic Settings*, enter the following:

Field	Value
Name	*Corp_remote*
Remote Gateway	*203.0.113.249*
Authentication Method	*Pre-shared Key*
Pre-Shared Key	Enter pre-shared key that was configured on FortiGate's dialup VPN configuration.
Prompt for Username	*Enabled*

6. Click *VPN Settings*, enter the following:

Field	Value
IKE	*Version 2*
Address Assignment	*Mode Config*
Encapsulation	*IKE UDP Port*
IKE TCP Port	*500*

7. Click *Phase 1*, enter the following details:

Field		Value
IKE Proposal		
	Encryption, Authentication	*AES128, SHA256*
	Encryption, Authentication	*AES256, SHA256*
DH Groups		*20, 21*
Key Life		*86400*
EAP Authentication Method		*EAP-TTLS*
Dead Peer Detection		*Enable*
NAT Traversal		*Enable*

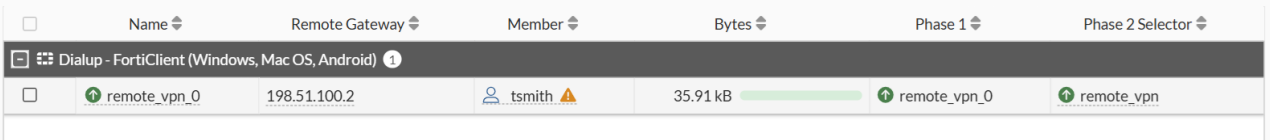
8. Click *Phase 2*, enter the following details:

Field		Value
IKE Proposal		
	Encryption, Authentication	*AES128, SHA256*
	Encryption, Authentication	*AES256, SHA256*
DH Groups		*20*
Key Life		*43200*
Enable Replay Detection		*Enable*
Enable Perfect Forward Secrecy (PFS)		*Enable*

9. Click *Save* to save the VPN Tunnel configuration.
10. Click *Save* to save *Remote Access* endpoint profile.

To verify the connection:

1. On the FortiClient, go to *Remote Access* and find the VPN tunnel Corp_remote to connect.
2. Click *Connect*.
3. When prompted, enter the LDAP username and password.
4. Once connected, go to the *FortiGate > Dashboard > Network Monitor > VPN*.
   The LDAP user will be shown in the *Member* column:



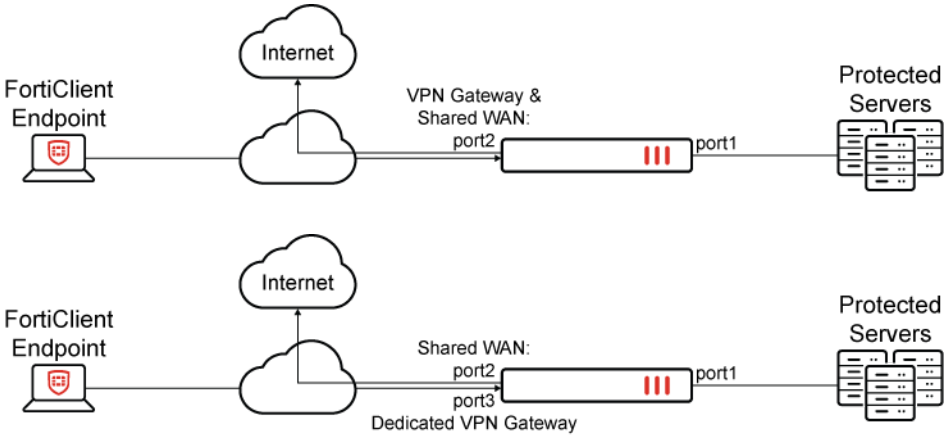
5. In the debug command `diagnose vpn ike gateway list`, the `eap-user` field displays the LDAP user name:

```
# diagnose vpn ike gateway list

vd: root/0
name: remote_vpn_0
version: 2
interface: port3 5
addr: 203.0.113.249:4500 -> 198.51.100.2:58861
tun_id: 10.10.2.1/::10.0.0.6
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 1048s ago
eap-user: tsmith
2FA: no
peer-id: 198.51.100.2
peer-id-auth: no
FortiClient UID: 9A016B5A6E914B42AD4168C066EB04CA
assigned IPv4 address: 10.10.2.1/255.255.255.255
```

Configure FortiClient SIA for IPsec VPN tunnels

The FortiClient *Secure Internet Access (SIA)* template for the *VPN Wizard* enables the configuration of a remote access IPsec VPN to ensure all FortiClient traffic is routed through the FortiGate IPsec VPN tunnel for security inspection. The template allows administrators to select the desired security profile, including certificate or deep inspection, and configure policies to block access to botnet and C&C servers. Additionally, it provides an option to allow remote VPN users access to specified local subnets and local interfaces.

Typical deployment scenarios can be broken down into these topologies:



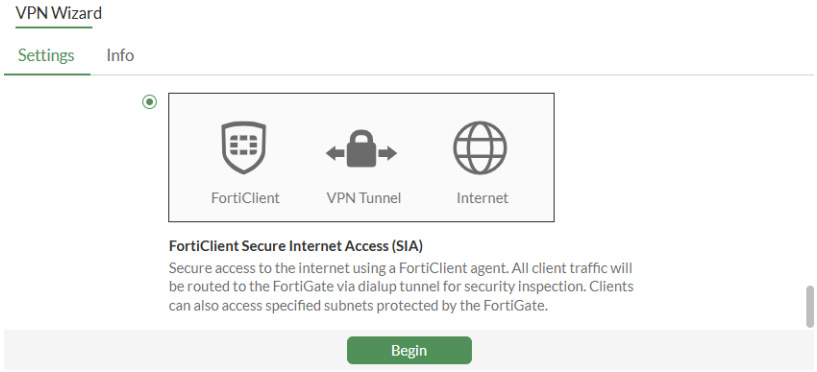
In the first use case, port2 acts as the VPN Gateway interface as well as the shared WAN interface. Therefore, when VPN is established, a policy needs to be created to allow traffic from the VPN interface to port2.

In the second use case, port3 acts as a dedicated VPN Gateway interface. Traffic entering the VPN interface will egress on port2 which is the acting shared WAN.

The following example uses the second use case.

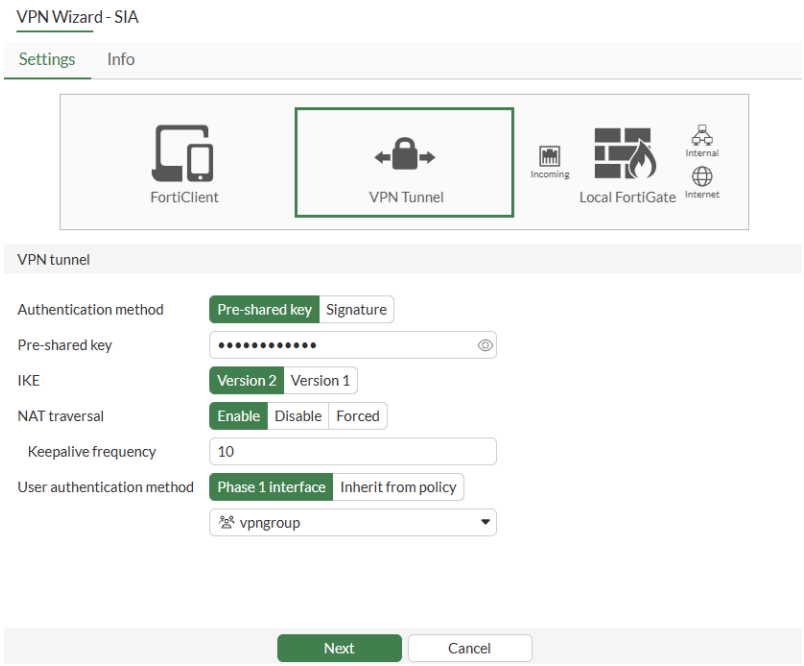
To configure IPsec VPN with FortiClient as the dialup client in the GUI:

1. Configure a user and user group:
   a. Go to *User & Authentication* > *User Definition* and create a user. See Users on page 3485 for different types of users.
   b. Go to *User & Authentication* > *User Groups* and create a user group with the new user added as a member. See User groups on page 3487.
2. Go to *VPN* > *VPN Wizard* and configure the tunnel settings:
   a. Enter a VPN *Tunnel name*.
   b. For *Select a template*, select *FortiClient Secure Internet Access (SIA)*.



   c. Click *Begin*.
3. Configure the following *VPN tunnel* settings:
   a. Set *Authentication method* to *Pre-shared key*.
   b. Enter the *Pre-shared key*.
   c. Set *IKE* to *Version 2*.
   d. For *User authentication method*, select *Phase 1 interface*.

> One or more user groups for user authentication can be configured under a single IPsec VPN tunnel. See Using single or multiple user groups for user authentication on page 3024 for details.



e. Click *Next*.

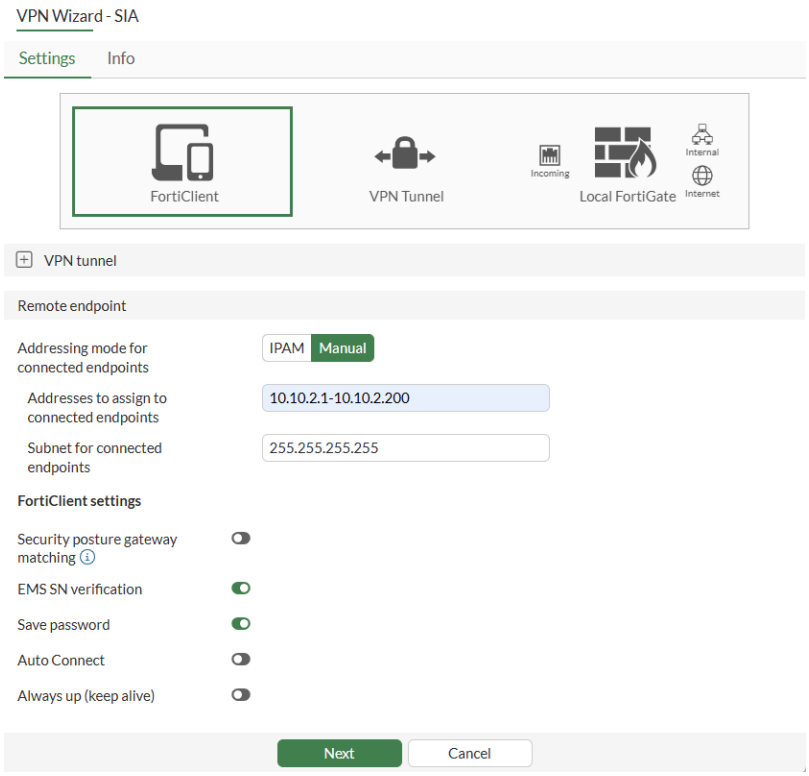
4. Configure the following *Remote endpoint* settings:

a. In Address mode, the default is *IPAM*. To use this option, IPAM must be enabled and a subnet must be assigned, then you can configure the *Network size* and *Subnet for connected endpoints* according to the number of remote endpoints.

Alternatively, change the Address mode to *Manual*. Configure *Addresses to assign to connected endpoints* by entering the client address range manually. Enter the *Subnet* for connected endpoints.

b. Enable *FortiClient settings* as needed:

Field	Description
Security posture gateway matching	A tunnel connection will only be established when the endpoint is matched by a security posture tag. See Enforcing security posture verification before dial-up VPN connection on page 3174.
EMS SN verification	Enable/disable verification of EMS serial number. Requires both the FortiGate and FortiClient endpoints must be connected to the same FortiClient EMS. See Enhancing IPsec security using EMS SN verification on page 3146.
Save password	Enable/disable saving username and password on VPN clients.
Auto Connect	Enable/disable allowing the VPN client to bring up the tunnel when there is no traffic.
Always up (keep alive)	Enable/disable allowing the VPN client to keep the tunnel up when there is no traffic.



c. Click *Next*.

5. Configure the following *Local FortiGate* and *Secure Internet Access (SIA)* settings:

a. From the *Incoming interface that binds to tunnel* dropdown list, select the port. This port may be your WAN interface, or a dedicated interface designated for establishing the IPsec tunnel on.

b. Enable *Local subnets that remote endpoints can access*.

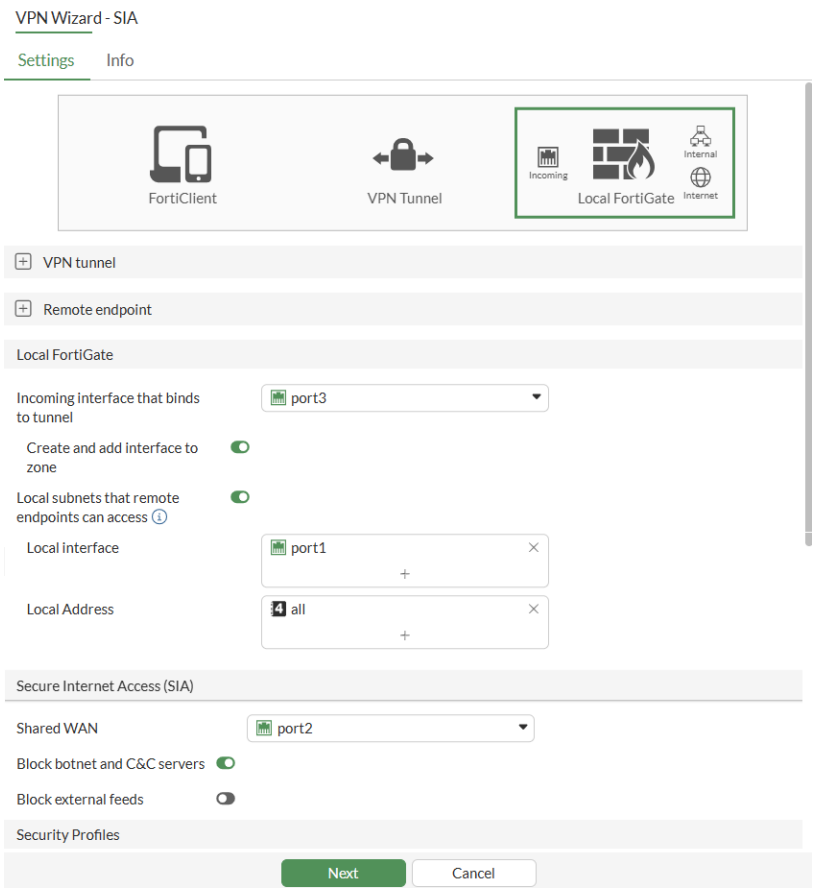
c. Set the *Local interface* and *Local Address*.

d. From the *Shared WAN* dropdown list, select WAN interface. This interface can also be the same interface used for establishing the IPsec tunnel if your internet access is through it.

e. Enable the other *Secure Internet Access (SIA)* fields, as needed.

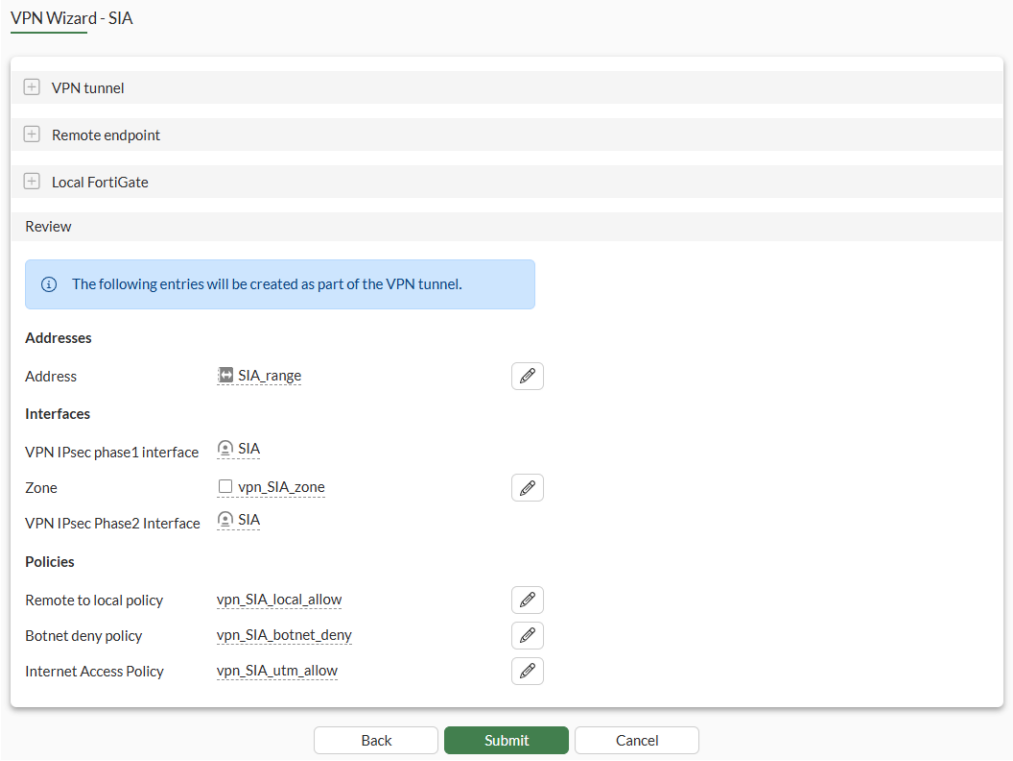
> The *Block external feeds* field is an optional feature that allows you to block specific external feeds. After enabling the field, select an *Address External Feed* or *Dynamic Address* option to proceed.

f. Enable the required *Security Profiles*. These profiles protect the internet traffic that is passing through the SIA connection.



g. Click *Next*.

6. Click *Submit*.



The VPN wizard creates three policies:

- *Remote to local policy* allows traffic to the local subnet without any inspection.
- *Botnet deny policy* blocks any internet traffic to Botnet and C&C server destinations.
- *Internet Access Policy* applies the Security Profiles to the Secure Internet Access policy.

To configure IPsec tunnels for FortiClient Secure Internet Access (SIA) in the CLI:

1. Configure the user and user group:

```
config user local
    edit "vpnuser1"
        set type password
        set passwd ********
    next
end
config user group
    edit "vpngroup"
        set member "vpnuser1"
    next
end
```

2. Configure the LAN interface, such as port1:

```
config system interface
    edit "port1"
        set vdom "root"
        set ip 192.168.5.4 255.255.255.0
```

```
    next
end
config firewall address
    edit "local_subnet_1"
        set subnet 192.168.5.0 255.255.255.0
    next
end
config firewall addrgrp
    edit "local_network"
        set member "local_subnet_1"
    next
end
```

3. Configure the WAN interface for establishing the IPsec tunnel. Additionally, configure a separate WAN interface for internet access if applicable. If both functions use the same interface, configure only that single interface.
   a. Configure the interface on which the IPsec tunnel is established:

```
config system interface
    edit "port3"
        set vdom "root"
        set ip 172.16.200.4 255.255.255.0
    next
end
```

   b. Configure the interface on which Internet access is enabled:

```
config system interface
    edit "port2"
        set vdom "root"
        set ip 172.17.200.4 255.255.255.0
    next
end
```

4. Configure the client address pool:

```
config firewall address
    edit "client_range"
        set type iprange
        set comment "VPN client range"
        set start-ip 10.10.2.1
        set end-ip 10.10.2.200
    next
end
```

5. Configure the IPsec Phase 1 interface:

```
config vpn ipsec phase1-interface
    edit "SIA"
        set type dynamic
        set interface "port3"
        set ike-version 2
```

```
        set peertype any
        set mode-cfg enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set dhgroup 20 21
        set eap enable
        set authusrgrp "vpngroup"
        set assign-ip-from name
        set ipv4-start-ip 10.10.2.1
        set ipv4-end-ip 10.10.2.200
        set dns-mode auto
        set eap-identity send-request
        set save-password enable
        set psksecret ********
        set transport auto
    next
end
```

One or more user groups for user authentication can be configured under a single IPsec VPN tunnel. See Using single or multiple user groups for user authentication on page 3024 for details.

6. Configure the IPsec Phase 2 interface:

```
config vpn ipsec phase2-interface
    edit "SIA"
        set phase1name "SIA"
        set dhgroup 20 21
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
    next
end
```

7. Configure the firewall policy to allow client traffic to flow over the IPsec VPN tunnel:

```
config firewall policy
    edit 1
        set name "vpn_SIA_local_allow"
        set srcintf "SIA"
        set dstintf "port1"
        set action accept
        set srcaddr "SIA_range"
        set dstaddr "local_network"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
    edit 2
        set name "vpn_SIA_botnet_deny"
        set srcintf "SIA"
        set dstintf "port2"
        set srcaddr "all"
```

```
        set internet-service enable
        set internet-service-name "Botnet-C&C.Server"
        set schedule "always"
        set logtraffic all
    next
    edit 3
        set name "vpn_SIA_utm_allow"
        set srcintf "SIA"
        set dstintf "port2"
        set action accept
        set srcaddr "SIA_range"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set av-profile "default"
        set webfilter-profile "default"
        set dnsfilter-profile "default"
        set file-filter-profile "default"
        set ips-sensor "default"
        set application-list "default"
        set logtraffic all
        set nat enable
    next
end
```

8. Configure the FortiClient to connect to IPsec tunnel using the IP address configured on port 3. Review the IPsec tunnel details using CLI once it establishes:

```
# diagnose vpn ike gateway list

vd: root/0
name: SIA_0
version: 2
interface: port3 7
addr: 172.16.200.4:500 -> 172.16.200.242:500
tun_id: 10.10.2.1/::10.0.0.3
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 119s ago
eap-user: userc
2FA: no
peer-id: 172.16.200.242
peer-id-auth: no
FortiClient UID: 6108A9179A5C40D7BD57504E15114C1F
assigned IPv4 address: 10.10.2.1/255.255.255.255
pending-queue: 0
PPK: no
IKE SA: created 1/1  established 1/1  time 20/20/20 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms
```

```
  id/spi: 1 f5244df351756ec7/8183f1f8f03b9f05
  direction: responder
  status: established 119-119s ago = 20ms
  proposal: aes256-sha256
  child: no
  SK_ei: 1d527193fb58b2dd-7019c0a5270cd57c-37e49e110ca64975-072b54563e4c1556
  SK_er: 4b393c9433147a16-eb5f2232fc46af80-bdfa28048275f780-7e7f0bf6da06db2c
  SK_ai: 98ed685e72dfd297-a049598cde672f86-e1e65682148cbe5b-ad6ceca024fcced4
  SK_ar: 655ee500ab17ed12-6fdb69e82ac827ee-21943cd3f206b69a-9e81df2b21ee0aab
  PPK: no
  message-id sent/recv: 0/29
  QKD: no
  PQC-KEM (IKE): no
  PQC-KEM (all IPsec): no
  lifetime/rekey: 86400/86010
  DPD sent/recv: 00000000/00000000
  peer-id: 172.16.200.242
```

9. Review the tunnel information:

```
# diagnose vpn tunnel list

list all ipsec tunnel in vd 0
------------------------------------------------------
name=SIA ver=2 serial=3 172.16.200.4:0->0.0.0.0:0 nexthop= tun_id=10.0.0.2 tun_id6=::10.0.0.2
status=up dst_mtu=0 weight=1
bound_if=7 real_if=0 lgwy=static/1 tun=intf mode=dialup/2 encap=none/552 options[0228]=npu
frag-rfc  role=primary accept_traffic=1 overlay_id=0

proxyid_num=0 child_num=1 refcnt=3 ilast=42968340 olast=42968340 ad=/0
stat: rxp=681 txp=0 rxb=92088 txb=0
dpd: mode=on-demand on=0 status=ok idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
------------------------------------------------------
name=SIA_0 ver=2 serial=4 172.16.200.4:0->172.16.200.242:0 nexthop= tun_id=10.10.2.1 tun_
id6=::10.0.0.3 status=up dst_mtu=0 weight=1
bound_if=7 real_if=0 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/74408 options
[122a8]=npu rgwy-chg frag-rfc  run_state=0 role=primary accept_traffic=1 overlay_id=0

parent=SIA index=0
proxyid_num=1 child_num=0 refcnt=5 ilast=0 olast=42968340 ad=/0
stat: rxp=681 txp=0 rxb=92088 txb=0
dpd: mode=on-demand on=1 status=ok idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=SIA proto=0 sa=1 ref=2 serial=1 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.10.2.1-10.10.2.1:0
  SA:  ref=4 options=6a6 type=00 soft=0 mtu=1280 expire=43058/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000003 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43186/43200
```

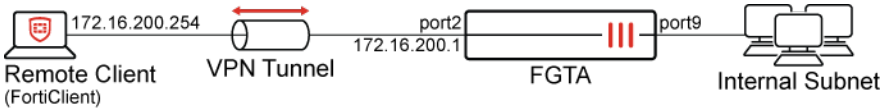
```
  dec: spi=731984eb esp=aes key=16 cff89f9d407cb6e89f385a3cc92d1d2f
       ah=sha1 key=20 aaa09fadca2939fab551eaf1a7c79889ec6f1fb5
  enc: spi=15ee0fee esp=aes key=16 73f1af3bebd239e395aa42fe7e6a9687
       ah=sha1 key=20 814b04456e1f04eb8c6fadd057b54baa3dab47fa
  dec:pkts/bytes=3/120, enc:pkts/bytes=0/0
  npu_flag=02 npu_rgwy=172.16.200.242 npu_lgwy=172.16.200.4 npu_selid=1 dec_npuid=1 enc_
npuid=0
```

Remote access VPNs with IPAM addressing NEW

The VPN wizard for creating remote access and SIA tunnels defaults to using IPAM for addressing, instead of manually entering the client IP range. When IPAM is not enabled, an option is displayed in the wizard to enable it and to assign basic IPAM parameters. Once enabled, administrators only need to select a network size in order to automatically assign a block of addresses from the IPAM address pool to the VPN clients.

Example

In this example, we enable IPAM and assign a default block of addresses for use by IPAM. Using the Remote Access VPN Wizard, we create the VPN settings and reserve a 256 address block for VPN client address assignments. IPAM automatically assigns the address range with 256 addresses from the default address block. This simplifies the role of the admin in reserving and defining an address range manually for client endpoints.

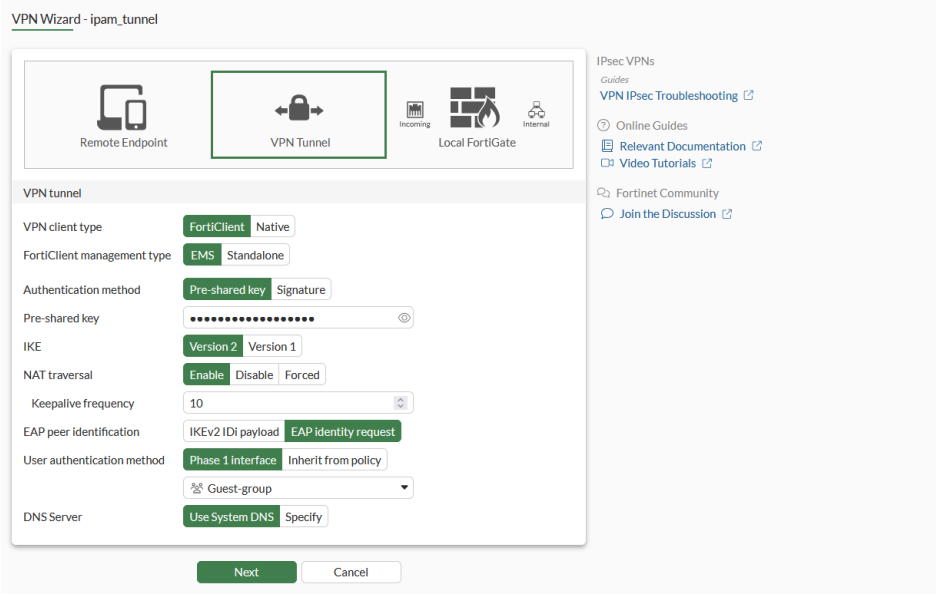


To enable IPAM and IPAM Subnet in the GUI:

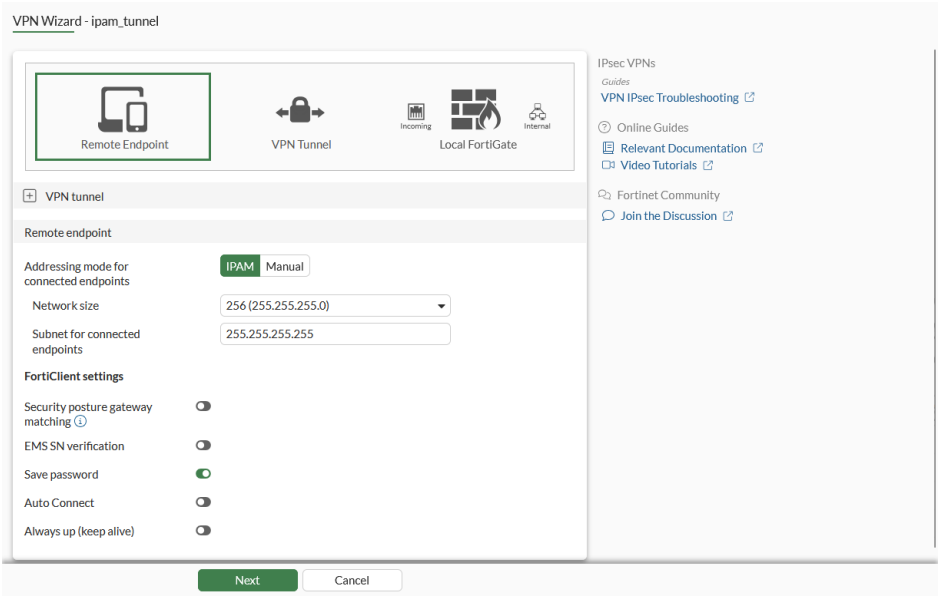
1. Go to *Network* > *IPAM* > *IPAM Settings*.
2. Set *Status* to *Enabled*.
3. Click *OK*.
4. Go to the *IPAM Subnets* tab.
5. Define a subnet to allow IPAM to assign addresses from, for example 172.31.0.0 255.255.0.0.
6. Click *OK*.

To configure an IPsec tunnel from template in the GUI:

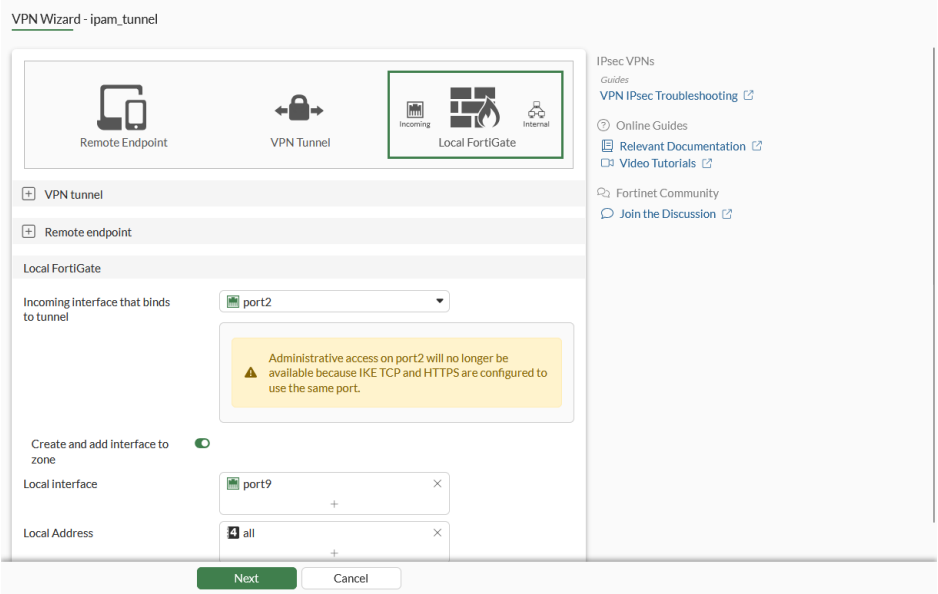
1. Go to *VPN* > *VPN Wizard*.
2. Enter a *Tunnel name* and select *Remote Access*, then click *Begin*.
3. Set *VPN client type* to *FortiClient* and *FortiClient management type* to *EMS*.
4. Set *Authentication method* to *Pre-shared key* and enter the *Pre-shared key*.
5. Set *User authentication method* to *Phase 1 interface* and select a user group.



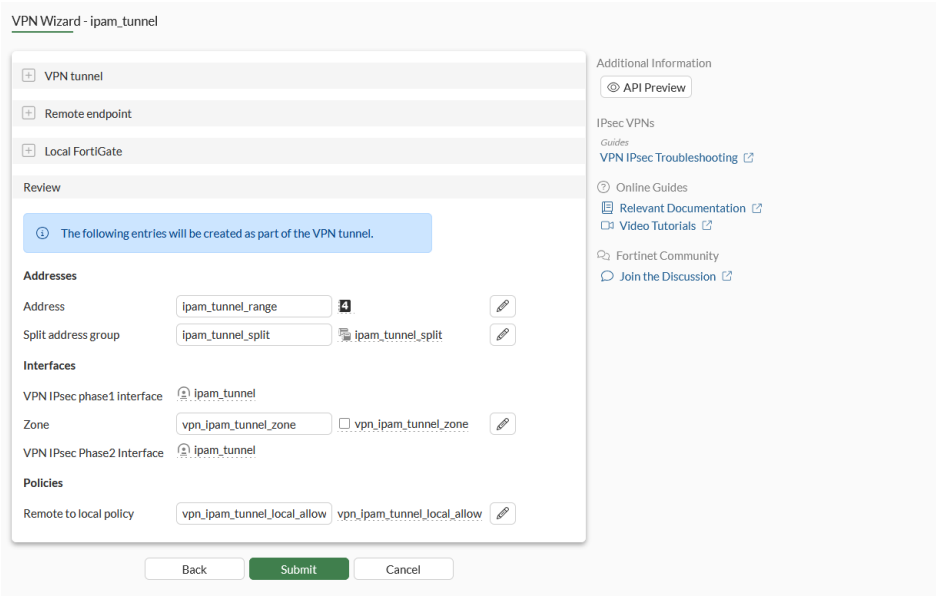
6. Click *Next*.
   Addressing mode for connected endpoints is set to *IPAM* by default. If IPAM was not previously enabled, an option *Edit IPAM Settings* is shown.
7. Set the *Network size*. This will reserve a range of addresses in the IPAM address range for VPN clients.



8. Leave *Subnet for connected endpoints* as is, and click *Next*.
9. Set the *Incoming interface that binds to tunnel*, *Local interface*, and *Local Address*.



10. Click *Next*.



11. Review the settings, then click *Submit*.

To enable IPAM and IPAM Subnet in the CLI:

```
config system ipam
    set status enable
    config pools
        edit "default-pool"
            set subnet 172.31.0.0 255.255.0.0
        next
    end
end
```

To configure an IPsec tunnel in the CLI:

```
config vpn ipsec phase1-interface
    edit "ipam_tunnel"
        set type dynamic
        set interface "port1"
        set ike-version 2
        set peertype any
        set net-device disable
        set mode-cfg enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set comments "VPN: ipam_tunnel -- Created by VPN wizard"
        set dhgrp 20 21
        set eap enable
        set eap-identity send-request
        set wizard-type dialup-forticlient
        set authusrgrp "local-group"
        set assign-ip-from name
        set dns-mode auto
        set ipv4-split-include "ipam_tunnel_split"
        set ipv4-name "ipam_tunnel_range"
        set save-password enable
        set psksecret PLAIN "0x313233343536"
    next
end
config vpn ipsec phase2-interface
    edit "ipam_tunnel"
        set phase1name "ipam_tunnel"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
        set comments "VPN: ipam_tunnel -- Created by VPN wizard"
    next
end
config firewall address
    edit "ipam_tunnel_range"
        set type ipam
        set comment "VPN: ipam_tunnel -- Created by VPN wizard"
        set subnet 172.31.0.0 255.255.255.0
    next
end

config firewall policy
    edit 1
        set srcintf "any"
        set dstintf "any"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
```

```
    next
    edit 2
        set name "vpn_ipam_tunnel_local_allow"
        set srcintf "ipam_tunnel"
        set dstintf "port9"
        set action accept
        set srcaddr "ipam_tunnel_range"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
        set comments "VPN: ipam_tunnel -- Created by VPN wizard"
    next
end
```

The firewall address type is set to `ipam`, see IPAM firewall address New on page 2051.

LDAP authentication with TCP as transport

EAP-TTLS for LDAP based user authentication in IKEv2

When configuring IPsec VPNs, the user authentication mechanism used depends on the IKE version in use:

- IKE version 1 uses XAUTH (Extended Authentication) for user authentication.
- IKE version 2 uses EAP (Extensible Authentication Protocol) for user authentication.

XAUTH in IKEv1 natively supports LDAP based user authentication. In IKEv2, LDAP based user authentication is not directly supported through all EAP methods. To authenticate users against an LDAP server in IKEv2, you must use EAP-TTLS (Tunneled Transport Layer Security). EAP-TTLS allows credentials to be securely transmitted between FortiClient and FortiGate over a TLS tunnel, ensuring secure user authentication.

EAP-TTLS configuration is supported on FortiClient EMS and FortiClient version 7.4.3 and later, see EAP-TTLS support for IPsec VPN.

Both FortiClient and FortiClient EMS must be running compatible versions to enable this functionality, see EMS Compatibility chart.

Transport options: UDP, TCP or Auto mode

Different options can be selected between transport methods such as UDP, TCP, or Auto mode for IKE negotiation and encapsulation of ESP packets.

Traditionally, dial-up IPsec VPNs use UDP as the transport protocol for IKE negotiations and encrypted traffic. IPsec VPNs can now operate over TCP, providing greater flexibility and compatibility in restrictive network environments. Using TCP as transport is beneficial in scenarios where ISPs, firewalls, or proxy servers only allow inbound and outbound TCP traffic and block UDP traffic. You can also specify a custom TCP port for IPsec communication for both FortiGate and FortiClient. For details on configuring a custom TCP port on FortiClient,

see IPsec VPN over TCP. For details on configuring a custom TCP port on FortiGate, see Dialup IPsec VPN using custom TCP port on page 3159.

Requirements for using TCP as transport:

- IKE version 2
- FortiClient version 7.4.1 or later, that supports TCP as transport

VPN over TCP can be enabled on a per-VDOM basis. When configured, a dialup VPN tunnel defaults to Auto mode. In Auto mode, IKE initially uses UDP encapsulation. If the UDP connection fails to establish within the defined time threshold, it uses TCP encapsulation as fallback.

To configure the transport mode in the GUI:

1. Go to *VPN > VPN Tunnels > Settings*.
2. Enable *Allow VPN negotiation over TCP*.
   Use TLS is supported with FortiGate as a client. It is not currently supported by FortiClient in 7.4.
3. Change TCP port for IKE/IPsec traffic as needed.

To configure the transport mode in the CLI:

```
config system settings
    set ike-tcp-service enable
    set ike-tcp-port <port>
end
```

On FortiClient EMS, you can configure the transport method using the FortiClient EMS GUI (see Example) or the XML tag `<transport_mode>`.

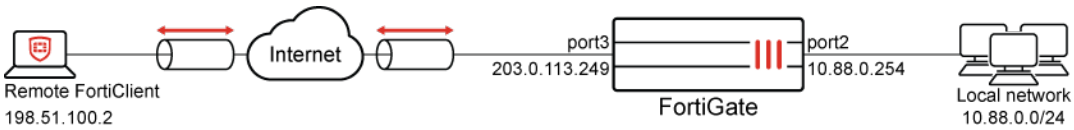
Example

This example demonstrates how to configure an IPsec VPN tunnel using IKEv2 with Auto transport protocol and LDAP for user authentication.

On FortiGate, an LDAP server named *LDAP-Server* is pre-configured to connect to LDAP directory server and is referenced in user group named *LDAP-Group*. For configuration steps, see Configuring an LDAP server on page 3509 and User groups on page 3487.

The *Group Name* option is configured to match users who belong to the *Domain Users* group on the LDAP server. As a result, only members of the *Domain Users* group are permitted to connect to the IPsec VPN. For more information on group-based user matching, see Tracking users in each Active Directory LDAP group on page 3518.

FortiClient is managed by FortiClient EMS. To enable support for both TCP transport and EAP-TTLS (required for secure LDAP authentication with IKEv2), ensure that both FortiClient and FortiClient EMS are running version 7.4.3 or later.



To configure dialup IPsec VPN using IKE version 2 and TCP as transport using GUI:

1. Enable VPN over TCP:
   a. Go to *VPN > VPN Tunnels > Settings*.
   b. Enable *Allow VPN negotiation over TCP*.
   c. Optionally, change the TCP port.
   d. Click *Apply*.
2. Go to *VPN > VPN Wizard*.
3. Enter a name for the VPN in the *Tunnel name* field. In this example, it is *dialup_vpn*.
4. Set *Select a template* to *Remote Access*.
5. Click *Begin*.
6. Configure the *VPN tunnel* settings:

Field	Value
VPN client type	*FortiClient*
FortiClient Management Type	*EMS*
Authentication method	*Pre-shared key*
Pre-shared key	Enter a suitable key. The same pre-shared key is configured on FortiClient EMS to connect to this IPsec tunnel.
IKE	*Version 2*
NAT traversal	*Enable*
Keepalive frequency	*10*
EAP peer identification	*EAP identity request*
User authentication method	*Phase 1 interface* Select *LDAP-Group* from the dropdown list. Optionally, if you have multiple user groups configured, select *Inherit from policy* and then specify the groups. See Using single or multiple user groups for user authentication on page 3024 for details.
DNS Server	*Use System DNS*

7. Click *Next*.
8. Configure the *Remote Endpoint* settings:

Field	Value
Addresses to assign to connected endpoints	*10.10.2.1-10.10.2.200*
Subnet for connected endpoints	*255.255.255.255*

Field	Value
Security posture gateway matching	*Disable*
EMS SN verification	*Disable*
Save password	*Enable*
Auto Connect	*Disable*
Always up (keep alive)	*Disable*

9. Click *Next*.
10. Configure the *Local FortiGate* settings:

Field	Value
Incoming interface that binds to tunnel	*port3*
Create and add interface to zone	*Disable*
Local interface	*port2*
Local Address	*local_network* Click *+* to create a new address object with *Type* set to *Subnet* and *IP/Netmask* set to *10.88.0.0/24*, if not already configured.

11. Click *Next*.
12. Review the configuration and click *Submit*.

To configure dialup IPsec VPN using IKE version 2 and TCP as transport using CLI:

1. Configure a LDAP server on FortiGate:

```
config user ldap
    edit "ldaps-server"
        set server "10.88.0.1"
        set server-identity-check disable
        set cnid "sAMAccountName"
        set dn "dc=fortiad,dc=info"
        set type regular
        set username "fortiad\\Administrator"
        set password ENC **********
        set secure ldaps
        set ca-cert "LDAPS-CA"
        set port 636
    next
end
```

2. Configure a user group that uses the LDAP server along with group matching (`group-name`):

```
config user group
    edit "LDAP-Group"
        set member "ldaps-server"
        config match
            edit 1
                set server-name "ldaps-server"
                set group-name "CN=Administrators,CN=Builtin,DC=fortiad,DC=info"
            next
        end
    next
end
```

3. Configure the internal interface:

```
config system interface
    edit "port2"
        set vdom "root"
        set ip 10.88.0.254 255.255.255.0
    next
end
config firewall address
    edit "local_subnet_1"
        set subnet 10.88.0.0 255.255.255.0
    next
end
config firewall addrgrp
    edit "local_network"
        set member "local_subnet_1"
    next
end
```

4. Configure the WAN interface.

   The WAN interface is the interface connected to the ISP. It can be configured in static mode (as shown here), DHCP, or PPPoE mode. It is recommended to configure it with a static IP address to ensure that the IPsec VPN configuration on the FortiClient stays unchanged if the WAN IP changes on the FortiGate. The IPsec tunnel is established over the WAN interface.

```
config system interface
    edit "port3"
        set vdom "root"
        set ip 203.0.113.249 255.255.255.0
    next
end
```

5. Configure the client address pool.

   You must create a firewall address to assign an IP address to a client from the address pool.

```
config firewall address
    edit "client_range"
        set type iprange
        set start-ip 10.10.2.1
        set end-ip 10.10.2.200
```

```
    next
end
```

6. Enable VPN over TCP. Optionally change the TCP port:

```
config system settings
    set ike-tcp-service enable
    set ike-tcp-port <port>
end
```

When using TCP port 443 for IKE/IPsec traffic, GUI access can be affected for interfaces that are bound to an IPsec tunnel when the GUI admin port is also using port 443. To ensure continued functionality, change either the IKE/IPsec port or the administrative access port.

To change the administrative access port:

```
config system global
    set admin-sport <port>
end
```

Command	Description
`admin-sport <port>`	Set the administrative access port for HTTPS (1 - 65535, default = 443).

For port conflicts with ZTNA and SSL VPN, ZTNA and SSL VPN will take precedence. To avoid any port conflicts with other services, review the FortiOS Ports guide for other incoming ports used on the FortiGate.

7. Configure the IPsec phase 1 interface:

```
config vpn ipsec phase1-interface
    edit "dialup_vpn"
        set type dynamic
        set interface "port3"
        set ike-version 2
        set peertype any
        set net-device disable
        set mode-cfg enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set dhgrp 20 21
        set eap enable
        set eap-identity send-request
        set authusrgrp "LDAP-Group"
        set assign-ip-from name
        set ipv4-name "client range"
        set dns-mode auto
        set ipv4-split-include "local_network"
        set save-password enable
        set psksecret ******
```

```
    next
end
```

8. Configure the IPsec phase 2 interface:

```
config vpn ipsec phase2-interface
    edit "dialup_vpn"
        set phase1name "dialup_vpn"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
    next
end
```

9. Configure a firewall policy to allow client traffic flow over the IPsec VPN tunnel to the FortiGate's LAN (port2) interface:

```
config firewall policy
    edit 1
        set name "inbound"
        set srcintf "dialup_vpn"
        set dstintf "port2"
        set srcaddr "client_range"
        set dstaddr "local_network"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end
```

One or more user groups for user authentication can be configured under a single IPsec VPN tunnel. See Using single or multiple user groups for user authentication on page 3024 for details.

To configure the Remote Access profile to use EAP-TTLS with IKEv2 on FortiClient EMS:

1. On FortiClient EMS, navigate to *Endpoint Profiles* > *Remote Access* and click *Edit* on the *Default* endpoint profile.
2. Under *VPN Tunnels*, click *Add Tunnel*.
3. Set *VPN type* as *Manual*.
4. Click *Next*.
5. Under *Basic Settings*, enter the following:

Field	Value
Name	*IPsec-IKEv2*
Remote Gateway	*203.0.113.249*
Authentication Method	*Pre-shared Key*

Field	Value
Pre-Shared Key	Enter pre-shared key that was configured on FortiGate's dialup VPN configuration.
Prompt for Username	*Enabled*

6. Click *VPN Settings*, enter the following:

Field	Value
IKE	*Version 2*
Address Assignment	*Mode Config*
Encapsulation	*Auto* (Optional) To use TCP mode only, set to *IPsec over TCP*. Otherwise Auto will attempt UDP first, and fallback to TCP.
IKE TCP Port	*443*
IKE UDP Port	*500*

7. Click *Phase 1*, enter the following details:

Field	Value
IKE Proposal	
Encryption, Authentication	*AES128, SHA256*
Encryption, Authentication	*AES256, SHA256*
DH Groups	*20, 21*
Key Life	*86400*
EAP Authentication Method	*EAP-TTLS* This setting is available in the GUI of FortiClient 7.4.4 and later.
Dead Peer Detection	*Enable*
NAT Traversal	*Enable*

8. Click *Phase 2*, enter the following details:

Field	Value
IKE Proposal	
Encryption, Authentication	*AES128, SHA256*
Encryption, Authentication	*AES256, SHA256*

Field	Value
DH Groups	*20*
Key Life	*43200*
Enable Replay Detection	*Enable*
Enable Perfect Forward Secrecy (PFS)	*Enable*

9. Click *Save* to save the VPN Tunnel configuration.
10. Click *Save* to save *Remote Access* endpoint profile.

To connect and fallback to TCP:

In this example, UDP is blocked by the ISP. FortiClient will detect that connection over TCP fails, so it will connect over TCP.

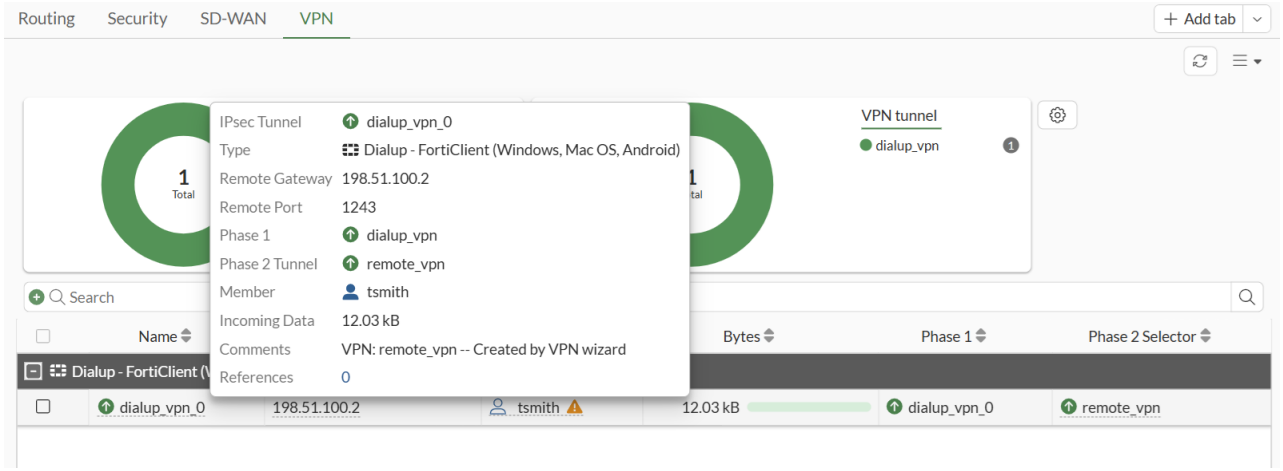
1. On FortiClient endpoint that received the configuration, in the *Remote Access* tab, find the tunnel *IPsec-IKEv2* and click *Connect*.
   The client will first initiate the connection over UDP. After some time, it will fallback to TCP.
2. Once the VPN connection is initiated over TCP, user is prompted for their LDAP credentials. Do not wait too long to enter the username and password or the connection will timeout.
   Once the correct credentials are entered, the connection will establish.

To verify VPN tunnel status using FortiGate GUI:

1. On the FortiGate, go to *Dashboard > Network Monitor > VPN*.
2. Find the new tunnel, *dialup_vpn_0* in this example, and check the statuses of the *Phase 1* and *Phase 2* selectors.

   Successfully established dialup IPsec VPN tunnels are displayed using a parent name followed by an index, creating a structured and organized association with the parent tunnel. For example, the parent name *dialup_vpn* results in tunnel names such as *dialup_vpn_0*, *dialup_vpn_1*, and so on.

3. Hover the cursor over the tunnel name to see additional details.



To verify VPN tunnel status:

1. Verify the IPsec Phase 1 tunnel status on the FortiGate, either using the CLI or the Command palette (CTRL+P) > CLI diagnostics:

```
# diagnose vpn ike gateway list name dialup_vpn_0

vd: root/0
name: dialup_vpn_0
version: 2
interface: port3 5
addr: 203.0.113.249:443 -> 198.51.100.2:1243
tun_id: 10.10.2.1/::10.0.0.8
remote_location: 0.0.0.0
network-id: 0
transport: TCP
tcp-tx-1m-rate : 0 kbps
tcp-tx-5m-rate : 0 kbps
tcp-tx-drop    : 0
tcp-rx-1m-rate : 0 kbps
tcp-rx-5m-rate : 0 kbps
tcp-rx-drop    : 0
created: 276s ago
eap-user: tsmith
2FA: no
peer-id: 198.51.100.2
peer-id-auth: no
FortiClient UID: 9A016B5A6E914B42AD4168C066EB04CA
assigned IPv4 address: 10.10.2.1/255.255.255.255
nat: me peer
pending-queue: 0
PPK: no
IKE SA: created 1/1  established 1/1  time 15390/15390/15390 ms
IPsec SA: created 1/1  established 1/1  time 20/20/20 ms

  id/spi: 22915 7d15b5fbc22c6251/41ed29e125e71ad3
```

```
  direction: responder
  status: established 276-261s ago = 15390ms
  proposal: aes128-sha256
  child: no
  SK_ei: c8bde5a7fb91c50c-b2e8f23654b5f4e7
  SK_er: e9a54853408a154f-9e2221740b993822
  SK_ai: 70eec5f71fa1c630-307098f3d8d3c88e-1ce71f56708aecde-29d779709f7f59cf
  SK_ar: c7cf4ea6936ac970-c45518b44b3b4820-6e0ed1e34b3f140e-37e24a58cf868a29
  PPK: no
  message-id sent/recv: 0/37
  QKD: no
  PQC-KEM (IKE): no
  PQC-KEM (all IPsec): no
  lifetime/rekey: 86400/85868
```

Note that FortiGate acts as the responder, establishes the IPsec tunnel with remote FortiClient user with IP 198.51.100.2 over TCP port 443 on its port3 interface, and dynamically assigns a VPN IP of 10.10.2.1 to FortiClient endpoint. The authenticated username is also displayed in eap-user field.

2. Verify the IPsec Phase 2 tunnel status on the FortiGate:

```
# diagnose vpn tunnel list name dialup_vpn_0

list all ipsec tunnel in vd 0
------------------------------------------------------
name=dialup_vpn_0 ver=2 serial=d 203.0.113.249:443->198.51.100.2:1243 nexthop= tun_
id=10.10.2.1 tun_id6=::10.0.0.8 status=up dst_mtu=0 weight=1 country=ZZ
bound_if=5 real_if=0 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none options[0x23a8]=npu
rgwy-chg rport-chg frag-rfc  run_state=0 role=primary accept_traffic=1 overlay_id=0

parent=dialup_vpn index=0
proxyid_num=1 child_num=0 refcnt=6 ilast=27 olast=44096403 ad=/0
stat: rxp=404 txp=0 rxb=32239 txb=0
dpd: mode=on-demand on=1 status=ok idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=remote_vpn proto=0 sa=1 ref=2 serial=1 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.10.2.1-10.10.2.1:0
  SA:  ref=3 options=20682 type=00 soft=0 mtu=1280 expire=42697/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000194 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43186/43200
  dec: spi=0cbfd916 esp=aes key=16 8b28c31eb8705e1f9ccce460dc38c0bd
       ah=sha256 key=32 deaabb9a355830bcad14a28a8ed46932cf52ea8b4eec0dc381714d90c7173931
  enc: spi=215028aa esp=aes key=16 f4f6b1d4bde88bda7b98bd1210e298e6
       ah=sha256 key=32 d4da5de14ef9b6270540e5a3f75b69b229b0506c6fc7ecc4dc1816e2c15d861a
  dec:pkts/bytes=404/32239, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=0.0.0.0:0 npu_lgwy=0.0.0.0:0 npu_selid=0
  dec_npuid=0 enc_npuid=0 dec_engid=-1 enc_engid=-1 dec_saidx=-1 enc_saidx=-1
```

Note the decryption, encryption, and authentication keys negotiated in Phase 2 to protect subnets under the `src` and `dst` fields. FortiOS also automatically adds the dynamically assigned VPN IP of 10.10.2.1 to its traffic selectors for dialup tunnel dialup_vpn_0.

To verify traffic through the IPsec tunnel:

1. On the remote FortiClient user with IP address 198.51.100.2, initiate ICMP traffic to a machine in the 10.88.0.0/24 network and verify that traffic goes through:

```
# ping 10.88.0.7
Pinging 10.88.0.7 with 32 bytes of data:
Reply from 10.88.0.7: bytes=32 time=2ms TTL=63
Reply from 10.88.0.7: bytes=32 time=2ms TTL=63
```

2. On FortiGate, run a sniffer trace for ICMP packets destined to 10.10.111.3:

```
# diagnose sniffer packet any 'host 10.10.111.3’ 4 0 l

Using Original Sniffing Mode
interfaces=[any]
filters=[host 10.88.0.7 and icmp]
2026-04-13 21:06:46.543624 dialup_vpn in 10.10.2.1 -> 10.88.0.7: icmp: echo request
2026-04-13 21:06:46.543694 port2 out 10.88.0.254 -> 10.88.0.7: icmp: echo request
2026-04-13 21:06:46.544223 port2 in 10.88.0.7 -> 10.88.0.254: icmp: echo reply
2026-04-13 21:06:46.544247 dialup_vpn out 10.88.0.7 -> 10.10.2.1: icmp: echo reply
```

   The ICMP request packets are received on dialup_vpn_0 tunnel interface and routed to the port2 interface. ICMP reply packets are received on the port2 interface and routed back out on dialup_vpn_0 tunnel interface.

   For more information, see Performing a sniffer trace or packet capture on page 4966.

3. Optionally, run real time IKE debug using appropriate filters for troubleshooting if the IPsec tunnel does not come up:

```
# diagnose vpn ike log filter rem-addr4 198.51.100.2
# diagnose debug application ike -1
# diagnose debug enable
```

   For more information, see VPN IPsec troubleshooting on page 3335.

To verify LDAP authentication using EAP-TTLS on FortiClient:

1. Ensure that the logging level is set to *Debug* on FortiClient.
2. After the VPN tunnel establishes on FortiClient, go to the folder located under *C:\%ProgramFiles%\Fortinet\FortiClient\logs\trace* and open the file *FortiIKE_1.log*. The following shows expected log output for this feature:

```
[7868:11132] [FortiIKE  524    info] IPsec daemon start ipsec.exe -s FC_{73EFB30F-1CAD-4a7a-
AE2E-150282B6CE25}_000002 -i 2 -g 198.51.100.2 IPsec-IKEv2

[7868:11132] [FortiIKE  419   debug] EAP-TTLS method loaded
[7868:11132] [FortiIKE  427   debug] EAP-GTC method loaded

[7868:11132] [FortiIKE  4939   debug] 0:IPsec-IKEv2:0: initiator received EAP msg
[7868:11132] [FortiIKE  400   debug] EAP: Initialize selected EAP method: vendor 0 method 21
(TTLS)
```

```
[7868:11132] [FortiIKE  145   debug] EAP-TTLS: Phase2 type: PAP
[7868:11132] [FortiIKE  4779   debug] 0:IPsec-IKEv2:0: authentication succeeded
```

VPN 2FA with EAP and certificate authentication

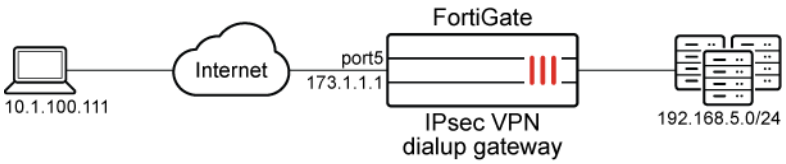
IPsec IKEv2 VPNs now support certificate authentication and EAP authentication at the same time from a dialup FortiClient. With the eap-cert-auth setting enabled, FortiGate will validate the X.509 certificate sent from FortiClient when establishing an IKEv2 tunnel. After it is authenticated, FortiGate will continue to perform EAP authentication.

```
config vpn ipsec phase1-interface
    edit <tunnel>
        set eap enable
        set eap-cert-auth {*disable | enable}
        set peer <user peer>
    next
end
```

Setting	Description
`eap enable`	Enable EAP to perform user authentication.
`eap-cert-auth {*disable | enable}`	Enable/Disable certificate authentication in addition to EAP (default = disable).
`peer <user peer>`	When eap-cert-auth is enabled, the peer certificate is validated if `peer <user peer>` is configured. If eap-cert-auth is disabled, the peer certificate is not validated even if `peer <user peer>` is configured.

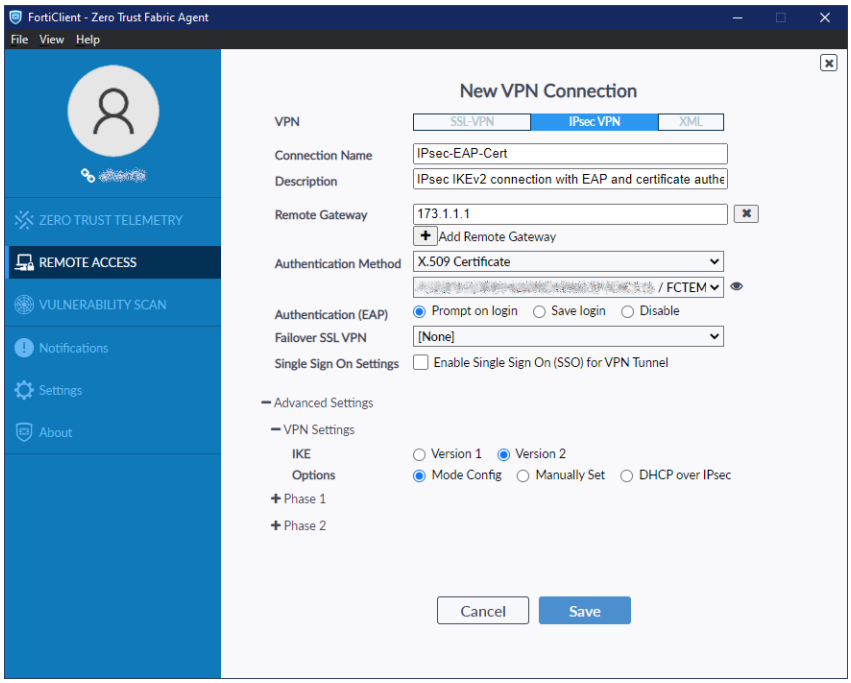
Example

In this example, FortiClient is configured to use an X.509 certificate and EAP to perform certificate and user authentication. The certificate is issued by a CA that is trusted by the FortiGate.



To configure FortiClient:

1. Go to *Remote Access*, click the sandwich menu, and select *Add a new connection*.
2. Configure the new VPN connection.



Field	Value
VPN	IPsec VPN
Connection Name	IPsec-EAP-Cert
Description	(Optional)
Remote Gateway	173.1.1.1
Authentication Method	X.509 Certificate
Authentication (EAP)	Prompt on login
Failover SSL VPN	[None]
Advanced Settings > VPN Settings	
IKE	Version 2
Options	Mode Config

3. Click *Save*.

To configure local users and peer (certificate authentication):

```
config user group
    edit "local-group"
        set member "usera" "userc" "bi_local"
    next
end
config user peer
    edit "peer-test3"
        set ca "Eternity-CA"
```

```
        set subject "test3"
    next
end
```

A certificate issued by Eternity-CA with `test3` in the subject will be trusted by the FortiGate.

To configure the IPsec VPN phase1:

```
config vpn ipsec phase1-interface
    edit "fct-eapcert"
        set type dynamic
        set interface "port5"
        set ike-version 2
        set authmethod signature
        set net-device disable
        set mode-cfg enable
        set proposal aes128-sha1 aes256-sha256
        set comments "VPN: fct3459 -- Created by VPN wizard"
        set dhgrp 20
        set eap enable
        set eap-identity send-request
        set eap-cert-auth enable
        set wizard-type dialup-forticlient
        set authusrgrp "local-group"
        set certificate "test1"
        set peer "peer-test3"
        set ipv4-start-ip 3.3.1.1
        set ipv4-end-ip 3.3.1.100
        set dns-mode auto
        set ipv4-split-include "fct3459_split1"
        set save-password enable
        set client-auto-negotiate enable
        set client-keep-alive enable
    next
end
```

The FortiGate will validate the client certificate based on the eap-cert-auth setting. The following examples show different outcomes from different configuration scenarios.

Scenario 1: eap-cert-auth is enabled and FortiGate receives a valid and trusted certificate

Using IKE debugs, we can observe the steps when FortiGate receives the connection request with the client certificate, validates it, and continues to perform EAP authentication before establishing the VPN connection.

```
# diagnose debug application ike -1
# diagnose debug enable
ike V=root:0:fct-eapcert:11: received FCT-UID : 4E98BCE511844FA8B161FA5DF047B12C
ike V=root:0:fct-eapcert:11: received peer identifier DER_ASN1_DN 'C = CA, ST = BC, L = Burnaby, O
= Fortinet, OU = QA, CN = test3'
ike V=root:0:fct-eapcert:11: re-validate gw ID
```

```
ike V=root:0:fct-eapcert:11: gw validation OK
ike V=root:0:fct-eapcert:11: Validating X.509 certificate
ike V=root:0:fct-eapcert:11: peer cert, subject='test3', issuer='Eternity-CA'
ike V=root:0:fct-eapcert:11: peer ID verified
ike V=root:0:fct-eapcert:11: building fnbam peer candidate list
ike V=root:0:fct-eapcert:11: FNBAM_GROUP_NAME candidate 'peer-test3'
ike V=root:0:fct-eapcert:11: certificate validation before eap pending
ike V=root:0:fct-eapcert:11: fnbam reply 'peer-test3'
ike V=root:0:fct-eapcert:11: certificate validation complete
ike V=root:0:fct-eapcert:11: responder preparing EAP identity request
ike V=root:0:fct-eapcert:11: local cert, subject='test1', issuer='Eternity-CA'
ike 0:fct-eapcert:11: enc
…
ike V=root:0:fct-eapcert:11: sent IKE msg (AUTH_RESPONSE): 173.1.1.1:500->10.1.100.111:500,
len=1420, vrf=0, id=e955507be9a857eb/b48a41c47db35bdb:00000001, oif=13
ike V=root:0: comes 10.1.100.111:500->173.1.1.1:500,ifindex=13,vrf=0,len=92....
…
ike V=root:0:fct-eapcert:11: responder received EAP msg
ike V=root:0:fct-eapcert:11: send EAP message to FNBAM
ike V=root:0:fct-eapcert:11: initiating EAP authentication
ike V=root:0:fct-eapcert: EAP user "bi_local"
ike V=root:0:fct-eapcert: auth group local-group
ike V=root:0:fct-eapcert: EAP 1600011514 pending
ike V=root:0:fct-eapcert:11 EAP 1600011514 result FNBAM_CHALLENGED
ike V=root:0:fct-eapcert: EAP challenged for user "bi_local"
ike V=root:0:fct-eapcert:11: responder preparing EAP pass through message
…
```

Checking the IPsec VPN tunnel list:

```
# diagnose vpn tunnel list
list all ipsec tunnel in vd 0
…
name=fct-eapcert_0 ver=2 serial=6 173.1.1.1:0->10.1.100.111:0 tun_id=3.3.1.1 tun_id6=::10.0.0.5
dst_mtu=1500 dpd-link=on weight=1
bound_if=13 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/74408 options[122a8]=npu rgwy-chg
frag-rfc  run_state=0 role=primary accept_traffic=1 overlay_id=0
parent=fct-eapcert index=0
proxyid_num=1 child_num=0 refcnt=5 ilast=1 olast=13 ad=/0
stat: rxp=2 txp=97 rxb=16460 txb=53465
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=fct3459 proto=0 sa=1 ref=6 serial=1 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:3.3.1.1-3.3.1.1:0
…
```

The result shows that the tunnel is established.

Scenario 2: eap-cert-auth is enabled and FortiGate receives an invalid certificate

In this example, a user peer object is used that does not trust the issuer Eternity-CA.

```
# diagnose debug application ike -1
# diagnose debug enable
ke V=root:0:fct-eapcert:9: received FCT-UID : 4E98BCE511844FA8B161FA5DF047B12C
ike V=root:0:fct-eapcert:9: received peer identifier DER_ASN1_DN 'C = CA, ST = BC, L = Burnaby, O
= Fortinet, OU = QA, CN = test3'
ike V=root:0:fct-eapcert:9: re-validate gw ID
ike V=root:0:fct-eapcert:9: gw validation OK
ike V=root:0:fct-eapcert:9: Validating X.509 certificate
ike V=root:0:fct-eapcert:9: peer cert, subject='test3', issuer='Eternity-CA'
ike V=root:0:fct-eapcert:9: peer ID verified
ike V=root:0:fct-eapcert:9: building fnbam peer candidate list
ike V=root:0:fct-eapcert:9: FNBAM_GROUP_NAME candidate 'peer-test3'
ike V=root:0:fct-eapcert:9: certificate validation before eap pending
ike V=root:0:fct-eapcert:9: fnbam cert group matching failed
ike V=root:0:fct-eapcert:9: certificate validation failed
```

The result shows that the tunnel is not established.

Scenario 3: eap-cert-auth is disabled and FortiGate receives an invalid certificate

In this example, eap-cert-auth is disabled, so the incoming peer certificate is not validated.

```
# diagnose debug application ike -1
# diagnose debug enable
ike V=root:0:fct-eapcert:3: received FCT-UID : 4E98BCE511844FA8B161FA5DF047B12C
ike V=root:0:fct-eapcert:3: received peer identifier DER_ASN1_DN 'C = CA, ST = BC, L = Burnaby, O
= Fortinet, OU = QA, CN = test3'
ike V=root:0:fct-eapcert:3: re-validate gw ID
ike V=root:0:fct-eapcert:3: gw validation OK
ike V=root:0:fct-eapcert:3: responder preparing EAP identity request
ike V=root:0:fct-eapcert:3: local cert, subject='test1', issuer='Eternity-CA'
ike 0:fct-eapcert:3: enc
…
ike V=root:0:fct-eapcert:3: responder received EAP msg
ike V=root:0:fct-eapcert:3: send EAP message to FNBAM
ike V=root:0:fct-eapcert:3: initiating EAP authentication
ike V=root:0:fct-eapcert: EAP user "bi_local"
ike V=root:0:fct-eapcert: auth group local-group
ike V=root:0:fct-eapcert: EAP 1600011502 pending
ike V=root:0:fct-eapcert:3 EAP 1600011502 result FNBAM_CHALLENGED
ike V=root:0:fct-eapcert: EAP challenged for user "bi_local"
ike V=root:0:fct-eapcert:3: responder preparing EAP pass through message
```

Checking the IPsec VPN tunnel list:

```
# diagnose vpn tunnel list
list all ipsec tunnel in vd 0
…
name=fct-eapcert_0 ver=2 serial=5 173.1.1.1:0->10.1.100.111:0 tun_id=3.3.1.1 tun_id6=::10.0.0.4
dst_mtu=1500 dpd-link=on weight=1
bound_if=13 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/74408 options[122a8]=npu rgwy-chg
frag-rfc  run_state=0 role=primary accept_traffic=1 overlay_id=0
```

```
parent=fct-eapcert index=0
proxyid_num=1 child_num=0 refcnt=5 ilast=0 olast=0 ad=/0
stat: rxp=641 txp=747 rxb=16325 txb=471587
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=fct3459 proto=0 sa=1 ref=6 serial=1 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:3.3.1.1-3.3.1.1:0
```

The result shows that the tunnel is established even though the client sent an untrusted certificate.

Using single or multiple user groups for user authentication

User groups enable remote users to authenticate themselves before establishing a connection to the IPsec VPN tunnel. One or more user groups for user authentication can be configured under a single IPsec VPN tunnel. For information about configuring user groups, see User groups on page 3487.

When configuring dialup or remote access IPsec VPNs, user groups can be specified in either the phase 1 settings or the firewall policy associated with the tunnel. They must not be configured in both.

The workflow for configuring user groups varies depending on whether the IPsec tunnel is set up using the VPN Wizard or a custom IPsec configuration.

VPN Wizard IPsec tunnel configuration

When configuring the tunnel, you can specify one or more user groups in the *User authentication method* option.

- Single user group:

  Set *User Authentication Method* to either:

  - *Phase 1 Interface*: Select the required user group.
  - *Inherit from Policy*: Select the user group, which will automatically populate the *User/Group* fields in the firewall policies created by the wizard.
- Multiple user groups:

  Set *User Authentication Method* to *Inherit from Policy* and select the required user groups. These will be added to the *User/Group* fields in the firewall policies created by the wizard.
- For any additional firewall policies created outside the VPN Wizard, administrators must manually assign the appropriate user group or groups in the *User/Group* fields.

Custom IPsec tunnel configuration

The custom IPsec tunnel configuration offers more granular settings to configure IPsec VPN. To configure a custom IPsec VPN tunnel, go to *VPN > VPN Tunnels* and click *Create New > Custom IPsec tunnel*.

- Single user group:
  - GUI:

    In the VPN tunnel configuration, enable XAUTH for IKEv1 or EAP for IKEv2, and either:
    - Set *User Group* to *Specify* and select the required user group, or
    - Set *User Group* to *Inherit from Policy* and in the policy configuration, specify the user group in the *User/group* field.
  - CLI:

    For IKEv1, enable XAUTH:

    ```
    config vpn ipsec phase1-interface
        edit <tunnel-name>
            set xauthtype {pap | chap | auto}
        next
    end
    ```

    For IKEv2, enable EAP:

    ```
    config vpn ipsec phase1-interface
        edit <tunnel-name>
            set eap enable
            set eap-identity send-request
        next
    end
    ```

    In the VPN tunnel configuration, either:
    - Set the user group (`set authusrgrp <group-name>`), or
    - Remove any configured user group (`unset authusrgrp`), then specify the user group in the firewall policy (`set groups <group-name>`).
- Multiple user groups:
  - GUI:

    In the VPN tunnel configuration, enable XAUTH for IKEv1 or EAP for IKEv2, and set *User Group* to *Inherit from Policy*. In the policy configuration, specify the user groups in the *User/group* field.
  - CLI:

    For IKEv1, enable XAUTH:

    ```
    config vpn ipsec phase1-interface
        edit <tunnel-name>
            set xauthtype {pap | chap | auto}
        next
    end
    ```

    For IKEv2, enable EAP:

    ```
    config vpn ipsec phase1-interface
        edit <tunnel-name>
            set eap enable
            set eap-identity send-request
        next
    end
    ```

In the VPN tunnel configuration, remove any configured user group (`unset authusrgrp`), then specify the user groups in the firewall policy (`set groups <group-name> ... <group-name>`).

Starting with FortiClient 7.4.4, IKEv1 is no longer supported on the client. IKEv1 is now considered a legacy configuration.

Also, FortiClient 7.4.4 does not support IPv6. Use FortiClient 7.4.6 or later.

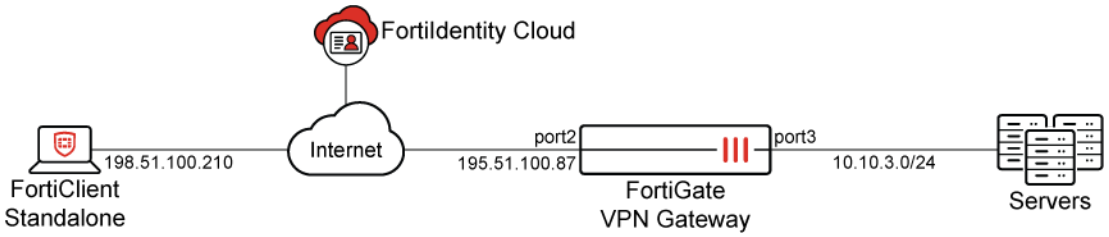
VPN with FortiClient Standalone and FortiIdentity Cloud NEW

The FortiClient Standalone client can be licensed without FortiClient EMS, and operates as an unmanaged VPN client. This FortiClient is licensed through FortiCloud with the FortiClient Standalone license. Once registered, the end-users can be provisioned through FortiIdentity Cloud (FIC), which offers Single-Sign-On (SSO) via SAML.

The VPN Wizard on the FortiGate integrates FortiIdentity Cloud configurations into the wizard for simple Remote Access VPN setup for FortiClient Standalone. SAML configurations configured through the VPN wizard are automatically synchronized with FortiIdentity Cloud.

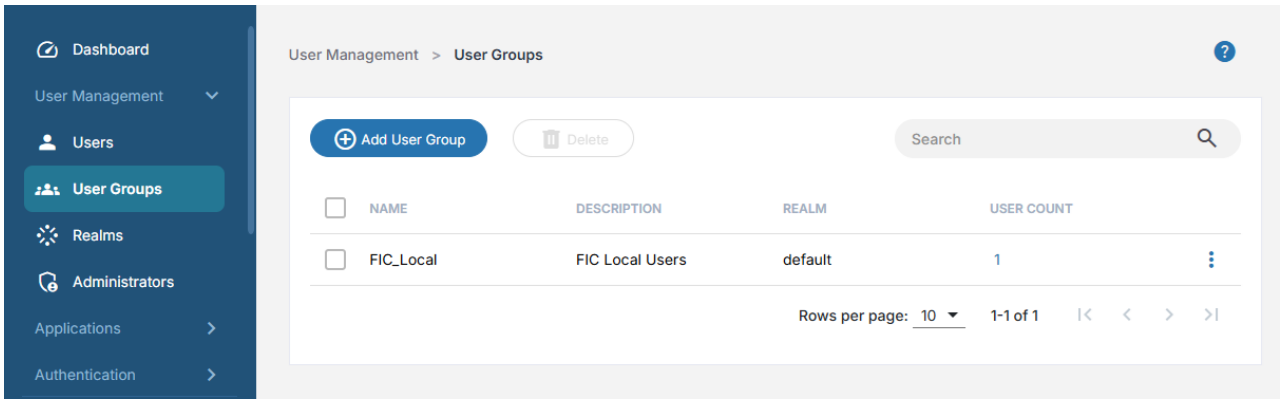
For more information about FortiIdentity Cloud SSO integration with the FortiGate, see SSO integration with FortiIdentity Cloud on page 3615.

Topology

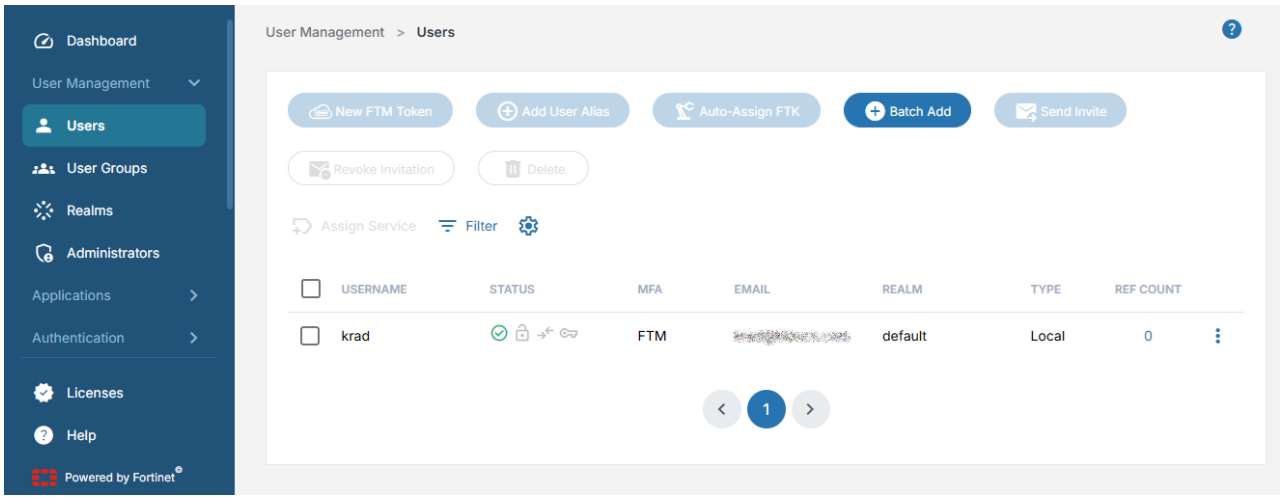


Prerequisites

- FortiClient Standalone license registered on the FortiCloud account
- FIC is licensed
- A realm and user group is already configured on FIC (the default realm can be used)



- A user has been activated and FortiClient invitation has been sent



- The FortiGate is registered to the same FortiCloud account as FIC
- The FortiClient Standalone client is installed on the user's endpoint
  FortiClient EMS is not required

Example

In this example, a Local User group with users hosted on FIC is provisioned. Users are activated and FortiClient VPN invitations are sent to each user. The FortiGate VPN Gateway is configured to allow remote access from standalone clients instead of EMS managed clients. Remote users can access the servers behind the FortiGate.

To configure the VPN with the VPN Wizard:

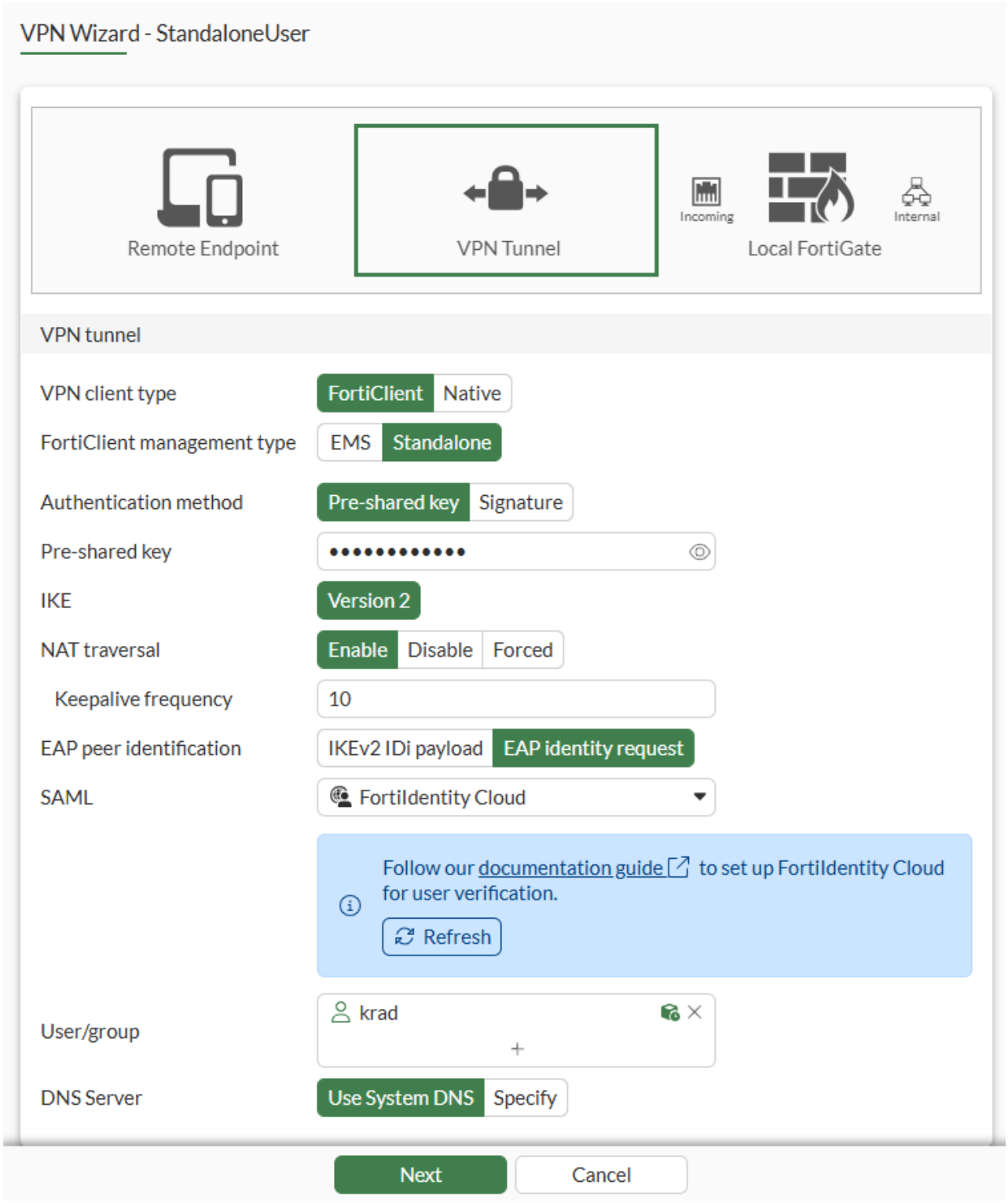
1. Go to *VPN > VPN Wizard*.
2. Name the VPN tunnel *StandaloneUser*, select *Remote Access* template, and click *Begin*
3. Set *VPN client type* to *FortiClient*.
4. Set *FortiClient management type* to *Standalone*.
5. In the pop-up, click *Setup SSO*:

a. Edit the *SSO Service Provider Configuration* as needed.
b. Under *Identity Provider Configuration* select the following values, which are retrieved from the FIC account:

Field	Value
Realm	default
User source	LocalIDP_default
Certificate	FIC_Default_Cert

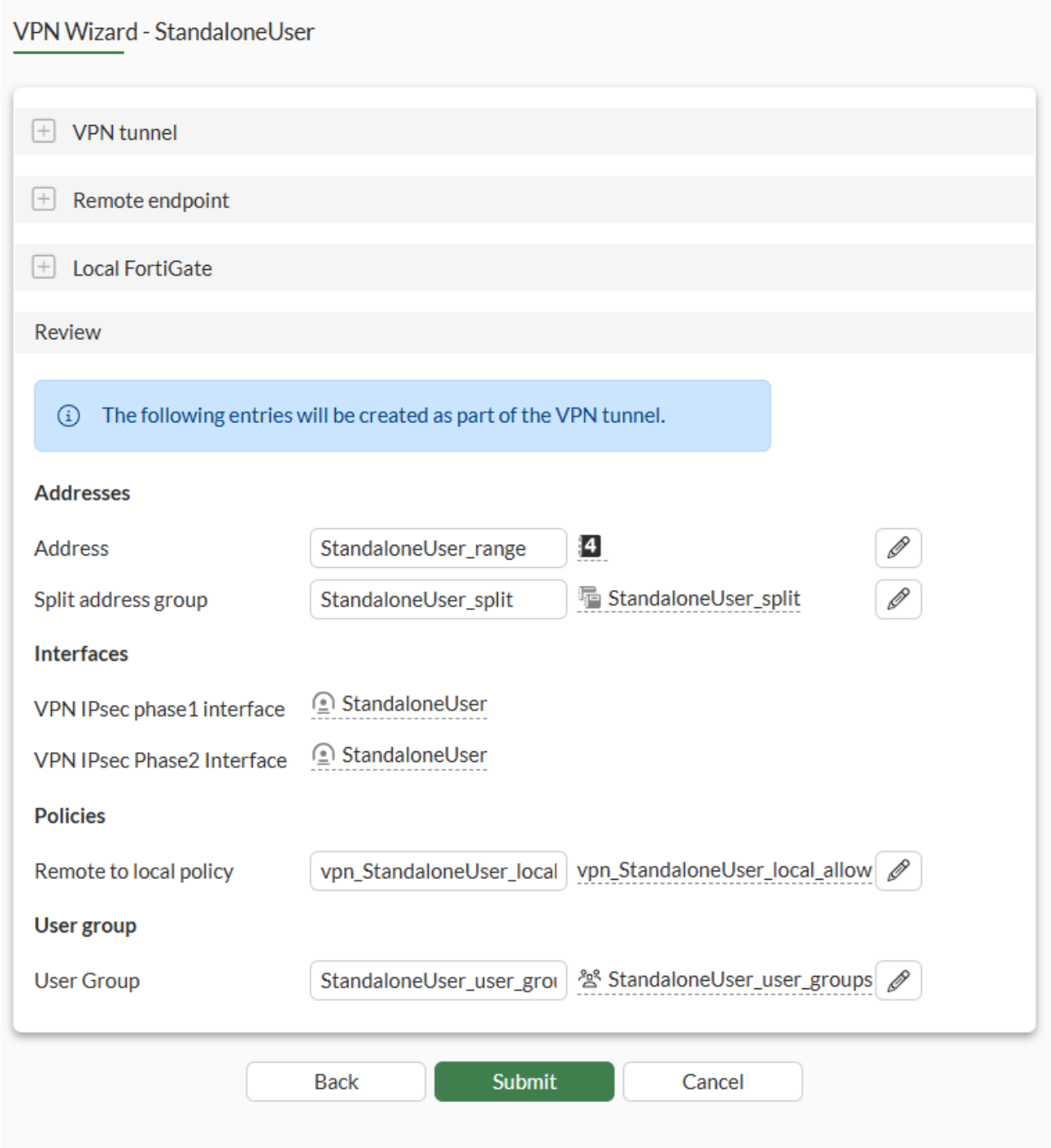
c. Under *Additional SAML Attributes*, set *Attribute used to identify groups* to *group*.
d. Click *OK*.

6. Enter *Pre-Shared key* or switch to *Signature* for certificate authentication.
7. Verify *EAP peer identification* is set to *EAP identity request*.
8. Ensure *SAML* is configured to the SAML server created for FIC.
9. For *User/group*, select the individual users.



10. Click *Next*.
11. For *Addressing mode*, either use *IPAM* and select your *Network size*, or switch to *Manual* and assign address range for your endpoints.
12. Click *Next*.

13. Select *Incoming interface*, *Local interface*, and *Local Address*.
14. Disable *Create and add interface to zone*.
15. Click *Next*.
16. Review your settings.

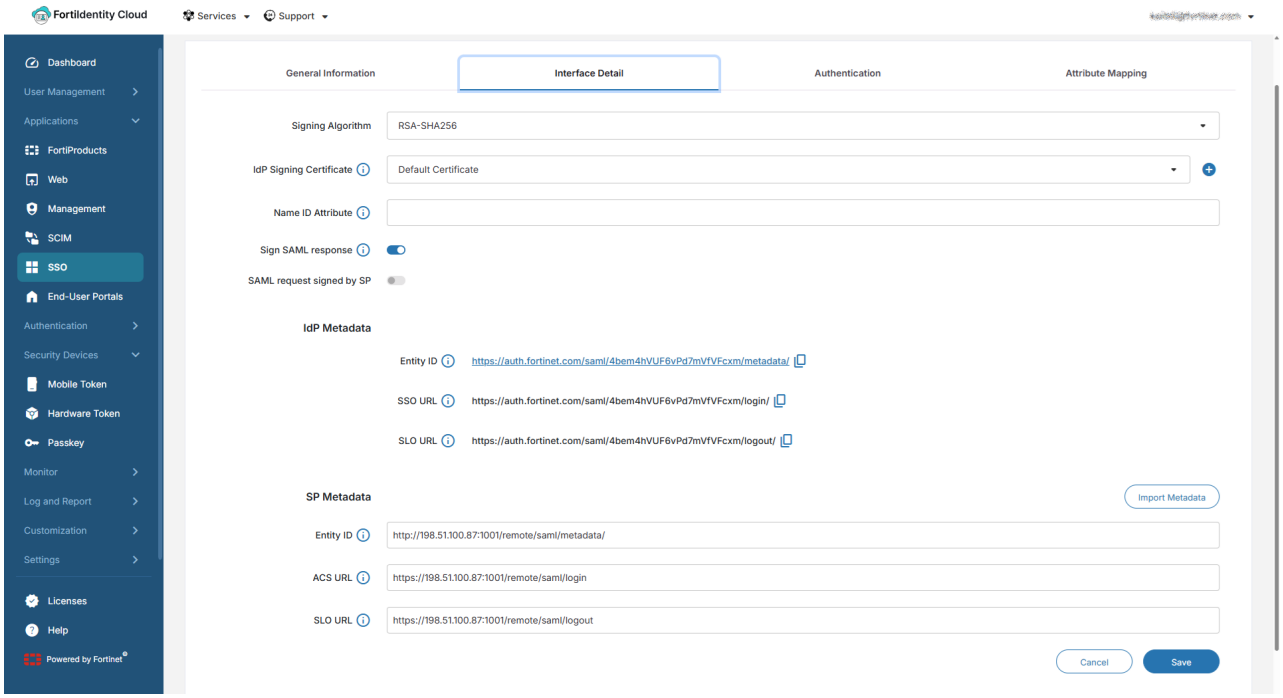


- Split tunneling will be enabled to route only traffic to the local subnet to the tunnel.
- A policy will be created to allow traffic to the local subnet.
- A user group will be created containing the user(s) you selected.

17. Click *Submit*.

To verify SAML settings on FortiIdentity Cloud:

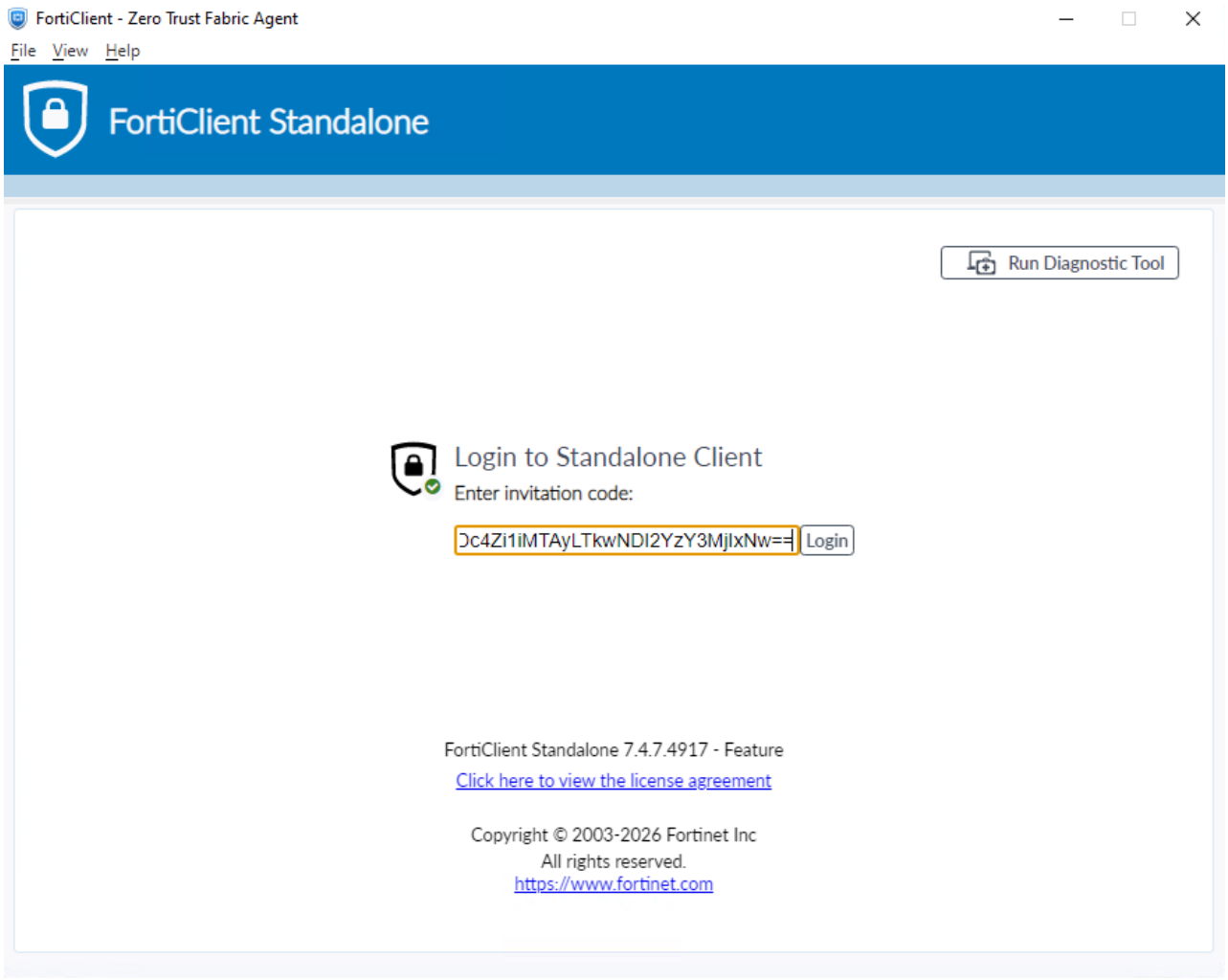
1. Login to FIC.
2. Go to *Applications* > *SSO*.
   A new SSO Application called “FortiIdentity Cloud” is automatically created when the FortiGate SSO configurations were set up.
3. Click *Edit* to see the details.
4. On the *Interface Detail* tab, the IdP and SP information are displayed:



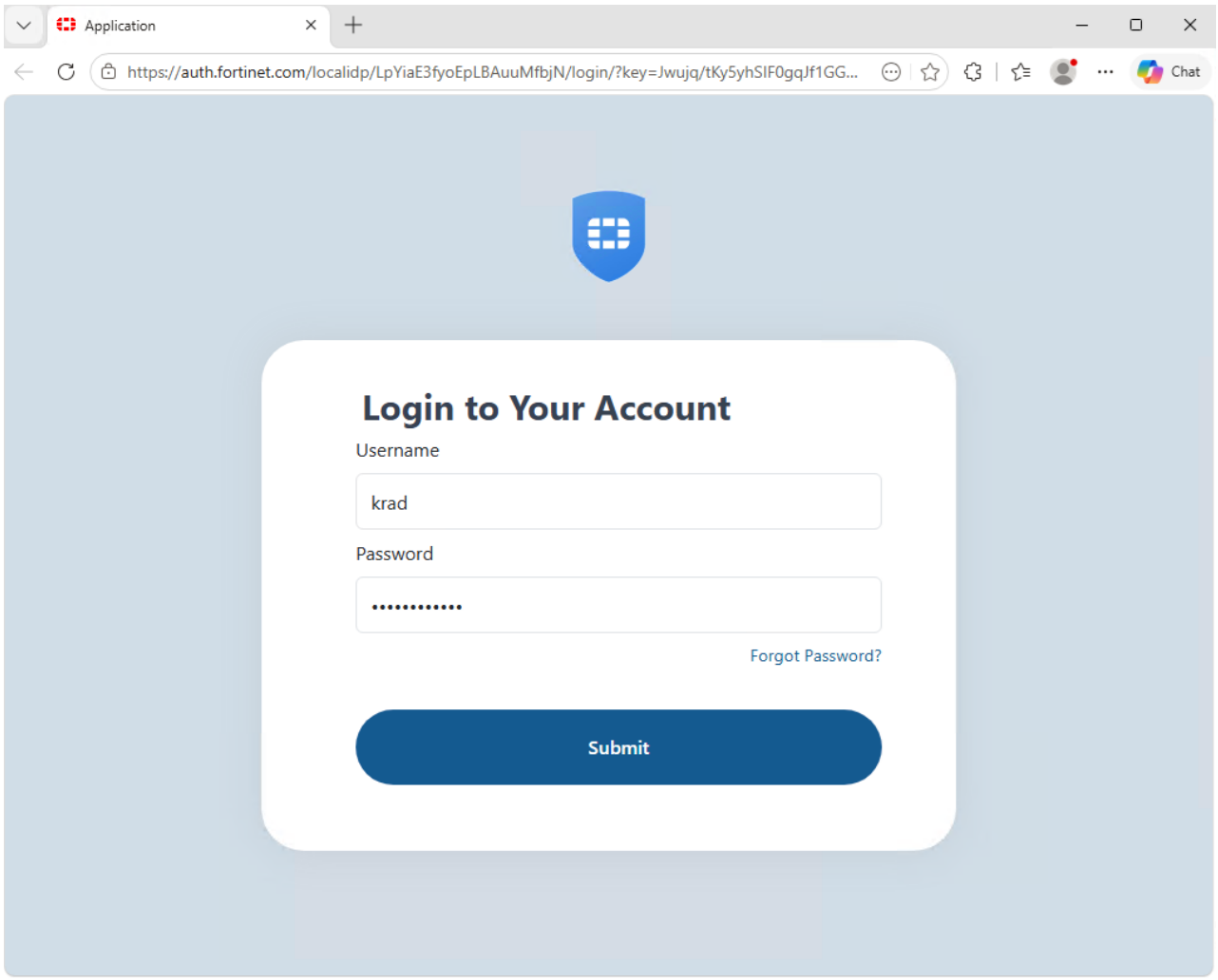
5. Click *Cancel* to exit the page.

To configure the VPN on FortiClient Standalone:

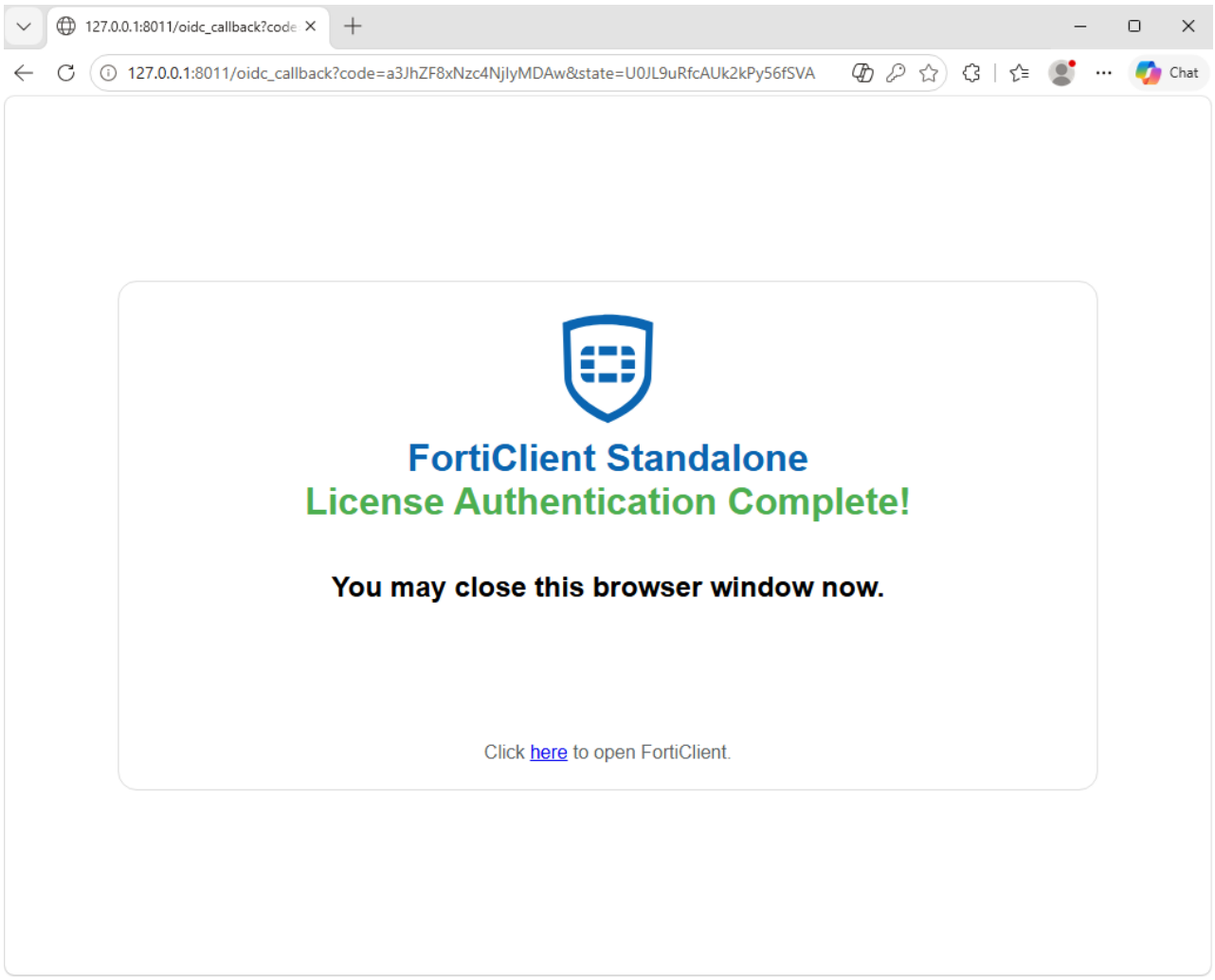
1. On the remote endpoint, open FortiClient Standalone.
2. Use the invitation code sent to the user email when the FIC account was created to login and activate FortiClient Standalone.



3. A SAML authentication window will open on a browser requiring the end user to authenticate to FortiIdentity Cloud. Enter the user credentials and password.



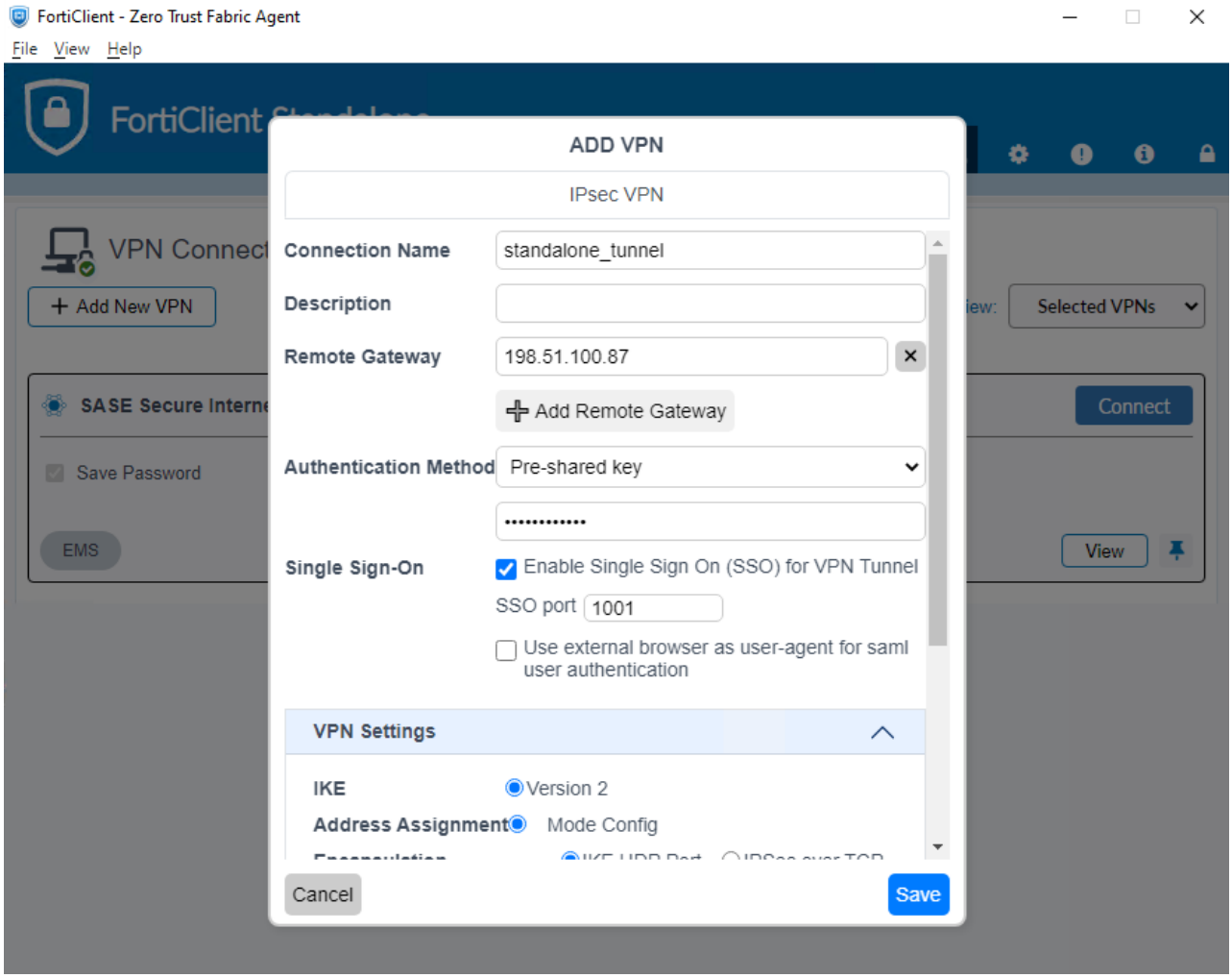
4. Once successful, a page will appear indicating the license is activated.



5. Go to the FortiClient Standalone Client, and click *Add New VPN*.
6. Configure the following:

Field	Value
Connection Name	standalone_tunnel
Remote Gateway	198.51.100.87
Authentication Method	Pre-shared key
Key	<enter key>
Single Sign-On	Click *Enable Single Sign On (SSO) for VPN Tunnel*
SSO port	1001
Use external browser as user-agent for saml user authentication	Enable if you want to use your browser to authenticate

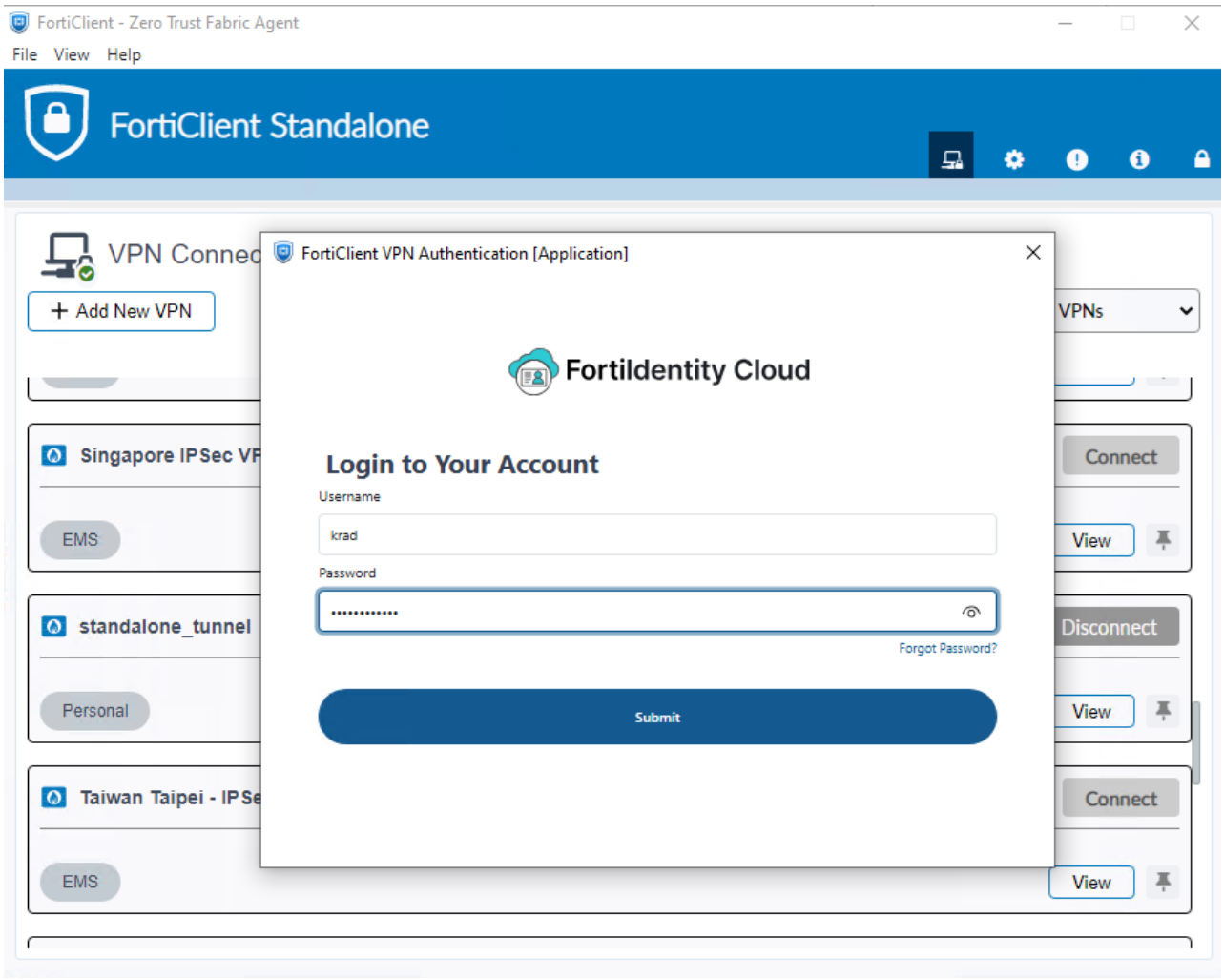
Leave the remaining settings as their default values.



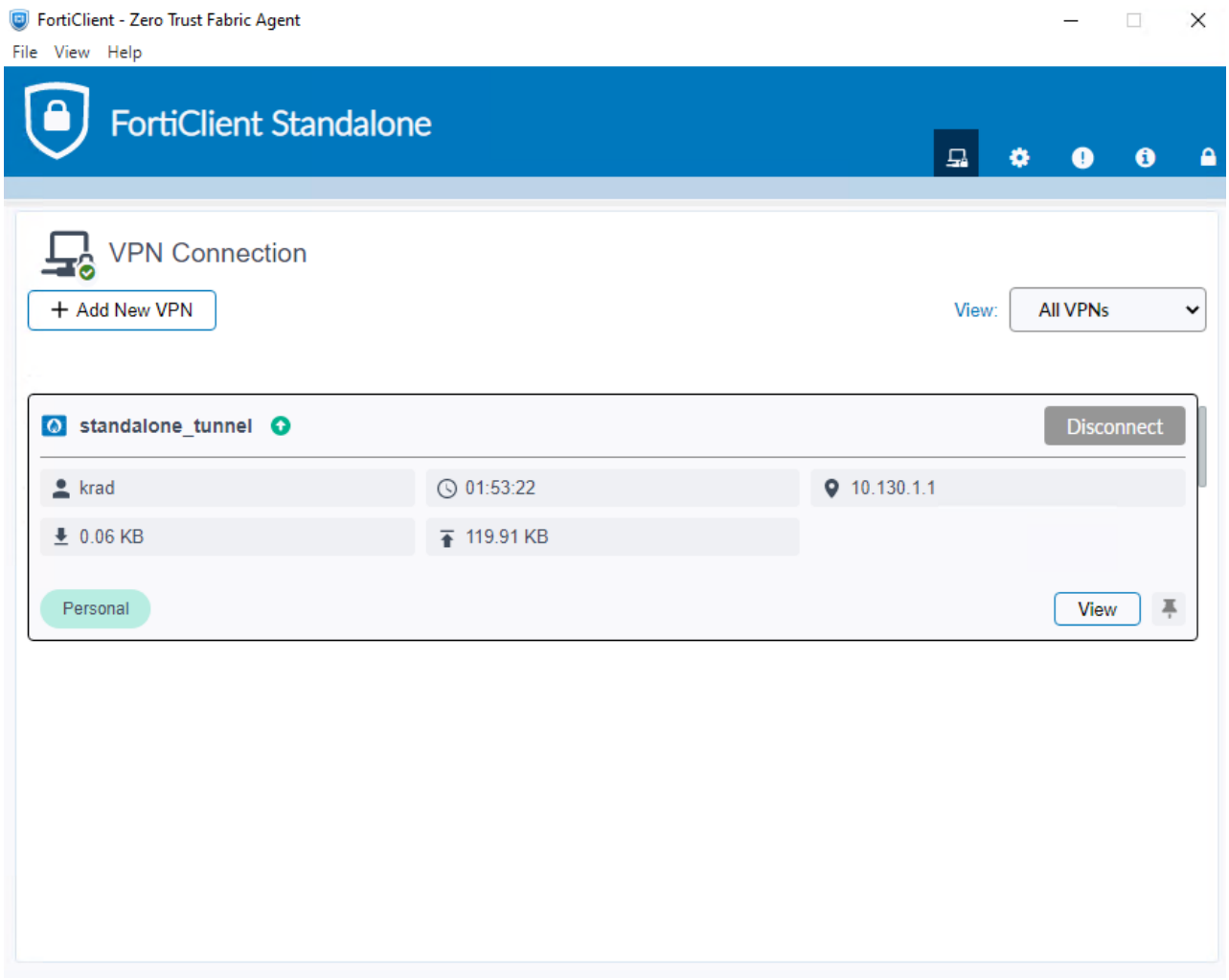
Testing and verification

To test the connection from the FortiClient:

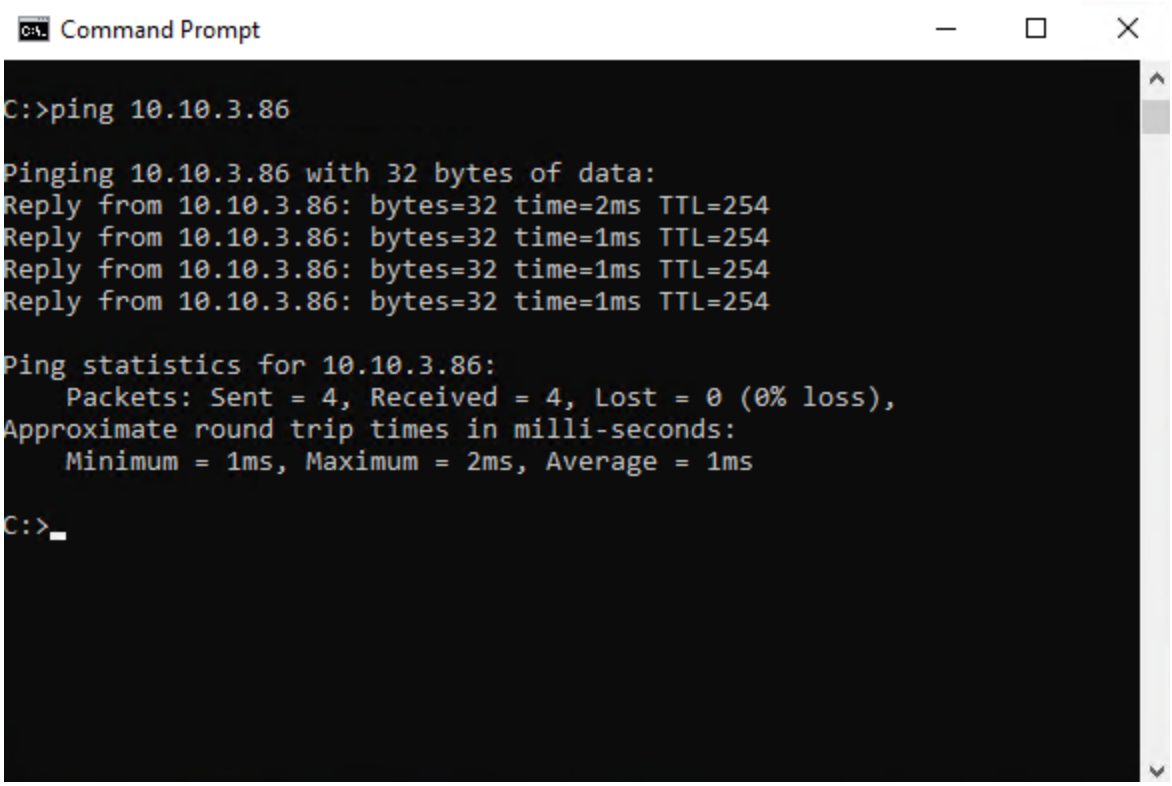
1. Find the new tunnel and click *Connect*.
2. When prompted, enter the FIC credentials.



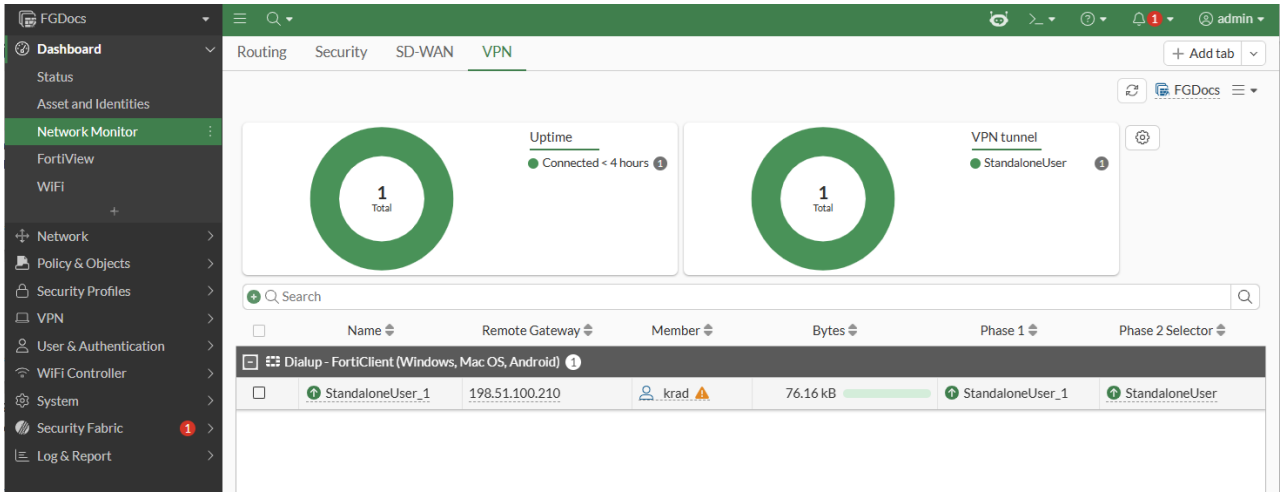
3. After passing 2-factor authentication, the user will be authenticated and the tunnel will come up.



4. In the CLI, try to ping a server behind the FortiGate:



5. Login to FortiIdentity Cloud.
6. Go to the *Monitor* > *IdP Sessions* page. The user that logged into VPN is displayed.
7. Click on Details to see more information about the User and session.
8. On the FortiGate, go to *Dashboard* > *Network Monitor* > *VPN* to view the user that is connected.



9. From the Command palette (CTRL+P) > CLI Diagnostics, search for *ike gateway* to list all VPN gateway tunnels.

```
vd: root/0
name: StandaloneUser_1
version: 2
interface: port2 6
```

```
addr: 198.51.100.87:4500 -> 198.51.100.210:58718
tun_id: 10.130.1.1/::10.0.0.1
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 7326s ago
eap-user: krad
2FA: no
peer-id: 198.51.100.210
peer-id-auth: no
FortiClient UID: 7B95C78E064144C2B69BF772A8BDF999
assigned IPv4 address: 10.130.1.1/255.255.255.255
nat: me peer
pending-queue: 0
PPK: no
IKE SA: created 1/1  established 1/1  time 1030/1030/1030 ms
IPsec SA: created 1/1  established 1/1  time 40/40/40 ms

  id/spi: 7 110117e334c433de/38a77cd3d2bfccab
  direction: responder
  status: established 7326-7325s ago = 1030ms
  proposal: aes128-sha256
  child: no
  SK_ei: eec35c4e030308d3-fa7dcd8befebf685
  SK_er: 263313c9ddfdecaa-6b6c4af04f25755d
  SK_ai: 8bf501d144851c9e-4e53f231ee00e426-2f8325e2cba85127-21b5e615baea764a
  SK_ar: 40c40ae07a71187a-78fb7e7c42e29101-f5a517d281467544-0c20b16413f715a6
  PPK: no
  message-id sent/recv: 0/372
  QKD: no
  PQC-KEM (IKE): no
  PQC-KEM (all IPsec): no
  lifetime/rekey: 86400/78804
  DPD sent/recv: 00000000/00000000
  peer-id: 198.51.100.210
```

10. For troubleshooting real-time IKE debugs, use the following commands and observe the output when a tunnel is negotiated and authenticated:

```
# diagnose debug enable

# diagnose debug application ike -1
Debug messages will be on for 30 minutes.

# diagnose debug application fnbamd -1
Debug messages will be on for 29 minutes.

# diagnose debug application samld -1
Debug messages will be on for 29 minutes.

...
<saml:NameID xmlns:saml="urn:oasis:names:tc:SAML:2.0:assertion"
```

```
Format="urn:oasis:names:tc:SAML:2.0:attrname-format:unspecified" SPNameQualifier=
"http://198.51.100.87:1001/remote/saml/metadata/">krad</saml:NameID>
</NidAndSessionIndex>
</Session>
samld_send_common_reply [134]:     Attr: 17, 31, magic=050c0d8097995b5e
samld_send_common_reply [134]:     Attr: 18, 29, 2026-05-12T18:30:49Z
samld_send_common_reply [131]:     Attr: 10, 22, 'username' 'krad'
samld_send_common_reply [131]:     Attr: 10, 25, 'first_name' 'Kenny'
samld_send_common_reply [131]:     Attr: 10, 22, 'last_name' 'Rad'
samld_send_common_reply [131]:     Attr: 10, 30, 'email' 'krad@xxxxx.com'
samld_send_common_reply [131]:     Attr: 10, 25, 'fic_tags' 'fct-vpn'
...
ike V=root:0: udp comes 198.51.100.210:58718->198.51.100.87:4500,ifindex=6,vrf=0,len=610....
ike V=root:0: IKEv2 exchange=SA_INIT id=110117e334c433de/0000000000000000 len=606
ike 0: in
110117E334C433DE0000000000000000212022080000000000000025E220001140200004401010007030000080300000
0C0300000C0100000C800E008003000008040000 1

4030000080200000503000008020000060300000802000007000000080200000202000044020100070300000803000
00C0300000C0100000C800E00800300000804000015030000 08

0200000503000008020000060300000802000007000000080200000202000044030100070300000803000000C030000
0C0100000C800E010003000008040000140300000802000005 0

3000008020000060300000802000007000000080200000200000044040100070300000803000000C0300000C0100000
C800E010003000008040000150300000802000005030000080 2

0000060300000802000007000000080200000228000068001400 00D7979154F89601E66A62A2E1BA98D026E97F75E5
759F1D5A80EC8A83BE94993C967AEAAFD3D2FDB6FA7B04E5C41

2DD1183F1C825A76A7036B83550355C975F49978101C8BED3474AA1BFEABDC01F2EF0C85C8A785BC7E9DE6A491DFF3
72609572B0000249215BB5D55C4A4538D90B1D5DC48A5E6B32C

F780E22B72FA2474E8C4388F299F2B00001C4C53427B6D465D1B337BB755A37A7FEF006C0054F77F00002B00001CB4
F01CA951E9DA8D0BAFBBD34AD3044E206C0054F77F000029000

01CC1DC4350476B98A429B91781914CA43E7097189DB702000029000008000 0402E2900001C0000400489E6FC10A83
44FE335C4980062904516C2A810C62900001C00004005AE8829
933D63E8A43390B613553733D4D705FAD50000000E0000402F000200030004
ike V=root:0:110117e334c433de/0000000000000000:7: responder received SA_INIT msg
ike V=root:0:110117e334c433de/0000000000000000:7: VID forticlient connect license
4C53427B6D465D1B337BB755A37A7FEF006C0054F77F0000
ike V=root:0:110117e334c433de/0000000000000000:7: VID Fortinet Endpoint Control
B4F01CA951E9DA8D0BAFBBD34AD3044E206C0054F77F0000
ike V=root:0:110117e334c433de/0000000000000000:7: VID Forticlient EAP Extension
C1DC4350476B98A429B91781914CA43E7097189DB7020000
ike V=root:0:110117e334c433de/0000000000000000:7: received notify type FRAGMENTATION_SUPPORTED
ike V=root:0:110117e334c433de/0000000000000000:7: received notify type NAT_DETECTION_SOURCE_IP
ike V=root:0:110117e334c433de/0000000000000000:7: received notify type NAT_DETECTION_
DESTINATION_IP
ike V=root:0:110117e334c433de/0000000000000000:7: received notify type SIGNATURE_HASH_
ALGORITHMS
```

```
ike V=root:0:110117e334c433de/0000000000000000:7: incoming proposal:
ike V=root:0:110117e334c433de/0000000000000000:7: proposal id = 1:
ike V=root:0:110117e334c433de/0000000000000000:7:   protocol = IKEv2:
ike V=root:0:110117e334c433de/0000000000000000:7:      encapsulation = IKEv2/none
ike V=root:0:110117e334c433de/0000000000000000:7:         type=ENCR, val=AES_CBC (key_len =
128)
ike V=root:0:110117e334c433de/0000000000000000:7:         type=INTEGR, val=AUTH_HMAC_SHA2_256_
128
ike V=root:0:110117e334c433de/0000000000000000:7:         type=PRF, val=PRF_HMAC_SHA2_256
ike V=root:0:110117e334c433de/0000000000000000:7:         type=PRF, val=PRF_HMAC_SHA2_384
ike V=root:0:110117e334c433de/0000000000000000:7:         type=PRF, val=PRF_HMAC_SHA2_512
ike V=root:0:110117e334c433de/0000000000000000:7:         type=PRF, val=PRF_HMAC_SHA
ike V=root:0:110117e334c433de/0000000000000000:7:         type=DH_GROUP, val=ECP384.
ike V=root:0:110117e334c433de/0000000000000000:7: proposal id = 2:
ike V=root:0:110117e334c433de/0000000000000000:7:   protocol = IKEv2:
ike V=root:0:110117e334c433de/0000000000000000:7:      encapsulation = IKEv2/none
ike V=root:0:110117e334c433de/0000000000000000:7:         type=ENCR, val=AES_CBC (key_len =
128)
ike V=root:0:110117e334c433de/0000000000000000:7:         type=INTEGR, val=AUTH_HMAC_SHA2_256_
128
ike V=root:0:110117e334c433de/0000000000000000:7:         type=PRF, val=PRF_HMAC_SHA2_256
ike V=root:0:110117e334c433de/0000000000000000:7:         type=PRF, val=PRF_HMAC_SHA2_384
ike V=root:0:110117e334c433de/0000000000000000:7:         type=PRF, val=PRF_HMAC_SHA2_512
ike V=root:0:110117e334c433de/0000000000000000:7:         type=PRF, val=PRF_HMAC_SHA
ike V=root:0:110117e334c433de/0000000000000000:7:         type=DH_GROUP, val=ECP521.
ike V=root:0:110117e334c433de/0000000000000000:7: proposal id = 3:
ike V=root:0:110117e334c433de/0000000000000000:7:   protocol = IKEv2:
ike V=root:0:110117e334c433de/0000000000000000:7:      encapsulation = IKEv2/none
ike V=root:0:110117e334c433de/0000000000000000:7:         type=ENCR, val=AES_CBC (key_len =
256)
ike V=root:0:110117e334c433de/0000000000000000:7:         type=INTEGR, val=AUTH_HMAC_SHA2_256_
128
ike V=root:0:110117e334c433de/0000000000000000:7:         type=PRF, val=PRF_HMAC_SHA2_256
ike V=root:0:110117e334c433de/0000000000000000:7:         type=PRF, val=PRF_HMAC_SHA2_384
ike V=root:0:110117e334c433de/0000000000000000:7:         type=PRF, val=PRF_HMAC_SHA2_512
ike V=root:0:110117e334c433de/0000000000000000:7:         type=PRF, val=PRF_HMAC_SHA
ike V=root:0:110117e334c433de/0000000000000000:7:         type=DH_GROUP, val=ECP384.
ike V=root:0:110117e334c433de/0000000000000000:7: proposal id = 4:
ike V=root:0:110117e334c433de/0000000000000000:7:   protocol = IKEv2:
ike V=root:0:110117e334c433de/0000000000000000:7:      encapsulation = IKEv2/none
ike V=root:0:110117e334c433de/0000000000000000:7:         type=ENCR, val=AES_CBC (key_len =
256)
ike V=root:0:110117e334c433de/0000000000000000:7:         type=INTEGR, val=AUTH_HMAC_SHA2_256_
128
ike V=root:0:110117e334c433de/0000000000000000:7:         type=PRF, val=PRF_HMAC_SHA2_256
ike V=root:0:110117e334c433de/0000000000000000:7:         type=PRF, val=PRF_HMAC_SHA2_384
ike V=root:0:110117e334c433de/0000000000000000:7:         type=PRF, val=PRF_HMAC_SHA2_512
ike V=root:0:110117e334c433de/0000000000000000:7:         type=PRF, val=PRF_HMAC_SHA
ike V=root:0:110117e334c433de/0000000000000000:7:         type=DH_GROUP, val=ECP521.
ike V=root:0: cache rebuild start
ike V=root:0:StandaloneUser: cached as dynamic
```

```
ike V=root:0:auto-nego: local:10.152.95.87, remote:10.11.23.45
ike V=root:0:auto-nego: cached as static-ddns.
ike V=root:0: cache rebuild done
ike V=root:0:110117e334c433de/0000000000000000:7: matched proposal id 1
ike V=root:0:110117e334c433de/0000000000000000:7: proposal id = 1:
ike V=root:0:110117e334c433de/0000000000000000:7:   protocol = IKEv2:
ike V=root:0:110117e334c433de/0000000000000000:7:      encapsulation = IKEv2/none
ike V=root:0:110117e334c433de/0000000000000000:7:         type=ENCR, val=AES_CBC (key_len =
128)
ike V=root:0:110117e334c433de/0000000000000000:7:         type=INTEGR, val=AUTH_HMAC_SHA2_256_
128
ike V=root:0:110117e334c433de/0000000000000000:7:         type=PRF, val=PRF_HMAC_SHA2_256
ike V=root:0:110117e334c433de/0000000000000000:7:         type=DH_GROUP, val=ECP384.
ike V=root:0:110117e334c433de/0000000000000000:7: lifetime=86400
ike V=root:0:110117e334c433de/0000000000000000:7: SA proposal chosen, matched gateway
StandaloneUser
ike V=root:0:StandaloneUser:StandaloneUser: created connection: 0x556befa429a0 6
198.51.100.87->198.51.100.210:58718.
...
ike V=root:0:StandaloneUser:7: responder received EAP msg
ike V=root:0:StandaloneUser:7: send EAP message to FNBAM
ike V=root:0:StandaloneUser:7: initiating EAP authentication
ike V=root:0:StandaloneUser: EAP user "7B95C78E064144C2B69BF772A8BDF999"
ike V=root:0:StandaloneUser: auth group StandaloneUser_user_groups
ike V=root:0:StandaloneUser: EAP 12708923928577 pending
...
[781] fnbam_user_auth_group_match-req id: 12708923928577, server: FortiIdentity Cloud, local
auth: 0, dn match: 0
[658] __group_match-Updating saml username from 7B95C78E064144C2B69BF772A8BDF999 to krad

[672] __group_match-Use 'krad' for user group matching.
[704] __group_match-Group 'StandaloneUser_user_groups' passed group matching
[707] __group_match-Add matched group 'StandaloneUser_user_groups'(3)
[269] find_matched_usr_grps-Passed group matching
[1012] update_auth_token_session-config does not require 2fa
[357] fnbamd_comm_send_result-Sending result 0 (nid 0) for req 12708923928577, len=2784
[2048] handle_child_rsp-Auth rsp 12708923928577, session created, line 15
ike V=root:0:StandaloneUser:7 EAP 12708923928577 result FNBAM_SUCCESS
ike V=root:0:StandaloneUser: EAP succeeded for user "7B95C78E064144C2B69BF772A8BDF999" group
"StandaloneUser_user_groups" 2FA=no
ike V=root:0:StandaloneUser:7: responder preparing EAP pass through message
...
ike V=root:0:StandaloneUser:7:StandaloneUser:0: comparing selectors
ike V=root:0:StandaloneUser:7:StandaloneUser:0: matched by rfc-rule-2
ike V=root:0:StandaloneUser:7:StandaloneUser:0: phase2 matched by subset
ike V=root:0:StandaloneUser:7:StandaloneUser:0: using mode-cfg override 0:10.130.1.1-
10.130.1.1:0
ike V=root:0:StandaloneUser:7:StandaloneUser:0: accepted proposal:
ike V=root:0:StandaloneUser:7:StandaloneUser:0: TSi_0 0:10.130.1.1-10.130.1.1:0
ike V=root:0:StandaloneUser:7:StandaloneUser:0: TSr_0 0:0.0.0.0-255.255.255.255:0
...
ike V=root:0:StandaloneUser_1:7: processed INITIAL-CONTACT
```

```
ike V=root:0:StandaloneUser_1:7: mode-cfg assigned (1) IPv4 address 10.130.1.1
ike V=root:0:StandaloneUser_1:7: mode-cfg assigned (2) IPv4 netmask 255.255.255.255
ike V=root:0:StandaloneUser_1:7: mode-cfg send (13) 0:10.10.3.0/255.255.255.0:0
ike V=root:0:StandaloneUser_1:7: mode-cfg send (3) IPv4 DNS(1) 96.45.45.45
ike V=root:0:StandaloneUser_1:7: mode-cfg send (3) IPv4 DNS(2) 96.45.46.46
ike V=root:0:StandaloneUser_1:7: mode-cfg send INTERNAL_IP6_SUBNET
ike V=root:0:StandaloneUser_1:7: mode-cfg IPv6 DNS ignored, no IPv6 DNS servers found
ike V=root:0:StandaloneUser_1:7: mode-cfg send (28673) UNITY_SAVE_PASSWD
ike V=root:0:StandaloneUser_1:7: client auto-negotiate is disabled
ike V=root:0:StandaloneUser_1:7: client-keep-alive is disabled
ike V=root:0:StandaloneUser_1:7:StandaloneUser:0: replay protection enabled
ike V=root:0:StandaloneUser_1:7:StandaloneUser:0: set sa life soft seconds=43186.
ike V=root:0:StandaloneUser_1:7:StandaloneUser:0: set sa life hard seconds=43200.
ike V=root:0:StandaloneUser_1:7:StandaloneUser:0: IPsec SA selectors #src=1 #dst=1
ike V=root:0:StandaloneUser_1:7:StandaloneUser:0: src 0 7 0:0.0.0.0-255.255.255.255:0
ike V=root:0:StandaloneUser_1:7:StandaloneUser:0: dst 0 7 0:10.130.1.1-10.130.1.1:0
ike V=root:0:StandaloneUser_1:7:StandaloneUser:0: add dynamic IPsec SA selectors 682
ike V=root:0:StandaloneUser_1:7:StandaloneUser:0: added dynamic IPsec SA proxyids new 1 682
ike V=root:0:StandaloneUser:0: add route 10.130.1.1/255.255.255.255 gw 10.130.1.1 oif
StandaloneUser(45) metric 15 priority 1
ike V=root:0:StandaloneUser_1:7:StandaloneUser:0: tunnel 1 of VDOM limit 0/0
ike V=root:0:StandaloneUser_1:7:StandaloneUser:0: add IPsec SA: SPIs=5aaaa081/07ee57c0
ike 0:StandaloneUser_1:7:StandaloneUser:0: IPsec SA dec spi 5aaaa081 key
16:60E1E90B6E808912FBD1CF183EADF8D3 auth 32:38632B9AFC5B5E4BB70D7166F83D
3366D9B976B0F97206B90DE9D003A3A5163B
ike 0:StandaloneUser_1:7:StandaloneUser:0: IPsec SA enc spi 07ee57c0 key
16:560B856E06BB458E90901285A6DA0163 auth 32:F7D180A7EA6FF227196964C28EA1
319B6ADA092126382F78D1422231728FE746
ike V=root:0:StandaloneUser_1:7:StandaloneUser:0: added IPsec SA: SPIs=5aaaa081/07ee57c0
ike V=root:0:StandaloneUser_1: scope=(3):(0), sync=1. Ignored
ike V=root:0:StandaloneUser_1:7:StandaloneUser:0: sending SNMP tunnel UP trap
ike V=root:0:StandaloneUser_1: tunnel up event assigned address 10.130.1.1
```

Native VPN as dialup client - Windows NEW

Native VPN remote access configurations have been improved on the VPN wizard. For supported OS's, configurations from the VPN wizard will work out of the box. Native VPN client defaults to using L2TP over IPsec for Windows, Android, and macOS/iOS clients. Administrators can also configure IKEv2 for Windows and Android clients. DH groups and proposals have been updated to work with native clients.

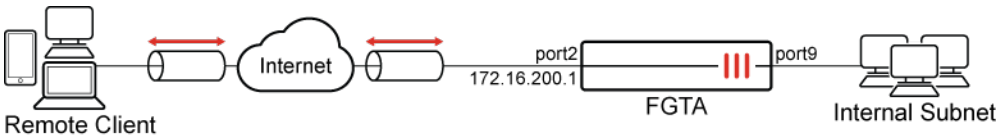
In L2TP over IPsec mode:

- Only pre-shared key is supported; signature cannot be chosen as Authentication method.
- L2TP over IPsec connects over IKEv1.
- Addressing mode must be set to Manual.
- Only one L2TP over IPsec tunnel can be configured per VDOM.

These limitations do not apply to IKEv2 mode.

Organizations that manage a mixed environment (Windows, macOS, iOS, Android) can use the wizard to deploy a VPN tunnel that automatically aligns with each OS's native VPN client defaults.

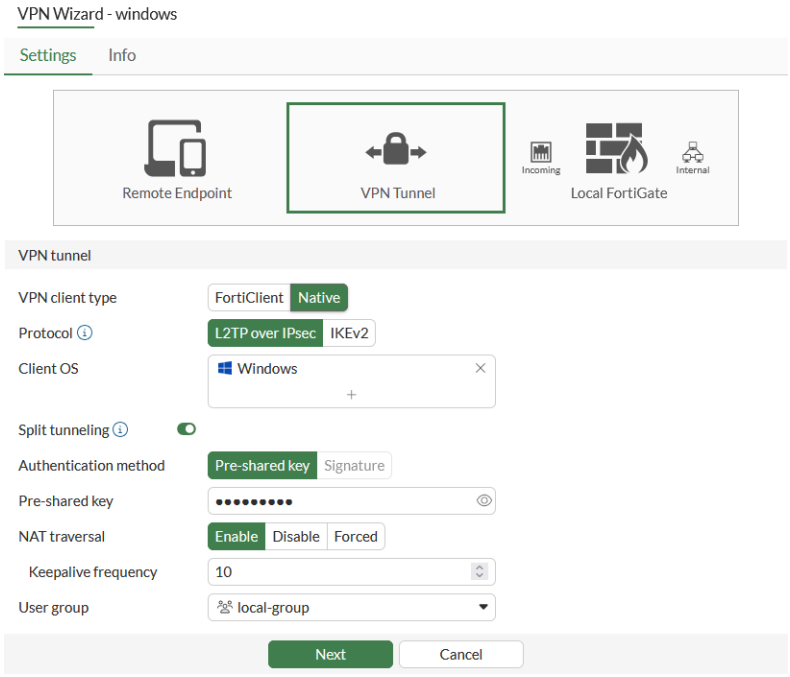
Example 1: Configure L2TP over IPsec tunnel for native VPN clients



In this example, the FortiGate acts as a VPN gateway to provide secure internet access (SIA) through the shared WAN on port2, and for remote access to the Internal network through port9. The settings are compatible for Windows, Android, and macOS/iOS clients connecting through L2TP over IPsec. However, connecting from a Windows client is demonstrated here. User authentication is based on pre-defined usergroup *local-group* that contains a local user called *userc*.

To configure L2TP IPsec tunnel from a template for Windows, Android, and macOS:

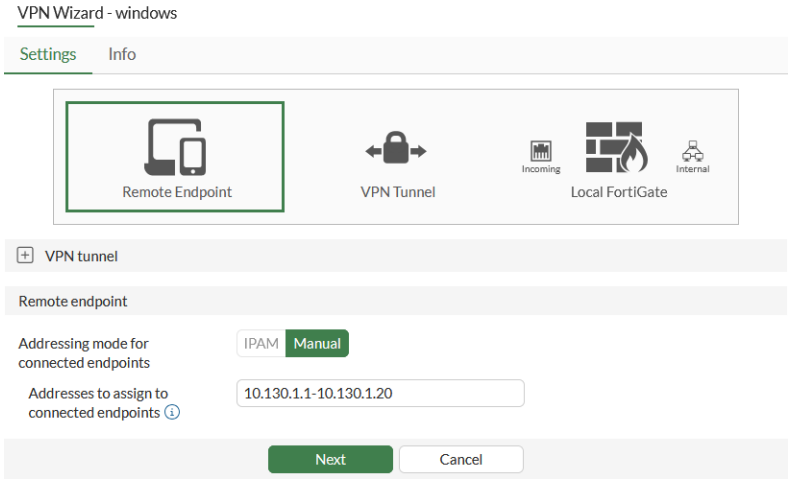
1. Go to *VPN > VPN Wizard*.
2. Name the VPN tunnel, select *Remote Access* template, and click *Begin*.
3. Configure the VPN tunnel settings, then click *Next*:



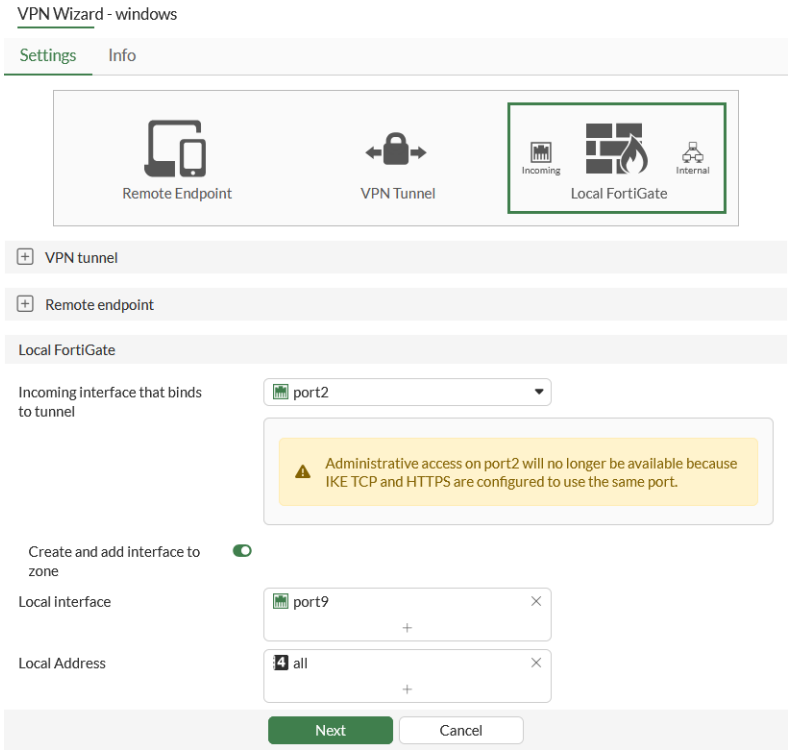
Parameter	Value
VPN client type	Native
Protocol	L2TP over IPsec
Client OS	Windows
Authentication method	Pre-Shared key
Pre-Shared key	Enter the pre-shared key

Parameter	Value
User group	local-group

4. Set *Addresses to assign to connected endpoints* to *10.130.1.1-10.130.1.20*, then click *Next*.



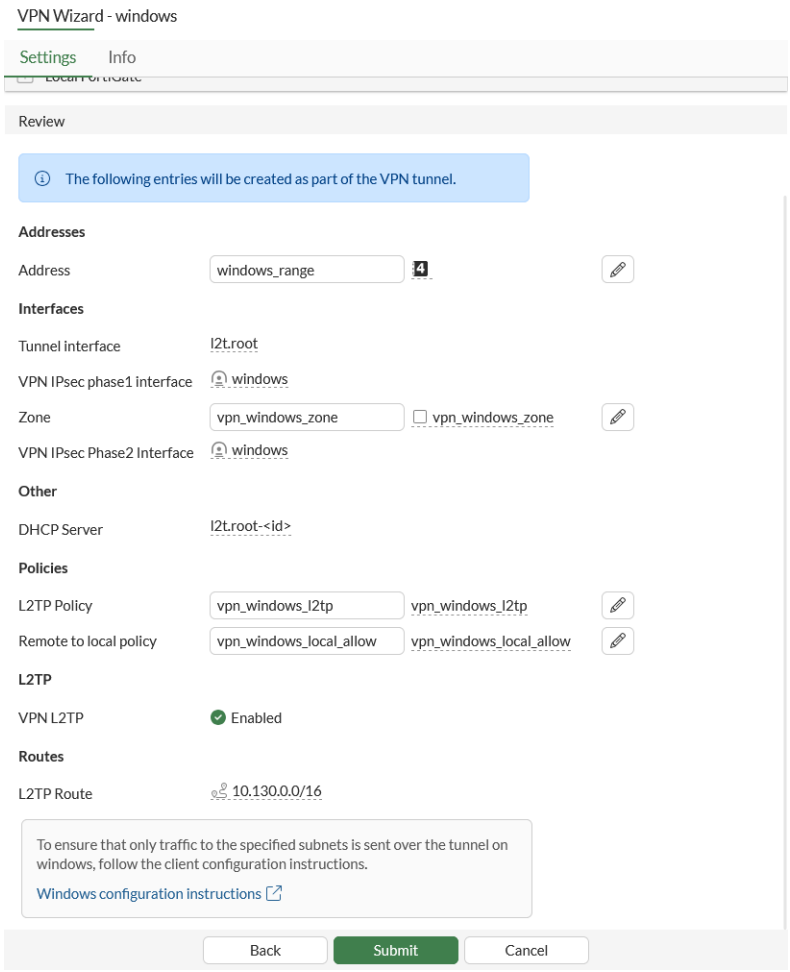
5. Configure the local FortiGate options, then click *Next*:



Parameter	Value
Incoming interface that binds to tunnel	port2

Parameter	Value
Local interface	port9
Local Address	all

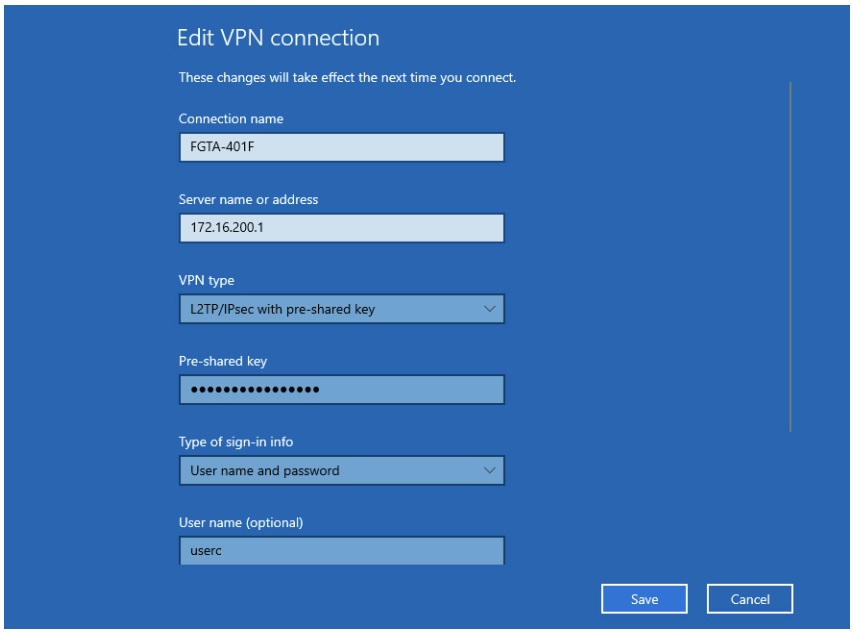
6. Review the configuration, then click *Submit*.



Only Android 12 and below are supported for L2TP VPN.

To configure L2TP connection for Windows:

1. Open *VPN* or go to *Settings* > *Network & internet*.
2. Add and configure a VPN:



Parameter	Value
VPN provider	Windows (built-in)
Connection name	Enter a connection name.
Server name or address	Enter your FortiGate Firewall's hostname or IP address.
VPN type	L2TP/IPsec with pre-shared key
Pre-shared key	Enter the pre-shared key.
Type of sign-in info	Username and password
Username (optional)	Enter the username.
Password (optional)	Enter the password.

3. Click *Save*.

To configure L2TP IPsec tunnel in the CLI:

```
config user group
    edit "local-group"
        set member "usera" "userc"
    next
end
```

```
config vpn ipsec phase1-interface
    edit "windows"
        set type dynamic
        set interface "port1"
        set ike-version 1
        set peertype any
```

```
        set net-device disable
        set proposal aes256-sha512 aes256-sha256 aes256-sha1
        set dhgrp 19 14 2
        set wizard-type dialup-l2tp
        set psksecret <password>
    next
end
```

```
config vpn ipsec phase2-interface
    edit "windows"
        set phase1name "windows"
        set proposal aes256-sha512 aes256-sha256 aes256-sha1
        set pfs disable
        set keepalive enable
        set encapsulation transport-mode
        set l2tp enable
    next
end
```

```
config vpn l2tp
    set status enable
    set eip 10.130.1.20
    set sip 10.130.1.1
    set enforce-ipsec enable
    set usrgrp "local-group"
end
```

```
config firewall address
    edit "windows_range"
        set type iprange
        set start-ip 10.130.1.1
        set end-ip 10.130.1.20
    next
end
```

Three policies are created:

- One to allow L2TP traffic into the WAN interface
- One to allow VPN traffic to the local subnet
- One to allow VPN traffic to the Internet

```
config firewall policy
    edit 3
        set name "vpn_windows_l2tp"
        set srcintf "windows"
        set dstintf "port2"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "L2TP"
    next
```

```
    edit 4
        set name "vpn_windows_local_allow"
        set srcintf "l2t.root"
        set dstintf "port9"
        set action accept
        set srcaddr "windows_range"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
    edit 5
        set name "vpn_windows_utm_allow"
        set srcintf "l2t.root"
        set dstintf "port2"
        set action accept
        set srcaddr "windows_range"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

```
config router static
    edit 2
        set dst 10.130.0.0 255.255.0.0
        set device "l2t.root"
    next
end
```

To check the IPsec phase1/phase2 interface status:

```
# diagnose vpn ike gateway list

vd: root/0
name: windows_0
version: 1
interface: port1 7
addr: 172.16.200.1:4500 -> 172.16.200.254:64916
tun_id: 172.16.200.254/::10.0.0.15
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 19s ago
peer-id: 31.1.1.198
peer-id-auth: no
nat: me peer
pending-queue: 0
IKE SA: created 1/1  established 1/1  time 20/20/20 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms
```

```
  id/spi: 16 ccfbf5c901e3899f/547217d62b9bc1ef
  direction: responder
  status: established 19-19s ago = 20ms
  proposal: aes256-sha1
  key: cd7c7a2385ef9085-30a95bab3c1a6178-4dea8ede135046e0-fbb2c44074b6fd74
  QKD: no
  PQC-KEM (IKE): no
  PQC-KEM (all IPsec): no
  lifetime/rekey: 28800/28510
  DPD sent/recv: 00000000/00000000
  peer-id: 31.1.1.198
```

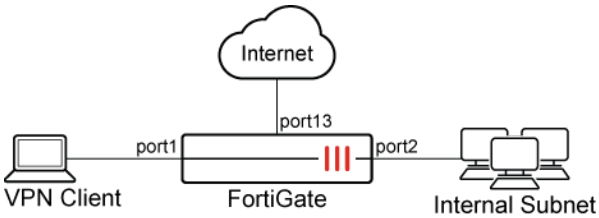
```
# diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=windows_0 ver=1 serial=10 172.16.200.1:4500->172.16.200.254:64916 nexthop=0.0.0.0 tun_
id=172.16.200.254 tun_id6=::10.0.0.15 status=up dst_mtu=1500 weight=1 country=ZZ
bound_if=7 real_if=7 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none options[0x23a8]=npu rgwy-
chg rport-chg frag-rfc  run_state=0 role=primary accept_traffic=1 overlay_id=0

parent=windows index=0
proxyid_num=1 child_num=0 refcnt=6 ilast=0 olast=42955707 ad=/0
stat: rxp=6 txp=0 rxb=696 txb=0
dpd: mode=on-demand on=1 status=ok idle=60000ms retry=3 count=0 seqno=0
natt: mode=keepalive draft=32 interval=10 remote_port=64916
fec: egress=0 ingress=0
proxyid=windows proto=17 sa=1 ref=2 serial=1 transport-mode add-route
  src: 17:172.16.200.1-172.16.200.1:1701
  dst: 17:172.16.200.254-172.16.200.254:0
  SA:  ref=3 options=1a7 type=00 soft=0 mtu=1454 expire=3563/0B replaywin=2048
       seqno=401 esn=0 replaywin_lastseq=00000006 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=3589/3600
  dec: spi=a91127c2 esp=aes key=32
e54c49f9205f06ff55e9e974eefa00079bd57466041cf0a0512a11193a533a36
       ah=sha1 key=20 3eca7a57dda0ae2dbdf589f2798a2539b3ca774b
  enc: spi=c8be2f10 esp=aes key=32
008379628c4d67a232fb43dd2b7606bb43c8e02a731df7b90833485e2bb9f522
       ah=sha1 key=20 0291e163d22963ebe01bc8cc075cfabef935c10d
  dec:pkts/bytes=6/696, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=0.0.0.0:0 npu_lgwy=0.0.0.0:0npu_selid=e
  dec_npuid=0 enc_npuid=0 dec_engid=-1 enc_engid=-1 dec_saidx=-1 enc_saidx=-1
------------------------------------------------------
name=windows ver=1 serial=2 172.16.200.1:0->0.0.0.0:0 nexthop=0.0.0.0 tun_id=10.0.0.1 tun_
id6=::10.0.0.1 status=up dst_mtu=0 weight=1
bound_if=7 real_if=0 lgwy=static/1 tun=intf mode=dialup/2 encap=none options[0x228]=npu frag-rfc
role=primary accept_traffic=1 overlay_id=0

proxyid_num=0 child_num=1 refcnt=3 ilast=42955707 olast=42955707 ad=/0
stat: rxp=30 txp=0 rxb=3480 txb=0
dpd: mode=on-demand on=0 status=ok idle=20000ms retry=3 count=0 seqno=0
```

```
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
```

```
# diagnose debug enable
# diagnose debug app ike -1
# diagnose debug app fnbam -1
ike V=root:0:ccfbf5c901e3899f/0000000000000000:16: responder: main mode get 1st message...
...
ike V=root:0:windows: adding new dynamic tunnel for 172.16.200.254:64916
ike V=root:0:windows_0: tunnel created tun_id 172.16.200.254/::10.0.0.15 remote_location 0.0.0.0
ike V=root:0:windows_0: added new dynamic tunnel for 172.16.200.254:64916
ike V=root:0:windows_0:16: established IKE SA ccfbf5c901e3899f/547217d62b9bc1ef
ike V=root:0:windows_0: DPD disabled, not negotiated
ike V=root:0:windows_0:16: check peer route: if_addr4_rcvd=0, if_addr6_rcvd=0, mode_cfg=0
ike V=root:0:windows_0:16: no pending Quick-Mode negotiations
ike V=root:0: udp comes 172.16.200.254:64916->172.16.200.1:4500,ifindex=7,vrf=0,len=448....
ike V=root:0: IKEv1 exchange=Quick id=ccfbf5c901e3899f/547217d62b9bc1ef:00000001 len=444 vrf=0
...
ike V=root:0:windows_0:16:61: received NATOA-i 31.1.1.198
ike V=root:0:windows_0:16:61: received NATOA-r 172.18.68.206
ike V=root:0:windows_0:16:61: peer proposal is: peer:17:31.1.1.198-31.1.1.198:1701,
me:17:172.18.68.206-172.18.68.206:1701
ike V=root:0:windows_0:16:windows:61: trying
ike V=root:0:windows_0:16:61: L2TP selectors 17:172.18.68.206-172.18.68.206:1701 -> 17:31.1.1.198-
31.1.1.198:1701
ike V=root:0:windows_0:16:61: transport mode, override with 17:172.16.200.1-172.16.200.1:1701 ->
17:172.16.200.254-172.16.200.254:0
ike V=root:0:windows_0:16:windows:61: matched phase2
ike V=root:0:windows_0:16:windows:61: dynamic client
ike V=root:0:windows_0:16:windows:61: my proposal:
ike V=root:0:windows_0:16:windows:61: proposal id = 1:
ike V=root:0:windows_0:16:windows:61:   protocol id = IPSEC_ESP:
ike V=root:0:windows_0:16:windows:61:      trans_id = ESP_AES_CBC (key_len = 256)
ike V=root:0:windows_0:16:windows:61:      encapsulation = ENCAPSULATION_MODE_TRANSPORT
ike V=root:0:windows_0:16:windows:61:         type = AUTH_ALG, val=SHA2_512
ike V=root:0:windows_0:16:windows:61:      trans_id = ESP_AES_CBC (key_len = 256)
ike V=root:0:windows_0:16:windows:61:      encapsulation = ENCAPSULATION_MODE_TRANSPORT
ike V=root:0:windows_0:16:windows:61:         type = AUTH_ALG, val=SHA2_256
ike V=root:0:windows_0:16:windows:61:      trans_id = ESP_AES_CBC (key_len = 256)
ike V=root:0:windows_0:16:windows:61:      encapsulation = ENCAPSULATION_MODE_TRANSPORT
ike V=root:0:windows_0:16:windows:61:         type = AUTH_ALG, val=SHA1
ike V=root:0:windows_0:16:windows:61: incoming proposal:
ike V=root:0:windows_0:16:windows:61: proposal id = 1:
ike V=root:0:windows_0:16:windows:61:   protocol id = IPSEC_ESP:
ike V=root:0:windows_0:16:windows:61:      trans_id = ESP_AES_CBC (key_len = 256)
ike V=root:0:windows_0:16:windows:61:      encapsulation = UDP_ENCAPSULATION_MODE_TRANSPORT_
RFC3947
ike V=root:0:windows_0:16:windows:61:         type = AUTH_ALG, val=SHA1
ike V=root:0:windows_0:16:windows:61: negotiation result
ike V=root:0:windows_0:16:windows:61: proposal id = 1:
ike V=root:0:windows_0:16:windows:61:   protocol id = IPSEC_ESP:
ike V=root:0:windows_0:16:windows:61:      trans_id = ESP_AES_CBC (key_len = 256)
```

```
ike V=root:0:windows_0:16:windows:61:      encapsulation = ENCAPSULATION_MODE_TRANSPORT
ike V=root:0:windows_0:16:windows:61:         type = AUTH_ALG, val=SHA1
ike V=root:0:windows_0:16:windows:61: using udp transport mode.
ike V=root:0:windows_0:16:61: sending NATOA-i 172.16.200.254
ike V=root:0:windows_0:16:61: sending NATOA-r 172.16.200.1
ike V=root:0:windows_0:16:windows:61: replay protection enabled
ike V=root:0:windows_0:16:windows:61: SA life soft seconds=3589.
ike V=root:0:windows_0:16:windows:61: SA life hard seconds=3600.
ike V=root:0:windows_0:16:windows:61: IPsec SA selectors #src=1 #dst=1
ike V=root:0:windows_0:16:windows:61: src 0 7 17:172.16.200.1-172.16.200.1:1701
ike V=root:0:windows_0:16:windows:61: dst 0 7 17:172.16.200.254-172.16.200.254:0
ike V=root:0:windows_0:16:windows:61: add dynamic IPsec SA selectors 183
ike V=root:0:windows_0:16:windows:61: added dynamic IPsec SA proxyids new 1 183
ike V=root:0:windows:61: add route 172.16.200.254/255.255.255.255 gw 172.16.200.254 oif windows
(51) metric 15 priority 1
ike V=root:0:windows_0:16:windows:61: tunnel 1 of VDOM limit 0/0
ike V=root:0:windows_0:16:windows:61: add IPsec SA: SPIs=a91127c2/c8be2f10
ike 0:windows_0:16:windows:61: IPsec SA dec spi a91127c2 key
32:E54C49F9205F06FF55E9E974EEFA00079BD57466041CF0A0512A11193A533A36 auth
20:3ECA7A57DDA0AE2DBDF589F2798A2539B3CA774B
ike 0:windows_0:16:windows:61: IPsec SA enc spi c8be2f10 key
32:008379628C4D67A232FB43DD2B7606BB43C8E02A731DF7B90833485E2BB9F522 auth
20:0291E163D22963EBE01BC8CC075CFABEF935C10D
ike V=root:0:windows_0:16:windows:61: transport mode encapsulation is enabled
ike V=root:0:windows_0:16:windows:61: added IPsec SA: SPIs=a91127c2/c8be2f10
ike V=root:0:windows_0:16:windows:61: sending SNMP tunnel UP trap
ike V=root:0:windows_0: tunnel up event
...
```

Example 2: Configure IKEv2 IPsec tunnel for native VPN clients



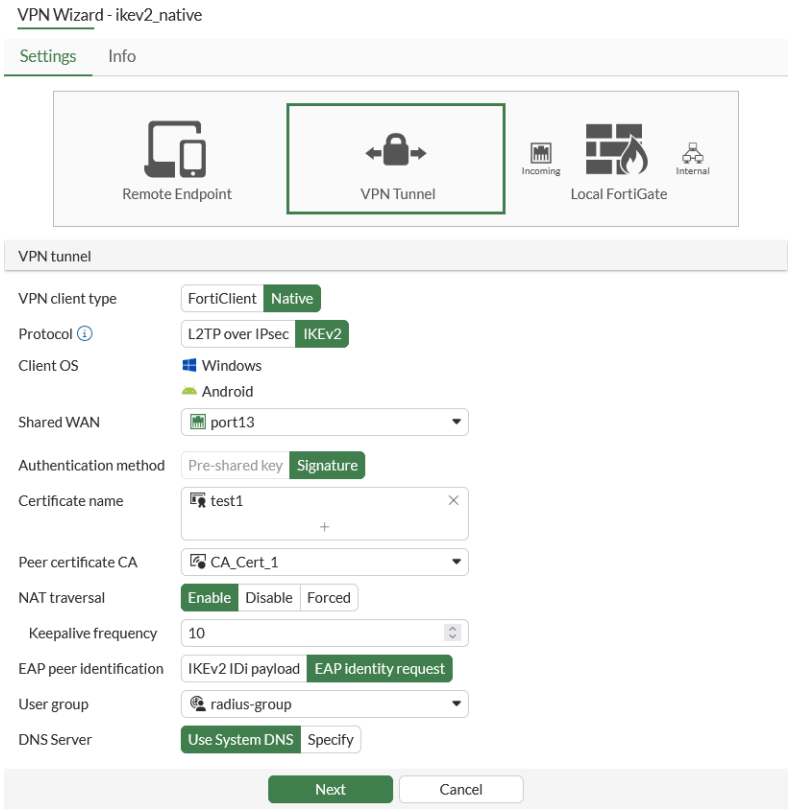
In this example, the FortiGate acts as a VPN gateway to provide secure internet access (SIA) through the shared WAN on port13, and for remote access to the Internal network through port2. The settings are compatible for Windows and Android clients only connecting through IKEv2. However, connecting from a Windows client is demonstrated here. Signature authentication is used, and user authentication is based on RADIUS authentication. Alternatively, local user authentication can also be used.

Signature authentication must be used, which means an appropriate server certificate signed by a CA should be used. The CA certificate also needs to be installed on the endpoints to verify the server certificate. Furthermore, the certificate requires a proper SAN definition. On the client configuration, the server IP or FQDN must match the SAN.

This example assumes certificates and RADIUS server have already been configured and installed. For detailed information about certificate and RADIUS configurations, see Windows IKEv2 native VPN with user certificate on page 3070.

To configure IKEv2 IPsec tunnel from a template for Windows:

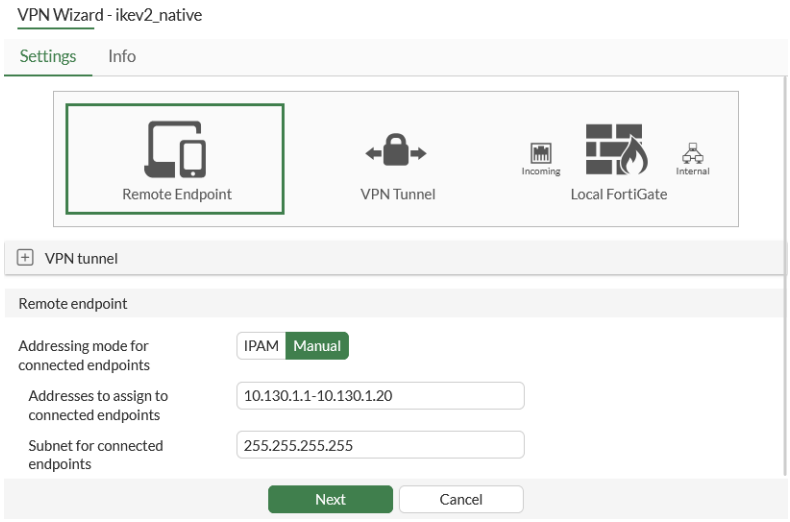
1. Go to *VPN > VPN Wizard*.
2. Name the VPN tunnel, select *Remote Access* template, and click *Begin*.
3. Configure the VPN tunnel settings, then click *Next*:



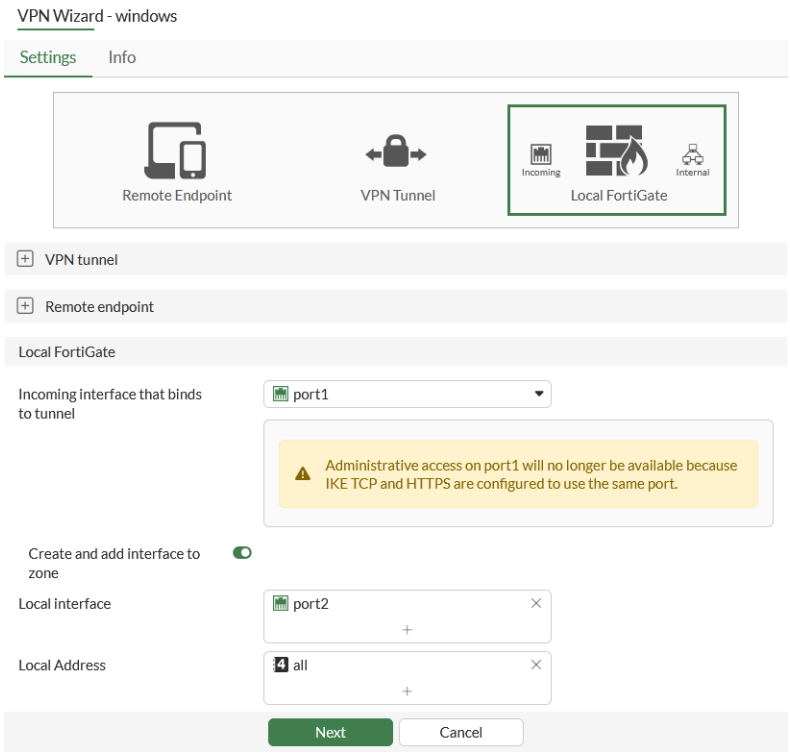
Parameter	Value
VPN client type	Native
Protocol	IKEv2
Shared WAN	port13
Authentication method	Signature
Certificate names	test1 The certificate should have a proper SAN defined with either the IP or FQDN of the VPN gateway.
Peer Certificate CA	CA_Cert_1 CA certificate to verify the client's user certificate. Must be configured but not applicable in this scenario.

Parameter	Value
User group	radius-group Alternatively, you can also use a local user and group here.

4. Set *Addressing mode for connected endpoints* to *Manual* and set *Addresses to assign to connected endpoints* to *10.130.1.1-10.130.1.20*, then click *Next*.

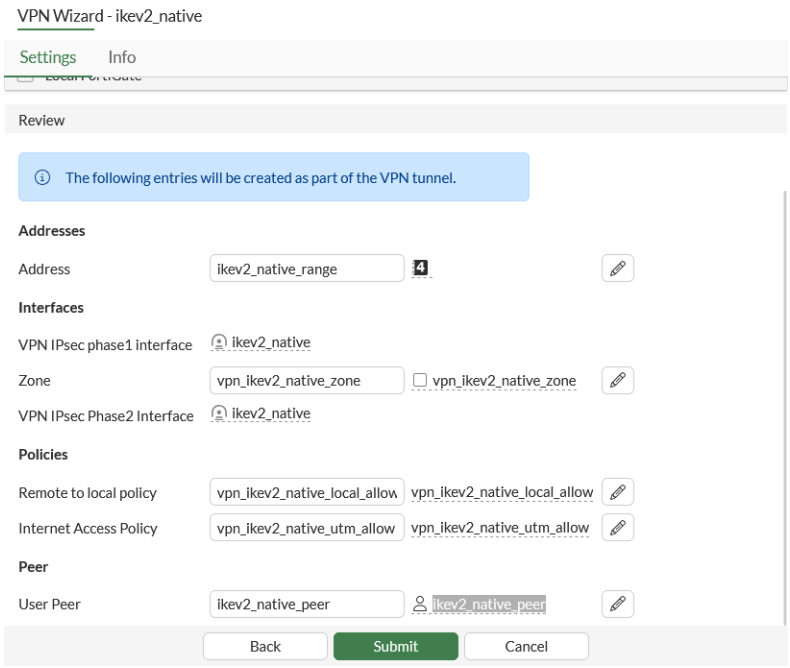


5. Configure the local FortiGate options, then click *Next*:



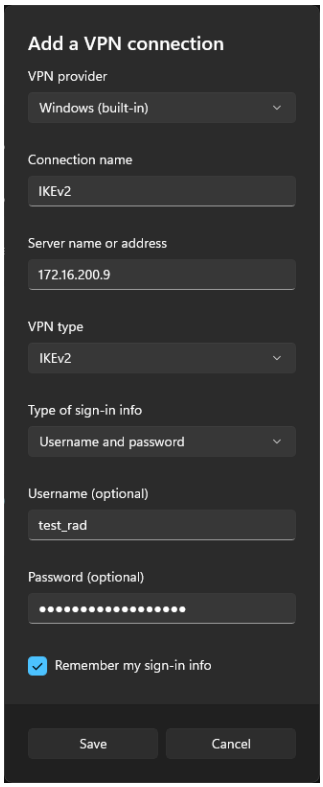
Parameter	Value
Incoming interface that binds to tunnel	port1
Local interface	port2
Local Address	all

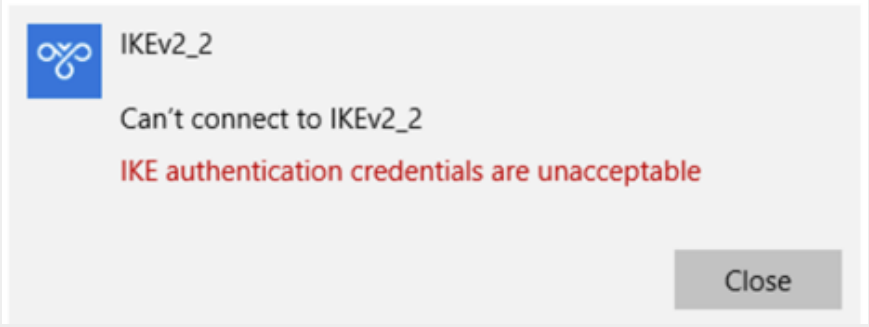
6. Review the configuration, then click *Submit*.



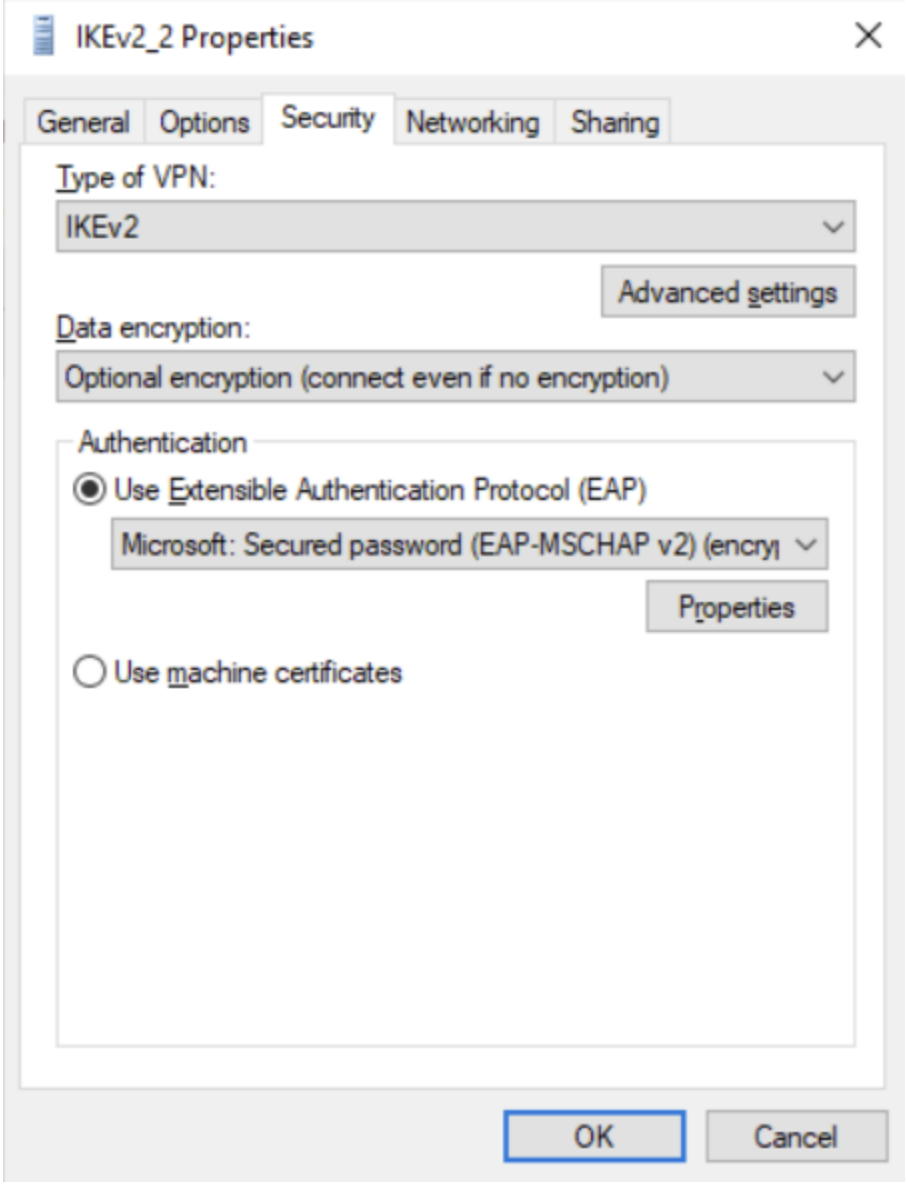
To configure IKEv2 VPN connection for Windows:

1. Open *VPN* or go to *Settings* > *Network & internet*.
2. Add and configure a VPN:



Parameter	Value
VPN provider	Windows (built-in)
Connection name	Enter a connection name.
Server name or address	Enter your VPN Gateway's FQDN or IP address. Windows uses the server name or address to verify the SAN field. Therefore this must match. In case it doesn't match, the connection will fail with an error: 
VPN type	IKEv2
Type of sign-in info	Username and password
Username (optional)	Enter the username.
Password (optional)	Enter the password.

3. Click *Save*.
   Windows VPN can support many types of EAP methods. When username and password is used, it defaults to EAP-MSCHAP v2. This can be seen when you edit the properties of the VPN network adapter, and view the *Security tab*.



To configure IKEv2 IPsec tunnel for Windows in the CLI:

```
config vpn ipsec phase1-interface
    edit "ikev2_native"
        set type dynamic
        set interface "port1"
        set ike-version 2
        set authmethod signature
        set net-device disable
```

```
        set mode-cfg enable
        set proposal aes256-sha256 aes256gcm-prfsha256 aes256gcm-prfsha384
        set dhgrp 20 14 2
        set eap enable
        set eap-identity send-request
        set wizard-type dialup-ikev2
        set authusrgrp "radius-group"
        set certificate "test1"
        set peer "ikev2_native_peer"
        set assign-ip-from name
        set dns-mode auto
        set ipv4-name "ikev2_native_range"
    next
end
```

```
config vpn ipsec phase2-interface
    edit "ikev2_native"
        set phase1name "ikev2_native"
        set proposal aes256gcm aes256-sha1 aes128-sha1
        set pfs disable
        set keepalive enable
    next
end
```

```
config firewall policy
    edit 1
        set name "vpn_ikev2_native_local_allow"
        set uuid 402dbbe2-0de4-51f1-202f-ea123f6da629
        set srcintf "ikev2_native"
        set dstintf "port2"
        set action accept
        set srcaddr "ikev2_native_range"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
    edit 2
        set name "vpn_ikev2_native_utm_allow"
        set uuid 402fbab4-0de4-51f1-97c2-2e4456a64492
        set srcintf "ikev2_native"
        set dstintf "port13"
        set action accept
        set srcaddr "ikev2_native_range"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

```
config firewall address
    edit "ikev2_native_range"
        set uuid 402b4d9e-0de4-51f1-c355-1375e255836b
        set type iprange
        set start-ip 10.130.1.1
        set end-ip 10.130.1.20
    next
end
```

```
config router static
    edit 1
        set gateway 172.16.200.254
        set device "port1"
    next
end
```

To check the IPsec phase1/phase2 interface status:

```
# diagnose vpn ike gateway list

vd: root/0
name: ikev2_native_0
version: 2
interface: port1 7
addr: 172.16.200.4:4500 -> 172.16.200.254:64917
tun_id: 10.130.1.1/::10.0.0.5
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 9s ago
eap-user: test_rad
2FA: no
assigned IPv4 address: 10.130.1.1/0.0.0.0
nat: me peer
pending-queue: 0
PPK: no
IKE SA: created 1/1  established 1/1  time 230/230/230 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 89 bc5aa3bb99c0963c/290cf660b0977bf5
  direction: responder
  status: established 9-8s ago = 230ms
  proposal: aes256-sha256
  child: no
  SK_ei: 1f769b39b6e45680-bc358b9bbd82e462-5f0843ec67ac6a5a-56ef87e3cd629067
  SK_er: 094a6134c1e9e472-8730eb86ceb92a04-292ea04fa3514643-1ad51077fc429420
  SK_ai: 1cb620c46998be6e-aaa2136a04ccae9a-d478963782112a87-c35806662125531b
  SK_ar: 9d3c9205a68e9efb-3d95fce5b3bbb81a-b2642e584c509f6c-08be1b3cd27a2a02
  PPK: no
  message-id sent/recv: 0/7
  QKD: no
```

```
  PQC-KEM (IKE): no
  PQC-KEM (all IPsec): no
  lifetime/rekey: 86400/86121
  DPD sent/recv: 00000000/00000000
```

```
# diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=ikev2_native ver=2 serial=4 172.16.200.4:0->0.0.0.0:0 nexthop= tun_id=10.0.0.2 tun_
id6=::10.0.0.2 status=up dst_mtu=0 weight=1
bound_if=7 real_if=0 lgwy=static/1 tun=intf mode=dialup/2 encap=none options[0x228]=npu frag-rfc
role=primary accept_traffic=1 overlay_id=0

proxyid_num=0 child_num=1 refcnt=3 ilast=42973580 olast=42973580 ad=/0
stat: rxp=203 txp=0 rxb=31104 txb=0
dpd: mode=on-demand on=0 status=ok idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
------------------------------------------------------
name=ikev2_native_0 ver=2 serial=6 172.16.200.4:4500->172.16.200.254:64917 nexthop= tun_
id=10.130.1.1 tun_id6=::10.0.0.5 status=up dst_mtu=0 weight=1 country=ZZ
bound_if=7 real_if=0 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none options[0x23a8]=npu rgwy-
chg rport-chg frag-rfc  run_state=0 role=primary accept_traffic=1 overlay_id=0

parent=ikev2_native index=0
proxyid_num=1 child_num=0 refcnt=6 ilast=0 olast=42973580 ad=/0
stat: rxp=72 txp=0 rxb=11248 txb=0
dpd: mode=on-demand on=1 status=ok idle=20000ms retry=3 count=0 seqno=0
natt: mode=keepalive draft=0 interval=10 remote_port=64917
fec: egress=0 ingress=0
proxyid=ikev2_native proto=0 sa=1 ref=2 serial=1 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.130.1.1-10.130.1.1:0
  SA:  ref=4 options=4a7 type=00 soft=0 mtu=1280 expire=43173/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000002 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43189/43200
  dec: spi=5c8e4a53 esp=aes key=32
34420e33f9e18eea020278fdeadfc111aa099f84e1dd2a0f25ec20fd242bff9f
       ah=sha1 key=20 4a4f61a0a89017d5ac711be76adb3631b1f24be5
  enc: spi=ec7bfcfc esp=aes key=32
80a0b64bdf7e1caca170a0641dca00dd657629af8913c5baeacc209edc4ccac3
       ah=sha1 key=20 ebd8bc8a2f88cd2dc8a2232fdadb7fc627694a77
  dec:pkts/bytes=72/11248, enc:pkts/bytes=0/0
  npu_flag=02 npu_rgwy=172.16.200.254:64917 npu_lgwy=172.16.200.4:4500npu_selid=3
  dec_npuid=1 enc_npuid=0 dec_engid=-1 enc_engid=-1 dec_saidx=3 enc_saidx=-1
```

```
# diagnose debug enable
# diagnose debug app ike -1
# diagnose debug app fnbam -1

ike V=root:0: udp comes 172.16.200.254:1012->172.16.200.4:500,ifindex=7,vrf=0,len=624....
ike V=root:0: IKEv2 exchange=SA_INIT id=bc5aa3bb99c0963c/0000000000000000 len=624
```

```
...
ike V=root:0:ikev2_native:89: reassembled fragmented message
ike V=root:0:ikev2_native:89: responder received AUTH msg
ike V=root:0:ikev2_native:89: processing notify type MOBIKE_SUPPORTED
ike V=root:0:ikev2_native:89: peer identifier IPV4_ADDR 172.17.249.189
ike V=root:0:ikev2_native:89: eap-peer=yes
ike V=root:0:ikev2_native:89: re-validate gw ID
ike V=root:0:ikev2_native:89: gw validation OK
...
ike V=root:0:ikev2_native:89: responder preparing EAP identity request
ike V=root:0:ikev2_native:89: local cert, subject='172.18.68.209', issuer='eku-ca'
ike V=root:0:ikev2_native:89: local CA cert, subject='eku-ca', issuer='eku-ca'
...
ike V=root:0:ikev2_native:89: sent IKE msg (AUTH_RESPONSE): 172.16.200.4:4500-
>172.16.200.254:64917, len=148, vrf=0, id=bc5aa3bb99c0963c/290cf660b0977bf5:00000001, oif=7
ike V=root:0: udp comes 172.16.200.254:64917->172.16.200.4:4500,ifindex=7,vrf=0,len=100....
ike V=root:0: IKEv2 exchange=AUTH id=bc5aa3bb99c0963c/290cf660b0977bf5:00000002 len=96
ike 0: in
BC5AA3BB99C0963C290CF660B0977BF52E2023080000000200000060300000442BFE21A653DCF88A4F0CA11AD6C68644EC
CF0388EE7FB9AB1E00C94DD75AE4A1D24007C230347591E0B69DADA0D9878F315EDC6D47851E78EF816D43CBA95A67
ike 0:ikev2_native:89: dec
BC5AA3BB99C0963C290CF660B0977BF52E2023080000000200000031300000040000011025A000D01746573745F726164
ike V=root:0:ikev2_native:89: responder received EAP msg
ike V=root:0:ikev2_native:89: send EAP message to FNBAM
ike V=root:0:ikev2_native:89: initiating EAP authentication
ike V=root:0:ikev2_native: EAP user "test_rad"
ike V=root:0:ikev2_native: auth group radius-group
ike V=root:0:ikev2_native: EAP 24730597347332 pending
...
ike V=root:0:ikev2_native:89 EAP 24730597347332 result FNBAM_CHALLENGED
[1098] __rad_del_job_timer-
[1386] freeze_auth_session-
ike V=root:0:ikev2_native: EAP challenged for user "test_rad"
ike V=root:0:ikev2_native:89: responder preparing EAP pass through message
ike 0:ikev2_native:89: enc 0000001A015B00160410812E67D63D7945B8A54A6049EEA98733050403020105
ike 0:ikev2_native:89: out
BC5AA3BB99C0963C290CF660B0977BF52E2023200000000200000060300000044669A4D0DFD7FC82E33253FD3AEB8ED5D55
0976ED0D67F21B71DB16A0ABEB36BC652DCC055A81D59E5A36C27E1BE94EF9135A01ED5E27E4009BBB1DEF26EC8925
ike V=root:0:ikev2_native:89: sent IKE msg (AUTH_RESPONSE): 172.16.200.4:4500-
>172.16.200.254:64917, len=96, vrf=0, id=bc5aa3bb99c0963c/290cf660b0977bf5:00000002, oif=7
ike V=root:0: udp comes 172.16.200.254:64917->172.16.200.4:4500,ifindex=7,vrf=0,len=84....
ike V=root:0: IKEv2 exchange=AUTH id=bc5aa3bb99c0963c/290cf660b0977bf5:00000003 len=80
ike 0: in
BC5AA3BB99C0963C290CF660B0977BF52E2023080000000300000050300000343 9E6158679F4F96936D70706661FE64307
3C60362D82775DCB70395B4B3FC89F31EF9D45E142EB6BB224B76BE92F1EE3
ike 0:ikev2_native:89: dec
BC5AA3BB99C0963C290CF660B0977BF52E202308000000030000002A30000004000000 0A025B0006031A
ike V=root:0:ikev2_native:89: responder received EAP msg
ike V=root:0:ikev2_native:89: send EAP message to FNBAM
ike V=root:0:ikev2_native: EAP 24730597347332 pending
...
[672] __group_match-Use 'test_rad' for user group matching.
```

```
[710] __group_match-Check if radius-server is a group member
[716] __group_match-Group 'radius-group' passed group matching
[719] __group_match-Add matched group 'radius-group'(7)
[276] find_matched_usr_grps-Passed group matching
[1024] update_auth_token_session-config does not require 2fa
[336] fnbamd_comm_send_result-Sending result 0 (nid 0) for req 24730597347332, len=2784
[668] destroy_auth_silent-delete session 24730597347332
[1938] handle_child_rsp-Auth rsp 24730597347332, session created, line 126
[1650] fnbamd_rad_stop-
[1271] __rad_stop-
[545] __rad_udp_close-closed.
[1266] __rad_conn_stop-Stop rad conn timer.
[1098] __rad_del_job_timer-
ike V=root:0:ikev2_native:89 EAP 24730597347332 result FNBAM_SUCCESS
ike V=root:0:ikev2_native: EAP succeeded for user "test_rad" group "radius-group" 2FA=no
ike V=root:0:ikev2_native:89: responder preparing EAP pass through message
...
ike V=root:0:ikev2_native:89: responder received AUTH msg
ike V=root:0:ikev2_native:89: peer identifier IPV4_ADDR 172.17.249.189
ike V=root:0:ikev2_native:89: eap-peer=yes
ike V=root:0:ikev2_native:89: auth verify done
ike V=root:0:ikev2_native:89: responder AUTH continuation
ike V=root:0:ikev2_native:89: authentication succeeded
ike V=root:0:ikev2_native:89: responder creating new child
ike V=root:0:ikev2_native:89: mode-cfg type 1 request 0:''
ike V=root:0:ikev2_native: mode-cfg allocate 10.130.1.1/0.0.0.0
ike V=root:0:ikev2_native:89: mode-cfg using allocated IPv4 10.130.1.1
...
ike V=root:0:ikev2_native_0: tunnel created tun_id 10.130.1.1/::10.0.0.5 remote_location 0.0.0.0
ike V=root:0:ikev2_native_0: added new dynamic tunnel for 172.16.200.254:64917
ike V=root:0:ikev2_native_0:89: established IKE SA bc5aa3bb99c0963c/290cf660b0977bf5
ike V=root:0:ikev2_native_0:89: check peer route: if_addr4_rcvd=0, if_addr6_rcvd=0, mode_cfg=1
ike V=root:0:ikev2_native_0:89: mode-cfg assigned (1) IPv4 address 10.130.1.1
ike V=root:0:ikev2_native_0:89: mode-cfg send (3) IPv4 DNS(1) 172.17.254.148
ike V=root:0:ikev2_native_0:89: mode-cfg send (3) IPv4 DNS(2) 8.8.8.8
ike V=root:0:ikev2_native_0:89: mode-cfg IPv6 DNS ignored, no IPv6 DNS servers found
ike V=root:0:ikev2_native_0:89:ikev2_native:1: replay protection enabled
ike V=root:0:ikev2_native_0:89:ikev2_native:1: set sa life soft seconds=43189.
ike V=root:0:ikev2_native_0:89:ikev2_native:1: set sa life hard seconds=43200.
ike V=root:0:ikev2_native_0:89:ikev2_native:1: IPsec SA selectors #src=1 #dst=1
ike V=root:0:ikev2_native_0:89:ikev2_native:1: src 0 7 0:0.0.0.0-255.255.255.255:0
ike V=root:0:ikev2_native_0:89:ikev2_native:1: dst 0 7 0:10.130.1.1-10.130.1.1:0
ike V=root:0:ikev2_native_0:89:ikev2_native:1: add dynamic IPsec SA selectors 483
ike V=root:0:ikev2_native_0:89:ikev2_native:1: added dynamic IPsec SA proxyids new 1 483
ike V=root:0:ikev2_native:1: add route 10.130.1.1/255.255.255.255 gw 10.130.1.1 oif ikev2_native
(49) metric 15 priority 1
ike V=root:0:ikev2_native_0:89:ikev2_native:1: tunnel 1 of VDOM limit 0/0
ike V=root:0:ikev2_native_0:89:ikev2_native:1: add IPsec SA: SPIs=5c8e4a53/ec7bfcfc
ike 0:ikev2_native_0:89:ikev2_native:1: IPsec SA dec spi 5c8e4a53 key
32:34420E33F9E18EEA020278FDEADFC111AA099F84E1DD2A0F25EC20FD242BFF9F auth
20:4A4F61A0A89017D5AC711BE76ADB3631B1F24BE5
ike 0:ikev2_native_0:89:ikev2_native:1: IPsec SA enc spi ec7bfcfc key
```

```
32:80A0B64BDF7E1CACA170A0641DCA00DD657629AF8913C5BAEACC209EDC4CCAC3 auth
20:EBD8BC8A2F88CD2DC8A2232FDADB7FC627694A77
ike V=root:0:ikev2_native_0:89:ikev2_native:1: added IPsec SA: SPIs=5c8e4a53/ec7bfcfc
ike V=root:0:ikev2_native_0:89:ikev2_native:1: sending SNMP tunnel UP trap
ike V=root:0:ikev2_native_0: tunnel up event assigned address 10.130.1.1
...
```

Native VPN as dialup client - iOS

When FortiGate is configured as a dialup VPN server, Apple devices, such as iPhones or iPads, with dynamic or unknown IP addresses can establish secure connections over the Internet to access private resources protected by FortiGate.

In this example, the FortiGate protects a local network (10.10.111.0/24) that a remote user on an iOS device needs to securely access over the Internet using a VPN connection. The FortiGate is configured as a dialup VPN server on wan1, and the iOS device is the dial-up IPsec VPN client.

You can configure dialup IPsec VPN with an iOS device as the dialup client using the VPN Wizard in the GUI or the CLI.

To configure IPsec VPN with an iOS device as the dialup client in the GUI:

1. Configure a local user:
   a. Go to *User & Authentication* > *User Definition* and click *Create New*.
   b. Set the *User Type* to *Local User*, then click *Next*.
   c. Set the *Login Credentials*, then click *Next*:
      - *Username*: *vpnuser1*
      - *Password*: Enter a suitable password
   d. Disable *Two-factor Authentication*, then click *Next*.
   e. Set *User Account Status* to *Enabled* and disable *User Group*, then click *Submit*.
2. Configure a local user group:
   a. Go to *User & Authentication* > *User Groups* and click *Create New*.
   b. Configure the following:

Field	Value
Name	vpngroup
Type	Firewall
Members	vpnuser1

   c. Click *OK*.

   For other types of authentication, such as LDAP, PKI, or SAML, see User & Authentication on page 3484.

3. Go to *VPN* > *VPN Wizard*.
4. Enter a name for the VPN in the *Tunnel name* field. In this example, *ios_client*
5. Set *Select a template* to *Remote Access*.
6. Click *Begin*.
7. Configure the *VPN tunnel* settings:

Field	Value
VPN client type	Native
Protocol	L2TP over IPsec
Client OS	macOS/iOS
Share WAN	wan1
Authentication method	Pre-shared key
Pre-shared key	Enter a suitable key. The same pre-shared key is used by the remote user on the iOS device to connect to the dialup VPN.
User group	vpngroup

8. Click *Next*.
9. Configure the *Remote Endpoint* settings:

Field	Value
Addresses to assign to connected endpoints	10.10.2.1-10.10.2.200

10. Click *Next*.
11. Configure the *Local FortiGate* settings:

Field	Value
Incoming interface that binds to tunnel	wan1 Disable Create and add interface to zone.
Local interface	lan
Local address	local_network Click + to create a new address object with *Type* set to *Subnet* and *IP/Netmask* set to *10.10.111.0/24*, if not already configured.

12. Click *Next*.
13. Review the configuration and, if everything is correct, click *Submit*.
    The IPsec VPN is configured and is visible on the *VPN* > *VPN Tunnels* page.

To configure IPsec VPN with an iOS device as the dialup client in the CLI:

1. Configure a local user:

```
config user local
    edit "vpnuser1"
        set type password
        set passwd ******
    next
end
```

2. Configure a local user group:

```
config user group
    edit "vpngroup"
        set member "vpnuser1"
    next
end
```

3. Configure the internal interface.

   The LAN interface connects to the corporate internal network. Traffic from this interface routes out the IPsec VPN tunnel. Creating an address group for the protected network behind this FortiGate causes traffic to this network group to go through the IPsec tunnel.

```
config system interface
    edit "lan"
        set vdom "root"
        set ip 10.10.111.1 255.255.255.0
    next
end
```

```
config firewall address
    edit "local_subnet_1"
        set subnet 10.10.111.0 255.255.255.0
    next
end
```

```
config firewall addrgrp
    edit "local_network"
        set member "local_subnet_1"
    next
end
```

4. Configure the WAN interface.

   The WAN interface is the interface connected to the ISP. It can be configured in static mode (as shown here), DHCP, or PPPoE mode. It is recommended to configure it with a static IP address to ensure that the IPsec VPN configuration on the on the iOS device stays unchanged if the WAN IP changes on the FortiGate. The IPsec tunnel is established over the WAN interface.

```
config system interface
    edit "wan1"
        set vdom "root"
```

```
        set ip 203.0.113.249 255.255.255.0
    next
end
```

5. Configure the client address pool.
   You must create a firewall address to assign an IP address to a client from the address pool.

```
config firewall address
    edit "client_range"
        set type iprange
        set comment "VPN client range"
        set start-ip 10.10.2.1
        set end-ip 10.10.2.200
    next
end
```

6. Configure the IPsec phase1-interface.
   PSK is used as the authentication method in this example. Signature authentication is also an option.

```
config vpn ipsec phase1-interface
    edit "ios_client"
        set type dynamic
        set interface "wan1"
        set ike-version 2
        set peertype any
        set net-device disable
        set mode-cfg enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set dpd on-demand
        set dpd-retryinterval 20
        set xauthtype auto
        set authusrgrp "vpngroup"
        set assign-ip-from name
        set ipv4-name "client_range"
        set dns-mode auto
        set ipv4-split-include "local_network"
        set save-password enable
        set psksecret *****
    next
end
```

7. Configure the IPsec phase2-interface:

```
config vpn ipsec phase2-interface
    edit "ios_client"
        set phase1name "ios_client"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set pfs disable
    next
end
```

8. Configure a firewall policy to allow client traffic flow over the IPsec VPN tunnel to the FortiGate's lan interface:

```
config firewall policy
    edit 1
        set name "inbound"
        set srcintf "ios_client"
        set dstintf "lan"
        set srcaddr "client_range"
        set dstaddr "local_network"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end
```

To configure the iOS device:

1. On the iOS device, go to *Settings* > *General* > *VPN & Device Management* > *VPN* and select *Add VPN Configuration*.
2. Configure the following details:

Field	Value
Type	IPsec
Description	Enter a suitable description.
Server	203.0.113.249 (the wan1 IP address of the FortiGate the is configured as the IPsec VPN dialup server)
Account	vpnuser1 (the local user configured on the FortiGate)
Password	Enter the password configured for the user.
Secret	Enter the pre-shared key that is configured on the FortiGate.

3. Leave the remaining settings as their default values and press *Done*.
4. Go to *Settings* > *General* > *VPN & Device Management* > *VPN*
5. Highlight the IPsec VPN configuration (indicated by a checkmark) and press the *Not Connected* toggle to enable the VPN.
6. If prompted, enter your user credentials.
7. The *VPN Status* changes to *Connected* and a VPN icon appears at the top of the screen.

To verify the tunnel status in the GUI:

1. On the FortiGate, go to *Dashboard* > *Network Monitor* > *VPN*.
2. Find the new tunnel, *ios_client_0* in this example, and check the statuses of the *Phase 1* and *Phase 2* selectors.

> Successfully established dialup IPsec VPN tunnels are displayed using a parent name followed by an index, creating a structured and organized association with the parent tunnel. For example, the parent name *ios_client* results in tunnel names such as *ios_client_0*, *ios_client_1*, and so on.

3. Hover the cursor over the tunnel name to see additional details.

To verify the tunnel status in the CLI:

1. Verify the IPsec Phase 1 tunnel status on the FortiGate:

```
FortiGate # diagnose vpn ike gateway list name ios_client_0

vd: root/0
name: ios_client_0
version: 1
interface: wan1 3
addr: 203.0.113.249:4500 -> 198.51.100.171:40398
tun_id: 198.51.100.171/::10.0.0.8
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 22s ago
xauth-user: vpnuser1
2FA: no
peer-id: 0.0.0.0
peer-id-auth: no
assigned IPv4 address: 10.10.2.1/255.255.255.255
nat: me peer
pending-queue: 0
IKE SA: created 1/1  established 1/1  time 330/330/330 ms
IPsec SA: created 1/1  established 1/1  time 280/280/280 ms

  id/spi: 5 456ee87e1ccab0d5/a2d3d3f43d7c1147
  direction: responder
  status: established 22-22s ago = 330ms
  proposal: aes256-sha256
  key: c5b56bf00a715ad4-7332f707363766b5-9d7fc85b5003227c-18ab12ec2a8c4823
  QKD: no
  PQC-KEM (IKE): no
  PQC-KEM (all IPsec): no
  lifetime/rekey: 3600/3307
  DPD sent/recv: 00000000/00000000
  peer-id: 0.0.0.0
```

Note that FortiGate acts as the responder, establishes the IPsec tunnel with remote iOS device with IP 198.51.100.171 over UDP NAT-T port 4500 on its wan1 interface, and dynamically assigns a VPN IP of 10.10.2.1 to the iOS device. The authenticated username is also displayed in `xauth-user` field.

2. Verify the IPsec Phase 2 tunnel status on either the Branch or HQ:

```
FortiGate # diagnose vpn tunnel list name ios_client_0
list ipsec tunnel by names in vd 0
------------------------------------------------------
name=ios_client_0 ver=1 serial=a 203.0.113.249:4500->198.51.100.171:40398
nexthop=203.0.113.250 tun_id=198.51.100.171 tun_id6=::10.0.0.8 status=up dst_mtu=1200 weight=1
bound_if=3 real_if=3 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/74664 options
[123a8]=npu rgwy-chg rport-chg frag-rfc  run_state=0 role=primary accept_traffic=1 overlay_
id=0

parent=ios_client index=0
proxyid_num=1 child_num=0 refcnt=7 ilast=42956699 olast=42956699 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-demand on=1 status=ok idle=20000ms retry=3 count=0 seqno=0
natt: mode=keepalive draft=32 interval=10 remote_port=40398
fec: egress=0 ingress=0
proxyid=ios_client proto=0 sa=1 ref=2 serial=1 add-route
  src: 0:10.10.111.0-10.10.111.255:0
  dst: 0:10.10.2.1-10.10.2.1:0
  SA:  ref=3 options=20083 type=00 soft=0 mtu=1118 expire=3032/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=3588/3600
  dec: spi=1bddf144 esp=aes key=32
7b869ab5470bcd36ea84772b526081067fedbfb3c82347d67da27624d757750c
       ah=sha256 key=32 e25c84a309cdb0ab1dd2b1fd2953e5c74a4b0b9a8ad20cbcf046edf5edd3174e
  enc: spi=01e1446e esp=aes key=32
030d2fd16ce78591ccc87ff28e8bc74b7a4a11acaab7fe54f5abea9609f02cd1
       ah=sha256 key=32 45a02f1476faa30afcf55fdb4a058464702983076bf7877ef27201e75e9ba786
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=198.51.100.171 npu_lgwy=203.0.113.249 npu_selid=4 dec_npuid=0 enc_
npuid=0
```

Note the decryption, encryption, and authentication keys negotiated in Phase 2 to protect subnets under the `src` and `dst` fields. FortiOS also automatically adds the dynamically assigned VPN IP of 10.10.2.1 to its traffic selectors for dialup tunnel for_client_0.

To verify traffic through the IPsec tunnel:

1. On the iOS device with public IP 198.51.100.171 open an app that has the ping utility and initiate ICMP requests the a machine in the 10.10.111.0/24 network, then verify that traffic goes through:

```
# ping 10.10.111.3
PING 10.10.111.3 (10.10.111.3) 56(84) bytes of data.
64 bytes from 10.10.111.3: icmp_seq=1 ttl=62 time=9.82 ms
64 bytes from 10.10.111.3: icmp_seq=2 ttl=62 time=0.809 ms
```

2. On FortiGate, run a sniffer trace for ICMP packets destined to 10.10.111.3:

```
# diagnose sniffer packet any 'host 10.10.111.3’ 4 0 l
Using Original Sniffing Mode
interfaces=[any]
filters=[host 10.10.111.3]
2024-11-05 12:06:05.509711 ios_client_0 in 10.10.2.1 -> 10.10.111.3: icmp: echo request
```

```
2024-11-05 12:06:05.509791 lan out 10.10.2.1 -> 10.10.111.3: icmp: echo request
2024-11-05 12:06:05.518809 lan in 10.10.111.3 -> 10.10.2.1: icmp: echo reply
2024-11-05 12:06:05.518899 ios_client_0 out 10.10.111.3 -> 10.10.2.1: icmp: echo reply
```

The ICMP request packets are received on ios_client_0 tunnel interface and routed to the lan interface. ICMP reply packets are received on the lan interface and routed back out on ios_client_0 tunnel interface.

For more information, see Performing a sniffer trace or packet capture on page 4966.

3. Optionally, run real time IKE debug using appropriate filters for troubleshooting if the IPsec tunnel does not come up:

```
FortiGate # diagnose vpn ike log filter rem-addr4 198.51.100.171
FortiGate # diagnose debug application ike -1
FortiGate # diagnose debug enable
```

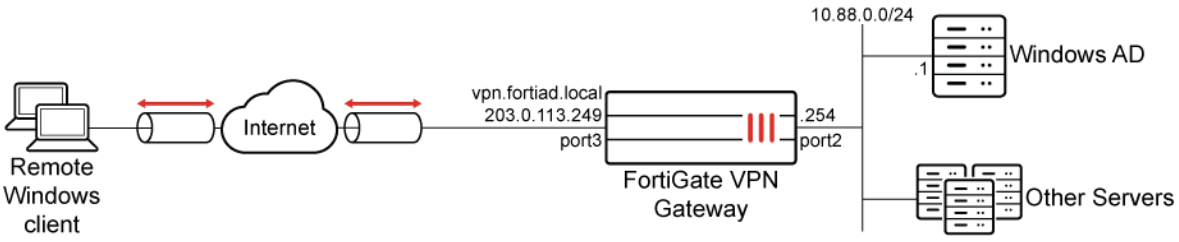
For more information, see VPN IPsec troubleshooting on page 3335.

Windows IKEv2 native VPN with user certificate

In this example, we configure remote access VPN for a Windows environment where remote clients use the built-in VPN client to connect over IKEv2 to the FortiGate VPN gateway. IKEv2 with Extensible Authentication Protocol – Transport Layer Security (EAP-TLS) using mutual certificate authentication is configured. Mutual certificate authentication means that both the client and server use certificates to identify themselves. Pre-shared Keys and password authentication are not used.

EAP-TLS is compatible with RADIUS, which is handled by the Network Policy Server (NPS) on the Windows server. Certificates are generated and distributed through Active Directory Certificate Services (AD CS). An additional certificate is used to identify the VPN gateway.

Topology



Prerequisites

Windows server

This example assumes that the following Windows server roles are installed and available:

- NPS (RADIUS)
- AD CS with a generated CA
- Group Policy Management

- DNS server

It is also assumed that a connection is established between the NPS and FortiGate, and a DNS entry exists for the NPS that the FortiGate can resolve.

Domain and users

Users are defined in the Active Directory and belong to various groups. The NPS (RADIUS) server defines a different policy for each group and returns the name of each group in a FortiGate Vendor-specific attribute (12356).

This example has 2 groups in the fortiad.info domain:

AD Group	Vendor-specific attribute group name	User
FortiAD\MIS	rad_mis	tsmith (Tom Smith)
FortiAD\Finance	rad_finance	dparker (Dan Parker)

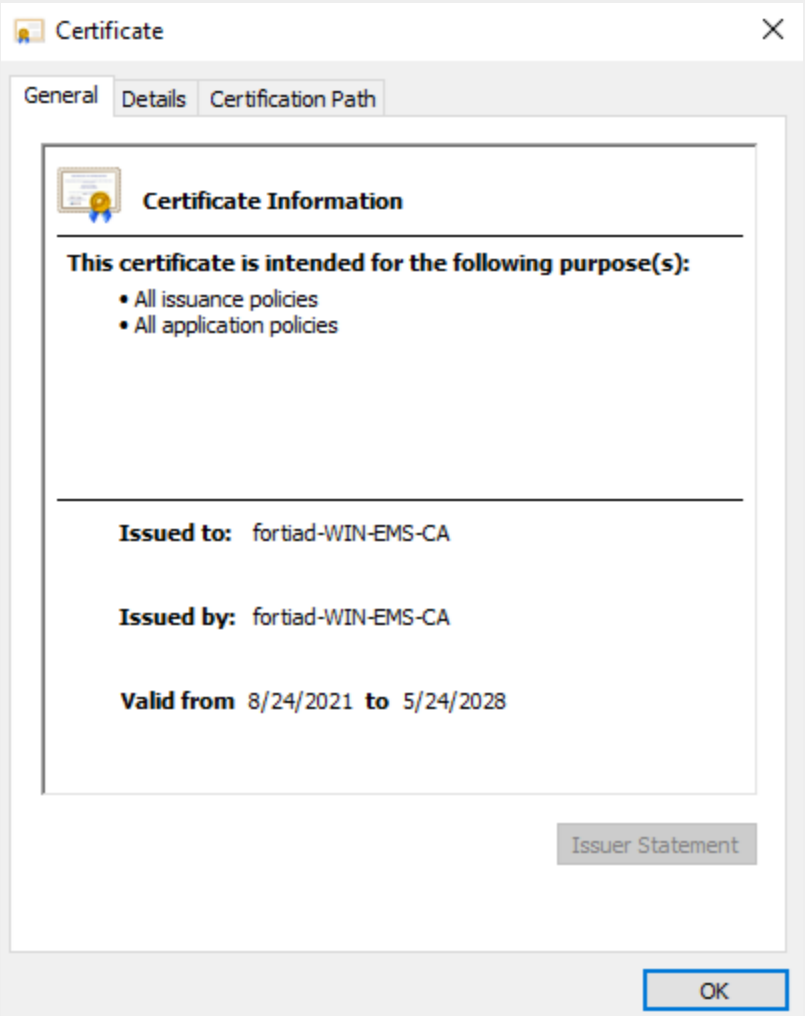
Certificates

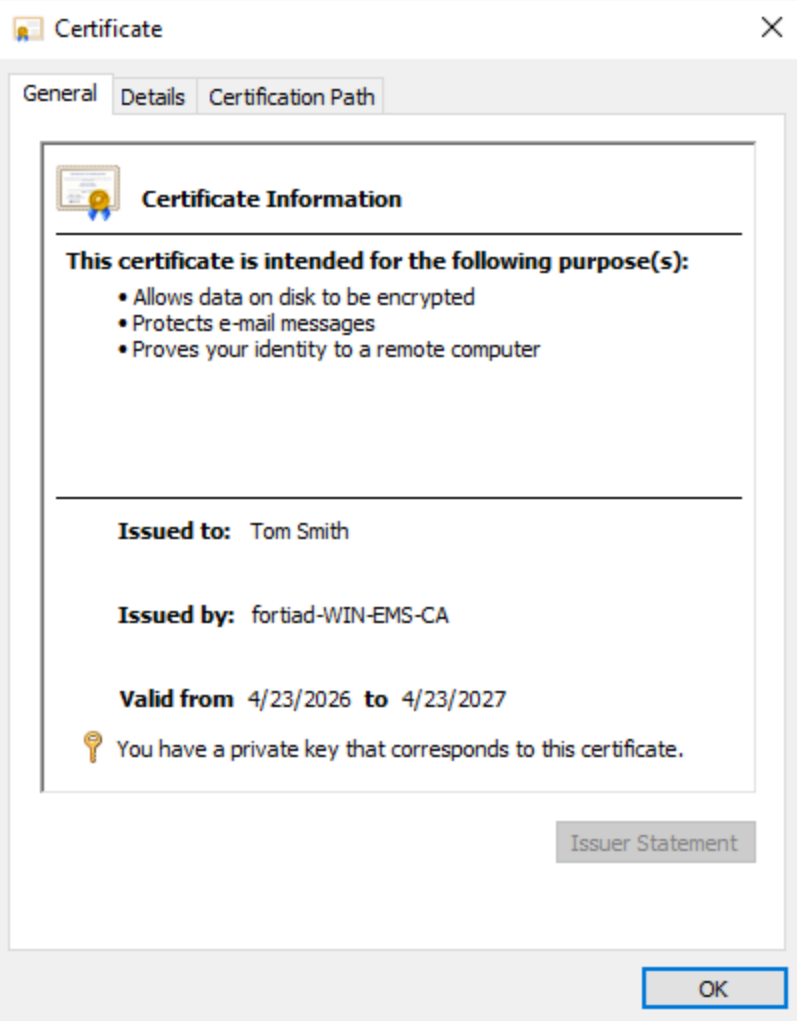
The following certificates are required:

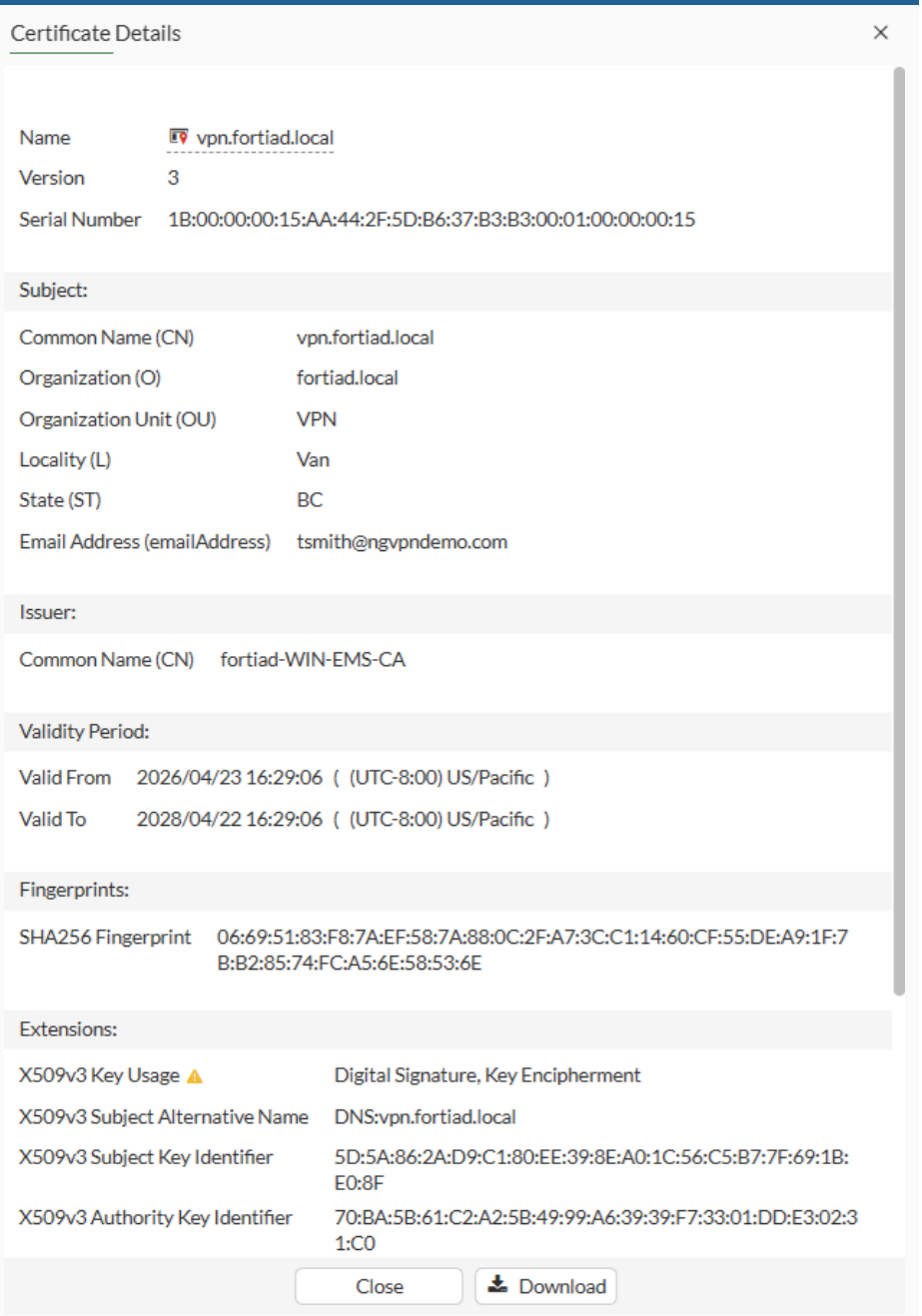
- CA certificate for EAP-TLS to sign the client and server certificates.
- Client certificate for EAP-TLS used by the windows client.
- Server certificate for EAP-TLS used by the server providing RADIUS authentication.
- VPN certificate used to identify the FortiGate dialup gateway.

To further detail the purpose of each type of certificate:

Certificate Type	Details
CA certificate	The CA certificate must be able to sign other certificates. It is created after AD CSs CA role installation.The CA certificate is automatically installed on the server that is hosting the AD CS role. In this example, that server is also hosting the NPS and DNS server. The *Key Usage* specifies *Certificate Signing*. In this example:

Certificate Type	Details
	
Client certificate	The client certificate is stored in the personal user certificate store and is used to authenticate the user. The certificate has *Client Authentication* and a SAN of the user's name, and is signed by the CA. The CA is stored in *Current User > Trusted Root Certification Authorities*. In this example:

Certificate Type	Details
	
Server certificate	The NPS certificate must be in the hosting server's certificate store so that the NPS can access it. It has *Server Authentication* and a SAN DNS name to match the server's IP address.
VPN gateway certificate	The FortiGate generates a CSR and stores the private key for this VPN gateway certificate. This CSR is signed by the Windows CA and uploaded to the FortiGate. The certificate has a SAN indicating the FQDN which resolves to the VPN gateway's WAN interface. Clients should use the FQDN to connect to the VPN. In this example:

Certificate Type	Details
	

For instructions on how to configure various parts of the Windows server and certificates, see the following sections:

- Configure the Windows server on page 3088
  - To create a certificate template to enable automatic enrollment for the user groups: on page 3088
  - To create a group policy to enable automatic enrollment: on page 3089
  - To verify that users are receiving certificates: on page 3090
  - To generate and sign a CSR and import the signed certificate to the FortiGate: on page 3090

- To configure network policies on the RADIUS server: on page 3091
- To add the FortiGate as a RADIUS client: on page 3093
- To create a DNS entry for the VPN connection: on page 3094

Configure the FortiGate

An VPN tunnel is configured using the VPN wizard to allow native windows clients to connect to the VPN using their built-in VPN clients. In this example, EAP-TLS is used, where each client uses their CA issued user certificate to authenticate against the NPS (RADIUS) server. FortiGate acts as a relay to pass the authentication requests to the RADIUS server. Subsequently, when the RADIUS identifies the user and user group from the client certificate, it returns the group-name via the Fortinet Vendor-Specific attribute 12356 to the FortiGate.

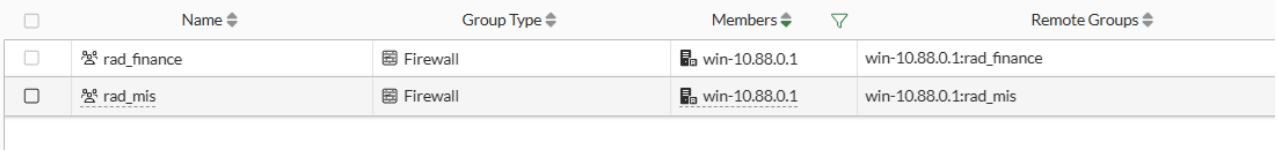
All traffic, including internet traffic, is routed through the tunnel. Role-based access control is used with different policies defined for each user group.

To configure the RADIUS server and user groups in the GUI:

1. Go to *User & Authentication* > *RADIUS Servers* and click *Create New*.
2. Configure the following:

Field	Value
Name	win-10.88.0.1
IP/Name	10.88.0.1
Secret	<secret>

   For better security, use RADSEC.
3. Click *OK*.
4. Go to *User & Authentication* > *User Groups* and click *Create new*.
5. Set *Name* to *rad_finance*.
6. In the *Remote Groups* section, click *Add*.
   a. Set *Remote Server* to *win-10.88.0.1*.
   b. Set *Groups* to Specify and enter the group name: *rad_finance*
   c. Click *OK*.
7. Click *OK*.
8. Click *Create new* again to create a second group named *rad_mis*.

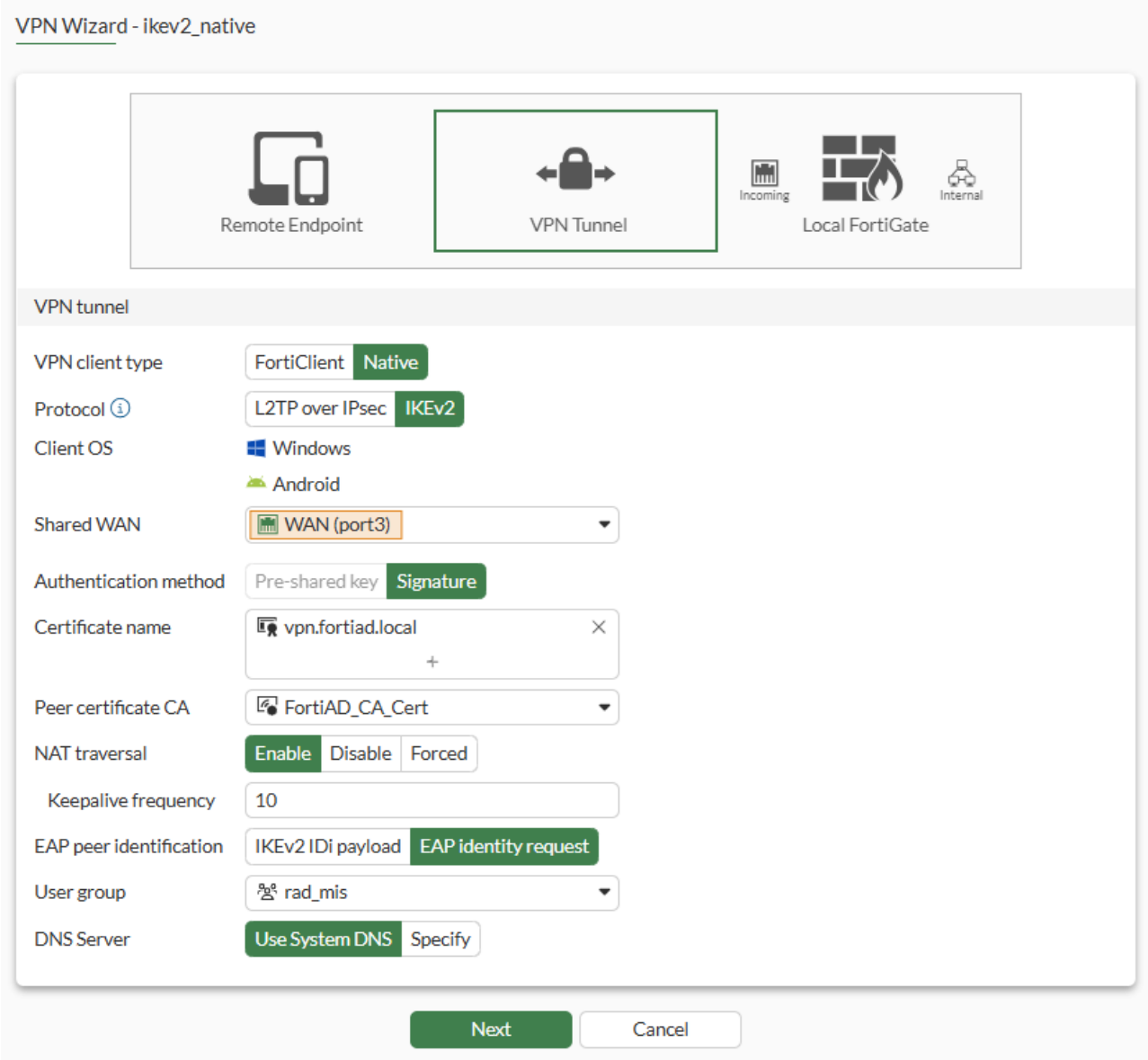


To configure IPsec VPN in the GUI:

1. Go to *VPN* > *IPsec Wizard*.
2. Enter a name for the VPN, such as *ikev2_native*.
3. Set *Template type* to *Remote Access*, then click *Begin*.

4. In the *VPN tunnel* settings page, configure the following:

Field	Value
VPN Client Type	Native
Protocol	IKEv2
Client OS	Defaults to Windows and Android
Shared WAN	WAN (port3)
Authentication method	Signature (For IKEv2 native VPNs, PSK is not supported)
Certificate name	vpn.fortiad.local The VPN gateway certificate
Peer certificate CA	FortiAD_CA_Cert CA certificate the signed the remote, server and VPN certificates
EAP peer identification	EAP identity request
User group	rad_mis Authentication on individual policies is configured later.
DNS Server	Use System DNS



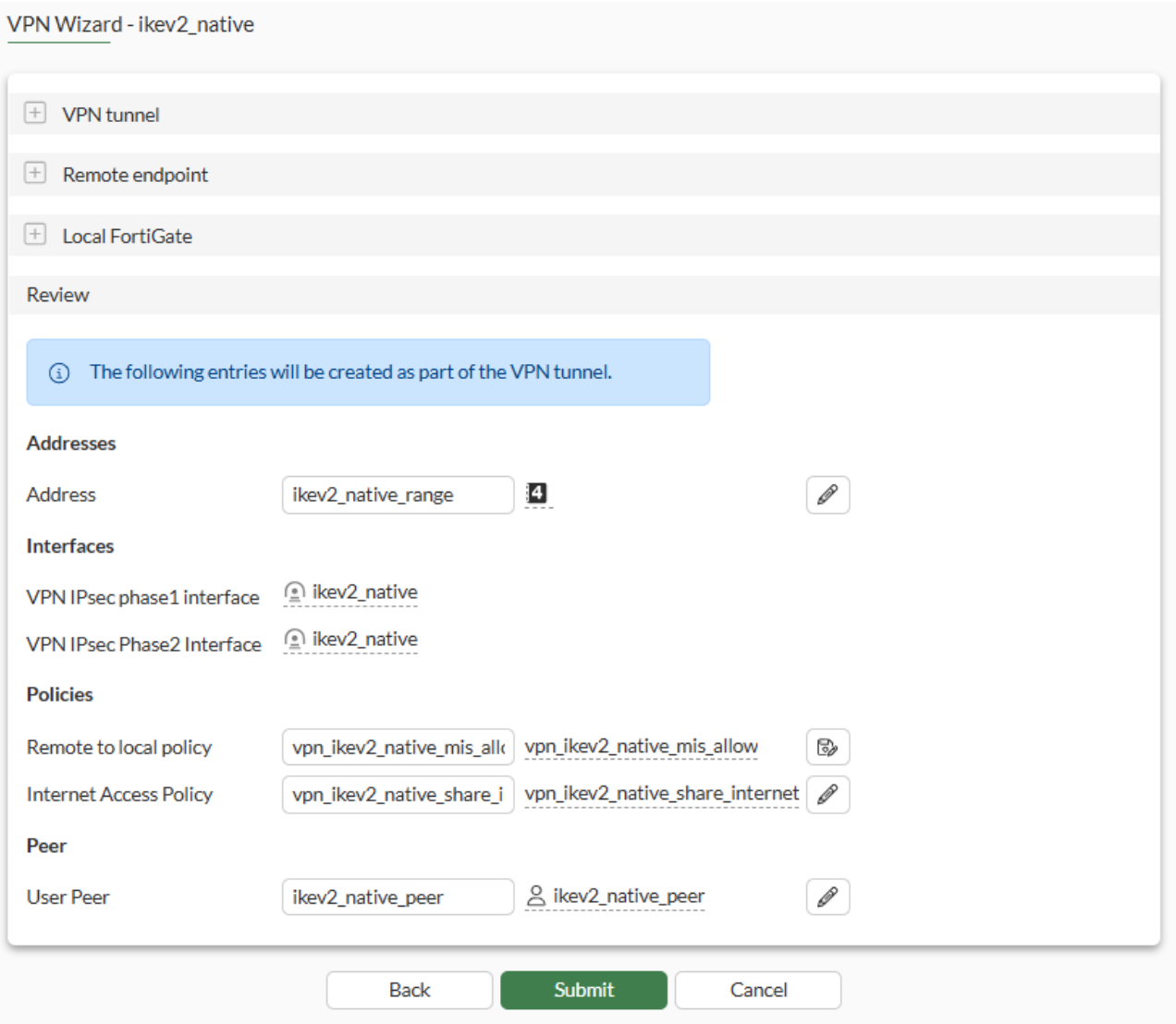
5. Click *Next*.
6. In the *Remote endpoint* settings page, configure the following:

Field	Value
Addressing mode for connected endpoints	Manual
Addresses to assign to connected endpoints	10.130.1.1-10.130.1.20
Subnet for connected endpoints	255.255.255.255

7. Click *Next*.
8. In the *Local FortiGate* settings page, configure the following:

Field	Value
Incoming interface that binds to tunnel	(WAN) port3
Create and add interface to zone	Disable
Local interface	DMZ (port2)
Local Address	DMZ_Network (10.88.0.0/24)

9. Click *Next*.
10. On the final review page, review the settings that will be installed.
    Since we will configure additional local policies, we will rename the current Remote to local policy to *vpn_ikev2_native_mis_allow*.

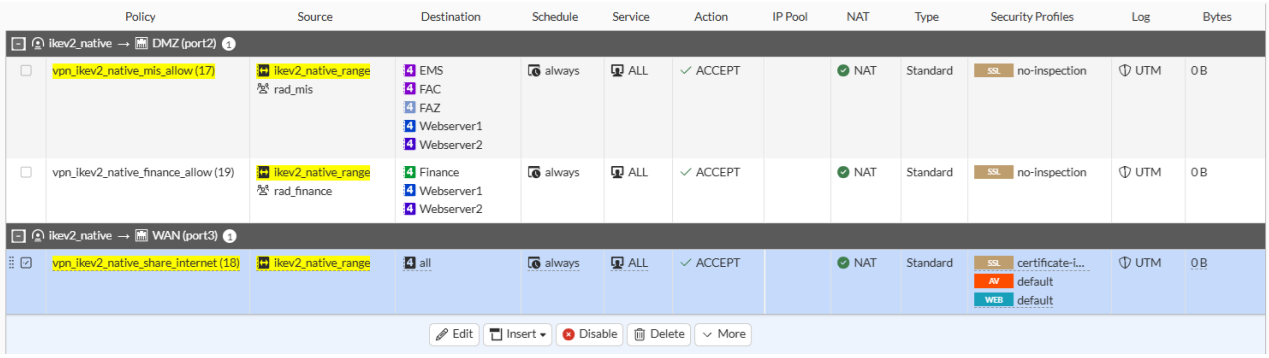


11. Click *Submit*.
12. After the VPN tunnel is configured, edit the tunnel:

a. Under *Authentication*, change *User group* from *Specify* to *Inherit from policy*.
b. Click *OK*.

To configure the firewall policies in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy*.
   Two policies are configured by default for from the ikev2_native to local, and ikev2_native to Internet.
2. Edit the ikev2_native to local policy.
   a. Set *User/group* to *rad_mis*.
   b. Set *Destination* to specific servers accessible by MIS. For instance, EMS, FAZ, FAC, Webserver1 and Webserver2.
   c. Click *OK*.
3. Copy and Paste the previous policy then click *Edit*.
   a. Set the name to *vpn_ikev2_native_finance_allow*.
   b. Change the *User/group* to *rad_finance*.
   c. Set the *Destination* to specific servers accessible by MIS. For instance, Webserver1, Webserver2 and Finance.
   d. Enable the policy.
   e. Click *OK*.
4. Edit the ikev2_native to Internet policy to apply inspection that is appropriate for Internet traffic.



To review the configurations in CLI:

```
config vpn ipsec phase1-interface
    edit "VPN1"
        set type dynamic
        set interface "port1"
        set ike-version 2
        set authmethod signature
        set peertype any
        set net-device disable
        set mode-cfg enable
        set ipv4-dns-server1 192.168.1.100
        set proposal aes128-sha256 aes256-sha256 aes128-sha1
        set localid "vpn.lab.local"
        set dpd on-idle
        set dhgrp 14 5 2
```

```
        set eap enable
        set eap-identity send-request
        set certificate "vpn.lab.local"
        set ipv4-start-ip 10.58.58.1
        set ipv4-end-ip 10.58.58.10
        set ipv4-split-include "10/8_net"
        set dpd-retryinterval 60
    next
end
config vpn ipsec phase2-interface
    edit "VPN1"
        set phase1name "VPN1"
        set proposal aes128-sha256 aes256-sha256 aes128-sha1
        set pfs disable
        set keepalive enable
        set src-addr-type name
        set dst-addr-type name
        set src-name "all"
        set dst-name "all"
    next
end
```

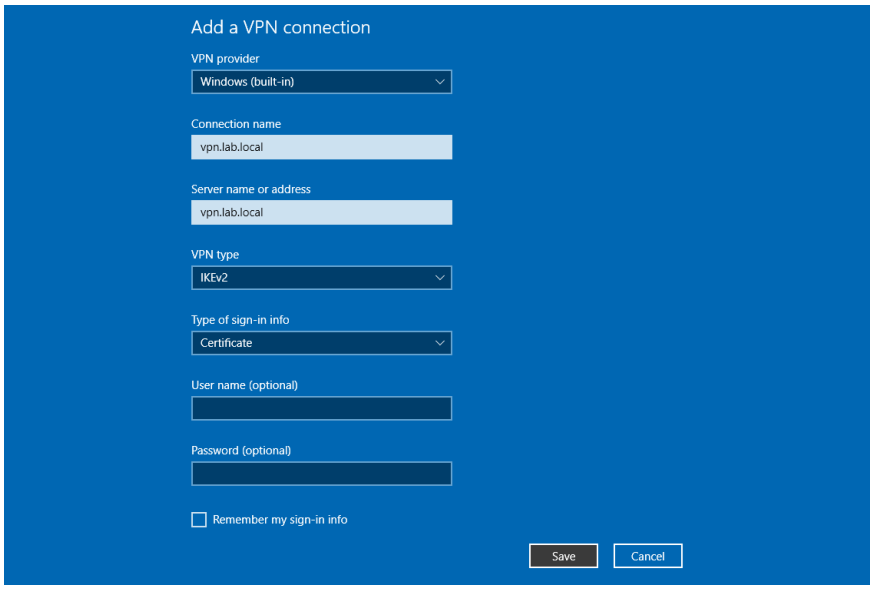
Configure the Windows client

The configuration is done on a Windows 10 endpoint.

To add VPN connection and configure a VPN interface:

1. Open the *Settings* page and go to *Network & Internet > VPN*.
2. Click *Add a VPN connection*.
3. Configure the following:

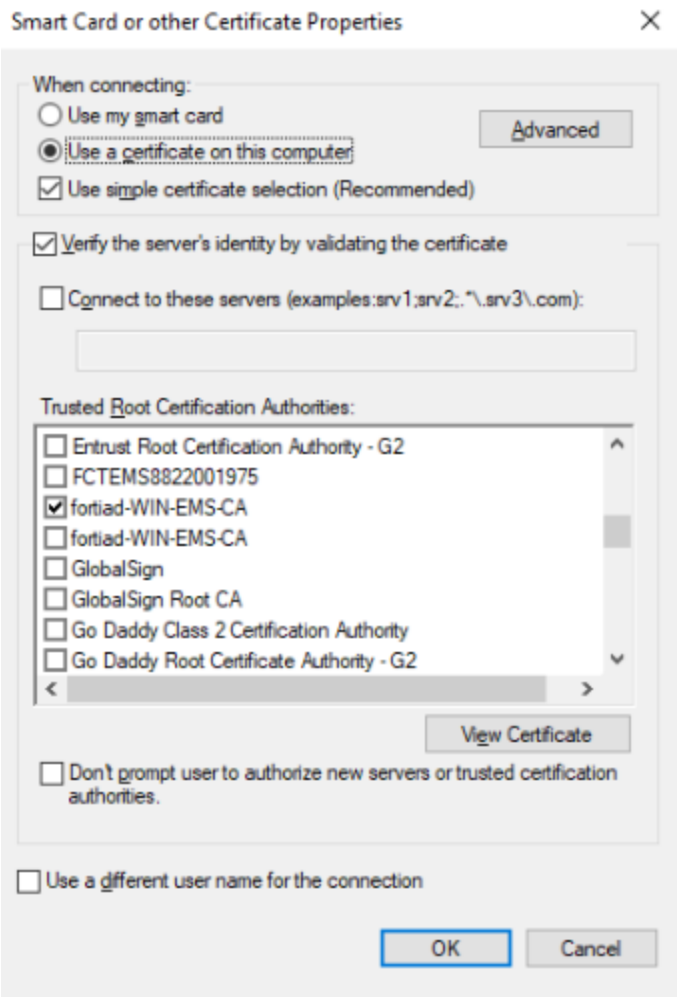
VPN provider	Windows (built-in)
Connection name	vpn.lab.local
Server name or address	vpn.lab.local
VPN type	IKEv2
Type of sign-in info	Certificate



4. Click *Save*.
5. If you connected immediately but not all the certificates are in place or trusted, you may get an error like the following:

Click *Connect* to continue anyway.

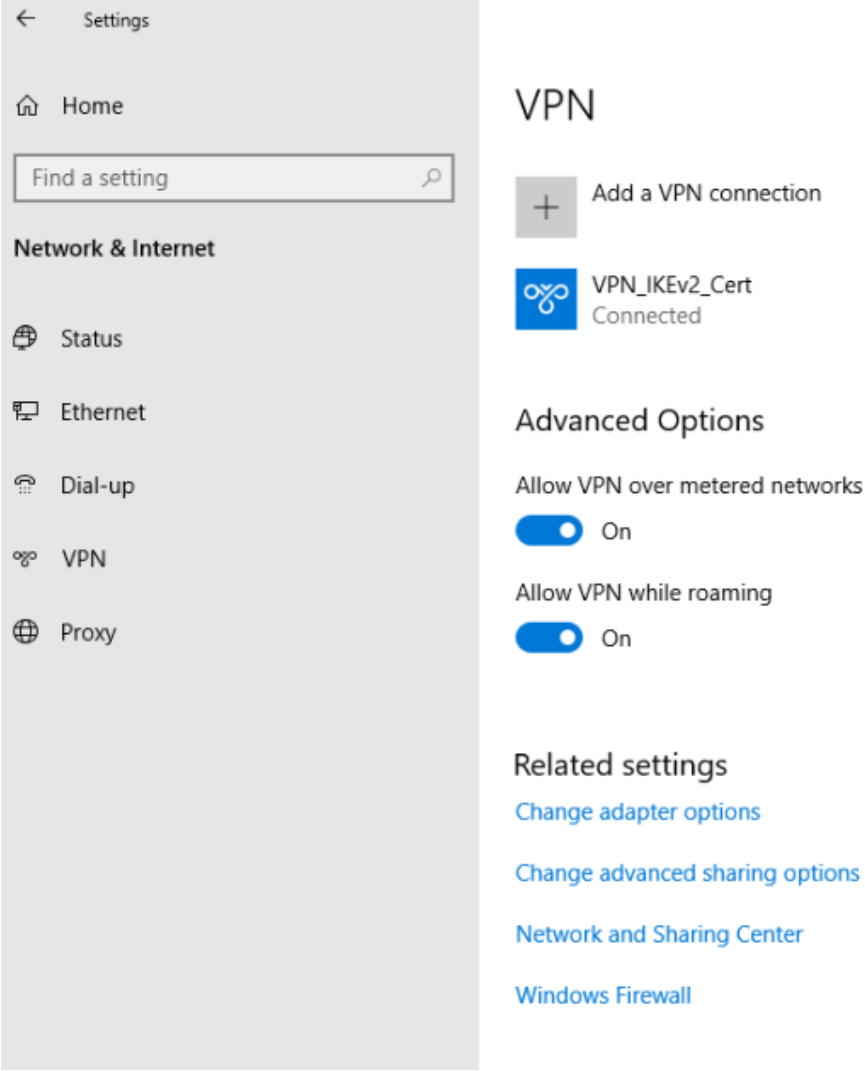
6. To explicitly trust the signing CA, go to *Network & Internet > Status* and, under *Advanced network settings*, click *Change adapter options*.
7. Select the VPN connection then click *Change settings of this connection*, or right-click on the connection and select *Properties*:
   a. Go to the *Security* tab and, in the *Authentication* section, click *Properties*.
   b. Enable *Verify the server's identity by validating the certificate*.
   c. Disable *Connect to these servers*.
   d. In the *Trusted Root Certificate Authorities* list, select the CA *fortiad-WIN-EMS-CA*.



e. Click *OK*, then click *OK* again.

To test the connection:

1. Log in to the Windows endpoint as user1.
2. Open the network settings and connect to the *VPN_IKEv2_Cert* VPN.

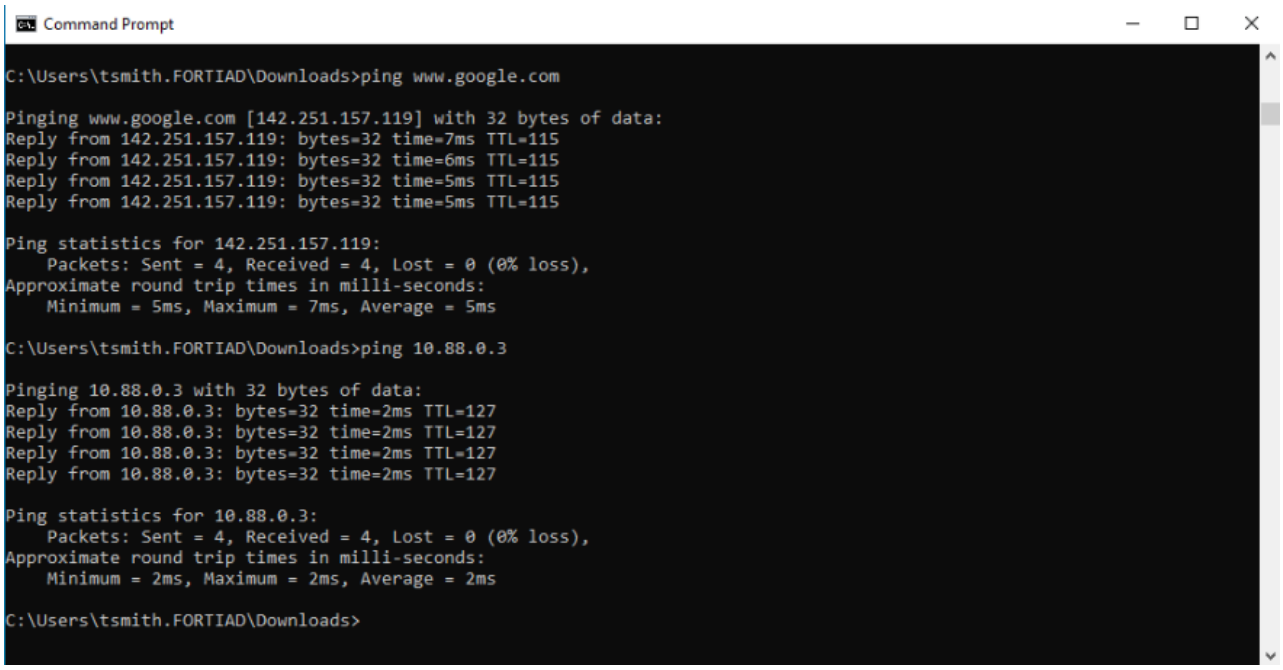


3. In the Windows command prompt, use ipconfig /all to see the interface details:

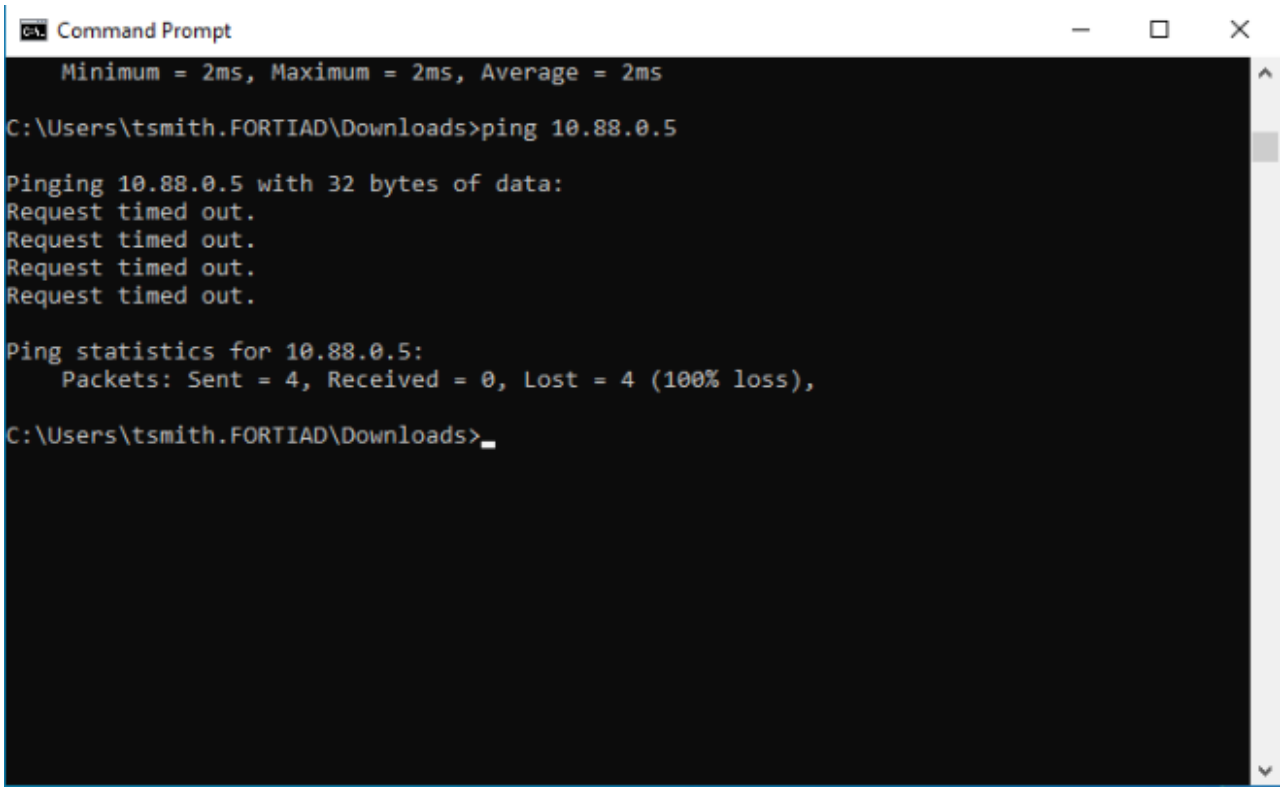
```
C:\Users\tsmith.FORTIAD\Downloads>ipconfig /all

Windows IP Configuration

   Host Name . . . . . . . . . . . . : WIN10-01
   Primary Dns Suffix  . . . . . . . : fortiad.info
   Node Type . . . . . . . . . . . . : Hybrid
   IP Routing Enabled. . . . . . . . : No
   WINS Proxy Enabled. . . . . . . . : No
   DNS Suffix Search List. . . . . . : fortiad.info

…

PPP adapter VPN_IKEv2_Cert:

   Connection-specific DNS Suffix  . :
   Description . . . . . . . . . . . : VPN_IKEv2_Cert
```

```
    Physical Address. . . . . . . . . :
    DHCP Enabled. . . . . . . . . . . : No
    Autoconfiguration Enabled . . . . : Yes
    IPv4 Address. . . . . . . . . . . : 10.130.1.1 (Preferred)
    Subnet Mask . . . . . . . . . . . : 255.255.255.255
    Default Gateway . . . . . . . . . : 0.0.0.0
    DNS Servers . . . . . . . . . . . : 8.8.8.8
                                        208.91.112.52
    NetBIOS over Tcpip. . . . . . . . : Enabled
```

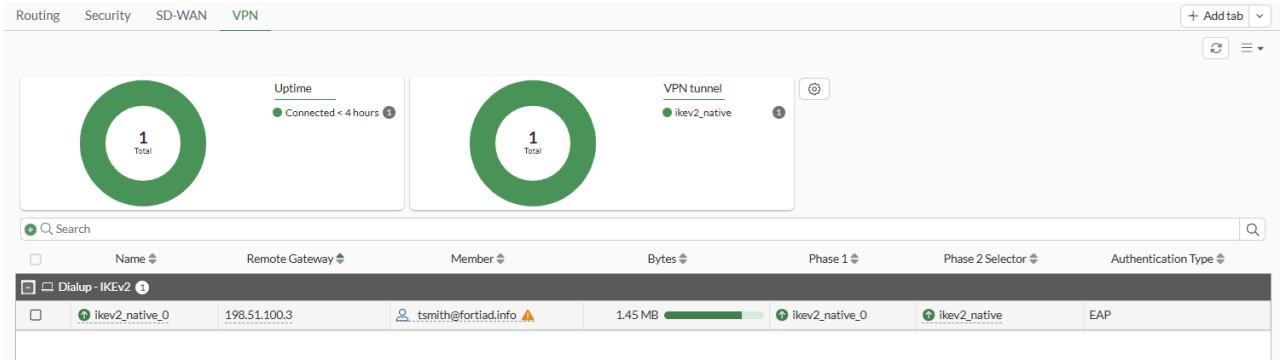
4. Pinging the internet and server should be successful.



5. Pinging unauthorized devices should be unsuccessful.



6. On the FortiGate, view the remote user on the *Dashboard* > *Network Monitor* > *VPN* page.



7. Go to the *Command Palette* (CTRL+P) > *CLI Diagnostics*. Search for *vpn ike gateway* and open the debug console. The outputs of the tunnel debugs are displayed.

```
vd: root/0
name: ikev2_native_0
version: 2
interface: port3 5
addr: 203.0.113.249:4500 -> 198.51.100.3:4500
tun_id: 10.130.1.1/::10.0.0.18
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 4678s ago
eap-user: tsmith@fortiad.info
2FA: no
```

```
groups:
  rad_mis 9
assigned IPv4 address: 10.130.1.1/0.0.0.0
pending-queue: 0
PPK: no
IKE SA: created 1/1  established 1/1  time 80/80/80 ms
IPsec SA: created 1/2  established 2/2  time 0/0/0 ms

  id/spi: 6 020e1a72f1bf999a/e218cfa16a453939
  direction: responder
  status: established 4678-4678s ago = 80ms
  proposal: aes256-sha256
  child: no
  SK_ei: 7da383e491891e5d-d62d626f5acbe51d-8832e851c4f073ea-87789a94fde89285
  SK_er: 1f47117a7826662b-ee20090c11fa947a-6688c2ee7a2aa15e-19de24909f48cda6
  SK_ai: 9e7b3ee3b514acab-702ebb53eba7495a-7c82f9dcf03c8cb3-9968d852aca1ddbf
  SK_ar: 9cd8ac3ac823e4ab-a641c7ce3610ba72-403e39f1e17a25a7-1029eb2b51c49d4a
  PPK: no
  message-id sent/recv: 25/11
  QKD: no
  PQC-KEM (IKE): no
  PQC-KEM (all IPsec): no
  lifetime/rekey: 86400/81451
  DPD sent/recv: 00000019/00000019
```

8. Running real-time debugs shows the negotiations as the tunnel comes up.

```
# diagnose debug application ike -1
Debug messages will be on for 30 minutes.

# diagnose debug application fnbamd -1
Debug messages will be on for 30 minutes.

# diagnose debug enable

# ike V=root:0: udp comes 198.51.100.3:500->203.0.113.249:500,ifindex=5,vrf=0,len=1104....
…
ike V=root:0:1b7426f242a51912/0000000000000000:7: SA proposal chosen, matched gateway ikev2_
native
ike V=root:0:ikev2_native:ikev2_native: created connection: 0x55f5098c82f0 5 203.0.113.249-
>198.51.100.3:500.
ike V=root:0:ikev2_native:7: processing notify type NAT_DETECTION_SOURCE_IP
ike V=root:0:ikev2_native:7: processing NAT-D payload
ike V=root:0:ikev2_native:7: NAT not detected
ike V=root:0:ikev2_native:7: process NAT-D
ike V=root:0:ikev2_native:7: processing notify type NAT_DETECTION_DESTINATION_IP
ike V=root:0:ikev2_native:7: processing NAT-D payload
ike V=root:0:ikev2_native:7: NAT not detected
ike V=root:0:ikev2_native:7: process NAT-D
ike V=root:0:ikev2_native:7: processing notify type FRAGMENTATION_SUPPORTED
ike V=root:0:ikev2_native:7: responder preparing SA_INIT msg
ike V=root:0:ikev2_native:7: create NAT-D hash local 203.0.113.249/500 remote 198.51.100.3/500
```

```
ike V=root:0:ikev2_native:7: sending CERTREQ payload (len=21)
…
ike V=root:0:ikev2_native:7: sent IKE msg (SA_INIT_RESPONSE): 203.0.113.249:500-
>198.51.100.3:500, len=341, vrf=0, id=1b7426f242a51912/63966ac9bbf0227d, oif=5
…
ike V=root:0: IKEv2 exchange=AUTH
…
ike V=root:0:ikev2_native:7: reassembled fragmented message
ike V=root:0:ikev2_native:7: responder received AUTH msg
ike V=root:0:ikev2_native:7: processing notify type MOBIKE_SUPPORTED
ike V=root:0:ikev2_native:7: peer identifier IPV4_ADDR 198.51.100.3
ike V=root:0:ikev2_native:7: eap-peer=yes
ike V=root:0:ikev2_native:7: re-validate gw ID
ike V=root:0:ikev2_native:7: gw validation OK
ike V=root:0:ikev2_native:7: received peer certreq '0D905DFB98916D025BF658D03BF82C3C1F399EFD'
…
ike V=root:0:ikev2_native:7: responder preparing EAP identity request
ike V=root:0:ikev2_native:7: local cert, subject='vpn.fortiad.local', issuer='fortiad-WIN-EMS-
CA'
ike V=root:0:ikev2_native:7: local CA cert, subject='fortiad-WIN-EMS-CA', issuer='fortiad-WIN-
EMS-CA'
ike V=root:0:ikev2_native:7: splitting payload len=2759 into 3 fragments
…
ike V=root:0:ikev2_native:7: sent IKE msg (AUTH_RESPONSE): 203.0.113.249:4500-
>198.51.100.3:4500, len=1124, vrf=0, id=1b7426f242a51912/63966ac9bbf0227d:00000001, oif=5
…
ike V=root:0:ikev2_native:7: responder received EAP msg
ike V=root:0:ikev2_native:7: send EAP message to FNBAM
ike V=root:0:ikev2_native:7: initiating EAP authentication
ike V=root:0:ikev2_native: EAP user "tsmith@fortiad.info"
ike V=root:0:ikev2_native: auth candidate group 'rad_mis' 9
ike V=root:0:ikev2_native: auth candidate group 'rad_finance' 10
…
ike V=root:0:ikev2_native:7 EAP 128797701255176 result FNBAM_CHALLENGED
ike V=root:0:ikev2_native: EAP challenged for user "tsmith@fortiad.info"
ike V=root:0:ikev2_native:7: responder preparing EAP pass through message
…
[672] __group_match-Use 'tsmith@fortiad.info' for user group matching.
[710] __group_match-Check if win-10.88.0.1 is a group member
[716] __group_match-Group 'rad_mis' passed group matching
[719] __group_match-Add matched group 'rad_mis'(9)
[710] __group_match-Check if win-10.88.0.1 is a group member
[269] find_matched_usr_grps-Passed group matching
[1012] update_auth_token_session-config does not require 2fa
[357] fnbamd_comm_send_result-Sending result 0 (nid 0) for req 128797701255176, len=2838
[2048] handle_child_rsp-Auth rsp 128797701255176, session created, line 36
ike V=root:0:ikev2_native:7 EAP 128797701255176 result FNBAM_SUCCESS
ike V=root:0:ikev2_native: user 'tsmith@fortiad.info' authenticated group 'rad_mis' 9
ike V=root:0:ikev2_native:7: responder preparing EAP pass through message
…
ike V=root:0:ikev2_native:7: responder received AUTH msg
```

```
ike V=root:0:ikev2_native:7: peer identifier IPV4_ADDR 198.51.100.3
ike V=root:0:ikev2_native:7: eap-peer=yes
ike V=root:0:ikev2_native:7: auth verify done
ike V=root:0:ikev2_native:7: responder AUTH continuation
ike V=root:0:ikev2_native:7: authentication succeeded
ike V=root:0:ikev2_native:7: responder creating new child
ike V=root:0:ikev2_native:7: mode-cfg type 1 request 0:''
ike V=root:0:ikev2_native: mode-cfg allocate 10.130.1.1/0.0.0.0
ike V=root:0:ikev2_native:7: mode-cfg using allocated IPv4 10.130.1.1
…
ike V=root:0:ikev2_native:7:ikev2_native:38: comparing selectors
ike V=root:0:ikev2_native:7:ikev2_native:38: matched by rfc-rule-3
ike V=root:0:ikev2_native:7:ikev2_native:38: phase2 matched by subset
ike V=root:0:ikev2_native:7:ikev2_native:38: using mode-cfg override 0:10.130.1.1-10.130.1.1:0
…
ike V=root:0:ikev2_native_0:7:ikev2_native:38: added IPsec SA: SPIs=e996330c/449bb83a
ike V=root:0:ikev2_native_0:7:ikev2_native:38: sending SNMP tunnel UP trap
ike V=root:0:ikev2_native_0: tunnel up event assigned address 10.130.1.1
ike V=root:0:ikev2_native_0: user 'tsmith@fortiad.info' 10.130.1.1 groups 1
```

Configure the Windows server

The Windows server includes AD-CS, a RADIUS server, and a DNS server.

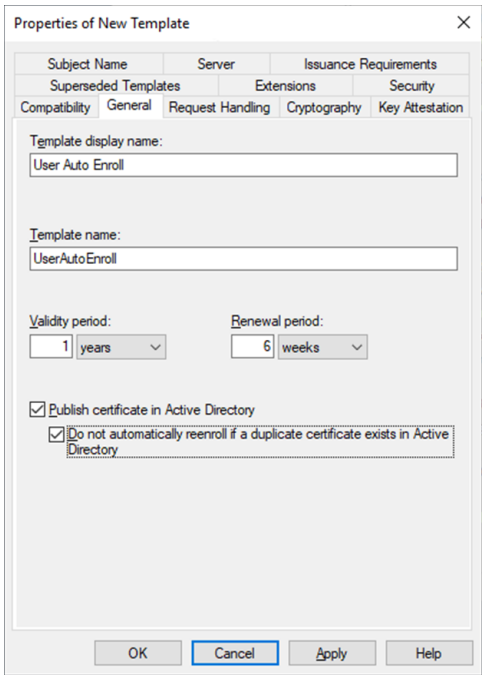
After the AD CS role has been installed and configured, the CA is ready to sign certificates.

Users and groups are defined first. The groups are configured to automatically receive certificates and relay membership to the FortiGate for granular access control through group matching in policies.

RADIUS is used to authorize connecting users. The RADIUS server returns users' groups with the access-accept response, to indicate to the FortiGate what groups the users belong to.

To create a certificate template to enable automatic enrollment for the user groups:

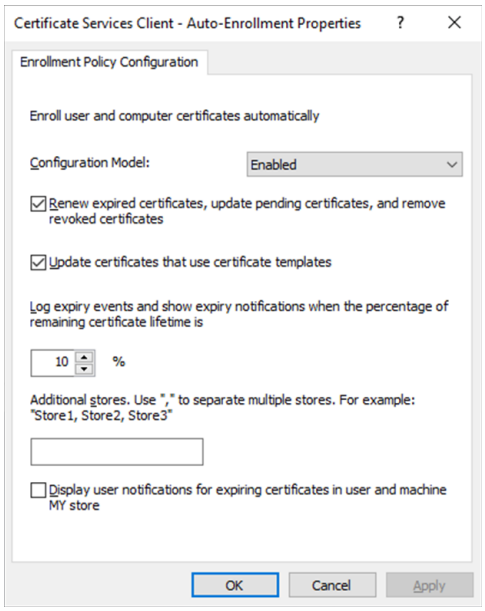
1. Open *Certification Authority*.
2. In the navigation pane, expand the new CA, right-click *Certificate Template* and click *Manage*.
3. Configure a new certificate template:
   a. Right-click the *User* template and click *Duplicate Template*.
   b. On the *General* tab, enter a *Template display name*, such as *User Auto Enroll*.
   c. Enable *Publish certificate in Active Directory* and *Do not automatically reenroll...*.



   d. Configure the remaining settings as required, then go to the *Request Handling* tab.
   e. Disable *Allow private key to be exported* and select *Enroll subject without requiring any user input*.
   f. On the *Security* tab, in *Group or user name*, click *Add*.
   g. Add AD groups where users needs to enroll.
   h. Select each group and, under *Permissions*, enable *Read*, *Enroll*, and *Autoenroll*.
   i. On the *Extensions* tab, click *Application Policies* then click *Edit*.
   j. Remove all of the policies expect for *Client Authentication*.
   k. Click *OK* then close the *Certificate Templates* console.
4. In the navigation pane, right-click *Certificate Template* and click *New > Certificate Template to Issue*.
5. Select the new certificate template, *User Auto Enroll*, then click *OK*.

To create a group policy to enable automatic enrollment:

1. Open the *Group Policy Management* console.
2. In the navigation pane, go to *Forest:fortiad.info > Domains > fortiad.info*, and then click *Group Policy Objects*.
3. Click *Action*, and then click *New*.
4. Set a *Name* for the new GPO then click *OK*.
5. Right-click the new GPO and click *Edit*.
6. In the *Group Policy Management Editor* navigation pane, go to *User configuration > Policies > Windows Settings > Security Settings > Public Key Policies*.
7. In the content pane, double-click *Certificate Services Client - Auto-Enrollment*.
8. Set *Configuration Model* to *Enabled*.
9. Enable *Renew expired certificates...* and *Update certificates...*.



10. Click *OK*.

To verify that users are receiving certificates:

1. Log into an endpoint with a domain user.
2. On the server, open Certification Authority.
3. Expand the CA and select *Issued Certificates*.
4. Verify that the user logged into the endpoint is listed under *Requested Name*. You can also check the local user certificate store on the endpoint.

To generate and sign a CSR and import the signed certificate to the FortiGate:

1. On the FortiGate and go to *System* > *Certificates* and click *Create/Import* > *Generate CSR*.
2. Configure the CSR:

Field	Value
Certificate Name	vpn.fortiad.local
ID Type	Domain Name
Domain Name	vpn.fortiad.local
Subject Alternative Name	DNS:vpn.fortiad.local

3. Configure the remaining settings as required, then click *OK*.
4. Download the CSR to a location that is accessible to the CA server.
5. Sign the CSR with the previously created CA:
   a. Open the command prompt as an administrator and enter the following:

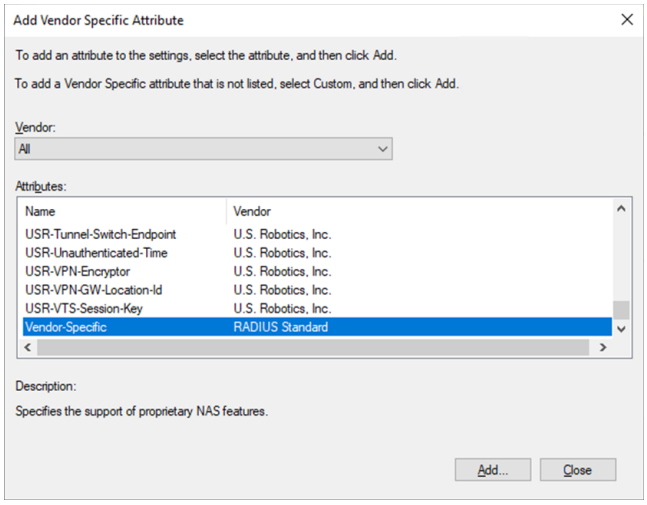
```
certreq -submit -attrib "CertificateTemplate:WebServer" C:\CSR\vpn.fortiad.local.csr
```

   The *Certification Authority List* window opens.

   b. Select the CA and click *OK*.
   c. Save the signed certificate with a .cer file extension to a location that is accessible from the FortiGate.
6. Import the signed certificate to the FortiGate:
   a. On the FortiGate, go to *System* > *Certificates* and click *Create/Import* > *Certificate*.
   b. Click *Import Certificate*.
   c. Set *Type* to *Local Certificate*.
   d. Click *Upload* and locate and select the signed certificate
   e. Click *Create* then click *OK*.

To configure network policies on the RADIUS server:

1. Open the *Network Policy Server* and, in the console tree, expand *Policies*.
2. Right-click on *Network Policies* and click *New*.
3. Enter a *Policy name*, such as *VPN-group-MIS*, then click *Next*.
4. Under *Condition description* click *Add*:
   a. Select *User Groups*, then click *Add*.
   b. Click *Add Groups*.
   c. Enter the group name, *MIS*, click *Check Names* to confirm the group.
   d. Click *OK* in both windows.
5. Click *Next*.
6. Make sure that *Access granted* is selected, then click *Next*.
7. On the *Configure Authentication Methods* page, click *Add* and add the EAP type *Microsoft: Smart Care or other certificate*.
8. Edit the EAP type, select the previously generated certificate, then click *OK*.
9. Deselect all of the *Less secure authentication methods* then click *Next*.
10. Configure constraints as needed, then click *Next*.
11. On the *Configure Settings* page, under *RADIUS Attributes*, select *Vendor Specific*, then click *Add*:
    a. In the *Attributes* list, select *Vendor-Specific*, then click *Add*.

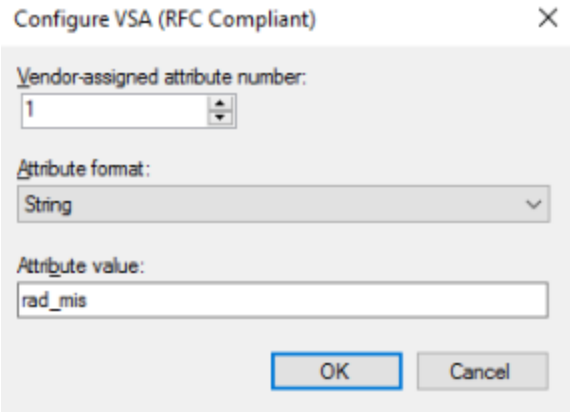


    b. In the *Attribute Information* window, click *Add*.

c. In the *Vendor-Specific Attribute Information* window, enter the *Vendor Code*, *12356*, and select *Yes. It conforms*.

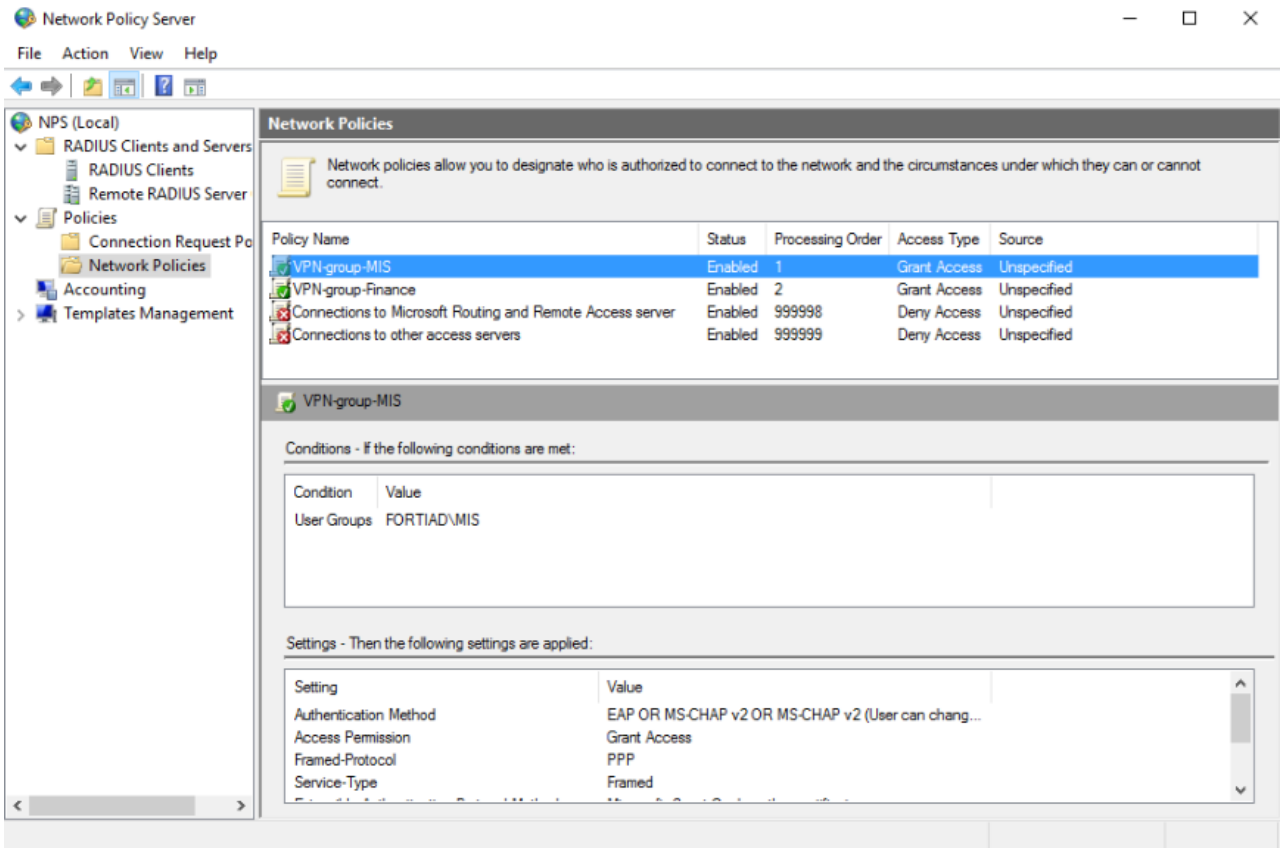
d. Click *Configure Attribute* and configure the following:

Field	Value
Vendor-assigned attribute number	1
Attribute format	String
Attribute value	<Group Name returned to FortiGate>



e. Click *OK* on all three windows and on the *Add Vendor Specific Attribute* window click *Close*.

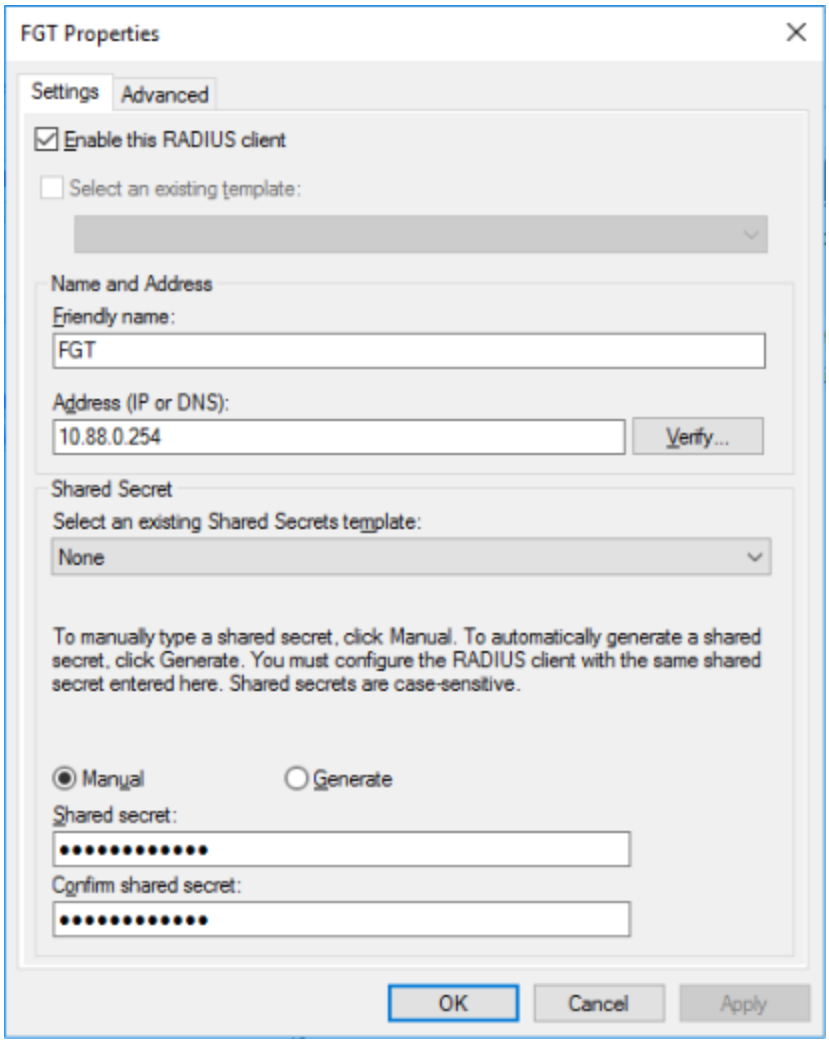
12. Click *Next*.
13. On the *Completing New Network Policy* page, review the configuration, then click *Finish*.
14. Duplicate the policy for *Finance*, and call the new policy *VPN-group-Finance*.
15. Reorder the policies so that the two groups are one and two in the processing order.



To add the FortiGate as a RADIUS client:

1. Open the *Network Policy Server* and, in the console tree, expand *RADIUS Clients and Servers*.
2. Right-click on *RADIUS Clients* and click *New*.
3. Add the FortiGate as a RADIUS client:

Field	Value
Friendly name	FGT
Address	10.88.0.254
Shared Secret	Manually enter the shared secret.



4. Click *OK*.

To create a DNS entry for the VPN connection:

1. Open the *DNS Manager*.
2. Go to *DC > Forward Lookup Zones* and select *lab.local*.
3. Right click in the content pane and select *New Host (A or AAAA)*.
4. Enter the VPN name. The FQDN should be auto-filled with *vpn.fortiad.local*.
5. Enter an IP address (203.0.113.249).
6. Click *Add Host*.

Dialup VPN with certificate authentication

In a dialup VPN setup, a company may choose to use X.509 certificates as their authentication solution for remote users. This method verifies the remote user using a user certificate instead of a username and password. This method can be simpler for end users and more secure as it uniquely identifies each user.

Administrators need to issue unique user certificates to each user for remote access management. The user certificate can be verified by one of three methods: the subject field, common name, or the subject identity such as the user principle name in the Subject Alternative Name (SAN) field.

Subject field verification

This is the basic method that verifies the subject string defined in the PKI user setting matches a substring in the subject field of the user certificate. For example:

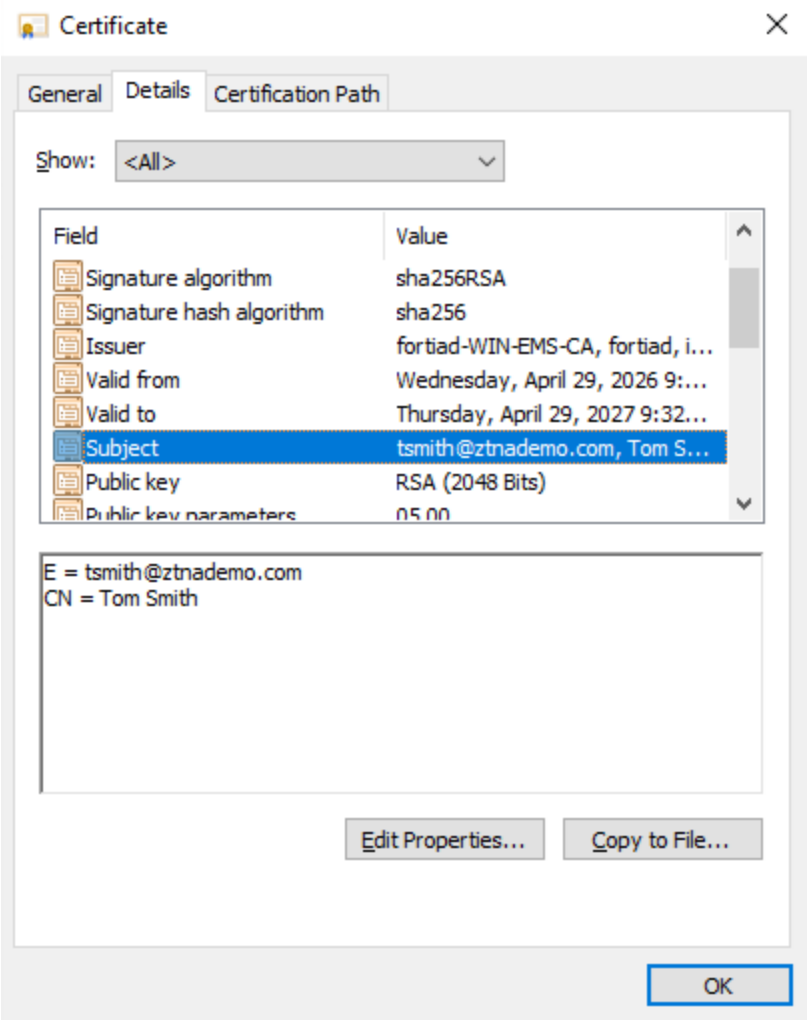
```
config user peer
    edit "tsmith"
        set ca "CA_Cert_1"
        set subject "CN=Tom Smith"
    next
end
```

Common name verification

In this method, administrators can define the CN string to match the common name (CN) in the subject field of the certificate. For example:

```
config user peer
    edit "tsmith"
        set ca "CA_Cert_1"
        set cn "Tom Smith"
    next
end
```

The matching certificate looks like the following:



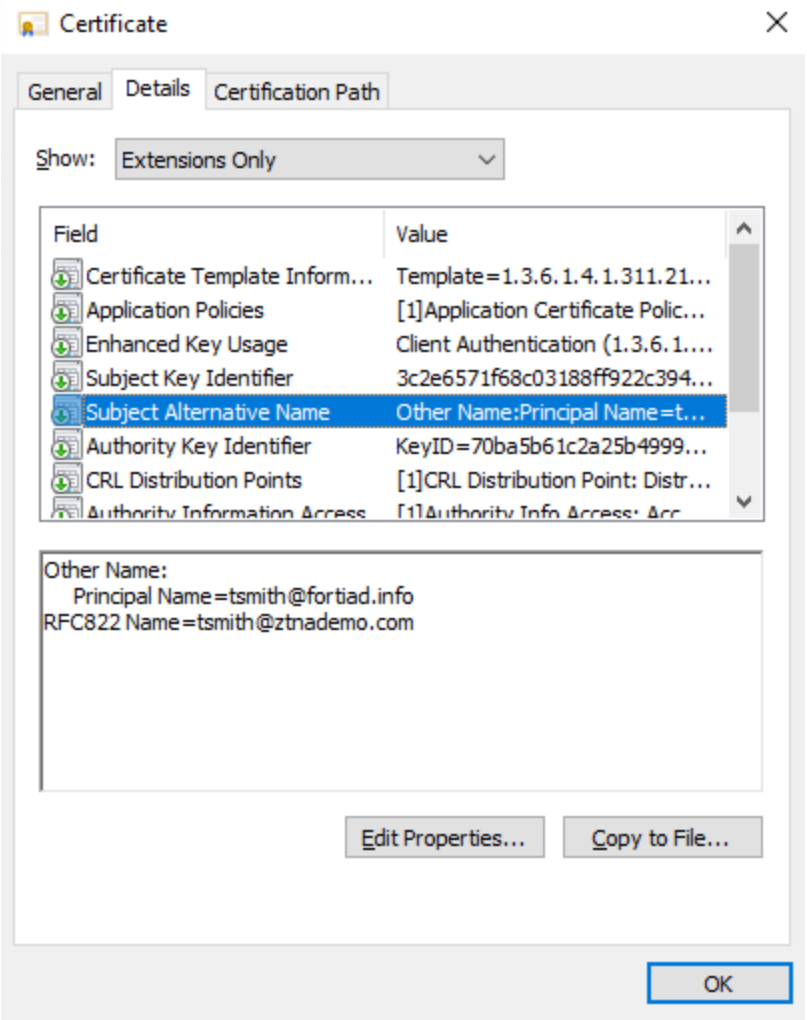
A PKI user must be created on the FortiGate for each remote user that connects to the VPN with a unique user certificate.

Principal name with LDAP integration

In this method, the PKI user setting references an LDAP server. When `mfa-mode` is set to `subject-identity`, the UPN in the user certificate's SAN field is used to look up the user in the LDAP directory. If a match is found, then authentication succeeds. For example:

```
config user peer
    edit "ldap-peer"
        set ca "CA_Cert_1"
        set mfa-mode subject-identity
        set mfa-server "LDAP-fortiad"
    next
end
```

The matching certificate looks like the following:



This method is more scalable because only one PKI user needs to be created on the FortiGate. Remote users connect with their unique user certificate that are matched against users in the LDAP server.

The `two-factor` and `passwd` settings in the user peer configuration are not supported by IPsec. If enabled, the setting will be ignored.

```
config user peer
    edit <user>
        set two-factor enable
        set passwd **********
    next
end
```

Certificate management

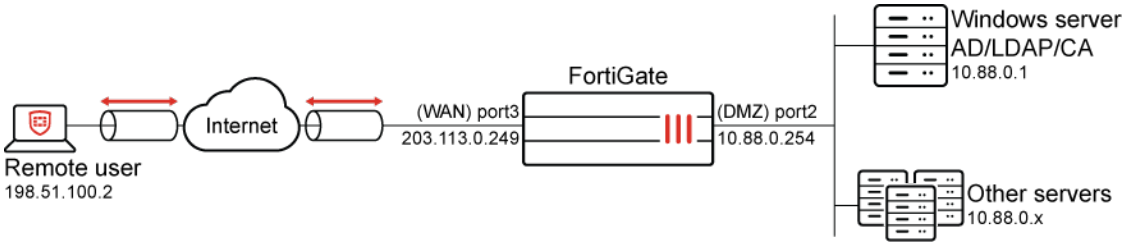
Dialup IPsec VPN with certificate authentication requires careful certificate management planning. Assuming that a company's private certificate authority (CA) is used to generate and sign all the certificates, the following

certificates are needed:

Certificate type	Description
Server certificate	The server certificate is used to identify the FortiGate IPsec dialup gateway. A CSR can be generated on the FortiGate and signed by the CA, or the CA can generate the private and public keys and export the certificate package to the FortiGate.
User certificate	The user certificate is generated and signed by the CA with unique CNs in the subject field and/or unique Principal Names in the SAN field. They are used to identify the user that is connecting to the VPN. User certificates must be installed on client machines. They should have an Enhanced Key Usage with Client Authentication.
CA certificate	The root CA certificate, and any subordinate CA that signed the actual user and server certificates, must be imported into the FortiGate and client machines. The CA certificate is used to verify the certificate chain of the server and user certificates.

Example

In this example, a dialup IPsec VPN tunnel is configured with certificate authentication using the subject field verification method and the LDAP integration method.



The company CA, named root CA, signs all the server and user certificates. The user, tsmith@fortiad.info, has a user certificate signed by root CA installed on their endpoint. The corresponding user account is also present under the company's Active Directory.

This example shows VPN configurations in IKEv2. Starting with FortiClient 7.4.4, IKEv1 is no longer supported on the client. Therefore, plan accordingly when choosing your IKE version. Use IKEv2 if you plan on deploying FortiClient 7.4.4 and later.

There are five major steps to configure this example:

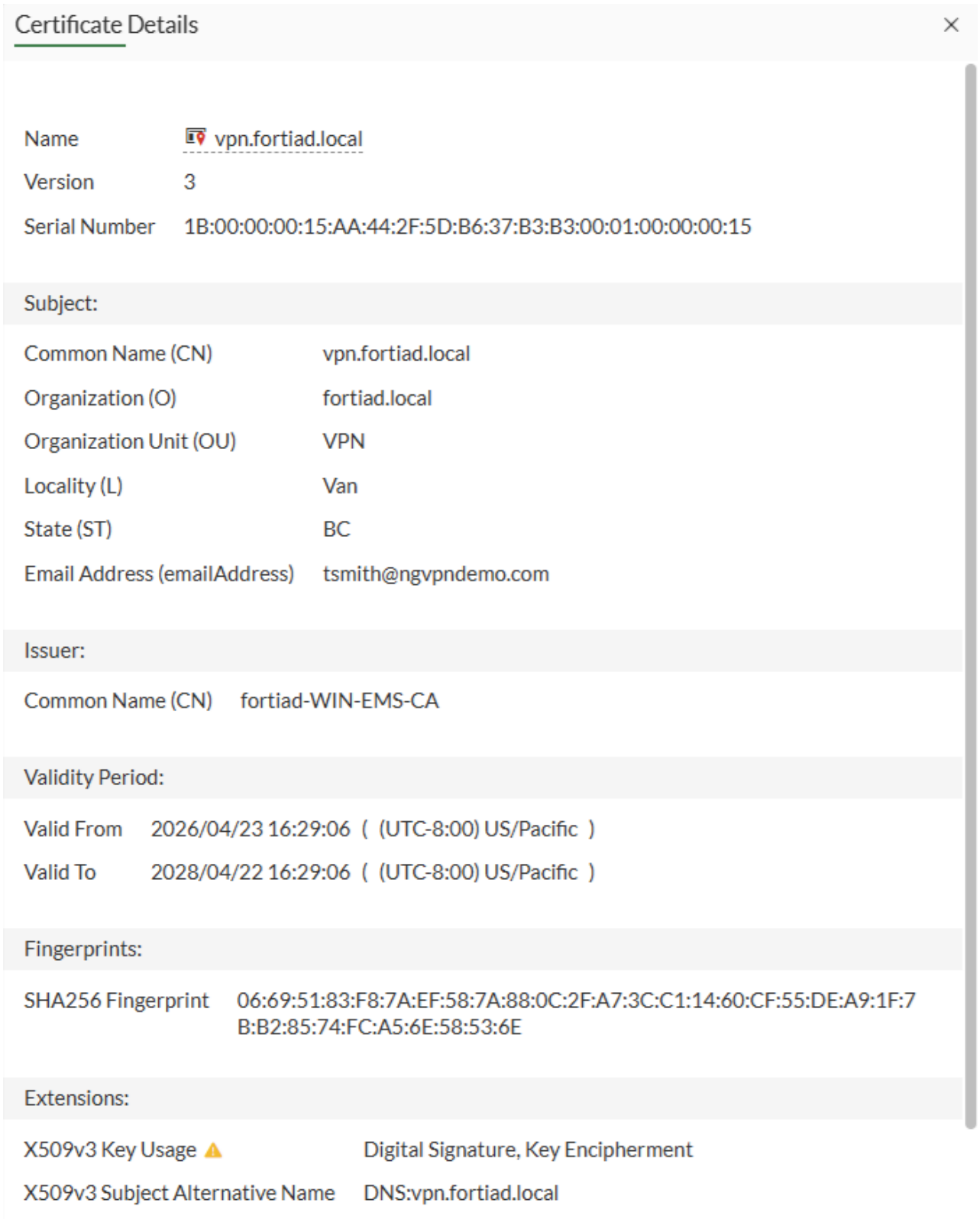
1. Importing the certificates
2. Configuring user authentication
3. Configuring the VPN
4. Configuring FortiClient and the endpoints
5. Testing and verifying the certificate authentication

Importing the certificates

The server certificate and CA certificate need to be imported into the FortiGate.

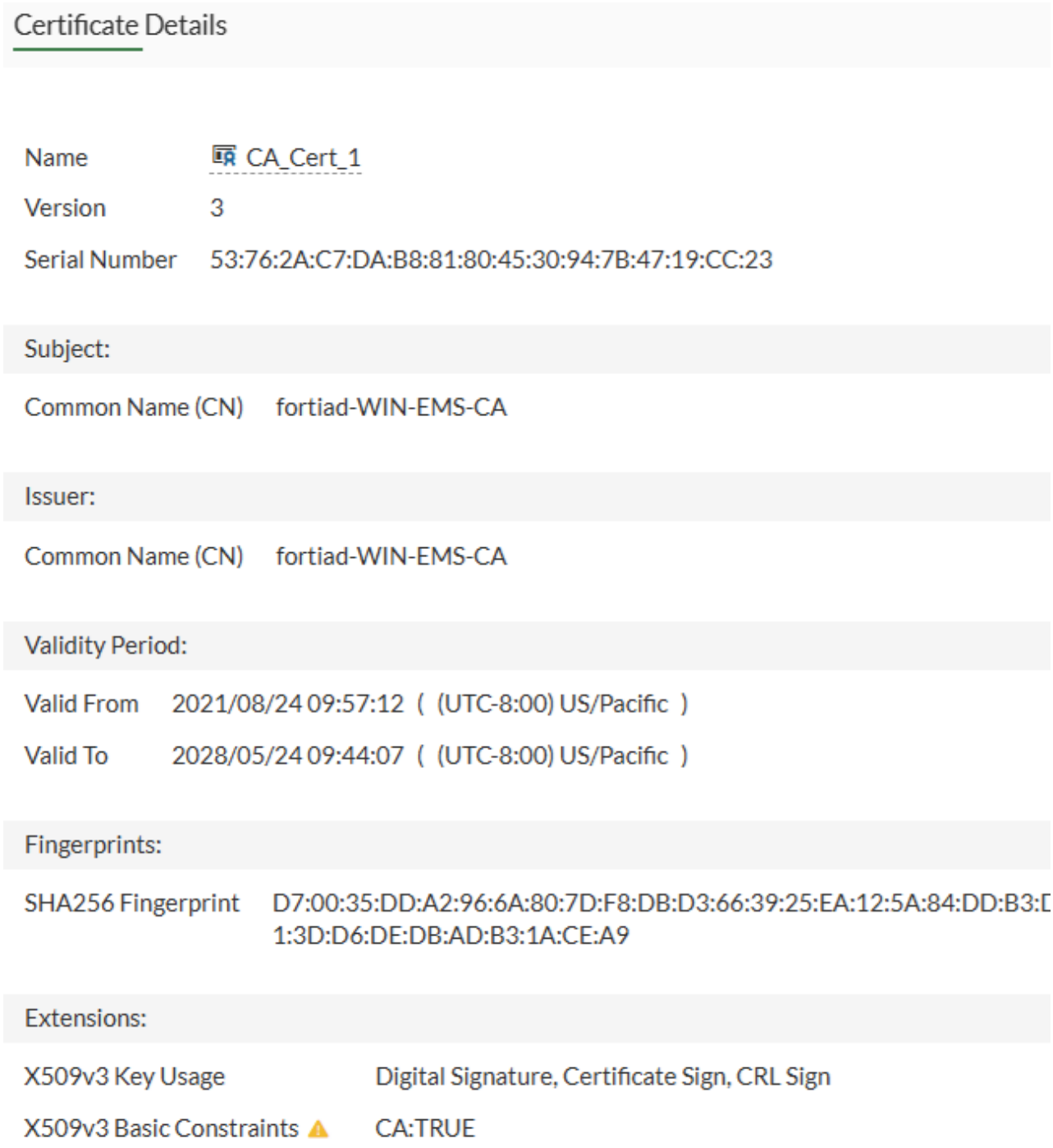
To import the server certificate:

1. Go to *System* > *Certificates* and select *Import* > *Local Certificate*.
2. For *Type*, select *PKCS #12 Certificate*.
3. Upload the key file exported from the CA and enter the password.
4. Click *OK*. The certificate now appears in the *Local Certificate* section.


Certificate Details
Name
vpn.fortiad.local
Version
3
Serial Number
1B:00:00:00:15:AA:44:2F:5D:B6:37:B3:B3:00:01:00:00:00:15
Subject:
Common Name (CN)
vpn.fortiad.local
Organization (O)
fortiad.local
Organization Unit (OU)
VPN
Locality (L)
Van
State (ST)
BC
Email Address (emailAddress)
tsmith@ngvpndemo.com
Issuer:
Common Name (CN)
fortiad-WIN-EMS-CA
Validity Period:
Valid From
2026/04/23 16:29:06 ((UTC-8:00) US/Pacific)
Valid To
2028/04/22 16:29:06 ((UTC-8:00) US/Pacific)
Fingerprints:
SHA256 Fingerprint
06:69:51:83:F8:7A:EF:58:7A:88:0C:2F:A7:3C:C1:14:60:CF:55:DE:A9:1F:7
B:B2:85:74:FC:A5:6E:58:53:6E
Extensions:
X509v3 Key Usage
Digital Signature, Key Encipherment
X509v3 Subject Alternative Name
DNS:vpn.fortiad.local

To import the CA certificate:

1. Go to *System* > *Certificates* and select *Import* > *CA Certificate*.
2. For *Type*, select *File*.
3. Upload the CA certificate (usually a .CRT file). This certificate only contains the public key.
4. Click *OK*. The certificate now appears in the *Remote CA Certificate* section.



If any subordinate CA is involved in signing the certificates, you need to import its certificate.

Configuring user authentication

FortiGate PKI users do not appear in the GUI until at least one PKI user has been created in the CLI. The following instructions create the PKI users in the CLI.

To configure PKI users for subject field verification:

1. Create the PKI user and choose the CA certificate that was imported (if the certificate was signed by a subordinate CA, choose the subordinate CA's certificate):

```
config user peer
    edit "tsmith"
        set ca "CA_Cert_1"
        set subject "CN=Tom Smith"
    next
end
```

   For an example of CN field matching, see Common name verification.
2. Create additional users as needed.
3. Place the users into a peer group:

```
config user peergrp
    edit "pki-users"
        set member "tsmith" <user> ... <user>
    next
end
```

To configure PKI users for LDAP integration:

1. Configure the LDAP server that users connect to for authentication:

```
config user ldap
    edit "LDAP-fortiad"
        set server "10.88.0.1"
        set cnid "sAMAccountName"
        set dn "dc=fortiad,dc=info"
        set type regular
        set username "fortiad\\Administrator"
        set password ************
        set secure ldaps
        set ca-cert "CA_Cert_1"
        set port 636
    next
end
```

2. Configure the PKI user to reference the LDAP server using the CA certificate that was imported:

```
config user peer
    edit "ldap-peer"
        set ca "CA_Cert_1"
        set mfa-mode subject-identity
        set mfa-server "LDAP-fortiad"
    next
end
```

3. Place the user into a peer group:

```
config user peergrp
    edit "pki-ldap"
        set member "ldap-peer"
    next
end
```

Configuring the VPN

To configure the tunnel, use the custom IPsec tunnel configuration method.

To configure the remote access VPN:

1. Go to *VPN* > *VPN Tunnels*.
2. Click *Create new* > *Custom IPsec tunnel*.
3. Enter the following *Network* configurations:

Field	Value
Name	dialup_cert
IP version	IPv4
Remote gateway	Dialup user
Interface	port3
Mode config	Enabled. Select IPv4
Assign client IP	Enabled. Select Range
IPv4 client IP range	172.18.200.10-172.18.200.99
IPv4 subnet mask	255.255.255.255
Certificate name	Choose your server certificate. In this example, vpn.fortiad.local
DNS	Use System DNS
IPv4 split tunnel	Enable if you do not want Internet traffic routed through this tunell. Select the subnet range which will be routed to this tunnel.

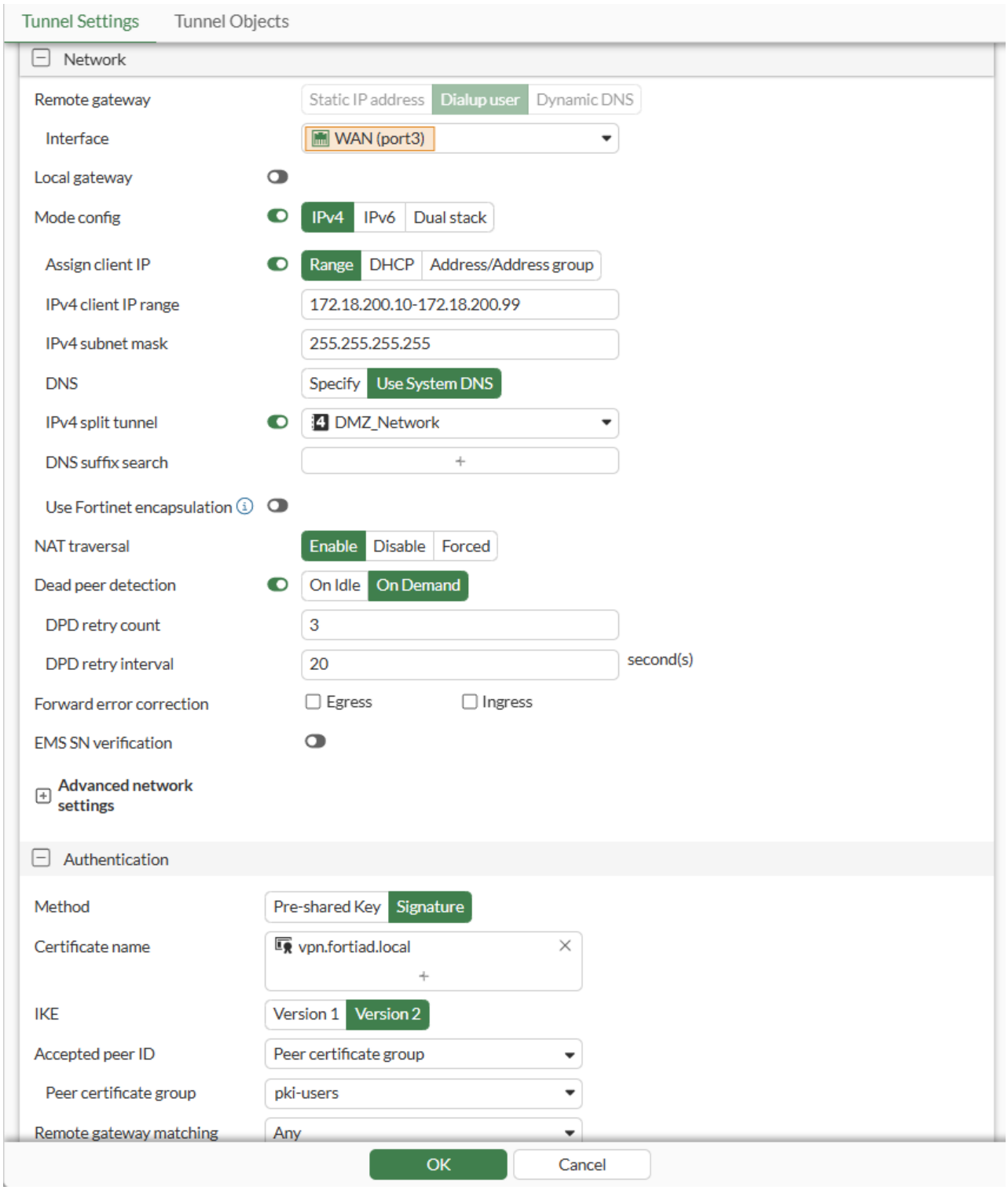
4. Enter the following *Authentication* configurations:

Field	Value
Method	Signature
Certificate name	Choose your server certificate, in this example, *vpn.fortiad.local*.
IKE	Version 2
Accepted peer ID	Peer certificate group
Peer certificate group	Choose the PKI peer group based on the method you chose. For subject field verification, select *pki-users*. For LDAP integration, select *pki-ldap*.
EAP	Disabled.

5. For Phase 1 proposal, use the default or select your custom settings.
6. For Phase 2 selectors, click *Create new*:

Field	Value
Name	dialup_cert
Local address	Subnet Address. Enter the address of the local subnet, in this case *10.88.0.0 255.255.255.255*.

7. Configure the remaining as needed then click *OK* to save.
8. Review the tunnel settings:


Tunnel Settings
Tunnel Objects
Network
Remote gateway
Static IP address
Dialup user
Dynamic DNS
Interface
WAN (port3)
Local gateway
Mode config
IPv4
IPv6
Dual stack
Assign client IP
Range
DHCP
Address/Address group
IPv4 client IP range
172.18.200.10-172.18.200.99
IPv4 subnet mask
255.255.255.255
DNS
Specify
Use System DNS
IPv4 split tunnel
DMZ_Network
DNS suffix search
Use Fortinet encapsulation
NAT traversal
Enable
Disable
Forced
Dead peer detection
On Idle
On Demand
DPD retry count
3
DPD retry interval
20
second(s)
Forward error correction
Egress
Ingress
EMS SN verification
Advanced network settings
Authentication
Method
Pre-shared Key
Signature
Certificate name
vpn.fortiad.local
IKE
Version 1
Version 2
Accepted peer ID
Peer certificate group
Peer certificate group
pki-users
Remote gateway matching
Any
OK
Cancel

To configure the address objects:

1. Create the address range for the dialup clients:
   a. Go to *Policy & Objects* > *Addresses* and select *Address*.
   b. Click *Create new*.
   c. For *Name*, enter *dialup_cert_client_range* .
   d. For *Type*, select *IP Range* and enter *172.18.200.10-172.18.200.99* in the *IP Range* field.
   e. Click *OK*.
2. Create the address subnet for the destination 10.88.0.0/24:
   a. Click *Create new*.
   b. For *Name*, enter *DMZ_Network*.
   c. For *Type*, select *Subnet* and enter *10.88.0.0/24* in the *IP/Netmask* field.
   d. Click *OK*.

To configure the firewall policy:

1. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
2. Configure the following:

Field	Value
Name	dialup_cert_in
Schedule	always
Action	ACCEPT
Incoming interface	dialup_cert
Outgoing Interface	port2
Source	dialup_cert_client_range
Destination	DMZ_Network
Service	ALL
NAT	Enabled

3. Configure the other settings as needed.
4. Click *OK*.

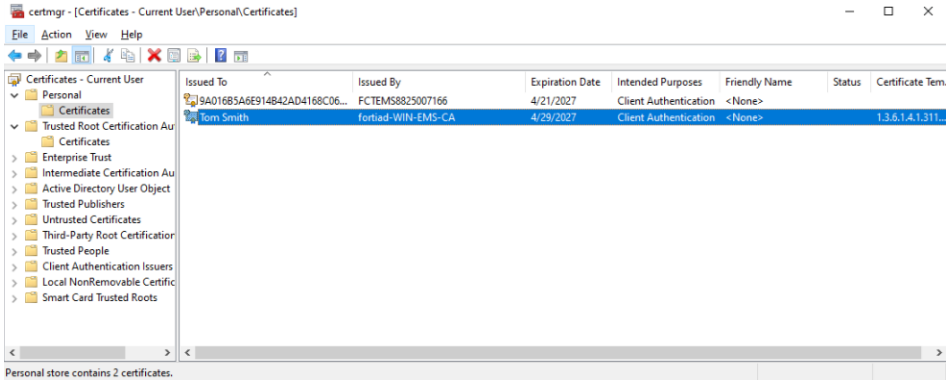
Configuring FortiClient and the endpoints

The following example is configured on a Windows PC with FortiClient 7.4.4. The VPN tunnel is directly configured on the client. However, for mass deployments, it is recommended to use EMS.

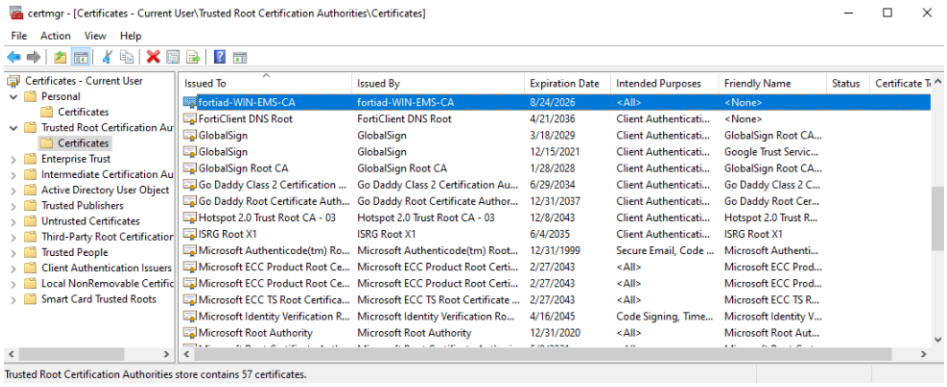
The user certificate and CA certificate must be installed on the endpoint device. They may be pushed by the administrator through group policies or another method. This example assumes that the user certificate and CA certificate are already installed on the endpoint.

To verify the user and CA certificates:

1. Open the Windows certificate manager (certmgr):
   a. In the Control Panel, type *Manage user certificate* in the search box.
   b. Click the result, *Manage user certificates*.
2. Go to *Personal* > *Certificate*. The user certificate should be listed.



3. Go to *Trusted Root Certification Authorities* > *Certificates*. The company CA certificate should be listed.

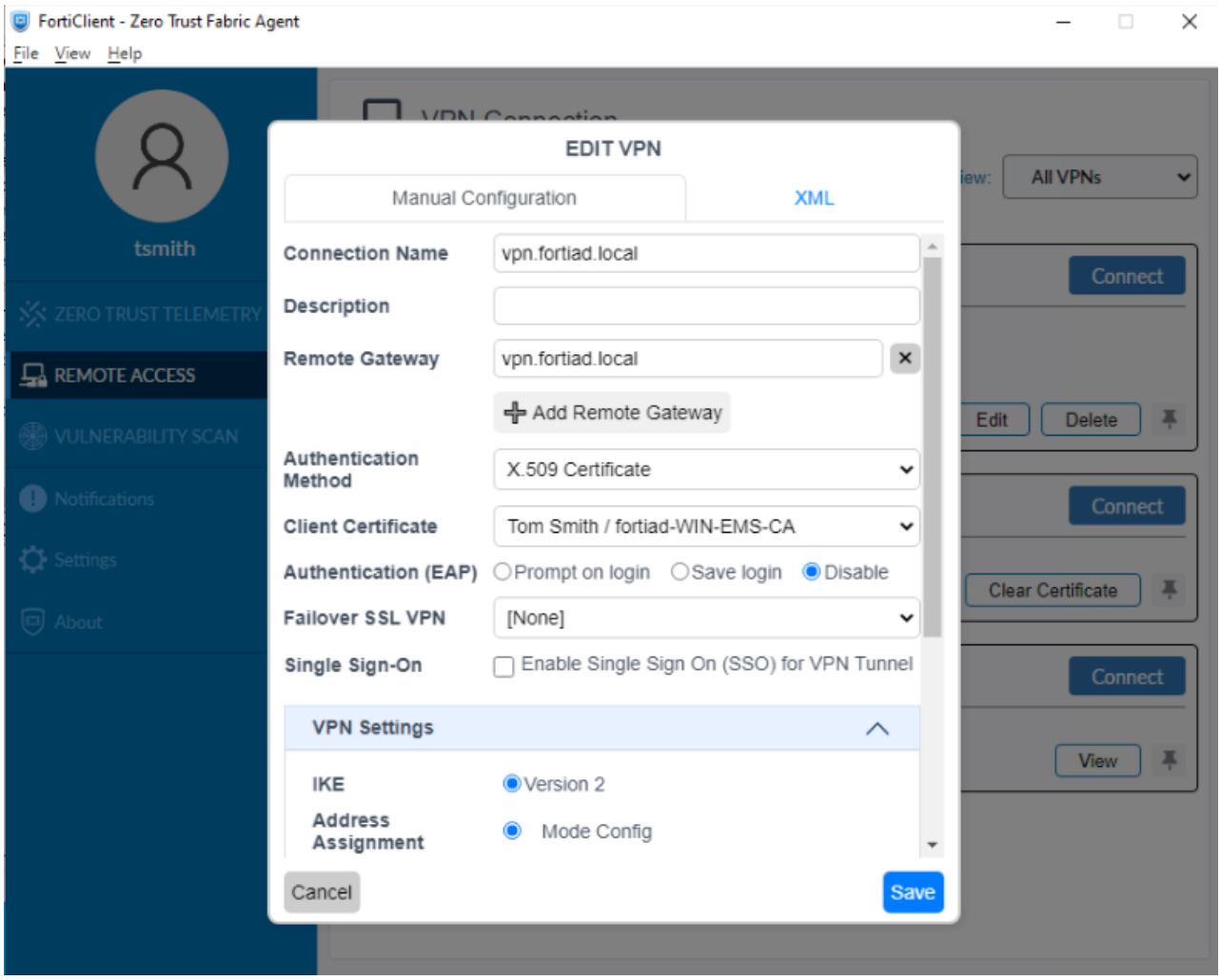


To configure the FortiClient endpoint settings:

1. In FortiClient, click the *Remote Access* tab and add a new connection.
2. Configure the following:

Field	Value
Connection Name	vpn.fortiad.local
Remote Gateway	vpn.fortiad.local (This should resolve to 203.113.0.249)
Authentication Method	X.509 Certificate
Client Certificate	Select the user certificate from the Certificate Store.
Authentication (EAP)	Disable

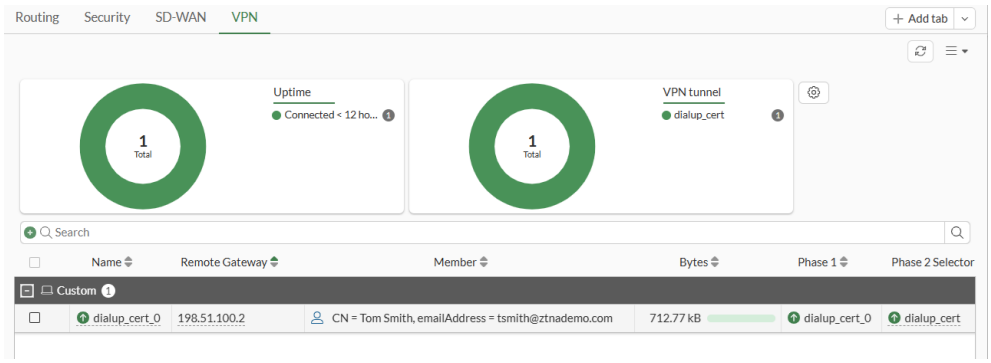
3. For other VPN settings, you can leave as default or update to match your FortiGate VPN settings.



4. Click *Save*.

Testing and verifying the certificate authentication

1. On the client PC, open FortiClient and click the *Remote Access* tab.
2. Select the VPN tunnel, *vpn.fortiad.local*, and click *Connect*.
   If the connection is successful, the tunnel will flash Connected and then minimize. When you open the *Remote Access* window, it now displays *VPN Connected* and the associated VPN tunnel details.
3. On the FortiGate, go to *Dashboard* > *Network Monitor* > *VPN* to view the VPN tunnel monitor.
   The widget displays tunnel information, including the *Peer ID* containing the subject field of the user certificate.



4. Go to *Log & Report > System Events* and switch to *VPN Events* from the drop-down. Several tunnel related logs are recorded.
5. The same logs can be viewed in the CLI:

```
# execute log filter category 1
# execute log filter field subtype vpn
# execute log display
…
1: date=2026-05-01 time=09:07:17 eventtime=1777651637248046212 tz="-0700" logid="0101037141"
type="event" subtype="vpn" level="notice" vd="root" logdesc="IPsec tunnel statistics"
msg="IPsec tunnel statistics" action="tunnel-stats" remip=198.51.100.2 locip=203.0.113.249
remport=50740 locport=4500 outintf="port3" srccountry="Reserved"
cookies="7db260751a851715/6ecbb03d925502b2" user="CN = Tom Smith, emailAddress =
tsmith@ztnademo.com" group="pki-users" useralt="CN = Tom Smith, emailAddress =
tsmith@ztnademo.com" eapuser="N/A" eapauthgroup="N/A" assignip=172.18.200.10
vpntunnel="dialup_cert_0" tunnelip=172.18.200.10 tunnelid=244014039 tunneltype="ipsec"
duration=31218 sentbyte=0 rcvdbyte=714964 nextstat=600
fctuid="9A016B5A6E914B42AD4168C066EB04CA" advpnsc=0
```

6. To view tunnel info, go to the command palette (CTRL+P) > CLI diagnostics and look for `ike gateway` list to see the detail tunnel information:

```
vd: root/0
name: dialup_cert_0
version: 2
interface: port3 5
addr: 203.0.113.249:4500 -> 198.51.100.2:50740
tun_id: 172.18.200.10/::10.0.0.6
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 31508s ago
peer-id: CN = Tom Smith, emailAddress = tsmith@ztnademo.com
peer-id-auth: yes
FortiClient UID: 9A016B5A6E914B42AD4168C066EB04CA
assigned IPv4 address: 172.18.200.10/255.255.255.255
nat: me peer
pending-queue: 0
PPK: no
IKE SA: created 1/1  established 1/1  time 400/400/400 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms
```

```
  id/spi: 5 7db260751a851715/6ecbb03d925502b2
  direction: responder
  status: established 31508-31507s ago = 400ms
  proposal: aes256-sha256
  child: no
  SK_ei: 04ef0646caca17d4-97649b02b2af7afc-63ab28adde368e9a-ba7b4104f4af99a1
  SK_er: 88b93d53c4171875-c8b4a34ec08766c4-0864cac0c9f19d62-c01e15d4228fd5a6
  SK_ai: a8e30642d7887e0f-772a8f8d0895b185-9921013fcdbe3c4d-26a5f794bd4d7258
  SK_ar: d92ac300d86c8cdc-6bb9ec2b86c1412e-46ccb2c49458461a-da618ac5e2b5deca
  PPK: no
  message-id sent/recv: 0/2
  QKD: no
  PQC-KEM (IKE): no
  PQC-KEM (all IPsec): no
  lifetime/rekey: 86400/54622
  DPD sent/recv: 00000000/00000000
  peer-id: CN = Tom Smith, emailAddress = tsmith@ztnademo.com
  peer-group: pki-users
```

7. If any issues arise during the connection, run the following debug commands to troubleshoot the issue:

```
# diagnose debug application ike -1
# diagnose debug application fnbamd -1
# diagnose debug enable
```

Successful connection should look similar to this:

```
ike V=root:0: udp comes 198.51.100.2:55434->203.0.113.249:4500,ifindex=5,vrf=0,len=466....
ike V=root:0: IKEv2 exchange=SA_INIT id=312cb44890199231/0000000000000000 len=462
ike 0: in …
 chosen, matched gateway dialup_cert
ike V=root:0:dialup_cert:dialup_cert: created connection: 0x55dd379857f0 5 203.0.113.249-
>198.51.100.2:55434.
ike V=root:0:dialup_cert:7: processing notify type NAT_DETECTION_SOURCE_IP
ike V=root:0:dialup_cert:7: processing NAT-D payload
ike V=root:0:dialup_cert:7: NAT detected: PEER
ike V=root:0:dialup_cert:7: process NAT-D
ike V=root:0:dialup_cert:7: processing notify type NAT_DETECTION_DESTINATION_IP
ike V=root:0:dialup_cert:7: processing NAT-D payload
ike V=root:0:dialup_cert:7: NAT detected: ME PEER
ike V=root:0:dialup_cert:7: process NAT-D
ike V=root:0:dialup_cert:7: processing notify type SIGNATURE_HASH_ALGORITHMS
ike V=root:0:dialup_cert:7: FEC vendor ID received FEC but IP not set
ike 0:dialup_cert:7: FCT EAP 2FA extension vendor ID received
ike V=root:0:dialup_cert:7: responder preparing SA_INIT msg
ike V=root:0:dialup_cert:7: create NAT-D hash local 203.0.113.249/4500 remote
198.51.100.2/55434
ike V=root:0:dialup_cert:7: sending CERTREQ payload (len=21)
ike V=root:0:dialup_cert:7: certreq[0]: '000E16E711D247159D734740B429AEC35543F80E'
ike 0:dialup_cert:7: out
…
```

```
ike V=root:0:dialup_cert:7: sent IKE msg (SA_INIT_RESPONSE): 203.0.113.249:4500-
>198.51.100.2:55434, len=301, vrf=0, id=312cb44890199231/816a1bc34346d2e3, oif=5
…
ike V=root:0: udp comes 198.51.100.2:55434->203.0.113.249:4500,ifindex=5,vrf=0,len=2468....
ike V=root:0: IKEv2 exchange=AUTH id=312cb44890199231/816a1bc34346d2e3:00000001 len=2464
ike 0: in
…
ike V=root:0:dialup_cert:7: responder received AUTH msg
ike V=root:0:dialup_cert:7: processing notify type INITIAL_CONTACT
ike V=root:0:dialup_cert:7: received peer identifier DER_ASN1_DN 'CN = Tom Smith, emailAddress
= tsmith@ztnademo.com'
ike V=root:0:dialup_cert:7: eap-peer=no
ike V=root:0:dialup_cert:7: match gw peer ID by FNBAM
ike V=root:0:dialup_cert:7: Validating X.509 certificate
ike V=root:0:dialup_cert:7: peer cert, subject='Tom Smith', issuer='fortiad-WIN-EMS-CA'
ike V=root:0:dialup_cert:7: peer ID verified
ike V=root:0:dialup_cert:7: building fnbam peer candidate list
ike V=root:0:dialup_cert:7: FNBAM_GROUP_NAME candidate 'pki-users'
ike V=root:0:dialup_cert:7: certificate validation pending
…
[554] fnbamd_cert_verify-Following cert chain depth 0
[628] fnbamd_cert_verify-Issuer found: CA_Cert_1 (SSL_DPI opt 1)
[554] fnbamd_cert_verify-Following cert chain depth 1
[1057] __cert_verify-peer_info.no_ocsp_query:0 cert->status:640.
[769] fnbamd_cert_check_group_list-checking group with name 'pki-users'
[581] __check_add_peer-check 'tsmith'
[440] peer_san_subject_check-SAN: host_good 1; email_good 1; ip_good 1
[454] peer_subject_cn_check-Cert subject 'CN = Tom Smith, emailAddress = tsmith@ztnademo.com'
[349] __RDN_match-Checking 'CN' val 'Tom Smith' -- match.
[378] __cert_subject_RDN_compare-Total matched RDNs in cert: 1
[479] peer_subject_cn_check-Subject is good.
[589] __check_add_peer-'tsmith' check ret:good
[704] __peer_user_clear_unmatched-Clear all user(s) other than 'tsmith'
[724] __peer_user_clear_unmatched-
[202] __get_default_ocsp_ctx-def_ocsp_ctx=(nil), no_ocsp_query=0, ocsp_enabled=0
[841] fnbamd_cert_check_group_list-Peer users
[848] fnbamd_cert_check_group_list-    'tsmith' ('N/A','N/A','N/A')
[1069] __cert_verify_do_next-req_id=9711170703366
[109] __cert_chg_st- 'Validation' -> 'Done'
[1163] __cert_done-req_id=9711170703366
[1742] fnbamd_auth_session_done-Session done, id=9711170703366
[1209] __fnbamd_cert_auth_run-Exit, req_id=9711170703366
[1786] create_auth_cert_session-fnbamd_cert_auth_init returns 0, id=9711170703366
[1698] auth_cert_success-id=9711170703366
[1321] fnbamd_cert_auth_copy_cert_status-req_id=9711170703366
[1330] fnbamd_cert_auth_copy_cert_status-Matched peer user 'tsmith'
[969] fnbamd_cert_check_matched_groups-checking group with name 'pki-users', peer_ctx->peer_
user->setting.name:tsmith
[1033] fnbamd_cert_check_matched_groups-matched
[1362] fnbamd_cert_auth_copy_cert_status-Leaf cert status is unchecked.
[1459] fnbamd_cert_auth_copy_cert_status-Cert st 2c0, req_id=9711170703366
```

```
[357] fnbamd_comm_send_result-Sending result 0 (nid 672) for req 9711170703366, len=2787
[2048] handle_child_rsp-Auth rsp 9711170703366, session not created, line 85
ike V=root:0:dialup_cert:7: fnbam reply 'pki-users'
ike V=root:0:dialup_cert:7: fnbam matched peergrp 'pki-users'
ike V=root:0:dialup_cert:7: certificate validation succeeded
ike V=root:0:dialup_cert:7: signature verification succeeded
ike V=root:0:dialup_cert:7: auth verify done
ike V=root:0:dialup_cert:7: responder AUTH continuation
ike V=root:0:dialup_cert:7: authentication succeeded
ike V=root:0:dialup_cert:7: processing notify type FORTICLIENT_CONNECT
ike V=root:0:dialup_cert:7: received FCT data len = 288, data = 'VER=1
FCTVER=7.4.4.1887
UID=9A016B5A6E914B42AD4168C066EB04CA
IP=172.16.7.3
MAC=02-09-0f-00-03-03;02-09-0f-00-04-03;
HOST=WIN10-01
USER=tsmith
OSVER=Microsoft Windows 8.0 Professional Edition, 64-bit (build 9200)
REG_STATUS=0
EMSSN=FCTEMS8825007166
EMSID=00000000000000000000000000000000
'
…
ike V=root:0:dialup_cert:7:dialup_cert:11: phase2 matched by intersection
ike V=root:0:dialup_cert:7:dialup_cert:11: using mode-cfg override 0:172.18.200.10-
172.18.200.10:0
…
ike V=root:0:dialup_cert:7: responder preparing AUTH msg
ike V=root:0:dialup_cert: adding new dynamic tunnel for 198.51.100.2:55434
ike V=root:0:dialup_cert_0: tunnel created tun_id 172.18.200.10/::10.0.0.7 remote_location
0.0.0.0
…
ike V=root:0:dialup_cert_0:7: established IKE SA 312cb44890199231/816a1bc34346d2e3
…
ike V=root:0:dialup_cert_0:7:dialup_cert:11: add dynamic IPsec SA selectors 682
ike V=root:0:dialup_cert_0:7:dialup_cert:11: added dynamic IPsec SA proxyids new 1 682
ike V=root:0:dialup_cert:11: add route 172.18.200.10/255.255.255.255 gw 172.18.200.10 oif
dialup_cert(13) metric 15 priority 1
ike V=root:0:dialup_cert_0:7:dialup_cert:11: tunnel 1 of VDOM limit 0/0
ike V=root:0:dialup_cert_0:7:dialup_cert:11: add IPsec SA: SPIs=4bac9b08/f4f33fa6
…
ike V=root:0:dialup_cert_0:7:dialup_cert:11: added IPsec SA: SPIs=4bac9b08/f4f33fa6
ike V=root:0:dialup_cert_0:7:dialup_cert:11: sending SNMP tunnel UP trap
ike V=root:0:dialup_cert_0: tunnel up event assigned address 172.18.200.10
ike V=root:0:dialup_cert_0: sent tunnel-up message to EMS: (fct-
uid=9A016B5A6E914B42AD4168C066EB04CA, intf=dialup_cert_0, addr=172.18.200.10, vdom=root)
```

Alternatively, if LDAP peer user is used, the user authentication will look slightly different:

```
ike V=root:0:dialup_cert:8: Validating X.509 certificate
ike V=root:0:dialup_cert:8: peer cert, subject='Tom Smith', issuer='fortiad-WIN-EMS-CA'
ike V=root:0:dialup_cert:8: peer ID verified
```

```
ike V=root:0:dialup_cert:8: building fnbam peer candidate list
ike V=root:0:dialup_cert:8: FNBAM_GROUP_NAME candidate 'pki-ldap'
ike V=root:0:dialup_cert:8: certificate validation pending
…
[554] fnbamd_cert_verify-Following cert chain depth 0
[628] fnbamd_cert_verify-Issuer found: CA_Cert_1 (SSL_DPI opt 1)
[554] fnbamd_cert_verify-Following cert chain depth 1
[1057] __cert_verify-peer_info.no_ocsp_query:0 cert->status:640.
[769] fnbamd_cert_check_group_list-checking group with name 'pki-ldap'
[581] __check_add_peer-check 'ldap-peer'
[440] peer_san_subject_check-SAN: host_good 1; email_good 1; ip_good 1
[454] peer_subject_cn_check-Cert subject 'CN = Tom Smith, emailAddress = tsmith@ztnademo.com'
[84] fnbamd_peer_ldap_push-Check LDAP setting of peer user 'ldap-peer'
[351] fnbamd_ldap_get-vfid=0, name='LDAP-fortiad'
[589] __check_add_peer-'ldap-peer' check ret:pending
[802] fnbamd_cert_check_group_list-LDAP servers
[808] fnbamd_cert_check_group_list-    'LDAP-fortiad', (Principle-Name), ref=2
[202] __get_default_ocsp_ctx-def_ocsp_ctx=(nil), no_ocsp_query=0, ocsp_enabled=0
[841] fnbamd_cert_check_group_list-Peer users
[848] fnbamd_cert_check_group_list-    'ldap-peer' ('LDAP-fortiad','N/A','N/A')
..
[771] __get_san_value-Obtained GEN_OTHERNAME 'tsmith@fortiad.info'
[472] __cert_ldap_query-cert_field_val: 'tsmith@fortiad.info'
[1899] fnbamd_ldap_auth_ctx_init-User: tsmith@fortiad.info, password query: 0, group list
query: 0, group only: 0, UPN query: 1, user domain query: 0
…
[1482] __ldap_tcps_connect-tcps_connect(10.88.0.1) is established. Current state: Connecting.
[1161] __ldap_auth_ctx_reset-
[1004] __ldap_next_state-State: Connecting -> Admin Binding
…
[1004] __ldap_next_state-State: Admin Binding -> DN Search
[1443] __ldap_rxtx-Start ldap conn timer.
[1341] __ldap_rxtx-fd 14, state 4(DN Search)
[1342] __ldap_rxtx-Stop ldap conn timer.
[1349] __ldap_rxtx-
[902] fnbamd_ldap_build_dn_search_req-base:'dc=fortiad,dc=info' filter:(&
(userPrincipalName=tsmith@fortiad.info)(!(UserAccountControl:1.2.840.113556.1.4.803:=2)))
[1273] fnbamd_ldap_send-sending 146 bytes to 10.88.0.1
…
[1500] fnbamd_ldap_recv-Response len: 16, svr: 10.88.0.1
[1176] fnbamd_ldap_parse_response-Got one MESSAGE. ID:2, type:search-result
[1211] fnbamd_ldap_parse_response-ret=0
[806] __ldap_membership_next-Auth accepted
[1161] __ldap_auth_ctx_reset-
[1004] __ldap_next_state-State: DN Search -> Done
[403] __cert_ldap_query_cb-LDAP ret=0, server='LDAP-fortiad', req_id=9711170703367
[414] __cert_ldap_query_cb-Matched peer 'ldap-peer'
…
[1330] fnbamd_cert_auth_copy_cert_status-Matched peer user 'ldap-peer'
[969] fnbamd_cert_check_matched_groups-checking group with name 'pki-ldap', peer_ctx->peer_
user->setting.name:ldap-peer
```

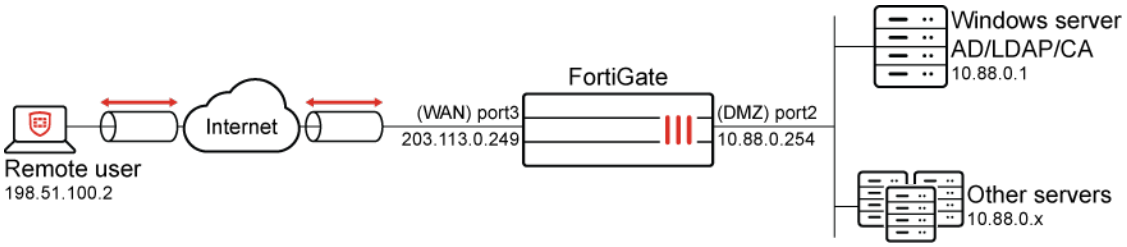
```
[1033] fnbamd_cert_check_matched_groups-matched
…
ike V=root:0:dialup_cert:8: fnbam reply 'pki-ldap'
ike V=root:0:dialup_cert:8: fnbam matched peergrp 'pki-ldap'
ike V=root:0:dialup_cert:8: certificate validation succeeded
ike V=root:0:dialup_cert:8: signature verification succeeded
ike V=root:0:dialup_cert:8: auth verify done
ike V=root:0:dialup_cert:8: responder AUTH continuation
ike V=root:0:dialup_cert:8: authentication succeeded
```

Dialup VPN with certificate and LDAP authentication

The previous example demonstrated how to use a client certificate as the primary authentication method for connecting to an IKEv2 remote access VPN. This example extends on that by utilizing EAP to authentication the client using username and password against the same Active Directory.

The topology and configurations remain largely the same. We will use the Principal name with LDAP integration verification method for the user certificate.

```
config user peer
    edit "ldap-peer"
        set ca "CA_Cert_1"
        set mfa-mode subject-identity
        set mfa-server "LDAP-fortiad"
    next
end
```

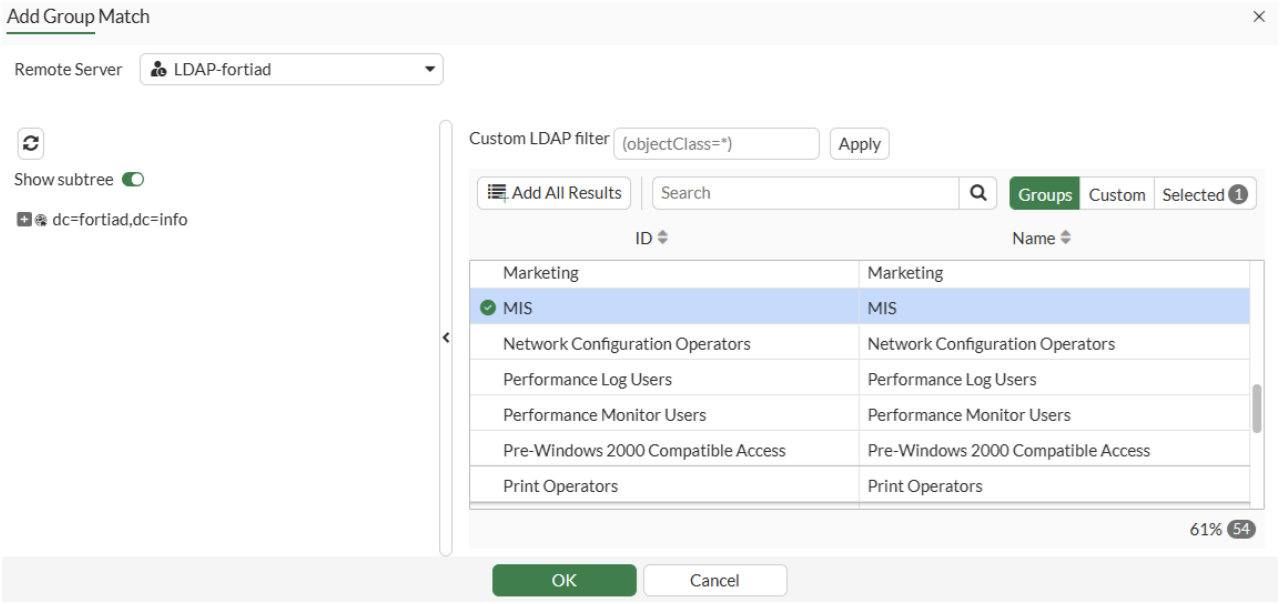


This example extends on the previous configurations with the addition of an LDAP user group for the user Tom Smith, that belongs to the MIS group on Active Directory.

Additionally, since the FortiClient endpoint must use EAP-TTLS for user authentication, it is important to configure the endpoint VPN settings on FortiClient EMS. The Remote Access endpoint profile settings will be pushed out to all connected FortiClient endpoints.

To configure an LDAP user group and assign it to the VPN:

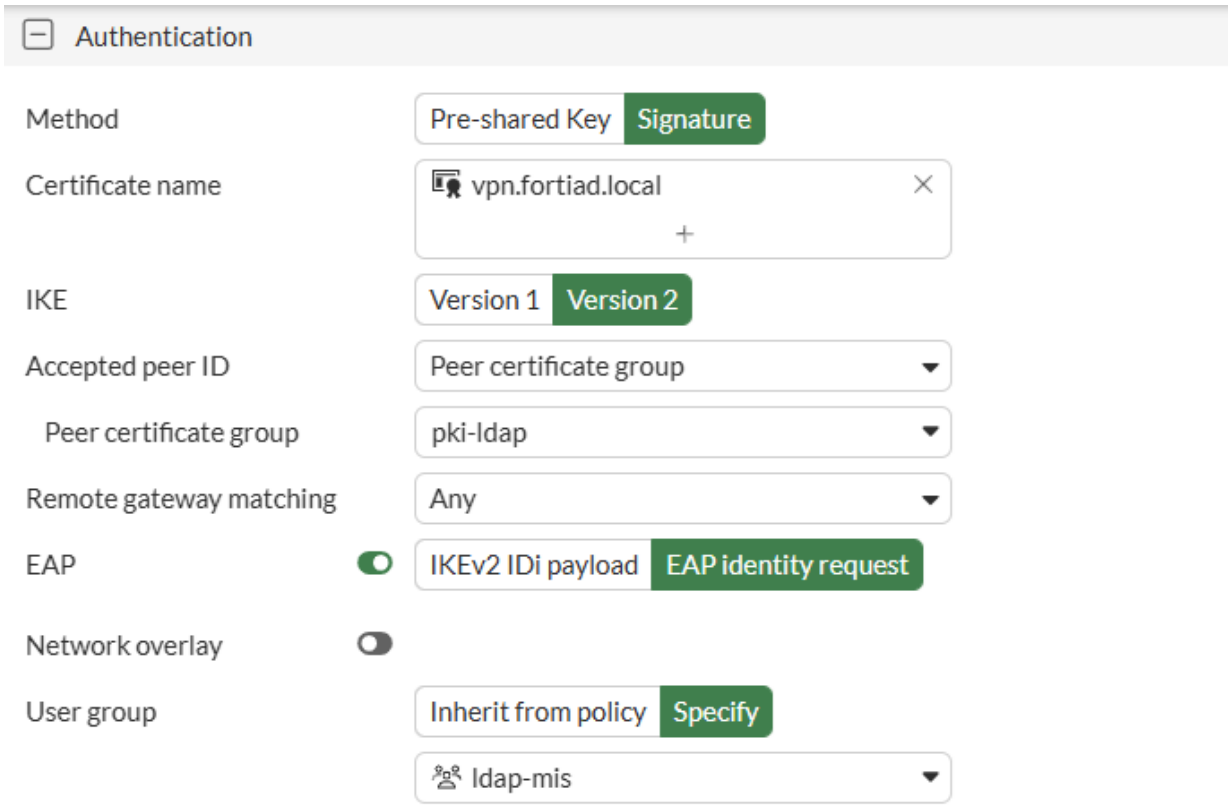
1. On the FortiGate, go to *User & Authentication* > *User Groups* and click *Create new*.
2. Set the name to *ldap-mis*.
3. Under *Remote Groups*, click *+Add*.
4. For the *Remote Server*, select *LDAP-fortiad*.
5. When the groups are displayed, find the MIS group, right-click and select *Add Selected*.



6. Click *OK*.
7. Click *OK* to save the new user group.

To assign the LDAP user group to the VPN tunnel:

1. Go to *VPN* > *VPN Tunnels*.
2. Edit the *dialup_cert tunnel*.
3. In the *Authentication* section, configure the following:
   a. Enable *EAP* and select *EAP Identity request*.
   b. Set *User group* to *Specify*.
   c. Select the *ldap-mis* user group



4. Click *OK* to save the VPN changes.

To configure a remote access endpoint profile on EMS:

1. On FortiClient EMS, go to *Endpoint Profiles* > *Remote Access*.
2. Edit the *Default* profile.
3. Enable *IPsec VPN*.
   a. Enable *User Windows Store Certificates*.
   b. Enable *Current User Windows Store Certificates*.
4. Under *VPN Tunnels*, click *Add Tunnel*.
5. Select *Manual* and click *Next*.
6. In *Basic settings*, configure the following:

Field	Value
Name	ldap_cert_vpn
Remote Gateway	vpn.fortiad.local
Authentication Method	System Store Certificate
Prompt for Username	Enable

7. Switch to *VPN Settings* and confirm that the *IKE version* is *Version 2* and *Mode config* is selected.
8. Switch to *Phase 1* and edit the following:

Field	Value
IKE Proposal	AES128-SHA256 AES256-SHA256
DH Groups	20
EAP Authentication Method	EAP-TTLS

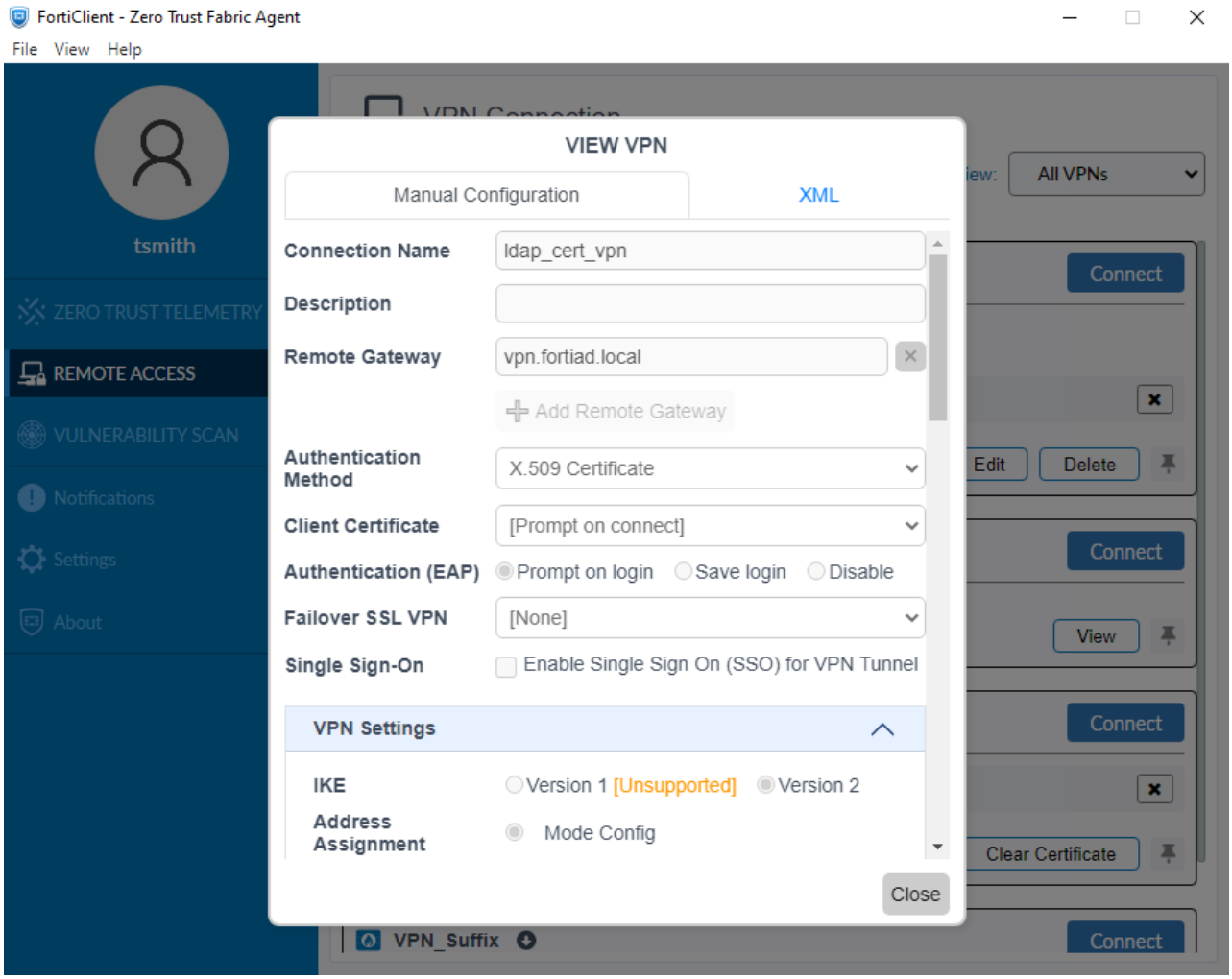
9. Switch to *Phase 2* and update the *IKE Proposal* to *AES128-SHA256* and *AES256-SHA256*.
10. Click *Save* to save the tunnel settings
11. Click *Save* to save the profile settings
12. Under *Endpoint Policy & Components*, ensure that the enabled profile has the Default VPN profile selected. The *Remote Access* profile and VPN tunnel configurations will be pushed to the FortiClient.

Verification

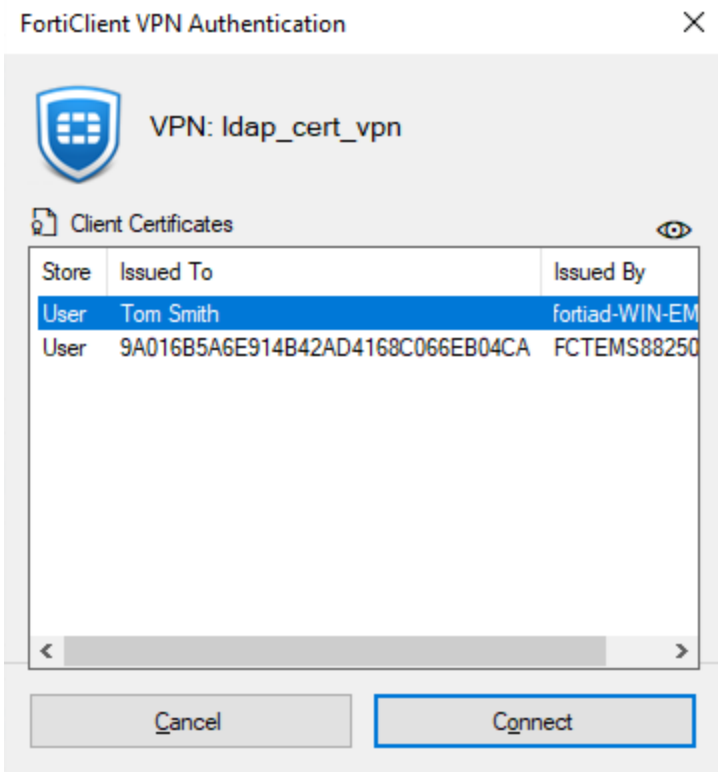
This example is based on using the Windows version of FortiClient 7.4.4. Using other versions of FortiClient may differ slightly.

To connect to the VPN:

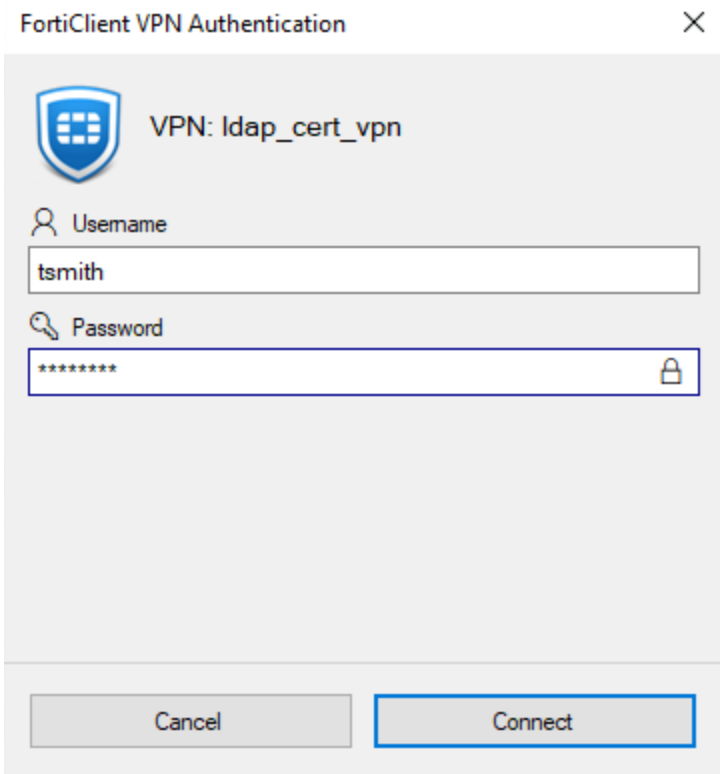
1. Open FortiClient on your endpoint and navigate to the *Remote Access* tab.
2. Find the new *ldap_cert_vpn* tunnel and click *View*.
3. Scroll through the VPN configurations to confirm the details.



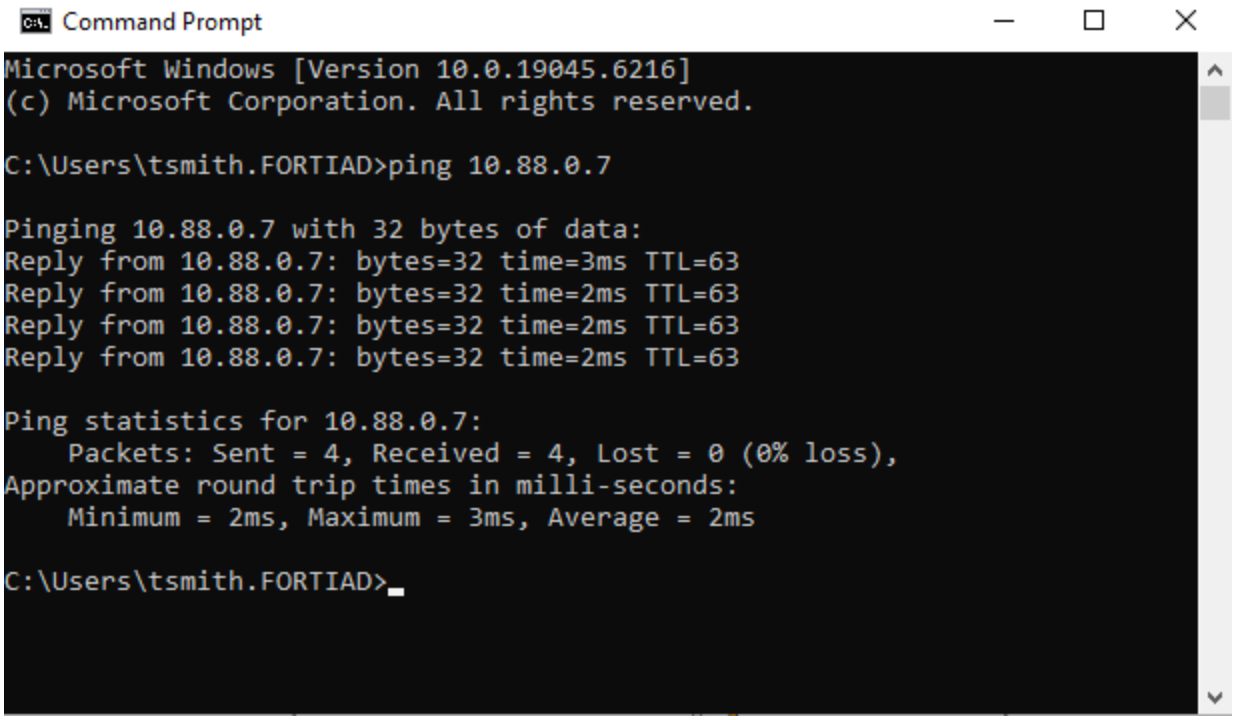
4. Click *Close*.
5. Click *Connect* to start the VPN connection
6. You will first be prompted to select your client certificate



7. Once selected, you will be prompted for the username and password

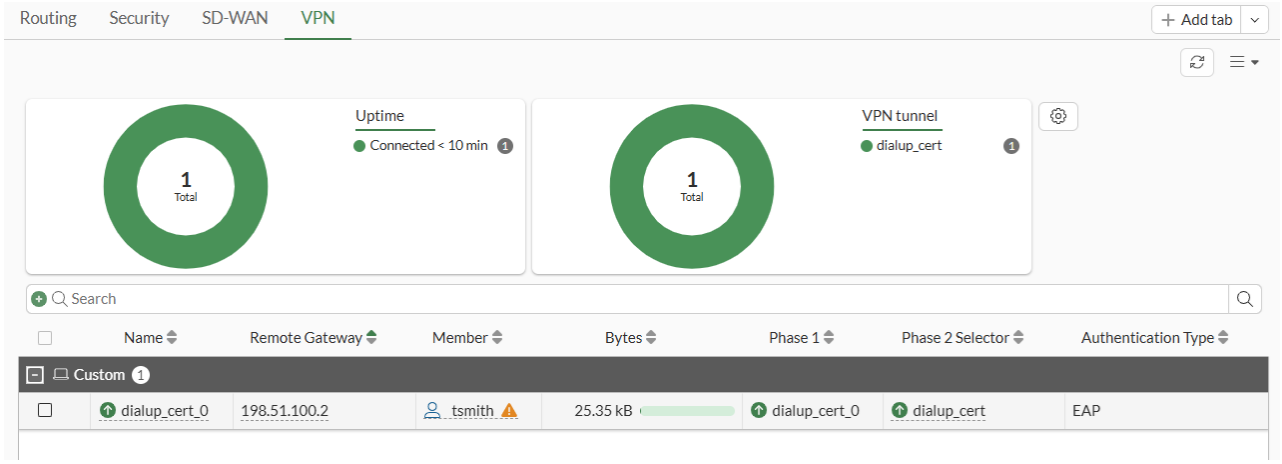


8. After both of these pass, your VPN will be connected.
9. From the command prompt, try to ping a server in the internal network.



To verify from the FortiGate:

1. On the FortiGate, go to Dashboard > Network Monitor > VPN to view the VPN tunnel monitor.
   The widget displays tunnel information and displays username from EAP authentication



2. Go to *Log & Report > System Events* and switch to *VPN Events* from the drop-down. Several tunnel related logs are recorded.
   The same logs can be viewed in the CLI:

```
# execute log filter category 1
# execute log filter field subtype vpn
# execute log display
1: date=2026-05-01 time=11:37:17 eventtime=1777660637248678009 tz="-0700" logid="0101037141"
type="event" subtype="vpn" level="notice" vd="root" logdesc="IPsec tunnel statistics"
```

```
msg="IPsec tunnel statistics" action="tunnel-stats" remip=198.51.100.2 locip=203.0.113.249
remport=62915 locport=4500 outintf="port3" srccountry="Reserved"
cookies="3019f6aa301bd504/49d22338a0eaf8a3" user="CN = Tom Smith, emailAddress =
tsmith@ztnademo.com" group="pki-ldap" useralt="CN = Tom Smith, emailAddress =
tsmith@ztnademo.com" eapuser="tsmith" eapauthgroup="ldap-mis" assignip=172.18.200.10
vpntunnel="dialup_cert_0" tunnelip=172.18.200.10 tunnelid=244014029 tunneltype="ipsec"
duration=335 sentbyte=240 rcvdbyte=25554 nextstat=600
fctuid="9A016B5A6E914B42AD4168C066EB04CA" advpnsc=0
```

3. To view tunnel info, go to the command palette (CTRL+P) > CLI diagnostics and look for `ike gateway` list to see the detail tunnel information:

```
vd: root/0
name: dialup_cert_0
version: 2
interface: port3 5
addr: 203.0.113.249:4500 -> 198.51.100.2:62915
tun_id: 172.18.200.10/::10.0.0.11
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 538s ago
eap-user: tsmith
2FA: no
peer-id: CN = Tom Smith, emailAddress = tsmith@ztnademo.com
peer-id-auth: yes
FortiClient UID: 9A016B5A6E914B42AD4168C066EB04CA
assigned IPv4 address: 172.18.200.10/255.255.255.255
nat: me peer
pending-queue: 0
PPK: no
IKE SA: created 1/1  established 1/1  time 18140/18140/18140 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 18 3019f6aa301bd504/49d22338a0eaf8a3
  direction: responder
  status: established 538-520s ago = 18140ms
  proposal: aes128-sha256
  child: no
  SK_ei: a17ecf0653ffdf34-1ec96e42ce289564
  SK_er: 28cdf2ba3fbf52e6-f447d097ba692e32
  SK_ai: 7319afe842921189-3cd73c44959294dc-f8c25592be553265-3bd47fafc816652e
  SK_ar: 0cb5eaee634df9d9-f7dce1763e9dbb3c-0bbcea6e5c1ef0b0-4bf63141d9af4124
  PPK: no
  message-id sent/recv: 0/62
  QKD: no
  PQC-KEM (IKE): no
  PQC-KEM (all IPsec): no
  lifetime/rekey: 86400/85609
  DPD sent/recv: 00000000/00000000
  peer-id: CN = Tom Smith, emailAddress = tsmith@ztnademo.com
  peer-group: pki-ldap
```

4. If any issues arise during the connection, run the following debug commands to troubleshoot the issue:

```
# diagnose debug application ike -1
# diagnose debug application fnbamd -1
# diagnose debug enable
```

5. Successful connection will have similar FNBAM debugs as these:

```
[554] fnbamd_cert_verify-Following cert chain depth 0
[628] fnbamd_cert_verify-Issuer found: CA_Cert_1 (SSL_DPI opt 1)
[554] fnbamd_cert_verify-Following cert chain depth 1
[1057] __cert_verify-peer_info.no_ocsp_query:0 cert->status:640.
[769] fnbamd_cert_check_group_list-checking group with name 'pki-ldap'
[581] __check_add_peer-check 'ldap-peer'
[440] peer_san_subject_check-SAN: host_good 1; email_good 1; ip_good 1
[454] peer_subject_cn_check-Cert subject 'CN = Tom Smith, emailAddress = tsmith@ztnademo.com'
[84] fnbamd_peer_ldap_push-Check LDAP setting of peer user 'ldap-peer'
[351] fnbamd_ldap_get-vfid=0, name='LDAP-fortiad'
[589] __check_add_peer-'ldap-peer' check ret:pending
[802] fnbamd_cert_check_group_list-LDAP servers
[808] fnbamd_cert_check_group_list-    'LDAP-fortiad', (Principle-Name), ref=2
[202] __get_default_ocsp_ctx-def_ocsp_ctx=(nil), no_ocsp_query=0, ocsp_enabled=0
[841] fnbamd_cert_check_group_list-Peer users
[848] fnbamd_cert_check_group_list-    'ldap-peer' ('LDAP-fortiad','N/A','N/A')
[1069] __cert_verify_do_next-req_id=9711170703385
[109] __cert_chg_st- 'Validation' -> 'Status-Query'
[793] __cert_status_query-req_id=9711170703385
[446] __cert_ldap_query-req_id=9711170703385
[457] __cert_ldap_query-LDAP query, idx 0
[771] __get_san_value-Obtained GEN_OTHERNAME 'tsmith@fortiad.info'
[472] __cert_ldap_query-cert_field_val: 'tsmith@fortiad.info'
[1899] fnbamd_ldap_auth_ctx_init-User: tsmith@fortiad.info, password query: 0, group list
query: 0, group only: 0, UPN query: 1, user
 domain query: 0
…
[1482] __ldap_tcps_connect-tcps_connect(10.88.0.1) is established. Current state: Connecting.
[1161] __ldap_auth_ctx_reset-
[1004] __ldap_next_state-State: Connecting -> Admin Binding
[1496] __ldap_tcps_connect-Start ldap conn timer.
[1341] __ldap_rxtx-fd 14, state 2(Admin Binding)
[1342] __ldap_rxtx-Stop ldap conn timer.
[1349] __ldap_rxtx-
[472] __ldap_build_bind_req-Binding to 'fortiad\Administrator'
…
[1004] __ldap_next_state-State: Admin Binding -> DN Search
[1443] __ldap_rxtx-Start ldap conn timer.
[1341] __ldap_rxtx-fd 14, state 4(DN Search)
[1342] __ldap_rxtx-Stop ldap conn timer.
[1349] __ldap_rxtx-
[902] fnbamd_ldap_build_dn_search_req-base:'dc=fortiad,dc=info' filter:(&
(userPrincipalName=tsmith@fortiad.info)(!(UserAccountControl
:1.2.840.113556.1.4.803:=2)))
…
```

```
[806] __ldap_membership_next-Auth accepted
[1161] __ldap_auth_ctx_reset-
[1004] __ldap_next_state-State: DN Search -> Done
[403] __cert_ldap_query_cb-LDAP ret=0, server='LDAP-fortiad', req_id=9711170703385
[414] __cert_ldap_query_cb-Matched peer 'ldap-peer'
…
[1330] fnbamd_cert_auth_copy_cert_status-Matched peer user 'ldap-peer'
[969] fnbamd_cert_check_matched_groups-checking group with name 'pki-ldap', peer_ctx->peer_
user->setting.name:ldap-peer
[1033] fnbamd_cert_check_matched_groups-matched
…
[2500] handle_req-Rcvd auth req 9711170703386 for tsmith in ldap-mis opt=00000000 prot=7 svc=9
[399] __compose_group_list_from_req-Group 'ldap-mis', type 1
[574] create_auth_session-Session created for req id 9711170703386
[317] radius_start-eap_local=1
[892] fnbamd_cfg_get_radius_list-
[457] fnbamd_rad_get-vfid=0, name='EAP_PROXY'
…
[2500] handle_req-Rcvd auth req 9552247767045 for tsmith in  opt=00000000 prot=0 svc=9
[399] __compose_group_list_from_req-Group 'ldap-mis', type 1
[763] fnbamd_saml_auth_cache_lookup-Authenticating 'tsmith'.
[574] create_auth_session-Session created for req id 9552247767045
[419] auth_local-started for tsmith
…
[1482] __ldap_tcps_connect-tcps_connect(10.88.0.1) is established. Current state: Connecting.
[1161] __ldap_auth_ctx_reset-
[1004] __ldap_next_state-State: Connecting -> Admin Binding
…
[472] __ldap_build_bind_req-Binding to 'fortiad\Administrator'
…
[1004] __ldap_next_state-State: Admin Binding -> DN Search
…
[1004] __ldap_next_state-State: DN Search -> User Binding
[1443] __ldap_rxtx-Start ldap conn timer.
[1341] __ldap_rxtx-fd 16, state 3(User Binding)
[1342] __ldap_rxtx-Stop ldap conn timer.
[1349] __ldap_rxtx-
[708] fnbamd_ldap_build_userbind_req-Trying DN 'CN=Tom Smith,CN=Users,DC=fortiad,DC=info'
[472] __ldap_build_bind_req-Binding to 'CN=Tom Smith,CN=Users,DC=fortiad,DC=info'
…
[1004] __ldap_next_state-State: User Binding -> Attr Query
[1443] __ldap_rxtx-Start ldap conn timer.
[1341] __ldap_rxtx-fd 16, state 5(Attr Query)
[1342] __ldap_rxtx-Stop ldap conn timer.
[1349] __ldap_rxtx-
[763] fnbamd_ldap_build_attr_search_req-Adding attr 'memberOf'
[780] fnbamd_ldap_build_attr_search_req-base:'CN=Tom Smith,CN=Users,DC=fortiad,DC=info'
filter:cn=*
…
[745] __retrieve_group_values- attr='memberOf', found 4 values
[754] __retrieve_group_values-val[0]='CN=MIS,CN=Users,DC=fortiad,DC=info'
```

```
[754] __retrieve_group_values-val[1]='CN=Domain Users,CN=Users,DC=fortiad,DC=info'
[754] __retrieve_group_values-val[2]='CN=Remote Desktop Users,CN=Builtin,DC=fortiad,DC=info'
[754] __retrieve_group_values-val[3]='CN=Administrators,CN=Builtin,DC=fortiad,DC=info'
…
[1004] __ldap_next_state-State: Primary Group Query -> Done
[1997] ldap_copy_grp_list-copied CN=MIS,CN=Users,DC=fortiad,DC=info
[1997] ldap_copy_grp_list-copied CN=Domain Users,CN=Users,DC=fortiad,DC=info
[1997] ldap_copy_grp_list-copied CN=Remote Desktop Users,CN=Builtin,DC=fortiad,DC=info
[1997] ldap_copy_grp_list-copied CN=Administrators,CN=Builtin,DC=fortiad,DC=info
[1997] ldap_copy_grp_list-copied CN=Remote-Allowed,CN=Users,DC=fortiad,DC=info
[781] fnbam_user_auth_group_match-req id: 9552247767045, server: LDAP-fortiad, local auth: 0,
dn match: 1
[672] __group_match-Use 'tsmith' for user group matching.
[710] __group_match-Check if LDAP-fortiad is a group member
[716] __group_match-Group 'ldap-mis' passed group matching
[719] __group_match-Add matched group 'ldap-mis'(10)
[269] find_matched_usr_grps-Passed group matching
[3442] fnbamd_ldap_result-Result for ldap svr LDAP-fortiad is SUCCESS
…
[672] __group_match-Use 'tsmith' for user group matching.
[710] __group_match-Check if LDAP-fortiad is a group member
[716] __group_match-Group 'ldap-mis' passed group matching
[719] __group_match-Add matched group 'ldap-mis'(10)
[269] find_matched_usr_grps-Passed group matching
[1012] update_auth_token_session-config does not require 2fa
[357] fnbamd_comm_send_result-Sending result 0 (nid 0) for req 9711170703386, len=3012
```

SAML-based authentication for FortiClient remote access VPNs

SAML-based authentication for FortiClient remote access VPNs can be configured using FortiClient 7.2.4 and later. Support is only available using IKEv2. Two factor authentication using FortiToken push is also supported.

The FortiGate authd daemon can support SAML authentication by listening for authentication messages from the FortiClient on the TCP port number configured in the `auth-ike-saml-port` setting (0 - 65535, default = 1001). This setting can be configured in the CLI:

```
config system global
    set auth-ike-saml-port <integer>
end
```

This allows the FortiGate to act as a SAML service provider (SP) for IKEv2 FortiClient remote access IPsec VPN clients by forwarding the FortiClient's SAML request to the configured SAML identity provider (IdP) for user authentication.

In 7.6.1 and later, FortiGate can also listen for SAML authentication messages on the port defined in `ike-tcp-port`. This is helpful in the scenario where administrators have configured IPsec VPN over TCP and they are constrained by the port(s) allowed by their ISP provider. Usually in these cases, VPN can only be established on port 443. As such, the single port defined in `ike-tcp-port` will be used by the iked process to listen for VPN

traffic as well as authentication messages. If the packet does not contain the IKE TCP special message, it will pass the traffic to the authd daemon.

In this scenario, it is recommended to use different ports for `ike-tcp-port` and `auth-ike-saml-port` to prevent the authd daemon from accidentally processing IKE TCP packets and dropping them.

```
config system setting
    set ike-tcp-port <port listened by iked>
end
```

```
config system global
    set auth-ike-saml-port <unique port>
end
```

Using TCP as transport requires FortiClient 7.4.1 or later. See IPsec VPN over TCP.

For more information and an example on using a single port for VPN over TCP and SAML authentication, see the community article Using the same TCP port for IPsec SAML authentication and IKE TCP encapsulation in FortiOS v7.6.1.

The `ike-saml-server` setting enables a configured SAML server to listen on a FortiGate interface for SAML authentication requests from FortiClient remote access IPsec VPN clients. It must be configured on the interface that is directly receiving the SAML authentication requests from FortiClient. This setting can be configured in the CLI:

```
config system interface
    edit <name>
        set ike-saml-server <saml_server>
    next
end
```

The `ike-saml-server` setting must be configured on the interface that is the first point of contact for FortiClient traffic.

For example, if FortiClient user SAML authentication traffic is always routed to the FortiGate on the WAN1 interface, then `ike-saml-server` must be configured for WAN1. If it is configured for WAN2, then the authentication traffic will not reach it on WAN1, even if the FortiGate allows traffic to flow from WAN1 to WAN2.

FortiClient will validate the certificate presented to it by FortiGate during its initial SAML connection. This certificate can be configured on the FortiGate from the GUI under *User & Authentication* > *Authentication Settings* > *Certificate* under *User Authentication Options*. To import the certificate on the FortiGate, see Import a certificate on page 3962.

This certificate can also be configured in the CLI:

```
config user setting
     set auth-cert <certificate>
end
```

To prevent an invalid server certificate prompt on FortiClient, the certificate's common name (CN) should match the IPsec VPN remote gateway's FQDN. If the certificate is signed by a custom Certificate Authority or one that is not well-known, the Certificate Authority's (CA) certificate should be imported in FortiClient endpoint's

Trusted Root Certificate Authority store. For details on installing a CA certificate on the endpoint, see Installing certificates on the client.

Dialup IPsec VPN with SAML using an external browser for authentication is supported starting from FortiOS 7.6.1, FortiClient (Windows) and (macOS) 7.2.5 and 7.4.1 and FortiClient (Linux) 7.4.3.

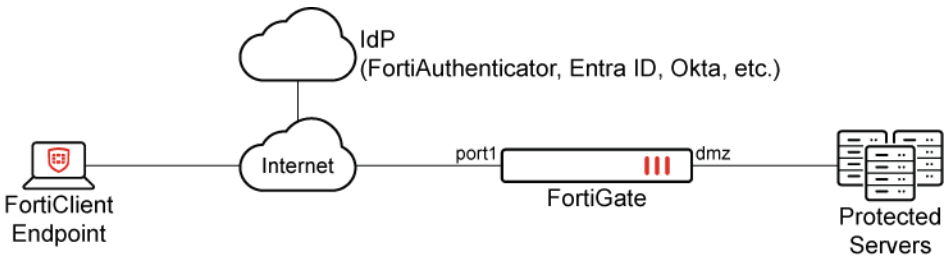
SAML authentication flow with IPsec

The SAML Authentication flow when using IPsec where FortiGate is the Service Provider (SP), FortiAuthenticator, Entra ID, Okta, or another SAML IdP is the Identity Provider (IdP) and FortiClient is the web-browser:

1. When the FortiClient user clicks on Connect on FortiClient to connect to IPsec VPN Gateway (i.e. FortiGate), FortiClient first initiates a connection to FortiGate on the `auth-ike-saml-port` configured on FortiGate.
2. The FortiGate sends a SAML Authentication Requests inside a redirect to FortiClient. The redirect consists of URLs to reach the IdP.
3. FortiClient uses these redirects to send SAML Authentication Request to the IdP after which the login page on the IdP opens up.
4. The user authenticates to the IdP using their SAML credentials configured on the IdP.
5. The IdP sends a SAML Authentication Response that contains the user and group information in form of SAML Assertions to FortiClient.
6. FortiClient sends a SAML Authentication Response to FortiGate.
7. The FortiGate consumes the SAML Authentication Response and SAML Assertions after verifying the IdP using its IdP's certificate and provides FortiClient with a temporary token ID.
8. FortiClient initiates IPsec tunnel and presents the token ID for authentication. Upon successful verification of token ID, IPsec tunnel establishes.

SAML configuration example with different IdPs

The following example shows configuring IPsec with SAML authentication using different IdPs on FortiGate and FortiClient. These examples are based on VPN over UDP.



The configuration steps on the FortiGate, different IdPs and FortiClient are as follows:

1. Configure IKE-SAML authentication port number on FortiGate.
2. Configure IPsec VPN certificate on FortiGate.
3. Configure SAML IdP and SAML SP.
4. Configure IPsec IKEv2 on FortiGate.

5. Configure firewall policies on FortiGate.
6. Configure IPsec VPN profile on FortiClient.
7. Verify the IPsec connection.

Only Configuring SAML IdP and SAML SP on page 3128 is unique to individual IdPs. All other steps listed above are the same on FortiGate and FortiClient when using different IdPs.

Configuring IKE-SAML authentication port number on FortiGate

Configure a suitable TCP port number for SAML authentication (`auth-ike-saml-port`) used by FortiGate. This example uses port 9443 and the setting is configurable using the CLI.

```
config system global
    set auth-ike-saml-port 9443
end
```

If VPN over TCP is used, then the port defined in ike-tcp-port takes precedence:

```
config system setting
    set ike-tcp-port <port listened by iked>
end
```

Use the `ike-tcp-port` in your user.saml definitions. Do not assign the same port to `ike-tcp-port` and `auth-ike-saml-port` when VPN over TCP is used.

Configuring IPsec VPN certificate

In this step, using either the GUI or the CLI, configure the IPsec VPN certificate that is presented to FortiClient upon its initial connection.

To configure the IPsec VPN certificate in the GUI:

1. Go to *User & Authentication* > *Authentication Settings*.
2. Select the certificate from the *Certificate* dropdown menu. To import the certificate on FortiGate, see Import a certificate on page 3962.

To configure the IPsec VPN certificate in the CLI:

If the certificate *VPN_Certificate* has already been imported on the FortiGate, then use the following CLI commands:

```
config user setting
    set auth-cert "VPN_Certificate"
end
```

Configuring SAML IdP and SAML SP

The SAML configuration on SP (FortiGate) will vary based on selected IdPs from the list below. Select the preferred combination of SP and IdP as per your requirement from the following list.

1. Configure FortiAuthenticator as SAML IdP and FortiGate as SAML SP
2. Configure Microsoft Entra ID as SAML IdP and FortiGate as SAML SP

SAML IdPs other than FortiAuthenticator or Microsoft Entra ID can be used. Please refer to the documentation of the respective SAML IdP for details.

Configuring IPsec IKEv2 on FortiGate

Configuring Remote access VPN on FortiGate enables FortiClient to connect to the IPsec VPN gateway configured on FortiGate. FortiClient 7.2.4 GA and above supports only IKEv2 for SAML authentication.

Different options can be selected between transport methods such as UDP, TCP, or Auto mode for IKE negotiation and encapsulation of ESP packets.

Traditionally, dial-up IPsec VPNs use UDP as the transport protocol for IKE negotiations and encrypted traffic. IPsec VPNs can now operate over TCP, providing greater flexibility and compatibility in restrictive network environments. Using TCP as transport is beneficial in scenarios where ISPs, firewalls, or proxy servers only allow inbound and outbound TCP traffic and block UDP traffic. You can also specify a custom TCP port for IPsec communication for both FortiGate and FortiClient. For details on configuring a custom TCP port on FortiClient, see IPsec VPN over TCP. For details on configuring a custom TCP port on FortiGate, see Dialup IPsec VPN using custom TCP port on page 3159.

Using TCP as transport requires IKEv2 and FortiClient 7.4.1 or later. See IPsec VPN over TCP.

The example discussed uses full-tunnel IPsec VPN. For split-tunnel configuration and other advanced configurations as per your requirement, see Remote access on page 2961.

To configure IPsec VPN on FortiGate with FortiClient as the dialup client:

1. Go to *VPN > VPN Tunnels*.
2. Click *Create New > Custom IPsec Tunnel*.
3. Configure the following options:

Field	Value
Name	*FCT_SAML*
Comments	(Optional)
Network	
IP Version	*IPv4*
Remote Gateway	*Dialup User*
Interface	*port1*

Field	Value
	Select the IPsec tunnel gateway interface.
Mode Config	*Enable* Select *IPv4*.
Assign client IP	*Enable* Select *Address/Address Group* from the dropdown list.
IPv4 client IP range	*VPN_Client_IP_Range* *VPN_Client_IP_Range* is configured from *10.212.134.1* to *10.212.134.200*. If it is not already created, select *Create > Address* from the dropdown menu to create a new address object. See IP range on page 2038 for more information.
Subnet Mask	*255.255.255.255*
DNS	*Specify*
IPv4 DNS	*8.8.8.8*
IPv4 split tunnel	*Disable*
Transport	*UDP* Optionally, if using FortiClient 7.4.1 or later for remote access: • Select *TCP encapsulation* to use TCP as transport. • Select *Auto* to use Auto mode.
Authentication	
Method	*Pre-shared key*
Pre-shared key	Enter the unique pre-shared key. Use the same key on FortiClient.
IKE	*version 2*
EAP	*Enable* Select *EAP identity request*
Network overlay	Enable when multiple dialup tunnels are configured. Specify a different network ID for each.
User group	*Inherit from policy* One or more user groups can be used; see Using single or multiple user groups for user authentication on page 3024.

4. In the *Phase 2 selectors* table click *Create New*:
   a. Set *Name* to *FCT_SAML*.
   b. Click *OK*.
5. Click *OK*. The newly created IPsec tunnel would be now visible under *VPN > IPsec Tunnels*.

For other advanced custom configurations as per your requirement, see Remote access on page 2961.

The SAML group configured, `<group-name>`, must be either configured inside the IPsec Phase 1 setting, `set authusrgrp <group-name>`, or in the firewall policy, `set groups <group-name>`, to allow the traffic to flow through the IPsec tunnel. If the SAML group is configured in both IPsec Phase 1 and firewall policy, the traffic stops to flow through the IPsec tunnel. In the example discussed, it is configured it in the firewall policy.

See Using single or multiple user groups for user authentication on page 3024 for more information.

Configuring firewall policies for IPsec tunnel

To configure firewall policies for IPsec tunnel:

1. Go to *Policy & Object > Firewall Policy*.
2. Click *Create New*.
3. Enter the following configuration:

Name	*IPsec to DMZ* Enter the desired name.
Incoming interface	*FCT_SAML* Select the configured IPsec tunnel.
Outgoing interface	*DMZ* Select the interfaces that FortiClient needs access to when it connects to VPN.
Source	Under *Address*, select *VPN_Client_IP_Range*. Under *User*, select *SAML-FAC-Group* (or *SAML-ENTRA-ID-Group*). The group under *User* is the SAML user group configured in the earlier steps.
Destination	*DMZ subnet* Click *Create* if it is not already created. See Subnet on page 2034 for more information.
Service	*ALL*

4. Click *OK*.
5. As IPsec tunnel configured as full-tunnel, create another policy to allow traffic from *IPsec to Internet*, to allow FortiClient to access Internet through IPsec tunnel.

For additional custom settings as per your requirement, see Firewall policy on page 1842.

Configuring IPsec VPN profile on FortiClient

To configure an IPsec VPN profile on FortiClient:

1. In FortiClient, go to *Remote Access* > *Configure VPN* or *Add a new connection*.
2. Set the following settings to configure an IPsec IKEv2 profile on FortiClient:

Connection Name	*VPN-Tunnel*
Remote Gateway	*<VPN Gateway FQDN>* or *<VPN Gateway IP>*
Authentication Method	*Pre-shared key* with *Enable Single Sign On (SSO) for VPN Tunnel* enabled.
Customize port	*9443*
Advanced Settings > VPN Settings	
IKE	*Version 2*
Options	*Mode Config*
Advanced Settings > Phase 1	
IKE Proposal	Encryption: *AES 128*, Authentication: *SHA 256* Encryption: *AES 256*, Authentication: *SHA 256*
DH Group	*20*, *21*
Advanced Settings > Phase 2	
IKE Proposal	Encryption: *AES 128*, Authentication: *SHA 256* Encryption: *AES 256*, Authentication: *SHA 256*
DH Group	*20*

3. Make sure that the Phase 1 and Phase 2 settings, including encryption algorithms, authentication methods, Diffie-Hellman (DH) groups, and other parameters, that are configured on FortiClient exactly match the corresponding Phase 1 and Phase 2 settings defined on the FortiGate IPsec tunnel. A mismatch in any of these parameters will cause the tunnel negotiation to fail.
4. Click *Save*.

To explore additional custom options to configure IPsec VPN profile, see Configuring an IPsec VPN connection.

FortiClient EMS 7.4.4 and later support configuring the Network ID on the IPsec VPN Phase 1 settings in the GUI. For earlier versions, refer to the XML reference guide for IKE settings for details on configuring the `<networkid>` setting.

Verifying IPsec connection

To verify the IPsec connection in the GUI:

1. On the client PC, open FortiClient and select the *Remote Access* tab.
2. Select the VPN tunnel, *VPN-Tunnel*, and click *Connect*.

3. If the connection is successful, a FortiClient pop-up will appear briefly indicating that the IKE negotiation succeeded. The *Remote Access* window now displays *VPN Connected* and the associated VPN tunnel details.
4. In FortiOS, go to *Dashboard > Network Monitor > VPN*.

To verify the IPsec connection in the CLI:

The following debugs are from FortiGate when used with FortiAuthenticator as the IdP. The debugs should be similar for other IdPs depending on the SAML attributes supported and sent by the IdP.

1. Verify the IKE gateway list:

```
# diagnose vpn ike gateway list
vd: root/0
name: FCT_SAML_0
version: 2
interface: port1 3
addr: 10.100.66.99:4500 -> 208.91.115.30:64917
tun_id: 10.212.134.1/::10.0.0.18
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 33s ago
eap-user: testuser
2FA: no
groups:
  SAML-FAC-Group 5
peer-id: 172.19.50.196
peer-id-auth: no
FortiClient UID: 19E1FA565259468FB46EDAA9D595176F
assigned IPv4 address: 10.212.134.1/255.255.255.255
nat: me peer
PPK: no
IKE SA: created 1/1  established 1/1  time 1680/1680/1680 ms
IPsec SA: created 1/1  established 1/1  time 40/40/40 ms

  id/spi: 1049 f883b783547b0c64/f45745cd8b228850
  direction: responder
  status: established 33-31s ago = 1680ms
  proposal: aes256-sha256
  child: no
  SK_ei: 09d0e99e4ee86518-82da5e46c7ef0425-0816ef283fed3ca6-3fa0eeb56ac863a5
  SK_er: 50e94be11ece32f8-aa13e54400e29531-684473a924ff04c5-8ebf45d854a59412
  SK_ai: 3d95eec2deb54cf1-a59a945f0156c214-fe9aa188a96dd70c-f2394e1f7bb647b0
  SK_ar: 0c0a478b800c7c9c-9dc56c05e9657200-7399b15d13ab8ad9-13984182abea936c
  PPK: no
  message-id sent/recv: 0/12
  QKD: no
  lifetime/rekey: 86400/86098
  DPD sent/recv: 00000000/00000000
  peer-id: 172.19.50.196
```

2. Verify the authd daemon debug output:

```
# diagnose debug application authd -1
...
[authd_http_on_method_post:5151]: src 10.1.100.253 flag 00008000
[authd_local_saml_auth:5602]: SAML login with UID '19E1FA565259468FB46EDAA9D595176F'.
[authd_http_prepare_javascript_redir:3852]: https://<VPN Gateway
FQDN>:9443/saml?0704048f9683e491
...
```

3. Verify the samld daemon debug output:

```
# diagnose debug application samld -1
...
</Session>
samld_send_common_reply [99]:     Attr: 17, 31, magic=040c07809dafc13e
samld_send_common_reply [99]:     Attr: 18, 29, 2024-03-19T21:42:21Z
samld_send_common_reply [95]:     Attr: 10, 26, 'username' 'testuser'
samld_send_common_reply [95]:     Attr: 10, 17, 'group' 'IT'
...
```

4. Verify the fnbamd daemon debug output:

```
# diagnose debug application fnbamd -1
...
[2426] handle_req-Rcvd auth cache message
[133] __saml_auth_cache_push-Auth cache created, user='19E1FA565259468FB46EDAA9D595176F',
SAML_server='saml-fac', vfid=0
[140] __saml_auth_cache_push-Hash bucket 227
[182] __saml_auth_cache_push-New auth cache entry is created,
user='19E1FA565259468FB46EDAA9D595176F', expires=1648598587, SAML_server='saml-fac', vfid=0
[1918] handle_req-Rcvd auth req 994781475 for 19E1FA565259468FB46EDAA9D595176F in ipsec
opt=00000000 prot=5
[466] __compose_group_list_from_req-Group 'saml-fac', type 1
[971] fnbamd_saml_auth_cache_lookup-Authneticating '19E1FA565259468FB46EDAA9D595176F'.
[1005] fnbamd_saml_auth_cache_lookup-Authentication passed.
```

5. Verify the IPsec daemon debug output:

```
# diagnose debug application ike -1
...
ike V=root:0:FCT_SAML: user 'testuser' authenticated group 'SAML-FAC-Group' 5
ike V=root:0:FCT_SAML:1180: responder preparing EAP pass through message
...
ike V=root:0:FCT_SAML_0:1180: mode-cfg assigned (1) IPv4 address 10.212.134.1
ike V=root:0:FCT_SAML_0:1180: mode-cfg assigned (2) IPv4 netmask 255.255.255.255
ike V=root:0:FCT_SAML_0:1180: mode-cfg send (13) 0:0.0.0.0/0.0.0.0:0
ike V=root:0:FCT_SAML_0:1180: mode-cfg send (3) IPv4 DNS(1) 8.8.8.8
...
ike V=root:0:FCT_SAML_0: sent tunnel-up message to EMS: (fct-
uid=19E1FA565259468FB46EDAA9D595176F, intf=FCT_SAML_0, addr=10.212.134.1, vdom=root)
ike V=root:0:FCT_SAML_0: user 'testuser' 10.212.134.1 groups 1
...
```

Configuring FortiAuthenticator as SAML IdP and FortiGate as SAML SP

This topic discusses the configuration steps required on FortiAuthenticator to act as the Identity Provider (IdP) and FortiGate to act as Service Provider (SP) during SAML Authentication for IPsec connection, as a part of the overall configuration in SAML-based authentication for FortiClient remote access VPNs on page 3124.

In FortiOS 7.6.1 and later, if TCP is used as the transport protocol for IPsec connections, you can use the same TCP port for both IKEv2 IPsec tunnels and the SAML Service Provider (SP) configuration. This reduces open ports and helps simplify deployment, particularly in restrictive network environments. When the same port is used, the `ike-tcp-port` setting takes precedence, and the `auth-ike-saml-port` setting is irrelevant; FortiGate automatically handles both services over the shared port.

Using TCP as transport requires FortiClient 7.4.1 or later. See IPsec VPN over TCP.

In the example discussed, the following assumptions and configuration steps are used:

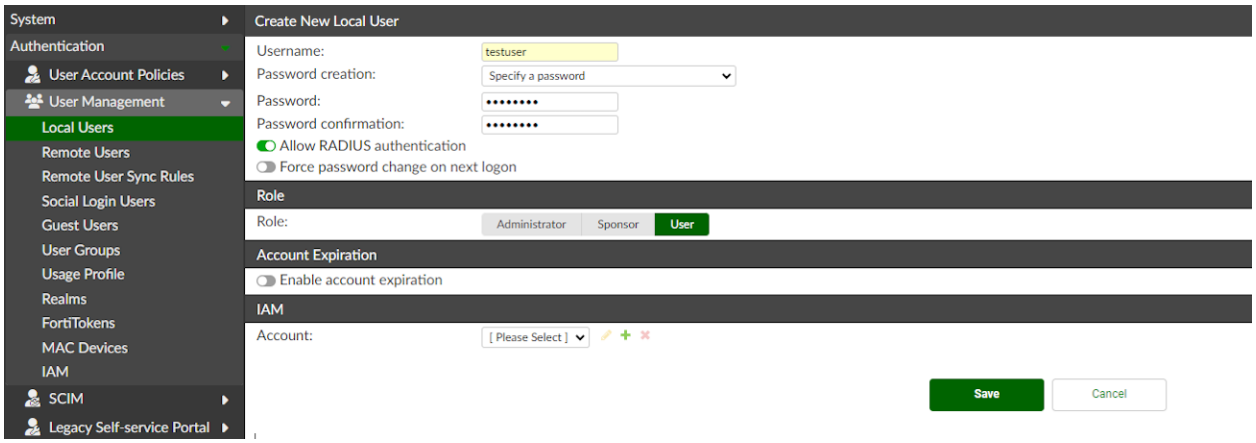
1. FortiAuthenticator is configured with local user (*testuser*) inside a User group (*IT*).
2. This user belongs to a unique realm (*samlrealm*).
3. FortiAuthenticator also requires a server certificate (also called as IdP certificate) for itself signed by a well-known CA or trusted by FortiClient endpoint and FortiGate.
4. The IdP certificate must be imported into FortiGate after which FortiAuthenticator can use the certificate to sign the SAML messages.
5. FortiAuthenticator must then be configured as SAML IdP and FortiGate as SAML SP.

To configure a local user on FortiAuthenticator:

1. Go to *Authentication* > *User Management* > *Local Users* and select *Create New*.
2. Enter the following details shown below and leave other settings as the defaults.

Username	*testuser*
Password creation	*Specify a password*
Password	Enter a desired password
Password confirmation	Re-enter the password

3. Click *Save*.



For more advanced and custom configuration options, see Local users.

To configure a user group on FortiAuthenticator:

1. Go to *Authentication* > *User Management* > *User Groups* and select *Create New*.
2. To create a user group with a local user:
   a. In the *Name* field, enter the group name as *IT*.
   b. Set the *Type* as *Local*.
   c. Under *Users*, from the *Available Users* table, select *testuser* and move it to the *Chosen Users* table.
3. Click *Save*.

For more advanced options, see User Groups.

To configure a user realm on FortiAuthenticator:

1. Go to *Authentication* > *User Management* > *Realms* and select *Create New*.
2. Name the realm as *samlrealm*.
3. In *User source*, from the dropdown, select *Local Users*.
4. Click *Save*.

For more advanced options, see Realms.

To import a server certificate on FortiAuthenticator:

1. Go to *Certificate Management* > *End Entities* > *Local Services* and select *Import*.
2. Depending on which file format your certificate is in, select the suitable *Type*.
3. Select *Upload a file* to locate the certificate file on your computer.
4. Click *Import*.

See End entities for more information on certificates on FortiAuthenticator.

To configure general SAML IdP settings on FortiAuthenticator for SAML:

1. Go to *Authentication* > *SAML IdP* > *General*, and select *Enable SAML Identity Provider portal*.
2. Configure the following settings:

Device FQDN	*<FortiAuthenticator FQDN>* To configure this setting, you must enter a *Device FQDN* in the *System Information* widget in the *Dashboard*.
Server address	*<FortiAuthenticator FQDN>* Enter the IP address or FQDN of the FortiAuthenticator device. This address must be accessible by the FortiClient endpoint.
Username input format	*username@realm*
Use default realm when user-provided realm is different from all configured realms	*Disabled*
Realms	*Realm*: *samlrealm* (Optional) *Groups* > *Filter*: *IT* Use *Groups* and *Filter* to add specific user groups. These user groups may be local users configured on the FortiAuthenticator itself or remote users populated from different remote authentication servers. See User Groups and Remote users for more information.
Legacy login sequence	*Disabled*
Default IdP certificate	*<IdP certificate>* This certificate is used by IdP to sign SAML messages before sending it to IdP. To import this certificate on FortiAuthenticator see Importing a server certificate. This certificate also needs to be imported on the SP (FortiGate) to be used in the SAML configuration. See Remote certificate on page 3968.

3. Select *Save* to apply any changes that you have made.

To configure service provider SAML settings on FortiAuthenticator for SAML:

1. Go to *SAML IdP* > *Service Providers*.
2. Click *Create New*.
3. Enter the following:

SP name	*FortiGate* Use any suitable SP name.
Server certificate	*<IdP certificate>* This certificate is used by IdP to sign SAML messages before sending it to IdP. To import this certificate on FortiAuthenticator, see Importing a server certificate. This certificate also needs to be imported on the SP (FortiGate) to be used in the SAML configuration. See Remote certificate on page 3968.
IdP Metadata	

Select an identifier to display IdP info	Click *+* to create a new IdP prefix. In the *IdP identifier* field, enter the prefix as *fac* and click *OK*. Fields such as *IdP entity id*, *IdP single sign-on URL*, and *IdP single logout URL* will populate automatically with this prefix information. These URLs must be accessible by FortiClient endpoints. Copy and paste this URL information in a separate notepad file as it will be used to configure IdP information on SP (FortiGate) later.
Authentication	
Authentication Method	*Password-only*
Assertion Attributes	Click *+* to expand it. Configure two SAML assertion attributes (*username* and *group*) as follows:
Assertion attribute	
SAML attribute	*username*
User attribute	*FortiAuthenticator > Username*
Assertion attribute	
SAML attribute	*group*
User attribute	*FortiAuthenticator > Group*

4. Click *Save*.

To export SAML IdP server certificate and import it on FortiGate:

1. On FortiAuthenticator, go to *Certificate Management > End Entities > Local Services*.
2. Select the certificate, *<IdP certificate>*, by selecting the left checkbox for the certificate entry in the table and clicking *Export Certificate*.
3. Go to the file location on your local computer and click *Save*.
4. On FortiGate, go to *System > Certificates*, and from the *Create/Import* dropdown, select *Remote Certificate*.
5. Select *Upload* to locate and upload the .cer remote certificate from your computer.
6. Click *OK*.
   The new certificate is now visible in the *System > Certificates* page under *Remote Certificate*.

To configure SAML server on FortiGate:

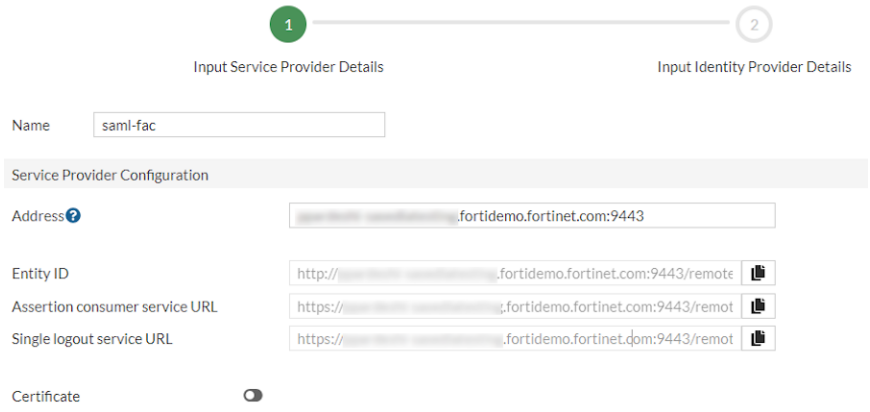
1. Go to *User & Authentication > Single Sign-On*.
2. Click *Create New*.
3. Enter the *Name* as *saml-fac*.
4. In the *Address* field, enter the FQDN/IP information in the following format:
   <ipsec-vpn-gateway-fqdn/ip-address>:<saml-ike-authentication-port>
   The *Address* field is used by FortiClient to initiate IPsec connection to FortiGate.
   If your IPsec VPN is configured to use TCP as the transport protocol on a specific port, the same TCP port can be configured for the saml-ike-authentication-port setting (FortiOS 7.4.1 and later). This allows using a

shared TCP port for both IPsec VPN and SAML SP configuration.

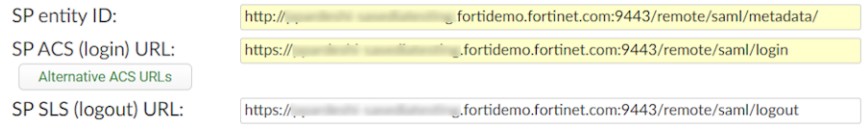
5. On the FortiGate, from *Service Provider Configuration*, copy the following URLs (*Entity ID*, *Assertion consumer service URL*, *Single logout service URL*). In FortiAuthenticator, enter it in *Authentication > SAML IdP > Service Providers > FortiGate*, inside the *SP Metadata* fields according to following mapping:

FortiGate settings	FortiAuthenticator settings
Entity ID	SP entity ID
Assertion consumer service URL	SP ACS (login) URL
Single logout service URL	SP SLS (logout) URL

The following demonstrates on the FortiGate:



The following demonstrates on FortiAuthenticator:

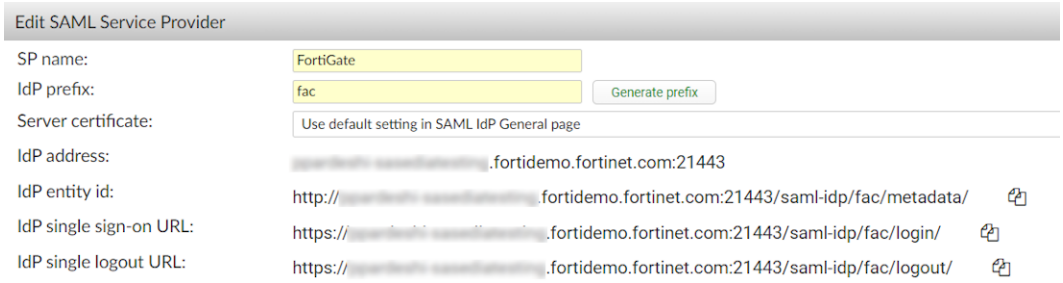


6. Click *Save* on FortiAuthenticator to save the SP URLs.
7. Return to FortiGate GUI, and click *Next*.

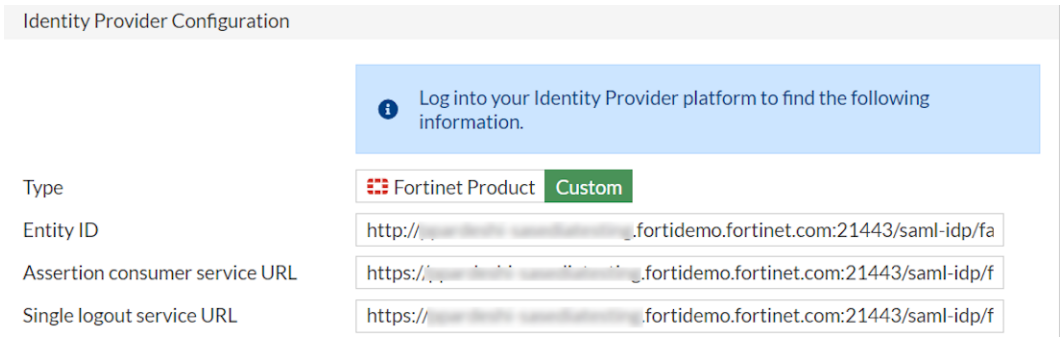
   Set the *Type* as *Custom*.The *Entity ID*, *Assertion consumer service URL*, and *Single logout service URL* fields required are available from FortiAuthenticator.
8. To copy the following URLs (*IdP entity id*, *IdP single sign-on URL*, *IdP single logout URL*) from FortiAuthenticator, go to *Authentication > SAML IdP > Service Providers > FortiGate* under *IdP Metadata*. Copy the URLs from FortiAuthenticator and paste it on FortiGate according to the following mapping:

FortiAuthenticator settings	FortiGate settings
IdP entity ID	Entity ID
IdP single sign-onURL	Assertion consumer service URL
IdP single logout URL	Single logout service URL

The following demonstrates on FortiAuthenticator:



The following demonstrates on the FortiGate:



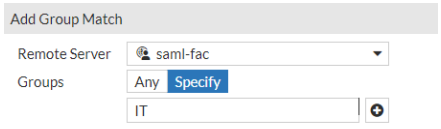
9. In the *Certificate* dropdown, select the IdP certificate that was imported on FortiGate.
10. Enter the following details in *Additional SAML Attributes*.

Attribute used to identify users	*username*
Attribute used to identify groups	*group*

11. Click *Submit* to save the changes.

To create SAML user group on FortiGate:

1. Go to *User & Authentication* > *User Groups* .
2. Click *Create New*.
3. Enter *Name* as *SAML-FAC-Group*.
4. In *Remote Groups*, click *Add*.
5. From the *Remote Server* dropdown, select *saml-fac* SAML server.
6. In the *Groups* field, click *Specify* and enter *IT*.



7. Click *OK*.
8. Click *OK*.

To associate SAML server with IPsec gateway interface:

Use FortiGate CLI to bind and associate the SAML server with the VPN gateway interface (port1) as follows:

```
config system interface
    edit "port1"
        set ike-saml-server "saml-fac"
    next
end
```

Configuring SAML on IdP and SP is now complete. The next step is to use SAML configuration inside IPsec configuration. To configure IPsec, see Configuring IPsec IKEv2 on FortiGate on page 3128.

Configuring Microsoft Entra ID as SAML IdP and FortiGate as SAML SP

This topic discusses the configurations steps required if your users are managed through Microsoft Entra ID (formerly Azure Active Directory), as a part of the overall configuration in SAML-based authentication for FortiClient remote access VPNs on page 3124. Microsoft Entra ID will be configured as Identity Provider (IdP) and FortiGate as Service Provider (SP) during SAML Authentication for IPsec connection.

In FortiOS 7.6.1 and later, if TCP is used as the transport protocol for IPsec connections, you can use the same TCP port for both IKEv2 IPsec tunnels and the SAML Service Provider (SP) configuration. This reduces open ports and helps simplify deployment, particularly in restrictive network environments. When the same port is used, the `ike-tcp-port` setting takes precedence, and the `auth-ike-saml-port` setting is irrelevant; FortiGate automatically handles both services over the shared port.

Using TCP as transport requires FortiClient 7.4.1 or later. See IPsec VPN over TCP.

To configure a user on Microsoft Entra ID:

1. Login to the Azure portal (portal.azure.com).
2. Search for *Microsoft Entra ID service* in the search bar and click on it.
3. In the left side menu, go to *Users*.
4. Select *New user* > *Create new user*.
5. In the *Basic* properties:
   a. In the *Display name* field, enter *testuser*.
   b. In the *User principal name* field, enter the username@companydomain.extension. For example, *testuser@<mydomain>.onmicrosoft.com*.
   c. Select *Show password* and then write down the value that's displayed in the *Password* box.
   d. Select *Review + create*.
6. Select *Create*.

To configure a Security group and add user to it on Microsoft Entra ID:

In this section, we create a security group named *IT* in Microsoft Entra ID for the *testuser*. FortiGate will use this security group to grant the user network access through the VPN.

1. In the Azure portal, navigate to *Microsoft Entra ID* service.
2. In the left side menu, go to *Groups*.

a. Select *New Group*.
b. In the *Group type* list, select *Security*.
c. In the *Group name* field, enter *IT*.
d. (Optional) In the *Group description* field, enter *Group for granting FortiGate VPN access*.
e. For the *Microsoft Entra roles can be assigned to the group (Preview) settings*, select *No*.
f. In the *Membership type* field, select *Assigned*.
g. Under *Members*, select *No members selected*.
h. In the *Users and groups* dialog field, select *testuser* from the *Users* list, and then click *Select*.
i. Select *Create*.

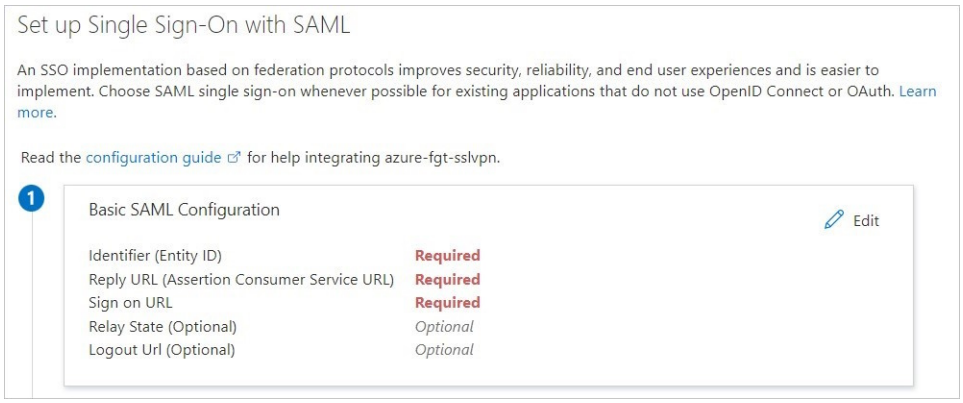
3. Back in the *Groups* section in Microsoft Entra ID, find the *IT* group and note the *Object Id*. This will be needed later.

To configure Enterprise application on Azure portal:

1. Configure user and groups:
   a. In the Azure portal, search for *Enterprise applications service* in the search bar.
   b. Click on *New application* and search for *FortiGate SSL VPN*.

      The application is named for "SSL VPN" but should still work with the IPsec VPN configuration.

   c. Once the application is found, select it, change the *Name* to *FortiGate IPsec VPN*, and click *Create*. It may take a few seconds to create the application.
   d. Once the application is created, go to *Enterprise application* > *All applications* > *FortiGate IPsec VPN*.
   e. On the application's overview page, in the *Manage* section, select *Users and groups*.
   f. Select *Add user/group*, then select *Users* in the *Add Assignment* dialog.
   g. In the *Users and groups* dialog, select *testuser* in the *Users* list, and then click *Select*.
   h. (Optional) If you are expecting any role value in the SAML assertion, in the *Select Role* dialog, select the appropriate role for the user from the list. Click *Select*.
   i. In the *Add Assignment* dialog, select *Assign*.
2. Configure single sign-on:
   a. Browse to *Enterprise application* > *All applications* > *FortiGate IPsec VPN application*.
   b. In the *Manage* section, select *Single sign-on*.
   c. In the *Select a single sign-on method* page, select *SAML*.
   d. In the *Set up Single Sign-On with SAML* page, select *Edit* for *Basic SAML Configuration* to edit the settings:



i. In *Identifier*, enter a URL in the pattern https://<FortiGate IP or FQDN address>:<Custom SAML-IKE port>/remote/saml/metadata
ii. In *Reply URL*, enter a URL in the pattern https://<FortiGate IP or FQDN address>:<Custom SAML-IKE port>/remote/saml/login
iii. In *Sign on URL*, enter a URL in the pattern https://<FortiGate IP or FQDN address>:< Custom SAML-IKE port >/remote/saml/login
iv. In *Logout URL*, enter a URL in the pattern https://<FortiGate IP or FQDN address>:< Custom SAML-IKE port >/remote/saml/logout

These URLs can be retrieved and copied directly from the FortiGate. Go to To configure SAML server on FortiGate: on page 3137 to get these URLs.

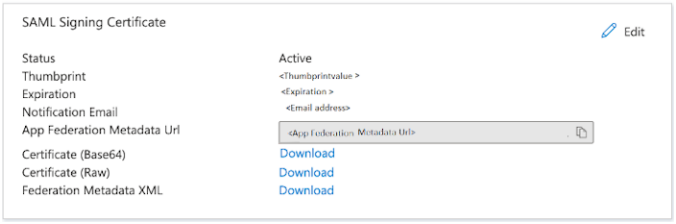
e. Click *Save*.
f. The FortiGate IPsec VPN application expects SAML assertions in a specific format, which requires you to add custom attribute mappings to the configuration.
g. The claims required by FortiGate IPsec VPN are shown in the following table. Names are case-sensitive.

Name	Source attribute
username	user.userprincipalname
group	user.groups

To create these additional claims:

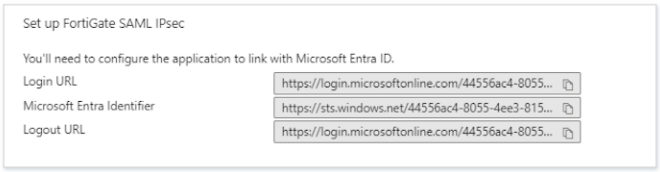
i. Next to *User Attributes & Claims*, select *Edit*.
ii. Select *Add new claim*.
iii. For *Name*, enter *username*.
iv. For *Source attribute*, select *user.userprincipalname*.
v. Select *Save*.
vi. Select *Add a group claim*.
vii. Select *All groups*.
viii. Under *Advanced options*, select the *Customize the name of the group claim*.
ix. For *Name*, enter *group*.
x. Select *Save*.

h. In the *Set up Single Sign-On with SAML* page, in the *SAML Signing Certificate* section, select *Download* next to *Certificate (Base64)* to download the certificate and save it on your computer. This will be needed in the step To export SAML IdP server certificate and import it on FortiGate: on page 3143



i. In the *SAML Signing Certificate* section, click *Edit*, set *Signing Option* to *Sign SAML response and assertion*, then click *Save*.

j. In the *Set up FortiGate SAML IPsec* section, copy the URLs (*Login URL*, *Microsoft Entra Identifier*, *Logout URL*) and paste it inside FortiGate's SAML server configuration, discussed in To configure SAML server on FortiGate: on page 3143.



To export SAML IdP server certificate and import it on FortiGate:

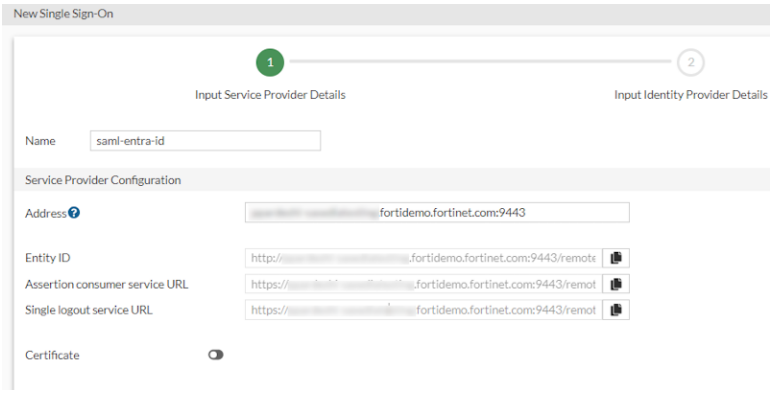
1. On FortiGate, go to *System* > *Certificates*, and from the *Create/Import* dropdown, select *Remote Certificate*.
2. Select *Upload* to locate and upload the .cer remote certificate from your computer.
3. Click *OK*. The certificate will now be visible under *System* > *Certificates* > *Remote Certificate*.

To configure SAML server on FortiGate:

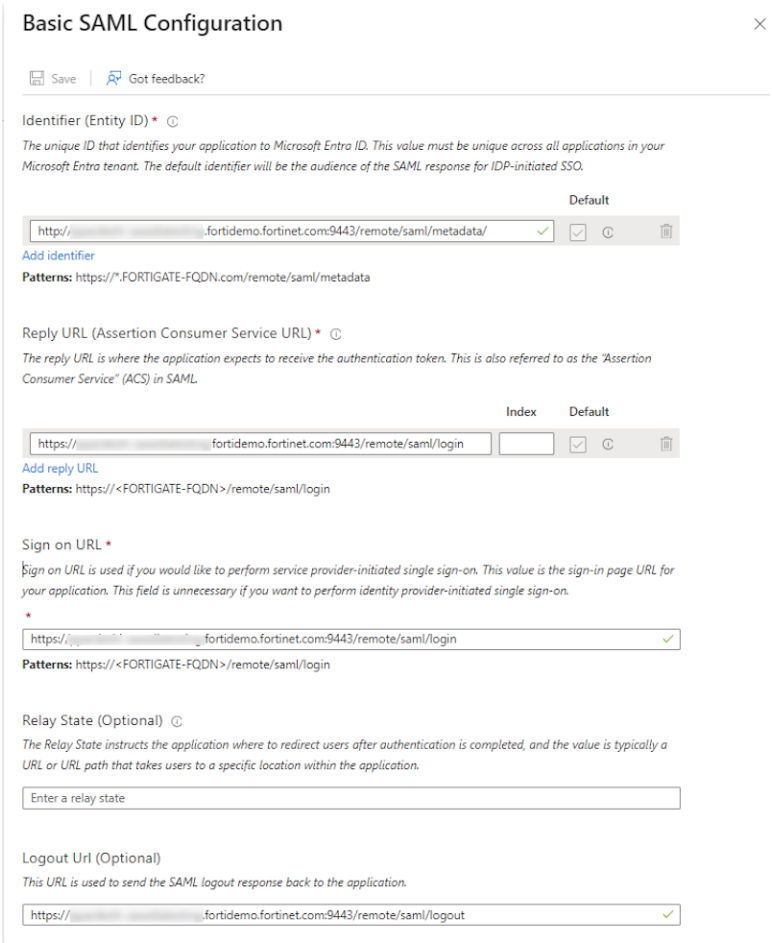
1. On the FortiGate, go to *User & Authentication* > *Single Sign-On* > *Create new*.
2. Enter the *Name* as *saml-entra-id*.
3. In the *Address* field, enter the FQDN/IP information in the following format:
   <ipsec-vpn-gateway-fqdn/ip-address>:<saml-ike-authentication-port>
   The *Address* field is used by FortiClient to initiate IPsec connection to FortiGate.
   If your IPsec VPN is configured to use TCP as the transport protocol on a specific port, the same TCP port can be configured for the saml-ike-authentication-port setting (FortiOS 7.4.1 and later). This allows using a shared TCP port for both IPsec VPN and SAML SP configuration.
4. In *Service Provider Configuration*, copy the following URLs (*Entity ID*, *Assertion consumer service URL*, *Single logout service URL*) and use it in *Azure in Enterprise application* > *All applications* > *FortiGate IPsec VPN* > *Single sign-on page* > *Basic SAML Configuration*. Use the following mapping to copy the required values:

FortiGate settings	Microsoft Entra ID settings
Entity ID	Identifier (Entity ID)
Assertion consumer service URL	Reply URL (Assertion Consumer Service URL)
Assertion consumer service URL	Sign on URL
Single logout service URL	Logout URL (Optional)

The following demonstrates on the FortiGate:



The following demonstrates on Microsoft Entra ID:



5. On the FortiGate GUI, click *Next*.
6. In *Identity Provider Details*, set the *Type* as *Custom*.
7. Paste the URLs copied from last step of section To configure Enterprise application on Azure portal: on page 3141 according to the following mapping:

Microsoft Entra IDsettings	FortiGate settings
Login URL	Assertion consumer service URL
Microsoft Entra Identifier	Entity ID
Logout URL	Single logout service URL

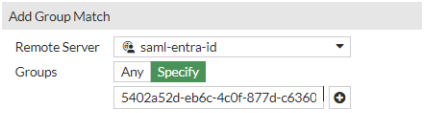
8. Select the *Certificate* from the dropdown. This certificate was imported into FortiGate in section To export SAML IdP server certificate and import it on FortiGate: on page 3143.
9. In the *Additional SAML Attributes* section, enter the following attributes:

Attribute used to identify users	*username*
Attribute used to identify groups	*group*

10. Click *Submit* to save the changes.

To configure SAML user group on FortiGate:

1. Go to *User & Authentication* > *User Groups* > *Create New*.
2. Enter *Name* as *SAML-ENTRA-ID-Group*.
3. In *Remote Groups*, click *Add*.
4. From the *Remote Server* dropdown, select *saml-entra-id SAML server*.
5. In *Groups*, click *Specify* and paste the Object ID copied in the section To configure a Security group and add user to it on Microsoft Entra ID: on page 3140.



6. Click *OK*.
7. Click *OK*.

To associate SAML server with IPsec gateway interface:

Use the FortiGate CLI to bind and associate the SAML server with the VPN gateway interface (port1) as follows:

```
config system interface
    edit "port1"
        set ike-saml-server "saml-entra-id"
    next
end
```

Configuring SAML on IdP and SP is now complete. The next step is to use SAML configuration inside IPsec configuration. To configure IPsec, see Configuring IPsec IKEv2 on FortiGate on page 3128.

Enhancing IPsec security using EMS SN verification

This feature ensures that only licensed FortiClient endpoints can establish an IPsec VPN connection with FortiGate. The FortiGate performs EMS SN verification, and for this feature to work, both the FortiGate and FortiClient endpoints must be connected to the same FortiClient EMS.

To enable the EMS SN verification in the CLI:

```
config vpn ipsec phase1-interface
    edit <name>
        set ems-sn-check {enable | disable}
    next
end
```

Command	Description
`set ems-sn-check`	Enable or disable EMS serial number verification.

IKE Mode Config clients

FortiGate supports dynamic configuration assignments to VPN clients using:

- IKEv1 Mode Config
- IKEv2 Configuration Payload (CP)

For both IKEv1 and IKEv2, FortiGate refers to these mechanisms simply as Mode Config. Mode Config allows the FortiGate to assign networking parameters to remote clients during tunnel establishment, enabling centralized configuration and simplifying client deployment.

Mode Config parameters

A FortiGate can be configured as an IKE Mode Config server or client. As a server (type dialup), FortiGate can assign common parameters such as:

- IPv4 or IPv6 address
- IPv4 subnets
- DNS servers
- WINS servers
- Domain suffix (IKEv1)
- Multiple DNS domains suffix (IKEv2)
- Split tunnel routes

With Mode Config extensions, FortiGate can also assign these in IKEv2:

- Split DNS
- Split services

As a Mode Config client (type static or ddns), the FortiGate can get IP address and other parameters from the server.

An IKE Mode Config server or client is configured using `config vpn ipsec phase1-interface` and involves the following parameters:

Parameter	Description
ike-version {1 | 2}	IKEv2 is the default for FortiGate IPsec VPNs. IKEv1 is considered legacy configuration.
mode-cfg {enable | disable}	Enable/disable IKE Mode Config.
type {static | dynamic | ddns}	If you set `type` to `dynamic`, an IKE Mode Config server is created. The other settings create an IKE Mode Config client.
assign-ip {enable | disable}	Enable to request an IP address from the server.
ip-version {4 | 6}	By default, IPsec VPNs use IPv4 addressing.
assign-ip-from { range | dhcp | name }	• range: Assign IP address from locally defined range. • dhcp: Assign IP address via DHCP. • name: Assign IP address from firewall address or group.
ipv4-start-ip {ipv4-address} ipv6-start-ip {ipv6-address}	The start of the IPv4 or IPv6 range.
ipv4-end-ip {ipv4-address} ipv6-end-ip {ipv6-address}	The end of the IPv4 or IPv6 range.
ipv4-netmask {ipv4-netmask}	The IPv4 network mask assigned to each client (default = 255.255.255.255).
ipv4-dns-server1 {ipv4-address} ipv4-dns-server2 {ipv4-address} ipv4-dns-server3 {ipv4-address} ipv6-dns-server1 {ipv6-address} ipv6-dns-server2 {ipv6-address} ipv6-dns-server3 {ipv6-address}	Up to 3 DNS servers.
Ipv4-wins-server1 {ipv4-address} Ipv4-wins-server2 {ipv4-address}	Up to 2 WINS servers.
ipv4-split-include <string> ipv6-split-include <string>	Mode Config server configuration. Applicable to IKEv1 and IKEv2.

Parameter	Description
	Specify the firewall address or address group that represents the subnets that the clients will have access to. This information is sent to the clients so that default traffic should not flow over the IPsec tunnel except for the specified subnets.
split-include-service <string>	Mode Config server configuration. Applicable to IKEv1 and IKEv2. Specify the service or service group that represents the services that the clients will have access to. This information is sent to the clients so that default traffic should not flow over the IPsec tunnel except for the specified services.
ipv4-split-exclude <string> ipv6-split-exclude <string>	Specify the subnets that should not be accessed over the IPsec tunnel. This information is sent to the clients so that all default traffic should flow over the IPsec tunnel except for the specified subnets. See Split-exclude in IKEv1.
domain <string>	Allows the configuration of a DNS suffix and is available only in IKEv1. To enable this feature, unity-support must be enabled using the `set unity-support enable` command in the phase 1 configuration. A maximum of one DNS suffix can be specified per IPsec tunnel.
internal-domain-list <domain name list>	Domains within the domain name list will be resolved by the tunnel's DNS. Other DNS queries will be resolved by the local DNS. This requires IKEv2. See Split DNS on page 3152 for information.
dns-suffix-search <dns suffix list>	Pass the configured DNS suffix list to the FortiClient to be installed when it connects over IKEv2.

Creating an IKE Mode Config client

In this example, the FortiGate connects to a VPN gateway with a static IP address that can be reached through port1. Only the port, gateway, and proposal information needs to be configured. All other configuration information will come from the IKE Mode Config server.

To configure an IKE Mode Config client:

```
config vpn ipsec phase1-interface
    edit vpn1
        set ip-version 4
        set type static
        set remote-gw <gw_address>
        set interface port1
        set proposal aes128-sha256 aes256-sha256
        set mode-cfg enable
        set assign-ip enable
    next
end
```

Creating an IKE Mode Config server

To configure IKE Mode config settings, the following must be configured first :

```
config vpn ipsec phase1-interface
    edit "vpn-p1"
        set type dynamic
        set interface <interface_name>
        set ike-version {1 | 2}
        set mode-cfg enable
        set proposal <encryption_combination>
        set ip-version {4 | 6}
    next
end
```

Assigning IP addresses

While mode config can do more than assigning IPs and DNS to clients, this is typically the most basic use case. For IKEv1, mode-config is triggered after IKE Phase 1 authentication. For IKEv2, Configuration Payload (CP) is exchanged during IKE_AUTH using CFG_REQUEST and CFG_REPLY payloads.

There are several methods to assign IPs from.

To assign an IP from an address range:

```
config vpn ipsec phase1-interface
    edit vpn1
        set ip-version 4
        set assign-ip enable
        set assign-ip-from range
        set ipv4-start-ip <range_start>
        set ipv4-end-ip <range_end>
        set ipv4-netmask <netmask>
    next
end
```

To assign an IP from a named firewall address or group:

```
config vpn ipsec phase1-interface
    edit vpn1
        set type dynamic
        set assign-ip-from name
        set ipv4-name <name>
        set ipv6-name <name>
    next
end
```

To assign an IP from IPAM:

This method is the default in configuring remote access using the VPN Wizard.

Internally, an address is defined using the IPAM type. See IPAM firewall address New on page 2051.

```
config firewall address
    edit "ipam"
        set type ipam
        set fabric-object enable
        set subnet 192.168.0.0 255.255.255.0
        set ipam-allocate-unique enable
    next
end
```

Once defined, the address object is applied to the phase1 configs:

```
config vpn ipsec phase1-interface
    edit vpn1
        set type dynamic
        set assign-ip-from name
        set ipv4-name “ipam”
    next
end
```

Assigning from a DHCP server

IKE Mode Config can use a remote DHCP server to assign the client IP addresses. Up to eight server addresses can be selected for either IPv4 or IPv6. The DHCP proxy must be enabled first.

To assign an IP from a DHCP server:

```
config system settings
    set dhcp-proxy enable
    set dhcp-server-ip <address>
    set dhcp6-server-ip <address>
end
```

```
config vpn ipsec phase1-interface
    edit vpn1
        set mode-cfg enable
        set assign-ip-from dhcp
    next
end
```

Assigning from a RADIUS server

If the client is authenticated by a RADIUS server, you can obtain the user’s IP address assignment from the Framed-IP-Address attribute. The user must be authenticated using XAuth in IKEv1 or EAP in IKEv2.

The users must be authenticated by a RADIUS server and assigned to the FortiGate user group `<grp_name>`.

To assign an IP from a RADIUS server:

```
config vpn ipsec phase1-interface
    edit vpn1
        set type dynamic
        set mode-cfg enable
        set assign-ip enable
        set assign-ip-from usrgrp
        set eap enable
        set authusrgrp <grp_name>
    next
end
```

Split tunneling

Split-include and Split-exclude

The `split-exclude` setting specifies that default traffic flows over the IPsec tunnel except for specified subnets. This is the opposite of `split-include`, which specifies that default traffic should not flow over the IPsec tunnel except for specified subnets. The `split-include` and `split-exclude` settings can be specified at the same time.

To configure split-exclude:

```
config vpn ipsec phase1-interface
    edit <name>
        set ike-version {1 | 2}
        set type dynamic
        set mode-cfg enable
        set ipv4-split-exclude <string>
        set ipv6-split-exclude <string>
    next
end
```

DNS suffix

The DNS suffix enables DNS resolution of network resources using their hostnames, without requiring clients to specify their fully qualified domain names (FQDN). This feature is particularly useful in environments where users access internal resources over VPN connections. By appending a DNS suffix to unqualified domain names (such as hostnames), it enables end systems to generate FQDNs required for DNS resolution.

IKEv2 DNS suffix search

The configuration of a DNS suffix on an IPsec IKEv2 tunnel requires phase 1 to be a dynamic gateway with mode-cfg enabled:

```
config vpn ipsec phase1-interface
    edit <tunnel_name>
```

```
        set mode-cfg enable
        set ike-version 2
        set type dynamic
        set dns-suffix-search example.com
    next
end
```

Command	Description
dns-suffix-search <suffix>	Specify one or more DNS suffixes in quotes separated by spaces.

For IKE version 2, DNS suffix is supported when connecting from a Windows FortiClient running 7.4.4 or later. More than one DNS suffix can be configured per IPsec tunnel.

For more information, see IPsec DNS suffix on page 3153.

IKEv1 DNS suffix

DNS suffix configuration is natively supported for IKE version 1. Only one DNS suffix is configurable per IPsec tunnel.

The configuration of a DNS suffix on an IPsec IKEv1 tunnel requires enabling unity-support in the phase 1 configuration of the IPsec tunnel:

```
config vpn ipsec phase1-interface
    edit <name>
        set unity-support enable
        set domain <string>
    next
end
```

For more information, see IPsec DNS suffix on page 3153.

Split DNS

This functionality empowers clients to determine whether DNS traffic should utilize the tunnel's DNS or the local DNS server for query resolution. This is achieved by letting users specify a list of FQDNs. Only FQDNs that match the specified list are directed to the tunnel for resolution, while all other queries are handled by the local DNS server.

The `internal-domain-list` option is available on IKEv2 phase1 dialup gateways if `mode-cfg` is enabled.

To enable IPsec Split DNS in the CLI:

```
config vpn ipsec phase1-interface
    edit <name>
        set type dynamic
        set ike-version 2
```

```
        set mode-cfg enable
        set dns-mode {manual | auto}
        set internal-domain-list <domain name>
    next
end
```

Command	Description
`internal-domain-list <domain name>`	One or more internal domain names in quotes separated by spaces.

Two scenarios need attention:

1. When there is no split tunnel, or the split tunnel is set to address *all*, the user must manually select the *Enable Local LAN* checkbox in the FortiClient by navigating to *Advanced Settings* > *Phase 1*. If not, only the FQDN matching the `internal-domain-list` will be resolved, discarding other DNS queries. However, once this setting is enabled on FortiClient, any non-matching DNS query will be resolved through the local DNS server.
2. If the `dns-mode` is set to `manual`, but the *ipv4-dns-server1* is not configured, the VPN tunnel's DNS will default to 0.0.0.0 and all DNS queries will be routed through the local DNS server.

IPsec DNS suffix

The DNS suffix enables DNS resolution of network resources using their hostnames, without requiring clients to specify their fully qualified domain names (FQDN). This feature is particularly useful in environments where users access internal resources over VPN connections. By appending a DNS suffix to unqualified domain names (such as hostnames), it enables end systems to generate FQDNs required for DNS resolution.

When a DNS suffix is configured for an IPsec tunnel, the configuration is pushed to FortiClient during VPN negotiations and is added to the DNS suffix list for the VPN adapter on the endpoint machine. This setting ensures accurate name resolution for unqualified domain names by appending the specified DNS suffix, which is essential for proper DNS resolution.

DNS suffix configuration is natively supported for IKE version 1. Only one DNS suffix is configurable per IPsec tunnel.

For IKE version 2, DNS suffix is supported when connecting from a Windows FortiClient running 7.4.4 or later. More than one DNS suffix can be configured per IPsec tunnel.

If we assume that a DNS suffix named example.com is set for an IPsec tunnel, when a VPN client performs DNS query for server1 using only its hostname server1, the end system appends the DNS suffix it to make the FQDN server1.example.com. A DNS query is sent using this FQDN to the DNS servers configured for VPN clients.

Split Tunneling:

If split tunneling is enabled on the IPsec tunnel, ensure that the address object used for split tunneling includes the IP address of the DNS server used by VPN clients. This ensures that DNS traffic flows correctly through the VPN adapter. Without this configuration, the DNS suffix is not applied, as DNS queries will bypass the VPN adapter.

Unqualified domains:

The DNS suffix is only appended if VPN clients make DNS queries for unqualified domain names. The DNS suffix is not appended for DNS queries made for FQDNs (such as server1.example.com) or partially qualified domain names (such as server1.example).

Configuration

DNS suffix for IKEv2 can be configured in the GUI on a custom VPN configuration. It requires phase 1 to be a dynamic gateway with mode-cfg enabled:

```
config vpn ipsec phase1-interface
    edit <tunnel_name>
        set mode-cfg enable
        set ike-version 2
        set type dynamic
        set dns-suffix-search <list of suffixes>
    next
end
```

Command	Description
`dns-suffix-search <list of suffixes>`	Specify one or more DNS suffixes in quotes separated by spaces.

Prerequisites:

- In the GUI, the DNS suffix search field is only visible when the IKE version is IKEv2, Mode config is enabled and DNS mode is set to auto OR a valid IP is specified.
- The internal-domain-list setting is not enabled on the phase1-interface configuration.

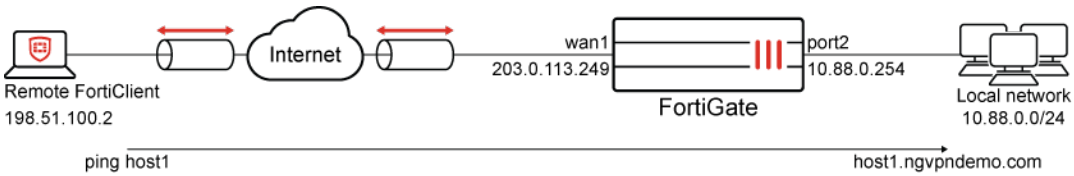
The configuration of a DNS suffix on an IPsec IKEv1 tunnel requires enabling unity-support in the phase 1 configuration of the IPsec tunnel:

```
config vpn ipsec phase1-interface
    edit <name>
        set unity-support enable
        set domain <string>
    next
end
```

Command	Description
`unity-support enable`	Enables unity support to allow pushing DNS suffixes to VPN clients.
`domain <string>`	Specify the DNS suffix that is required to be pushed to VPN clients.

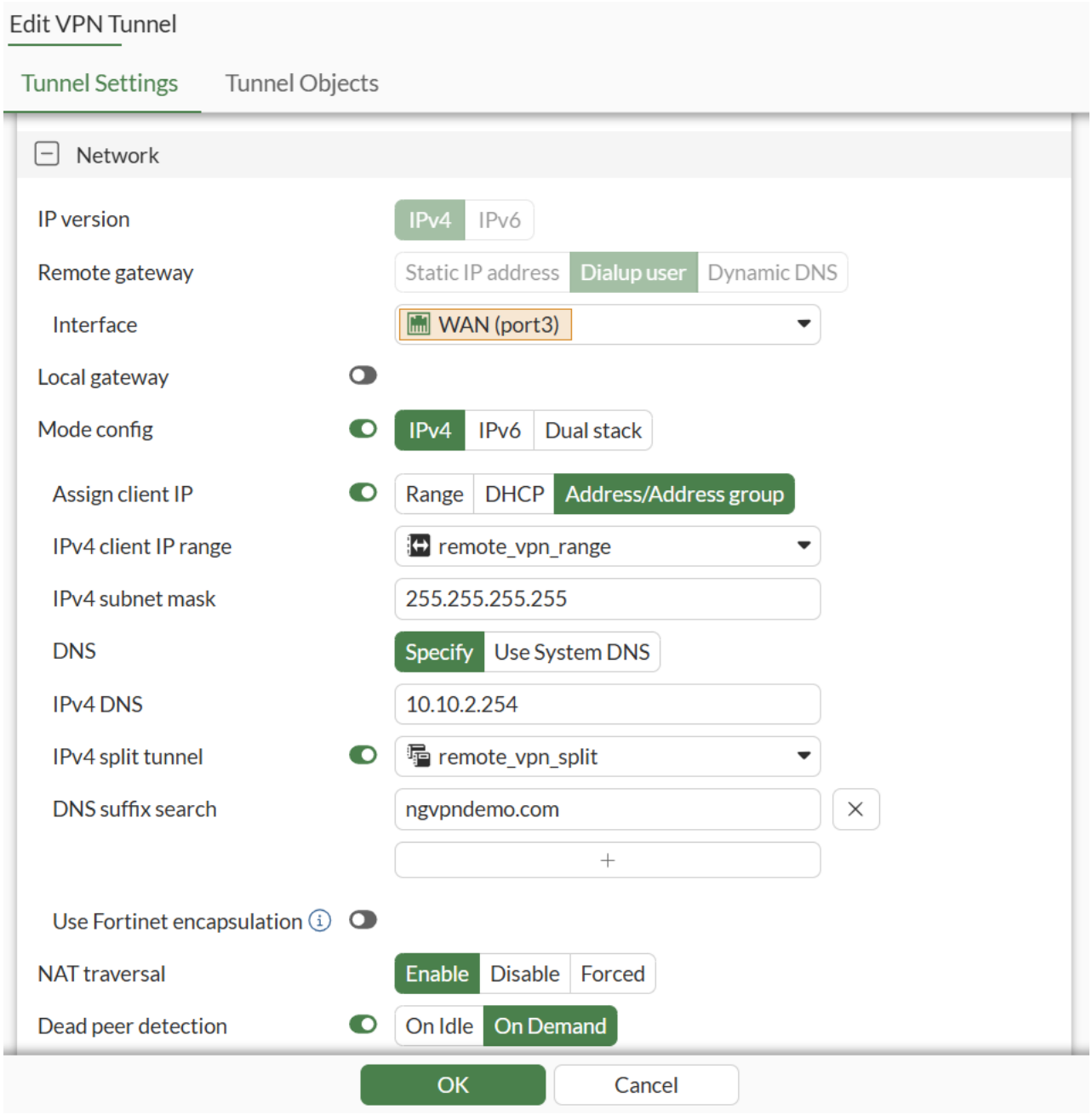
Example NEW

In a dialup VPN configuration, such as the one described in FortiClient as dialup client on page 2973, a DNS suffix is added to help the remote client resolve the names of devices in the local network. Subsequently, the client can reach a device simply by its hostname, and the domain suffix will be appended to it. This example demonstrates a client pinging *host1* and the suffix *ngvpndemo.com* added to the name.



It is assumed that the remote_vpn tunnel was already configured from FortiClient as dialup client on page 2973. The following settings are added to the remote_vpn tunnel

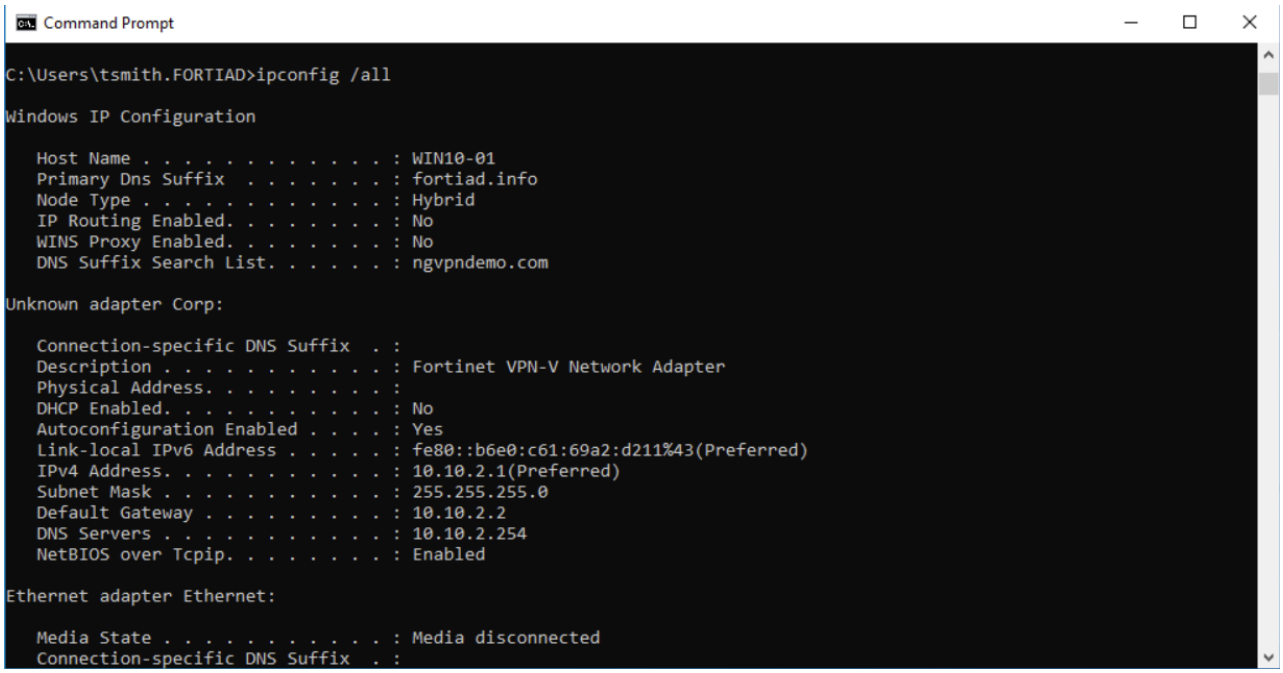
1. Go to *Network* > *Interfaces* and edit the *remote_vpn* interface.
2. Set the *IP address* to *10.10.2.254*.
3. Click *OK*.
4. Go to *VPN* > *VPN Tunnels* and edit the *remote_vpn* tunnel.
5. Set *DNS* to *Specify*, and enter *10.10.2.254*.
6. Add an entry for *DNS suffix search* and set it to *ngvpndemo.com*.



7. Click *OK*.
   This assumes that your VPN interface is configured as a DNS server or can recursively resolve the DNS queries. Use another DNS server if you have another internal DNS server.

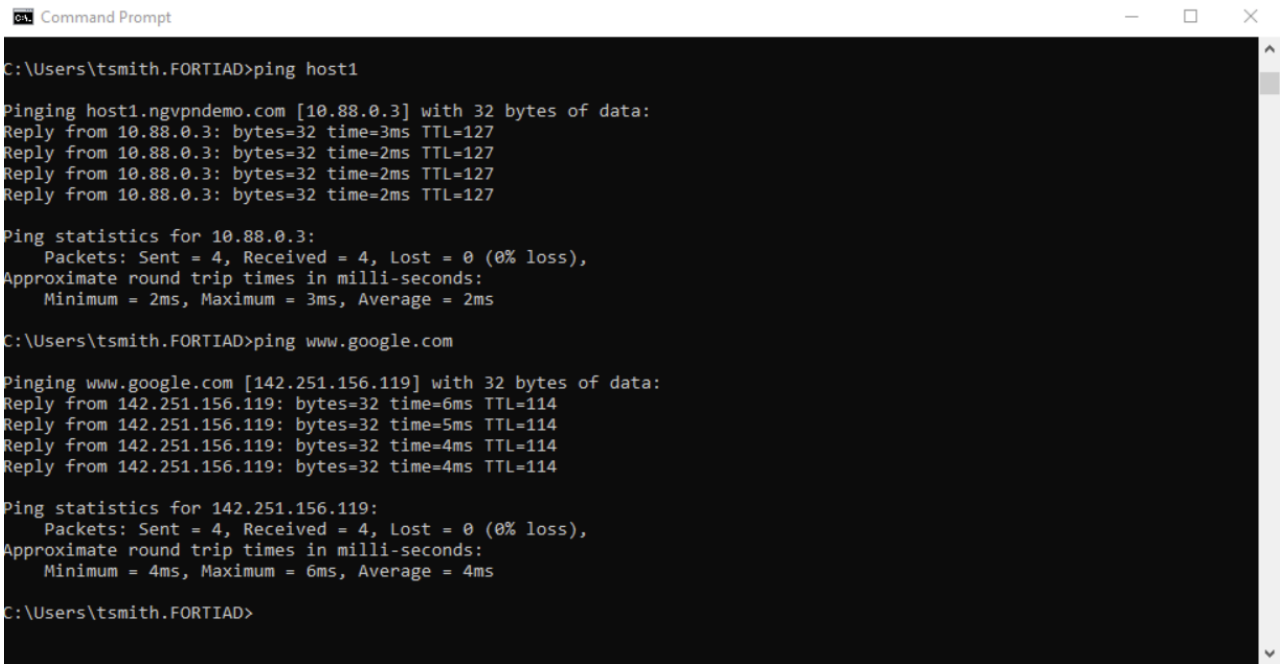
Testing and verification

1. On the client, connect to the tunnel with the user *vpnuser1*.
2. Once connected, open a command prompt and enter `ipconfig /all`.



FortiClient has installed the DNS suffix `ngvpndemo.com` onto this computer.

3. Ping the device `host1`.



The client appended the suffix and resolved `host1.ngvpndemo.com` to `10.88.0.3`.

Pinging www.google.com, the suffix is not added.

Troubleshooting and debugging

When the tunnel is formed, use real-time ike debugs (`diagnose debug application ike -1`) to see the `mode-cfg` message that FortiGate sends to the FortiClient.

Alternatively, go to *VPN > VPN Tunnel*, right-click on the *remote_vpn* tunnel and click *CLI diagnostics*. This will capture the real-time IKE debugs.

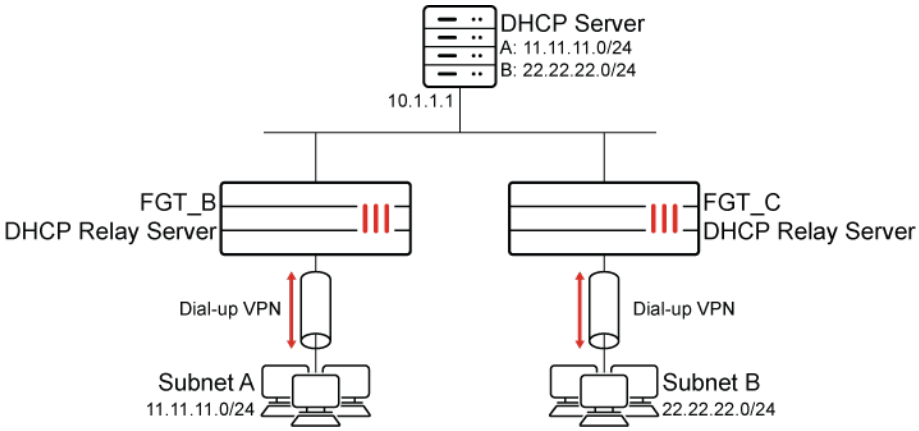
```
ike V=root:0:remote_vpn_0:3: processed INITIAL-CONTACT
ike V=root:0:remote_vpn_0:3: mode-cfg assigned (1) IPv4 address 10.10.2.1
ike V=root:0:remote_vpn_0:3: mode-cfg assigned (2) IPv4 netmask 255.255.255.255
ike V=root:0:remote_vpn_0:3: mode-cfg send (13) 0:10.88.0.0/255.255.255.0:0
ike V=root:0:remote_vpn_0:3: mode-cfg send (3) IPv4 DNS(1) 10.10.2.1
ike V=root:0:remote_vpn_0:3: mode-cfg send INTERNAL_IP6_SUBNET
ike V=root:0:remote_vpn_0:3: mode-cfg IPv6 DNS ignored, no IPv6 DNS servers found
ike V=root:0:remote_vpn_0:3: mode-cfg send (28673) UNITY_SAVE_PASSWD
ike V=root:0:remote_vpn_0:3: client auto-negotiate is disabled
ike V=root:0:remote_vpn_0:3: client-keep-alive is disabled
ike V=root:0:remote_vpn_0:3: mode-cfg send IKEV2_CFG_DNS_SUFFIX_SEARCH 'ngvpndemo.com'
```

Defining gateway IP addresses in IPsec with mode-config and DHCP

For an IPsec tunnel, the gateway IP address (giaddr) can be defined on a DHCP relay agent. Both IPv4 and IPv6 addresses are supported. An IPsec tunnel with mode-config and DHCP relay cannot specify a DHCP subnet range to the DHCP server.

The DHCP server assigns an IP address based on the giaddr set on the IPSec phase1 interface and sends an offer to this subnet. The DHCP server must have a route to the specified subnet giaddr.

Example



To define the gateway IP address on the DHCP relay server:

1. Configure the VPN IPsec phase1 interface:

```
config vpn ipsec phase1-interface
    edit "ipv4"
```

```
        set type dynamic
        set interface "port2"
        set peertype any
        set net-device disable
        set mode-cfg enable
        set proposal des-md5 des-sha1
        set dpd on-idle
        set dhgrp 5
        set assign-ip-from dhcp
        set dhcp-ra-giaddr 11.11.11.1
        set psksecret **********
        set dpd-retryinterval 60
    next
end
```

IPv6 could also be configured:

```
config vpn ipsec phase1-interface
    edit "ipv6"
        set type dynamic
        set interface "port2"
        set peertype any
        set net-device disable
        set mode-cfg enable
        set proposal des-md5 des-sha1
        set dpd on-idle
        set dhgrp 5
        set assign-ip-from dhcp
        set dhcp6-ra-linkaddr 2000:11:11:11::1
        set psksecret **********
        set dpd-retryinterval 60
    next
end
```

2. Enable DHCP proxy and configure the DHCP server IP address:

```
config system settings
    set dhcp-proxy enable
    set dhcp-server-ip "10.1.1.1"
end
```

3. Repeat the above steps for FGT_C and subnet B.

Dialup IPsec VPN using custom TCP port

Dialup IPsec VPN may operate over UDP or TCP, enabling VPN traffic from FortiClient to traverse restrictive firewalls that only permit TCP-based traffic. You can configure an IPsec VPN tunnel to exclusively use UDP, or you can configure the tunnel to automatically switch to TCP mode when the firewall blocks UDP.

In high-latency or congested networks, UDP-based VPN connections may suffer from packet loss or performance degradation. TCP, with its built-in error correction and retransmission mechanisms, enhances the reliability and stability of VPN connections in such environments.

Dialup IPsec over TCP is particularly advantageous in mobile or dynamic settings such as public WiFi, hotel networks, or cellular data where network conditions and restrictions often vary. This feature ensures more seamless and dependable VPN connectivity across a broader range of scenarios.

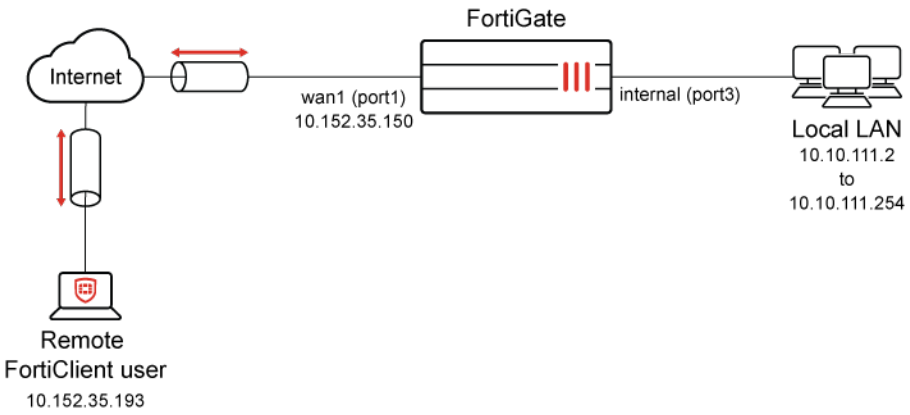
The custom TCP port functionality for IPsec is exclusively supported with IKE version 2 (IKEv2), and does not support NPU offloading.

Example

In this example, FortiGate is configured as a dialup IPsec server using IKE version 2 (IKEv2) and operating on a custom TCP port (5500). IKEv2 is configured to use EAP for user authentication. The initial setup leverages the VPN wizard to create the dialup IPsec tunnel.

On the client side, FortiClient is managed by FortiClient EMS and configured to act as the dialup IPsec client. The client is configured to connect to the FortiGate server over the custom TCP port 5500. This feature requires FortiClient 7.4.1 or later.

For a detailed description of the steps to configure FortiClient EMS to use the custom TCP port 5500 for IPsec VPN connections, see IPsec VPN over TCP.



To enable TCP and configure a custom port in the GUI:

1. Go to *VPN > VPN Tunnels* and select the *Settings* tab.
2. Enable *Allow VPN negotiation over TCP*.
3. Set *TCP port for IKE/IPsec traffic* to port *5500*.
4. Click *Apply*.

To enable TCP and configure a custom port in the CLI:

```
config system settings
    set ike-tcp-service enable
```

```
    set ike-tcp-port 5500
end
```

Command	Description
`ike-tcp-servicer {enable | disable}`	Enable/disable VPN over TCP. This is a per-vdom setting.
`ike-tcp-port <port>`	Set the TCP port for IKE/IPsec traffic (1 - 65535, default = 443).

When using TCP port 443 for IKE/IPsec traffic, GUI access can be affected for interfaces that are bound to an IPsec tunnel when the GUI admin port is also using port 443. To ensure continued functionality, change either the IKE/IPsec port or the administrative access port.

To change the administrative access port:

```
config system global
    set admin-sport <port>
end
```

Command	Description
`admin-sport <port>`	Set the administrative access port for HTTPS (1 - 65535, default = 443).

For port conflicts with ZTNA and Agentless VPN, ZTNA and Agentless VPN will take precedence. To avoid any port conflicts with other services, review the FortiOS Ports guide for other incoming ports used on the FortiGate.

To configure FortiGate as IPsec dialup server using VPN Wizard:

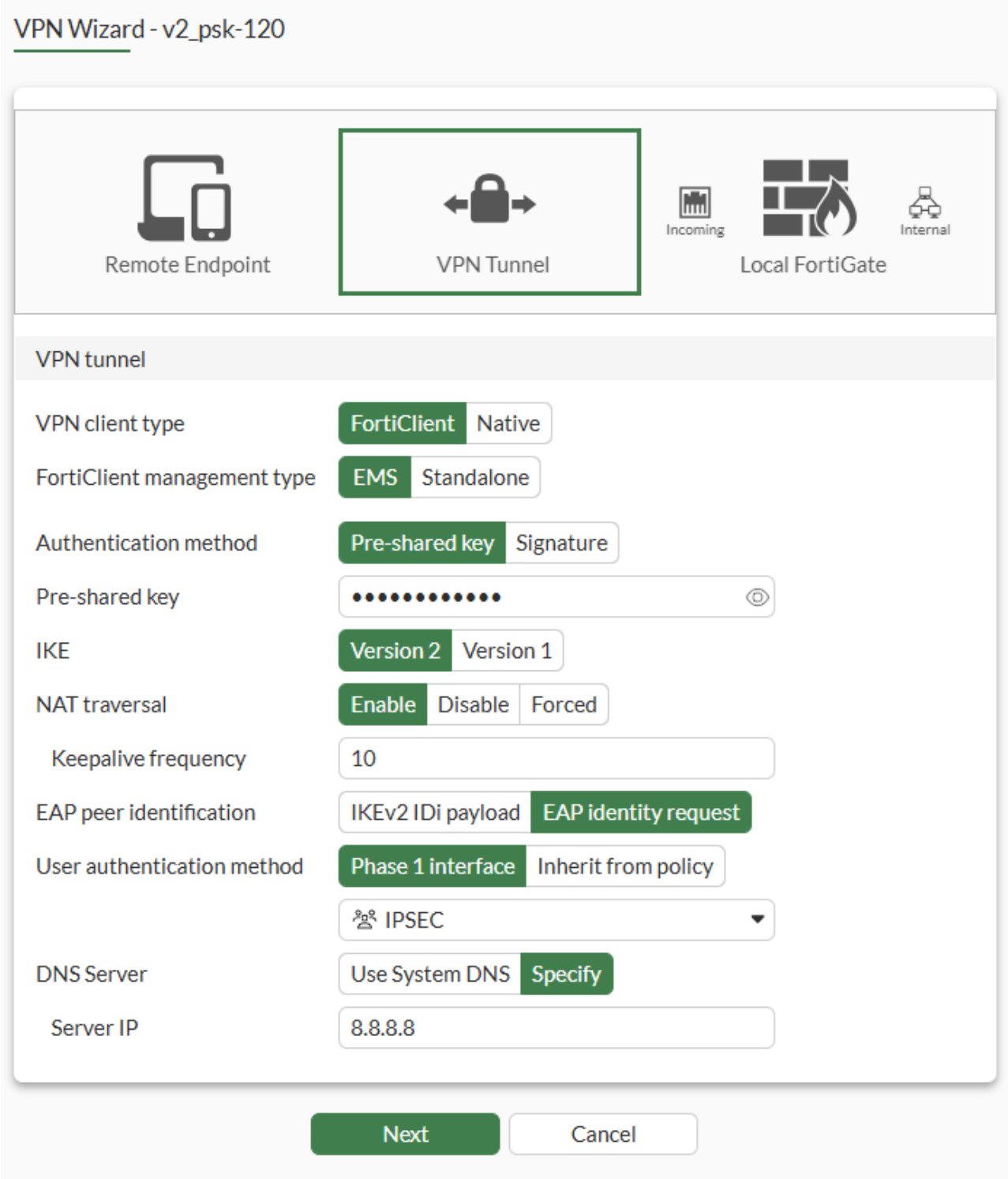
1. Go to *VPN* > *VPN Wizard*, and enter the following:

Field	Value
Tunnel name	v2_psk-120
Select a template	Remote Access

2. Click *Begin*.
3. In the *VPN Tunnel* section, enter the following:

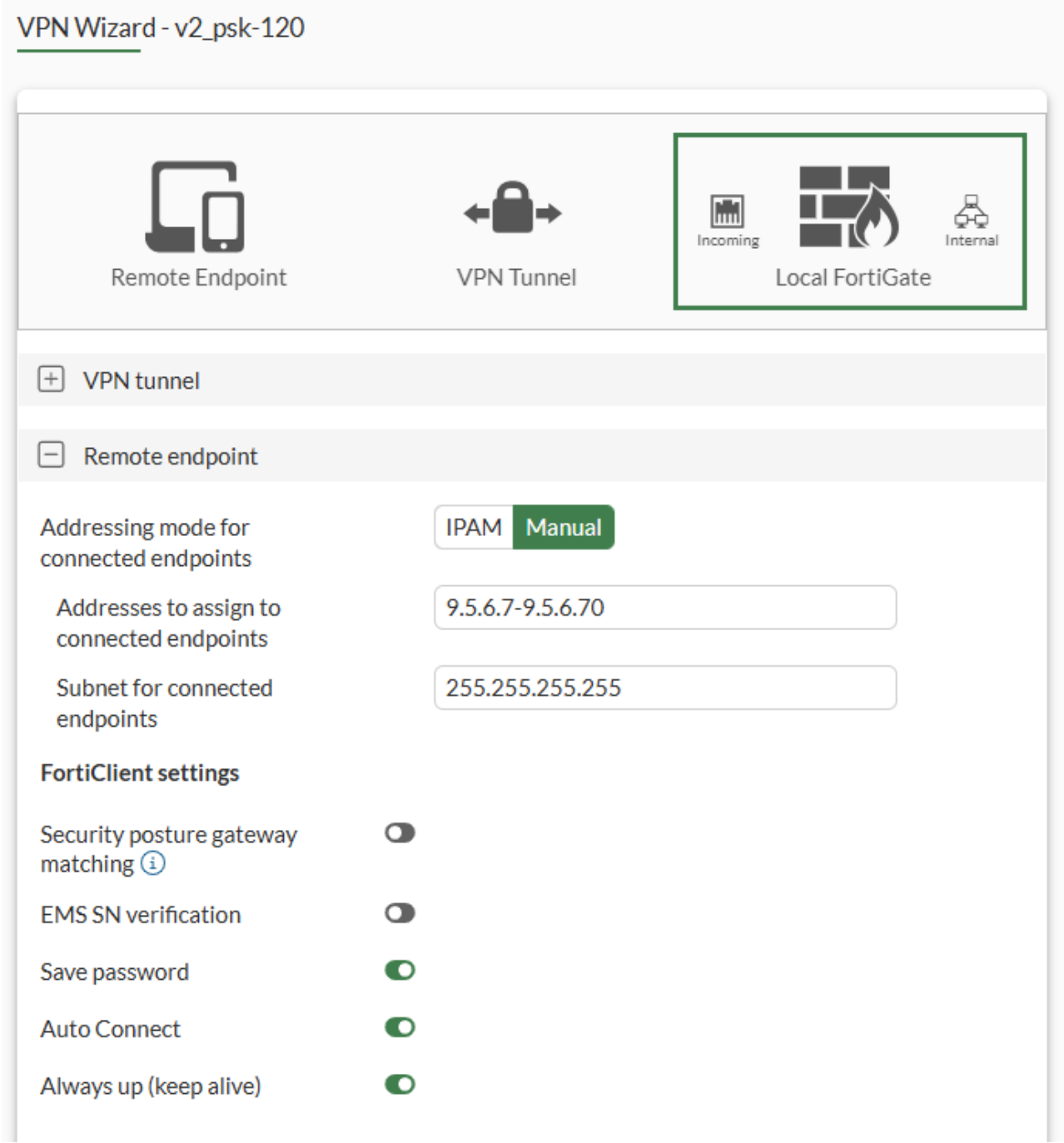
Field	Value
VPN client type	FortiClient
FortiClient management type	EMS
Authentication method	Pre-shared key
Pre-shared key	Enter suitable key
IKE	Version 2

Field	Value
NAT traversal	Enable
Keepalive frequency	10
EAP peer identification	EAP identity request
User authentication method	Phase 1 interface Use dropdown to select user group *IPSEC*. To configure user groups for authentication, see User groups on page 3487. (Optional) To use multiple user groups, select *Inherit from policy*. See Using single or multiple user groups for user authentication on page 3024 for details.
DNS Server	Specify
Server IP	8.8.8.8



4. Click *Next*.
5. Under *Remote Endpoint* section, enter the following:

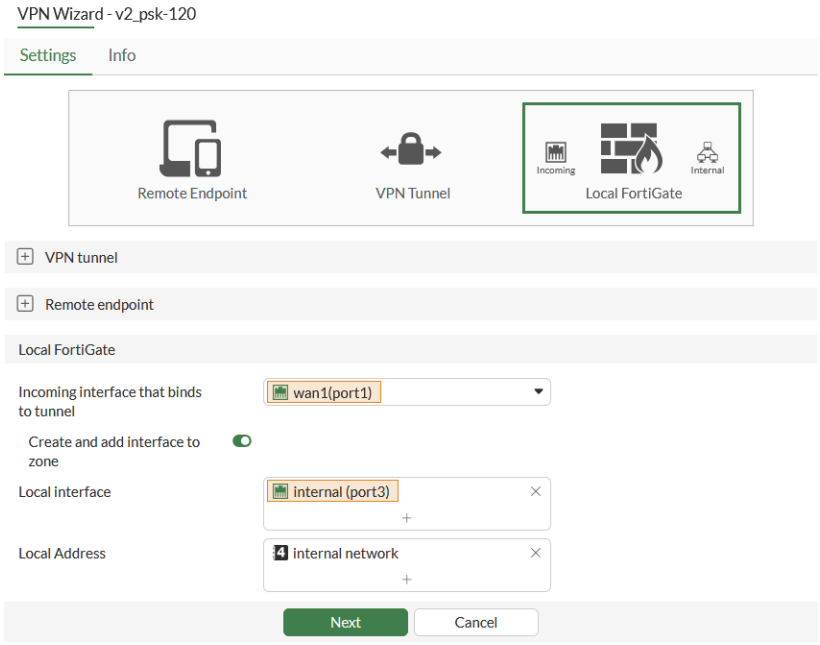
Field	Value
Addressing mode for connected endpoints	Manual
Address to assign to connected endpoints	9.5.6.7-9.5.6.70
Subnet for connected endpoints	255.255.255.255
FortiClient settings	
Security posture gateway matching	Disable
EMS SN verification	Disable
Save password	Enable
Auto Connect	Enable
Always up (keep alive)	Enable



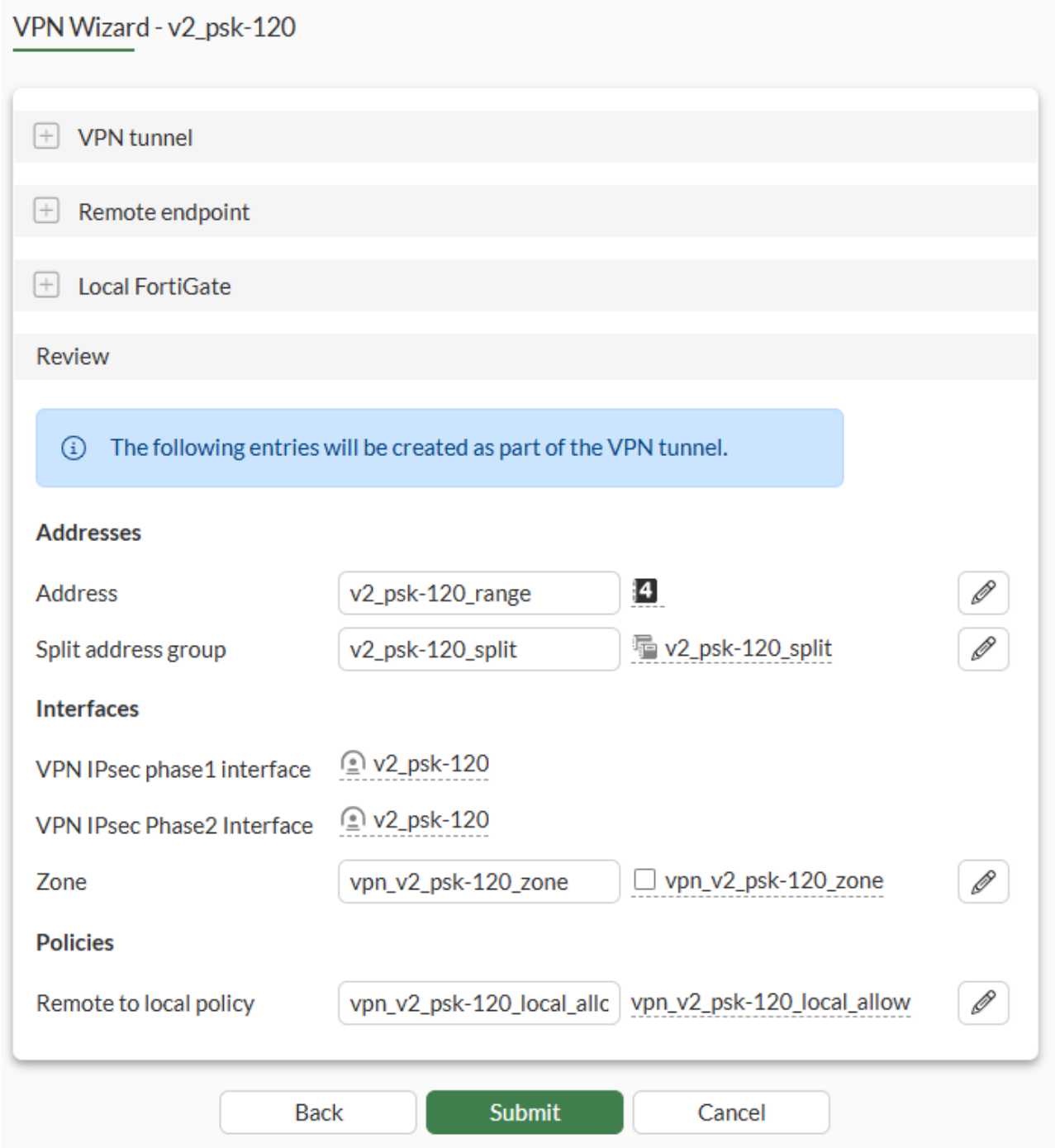
6. Click *Next*.
7. Under *Local FortiGate* section, enter the following:

Field	Value
Incoming interface that binds to tunnel	wan1(port1)

Field	Value
Create and add interface to Zone	Enable
Local interface	internal (port3)
Local Address	internal network

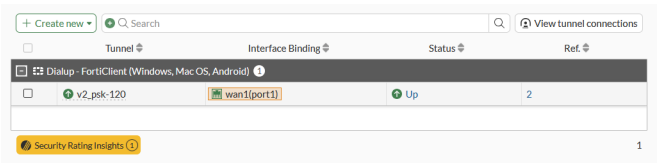


8. Click *Next*.
9. Under *Review* section, review the configuration pending configuration by the wizard.



10. Click *Submit*.

The tunnel is configured and visible under *VPN > VPN Tunnels*.



To configure FortiGate as IPsec dialup server using the CLI:

1. Configure a local user:

```
config user local
    edit "ipsec"
        set type password
        set passwd *****
    next
end
```

2. Configure a local user group:

```
config user group
    edit "IPSEC"
        set member "ipsec"
    next
end
```

3. Configure the internal interface and its address group.

   The internal interface connects to the corporate internal network. Traffic from this interface routes out the IPsec VPN tunnel. Creating an address group for the protected network behind this FortiGate causes traffic to this network group to go through the IPsec tunnel.

```
config system interface
    edit "port3"
        set vdom "root"
        set ip 10.100.55.1 255.255.255.0
        set type physical
        set alias "internal"
    next
end
config firewall address
    edit "internal network"
        set subnet 10.10.111.0 255.255.255.0
    next
end
config firewall addrgrp
    edit "v2_psk-120_split"
        set member "internal network"
    next
end
```

4. Configure the WAN interface and default route.

   The WAN interface is the interface connected to the ISP and it is recommended to configure it with a static IP address to ensure that the IPsec VPN configuration on the branch stays unchanged if the WAN IP changes on the HQ. The IPsec tunnel is established over the WAN interface.

```
config system interface
    edit "port1"
        set vdom "root"
        set ip 10.152.35.150 255.255.255.0
```

```
        set type physical
        set alias "wan1"
    next
end
config router static
    edit 1
        set gateway 10.152.35.151
        set device "port1"
    next
end
```

5. Configure IPsec Phase 1 interface:

```
config vpn ipsec phase1-interface
    edit "v2_psk-120"
        set type dynamic
        set interface "port1"
        set ike-version 2
        set peertype any
        set net-device disable
        set mode-cfg enable
        set ipv4-dns-server1 8.8.8.8
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set dhgrp 20 21
        set eap enable
        set eap-identity send-request
        set authusrgrp "IPSEC"
        set ipv4-start-ip 9.5.6.7
        set ipv4-end-ip 9.5.6.70
        set ipv4-split-include "v2_psk-120_split"
        set save-password enable
        set client-auto-negotiate enable
        set client-keep-alive enable
        set psksecret *******
    next
end
```

One or more user groups for user authentication can be configured under a single IPsec VPN tunnel. See Using single or multiple user groups for user authentication on page 3024 for details.

6. Configure IPsec Phase 2 interface:

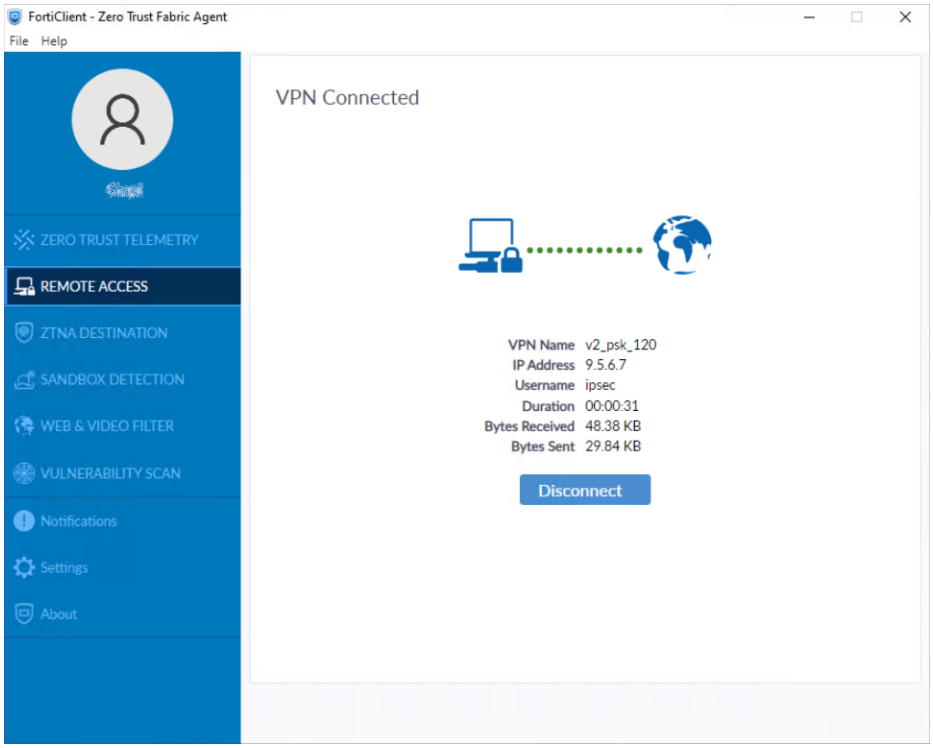
```
config vpn ipsec phase2-interface
    edit "v2_psk-120"
        set phase1name "v2_psk-120"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
    next
end
```

7. Configure firewall policies to allow traffic from IPsec tunnel to internal network:

```
config firewall policy
    edit 1
        set name "vpn_v2_psk-120_local_allow"
        set srcintf "vpn_v2_psk-120_zone"
        set dstintf "port3"
        set action accept
        set srcaddr "v2_psk-120_range"
        set dstaddr "internal network"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

To verify the VPN connection:

1. Using FortiClient, connect to the IPsec VPN gateway.

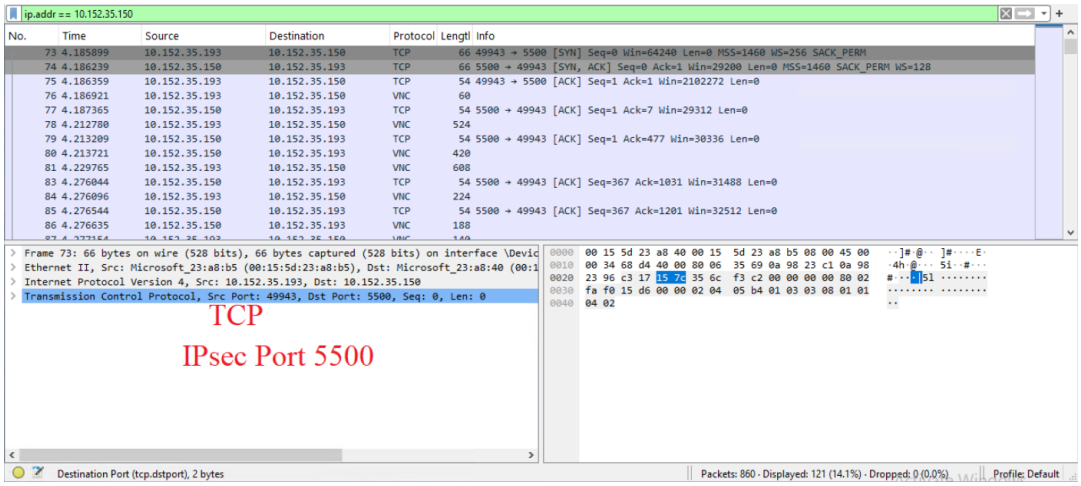


2. On FortiGate, run `diagnose vpn ike gateway list` to verify the IPsec VPN tunnel status. Note that `addr` shows the custom TCP port value, and `transport` shows `TCP`:

```
vd: root/0
name: v2_psk-120_0
version: 2
interface: port1 3
addr: 10.152.35.150:5500 -> 10.152.35.193:54854
tun_id: 9.5.6.7/::10.0.0.23
remote_location: 0.0.0.0
network-id: 0
```

```
transport: TCP
virtual-interface-addr: 169.254.1.1 -> 169.254.1.1
created: 592s ago
eap-user: ipsec
2FA: no
peer-id: 120
peer-id-auth: no
FortiClient UID: B70BAD123010487E86DB102969115E99
assigned IPv4 address: 9.5.6.7/255.255.255.255
nat: me peer
pending-queue: 0
PPK: no
IKE SA: created 1/1  established 1/1  time 80/80/80 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 23 93b6803bff7cff00/f89d6f9965fbf3a7
  direction: responder
  status: established 592-592s ago = 80ms
  proposal: aes256-sha256
  child: no
  SK_ei: f93108f3f8d9a94e-3e0a78289defb329-4d1ae67365f2cb56-e0d471a57ccb4f8d
  SK_er: 58b37cf4d2e96cb3-cb7e334a48905459-ac8e4ff743c86e5c-630454f2e35b97e6
  SK_ai: fc83b139808121a2-1dd68396d804e28d-bd619c0c4778dbda-9a1eb9e6fdf13808
  SK_ar: edad89ee56bf9ecc-81443426c00c78f5-0574d6b71163a43b-d9ebf04c3ae4b87f
  PPK: no
  message-id sent/recv: 0/124
  QKD: no
  lifetime/rekey: 86400/85537
  DPD sent/recv: 00000000/00000000
  peer-id: 120
```

3. Run a packet capture using the packet capture tool on FortiGate GUI under *Network* > *Diagnostic* tab for wan1(port1) interface with TCP port number 5500.



For more information, see Using the packet capture tool on page 1025.

4. (Optional) Run the packet capture using the following command:

```
diagnose sniffer packet wan1 “port 5500” 4 0 l.
```

For more information, see Performing a sniffer trace or packet capture on page 4966.

Because there is no NP offloading for the RFC compliant version of IPsec over TCP, its performance is lower than standard ESP and ESP over UDP, which fully utilizes NP offloading to accelerate the performance.

Matching IPsec tunnel gateway based on address parameters

FortiOS supports remote gateway matching in a dial-up IPsec tunnel connection based on the public source IP geolocation of dial-up clients. See Remote gateway matching on page 2766.

The following example uses the source IP address of the client to match the IPsec tunnel gateway based on the country/region parameters. The client, PC1, is behind a NAT'd device with address 160.106.x.x, which resolves to Canada. Two IPsec tunnels, `TestMatchA` and `TestMatchB`, will be configured on the phase1 interface to test remote gateway country matching. The tunnel that is assigned to Canada will match while the other will not.

This example only includes configurations related to the `remote-gw-match` feature. Other configurations, such as those for the phase2 interface, are omitted for brevity.

To match dialup IPsec tunnel gateway based on country:

1. On the phase1 interface, configure two IPsec tunnels on the FGT VPN Gateway, with `TestMatchA` set to the United States (US) and `TestMatchB` set to Canada (CA):

```
config vpn ipsec phase1-interface
    edit "TestMatchA"
        set type dynamic
        set ike-version 2
        set remote-gw-match geography
        set remote-gw-country "US"
    next
    edit "TestMatchB"
        set type dynamic
        set ike-version 2
        set remote-gw-match geography
        set remote-gw-country "CA"
    next
end
```

2. From PC2, initiate a dial-up VPN connection.

3. On the FortiGate, review the gateway list.

```
# diagnose vpn ike gateway list

vd: root/0
name: TestMatchB_0
version: 2
interface: port5 13
addr: 173.1.1.1:500 -> 160.106.x.x:500
tun_id: 160.106.x.x/::10.0.0.35
remote_location: 0.0.0.0
network-id: 0
created: 162s ago
peer-id: C = US, ST = California, L = Sunnyvale, O = Fortinet, OU = FortiGate, CN =
FG201EXXXXXXXXX, emailAddress = support@fortinet.com
peer-id-auth: yes
PPK: no
IKE SA: created 1/1  established 1/1  time 10/10/10 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 10884 54ab158a7d192cbc/ef82ff5e91d72f59
  direction: responder
  status: established 162-162s ago = 10ms
  proposal: aes128-sha256
  child: no
  SK_ei: f1d74e0f026674b1-7687368f42305b31
  SK_er: b693bc06ea670ad3-643a6562cca05617
  SK_ai: 7edea8cfc3f82ce0-9a8ac426e05205b5-b71efc76d940589c-e9725108e7309cf5
  SK_ar: da3eaa37cc171369-1261fc51d4404bc7-c38bbaa9efa1bcfe-de3c285f3eb18617
  PPK: no
  message-id sent/recv: 0/8
  lifetime/rekey: 86400/85967
  DPD sent/recv: 00000000/00000000
  peer-id: C = US, ST = California, L = Sunnyvale, O = Fortinet, OU = FortiGate, CN =
FG201EXXXXXXXXX, emailAddress = support@fortinet.com
```

Since the client IP address is geolocated in Canada, `TestMatchB` matched.

4. Change the country assignments of the two IPsec tunnels so that `TestMatchA` is set to Canada (CA) and `TestMatchB` is set to China (CN):

```
config vpn ipsec phase1-interface
    edit "TestMatchA"
        set type dynamic
        set ike-version 2
        set remote-gw-match geography
        set remote-gw-country "CA"
    next
    edit "TestMatchB"
        set type dynamic
        set ike-version 2
        set remote-gw-match geography
        set remote-gw-country "CN"
```

```
    next
end
```

5. From PC2, initiate a dial-up VPN connection.
6. On the FortiGate, review the gateway list again.

```
# diagnose vpn ike gateway list

vd: root/0
name: TestMatchA_0
version: 2
interface: port5 13
addr: 173.1.1.1:500 -> 160.106.x.x:500
tun_id: 160.106.x.x/::10.0.0.37
remote_location: 0.0.0.0
network-id: 0
created: 1856s ago
peer-id: C = US, ST = California, L = Sunnyvale, O = Fortinet, OU = FortiGate, CN =
FG201EXXXXXXXXX, emailAddress = support@fortinet.com
peer-id-auth: yes
PPK: no
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 10886 fec7cd972847a2ac/0c1ee0b54ddc155e
  direction: responder
  status: established 1856-1856s ago = 0ms
  proposal: aes128-sha256
  child: no
  SK_ei: 7e8c8d05a6a9adab-bfcf9ff2705e8965
  SK_er: bdd6ee61fc38cd81-202b5f142cefa5ce
  SK_ai: 30f905722136bbce-0c96d365dd52957c-3d05b83efd026140-831fbc76fc677456
  SK_ar: 4363f29c44d49f30-7d798777766efb09-aca39e8a8ca0e6d7-5b83c113e46b339d
  PPK: no
  message-id sent/recv: 0/89
  lifetime/rekey: 86400/84273
  DPD sent/recv: 00000000/00000000
  peer-id: C = US, ST = California, L = Sunnyvale, O = Fortinet, OU = FortiGate, CN =
FG201EXXXXXXXXX, emailAddress = support@fortinet.com
```

Since the client IP address is geolocated in Canada, `TestMatchA` matched.

Enforcing security posture verification before dial-up VPN connection

IPsec dial-up VPN can enforce ZTNA security posture verification before clients establish a VPN tunnel. This ensures the client has appropriate security posture before FortiGate grants a tunnel connection. The endpoint must be managed by FortiClient EMS, and its tags are synchronized with both EMS and FortiGate.

When a tag is defined on an IKEv2 IPsec tunnel, only the client IP addresses resolved by the tag can establish connection to the tunnel. When multiple tags are used, tags are checked sequentially until a match is made. If no tags match, then the client cannot establish a VPN connection.

Furthermore, to enhance resiliency, FortiClient endpoints can send security posture tags in JSON Web Token (JWT) format directly to FortiGate. The JWT tags are received during the VPN establishment process after authentication. They are also received periodically on tag changes and tag token expiration. This provides a backup mechanism for FortiGate to verify tags when FortiClient EMS is unreachable or slow to respond. FortiGate continues to prioritize querying FortiClient EMS for tag verification as the primary method.

JWT format security posture tags requires FortiClient 7.4.5 and FortiClient EMS 7.4.4 or above. See Improvements to tag compliance check for VPN connection for more background on FortiClient settings.

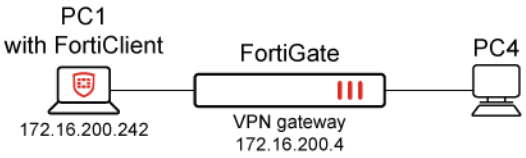
CLI syntax

```
config vpn ipsec phase1-interface
    edit <name>
        set ike-version 2
        set remote-gw-match {any | ipmask | iprange | geography | ztna}
        set remote-gw-ztna-tags <IPv4 ZTNA posture tags>
    next
end
```

When `set remote-gw-match ztna` is enabled, `remote-gw-ztna-tags` can be configured.

Example 1: basic configurations

A PC (172.16.200.242) makes a connection to the dial-up VPN gateway (172.16.200.4). The following example configuration and outputs show a successful connection with a matching ZTNA security posture tag (`EMS1_ZTNA_all_registered_clients`) and an unsuccessful connection when a ZTNA security posture tag cannot be matched.



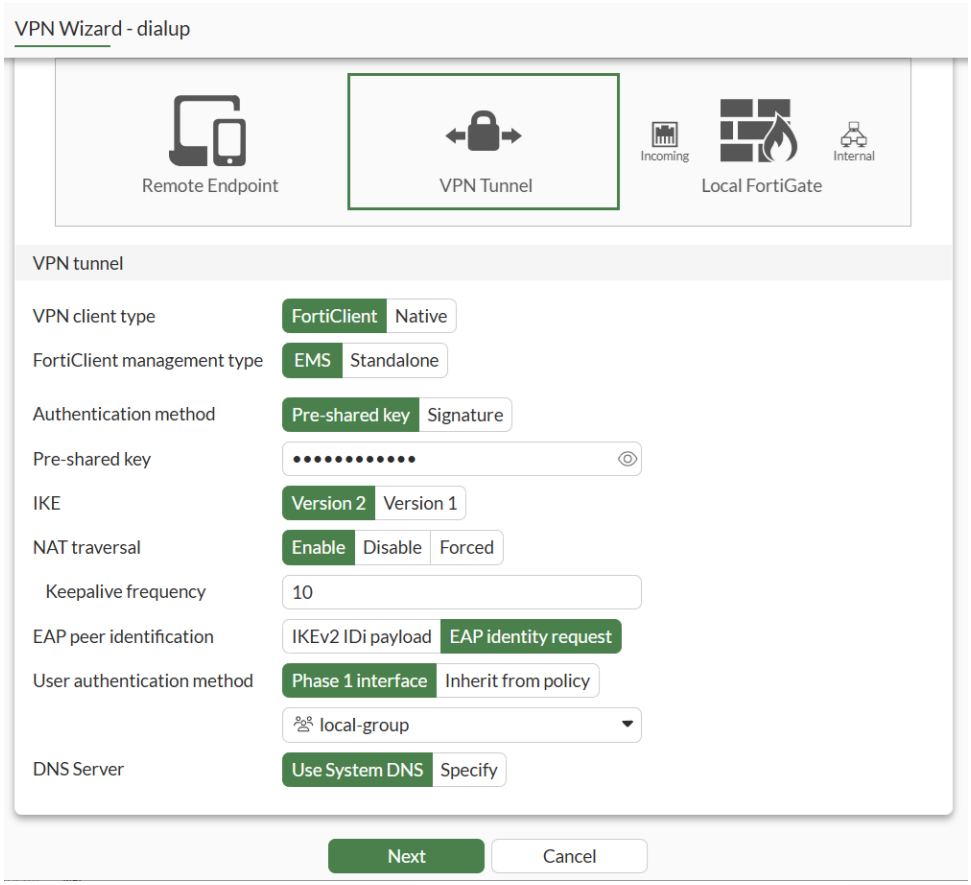
It is assumed that the following mandatory pre-configurations are complete before configuring VPN:

- FortiGate has established a connection with FortiClient EMS and has synchronized the ZTNA security posture tags, including the `EMS1_ZTNA_all_registered_clients` tag.
- FortiClient is registered to EMS and has the ZTNA security posture tag (`EMS1_ZTNA_all_registered_clients`).

In this example, the VPN Wizard is used in the GUI to configure the IPsec tunnel and use the ZTNA tag EMS1_ZTNA_all_registered_clients. This requires that IKEv2 is enabled. The example mostly uses the default settings in the wizard, but these can be modified as required by your use case.

To configure the VPN gateway with the GUI:

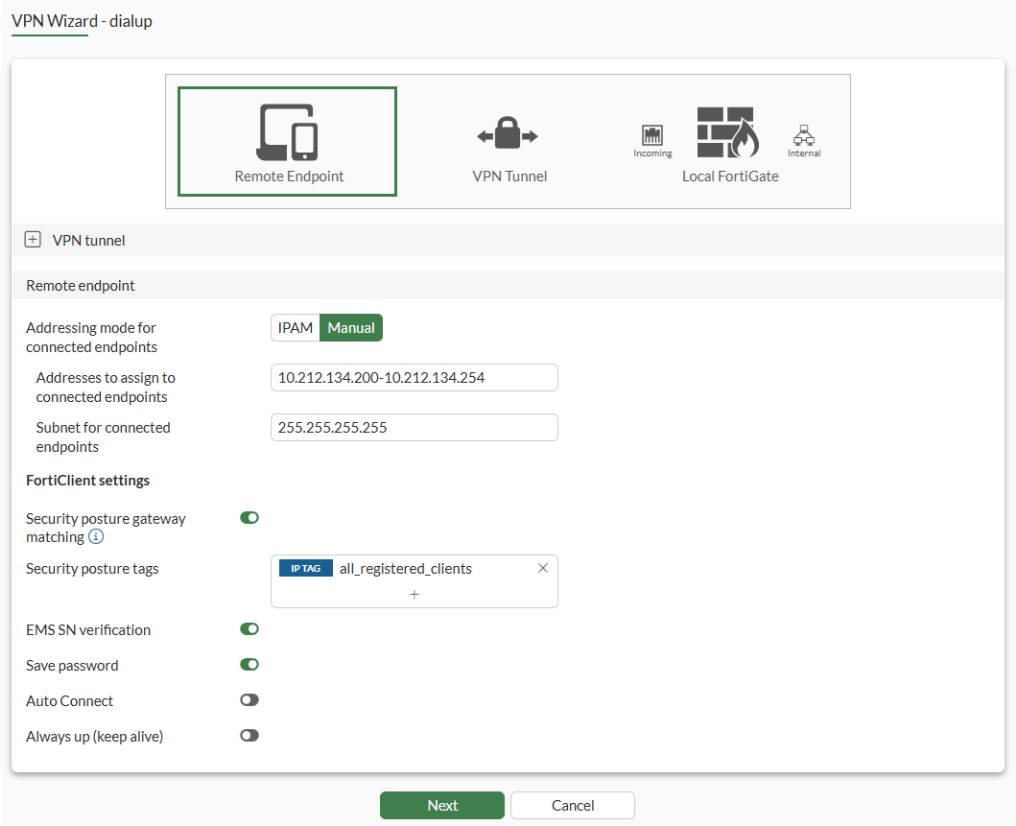
1. Go to *VPN* > *VPN Wizard*.
2. Enter the *Tunnel name*, such as *dialup*.
3. Set *Select a Template* to *Remote Access* then click *Begin*.
4. Configure the *VPN tunnel*:



VPN client type	FortiClient
FortiClient management typ	EMS
Authentication method	Pre-shared key
Pre-shared key	Enter the pre-shared key.
IKE	Version 2
NAT traversal	Enable
EAP peer identification	EAP identity request
User authentication method	Phase1 interface Select a user group for user authentication from the drop-down list.
DNS Server	Use System DNS

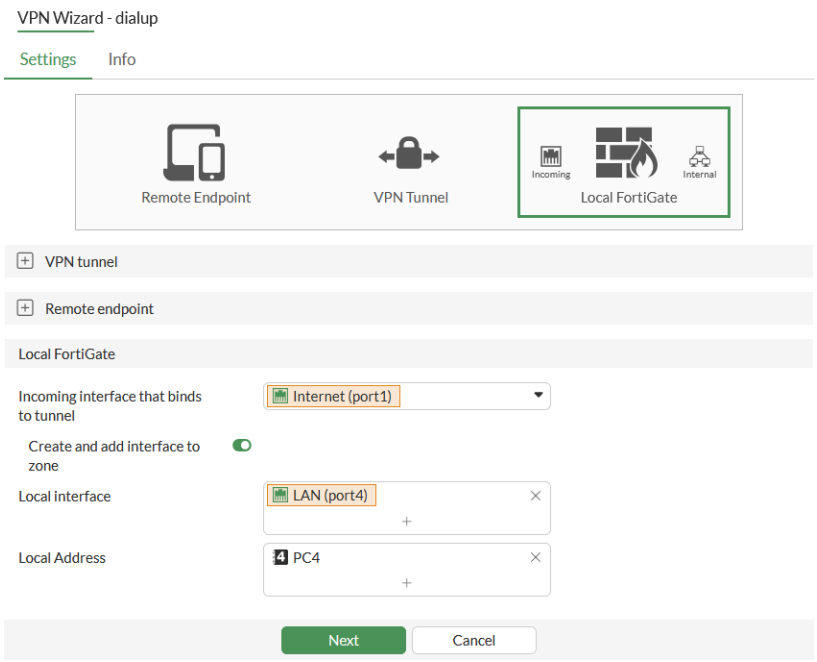
5. Click *Next*.

6. Configure the *Remote endpoint*:



Addressing mode for connected endpoints	IPAM or Manual. This example uses Manual.
Addresses to assign to connected endpoints	Enter the VPN IP address range that needs to be assigned to VPN endpoints: 10.212.134.200-10.212.134.254
Subnet for connected endpoints	255.255.255.255
Security posture gateway matching	Enable
Security posture tags	Add one or more ZTNA Tags, for example: IP TAG all_registered_clients
EMS SN verification	Enable
Save password	Enable
Auto Connect	Disable
Always up (keep alive)	Disable

7. Click *Next*.
8. Configure the *Local FortiGate*:



Incoming interface that binds to tunnel	Select the WAN Interface. In this example, that is the interface with IP address 172.16.200.4.
Create and add interface to zone	Enable
Local interface	Select the LAN interface. In this example, that is the interface that is connected to PC4.
Local Address	Select a local address to allow access from the VPN. In this example, that is the IP address of PC4.

9. Click *Next*.
10. Review the configuration. If everything looks correct, click *Submit*.

To configure the VPN gateway with the CLI:

```
config firewall address
    edit "dialup_range"
        set type iprange
        set start-ip 10.212.134.200
        set end-ip 10.212.134.254
    next
end
config vpn ipsec phase1-interface
    edit "dialup"
        set type dynamic
        set interface "port1"
        set ike-version 2
        set peertype any
```

```
        set net-device disable
        set mode-cfg enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set eap enable
        set eap-identity send-request
        set authusrgrp "local-group"
        set ems-sn-check enable
        set remote-gw-match ztna
        set assign-ip-from name
        set dns-mode auto
        set ipv4-split-include "dialup_split"
        set ipv4-name "dialup_range"
        set save-password enable
        set psksecret xxxxxxxx
        set remote-gw-ztna-tags "EMS1_ZTNA_all_registered_clients"
    next
end
config vpn ipsec phase2-interface
    edit "DY"
        set phase1name "dialup"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
    next
end
```

Results

When a client (172.16.200.242) has the appropriate ZTNA security posture tag, it is synchronized to FortiClient EMS and FortiGate.

To view the IP address that is resolved to tag:

```
# diagnose firewall dynamic list EMS1_ZTNA_all_registered_clients

CMDB name: EMS1_ZTNA_all_registered_clients
TAG name: all_registered_clients
EMS1_ZTNA_all_registered_clients: ID(6)
        ADDR(172.16.200.242)
Total IP dynamic range blocks: 0.
Total IP dynamic addresses: 1.
```

The address resolves to the IP address that is configured on the client endpoint. It does not resolve to the NAT'd public IP address when the client is behind NAT. In other words, the address resolves to the `IP Address` in the following output, and not the `Public IP Address`:

```
# diagnose endpoint ec-shm list

Record #0:
                IP Address = 172.16.200.242
                MAC Address = **:**:**:**:67:74
```

```
                MAC list =
                VDOM = root (0)
                TOKEN VDOM =  (-1)
                EMS serial number: FCTEMS**********
                EMS tenant id: ************1EDB589FBC4626
                Client cert SN: ************60ACA9FE283DEEC3B
                Public IP address: ************
                Quarantined: no
                Online status: online
                Registration status: registered
                On-net status: on-net
                Gateway Interface: port1
                FortiClient version: 7.2.4
                …
                Number of Routes: (1)
                        Gateway Route #0:
                                - IP:172.16.200.242, MAC: **:**:**:**:67:74, VPN: no
                                - Interface:port1, VDOM:root (0), SN: FG5H************
```

From the PC1 client, establish a tunnel with the FortiGate VPN gateway. When enabled, the following debug information displays the output when the security posture tag is matched:

```
# diagnose debug application ike -1
…
ike V=root:0:DY:155: received FCT-UID : 6108A9179A5C40D7BD57504E15114C1F
ike V=root:0:DY:155: received EMS SN : FCTEMS**********
ike V=root:0:DY:155: received EMS tenant ID : ************1EDB589FBC4626
ike V=root:0:DY:155: peer identifier IPV4_ADDR 172.16.200.242
ike V=root:0:DY:155: re-validate gw ID
ike V=root:0:DY:155: gw validation OK
ike V=root:0:DY:155: responder preparing EAP identity request
…
ike V=root:0:FCT:22: responder received AUTH msg
ike V=root:0:FCT:22: peer identifier IPV4_ADDR 172.16.200.242
ike V=root:0:FCT:22: auth verify done
ike V=root:0:FCT:22: responder AUTH continuation
ike V=root:0:FCT:22: authentication succeeded
ike V=root:0:FCT:22: EMS SN check passed
ike V=root:0:FCT:22: EMS ZTNA tags check success ZTNA_all_registered_clients
ike V=root:0:FCT:22: responder creating new child
```

The following tunnel output indicates a dial-up tunnel has been established:

```
# diagnose vpn ike gateway list
vd: root/0
name: dialup_0
version: 2
interface: port1 9
addr: 172.16.200.4:500 -> 172.16.200.242:500
tun_id: 10.212.134.200/::10.0.0.5
remote_location: 0.0.0.0
network-id: 0
```

```
transport: UDP
created: 34s ago
eap-user: userc
2FA: no
peer-id: 172.16.200.242
peer-id-auth: no
FortiClient UID: 6108A9179A5C40D7BD57504E15114C1F
assigned IPv4 address: 10.212.134.200/255.255.255.255
pending-queue: 0
PPK: no
IKE SA: created 1/1  established 1/1  time 10/10/10 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms
…

# diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=dialup ver=2 serial=1 172.16.200.4:0->0.0.0.0:0 nexthop= tun_id=10.0.0.1 tun_id6=::10.0.0.1
status=up dst_mtu=0 weight=1
bound_if=9 real_if=0 lgwy=static/1 tun=intf mode=dialup/2 encap=none/552 options[0228]=npu frag-
rfc  role=primary accept_traffic=1 overlay_id
=0

proxyid_num=0 child_num=1 refcnt=3 ilast=42978277 olast=42978277 ad=/0
stat: rxp=1290 txp=40 rxb=65588 txb=34472
dpd: mode=on-demand on=0 status=ok idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
------------------------------------------------------
name=dialup_0 ver=2 serial=5 172.16.200.4:0->172.16.200.242:0 nexthop=172.16.200.242 tun_
id=10.212.134.200 tun_id6=::10.0.0.5 status=up dst_mtu=1
500 weight=1
bound_if=9 real_if=9 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/74408 options[122a8]=npu
rgwy-chg frag-rfc  run_state=0 role=primary
accept_traffic=1 overlay_id=0

parent=dialup index=0
…
```

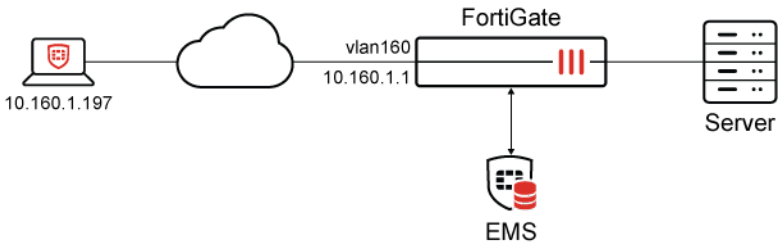
In the scenario where a client without a matching tag tries to establish a tunnel, the following debug information indicates that the tunnel cannot match the remote IP of the client.

```
ike V=root:0:FCT:23: responder received AUTH msg
ike V=root:0:FCT:23: peer identifier IPV4_ADDR 172.16.200.242
ike V=root:0:FCT:23: auth verify done
ike V=root:0:FCT:23: responder AUTH continuation
ike V=root:0:FCT:23: authentication succeeded
ike V=root:0:FCT:23: EMS SN check passed
ike V=root:0:FCT:23: EMS ZTNA tags check failure, abort
ike V=root:FCT Negotiate SA Error: EMS ZTNA tags check failed
ike V=root:0:FCT:23: sending IKE_AUTH ZTNA TAGS check failure notification response
```

Example 2: EMS unreachable

In this example, FortiClient EMS is configured to use ZTNA token, which enables FortiClient to send security posture tags in JWT format.

A remote employee connects to the FortiGate VPN gateway while the FortiClient EMS connection is temporarily unavailable. FortiGate falls back to verifying the JWT security posture tag and allows the VPN connection to be established.



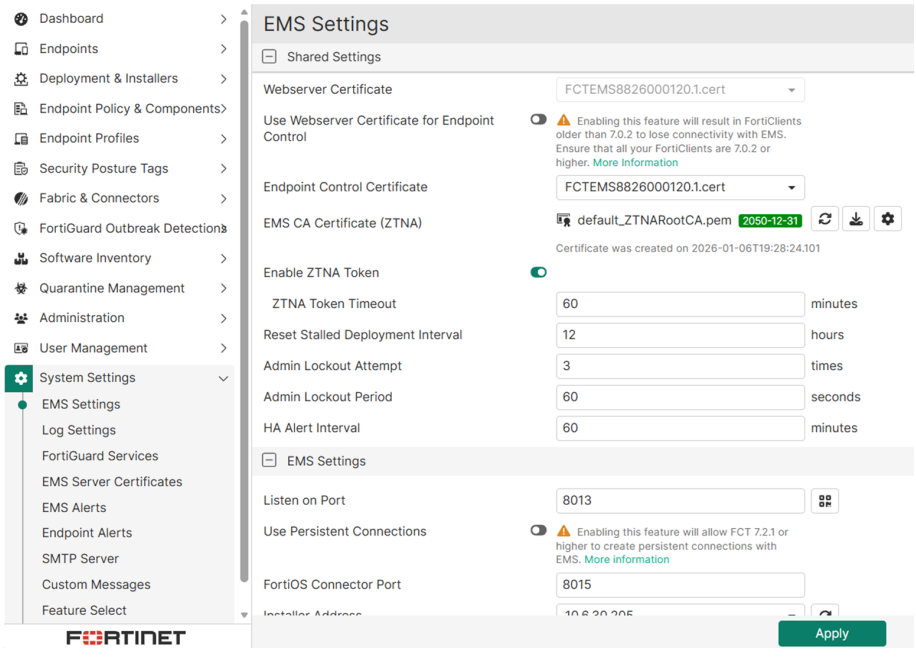
This example includes the following sections:

- Configuring FortiClient EMS on page 3182
- Configuring FortiGate on page 3183
- Verifying on page 3184

Configuring FortiClient EMS

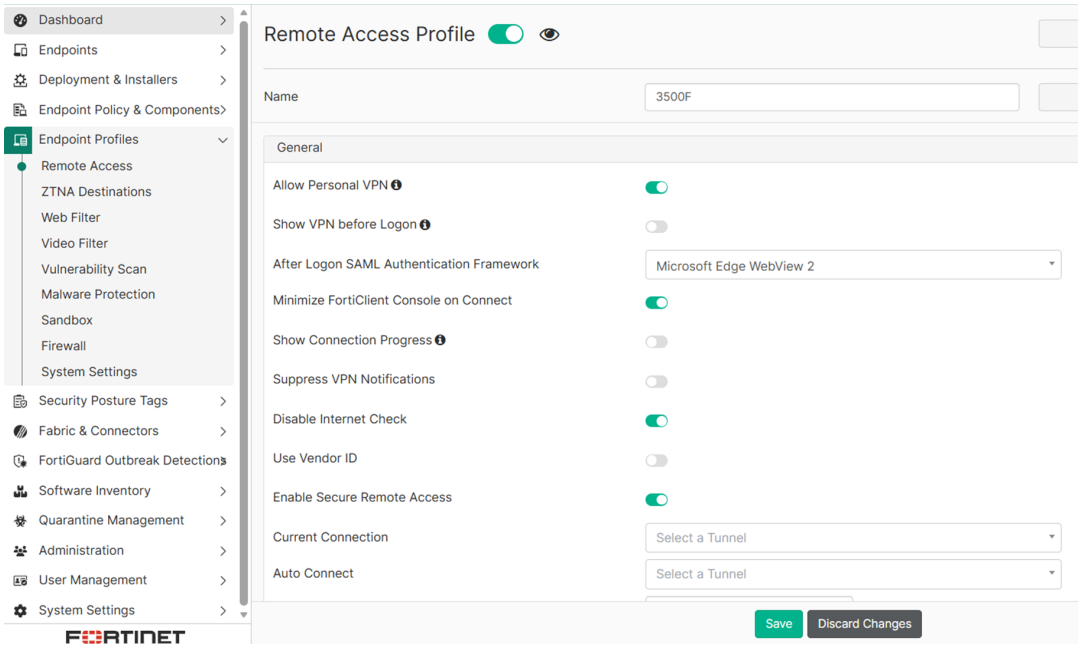
To configure FortiClient EMS:

1. On FortiClient EMS, go to *System Settings* > *EMS Settings*, and enable *Enable ZTNA Token*.
2. Set the *ZTNA Token Timeout* and interval time.



3. Go to *Endpoints Profiles* > *Remote Access*, and edit a Remote Access profile.

4. Under *General*, enable *Enable Secure Remote Access*.



5. Under *VPN Tunnels*, add or edit a tunnel.
6. In *Advanced Settings*, select the security posture tags.
7. Save the profile and push to FortiClient endpoints.

Configuring FortiGate

These steps assume FortiClient EMS fabric settings are already configured, and security posture tags have been synchronized to the FortiGate.

To configure the VPN tunnel on FortiGate:

1. On FortiGate, configure VPN phase1. Enable `remote-gw-match-ztna` and choose your ZTNA tags:

```
config vpn ipsec phase1-interface
    edit "FCT_EMS_TAG_ph2"
        set type dynamic
        set interface "vlan160"
        set ike-version 2
        set peertype any
        set net-device disable
        set mode-cfg enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set dpd on-idle
        set dhgrp 20 21
        set eap enable
        set eap-identity send-request
        set authusrgrp "local_group"
        set ems-sn-check enable
```

```
        set remote-gw-match ztna
        set ipv4-start-ip 10.139.1.2
        set ipv4-end-ip 10.139.1.20
        set dns-mode auto
        set ipv4-split-include "all"
        set save-password enable
        set psksecret <password>
        set dpd-retryinterval 60
        set remote-gw-ztna-tags "EMS3_ZTNA_a"
    next
end
```

2. Configure VPN phase2:

```
config vpn ipsec phase2-interface
    edit "FCT_EMS_TAG_ph2"
        set phase1name "FCT_EMS_TAG_ph2"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
    next
end
```

3. Configure a firewall policy:

```
config firewall policy
    edit 14
        set name "FCT_EMS_TAG_ph2"
        set srcintf "FCT_EMS_TAG_ph2"
        set dstintf "mgmt1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

Verifying

To verify:

1. Initiate a VPN connection from the remote FortiClient endpoint while the EMS connection failed.
2. During the VPN negotiation, use these commands to view the real-time debugs:

```
# diagnose debug application ike -1
# diagnose debug enable
ike V=root:0: udp comes 10.160.1.197:55846->10.160.1.1:4500,ifindex=1123,vrf=0,len=466....
…
ike V=root:0:FCT_EMS_TAG_ph2:FCT_EMS_TAG_ph2: created connection: 0x55e62c7d8a50 1123
10.160.1.1->10.160.1.197:55846.
```

```
,,,
ike V=root:0:FCT_EMS_TAG_ph2:40481: responder received AUTH msg
ike V=root:0:FCT_EMS_TAG_ph2:40481: processing notify type INITIAL_CONTACT
ike V=root:0:FCT_EMS_TAG_ph2:40481: peer identifier IPV4_ADDR 10.160.1.197
ike V=root:0:FCT_EMS_TAG_ph2:40481: eap-peer=yes
ike V=root:0:FCT_EMS_TAG_ph2:40481: re-validate gw ID
ike V=root:0:FCT_EMS_TAG_ph2:40481: gw validation OK
ike V=root:0:FCT_EMS_TAG_ph2:40481: responder preparing EAP identity request
…
ike V=root:0:FCT_EMS_TAG_ph2:40481: authentication succeeded
ike V=root:0:FCT_EMS_TAG_ph2:40481: processing notify type FORTICLIENT_CONNECT
ike V=root:0:FCT_EMS_TAG_ph2:40481: received FCT data len = 309, data = 'VER=1
FCTVER=7.4.5.1949
UID=F8A6AFF3531843B4972CBDBEEEB30E3C
IP=10.160.1.197
MAC=00-0c-29-2c-4d-d7;00-0c-29-2c-4d-e1;00-0c-29-2c-4d-eb;
HOST=DESKTOP-EMS-TAG-PH2
USER=tester
OSVER=Microsoft Windows 8.0 , 64-bit (build 9200)
REG_STATUS=0
EMSSN=FCTEMS8826000120
EMSID=00000000000000000000000000000000
FCTTAGS=1
```

The tag `FCTTAGS=1` indicates that FortiClient has JWT enabled and is sending security posture tags to FortiGate directly.

```
ike V=root:0:FCT_EMS_TAG_ph2:40481: received FCT-UID : F8A6AFF3531843B4972CBDBEEEB30E3C
…
ike V=root:0:FCT_EMS_TAG_ph2_0:40481: processing notify type ZTNA_TAG
```

This indicates FortiClient JWT has been received by the FortiGate:

```
...
ike V=root:0:FCT_EMS_TAG_ph2_0:40481: received ZTNA tags update token (len=791)
data=eyJhbGciOiJSUzI1NiIsInR5cCI6IkpXVCJ9.eyJlbmRwb2ludCI6eyJ1aWQiOiJGOEE2QUZGMzUzMTg0M0I0OTcy
Q0JEQkVFRUIzMEUzQyIsInRhZ3MiOlt7Im5hbWUiOiJhIiwidHlwZSI6MSwibGV2bCI6bnVsbH0seyJuYW1lIjoiYWxsX3
JlZ2lzdGVyZWRfY2xpZW50cyIsInR5cGUiOjEsImxldmwiOm51bGx9LHsibmFtZSI6IkxvdyIsInR5cGUiOjMsImxldmwi
Om51bGx9XSwibndpZnMiOm51bGwsInp0bmFfY2VydF9zbiI6IkIwOUQ5NzE3ODEyMUI1ODhFRDMwNTUxMzc1Nzc1MjJDN0
I2ODlCNzkiLCJvd25lciI6bnVsbH0sImV4cCI6MTc2OTc2MTM4NywiaWF0IjoxNzY5NzU3Nzg3fQ.GT4B2wuDJcVKn1LCb
6UW1by0pXxvW5vdQD2DzvCmRxODoy-gk9-XKaXZtvlRyFVAVG6BGDOZR0UIW--
Eduhps4zCr0PdQhy2cFnhsKu7tygdxSEEHpm_ESXi6t3EcIbxPb5y4xQflb1ny4TscB-OVN68pXT_
BwgfuszBcYON2ZzgEiRbcJmfwzAkmjThfHik59EZBOOC_lQv1pkQ3-
IQHJVSq0QhBIPx8seYhSIMtto9vJwIEciaZ8DPGbHtJ1WlOtm7SKbkoIDer4xSDVhKO03McJ0hlxFQpjc7AufONrN1jbgW
3ydJJ03u3tQHbWETufayz3-eN9GlufTpidUBgQ
ike V=root:0:FCT_EMS_TAG_ph2_0: EMS ZTNA token tags check success ZTNA_a
ike V=root:0:FCT_EMS_TAG_ph2_0:40481: send EMS_ZTNA_TAGS_CHECK_SUCCESS
```

This indicates the JWT security posture tags have been validated.

3. Verify the VPN is established:

```
# diagnose vpn ike gateway list name FCT_EMS_TAG_ph2_0
vd: root/0
name: FCT_EMS_TAG_ph2_0
version: 2
interface: vlan160 1123
addr: 10.160.1.1:4500 -> 10.160.1.197:55846
tun_id: 10.139.1.2/::10.0.0.183
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 39081s ago
eap-user: u1
2FA: no
peer-id: 10.160.1.197
peer-id-auth: no
FortiClient UID: F8A6AFF3531843B4972CBDBEEEB30E3C
assigned IPv4 address: 10.139.1.2/255.255.255.255
nat: me peer
pending-queue: 0
PPK: no
IKE SA: created 1/1  established 1/1  time 20/20/20 ms
IPsec SA: created 1/2  established 1/2  time 0/0/0 ms
  id/spi: 40481 db9b0112a09c6850/ac673ffc75f4cdca
  direction: responder
  status: established 39081-39081s ago = 20ms
...
```

```
# diagnose vpn tunnel list name FCT_EMS_TAG_ph2_0
list ipsec tunnel by names in vd 0
------------------------------------------------------
name=FCT_EMS_TAG_ph2_0 ver=2 serial=196 10.160.1.1:4500->10.160.1.197:55846 nexthop=0.0.0.0
tun_id=10.139.1.2 tun_id6=::10.0.0.183 status=up dst_mtu=1500 weight=1 country=ZZ
bound_if=1123 real_if=1123 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none options
[0x23a8]=npu rgwy-chg rport-chg frag-rfc  run_state=0 role=sync-primary accept_traffic=1
overlay_id=0
parent=FCT_EMS_TAG_ph2 index=0
proxyid_num=1 child_num=0 refcnt=7 ilast=15 olast=15 ad=/0
stat: rxp=11054 txp=11874 rxb=3338588 txb=4823896
dpd: mode=on-idle on=1 status=ok idle=60000ms retry=3 count=0 seqno=2
natt: mode=keepalive draft=0 interval=10 remote_port=55846
fec: egress=0 ingress=0
proxyid=FCT_EMS_TAG_ph2 proto=0 sa=1 ref=4 serial=1 add-route
```

Aggregate and redundant VPN

The following topics provide instructions on configuring aggregate and redundant VPNs:

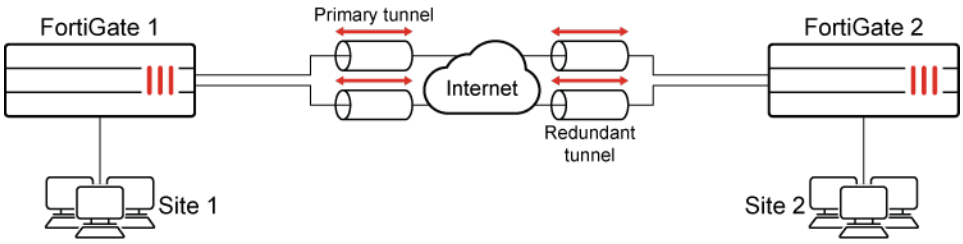
- Manual redundant VPN configuration on page 3187
- OSPF with IPsec VPN for network redundancy on page 3190

- IPsec VPN in an HA environment on page 3198
- Packet distribution and redundancy for aggregate IPsec tunnels on page 3205
- Packet distribution for aggregate dial-up IPsec tunnels using location ID on page 3217
- Packet distribution for aggregate static IPsec tunnels in SD-WAN on page 3222
- Packet distribution for aggregate IPsec tunnels using weighted round robin on page 3227
- Redundant hub and spoke VPN on page 3228

Manual redundant VPN configuration

A FortiGate with two interfaces connected to the internet can be configured to support redundant VPNs to the same remote peer. Four distinct paths are possible for VPN traffic from end to end. If the primary connection fails, the FortiGate can establish a VPN using the other connection.

Topology



The redundant configuration in this example uses route-based VPNs. The FortiGates must operate in NAT mode and use auto-keying.

This example assumes the redundant VPNs are essentially equal in cost and capability. When the original VPN returns to service, traffic continues to use the replacement VPN until the replacement VPN fails. If the redundant VPN uses more expensive facilities, only use it as a backup while the main VPN is down.

A redundant configuration for each VPN peer includes:

- One phase 1 configuration for each path between the two peers with dead peer detection enabled
- One phase 2 definition for each phase 1 configuration
- One static route for each IPsec interface with different distance values to prioritize the routes
- Two firewall policies per IPsec interface, one for each direction of traffic

To configure the phase 1 and phase 2 VPN settings:

1. Go to *VPN* > *IPsec Wizard* and select the *Custom* template.
2. Enter the tunnel name and click *Next*.
3. Enter the following phase 1 settings for path 1:

Remote Gateway	Static IP Address
IP Address	Enter the IP address of the primary interface of the remote peer.

Interface	Select the primary public interface of this peer.
Dead Peer Detection	On-Demand

4. Configure the remaining phase 1 and phase 2 settings as needed.
5. Click *OK*.
6. Repeat these steps for the remaining paths.
   a. Path 2:

Remote Gateway	Static IP Address
IP Address	Enter the IP address of the secondary interface of the remote peer.
Interface	Select the primary public interface of this peer.
Dead Peer Detection	On-Demand

   b. Path 3:

Remote Gateway	Static IP Address
IP Address	Enter the IP address of the primary interface of the remote peer.
Interface	Select the secondary public interface of this peer.
Dead Peer Detection	On-Demand

   c. Path 4:

Remote Gateway	Static IP Address
IP Address	Enter the IP address of the secondary interface of the remote peer.
Interface	Select the secondary public interface of this peer.
Dead Peer Detection	On-Demand

To configure the static routes:

1. Go to *Network* > *Static Routes* and click *Create New*.
2. In the *Destination* field, enter the subnet of the private network.
3. For *Interface*, select one of the IPsec interfaces on the local peer.
4. Enter a value for *Administrative Distance*.
5. Click *OK*.
6. Repeat these steps for the three remaining paths, and enter different values for *Administrative Distance* to prioritize the paths.

To configure the firewall policies:

1. Create the policies for the local primary interface:
   a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
   b. Enter the following:

Name	Enter a name for the policy.
Incoming Interface	Select the local interface to the internal (private) network.
Outgoing Interface	Select one of the virtual IPsec interfaces.
Source	All
Destination	All
Schedule	Always
Service	All
Action	ACCEPT

   c. Click *OK*.
   d. Click *Create New* and configure the policy for the other direction of traffic:

Name	Enter a name for the policy.
Incoming Interface	Select one of the virtual IPsec interfaces.
Outgoing Interface	Select the local interface to the internal (private) network.
Source	All
Destination	All
Schedule	Always
Service	All
Action	ACCEPT

   e. In the policy list, drag the VPN policies above any other policies with similar source and destination addresses.
2. Repeat these steps to create the policies for the three remaining paths.

Creating a backup IPsec interface

A route-based VPN can be configured to act as a backup IPsec interface when the main VPN is out of service. This can only be configured in the CLI.

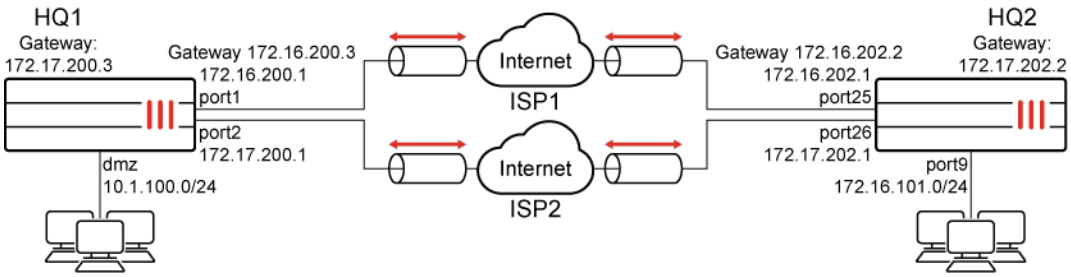
The backup feature works on interfaces with static addresses that have dead peer detection enabled. The `monitor` option creates a backup VPN for the specified phase 1 configuration.

To create a backup IPsec interface:

```
config vpn ipsec phase1-interface
    edit main_vpn
        set dpd on-demand
        set interface port1
        set nattraversal enable
        set psksecret ********
        set remote-gw 192.168.10.8
        set type static
    next
    edit backup_vpn
        set dpd on-demand
        set interface port2
        set monitor main_vpn
        set nattraversal enable
        set psksecret ********
        set remote-gw 192.168.10.8
        set type static
    next
end
```

OSPF with IPsec VPN for network redundancy

This is a sample configuration of using OSPF with IPsec VPN to set up network redundancy. Route selection is based on OSPF cost calculation. You can configure ECMP or primary/secondary routes by adjusting OSPF path cost.



Because the GUI can only complete part of the configuration, we recommend using the CLI.

To configure OSPF with IPsec VPN to achieve network redundancy using the CLI:

1. Configure the WAN interface and static route.

   Each FortiGate has two WAN interfaces connected to different ISPs. The ISP1 link is for the primary FortiGate and the IPS2 link is for the secondary FortiGate.

   a. Configure HQ1:

   ```
   config system interface
       edit "port1"
           set alias to_ISP1
   ```

```
        set ip 172.16.200.1 255.255.255.0
    next
    edit "port2"
        set alias to_ISP2
        set ip 172.17.200.1 255.255.255.0
    next
end
config router static
    edit 1
        set gateway 172.16.200.3
        set device "port1"
    next
    edit 2
        set gateway 172.17.200.3
        set device "port2"
        set priority 100
    next
end
```

b. Configure HQ2:

```
config system interface
    edit "port25"
        set alias to_ISP1
        set ip 172.16.202.1 255.255.255.0
    next
    edit "port26"
        set alias to_ISP2
        set ip 172.17.202.1 255.255.255.0
    next
end
config router static
    edit 1
        set gateway 172.16.202.2
        set device "port25"
    next
    edit 2
        set gateway 172.17.202.2
        set device "port26"
        set priority 100
    next
end
```

2. Configure the internal (protected subnet) interface:

a. Configure HQ1:

```
config system interface
    edit "dmz"
        set ip 10.1.100.1 255.255.255.0
    next
end
```

b. Configure HQ2:

```
config system interface
    edit "port9"
        set ip 172.16.101.1 255.255.255.0
    next
end
```

3. Configure IPsec phase1-interface and phase-2 interface. On each FortiGate, configure two IPsec tunnels: a primary and a secondary:

a. Configure HQ1:

```
config vpn ipsec phase1-interface
    edit "pri_HQ2"
        set interface "port1"
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set remote-gw 172.16.202.1
        set psksecret sample1
    next
    edit "sec_HQ2"
        set interface "port2"
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set remote-gw 172.17.202.1
        set psksecret sample2
    next
end
config vpn ipsec phase2-interface
    edit "pri_HQ2"
        set phase1name "pri_HQ2"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set auto-negotiate enable
    next
    edit "sec_HQ2"
        set phase1name "sec_HQ2"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set auto-negotiate enable
    next
end
```

b. Configure HQ2:

```
config vpn ipsec phase1-interface
    edit "pri_HQ1"
        set interface "port25"
        set peertype any
        set net-device enable
```

```
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set remote-gw 172.16.200.1
        set psksecret sample1
    next
    edit "sec_HQ1"
        set interface "port26"
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set remote-gw 172.17.200.1
        set psksecret sample2
    next
end
config vpn ipsec phase2-interface
    edit "pri_HQ1"
        set phase1name "pri_HQ1"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set auto-negotiate enable
    next
    edit "sec_HQ1"
        set phase1name "sec_HQ1"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set auto-negotiate enable
    next
end
```

4. Configure an inbound and outbound firewall policy for each IPsec tunnel:
   a. Configure HQ1:

```
config firewall policy
    edit 1
        set name "pri_inbound"
        set srcintf "pri_HQ2"
        set dstintf "dmz"
        set srcaddr "172.16.101.0"
        set dstaddr "10.1.100.0"
        set action accept
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set name "pri_outbound"
        set srcintf "dmz"
        set dstintf "pri_HQ2"
        set srcaddr "10.1.100.0"
        set dstaddr "172.16.101.0"
        set action accept
        set schedule "always"
        set service "ALL"
    next
```

```
    edit 3
        set name "sec_inbound"
        set srcintf "sec_HQ2"
        set dstintf "dmz"
        set srcaddr "172.16.101.0"
        set dstaddr "10.1.100.0"
        set action accept
        set schedule "always"
        set service "ALL"
    next
    edit 4
        set name "sec_outbound"
        set srcintf "dmz"
        set dstintf "sec_HQ2"
        set srcaddr "10.1.100.0"
        set dstaddr "172.16.101.0"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end
```

b. Configure HQ2:

```
config firewall policy
    edit 1
        set name "pri_inbound"
        set srcintf "pri_HQ1"
        set dstintf "port9"
        set srcaddr "10.1.100.0"
        set dstaddr "172.16.101.0"
        set action accept
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set name "pri_outbound"
        set srcintf "port9"
        set dstintf "pri_HQ1"
        set srcaddr "10.1.100.0"
        set dstaddr "172.16.101.0"
        set action accept
        set schedule "always"
        set service "ALL"
    next
    edit 3
        set name "sec_inbound"
        set srcintf "sec_HQ1"
        set dstintf "port9"
        set srcaddr "10.1.100.0"
        set dstaddr "172.16.101.0"
        set action accept
```

```
        set schedule "always"
        set service "ALL"
    next
    edit 4
        set name "sec_outbound"
        set srcintf "port9"
        set dstintf "sec_HQ1"
        set srcaddr "172.16.101.0"
        set dstaddr "10.1.100.0"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end
```

5. Assign an IP address to the IPsec tunnel interface:
   a. Configure HQ1:

```
config system interface
    edit "pri_HQ2"
        set ip 10.10.10.1 255.255.255.255
        set remote-ip 10.10.10.2 255.255.255.255
    next
    edit "sec_HQ2"
        set ip 10.10.11.1 255.255.255.255
        set remote-ip 10.10.11.2 255.255.255.255
    next
end
```

   b. Configure HQ2:

```
config system interface
    edit "pri_HQ1"
        set ip 10.10.10.2 255.255.255.255
        set remote-ip 10.10.10.1 255.255.255.255
    next
    edit "sec_HQ1"
        set ip 10.10.11.2 255.255.255.255
        set remote-ip 10.10.11.1 255.255.255.255
    next
end
```

6. Configure OSPF:
   a. Configure HQ1:

```
config router ospf
    set router-id 1.1.1.1
    config area
        edit 0.0.0.0
        next
    end
    config ospf-interface
```

```
        edit "pri_HQ2"
            set interface "pri_HQ2"
            set cost 10
            set network-type point-to-point
        next
        edit "sec_HQ2"
            set interface "sec_HQ2"
            set cost 20
            set network-type point-to-point
        next
    end
    config network
        edit 1
            set prefix 10.10.10.0 255.255.255.0
        next
        edit 2
            set prefix 10.10.11.0 255.255.255.0
        next
        edit 3
            set prefix 10.1.100.0 255.255.255.0
        next
    end
end
```

b. Configure HQ2:

```
config router ospf
    set router-id 2.2.2.2
    config area
        edit 0.0.0.0
        next
    end
    config ospf-interface
        edit "pri_HQ1"
            set interface "pri_HQ1"
            set cost 10
            set network-type point-to-point
        next
        edit "sec_HQ1"
            set interface "sec_HQ1"
            set cost 20
            set network-type point-to-point
        next
    end
    config network
        edit 1
            set prefix 10.10.10.0 255.255.255.0
        next
        edit 2
            set prefix 10.10.11.0 255.255.255.0
        next
        edit 3
```

```
            set prefix 172.16.101.0 255.255.255.0
        next
    end
end
```

To check VPN and OSPF states using diagnose and get commands:

1. Run the `HQ1 # diagnose vpn ike gateway list` command. The system should return the following:

```
vd: root/0
name: pri_HQ2
version: 1
interface: port1 11
addr: 172.16.200.1:500 -&gt; 172.16.202.1:500
virtual-interface-addr: 10.10.10.1 -&gt; 10.10.10.2
created: 1024s ago
IKE SA: created 1/1 established 1/1 time 0/0/0 ms
IPsec SA: created 1/3 established 1/2 time 0/5/10 ms
    id/spi: 45 d184777257b4e692/e2432f834aaf5658 direction: responder status: established
1024-1024s ago = 0ms proposal: aes128-sha256 key: 9ed41fb06c983344-189538046f5ad204
lifetime/rekey: 86400/85105 DPD sent/recv: 00000003/00000000
vd: root/0
name: sec_HQ2
version: 1
interface: port2 12
addr: 172.17.200.1:500 -&gt; 172.17.202.1:500
virtual-interface-addr: 10.10.11.1 -&gt; 10.10.11.2
created: 346s ago
IKE SA: created 1/1 established 1/1 time 0/0/0 ms
IPsec SA: created 1/1 established 1/1 time 0/10/15 ms
    id/spi: 48 d909ed68636b1ea5/163015e73ea050b8 direction: initiator status: established 0-0s
ago = 0ms proposal: aes128-sha256 key: b9e93c156bdf4562-29db9fbafa256152 lifetime/rekey:
86400/86099 DPD sent/recv: 00000000/00000000
```

2. Run the `HQ1 # diagnose vpn tunnel list` command. The system should return the following:

```
list all ipsec tunnel in vd 0
name=pri_HQ2 ver=1 serial=1 172.16.200.1:0-&gt;172.16.202.1:0 tun_id=172.16.202.1
bound_if=11 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/528 options[0210]=create_dev frag-
rfc accept_traffic=1
proxyid_num=1 child_num=0 refcnt=14 ilast=2 olast=2 ad=/0
stat: rxp=102 txp=105 rxb=14064 txb=7816
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=3
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=pri_HQ2 proto=0 sa=1 ref=2 serial=1 auto-negotiate
    src: 0:0.0.0.0/0.0.0.0:0 dst: 0:0.0.0.0/0.0.0.0:0 SA: ref=3 options=18227 type=00 soft=0
mtu=1438 expire=42254/0B replaywin=2048
        seqno=6a esn=0 replaywin_lastseq=00000067 itn=0
    life: type=01 bytes=0/0 timeout=42932/43200 dec: spi=1071b4ee esp=aes key=16
032036b24a4ec88da63896b86f3a01db
        ah=sha1 key=20 3962933e24c8da21c65c13bc2c6345d643199cdf
    enc: spi=ec89b7e3 esp=aes key=16 92b1d85ef91faf695fca05843dd91626
```

```
        ah=sha1 key=20 2de99d1376506313d9f32df6873902cf6c08e454
    dec:pkts/bytes=102/7164, enc:pkts/bytes=105/14936
name=sec_HQ2 ver=1 serial=2 172.17.200.1:0-&gt;172.17.202.1:0 tun_id=172.17.202.1
bound_if=12 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/528 options[0210]=create_dev frag-
rfc accept_traffic=1
proxyid_num=1 child_num=0 refcnt=14 ilast=3 olast=0 ad=/0
stat: rxp=110 txp=114 rxb=15152 txb=8428
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=3
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=sec_HQ2 proto=0 sa=1 ref=2 serial=1 auto-negotiate
    src: 0:0.0.0.0/0.0.0.0:0 dst: 0:0.0.0.0/0.0.0.0:0 SA: ref=3 options=18227 type=00 soft=0
mtu=1438 expire=42927/0B replaywin=2048
        seqno=2 esn=0 replaywin_lastseq=00000002 itn=0
    life: type=01 bytes=0/0 timeout=42931/43200 dec: spi=1071b4ef esp=aes key=16
bcdcabdb7d1c7c695d1f2e0f5441700a
        ah=sha1 key=20 e7a0034589f82eb1af41efd59d0b2565fef8d5da
    enc: spi=ec89b7e4 esp=aes key=16 234240b69e61f6bdee2b4cdec0f33bea
        ah=sha1 key=20 f9d4744a84d91e5ce05f5984737c2a691a3627e8
    dec:pkts/bytes=1/68, enc:pkts/bytes=1/136
```

3. Run the `HQ1 # get router info ospf neighbor` command. The system should return the following:

```
OSPF process 0, VRF 0:
Neighbor ID Pri State Dead Time Address Interface
2.2.2.2 1. Full/ - 00:00:37 10.10.10.2 pri_HQ2
2.2.2.2 1. Full/ - 00:00:32 10.10.11.2 sec_HQ2
```

4. Run the `HQ1 # get router info routing-table ospf` command. The system should return the following:

```
Routing table for VRF=0
O 172.16.101.0/24 [110/20] via 10.10.10.2, pri_HQ2 , 00:03:21
```

In case the primary tunnel is down after route convergence.

5. Run the `HQ1 # get router info routing-table ospf` command. The system should return the following:

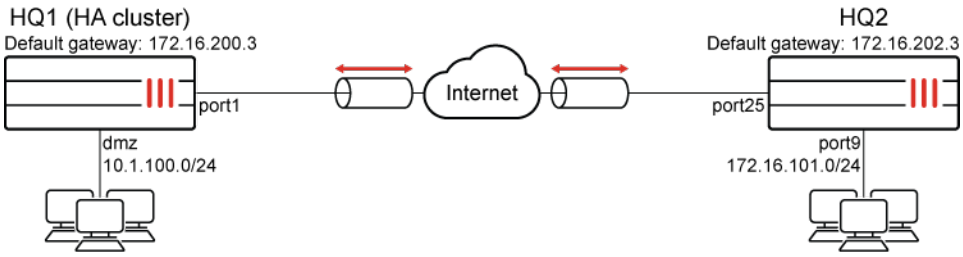
```
Routing table for VRF=0
O 172.16.101.0/24 [110/110] via 10.10.11.2, sec_HQ2 , 00:00:01
```

IPsec VPN in an HA environment

This is a sample configuration of site-to-site IPsec VPN in an HA environment.

For this example, set up HA as described in the HA topics. When setting up HA, enable the following options to ensure IPsec VPN traffic is not interrupted during an HA failover:

- `session-pickup` under HA settings.
- `ha-sync-esp-seqno` under IPsec phase1-interface settings.



You can configure IPsec VPN in an HA environment using the GUI or CLI.

In this example, the VPN name for HQ1 is "to_HQ2", and the VPN name for HQ2 is "to_HQ1".

To configure IPsec VPN in an HA environment in the GUI:

1. Set up IPsec VPN on HQ1 (the HA cluster):
   a. Go to *VPN > IPsec Wizard* and configure the following settings for *VPN Setup*:
      i. Enter a VPN name.
      ii. For *Template Type*, select *Site to Site*.
      iii. For *Remote Device Type*, select *FortiGate*.
      iv. For *NAT Configuration*, set *No NAT between sites*.
      v. Click *Next*.
   b. Configure the following settings for *Authentication*:
      i. For *Remote Device*, select *IP Address*.
      ii. In the *IP address* field, enter *172.16.202.1*.
      iii. For *Outgoing Interface*, select port1.
      iv. For *Authentication Method*, select *Pre-shared Key*.
      v. In the *Pre-shared Key* field, enter an example key.
      vi. Click *Next*.
   c. Configure the following settings for *Policy & Routing*:
      i. From the *Local Interface* dropdown menu, select the local interface.
      ii. Configure the *Local Subnets* as *10.1.100.0/24*.
      iii. Configure the *Remote Subnets* as *172.16.101.0/24*.
      iv. Click *Create*.
2. Set up IPsec VPN on HQ2:
   a. Go to *VPN > IPsec Wizard* and configure the following settings for *VPN Setup*:
      i. Enter a VPN name.
      ii. For *Template Type*, select *Site to Site*.
      iii. For *Remote Device Type*, select *FortiGate*.
      iv. For *NAT Configuration*, set *No NAT between sites*.
      v. Click *Next*.
   b. Configure the following settings for *Authentication*:
      i. For *Remote Device*, select *IP Address*.
      ii. In the *IP address* field, enter 172.16.200.1.
      iii. For *Outgoing Interface*, select port13.
      iv. For *Authentication Method*, select *Pre-shared Key*.

v. In the *Pre-shared Key* field, enter an example key.
vi. Click *Next*.

c. Configure the following settings for *Policy & Routing*:
i. From the *Local Interface* dropdown menu, select the desired local interface. In this example, it is port9.
ii. Configure the *Local Subnets* as 172.16.101.0.
iii. Configure the *Remote Subnets* as 10.1.100.0
iv. Click *Create*.

To configure IPsec VPN in an HA environment using the CLI:

1. Configure HA. In this example, two FortiGates work in active-passive mode. The HA heartbeat interfaces are WAN1 and WAN2:

```
config system ha
    set group-name "FGT-HA"
    set mode a-p
    set password sample
    set hbdev "wan1" 50 "wan2" 50
    set session-pickup enable
    set priority 200
    set override-wait-time 10
end
```

2. Configure the WAN interface and default route. The WAN interface is the interface connected to the ISP. It can work in static mode (as shown in this example), DHCP, or PPPoE mode. The IPsec tunnel is established over the WAN interface.
   a. Configure HQ1:

```
config system interface
    edit "port1"
        set vdom "root"
        set ip 172.16.200.1 255.255.255.0
    next
end
config router static
    edit 1
        set gateway 172.16.200.3
        set device "port1"
    next
end
```

   b. Configure HQ2:

```
config system interface
    edit "port25"
        set vdom "root"
        set ip 172.16.202.1 255.255.255.0
    next
end
```

```
config router static
    edit 1
        set gateway 172.16.202.2
        set device "port25"
    next
end
```

3. Configure the internal (protected subnet) interface. The internal interface connects to the corporate internal network. Traffic from this interface routes out the IPsec VPN tunnel.
   a. Configure HQ1:

```
config system interface
    edit "dmz"
        set vdom "root"
        set ip 10.1.100.1 255.255.255.0
    next
end
```

   b. Configure HQ2:

```
config system interface
    edit "port9"
        set vdom "root"
        set ip 172.16.101.1 255.255.255.0
    next
end
```

4. Configure the IPsec phase1-interface. This example uses PSK as the authentication method. You can also use signature authentication.
   a. Configure HQ1:

```
config vpn ipsec phase1-interface
    edit "to_HQ2"
        set interface "port1"
        set peertype any
        set net-device enable
        set ha-sync-esp-seqno  enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set remote-gw 172.16.202.1
        set psksecret sample
    next
end
```

   b. Configure HQ2:

```
config vpn ipsec phase1-interface
    edit "to_HQ1"
        set interface "port25"
        set peertype any
        set net-device enable
        set ha-sync-esp-seqno  enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
```

```
        set remote-gw 172.16.200.1
        set psksecret sample
    next
end
```

5. Configure the IPsec phase2-interface:
   a. Configure HQ1:

```
config vpn ipsec phase2-interface
    edit "to_HQ2"
        set phase1name "to_HQ2"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set auto-negotiate enable
    next
end
```

   b. Configure HQ2:

```
config vpn ipsec phase2-interface
    edit "to_HQ1"
        set phase1name "to_HQ1"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set auto-negotiate enable
    next
end
```

6. Configure static routes. Two static routes are added to reach the remote protected subnet. The blackhole route is important to ensure IPsec traffic does not match the default route when the IPsec tunnel is down.
   a. Configure HQ1:

```
config router static
    edit 2
        set dst 172.16.101.0 255.255.255.0
        set device "to_HQ2"
    next
    edit 3
        set dst 172.16.101.0 255.255.255.0
        set blackhole enable
        set distance 254
    next
end
```

   b. Configure HQ2:

```
config router static
    edit 2
        set dst 10.1.100.0 255.255.255.0
        set device "to_HQ1"
    next
    edit 3
```

```
        set dst 10.1.100.0 255.255.255.0
        set blackhole enable
        set distance 254
    next
end
```

7. Configure two firewall policies to allow bi-directional IPsec traffic flow over the IPsec tunnel:
   a. Configure HQ1:

```
config firewall policy
    edit 1
        set name "inbound"
        set srcintf "to_HQ2"
        set dstintf "dmz"
        set srcaddr "172.16.101.0"
        set dstaddr "10.1.100.0"
        set action accept
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set name "outbound"
        set srcintf "dmz"
        set dstintf "to_HQ2"
        set srcaddr "10.1.100.0"
        set dstaddr "172.16.101.0"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end
```

   b. Configure HQ2:

```
config firewall policy
    edit 1
        set name "inbound"
        set srcintf "to_HQ1"
        set dstintf "port9"
        set srcaddr "10.1.100.0"
        set dstaddr "172.16.101.0"
        set action accept
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set name "outbound"
        set srcintf "port9"
        set dstintf "to_HQ1"
        set srcaddr "172.16.101.0"
        set dstaddr "10.1.100.0"
        set action accept
```

```
        set schedule "always"
        set service "ALL"
    next
end
```

8. Use the following `diagnose` commands to check IPsec phase1/phase2 interface status including the sequence number on the secondary FortiGate. The `diagnose debug application ike -1` command is the key to troubleshoot why the IPsec tunnel failed to establish.
   a. Run the `HQ1 # diagnose vpn ike gateway list` command. The system should return the following:

```
vd: root/0
name: to_HQ2
version: 1
interface: port1 11
addr: 172.16.200.1:500 -> 172.16.202.1:500
created: 5s ago
IKE SA: created 1/1 established 1/1 time 0/0/0 ms
IPsec SA: created 2/2 established 2/2 time 0/0/0 ms
    id/spi: 12 6e8d0532e7fe8d84/3694ac323138a024 direction: responder status: established
5-5s ago = 0ms proposal: aes128-sha256 key: b3efb46d0d385aff-7bb9ee241362ee8d
lifetime/rekey: 86400/86124 DPD sent/recv: 00000000/00000000
```

   b. Run the `HQ1 # diagnose vpn tunnel list` command. The system should return the following:

```
list all ipsec tunnel in vd 0

name=to_HQ2 ver=1 serial=1 172.16.200.1:0->172.16.202.1:0 tun_id=172.16.202.1
bound_if=11 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/528 options[0210]=create_dev
frag-rfc accept_traffic=1
proxyid_num=1 child_num=0 refcnt=11 ilast=7 olast=87 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=to_HQ2 proto=0 sa=1 ref=2 serial=1 auto-negotiate
    src: 0:0.0.0.0/0.0.0.0:0 dst: 0:0.0.0.0/0.0.0.0:0 SA: ref=3 options=18227 type=00
soft=0 mtu=1438 expire=42927/0B replaywin=2048
        seqno=1 esn=0 replaywin_lastseq=00000000 itn=0
    life: type=01 bytes=0/0 timeout=42930/43200 dec: spi=ef9ca700 esp=aes key=16
a2c6584bf654d4f956497b3436f1cfc7
        ah=sha1 key=20 82c5e734bce81e6f18418328e2a11aeb7baa021b
    enc: spi=791e898e esp=aes key=16 0dbb4588ba2665c6962491e85a4a8d5a
        ah=sha1 key=20 2054b318d2568a8b12119120f20ecac97ab730b3
    dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
```

   ESP seqno synced to primary FortiGate every five minutes, and big gap between primary and secondary to ensure that no packet is dropped after HA failover caused by tcp-replay. Check ESP sequence number synced on secondary FortiGate.

   c. Run the `HQ1 # execute ha manage 0 admin` command.
   d. Run the `HQ1-Sec # diagnose vpn tunnel list` command. The system should return the following:

```
list all ipsec tunnel in vd 0

name=to_HQ2 ver=1 serial=1 172.16.200.1:0->172.16.202.1:0 tun_id=172.16.202.1
bound_if=11 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/528 options[0210]=create_dev
frag-rfc accept_traffic=1
proxyid_num=1 child_num=0 refcnt=11 ilast=13 olast=274 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=to_HQ2 proto=0 sa=1 ref=2 serial=1 auto-negotiate
    src: 0:0.0.0.0/0.0.0.0:0 dst: 0:0.0.0.0/0.0.0.0:0 SA: ref=3 options=27 type=00 soft=0
mtu=1280 expire=42740/0B replaywin=2048
        seqno=47868c01 esn=0 replaywin_lastseq=00000000 itn=0
    life: type=01 bytes=0/0 timeout=42930/43200 dec: spi=ef9ca700 esp=aes key=16
a2c6584bf654d4f956497b3436f1cfc7
        ah=sha1 key=20 82c5e734bce81e6f18418328e2a11aeb7baa021b
    enc: spi=791e898e esp=aes key=16 0dbb4588ba2665c6962491e85a4a8d5a
        ah=sha1 key=20 2054b318d2568a8b12119120f20ecac97ab730b3
    dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
```

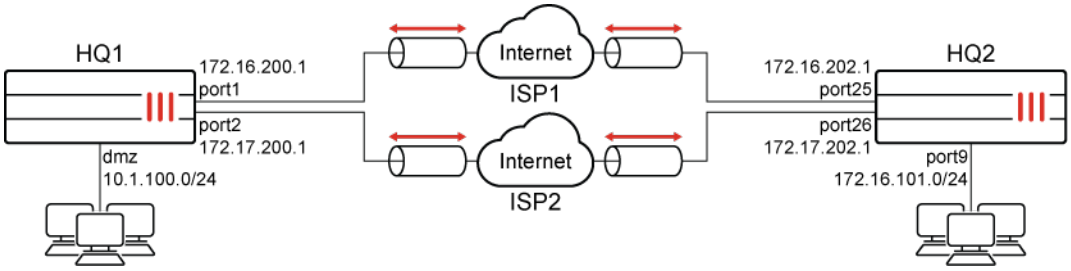
Packet distribution and redundancy for aggregate IPsec tunnels

This is a sample configuration of a multiple site-to-site IPsec VPN that uses an IPsec aggregate interface to set up redundancy and traffic load-balancing. The VPN tunnel interfaces must have `net-device` disabled in order to be members of the IPsec aggregate.

Each FortiGate has two WAN interfaces connected to different ISPs. OSPF runs over the IPsec aggregate in this configuration.

The supported load balancing algorithms are: L3, L4, round-robin (default), weighted round-robin, and redundant. The first four options allow traffic to be load-balanced, while the last option (redundant) uses the first tunnel that is up for all traffic.

Dynamic routing can run on the aggregate interface, and it can be a member interface in SD-WAN (not shown in this configuration).



Configuring the HQ1 FortiGate in the GUI

There are five steps to configure the FortiGate:

1. Create the IPsec tunnels.
2. Create the IPsec aggregate.
3. Configure the firewall policies.
4. Configure the aggregate VPN interface IPs.
5. Configure OSPF.

To create the IPsec tunnels:

1. Go to *VPN* > *IPsec Wizard* and select the *Custom* template.
2. For *Name*, enter pri_HQ2 and click *Next*.
3. Enter the following:

Phase 1	
IP Address	172.16.202.1
Interface	port1
Device creation	Disabled
Aggregate member	Enabled
Authentication Method	Pre-shared Key
Pre-shared Key	Enter the secure key
IKE Mode	Aggressive
Peer Options Accept Types	Any peer ID
Phase 2	
Auto-negotiate	Enable

4. Configure the other settings as needed.
5. Click *OK*.
6. Create another tunnel named sec_HQ2 with the following settings:

Phase 1	
IP Address	172.17.202.1
Interface	port2
Device creation	Disabled
Aggregate member	Enabled
Authentication Method	Pre-shared Key
Pre-shared Key	Enter the secure key
IKE Mode	Aggressive
Peer Options Accept Types	Any peer ID
Phase 2	
Auto-negotiate	Enable

To create the IPsec aggregate:

1. Go to *VPN > IPsec Tunnels* and click *Create New > IPsec Aggregate*.
2. For *Name*, enter agg_HQ2.
3. Select a load balancing algorithm.
4. From the *Tunnel* dropdown, select the tunnels that you created previously (*pri_HQ2* and *sec_HQ2*). If required, enter weights for each tunnel.
5. Click *OK*.

To configure the firewall policies:

1. Go to *Policy & Objects > Firewall Policy*.
2. Create an inbound traffic policy with the following settings:

Name	inbound
Incoming Interface	agg_HQ2
Outgoing Interface	dmz
Source	172.16.101.0
Destination	10.1.100.0
Schedule	always
Action	ACCEPT
Service	ALL

3. Click *OK*.
4. Create an outbound traffic policy with the following settings:

Name	outbound
Incoming Interface	dmz
Outgoing Interface	agg_HQ2
Source	10.1.100.0
Destination	172.16.101.0
Schedule	always
Action	ACCEPT
Service	ALL

To configure the aggregate VPN interface IPs:

1. Go to *Network > Interfaces* and edit agg_HQ2.
2. For *IP*, enter 10.10.10.1.
3. For *Remote IP/Netmask*, enter 10.10.10.2 255.255.255.255.
4. Click *OK*.

To configure OSPF:

1. Go to *Network* > *OSPF*.
2. For *Router ID*, enter 1.1.1.1.
3. In the *Areas* table, click *Create New*.
   a. For *Area ID*, enter 0.0.0.0.
   b. Click *OK*.
4. In the *Networks* table, click *Create New*.
   a. Set the *Area* to *0.0.0.0*.
   b. For *IP/Netmask*, enter 10.1.100.0/24.
   c. Click *OK*.
   d. Click *Create New*.
   e. For *IP/Netmask*, enter 10.10.10.0/24.
   f. Click *OK*.
5. Click *Apply*.

Configuring the HQ2 FortiGate in the GUI

There are five steps to configure the FortiGate:

1. Create the IPsec tunnels.
2. Create the IPsec aggregate.
3. Configure the firewall policies.
4. Configure the aggregate VPN interface IPs.
5. Configure OSPF.

To create the IPsec tunnels:

1. Go to *VPN* > *IPsec Wizard* and select the *Custom* template.
2. For *Name*, enter pri_HQ1 and click *Next*.
3. Enter the following:

Phase 1	
IP Address	172.16.200.1
Interface	port25
Device creation	Disabled
Aggregate member	Enabled
Authentication Method	Pre-shared Key
Pre-shared Key	Enter the secure key
IKE Mode	Aggressive
Peer Options Accept Types	Any peer ID

Phase 2	
Auto-negotiate	Enable

4. Configure the other settings as needed.
5. Click *OK*.
6. Create another tunnel named sec_HQ1 with the following settings:

Phase 1	
IP Address	172.17.200.1
Interface	port26
Device creation	Disabled
Aggregate member	Enabled
Authentication Method	Pre-shared Key
Pre-shared Key	Enter the secure key
IKE Mode	Aggressive
Peer Options Accept Types	Any peer ID
Phase 2	
Auto-negotiate	Enable

To create the IPsec aggregate:

1. Go to *VPN > IPsec Tunnels* and click *Create New > IPsec Aggregate*.
2. For *Name*, enter agg_HQ1.
3. Select a load balancing algorithm.
4. From the *Tunnel* dropdown, select the tunnels that you created previously (*pri_HQ1* and *sec_HQ1*). If required, enter weights for each tunnel.
5. Click *OK*.

To configure the firewall policies:

1. Go to *Policy & Objects > Firewall Policy*.
2. Create an inbound traffic policy with the following settings:

Name	inbound
Incoming Interface	agg_HQ1
Outgoing Interface	port9
Source	10.1.100.0
Destination	172.16.101.0
Schedule	always

Action	ACCEPT
Service	ALL

3. Click *OK*.
4. Create an outbound traffic policy with the following settings:

Name	outbound
Incoming Interface	port9
Outgoing Interface	agg_HQ1
Source	172.16.101.0
Destination	10.1.100.0
Schedule	always
Action	ACCEPT
Service	ALL

To configure the aggregate VPN interface IPs:

1. Go to *Network > Interfaces* and edit agg_HQ1.
2. For *IP*, enter 10.10.10.2.
3. For *Remote IP/Netmask*, enter 10.10.10.1 255.255.255.255.
4. Click *OK*.

To configure OSPF:

1. Go to *Network > OSPF*.
2. For *Router ID*, enter 2.2.2.2.
3. In the *Areas* table, click *Create New*.
   a. For *Area ID*, enter 0.0.0.0.
   b. Click *OK*.
4. In the *Networks* table, click *Create New*.
   a. Set the *Area* to *0.0.0.0*.
   b. For *IP/Netmask*, enter 172.16.101.0/24.
   c. Click *OK*.
   d. Click *Create New*.
   e. For *IP/Netmask*, enter 10.10.10.0/24.
   f. Click *OK*.
5. Click *Apply*.

Monitoring the traffic in the GUI

To monitor the traffic:

1. Go to *Dashboard* > *Network* , hover over the *IPsec* widget, then click *Expand to Full Screen*.
2. Expand the aggregate tunnel in the table to view statistics for each aggregate member.

Configuring the HQ1 FortiGate in the CLI

There are six steps to configure the FortiGate:

1. Configure the interfaces.
2. Configure two IPsec phase 1 and phase 2 interfaces.
3. Configure the IPsec aggregate.
4. Configure the firewall policies.
5. Configure the aggregate VPN interface IPs.
6. Configure OSPF.

To configure the interfaces:

1. Configure port1, port2, and dmz as shown in the topology diagram.

To configure two IPsec phase 1 and phase 2 interfaces:

```
config vpn ipsec phase1-interface
    edit "pri_HQ2"
        set interface "port1"
        set peertype any
        set net-device disable
        set aggregate-member enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set remote-gw 172.16.202.1
        set psksecret sample1
    next
    edit "sec_HQ2"
        set interface "port2"
        set peertype any
        set net-device disable
        set aggregate-member enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set remote-gw 172.17.202.1
        set psksecret sample2
    next
end
config vpn ipsec phase2-interface
    edit "pri_HQ2"
        set phase1name "pri_HQ2"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
```

```
        set auto-negotiate enable
    next
    edit "sec_HQ2"
        set phase1name "sec_HQ2"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
        set auto-negotiate enable
    next
end
```

To configure the IPsec aggregate:

```
config system ipsec-aggregate
    edit "agg_HQ2"
        set member "pri_HQ2" "sec_HQ2"
    next
end
```

To configure the firewall policies:

```
config firewall policy
    edit 1
        set name "inbound"
        set srcintf "agg_HQ2"
        set dstintf "dmz"
        set srcaddr "172.16.101.0"
        set dstaddr "10.1.100.0"
        set action accept
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set name "outbound"
        set srcintf "dmz"
        set dstintf "agg_HQ2"
        set srcaddr "10.1.100.0"
        set dstaddr "172.16.101.0"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end
```

To configure the aggregate VPN interface IPs:

```
config system interface
    edit "agg_HQ2"
        set ip 10.10.10.1 255.255.255.255
        set remote-ip 10.10.10.2 255.255.255.255
```

```
    next
end
```

To configure OSPF:

```
config router ospf
    set router-id 1.1.1.1
    config area
        edit 0.0.0.0
        next
    end
    config network
        edit 1
            set prefix 10.1.100.0 255.255.255.0
        next
        edit 2
            set prefix 10.10.10.0 255.255.255.0
        next
    end
end
```

Configuring the HQ2 FortiGate in the CLI

There are six steps to configure the FortiGate:

1. Configure the interfaces.
2. Configure two IPsec phase 1 and phase 2 interfaces.
3. Configure the IPsec aggregate.
4. Configure the firewall policies.
5. Configure the aggregate VPN interface IPs.
6. Configure OSPF.

To configure the interfaces:

1. Configure port25, port26, and port9 as shown in the topology diagram.

To configure two IPsec phase 1 and phase 2 interfaces:

```
config vpn ipsec phase1-interface
    edit "pri_HQ1"
        set interface "port25"
        set peertype any
        set net-device disable
        set aggregate-member enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set remote-gw 172.16.200.1
        set psksecret sample1
    next
```

```
    edit "sec_HQ1"
        set interface "port26"
        set peertype any
        set net-device disable
        set aggregate-member enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set remote-gw 172.17.200.1
        set psksecret sample2
    next
end
config vpn ipsec phase2-interface
    edit "pri_HQ1"
        set phase1name "pri_HQ1"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
        set auto-negotiate enable
    next
    edit "sec_HQ1"
        set phase1name "sec_HQ1"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
        set auto-negotiate enable
    next
end
```

To configure the IPsec aggregate:

```
config system ipsec-aggregate
    edit "agg_HQ1"
        set member "pri_HQ1" "sec_HQ1"
    next
end
```

To configure the firewall policies:

```
config firewall policy
    edit 1
        set name "inbound"
        set srcintf "agg_HQ1"
        set dstintf "port9"
        set srcaddr "10.1.100.0"
        set dstaddr "172.16.101.0"
        set action accept
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set name "outbound"
        set srcintf "port9"
        set dstintf "agg_HQ1"
        set srcaddr "172.16.101.0"
```

```
        set dstaddr "10.1.100.0"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end
```

To configure the aggregate VPN interface IPs:

```
config system interface
    edit "agg_HQ1"
        set ip 10.10.10.2 255.255.255.255
        set remote-ip 10.10.10.1 255.255.255.255
    next
end
```

To configure OSPF:

```
config router ospf
    set router-id 2.2.2.2
    config area
        edit 0.0.0.0
        next
    end
    config network
        edit 1
            set prefix 172.16.101.0 255.255.255.0
        next
        edit 2
            set prefix 10.10.10.0 255.255.255.0
        next
    end
end
```

Monitoring the traffic in the CLI

To view debugging information:

1. Verify the status of the phase 1 IKE SAs:

```
# diagnose vpn ike gateway list
vd: root/0
name: pri_HQ2
version: 1
interface: port1 11
addr: 172.16.200.1:500 -&gt; 172.16.202.1:500
 tun_id: 172.16.202.1
created: 1520s ago
IKE SA: created 1/2 established 1/1 time 10/10/10 ms
```

```
IPsec SA: created 2/2 established 1/1 time 0/0/0 ms
    id/spi: 173 dcdede154681579b/e32f4c48c4349fc0 direction: responder status: established
1498-1498s ago = 10ms proposal: aes128-sha256 key: d7230a68d7b83def-588b94495cfa9d38
lifetime/rekey: 86400/84631 DPD sent/recv: 0000000d/00000006
vd: root/0
name: sec_HQ2
version: 1
interface: port2 12
addr: 172.17.200.1:500 -&gt; 172.17.202.1:500
created: 1520s ago
IKE SA: created 1/2 established 1/1 time 10/10/10 ms
IPsec SA: created 2/2 established 1/1 time 0/0/0 ms
    id/spi: 174 a567bd7bf02a04b5/4251b6254660aee2 direction: responder status: established
1498-1498s ago = 10ms proposal: aes128-sha256 key: 9f44f500c28d8de6-febaae9d1e6a164c
lifetime/rekey: 86400/84631 DPD sent/recv: 00000008/0000000c
```

2. Verify the phase 2 IPsec tunnel SAs:

```
# diagnose vpn tunnel list
list all ipsec tunnel in vd 0
name=sec_HQ2 ver=1 serial=2 172.17.200.1:0-&gt;172.17.202.1:0 tun_id=172.17.202.1
bound_if=5 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/512 options[0200]=frag-rfc run_
state=1 accept_traffic=1
proxyid_num=1 child_num=0 refcnt=7 ilast=5 olast=5 ad=/0
stat: rxp=39 txp=40 rxb=5448 txb=2732
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=15
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=sec_HQ2 proto=0 sa=1 ref=2 serial=2 auto-negotiate
    src: 0:0.0.0.0/0.0.0.0:0 dst: 0:0.0.0.0/0.0.0.0:0 SA: ref=3 options=18227 type=00 soft=0
mtu=1438 expire=41230/0B replaywin=2048
        seqno=29 esn=0 replaywin_lastseq=00000028 itn=0
    life: type=01 bytes=0/0 timeout=42899/43200 dec: spi=1071b4f9 esp=aes key=16
1f4dbb78bea8e97650b52d8170b5ece7
        ah=sha1 key=20 cd9bf2de0f49296cf489dd915d7baf6d78bc8f12
    enc: spi=ec89b7ee esp=aes key=16 0546efecd0d1b9ba5944f635896e4404
        ah=sha1 key=20 34599bc7dc25e1ce63ac9615bd50928ce0667dc8
    dec:pkts/bytes=39/2796, enc:pkts/bytes=40/5456
name=pri_HQ2 ver=1 serial=1 172.16.200.1:0-&gt;172.16.202.1:0 tun_id=172.16.202.1
bound_if=11 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/512 options[0200]=frag-rfc run_
state=1 accept_traffic=1
proxyid_num=1 child_num=0 refcnt=5 ilast=15 olast=15 ad=/0
stat: rxp=38 txp=39 rxb=5152 txb=2768
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=20
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=pri_HQ2 proto=0 sa=1 ref=2 serial=2 auto-negotiate
    src: 0:0.0.0.0/0.0.0.0:0 dst: 0:0.0.0.0/0.0.0.0:0 SA: ref=3 options=18227 type=00 soft=0
mtu=1438 expire=41231/0B replaywin=2048
        seqno=28 esn=0 replaywin_lastseq=00000027 itn=0
    life: type=01 bytes=0/0 timeout=42900/43200 dec: spi=1071b4f8 esp=aes key=16
142cce377b3432ba41e64128ade6848c
        ah=sha1 key=20 20e64947e2397123f561584321adc0e7aa0c342d
    enc: spi=ec89b7ed esp=aes key=16 2ec13622fd60dacce3d28ebe5fe7ab14
```

```
        ah=sha1 key=20 c1787497508a87f40c73c0db0e835c70b3c3f42d
    dec:pkts/bytes=38/2568, enc:pkts/bytes=39/5432
```

3. Debug the IPsec aggregation list:

```
# diagnose sys ipsec-aggregate list
agg_HQ2 algo=RR member=2 run_tally=2
members:
    pri_HQ2
    sec_HQ2
```

4. Verify the OSPF neighbor information:

```
# get router info ospf neighbor
OSPF process 0, VRF 0:
Neighbor ID Pri State Dead Time Address Interface
2.2.2.2 1. Full/ - 00:00:34 10.10.10.2 agg1_HQ2
```

5. Verify the OSPF routing table:

```
# get router info routing-table ospf
Routing table for VRF=0
O 172.16.101.0/24 [110/20] via 10.10.10.2, agg1_HQ2 , 00:18:43
```

Packet distribution for aggregate dial-up IPsec tunnels using location ID

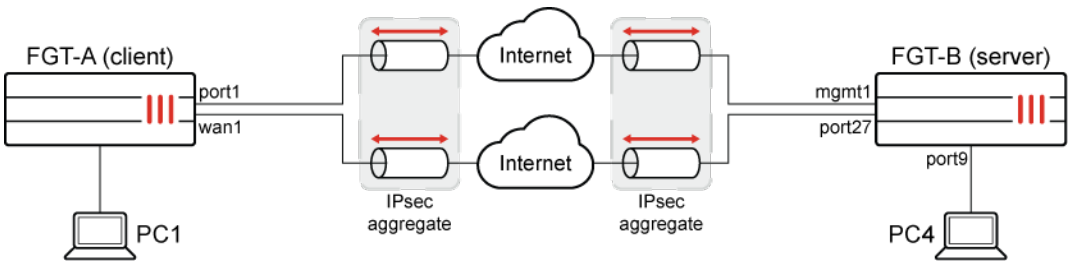
To support per-packet load balancing on aggregate dial-up IPsec tunnels between sites, each spoke must be configured with a location ID. On the hub, per-packet load balancing is performed on the tunnels in the IPsec aggregate that have the same location ID.

Multiple dial-up VPN tunnels from the same location can be aggregated on the VPN hub and load balanced based on the configured load balance algorithm.

IPsec traffic cannot be offloaded to the NPU.

Example

In this example, an IPsec aggregate tunnel is formed between two dial-up IPsec tunnels in order to support per-packet load balancing.



To configure the client FortiGate (FGT-A):

1. Configure the IPsec tunnels:

```
config vpn ipsec phase1-interface
    edit "client1"
        set interface "port1"
        set peertype any
        set net-device disable
        set aggregate-member enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set remote-gw 172.16.200.4
        set psksecret **********
    next
    edit "client2"
        set interface "wan1"
        set peertype any
        set net-device disable
        set aggregate-member enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set remote-gw 173.1.1.1
        set psksecret **********
    next
end
```

2. Configure an aggregate of the IPsec tunnels:

```
config system ipsec-aggregate
    edit "agg1"
        set member "client1" "client2"
    next
end
```

3. Configure the location ID:

```
config system settings
    set location-id 1.1.1.1
end
```

To configure the server FortiGate (FGT-B):

1. Configure the IPsec tunnels:

```
config vpn ipsec phase1-interface
    edit "server1"
        set type dynamic
        set interface "mgmt1"
        set peertype any
        set net-device disable
        set aggregate-member enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set dpd on-idle
```

```
        set psksecret **********
        set dpd-retryinterval 60
    next
    edit "server2"
        set type dynamic
        set interface "port27"
        set peertype any
        set net-device disable
        set aggregate-member enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set dpd on-idle
        set psksecret **********
        set dpd-retryinterval 60
    next
end
config vpn ipsec phase2-interface
    edit "server1"
        set phase1name "server1"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
    next
    edit "server2"
        set phase1name "server2"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
    next
end
```

2. Configure an aggregate of the IPsec tunnels:

```
config system ipsec-aggregate
    edit "server"
        set member "server1" "server2"
    next
end
```

3. Configure a firewall policy:

```
config firewall policy
    edit 1
        set srcintf "server"
        set dstintf "port9"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end
```

To check the IPsec tunnel and aggregate state:

1. List all of the VPN tunnels:

```
FGDocs # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=server1 ver=1 serial=1 172.16.200.4:500->0.0.0.0:500 tun_id=1.0.0.0 dst_mtu=0 dpd-link=on
remote_location=0.0.0.0 weight=1
bound_if=4 lgwy=static/1 tun=tunnel/15 mode=dialup/2 encap=none/4616 options[1208]=npu frag-
rfc  accept_traffic=1 overlay_id=0

proxyid_num=0 child_num=2 refcnt=4 ilast=14210 olast=14210 ad=/0
stat: rxp=798921 txp=819074 rxb=121435992 txb=68802216
dpd: mode=on-idle on=0 idle=60000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
run_tally=0
------------------------------------------------------
name=server2 ver=1 serial=2 173.1.1.1:500->0.0.0.0:500 tun_id=2.0.0.0 dst_mtu=0 dpd-link=on
remote_location=0.0.0.0 weight=1
bound_if=17 lgwy=static/1 tun=tunnel/15 mode=dialup/2 encap=none/4616 options[1208]=npu frag-
rfc  accept_traffic=1 overlay_id=0

proxyid_num=0 child_num=1 refcnt=3 ilast=14177 olast=14177 ad=/0
stat: rxp=836484 txp=819111 rxb=137429352 txb=80046050
dpd: mode=on-idle on=0 idle=60000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
run_tally=0
------------------------------------------------------
name=server1_0 ver=1 serial=8 172.16.200.4:500->172.16.200.1:500 tun_id=172.16.200.1 dst_
mtu=1500 dpd-link=on remote_location=1.1.1.1 weight=1
bound_if=4 lgwy=static/1 tun=tunnel/15 mode=dial_inst/3 encap=none/4744 options[1288]=npu
rgwy-chg frag-rfc  run_state=0 accept_traffic=1 overlay_id=0

parent=server1 index=0
proxyid_num=1 child_num=0 refcnt=5 ilast=45 olast=45 ad=/0
stat: rxp=17176 txp=17176 rxb=2610752 txb=1442784
dpd: mode=on-idle on=1 idle=60000ms retry=3 count=0 seqno=12
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=server1 proto=0 sa=1 ref=2 serial=1 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.1.100.0-10.1.100.255:0
  SA:  ref=3 options=2a6 type=00 soft=0 mtu=1438 expire=42342/0B replaywin=2048
       seqno=4319 esn=0 replaywin_lastseq=00004319 itn=0 qat=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43186/43200
  dec: spi=0aef2a07 esp=aes key=16 12738c8a1db02c23bfed73eb3615a5a1
       ah=sha1 key=20 0f3edd28e3165d184292b4cd397a6edeef9d20dc
  enc: spi=2cb75665 esp=aes key=16 982b418e40f0bb18b89916d8c92270c0
       ah=sha1 key=20 08cbf9bf78a968af5cd7647dfa2a0db066389929
  dec:pkts/bytes=17176/1442784, enc:pkts/bytes=17176/2610752
  npu_flag=00 npu_rgwy=172.16.200.1 npu_lgwy=172.16.200.4 npu_selid=6 dec_npuid=0 enc_npuid=0
------------------------------------------------------
name=server1_1 ver=1 serial=a 172.16.200.4:500->172.16.200.3:500 tun_id=172.16.200.3 dst_mtu=0
```

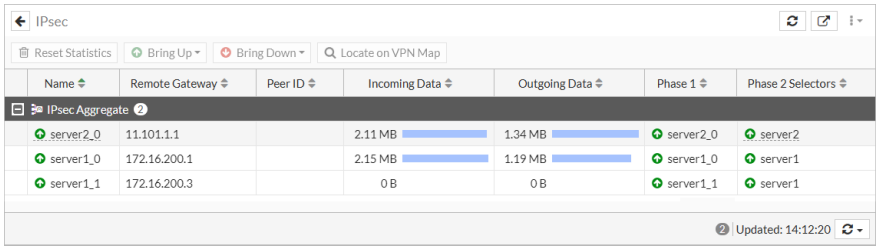
```
dpd-link=on remote_location=2.2.2.2 weight=1
bound_if=4 lgwy=static/1 tun=tunnel/15 mode=dial_inst/3 encap=none/4744 options[1288]=npu
rgwy-chg frag-rfc  run_state=0 accept_traffic=1 overlay_id=0

parent=server1 index=1
proxyid_num=1 child_num=0 refcnt=5 ilast=27 olast=27 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=1 idle=60000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=server1 proto=0 sa=1 ref=2 serial=1 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=3 options=2a6 type=00 soft=0 mtu=1280 expire=43167/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 itn=0 qat=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43187/43200
  dec: spi=0aef2a0a esp=aes key=16 4b7a17ba9d239e4ae5fe95ec100fca8b
       ah=sha1 key=20 7d3e058088f21e0c4f1c13c297293f06c8b592e7
  enc: spi=7e961809 esp=aes key=16 ecd1aa8657c5a509662aed45002d3990
       ah=sha1 key=20 d159e06c1cf0ded18a4e4ac86cbe5aa0315c21c9
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=172.16.200.3 npu_lgwy=172.16.200.4 npu_selid=9 dec_npuid=0 enc_npuid=0
------------------------------------------------------
name=server2_0 ver=1 serial=7 173.1.1.1:500->11.101.1.1:500 tun_id=11.101.1.1 dst_mtu=1500
dpd-link=on remote_location=1.1.1.1 weight=1
bound_if=17 lgwy=static/1 tun=tunnel/15 mode=dial_inst/3 encap=none/4744 options[1288]=npu
rgwy-chg frag-rfc  run_state=0 accept_traffic=1 overlay_id=0

parent=server2 index=0
proxyid_num=1 child_num=0 refcnt=5 ilast=45 olast=45 ad=/0
stat: rxp=16001 txp=17179 rxb=2113664 txb=1594824
dpd: mode=on-idle on=1 idle=60000ms retry=3 count=0 seqno=12
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=server2 proto=0 sa=1 ref=2 serial=1 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.1.100.0-10.1.100.255:0
  SA:  ref=6 options=2a6 type=00 soft=0 mtu=1438 expire=42342/0B replaywin=2048
       seqno=431a esn=0 replaywin_lastseq=00003e80 itn=0 qat=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43185/43200
  dec: spi=0aef2a08 esp=aes key=16 394d4e444e90ccb5184e744d49aabe3c
       ah=sha1 key=20 faabea35c2b9b847461cbd263c4856cfb679f342
  enc: spi=2cb75666 esp=aes key=16 0b3a2fbac4d5610670843fa1925d1207
       ah=sha1 key=20 97e99beff3d8f61a8638f6ef887006a9c323acd4
  dec:pkts/bytes=16001/2113596, enc:pkts/bytes=17179/2762792
  npu_flag=03 npu_rgwy=11.101.1.1 npu_lgwy=173.1.1.1 npu_selid=7 dec_npuid=1 enc_npuid=1
```

2. List the IPsec aggregate members:

```
# diagnose sys ipsec-aggregate list
server
members(3):
        server1_1
```

```
        server1_0
        server2_0
```

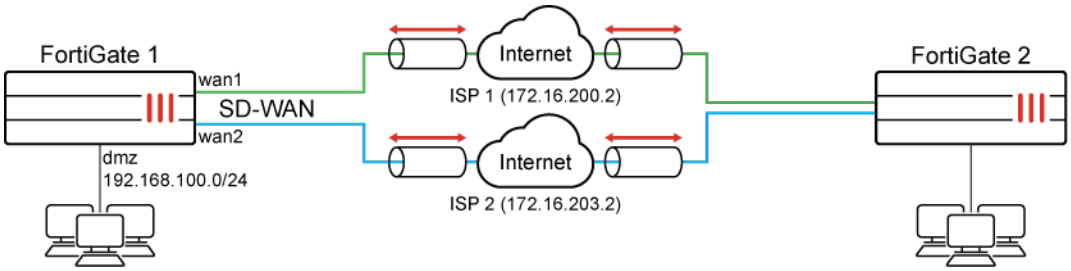
3. In the GUI, go to *Dashboard* > *Network Monitor* > *VPN* to review the traffic distributed over the aggregate members:



Packet distribution for aggregate static IPsec tunnels in SD-WAN

This is a sample configuration of aggregating IPsec tunnels by using per-packet load-balancing.

For example, a customer has two ISP connections, wan1 and wan2. On each FortiGate, two IPsec VPN interfaces are created. Next, an `ipsec-aggregate` interface is created and added as an SD-WAN member.



Configuring FortiGate 1

To create two IPsec VPN interfaces:

```
config vpn ipsec phase1-interface
    edit "vd1-p1"
        set interface "wan1"
        set peertype any
        set net-device disable
        set aggregate-member enable
        set proposal aes256-sha256
        set dhgrp 14
        set remote-gw 172.16.201.2
        set psksecret ftnt1234
    next
    edit "vd1-p2"
        set interface "wan2"
```

```
        set peertype any
        set net-device disable
        set aggregate-member enable
        set proposal aes256-sha256
        set dhgrp 14
        set remote-gw 172.16.202.2
        set psksecret ftnt1234
    next
end
```

```
config vpn ipsec phase2-interface
    edit "vd1-p1"
        set phase1name "vd1-p1"
    next
    edit "vd1-p2"
        set phase1name "vd1-p2"
    next
end
```

To create an IPsec aggregate interface:

```
config system ipsec-aggregate
    edit "agg1"
        set member "vd1-p1" "vd1-p2"
        set algorithm L3
    next
end
```

```
config system interface
    edit "agg1"
        set vdom "root"
        set ip 172.16.11.1 255.255.255.255
        set allowaccess ping
        set remote-ip 172.16.11.2 255.255.255.255
end
```

To configure the firewall policy:

```
config firewall policy
    edit 1
        set name "1"
        set srcintf "dmz"
        set dstintf ""virtual-wan-link""
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set nat enable
```

```
    next
end
```

To configure SD-WAN:

```
config system sdwan
    set status enable
    config members
        edit 1
            set interface "agg1"
            set gateway 172.16.11.2
        next
    end
end
```

Configuring FortiGate 2

To create two IPsec VPN interfaces:

```
config vpn ipsec phase1-interface
    edit "vd2-p1"
        set interface "wan1"
        set peertype any
        set net-device disable
        set proposal aes256-sha256
        set dhgrp 14
        set remote-gw 172.16.200.1
        set psksecret ftnt1234
    next
    edit "vd2-p2"
        set interface "wan2"
        set peertype any
        set net-device disable
        set proposal aes256-sha256
        set dhgrp 14
        set remote-gw 172.16.203.1
        set psksecret ftnt1234
    next
end
```

```
config vpn ipsec phase2-interface
    edit "vd2-p1"
        set phase1name "vd2-p1"
    next
    edit "vd2-p2"
        set phase1name "vd2-p2"
    next
end
```

To create an IPsec aggregate interface:

```
config system ipsec-aggregate
    edit "agg2"
        set member "vd2-p1" "vd2-p2"
        set algorithm L3
    next
end
```

```
config system interface
    edit "agg2"
        set vdom "root"
        set ip 172.16.11.2 255.255.255.255
        set allowaccess ping
        set remote-ip 172.16.11.1 255.255.255.255
    next
end
```

To configure the firewall policy:

```
config firewall policy
    edit 1
        set name "1"
        set srcintf "dmz"
        set dstintf ""virtual-wan-link""
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

To configure SD-WAN:

```
config system sdwan
    set status enable
    config members
        edit 1
            set interface "agg2"
            set gateway 172.16.11.1
        next
    end
end
```

Related diagnose commands

To display aggregate IPsec members:

```
# diagnose sys ipsec-aggregate list
agg1 algo=L3 member=2 run_tally=2
members:
        vd1-p1
        vd1-p2
```

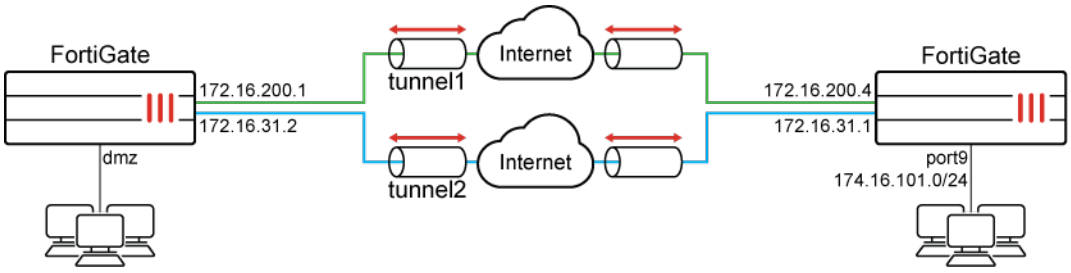
To check the VPN status:

```
# diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=vd1-p1 ver=1 serial=2 172.16.200.1:0->172.16.201.2:0 tun_id=172.16.201.2 dst_mtu=0
bound_if=10 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/520 options[0208]=npu frag-rfc  run_
state=1 accept_traffic=0

proxyid_num=1 child_num=0 refcnt=5 ilast=15 olast=676 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=vd1-p1 proto=0 sa=0 ref=1 serial=1
  src: 0:0.0.0.0/0.0.0.0:0
  dst: 0:0.0.0.0/0.0.0.0:0
------------------------------------------------------
name=vd1-p2 ver=1 serial=3 172.16.203.1:0->172.16.202.2:0 tun_id=172.16.202.2 dst_mtu=1500
bound_if=28 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/520 options[0208]=npu frag-rfc  run_
state=1 accept_traffic=1

proxyid_num=1 child_num=0 refcnt=12 ilast=1 olast=1 ad=/0
stat: rxp=1 txp=1686 rxb=16602 txb=111717
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=vd1-p2 proto=0 sa=1 ref=9 serial=1
  src: 0:0.0.0.0/0.0.0.0:0
  dst: 0:0.0.0.0/0.0.0.0:0
  SA:  ref=4 options=10226 type=00 soft=0 mtu=1438 expire=42164/0B replaywin=2048
       seqno=697 esn=0 replaywin_lastseq=00000002 itn=0 qat=0
  life: type=01 bytes=0/0 timeout=42902/43200
  dec: spi=f6ae9f83 esp=aes key=16 f6855c72295e3c5c49646530e6b96002
       ah=sha1 key=20 f983430d6c161d0a4cd9007c7ae057f1ff011334
  enc: spi=8c72ba1a esp=aes key=16 6330f8c532a6ca5c5765f6a9a6034427
       ah=sha1 key=20 e5fe385ed5f0f6a33f1d507601b15743a8c70187
  dec:pkts/bytes=1/16536, enc:pkts/bytes=1686/223872
  npu_flag=02 npu_rgwy=172.16.202.2 npu_lgwy=172.16.203.1 npu_selid=2 dec_npuid=1 enc_npuid=0
```

Packet distribution for aggregate IPsec tunnels using weighted round robin

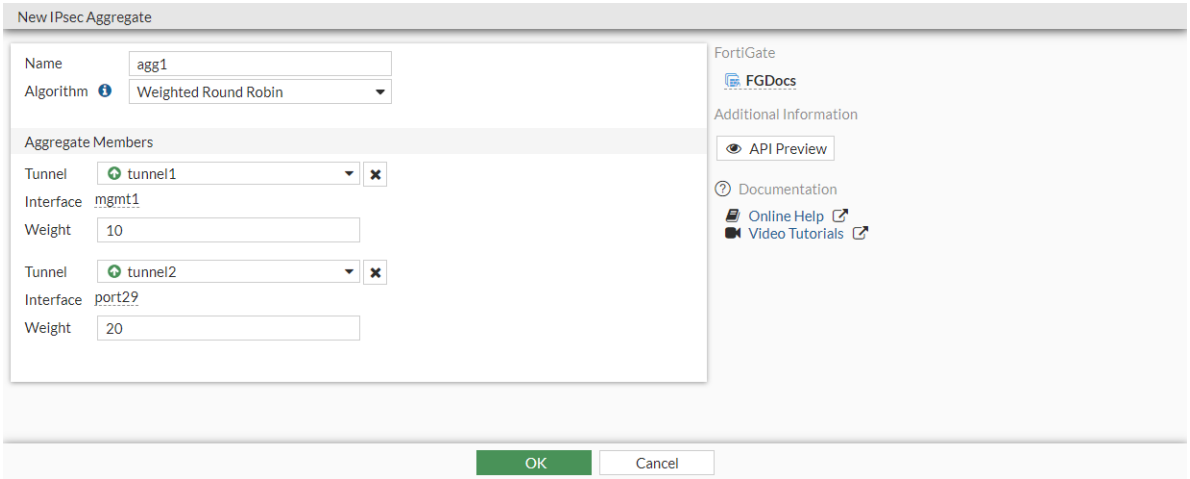
A weighted round robin algorithm can be used for IPsec aggregate tunnels to distribute traffic by the weight of each member tunnel.

In this example, the FortiGate has two IPsec tunnels put into IPsec aggregate. Traffic is distributed among the members, with one third over *tunnel1*, and two thirds over *tunnel2*. To achieve this, the weighted round robin algorithm is selected, *tunnel1* is assigned a weight of 10, and *tunnel2* is assigned a weight of 20.



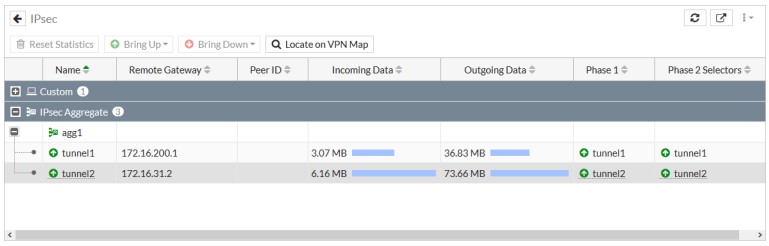
To create the IPsec aggregate in the GUI:

1. Go to *VPN* > *IPsec Tunnels* and click *Create New* > *IPsec Tunnel*.
2. Complete the wizard to create the *tunnel1* and *tunnel2* custom IPsec tunnels. Ensure that *Aggregate member* is *Enabled* for each tunnel under the *Network* > *Advanced* section.
3. Go to *VPN* > *IPsec Tunnels* and click *Create New* > *IPsec Aggregate*.
4. Enter a name for the aggregate, such as *agg1*, and ensure that *Algorithm* is *Weighted Round Robin*.
5. Add *tunnel1* as an aggregate members, and set *Weight* to *10*.
6. Add *tunnel2* as a second aggregate members, and set its *Weight* to *20*.



7. Click *OK*.

8. To view and monitor the aggregate tunnel statistics, go to the *IPsec* widget on the *Network* dashboard.



To create the IPsec aggregate in the CLI:

1. Create the *tunnel1* and *tunnel2* custom IPsec tunnels with aggregate-member enabled and aggregate-weight set for both tunnels:

```
config vpn ipsec phase1-interface
    edit "tunnel1"
        ...
        set aggregate-member enable
        set aggregate-weight 10
        ...
    next
    edit "tunnel2"
        ...
        set aggregate-member enable
        set aggregate-weight 20
        ...
    next
end
```

2. Create the IPsec aggregate:

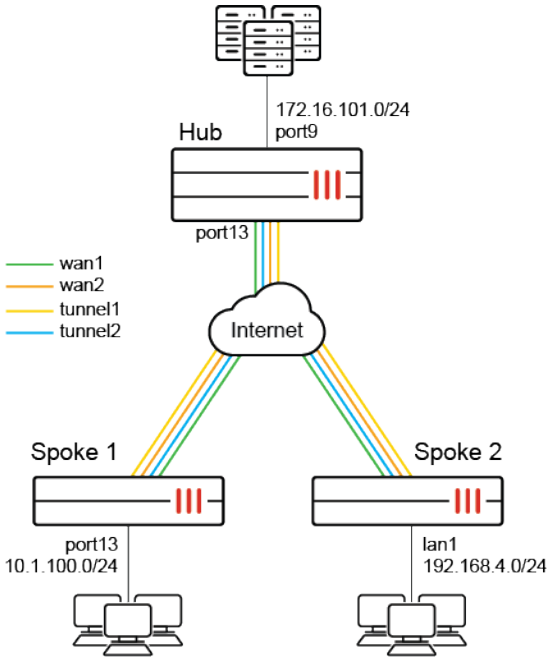
```
config system ipsec-aggregate
    edit "agg1"
        set member "tunnel1" "tunnel2"
        set algorithm weighted-round-robin
    next
end
```

Redundant hub and spoke VPN

A redundant hub and spoke configuration allows VPN connections to radiate from a central FortiGate unit (the hub) to multiple remote peers (the spokes). Traffic can pass between private networks behind the hub and private networks behind the remote peers. Traffic can also pass between remote peer private networks through the hub.

This is a sample configuration of hub and spoke IPsec VPN. The following applies for this scenario:

- The spokes have two WAN interfaces and two IPsec VPN tunnels for redundancy.
- The secondary VPN tunnel is up only when the primary tunnel is down by dead peer detection.



Because the GUI can only complete part of the configuration, we recommend using the CLI.

To configure redundant hub and spoke VPN using the FortiOS CLI:

1. Configure the hub.
   a. Configure the WAN, internal interface, and static route:

```
config system interface
    edit "port13"
        set alias "WAN"
        set ip 172.16.202.1 255.255.255.0
    next
    edit "port9"
        set alias "Internal"
        set ip 172.16.101.1 255.255.255.0
    next
end
config router static
    edit 1
        set gateway 172.16.202.2
        set device "port13"
    next
end
```

   b. Configure the IPsec phase1-interface and phase2-interface:

```
config vpn ipsec phase1-interface
    edit "hub"
        set type dynamic
        set interface "port13"
        set peertype any
```

```
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set dpd on-idle
        set psksecret sample
        set dpd-retryinterval 60
    next
end
config vpn ipsec phase2-interface
    edit "hub"
        set phase1name "hub"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
    next
end
```

c. Configure the firewall policy:

```
config firewall policy
    edit 1
        set name "spoke-hub"
        set srcintf "hub"
        set dstintf "port9"
        set srcaddr "all"
        set dstaddr "172.16.101.0"
        set action accept
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set name "spoke-spoke"
        set srcintf "hub"
        set dstintf "hub"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end
```

2. Configure the spokes.

a. Configure the WAN, internal interface, and static route:

i. Configure Spoke1:

```
config system interface
    edit "port1"
        set ip 172.16.200.1 255.255.255.0
    next
    edit "wan1"
        set mode dhcp
        set distance 10
```

```
        set priority 100
    next
    edit "dmz"
        set ip 10.1.100.1 255.255.255.0
    next
end
config router static
    edit 1
        set gateway 172.16.200.2
        set device "port1"
    next
end
```

ii. Configure Spoke2:

```
config system interface
    edit "wan1"
        set ip 172.16.200.3 255.255.255.0
    next
    edit "wan2"
        set mode dhcp
        set distance 10
        set priority 100
    next
    edit "lan1"
        set ip 192.168.4.1 255.255.255.0
    next
end
config router static
    edit 1
        set gateway 172.16.200.2
        set device "wan1"
    next
end
```

b. Configure IPsec phase1-interface and phase2-interface:

i. Configure Spoke1:

```
config vpn ipsec phase1-interface
    edit "primary"
        set interface "port1"
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set remote-gw 172.16.202.1
        set psksecret sample
        next
    edit "secondary"
        set interface "wan1"
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
```

```
        set remote-gw 172.16.202.1
        set monitor "primary"
        set psksecret sample
    next
end
config vpn ipsec phase2-interface
    edit "primary"
        set phase1name "primary"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set auto-negotiate enable
        set src-subnet 10.1.100.0 255.255.255.0
    next
    edit "secondary"
        set phase1name "secondary"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set auto-negotiate enable
        set src-subnet 10.1.100.0 255.255.255.0
    next
end
```

ii. Configure Spoke2:

```
config vpn ipsec phase1-interface
    edit "primary"
        set interface "wan1"
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set remote-gw 172.16.202.1
        set psksecret sample
        next
    edit "secondary"
        set interface "wan2"
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set remote-gw 172.16.202.1
        set monitor "primary"
        set psksecret sample
    next
end
config vpn ipsec phase2-interface
    edit "primary"
        set phase1name "primary"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set auto-negotiate enable
        set src-subnet 192.168.4.0 255.255.255.0
    next
    edit "secondary"
```

```
        set phase1name "secondary"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set auto-negotiate enable
        set src-subnet 192.168.4.0 255.255.255.0
    next
end
```

c. Configure the firewall policy:

i. Configure Spoke1:

```
config firewall policy
    edit 1
        set srcintf "dmz"
        set dstintf "primary" "secondary"
        set srcaddr "10.1.100.0"
        set dstaddr "172.16.101.0"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end
```

ii. Configure Spoke2:

```
config firewall policy
    edit 1
        set srcintf "lan1"
        set dstintf "primary" "secondary"
        set srcaddr "192.168.4.0"
        set dstaddr "172.16.101.0"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end
```

d. Configure the static route:

i. Configure Spoke1:

```
config router static
    edit 3
        set dst 172.16.101.0 255.255.255.0
        set distance 1
        set device "primary"
    next
    edit 4
        set dst 172.16.101.0 255.255.255.0
        set distance 3
        set device "secondary"
    next
end
```

ii. Configure Spoke2:

```
config router static
    edit 3
        set dst 172.16.101.0 255.255.255.0
        set distance 1
        set device "primary"
    next
    edit 4
        set dst 172.16.101.0 255.255.255.0
        set distance 3
        set device "secondary"
    next
end
```

3. Run `diagnose` and `get` commands.

a. Run the `Spoke1 # diagnose vpn tunnel list` command. The system should return the following:

```
name=primary ver=1 serial=1 172.16.200.1:0-&gt;172.16.202.1:0 tun_id=172.16.202.1
bound_if=11 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/528 options[0210]=create_dev
frag-rfc accept_traffic=1
proxyid_num=1 child_num=0 refcnt=15 ilast=0 olast=0 ad=/0
stat: rxp=1879 txp=1881 rxb=225480 txb=112860
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=1
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=primary proto=0 sa=1 ref=2 serial=2 auto-negotiate
    src: 0:10.1.100.0/255.255.255.0:0 dst: 0:0.0.0.0/0.0.0.0:0 SA: ref=3 options=18227
type=00 soft=0 mtu=1438 expire=41002/0B replaywin=2048
        seqno=758 esn=0 replaywin_lastseq=00000758 itn=0
    life: type=01 bytes=0/0 timeout=42901/43200 dec: spi=0908732f esp=aes key=16
20770dfe67ea22dd8ec32c44d84ef4d5
        ah=sha1 key=20 edc89fc2ec06309ba13de95e7e486f9b795b8707
    enc: spi=a1d9eed1 esp=aes key=16 8eeea2526fba062e680d941083c8b5d1
        ah=sha1 key=20 f0f5deaf88b2a69046c3154e9f751739b3f411f5
    dec:pkts/bytes=1879/112740, enc:pkts/bytes=1879/225480
name=secondary ver=1 serial=2 172.17.200.1:0-&gt;172.16.202.1:0 tun_id=172.16.202.1
bound_if=5 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/528 options[0210]=create_dev
frag-rfc accept_traffic=0
proxyid_num=1 child_num=0 refcnt=10 ilast=1892 olast=1892 ad=/0
       stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-demand on=0 idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=secondary proto=0 sa=0 ref=2 serial=2 auto-negotiate
    src: 0:10.1.100.0/255.255.255.0:0 dst: 0:0.0.0.0/0.0.0.0:0
```

b. Run the `Spoke1 # get router info routing-table static` command. The system should return the following:

```
Routing table for VRF=0
.................
S 172.16.101.0/24 [1/0] is directly connected, primary
```

ADVPN

Auto-Discovery VPN (ADVPN) allows the central hub to dynamically inform spokes about a better path for traffic between two spokes.

The following topics provide instructions on configuring ADVPN:

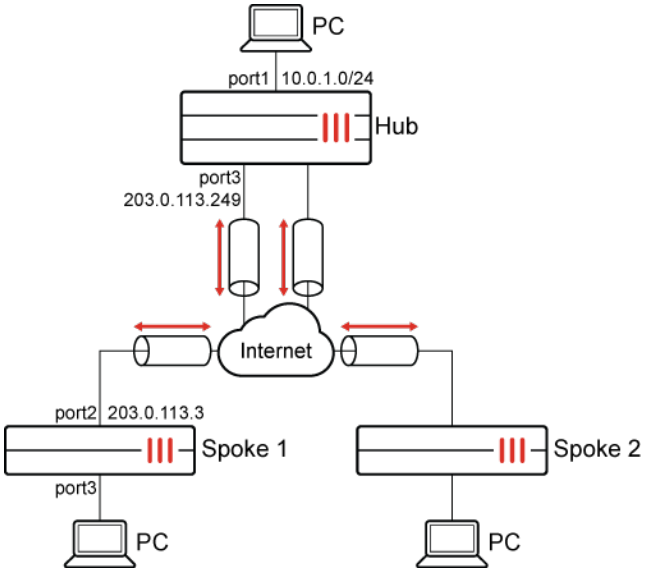
- IPsec VPN wizard hub-and-spoke ADVPN support on page 3235
- ADVPN with BGP as the routing protocol on page 3244
- ADVPN with OSPF as the routing protocol on page 3255
- ADVPN with RIP as the routing protocol on page 3264
- UDP hole punching for spokes behind NAT on page 3274

IPsec VPN wizard hub-and-spoke ADVPN support

Auto Discovery VPN (ADVPN) enables spokes in a hub-and-spoke topology to dynamically establish direct tunnels (shortcuts) with each other, bypassing the hub for spoke-to-spoke traffic. This reduces latency and avoids bottlenecking traffic through a central hub.

When using the IPsec VPN wizard to create a hub-and-spoke VPN, multiple local interfaces can be selected. At the end of the wizard, changes can be reviewed, real-time updates can be made to the local address group and tunnel interface, and easy configuration keys can be copied for configuring the spokes. These keys pre-populate spoke settings such as the hub tunnel IP, simplifying deployment and reducing the risk of misconfiguration.

This example shows the configuration of a hub with two spokes.



Configuration Steps

To configure the hub:

1. Go to *VPN > VPN Wizard*.
2. Enter the tunnel name *ADVPN*.
3. Select the Template *Hub and Spoke with ADVPN*.
4. Click *Begin*.
5. On the first page, select the Role *Hub*.
6. Click *Next*.
7. In the *VPN tunnel* configuration section:

Setting	Value
Authentication method	Pre-shared Key. For added security, use a digital signature
Pre-shared key	<key>
IKE	Version 2

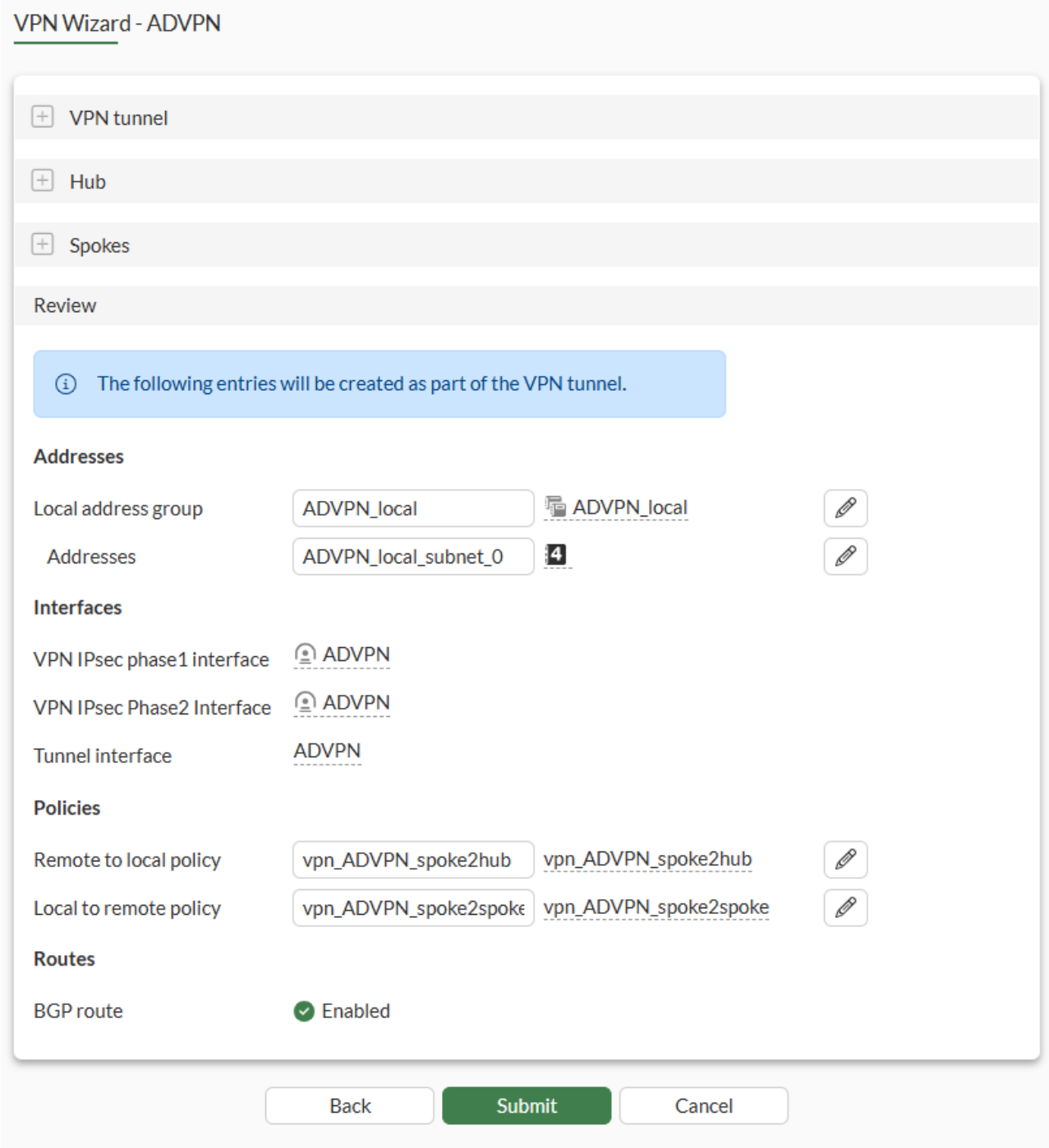
8. Click *Next*.
9. In the *Hub* configuration section:

Setting	Value
Incoming interface that binds to tunnel	port3
Create and add interface to zone	Disable. Enable to have the interface added to a zone
Hub tunnel IP/netmask	10.10.1.1/24
Local interface	port1
Local subnets that can access VPN	10.0.1.0/24
Local AS	65400

10. Click *Next*.
11. In the *Spoke* configuration section:

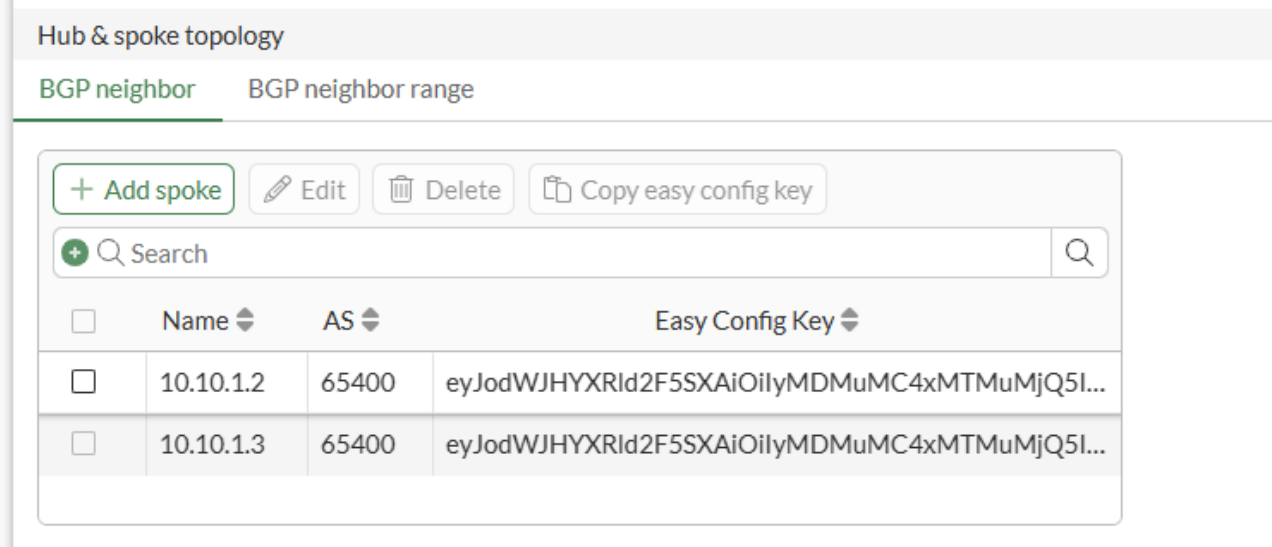
Setting	Value
Configure spoke tunnel IP	Individually
Spoke tunnel IP	10.10.1.2
	10.10.1.3

12. Click *Next*.
13. *Review Settings*:



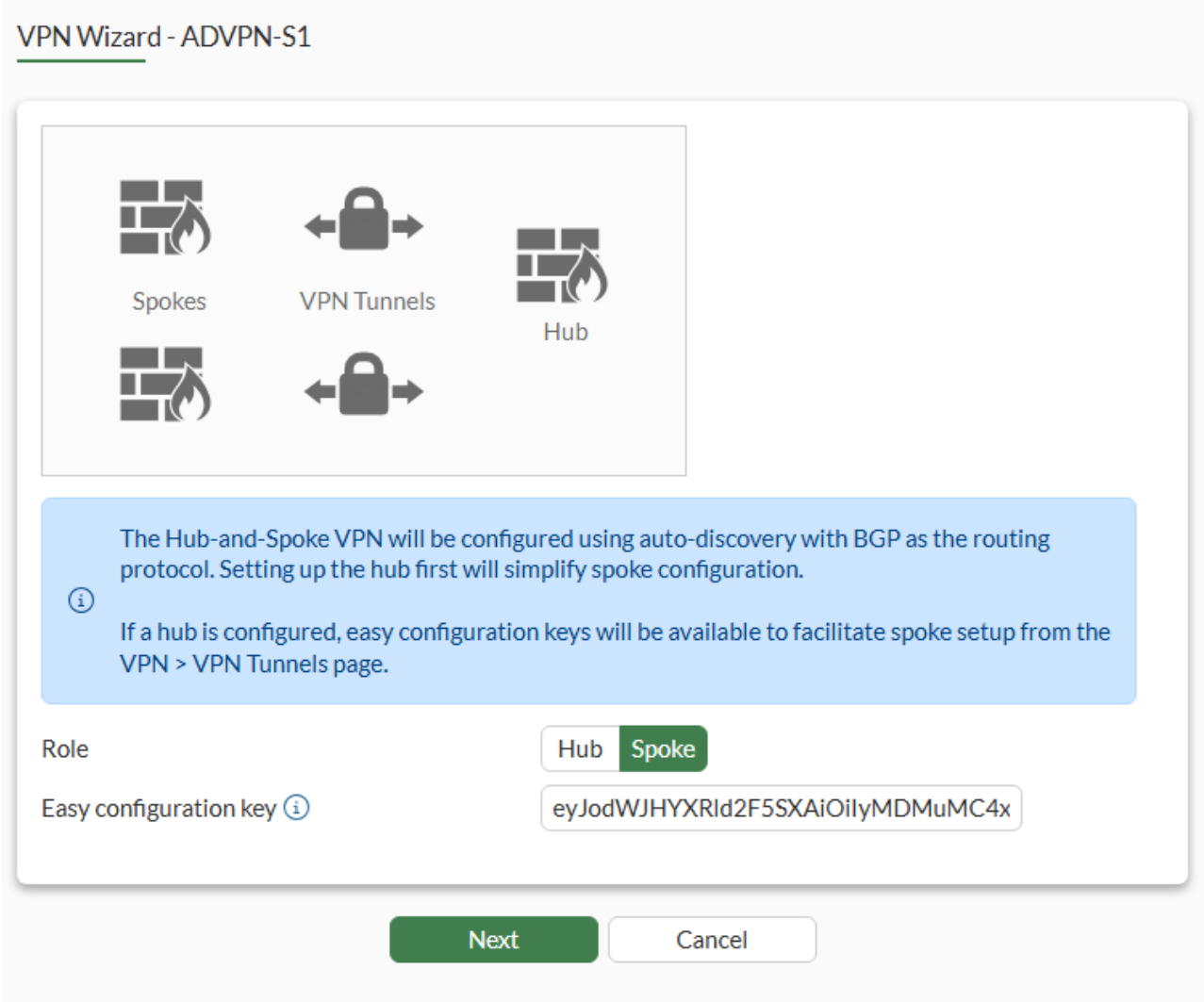
Confirm that the settings look correct, then click *Submit*.

14. In the VPN Tunnels list page, click the new ADVPN tunnel to view its settings.
15. Under Hub & spoke topology, see the 2 Spokes and their Easy Config Key. These can be used to configure the spoke FortiGates.



To configure the spokes:

1. Go to *VPN > IPsec Wizard*.
2. Enter the tunnel name *ADVPN-S1*
3. Select the Template *Hub and Spoke with ADVPN*
4. Click *Begin*.
5. On the first page, select the Role *Spoke*
6. Using the Easy Config Key for Spoke 1 (10.10.1.2), paste the key to Easy configuration key.



7. Click *Next*.
8. In the *VPN tunnel* configuration section:

Setting	Value
Authentication method	Pre-shared Key. For added security, use a digital signature
Pre-shared key	<key>
IKE	Version 2

9. Click Next
10. In the *Spoke* configuration section:

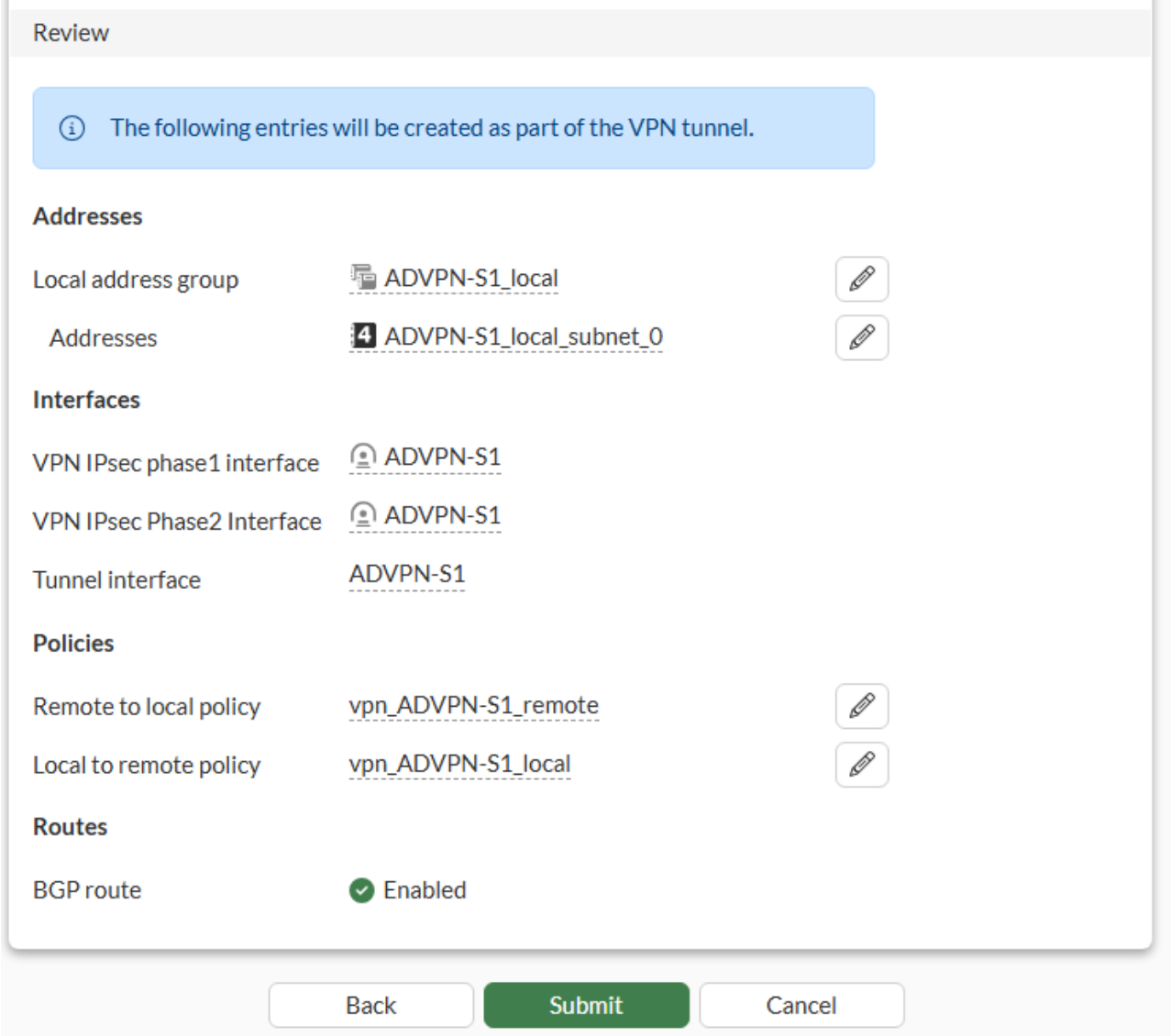
Setting	Value
Incoming interface that binds to tunnel	port2

Setting	Value
Create and add interface to zone	Disable. Enable to have the interface added to a zone
Spoke tunnel IP	10.10.1.2 (Pre-configured by the Key)
Local interface	port3
Local subnets that can access VPN	10.0.1.0/24
Local AS	65400

11. Click Next
12. In the *Hub* configuration section:

Setting	Value
Hub tunnel IP/netmask	10.10.1.1 255.255.255.0 (Pre-configured by the Key)
Hub public IP address	203.0.113.249 (Pre-configured by the Key)

13. Review settings.

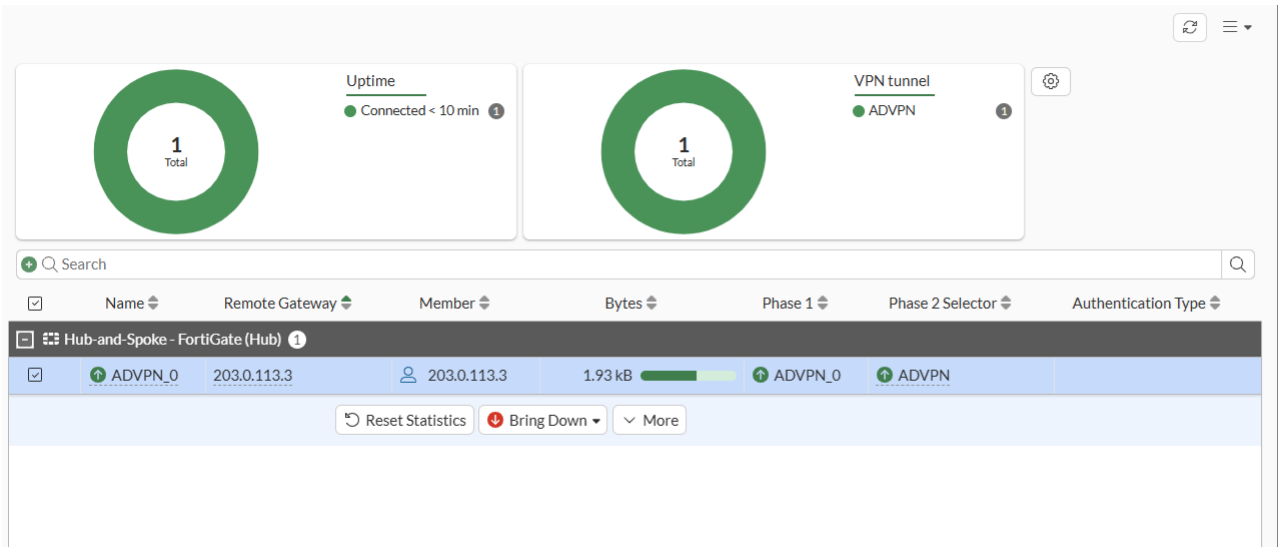


Confirm that the settings look correct, then click *Submit*.

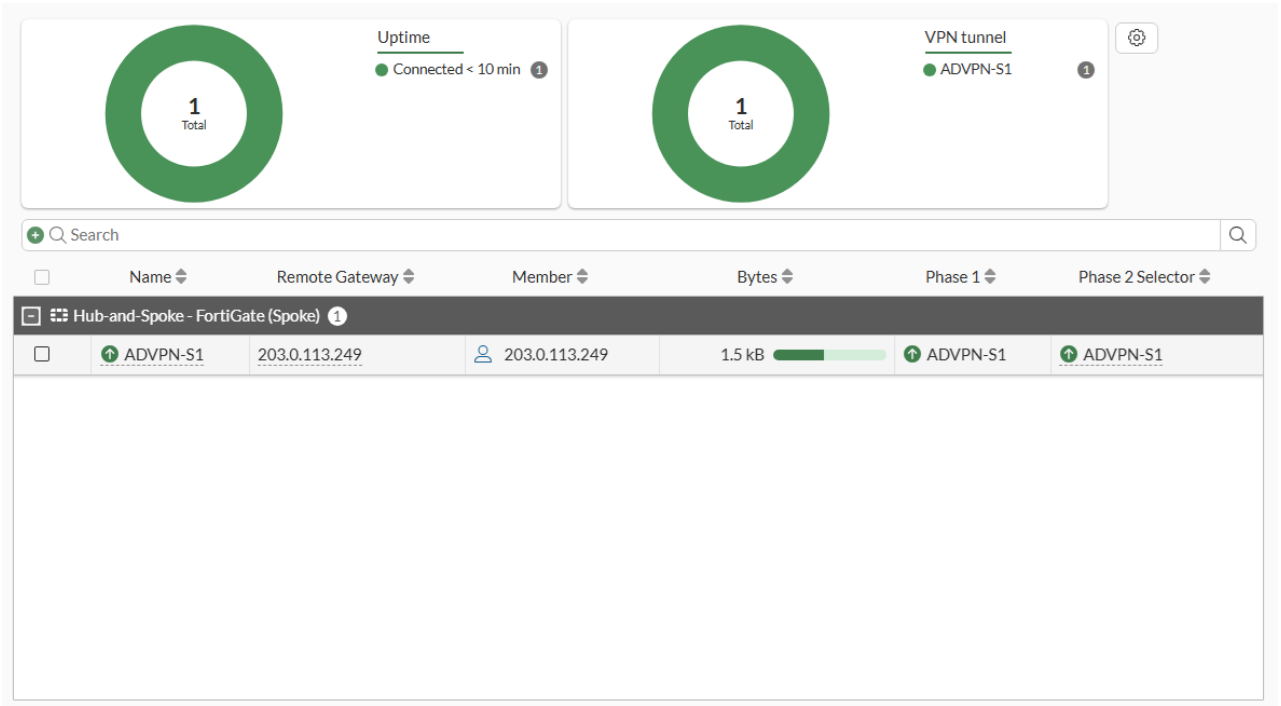
14. Follow the same steps to configure the second spoke. Use the Easy Config Key for 10.10.1.3

Tunnel Verification

1. On the hub FortiGate, go to *Dashboard* > *Network Monitor* > *VPN*.
   The tunnels to the spokes are established. Each Spoke will have a tunnel to the Hub.



2. On a spoke, go to *Dashboard* > *Network Monitor* > *VPN*.
   The tunnel to the hub is established. If another Spoke is up, a spoke to spoke shortcut is also established.



CLI configurations

Hub:

```
config vpn ipsec phase1-interface
    edit "ADVPN"
        set type dynamic
        set interface "port3"
```

```
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set add-route disable
        set dpd on-idle
        set dhgrp 20 21
        set wizard-type hub-fortigate-auto-discovery
        set auto-discovery-sender enable
        set psksecret ENC \<key\>
    next
end
```

```
config vpn ipsec phase2-interface
    edit "ADVPN"
        set phase1name "ADVPN"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
    next
end
```

```
config firewall policy
    edit 21
        set name "vpn_ADVPN_spoke2hub"
        set srcintf "ADVPN"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "ADVPN_local"
        set schedule "always"
        set service "ALL"
    next
    edit 22
        set name "vpn_ADVPN_spoke2spoke"
        set srcintf "ADVPN"
        set dstintf "ADVPN"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
    next
end
```

Spoke:

```
config vpn ipsec phase1-interface
    edit "ADVPN-S1"
        set interface "port2"
        set ike-version 2
```

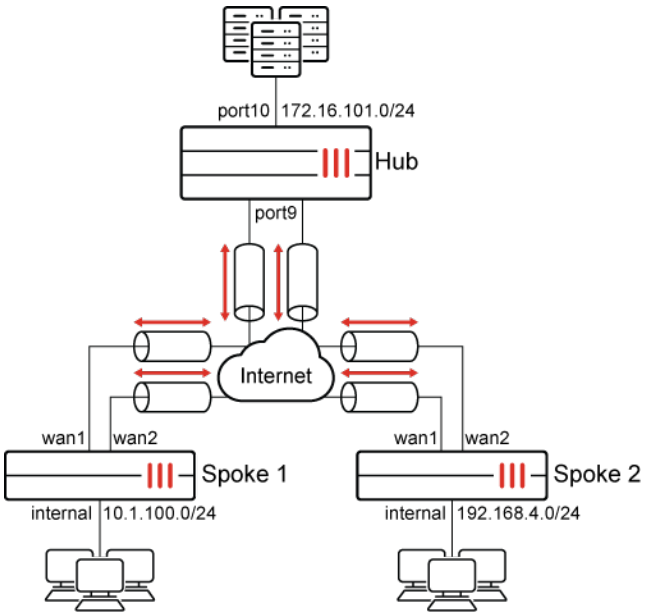
```
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set add-route disable
        set dpd on-idle
        set dhgrp 20 21
        set wizard-type spoke-fortigate-auto-discovery
        set auto-discovery-receiver enable
        set transport auto
        set remote-gw 203.0.113.249
        set psksecret ENC \<key\>
    next
end
config vpn ipsec phase2-interface
    edit "ADVPN-S1"
        set phase1name "ADVPN-S1"
        set proposal aes128-sha256 aes256-sha256 aes128gcm aes256gcm chacha20poly1305
        set dhgrp 20 21
    next
end
```

```
config firewall policy
    edit 5
        set name "vpn_ADVPN-S1_remote"
        set srcintf "ADVPN-S1"
        set dstintf "port3"
        set action accept
        set srcaddr "all"
        set dstaddr "ADVPN-S1_local"
        set schedule "always"
        set service "ALL"
    next
    edit 6
        set name "vpn_ADVPN-S1_local"
        set srcintf "port3"
        set dstintf "ADVPN-S1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
    next
end
```

ADVPN with BGP as the routing protocol

This is a sample configuration of ADVPN with BGP as the routing protocol. The following options must be enabled for this configuration:

- On the hub FortiGate, IPsec phase1-interface `net-device disable` must be run.
- IBGP must be used between the hub and spoke FortiGates.
- `bgp neighbor-group/neighbor-range` must be reused.



To configure ADVPN with BGP as the routing protocol using the CLI:

1. Configure hub FortiGate WAN interface, internal interface, and a static route:

```
config system interface
    edit "port9"
        set alias "WAN"
        set ip 22.1.1.1 255.255.255.0
    next
    edit "port10"
        set alias "Internal"
        set ip 172.16.101.1 255.255.255.0
    next
end
config router static
    edit 1
        set gateway 22.1.1.2
        set device "port9"
    next
end
```

2. Configure the hub FortiGate:
   a. Configure the hub FortiGate IPsec phase1-interface and phase2-interface:

```
config vpn ipsec phase1-interface
    edit "advpn-hub"
        set type dynamic
        set interface "port9"
```

```
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 3des-sha256 aes128-sha1 aes256-sha1 3des-
sha1
        set add-route disable
        set dpd on-idle
        set auto-discovery-sender enable
        set psksecret sample
        set dpd-retryinterval 5
    next
end
config vpn ipsec phase2-interface
    edit "advpn-hub"
        set phase1name "advpn-hub"
        set proposal aes128-sha1 aes256-sha1 3des-sha1 aes128-sha256 aes256-sha256 3des-
sha256
    next
end
```

When `net-device` is disabled, a tunnel ID is generated for each dynamic tunnel. This ID, in the form of an IP address, is used as the gateway in the route entry to that tunnel. The `tunnel-search` option is removed in FortiOS 7.0.0 and later.

b. Configure the hub FortiGate firewall policy:

```
config firewall policy
    edit 1
        set name "spoke2hub"
        set srcintf "advpn-hub"
        set dstintf "port10"
        set srcaddr "all"
        set dstaddr "172.16.101.0"
        set action accept
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set name "spoke2spoke"
        set srcintf "advpn-hub"
        set dstintf "advpn-hub"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end
```

c. Configure the hub FortiGate's IPsec tunnel interface IP address:

```
config system interface
    edit "advpn-hub1"
```

```
        set ip 10.10.10.254 255.255.255.255
        set remote-ip 10.10.10.253 255.255.255.0
    next
end
```

d. Configure the hub FortiGate's BGP:

```
config router bgp
    set as 65412
    config neighbor-group
        edit "advpn"
            set link-down-failover enable
            set remote-as 65412
            set route-reflector-client enable
        next
    end
    config neighbor-range
        edit 1
            set prefix 10.10.10.0 255.255.255.0
            set neighbor-group "advpn"
        next
    end
    config network
        edit 1
            set prefix 172.16.101.0 255.255.255.0
        next
    end
end
```

3. Configure the spoke FortiGates:

a. Configure the spoke FortiGates' WAN, internal interfaces, and static routes:

i. Configure Spoke1:

```
config system interface
    edit "wan1"
        set alias "primary_WAN"
        set ip 15.1.1.2 255.255.255.0
    next
    edit "wan2"
        set alias "secondary_WAN"
        set ip 12.1.1.2 255.255.255.0
    next
    edit "internal"
        set ip 10.1.100.1 255.255.255.0
    next
end
config router static
    edit 1
        set gateway 12.1.1.1
        set device "wan2"
        set distance 15
```

```
    next
    edit 2
        set gateway 15.1.1.1
        set device "wan1"
    next
end
```

ii. Configure the Spoke2:

```
config system interface
    edit "wan1"
        set alias "primary_WAN"
        set ip 13.1.1.2 255.255.255.0
    next
    edit "wan2"
        set alias "secondary_WAN"
        set ip 17.1.1.2 255.255.255.0
    next
    edit "internal"
        set ip 192.168.4.1 255.255.255.0
    next
end
config router static
    edit 1
        set gateway 17.1.1.1
        set device "wan2"
        set distance 15
    next
    edit 2
        set gateway 13.1.1.1
        set device "wan1"
    next
end
```

b. Configure the spoke FortiGates' IPsec phase1-interface and phase2-interface:

i. Configure Spoke1:

```
config vpn ipsec phase1-interface
    edit "spoke1"
        set interface "wan1"
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set add-route disable
        set dpd on-idle
        set auto-discovery-receiver enable
        set remote-gw 22.1.1.1
        set psksecret sample
        set dpd-retryinterval 5
    next
    edit "spoke1_backup"
        set interface "wan2"
```

```
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set add-route disable
        set dpd on-idle
        set auto-discovery-receiver enable
        set remote-gw 22.1.1.1
        set monitor "spoke1"
        set psksecret sample
        set dpd-retryinterval 5
    next
end
config vpn ipsec phase2-interface
    edit "spoke1"
        set phase1name "spoke1"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set auto-negotiate enable
    next
    edit "spoke1_backup"
        set phase1name "spoke1_backup"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set auto-negotiate enable
    next
end
```

ii. Configure Spoke2:

```
config vpn ipsec phase1-interface
    edit "spoke2"
        set interface "wan1"
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set add-route disable
        set dpd on-idle
        set auto-discovery-receiver enable
        set remote-gw 22.1.1.1
        set psksecret sample
        set dpd-retryinterval 5
    next
    edit "spoke2_backup"
        set interface "wan2"
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set add-route disable
        set dpd on-idle
        set auto-discovery-receiver enable
        set remote-gw 22.1.1.1
        set monitor "spoke2"
```

```
        set psksecret sample
        set dpd-retryinterval 5
    next
end
config vpn ipsec phase2-interface
    edit "spoke2"
        set phase1name "spoke2"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set auto-negotiate enable
    next
    edit "spoke2_backup"
        set phase1name "spoke2_backup"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set auto-negotiate enable
    next
end
```

c. Configure the spoke FortiGates' firewall policies:

i. Configure Spoke1:

```
config firewall policy
    edit 1
        set name "outbound_advpn"
        set srcintf "internal"
        set dstintf "spoke1" "spoke1_backup"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set name "inbound_advpn"
        set srcintf "spoke1" "spoke1_backup"
        set dstintf "internal"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end
```

ii. Configure Spoke2:

```
config firewall policy
    edit 1
        set name "outbound_advpn"
        set srcintf "internal"
        set dstintf "spoke2" "spoke2_backup"
```

```
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set name "inbound_advpn"
        set srcintf "spoke2" "spoke2_backup"
        set dstintf "internal"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end
```

d. Configure the spoke FortiGates' tunnel interface IP addresses:

i. Configure Spoke1:

```
config system interface
    edit "spoke1"
        set ip 10.10.10.1 255.255.255.255
        set remote-ip 10.10.10.254 255.255.255.0
    next
    edit "spoke1_backup"
        set ip 10.10.10.2 255.255.255.255
        set remote-ip 10.10.10.254 255.255.255.0
    next
end
```

ii. Configure Spoke2:

```
config system interface
    edit "spoke2"
        set ip 10.10.10.3 255.255.255.255
        set remote-ip 10.10.10.254 255.255.255.0
    next
    edit "spoke2_backup"
        set ip 10.10.10.4 255.255.255.255
        set remote-ip 10.10.10.254 255.255.255.0
    next
end
```

e. Configure the spoke FortiGates' BGP:

i. Configure Spoke1:

```
config router bgp
    set as 65412
    config neighbor
        edit "10.10.10.254"
```

```
            set advertisement-interval 1
            set link-down-failover enable
            set remote-as 65412
        next
    end
    config network
        edit 1
            set prefix 10.1.100.0 255.255.255.0
        next
    end
end
```

ii. Configure Spoke2:

```
config router bgp
    set as 65412
    config neighbor
        edit "10.10.10.254"
            set advertisement-interval 1
            set link-down-failover enable
            set remote-as 65412
        next
    end
    config network
        edit 1
            set prefix 192.168.4.0 255.255.255.0
        next
    end
end
```

4. Run `diagnose` and `get` commands on Spoke1 to check VPN and BGP states:

a. Run the `diagnose vpn tunnel list` command on Spoke1. The system should return the following:

```
list all ipsec tunnel in vd 0
----
name=spoke1 ver=1 serial=2 15.1.1.2:0->22.1.1.1:0 tun_id=22.1.1.1
bound_if=7 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/536 options[0218]=npu create_
dev frag-rfc  accept_traffic=1

proxyid_num=1 child_num=1 refcnt=19 ilast=1 olast=1 ad=r/2
stat: rxp=1 txp=160 rxb=16428 txb=8969
dpd: mode=on-idle on=1 idle=5000ms retry=3 count=0 seqno=628
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=spoke1 proto=0 sa=1 ref=6 serial=1 auto-negotiate adr
  src: 0:0.0.0.0/0.0.0.0:0
  dst: 0:0.0.0.0/0.0.0.0:0
  SA:  ref=6 options=1a227 type=00 soft=0 mtu=1438 expire=1225/0B replaywin=1024
       seqno=a1 esn=0 replaywin_lastseq=00000002 itn=0
  life: type=01 bytes=0/0 timeout=2369/2400
  dec: spi=c53a8f5b esp=aes key=16 cbe88682ad896a69290027b6dd8f7162
       ah=sha1 key=20 7bb704b388f83783ac76c2ab0b6c9f7dcf78e93b
  enc: spi=6e3633fc esp=aes key=16 1a0da3f4deed3d16becc9dda57537355
```

```
      ah=sha1 key=20 368544044bd9b82592d72476ff93d5055056da8d
  dec:pkts/bytes=1/16364, enc:pkts/bytes=160/19168
  npu_flag=03 npu_rgwy=22.1.1.1 npu_lgwy=15.1.1.2 npu_selid=1 dec_npuid=1 enc_npuid=1
----
name=spoke1_backup ver=1 serial=1 12.1.1.2:0->22.1.1.1:0 tun_id=22.1.1.1
bound_if=6 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/536 options[0218]=npu create_
dev frag-rfc  accept_traffic=0

proxyid_num=1 child_num=0 refcnt=11 ilast=0 olast=0 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=0 idle=5000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=spoke1_backup proto=0 sa=0 ref=2 serial=1 auto-negotiate adr
  src: 0:0.0.0.0/0.0.0.0:0
  dst: 0:0.0.0.0/0.0.0.0:0
```

b. Run the `get router info bgp summary` command on Spoke1. The system should return the following:

```
BGP router identifier 7.7.7.7, local AS number 65412
BGP table version is 2
1 BGP AS-PATH entries
0 BGP community entries

Neighbor             V         AS [[QualityAssurance62/MsgRcvd]]
[[QualityAssurance62/MsgSent]]   [[QualityAssurance62/TblVer]]  InQ OutQ Up/Down
State/PfxRcd
10.10.10.254       1.          65412     143     142         1.     1.     1. 00:24:45
               2

Total number of neighbors 1
```

c. Run the `get router info routing-table bgp` command on Spoke1. The system should return the following:

```
Routing table for VRF=0
B       172.16.101.0/24 [200/0] via 10.10.10.254, spoke1, 00:23:57
B       192.168.4.0/24 [200/0] via 10.10.10.254, spoke1, 00:22:03
```

d. Generate traffic between the spokes and check the shortcut tunnel and routing table. Run the `diagnose vpn tunnel list` command on Spoke1. The system should return the following:

```
list all ipsec tunnel in vd 0
----
name=spoke1 ver=1 serial=2 15.1.1.2:0->22.1.1.1:0 tun_id=22.1.1.1
bound_if=7 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/536 options[0218]=npu create_
dev frag-rfc  accept_traffic=1

proxyid_num=1 child_num=1 refcnt=19 ilast=2 olast=2 ad=r/2
stat: rxp=1 txp=268 rxb=16428 txb=31243
dpd: mode=on-idle on=1 idle=5000ms retry=3 count=0 seqno=714
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=spoke1 proto=0 sa=1 ref=6 serial=1 auto-negotiate adr
  src: 0:0.0.0.0/0.0.0.0:0
```

```
  dst: 0:0.0.0.0/0.0.0.0:0
  SA:  ref=6 options=1a227 type=00 soft=0 mtu=1438 expire=345/0B replaywin=1024
       seqno=10d esn=0 replaywin_lastseq=00000002 itn=0
  life: type=01 bytes=0/0 timeout=2369/2400
  dec: spi=c53a8f5b esp=aes key=16 cbe88682ad896a69290027b6dd8f7162
       ah=sha1 key=20 7bb704b388f83783ac76c2ab0b6c9f7dcf78e93b
  enc: spi=6e3633fc esp=aes key=16 1a0da3f4deed3d16becc9dda57537355
       ah=sha1 key=20 368544044bd9b82592d72476ff93d5055056da8d
  dec:pkts/bytes=1/16364, enc:pkts/bytes=268/48320
  npu_flag=03 npu_rgwy=22.1.1.1 npu_lgwy=15.1.1.2 npu_selid=1 dec_npuid=1 enc_npuid=1
----
name=spoke1_backup ver=1 serial=1 12.1.1.2:0->22.1.1.1:0 tun_id=22.1.1.1
bound_if=6 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/536 options[0218]=npu create_
dev frag-rfc  accept_traffic=0

proxyid_num=1 child_num=0 refcnt=11 ilast=8 olast=8 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=0 idle=5000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=spoke1_backup proto=0 sa=0 ref=2 serial=1 auto-negotiate adr
  src: 0:0.0.0.0/0.0.0.0:0
  dst: 0:0.0.0.0/0.0.0.0:0
----
name=spoke1_0 ver=1 serial=9 15.1.1.2:4500->13.1.1.2:4500 tun_id=13.1.1.2
bound_if=7 lgwy=static/1 tun=intf/0 mode=dial_inst/3 encap=none/728 options[02d8]=npu
create_dev no-sysctl rgwy-chg frag-rfc  accept_traffic=1

 parent=spoke1 index=0
proxyid_num=1 child_num=0 refcnt=17 ilast=4 olast=4 ad=r/2
stat: rxp=1 txp=100 rxb=112 txb=4686
dpd: mode=on-idle on=1 idle=5000ms retry=3 count=0 seqno=231
natt: mode=keepalive draft=32 interval=10 remote_port=4500
proxyid=spoke1 proto=0 sa=1 ref=5 serial=1 auto-negotiate adr
  src: 0:0.0.0.0/0.0.0.0:0
  dst: 0:0.0.0.0/0.0.0.0:0
  SA:  ref=6 options=1a227 type=00 soft=0 mtu=1422 expire=447/0B replaywin=1024
       seqno=65 esn=0 replaywin_lastseq=00000002 itn=0
  life: type=01 bytes=0/0 timeout=2368/2400
  dec: spi=c53a8f5c esp=aes key=16 73fd9869547475db78851e6c057ad9b7
       ah=sha1 key=20 6ad3a5b1028f6b33c82ba494a370f13c7f462635
  enc: spi=79cb0f2b esp=aes key=16 52ab0acdc830d58c00e5956a6484654a
       ah=sha1 key=20 baa82aba4106dc60618f6fe95570728656799239
  dec:pkts/bytes=1/46, enc:pkts/bytes=100/11568
  npu_flag=03 npu_rgwy=13.1.1.2 npu_lgwy=15.1.1.2 npu_selid=5 dec_npuid=1 enc_npuid=1
```

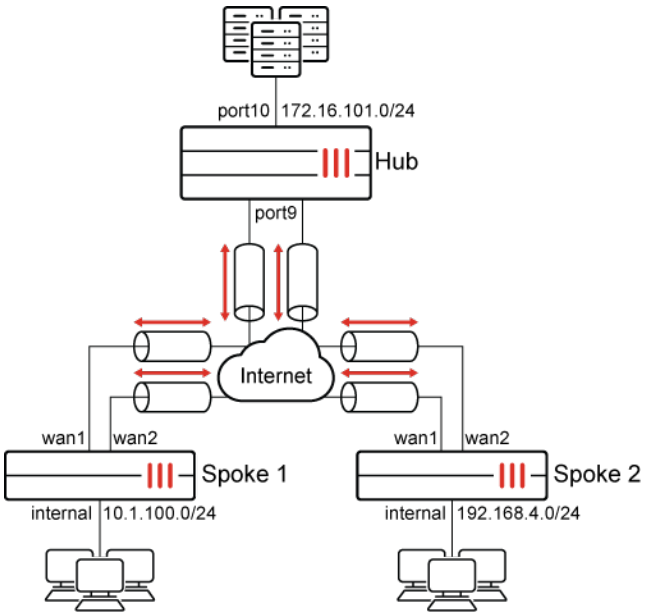
e. Run the `get router info routing-tale bgp` command. The system should return the following:

```
Routing table for VRF=0
B       172.16.101.0/24 [200/0] via 10.10.10.254, spoke1, 00:23:57
B       192.168.4.0/24 [200/0] via 10.10.10.3, spoke1_0 , 00:22:03
```

ADVPN with OSPF as the routing protocol

This is a sample configuration of ADVPN with OSPF as the routing protocol. The following options must be enabled for this configuration:

- On the hub FortiGate, IPsec phase1-interface `net-device enable` must be run.
- OSPF must be used between the hub and spoke FortiGates.



To configure ADVPN with OSPF as the routing protocol using the CLI:

1. Configure hub FortiGate's WAN, internal interface, and static route:

```
config system interface
    edit "port9"
        set alias "WAN"
        set ip 22.1.1.1 255.255.255.0
    next
    edit "port10"
        set alias "Internal"
        set ip 172.16.101.1 255.255.255.0
    next
end
config router static
    edit 1
        set gateway 22.1.1.2
        set device "port9"
    next
end
```

2. Configure the hub FortiGate:

a. Configure the hub FortiGate IPsec phase1-interface and phase2-interface:

```
config vpn ipsec phase1-interface
    edit "advpn-hub"
        set type dynamic
        set interface "port9"
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 3des-sha256 aes128-sha1 aes256-sha1 3des-
sha1
        set add-route disable
        set dpd on-idle
        set auto-discovery-sender enable
        set psksecret sample
        set dpd-retryinterval 5
    next
end
config vpn ipsec phase2-interface
    edit "advpn-hub"
        set phase1name "advpn-hub"
        set proposal aes128-sha1 aes256-sha1 3des-sha1 aes128-sha256 aes256-sha256 3des-
sha256
    next
end
```

When `net-device` is disabled, a tunnel ID is generated for each dynamic tunnel. This ID, in the form of an IP address, is used as the gateway in the route entry to that tunnel. The `tunnel-search` option is removed in FortiOS 7.0.0 and later.

b. Configure the hub FortiGate firewall policy:

```
config firewall policy
    edit 1
        set name "spoke2hub"
        set srcintf "advpn-hub"
        set dstintf "port10"
        set srcaddr "all"
        set dstaddr "172.16.101.0"
        set action accept
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set name "spoke2spoke"
        set srcintf "advpn-hub"
        set dstintf "advpn-hub"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
```

```
   next
end
```

c. Configure the hub FortiGate's IPsec tunnel interface IP address:

```
config system interface
    edit "advpn-hub1"
        set ip 10.10.10.254 255.255.255.255
        set remote-ip 10.10.10.253 255.255.255.0
    next
end
```

d. Configure the hub FortiGate's OSPF:

```
config router ospf
    set router-id 1.1.1.1
    config area
        edit 0.0.0.0
        next
    end
    config network
        edit 1
            set prefix 10.10.10.0 255.255.255.0
        next
        edit 2
            set prefix 172.16.101.0 255.255.255.0
        next
    end
end
```

3. Configure the spoke FortiGates:

a. Configure the spoke FortiGates' WAN, internal interfaces, and static routes:

i. Configure Spoke1:

```
config system interface
    edit "wan1"
        set alias "primary_WAN"
        set ip 15.1.1.2 255.255.255.0
    next
    edit "wan2"
        set alias "secondary_WAN"
        set ip 12.1.1.2 255.255.255.0
    next
    edit "internal"
        set ip 10.1.100.1 255.255.255.0
    next
end
config router static
    edit 1
        set gateway 12.1.1.1
        set device "wan2"
        set distance 15
```

```
    next
    edit 2
        set gateway 15.1.1.1
        set device "wan1"
    next
end
```

ii. Configure the Spoke2:

```
config system interface
    edit "wan1"
        set alias "primary_WAN"
        set ip 13.1.1.2 255.255.255.0
    next
    edit "wan2"
        set alias "secondary_WAN"
        set ip 17.1.1.2 255.255.255.0
    next
    edit "internal"
        set ip 192.168.4.1 255.255.255.0
    next
end
config router static
    edit 1
        set gateway 17.1.1.1
        set device "wan2"
        set distance 15
    next
    edit 2
        set gateway 13.1.1.1
        set device "wan1"
    next
end
```

b. Configure the spoke FortiGates' IPsec phase1-interface and phase2-interface:

i. Configure Spoke1:

```
config vpn ipsec phase1-interface
    edit "spoke1"
        set interface "wan1"
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set add-route disable
        set dpd on-idle
        set auto-discovery-receiver enable
        set remote-gw 22.1.1.1
        set psksecret sample
        set dpd-retryinterval 5
    next
    edit "spoke1_backup"
        set interface "wan2"
```

```
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set add-route disable
        set dpd on-idle
        set auto-discovery-receiver enable
        set remote-gw 22.1.1.1
        set monitor "spoke1"
        set psksecret sample
        set dpd-retryinterval 5
    next
end
config vpn ipsec phase2-interface
    edit "spoke1"
        set phase1name "spoke1"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set auto-negotiate enable
    next
    edit "spoke1_backup"
        set phase1name "spoke1_backup"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set auto-negotiate enable
    next
end
```

ii. Configure Spoke2:

```
config vpn ipsec phase1-interface
    edit "spoke2"
        set interface "wan1"
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set add-route disable
        set dpd on-idle
        set auto-discovery-receiver enable
        set remote-gw 22.1.1.1
        set psksecret sample
        set dpd-retryinterval 5
    next
    edit "spoke2_backup"
        set interface "wan2"
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set add-route disable
        set dpd on-idle
        set auto-discovery-receiver enable
        set remote-gw 22.1.1.1
        set monitor "spoke2"
```

```
        set psksecret sample
        set dpd-retryinterval 5
    next
end
config vpn ipsec phase2-interface
    edit "spoke2"
        set phase1name "spoke2"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set auto-negotiate enable
    next
    edit "spoke2_backup"
        set phase1name "spoke2_backup"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set auto-negotiate enable
    next
end
```

c. Configure the spoke FortiGates' firewall policies:

i. Configure Spoke1:

```
config firewall policy
    edit 1
        set name "outbound_advpn"
        set srcintf "internal"
        set dstintf "spoke1" "spoke1_backup"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set name "inbound_advpn"
        set srcintf "spoke1" "spoke1_backup"
        set dstintf "internal"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end
```

ii. Configure Spoke2:

```
config firewall policy
    edit 1
        set name "outbound_advpn"
        set srcintf "internal"
        set dstintf "spoke2" "spoke2_backup"
```

```
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set name "inbound_advpn"
        set srcintf "spoke2" "spoke2_backup"
        set dstintf "internal"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end
```

d. Configure the spoke FortiGates' tunnel interface IP addresses:

i. Configure Spoke1:

```
config system interface
    edit "spoke1"
        set ip 10.10.10.1 255.255.255.255
        set remote-ip 10.10.10.254 255.255.255.0
    next
    edit "spoke1_backup"
        set ip 10.10.10.2 255.255.255.255
        set remote-ip 10.10.10.254 255.255.255.0
    next
end
```

ii. Configure Spoke2:

```
config system interface
    edit "spoke2"
        set ip 10.10.10.3 255.255.255.255
        set remote-ip 10.10.10.254 255.255.255.0
    next
    edit "spoke2_backup"
        set ip 10.10.10.4 255.255.255.255
        set remote-ip 10.10.10.254 255.255.255.0
    next
end
```

e. Configure the spoke FortiGates' OSPF:

i. Configure Spoke1:

```
config router ospf
    set router-id 7.7.7.7
    config area
        edit 0.0.0.0
```

```
        next
    end
    config network
        edit 1
            set prefix 10.10.10.0 255.255.255.0
        next
        edit 2
            set prefix 10.1.100.0 255.255.255.0
        next
    end
end
```

ii. Configure Spoke2:

```
config router ospf
    set router-id 8.8.8.8
    config area
        edit 0.0.0.0
        next
    end
    config network
        edit 1
            set prefix 10.10.10.0 255.255.255.0
        next
        edit 2
            set prefix 192.168.4.0 255.255.255.0
        next
    end
end
```

4. Run `diagnose` and `get` commands on Spoke1 to check VPN and OSPF states:

a. Run the `diagnose vpn tunnel list` command on Spoke1. The system should return the following:

```
list all ipsec tunnel in vd 0
----
name=spoke1 ver=1 serial=2 15.1.1.2:0->22.1.1.1:0 tun_id=22.1.1.1
bound_if=7 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/536 options[0218]=npu create_
dev frag-rfc  accept_traffic=1

proxyid_num=1 child_num=1 refcnt=19 ilast=5 olast=2 ad=r/2
stat: rxp=1 txp=263 rxb=16452 txb=32854
dpd: mode=on-idle on=1 idle=5000ms retry=3 count=0 seqno=2283
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=spoke1 proto=0 sa=1 ref=5 serial=1 auto-negotiate adr
  src: 0:0.0.0.0/0.0.0.0:0
  dst: 0:0.0.0.0/0.0.0.0:0
  SA:  ref=6 options=1a227 type=00 soft=0 mtu=1438 expire=1057/0B replaywin=1024
       seqno=108 esn=0 replaywin_lastseq=00000003 itn=0
  life: type=01 bytes=0/0 timeout=2371/2400
  dec: spi=c53a8f78 esp=aes key=16 7cc50c5c9df1751f6497a4ad764c5e9a
       ah=sha1 key=20 269292ddbf7309a6fc05871e63ed8a5297b5c9a1
  enc: spi=6e363612 esp=aes key=16 42bd49bced1e85cf74a24d97f10eb601
```

```
      ah=sha1 key=20 13964f166aad48790c2e551d6df165d7489f524b
  dec:pkts/bytes=1/16394, enc:pkts/bytes=263/50096
  npu_flag=03 npu_rgwy=22.1.1.1 npu_lgwy=15.1.1.2 npu_selid=1 dec_npuid=1 enc_npuid=1
----
name=spoke1_backup ver=1 serial=1 12.1.1.2:0->22.1.1.1:0 tun_id=22.1.1.1
bound_if=6 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/536 options[0218]=npu create_
dev frag-rfc  accept_traffic=0

proxyid_num=1 child_num=0 refcnt=11 ilast=8 olast=8 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=0 idle=5000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=spoke1_backup proto=0 sa=0 ref=2 serial=1 auto-negotiate adr
  src: 0:0.0.0.0/0.0.0.0:0
  dst: 0:0.0.0.0/0.0.0.0:0
```

b. Run the `get router info ospf neighbor` command on Spoke1. The system should return the following:

```
OSPF process 0, VRF 0: Neighbor ID Pri State Dead Time Address Interface 8.8.8.8 1. Full/
- 00:00:35 10.10.10.254 spoke1 1.1.1.1 1. Full/ - 00:00:35 10.10.10.254 spoke1
```

c. Run the `get router info routing-table ospf` command on Spoke1. The system should return the following:

```
Routing table for VRF=0
O       172.16.101.0/24 [110/110] via 10.10.10.254, spoke1, 00:23:23
O       192.168.4.0/24   [110/110] via 10.10.10.254, spoke1, 00:22:35
```

d. Generate traffic between the spokes, then check the shortcut tunnel and routing table. Run the `diagnose vpn tunnel list` command on Spoke1. The system should return the following:

```
list all ipsec tunnel in vd 0
----
----
name=spoke1 ver=1 serial=2 15.1.1.2:0->22.1.1.1:0 tun_id=22.1.1.1
bound_if=7 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/536 options[0218]=npu create_
dev frag-rfc  accept_traffic=1

proxyid_num=1 child_num=1 refcnt=19 ilast=2 olast=2 ad=r/2
stat: rxp=1 txp=313 rxb=16452 txb=35912
dpd: mode=on-idle on=1 idle=5000ms retry=3 count=0 seqno=2303
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=spoke1 proto=0 sa=1 ref=3 serial=1 auto-negotiate adr
  src: 0:0.0.0.0/0.0.0.0:0
  dst: 0:0.0.0.0/0.0.0.0:0
  SA:  ref=6 options=1a227 type=00 soft=0 mtu=1438 expire=782/0B replaywin=1024
       seqno=13a esn=0 replaywin_lastseq=00000003 itn=0
  life: type=01 bytes=0/0 timeout=2371/2400
  dec: spi=c53a8f78 esp=aes key=16 7cc50c5c9df1751f6497a4ad764c5e9a
       ah=sha1 key=20 269292ddbf7309a6fc05871e63ed8a5297b5c9a1
  enc: spi=6e363612 esp=aes key=16 42bd49bced1e85cf74a24d97f10eb601
       ah=sha1 key=20 13964f166aad48790c2e551d6df165d7489f524b
```

```
  dec:pkts/bytes=1/16394, enc:pkts/bytes=313/56432
  npu_flag=03 npu_rgwy=22.1.1.1 npu_lgwy=15.1.1.2 npu_selid=1 dec_npuid=1 enc_npuid=1
----
name=spoke1_backup ver=1 serial=1 12.1.1.2:0->22.1.1.1:0 tun_id=22.1.1.1
bound_if=6 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/536 options[0218]=npu create_
dev frag-rfc  accept_traffic=0

proxyid_num=1 child_num=0 refcnt=11 ilast=13 olast=13 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=0 idle=5000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=spoke1_backup proto=0 sa=0 ref=2 serial=1 auto-negotiate adr
  src: 0:0.0.0.0/0.0.0.0:0
  dst: 0:0.0.0.0/0.0.0.0:0
----
name=spoke1_0 ver=1 serial=e 15.1.1.2:4500->13.1.1.2:4500 tun_id=13.1.1.2
bound_if=7 lgwy=static/1 tun=intf/0 mode=dial_inst/3 encap=none/728 options[02d8]=npu
create_dev no-sysctl rgwy-chg frag-rfc  accept_traffic=1

 parent=spoke1 index=0
proxyid_num=1 child_num=0 refcnt=19 ilast=4 olast=2 ad=r/2
stat: rxp=641 txp=1254 rxb=278648 txb=161536
dpd: mode=on-idle on=1 idle=5000ms retry=3 count=0 seqno=184
natt: mode=keepalive draft=32 interval=10 remote_port=4500
proxyid=spoke1_backup proto=0 sa=1 ref=10 serial=1 auto-negotiate adr
  src: 0:0.0.0.0/0.0.0.0:0
  dst: 0:0.0.0.0/0.0.0.0:0
  SA:  ref=6 options=1a227 type=00 soft=0 mtu=1422 expire=922/0B replaywin=1024
       seqno=452 esn=0 replaywin_lastseq=00000280 itn=0
  life: type=01 bytes=0/0 timeout=2370/2400
  dec: spi=c53a8f79 esp=aes key=16 324f8cf840ba6722cc7abbba46b34e0e
       ah=sha1 key=20 a40e9aac596b95c4cd83a7f6372916a5ef5aa505
  enc: spi=ef3327b5 esp=aes key=16 5909d6066b303de4520d2b5ae2db1b61
       ah=sha1 key=20 1a42f5625b5a335d8d5282fe83b5d6c6ff26b2a4
  dec:pkts/bytes=641/278568, enc:pkts/bytes=1254/178586
  npu_flag=03 npu_rgwy=13.1.1.2 npu_lgwy=15.1.1.2 npu_selid=a dec_npuid=1 enc_npuid=1
```

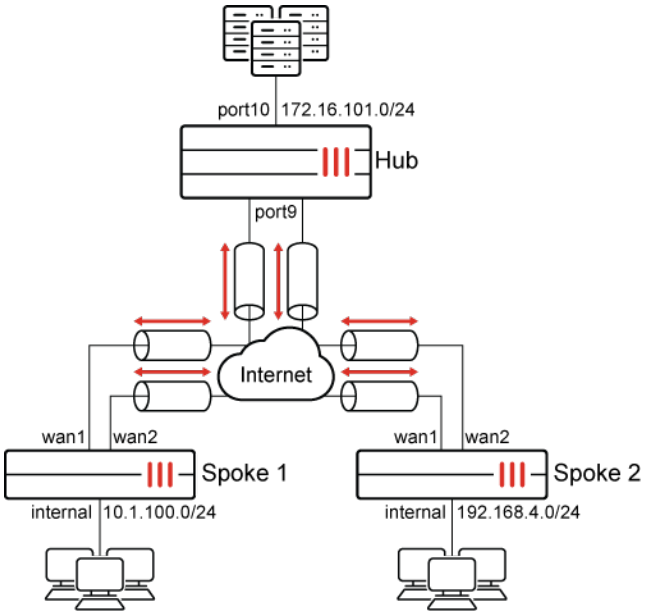
e. Run the `get router info routing-tale ospf` command. The system should return the following:

```
Routing table for VRF=0
O       172.16.101.0/24 [110/110] via 10.10.10.254, spoke1, 00:27:14
O       192.168.4.0/24  [110/110] via  10.10.10.3, spoke1_0, 00:26:26
```

ADVPN with RIP as the routing protocol

This is a sample configuration of ADVPN with RIP as routing protocol. The following options must be enabled for this configuration:

- On the hub FortiGate, IPsec phase1-interface `net-device disable` must be run.
- RIP must be used between the hub and spoke FortiGates.
- `split-horizon-status enable` must be run on the hub FortiGate.



To configure ADVPN with RIP as the routing protocol using the CLI:

1. In the CLI, configure hub FortiGate's WAN, internal interface, and static route:

```
config system interface
    edit "port9"
        set alias "WAN"
        set ip 22.1.1.1 255.255.255.0
    next
    edit "port10"
        set alias "Internal"
        set ip 172.16.101.1 255.255.255.0
    next
end
config router static
    edit 1
        set gateway 22.1.1.2
        set device "port9"
    next
end
```

2. Configure the hub FortiGate:
   a. Configure the hub FortiGate IPsec phase1-interface and phase2-interface:

```
config vpn ipsec phase1-interface
    edit "advpn-hub"
        set type dynamic
        set interface "port9"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 3des-sha256 aes128-sha1 aes256-sha1 3des-
sha1
```

```
        set add-route disable
        set dpd on-idle
        set auto-discovery-sender enable
        set psksecret sample
        set dpd-retryinterval 5
    next
end
config vpn ipsec phase2-interface
    edit "advpn-hub"
        set phase1name "advpn-hub"
        set proposal aes128-sha1 aes256-sha1 3des-sha1 aes128-sha256 aes256-sha256 3des-
sha256
    next
end
```

When `net-device` is disabled, a tunnel ID is generated for each dynamic tunnel. This ID, in the form of an IP address, is used as the gateway in the route entry to that tunnel. The `tunnel-search` option is removed in FortiOS 7.0.0 and later.

b. Configure the hub FortiGate firewall policy:

```
config firewall policy
    edit 1
        set name "spoke2hub"
        set srcintf "advpn-hub"
        set dstintf "port10"
        set srcaddr "all"
        set dstaddr "172.16.101.0"
        set action accept
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set name "spoke2spoke"
        set srcintf "advpn-hub"
        set dstintf "advpn-hub"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end
```

c. Configure the hub FortiGate's IPsec tunnel interface IP address:

```
config system interface
    edit "advpn-hub1"
        set ip 10.10.10.254 255.255.255.255
        set remote-ip 10.10.10.253 255.255.255.0
    next
end
```

d. Configure the hub FortiGate's RIP:

```
config router rip
    set default-information-originate enable
    config network
        edit 1
            set prefix 10.10.10.0 255.255.255.0
        next
        edit 2
            set prefix 172.16.101.0 255.255.255.0
        next
    end
    config interface
        edit "advpn-hub"
            set split-horizon-status disable
        next
    end
end
```

3. Configure the spoke FortiGates:

a. Configure the spoke FortiGates' WAN, internal interfaces, and static routes:

i. Configure Spoke1:

```
config system interface
    edit "wan1"
        set alias "primary_WAN"
        set ip 15.1.1.2 255.255.255.0
    next
    edit "wan2"
        set alias "secondary_WAN"
        set ip 12.1.1.2 255.255.255.0
    next
    edit "internal"
        set ip 10.1.100.1 255.255.255.0
    next
end
config router static
    edit 1
        set gateway 12.1.1.1
        set device "wan2"
        set distance 15
    next
    edit 2
        set gateway 15.1.1.1
        set device "wan1"
    next
end
```

ii. Configure the Spoke2:

```
config system interface
    edit "wan1"
```

```
        set alias "primary_WAN"
        set ip 13.1.1.2 255.255.255.0
    next
    edit "wan2"
        set alias "secondary_WAN"
        set ip 17.1.1.2 255.255.255.0
    next
    edit "internal"
        set ip 192.168.4.1 255.255.255.0
    next
end
config router static
    edit 1
        set gateway 17.1.1.1
        set device "wan2"
        set distance 15
    next
    edit 2
        set gateway 13.1.1.1
        set device "wan1"
    next
end
```

b. Configure the spoke FortiGates' IPsec phase1-interface and phase2-interface:

i. Configure Spoke1:

```
config vpn ipsec phase1-interface
    edit "spoke1"
        set interface "wan1"
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set add-route disable
        set dpd on-idle
        set auto-discovery-receiver enable
        set remote-gw 22.1.1.1
        set psksecret sample
        set dpd-retryinterval 5
    next
    edit "spoke1_backup"
        set interface "wan2"
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set add-route disable
        set dpd on-idle
        set auto-discovery-receiver enable
        set remote-gw 22.1.1.1
        set monitor "spoke1"
        set psksecret sample
        set dpd-retryinterval 5
    next
```

```
end
config vpn ipsec phase2-interface
    edit "spoke1"
        set phase1name "spoke1"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set auto-negotiate enable
    next
    edit "spoke1_backup"
        set phase1name "spoke1_backup"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set auto-negotiate enable
    next
end
```

ii. Configure Spoke2:

```
config vpn ipsec phase1-interface
    edit "spoke2"
        set interface "wan1"
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set add-route disable
        set dpd on-idle
        set auto-discovery-receiver enable
        set remote-gw 22.1.1.1
        set psksecret sample
        set dpd-retryinterval 5
    next
    edit "spoke2_backup"
        set interface "wan2"
        set peertype any
        set net-device enable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set add-route disable
        set dpd on-idle
        set auto-discovery-receiver enable
        set remote-gw 22.1.1.1
        set monitor "spoke2"
        set psksecret sample
        set dpd-retryinterval 5
    next
end
config vpn ipsec phase2-interface
    edit "spoke2"
        set phase1name "spoke2"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set auto-negotiate enable
    next
```

```
    edit "spoke2_backup"
        set phase1name "spoke2_backup"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm
aes256gcm chacha20poly1305
        set auto-negotiate enable
    next
end
```

c. Configure the spoke FortiGates' firewall policies:

i. Configure Spoke1:

```
config firewall policy
    edit 1
        set name "outbound_advpn"
        set srcintf "internal"
        set dstintf "spoke1" "spoke1_backup"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set name "inbound_advpn"
        set srcintf "spoke1" "spoke1_backup"
        set dstintf "internal"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end
```

ii. Configure Spoke2:

```
config firewall policy
    edit 1
        set name "outbound_advpn"
        set srcintf "internal"
        set dstintf "spoke2" "spoke2_backup"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
    edit 2
        set name "inbound_advpn"
        set srcintf "spoke2" "spoke2_backup"
        set dstintf "internal"
        set srcaddr "all"
```

```
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
    next
end
```

d. Configure the spoke FortiGates' tunnel interface IP addresses:

i. Configure Spoke1:

```
config system interface
    edit "spoke1"
        set ip 10.10.10.1 255.255.255.255
        set remote-ip 10.10.10.254 255.255.255.0
    next
    edit "spoke1_backup"
        set ip 10.10.10.2 255.255.255.255
        set remote-ip 10.10.10.254 255.255.255.0
    next
end
```

ii. Configure Spoke2:

```
config system interface
    edit "spoke2"
        set ip 10.10.10.3 255.255.255.255
        set remote-ip 10.10.10.254 255.255.255.0
    next
    edit "spoke2_backup"
        set ip 10.10.10.4 255.255.255.255
        set remote-ip 10.10.10.254 255.255.255.0
    next
end
```

e. Configure the spoke FortiGates' RIP:

i. Configure Spoke1:

```
config router rip
    config network
        edit 1
            set prefix 10.10.10.0 255.255.255.0
        next
        edit 2
            set prefix 10.1.100.0 255.255.255.0
        next
    end
end
```

ii. Configure Spoke2:

```
config router rip
    config network
```

```
        edit 1
            set prefix 10.10.10.0 255.255.255.0
        next
        edit 2
            set prefix 192.168.4.0 255.255.255.0
        next
    end
end
```

4. Run `diagnose` and `get` commands on Spoke1:
   a. Run the `diagnose vpn tunnel list` command on Spoke1. The system should return the following:

```
list all ipsec tunnel in vd 0
----
name=spoke1 ver=1 serial=2 15.1.1.2:0->22.1.1.1:0 tun_id=22.1.1.1
bound_if=7 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/536 options[0218]=npu create_
dev frag-rfc  accept_traffic=1

proxyid_num=1 child_num=1 refcnt=17 ilast=2 olast=2 ad=r/2
stat: rxp=1 txp=87 rxb=200 txb=6208
dpd: mode=on-idle on=1 idle=5000ms retry=3 count=0 seqno=1040
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=spoke1 proto=0 sa=1 ref=4 serial=1 auto-negotiate adr
  src: 0:0.0.0.0/0.0.0.0:0
  dst: 0:0.0.0.0/0.0.0.0:0
  SA:  ref=7 options=1a227 type=00 soft=0 mtu=1438 expire=1793/0B replaywin=1024
       seqno=57 esn=0 replaywin_lastseq=00000002 itn=0
  life: type=01 bytes=0/0 timeout=2370/2400
  dec: spi=c53a8f60 esp=aes key=16 6b54e32d54d039196a74d96e96d1cf14
       ah=sha1 key=20 e4903474614eafc96eda6400a3a5e88bbcb26a7f
  enc: spi=6e36349d esp=aes key=16 914a40a7993eda75c4dea2f42905f27d
       ah=sha1 key=20 8040eb08342edea2dae5eee058fd054a46688267
  dec:pkts/bytes=1/132, enc:pkts/bytes=86/11696
  npu_flag=03 npu_rgwy=22.1.1.1 npu_lgwy=15.1.1.2 npu_selid=1 dec_npuid=1 enc_npuid=1
----
name=spoke1_backup ver=1 serial=1 12.1.1.2:0->22.1.1.1:0 tun_id=22.1.1.1
bound_if=6 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/536 options[0218]=npu create_
dev frag-rfc  accept_traffic=0

proxyid_num=1 child_num=0 refcnt=11 ilast=0 olast=0 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=0 idle=5000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=spoke1_backup proto=0 sa=0 ref=2 serial=1 auto-negotiate adr
  src: 0:0.0.0.0/0.0.0.0:0
  dst: 0:0.0.0.0/0.0.0.0:0
```

   b. Run the `get router info rip database` command on Spoke1. The system should return the following:

```
Codes: R - RIP, Rc - RIP connected, Rs - RIP static, K - Kernel,
       C - Connected, S - Static, O - OSPF, I - IS-IS, B - BGP
```

```
   Network            Next Hop         Metric From            If     Time
Rc 10.1.100.0/24                             1.                 internal
Rc 10.10.10.2/32                               1.                 spoke1
R  172.16.101.0/24    10.10.10.254             1. 10.10.10.254      spoke1 02:28
R  192.168.4.0/24      10.10.10.254              1. 10.10.10.254       spoke1 02:44
```

c. Run the `get router info routing-table rip` command on Spoke1. The system should return the following:

```
Routing table for VRF=0
R       172.16.101.0/24 [120/2] via 10.10.10.254, spoke1, 00:08:38
R       192.168.4.0/24 [120/3] via 10.10.10.254, spoke1, 00:08:38
```

d. Generate traffic between the spokes, then check the shortcut tunnel and routing table. Run the `diagnose vpn tunnel list` command on Spoke1. The system should return the following:

```
list all ipsec tunnel in vd 0
----
name=spoke1 ver=1 serial=2 15.1.1.2:0->22.1.1.1:0 tun_id=22.1.1.1
bound_if=7 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/536 options[0218]=npu create_
dev frag-rfc  accept_traffic=1

proxyid_num=1 child_num=0 refcnt=19 ilast=3 olast=3 ad=r/2
stat: rxp=1 txp=78 rxb=200 txb=5546
dpd: mode=on-idle on=1 idle=5000ms retry=3 count=0 seqno=1039
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=spoke1 proto=0 sa=1 ref=5 serial=1 auto-negotiate adr
  src: 0:0.0.0.0/0.0.0.0:0
  dst: 0:0.0.0.0/0.0.0.0:0
  SA:  ref=7 options=1a227 type=00 soft=0 mtu=1438 expire=1807/0B replaywin=1024
       seqno=4e esn=0 replaywin_lastseq=00000002 itn=0
  life: type=01 bytes=0/0 timeout=2370/2400
  dec: spi=c53a8f60 esp=aes key=16 6b54e32d54d039196a74d96e96d1cf14
       ah=sha1 key=20 e4903474614eafc96eda6400a3a5e88bbcb26a7f
  enc: spi=6e36349d esp=aes key=16 914a40a7993eda75c4dea2f42905f27d
       ah=sha1 key=20 8040eb08342edea2dae5eee058fd054a46688267
  dec:pkts/bytes=1/132, enc:pkts/bytes=77/10456
  npu_flag=03 npu_rgwy=22.1.1.1 npu_lgwy=15.1.1.2 npu_selid=1 dec_npuid=1 enc_npuid=1
----
name=spoke1_backup ver=1 serial=1 12.1.1.2:0->22.1.1.1:0 tun_id=22.1.1.1
bound_if=6 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/536 options[0218]=npu create_
dev frag-rfc  accept_traffic=0

proxyid_num=1 child_num=0 refcnt=11 ilast=20 olast=20 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=0 idle=5000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=spoke1_backup proto=0 sa=0 ref=2 serial=1 auto-negotiate adr
  src: 0:0.0.0.0/0.0.0.0:0
  dst: 0:0.0.0.0/0.0.0.0:0
----
name=spoke1_0 ver=1 serial=a 15.1.1.2:4500->13.1.1.2:4500 tun_id=13.1.1.2
```

```
bound_if=7 lgwy=static/1 tun=intf/0 mode=dial_inst/3 encap=none/728 options[02d8]=npu
create_dev no-sysctl rgwy-chg frag-rfc  accept_traffic=1

 parent=spoke1 index=0
proxyid_num=1 child_num=0 refcnt=20 ilast=2 olast=0 ad=r/2
stat: rxp=1 txp=7 rxb=112 txb=480
dpd: mode=on-idle on=1 idle=5000ms retry=3 count=0 seqno=0
natt: mode=keepalive draft=32 interval=10 remote_port=4500
proxyid=spoke1 proto=0 sa=1 ref=8 serial=1 auto-negotiate adr
  src: 0:0.0.0.0/0.0.0.0:0
  dst: 0:0.0.0.0/0.0.0.0:0
  SA:  ref=6 options=1a227 type=00 soft=0 mtu=1422 expire=2358/0B replaywin=1024
       seqno=8 esn=0 replaywin_lastseq=00000002 itn=0
  life: type=01 bytes=0/0 timeout=2367/2400
  dec: spi=c53a8f61 esp=aes key=16 c66aa7ae9657068108ed47c048ff56b6
       ah=sha1 key=20 60661c68e20bbc913c2564ade85e01ea3769e703
  enc: spi=79cb0f30 esp=aes key=16 bf6c898c2e1c64baaa679ed5d79c3b58
       ah=sha1 key=20 146ca78be6c34eedb9cd66cc328216e08682ecb1
  dec:pkts/bytes=1/46, enc:pkts/bytes=7/992
  npu_flag=03 npu_rgwy=13.1.1.2 npu_lgwy=15.1.1.2 npu_selid=6 dec_npuid=1 enc_npuid=1
```

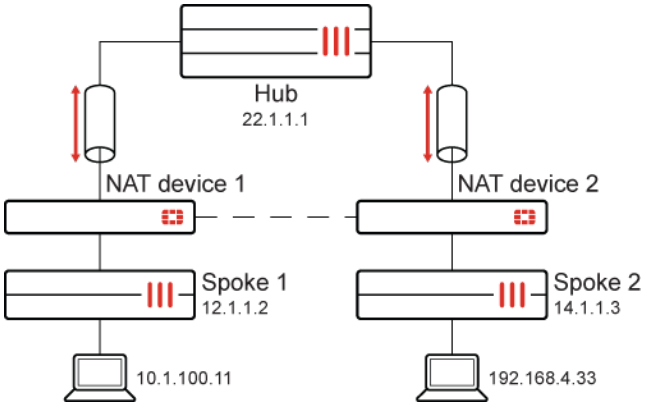
e. Run the `get router info routing-tale rip` command. The system should return the following:

```
Routing table for VRF=0
R       172.16.101.0/24 [120/2] via 10.10.10.254, spoke1, 00:09:04
R       192.168.4.0/24 [120/2] via 10.10.10.3, spoke1_0, 00:00:02
```

UDP hole punching for spokes behind NAT

UDP hole punching allows ADVPN shortcuts to be established through a UDP hole on a NAT device. The NAT device must support RFC 4787 Endpoint-Independent Mapping.

In the following example, device 10.1.100.11 behind Spoke1 needs to reach device 192.168.4.33 behind Spoke2. Spoke1 and Spoke2 are behind NAT devices and have established IPsec tunnels to the Hub. The hole punching creates a shortcut between Spoke1 and Spoke2 that bypasses the Hub.



To verify the ADVPN shortcut is established between both spokes behind NAT:

```
# diagnose debug enable
# diagnose debug application ike -1
ike 0: comes 22.1.1.1:4500->12.1.1.2:4500,ifindex=6....
ike 0: IKEv1 exchange=Informational id=3c10fb6a76f1e264/6c7b397100dffc63:58ac7c02 len=204
ike 0:toHub1:35: notify msg received: SHORTCUT-OFFER
ike 0:toHub1: shortcut-offer 10.1.100.11->192.168.4.33 psk 64 ppk 0 ver 1 mode 0
ike 0 looking up shortcut by addr 192.168.4.33, name toHub1
ike 0:toHub1: send shortcut-query 1438189781753480593 d3fdd1bfbc94caee/0000000000000000 12.1.1.2
10.1.100.11->192.168.4.33 psk 64 ttl 32 nat 1 ver 1 mode 0
ike 0:toHub1:35: sent IKE msg (SHORTCUT-QUERY): 12.1.1.2:4500->22.1.1.1:4500, len=236,
id=3c10fb6a76f1e264/6c7b397100dffc63:12e263f7
ike 0: comes 22.1.1.1:4500->12.1.1.2:4500,ifindex=6....
ike 0: IKEv1 exchange=Informational id=3c10fb6a76f1e264/6c7b397100dffc63:4976e1ac len=236
ike 0:toHub1:35: notify msg received: SHORTCUT-REPLY
ike 0:toHub1: recv shortcut-reply 1438189781753480593 d3fdd1bfbc94caee/16a1eb5b0f37ee23 14.1.1.3
to 10.1.100.11 psk 64 ppk 0 ver 1 mode 0 nat 55.1.1.2:64916
ike 0:toHub1: iif 22 192.168.4.33->10.1.100.11 route lookup oif 21
ike 0:toHub1: shortcut-reply received from 55.1.1.2:64916, local-nat=yes, peer-nat=yes
ike 0:toHub1: NAT hole punching to peer at 55.1.1.2:64916
ike 0:toHub1: created connection: 0x5e71f58 6 12.1.1.2->55.1.1.2:64916.     <==55.1.1.2:64916 this
is UDP hole of NAT device
ike 0:toHub1: adding new dynamic tunnel for 55.1.1.2:64916
ike 0:toHub1_0: added new dynamic tunnel for 55.1.1.2:64916
ike 0:toHub1_0:48: initiator: main mode is sending 1st message...
ike 0:toHub1_0:48: cookie d3fdd1bfbc94caee/16a1eb5b0f37ee23
ike 0:toHub1_0:48: sent IKE msg (ident_i1send): 12.1.1.2:4500->55.1.1.2:64916, len=632,
id=d3fdd1bfbc94caee/16a1eb5b0f37ee23
ike 0: comes 55.1.1.2:64916->12.1.1.2:4500,ifindex=6....
ike 0: IKEv1 exchange=Identity Protection id=d3fdd1bfbc94caee/16a1eb5b0f37ee23 len=252
ike 0:toHub1_0:48: initiator: main mode get 1st response...
…
…
ike 0:toHub1_0:48: negotiation result
ike 0:toHub1_0:48: proposal id = 1:
ike 0:toHub1_0:48:   protocol id = ISAKMP:
ike 0:toHub1_0:48:      trans_id = KEY_IKE.
ike 0:toHub1_0:48:      encapsulation = IKE/none
ike 0:toHub1_0:48:         type=OAKLEY_ENCRYPT_ALG, val=AES_CBC, key-len=128
ike 0:toHub1_0:48:         type=OAKLEY_HASH_ALG, val=SHA2_256.
ike 0:toHub1_0:48:         type=AUTH_METHOD, val=PRESHARED_KEY.
ike 0:toHub1_0:48:         type=OAKLEY_GROUP, val=MODP2048.
ike 0:toHub1_0:48: ISAKMP SA lifetime=86400
ike 0:toHub1_0:48: sent IKE msg (ident_i2send): 12.1.1.2:4500->55.1.1.2:64916, len=380,
id=d3fdd1bfbc94caee/16a1eb5b0f37ee23
ike 0: comes 55.1.1.2:64916->12.1.1.2:4500,ifindex=6....
ike 0: IKEv1 exchange=Identity Protection id=d3fdd1bfbc94caee/16a1eb5b0f37ee23 len=380
ike 0:toHub1_0:48: initiator: main mode get 2nd response...
…
…
ike 0:toHub1_0:48: add INITIAL-CONTACT
```

```
ike 0:toHub1_0:48: add INTERFACE-ADDR4 10.10.1.100
ike 0:toHub1_0:48: sent IKE msg (ident_i3send): 12.1.1.2:4500->55.1.1.2:64916, len=140,
id=d3fdd1bfbc94caee/16a1eb5b0f37ee23
ike 0: comes 55.1.1.2:64916->12.1.1.2:4500,ifindex=6....
ike 0: IKEv1 exchange=Identity Protection id=d3fdd1bfbc94caee/16a1eb5b0f37ee23 len=124
ike 0:toHub1_0:48: initiator: main mode get 3rd response...
ike 0:toHub1_0:48: received p1 notify type INTERFACE-ADDR4
ike 0:toHub1_0:48: INTERFACE-ADDR4 10.10.1.102
ike 0:toHub1_0:48: peer identifier IPV4_ADDR 14.1.1.3
ike 0:toHub1_0:48: PSK authentication succeeded
ike 0:toHub1_0:48: authentication OK
ike 0:toHub1_0:48: established IKE SA d3fdd1bfbc94caee/16a1eb5b0f37ee23
ike 0:toHub1_0:48: auto-discovery receiver
ike 0:toHub1_0:48: auto-discovery 2
ike 0:toHub1_0: add R/32 route 10.10.1.102 via 10.10.1.102, intf=toHub1(22)
ike 0:toHub1_0: add peer route 10.10.1.102
ike 0:toHub1: schedule auto-negotiate
ike 0:toHub1_0:48: no pending Quick-Mode negotiations
ike 0:toHub1_0:toHub1: IPsec SA connect 6 12.1.1.2->55.1.1.2:64916
ike 0:toHub1_0:toHub1: using existing connection
ike 0:toHub1_0:toHub1: traffic triggered, serial=1 1:10.1.100.11:2048->1:192.168.4.33:0
ike 0:toHub1:toHub1: config found
ike 0:toHub1_0:toHub1: IPsec SA connect 6 12.1.1.2->55.1.1.2:64916 negotiating
ike 0:toHub1_0:48: cookie d3fdd1bfbc94caee/16a1eb5b0f37ee23:8465e467
ike 0:toHub1_0:48:toHub1:109: natt flags 0x1f, encmode 1->3
ike 0:toHub1_0:48:toHub1:109: initiator selectors 0 0:0.0.0.0/0.0.0.0:0:0->0:0.0.0.0/0.0.0.0:0:0
ike 0:toHub1_0:48: sent IKE msg (quick_i1send): 12.1.1.2:4500->55.1.1.2:64916, len=620,
id=d3fdd1bfbc94caee/16a1eb5b0f37ee23:8465e467
ike 0: comes 55.1.1.2:64916->12.1.1.2:4500,ifindex=6....
ike 0: IKEv1 exchange=Quick id=d3fdd1bfbc94caee/16a1eb5b0f37ee23:8465e467 len=444
ike 0:toHub1_0:48:toHub1:109: responder selectors 0:0.0.0.0/0.0.0.0:0->0:0.0.0.0/0.0.0.0:0
ike 0:toHub1_0:48:toHub1:109: my proposal:
…
…
ike 0:toHub1_0:48:toHub1:109: add IPsec SA: SPIs=79654cf1/5e9936a5
ike 0:toHub1_0:48:toHub1:109: IPsec SA dec spi 79654cf1 key 16:5E21180992B8892DE5142E1F53ABD29E
auth 20:49AA4AE14994A39A138392AC517B6E79D98CA673
ike 0:toHub1_0:48:toHub1:109: IPsec SA enc spi 5e9936a5 key 16:BE16B8EF4E75F7B3CF97A1D58D996890
auth 20:2F46B57CAC6F3185BB182F9280312263325F6BAF
ike 0:toHub1_0:48:toHub1:109: added IPsec SA: SPIs=79654cf1/5e9936a5
ike 0:toHub1_0:48:toHub1:109: sending SNMP tunnel UP trapp
```

To verify the spoke-to-spoke IPsec phase 1 tunnel shortcut is established:

```
# diagnose vpn ike gateway list
vd: root/0
name: toHub1
version: 1
interface: wan2 6
addr: 12.1.1.2:4500 -> 22.1.1.1:4500
tun_id: 22.1.1.1
created: 503s ago
```

```
assigned IPv4 address: 10.10.1.100/255.255.255.0
nat: me
auto-discovery: 2 receiver
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/3  established 1/3  time 0/0/0 ms

  id/spi: 35 3c10fb6a76f1e264/6c7b397100dffc63
  direction: initiator
  status: established 503-503s ago = 0ms
  proposal: aes128-sha256
  key: 7fca86063ea2e72f-4efea6f1bec23948
  lifetime/rekey: 86400/85596
  DPD sent/recv: 00000000/00000000

vd: root/0
name: toHub1_0
version: 1
interface: wan2 6
addr: 12.1.1.2:4500 -> 55.1.1.2:64916
created: 208s ago
nat: me peer
auto-discovery: 2 receiver
IKE SA: created 1/1  established 1/1  time 20/20/20 ms
IPsec SA: created 1/1  established 1/1  time 10/10/10 ms

  id/spi: 48 d3fdd1bfbc94caee/16a1eb5b0f37ee23
  direction: initiator
  status: established 208-208s ago = 20ms
  proposal: aes128-sha256
  key: 9bcac400d8e14e11-fffde33eaa3a8263
  lifetime/rekey: 86400/85891
  DPD sent/recv: 0000000a/00000000
```

Fabric Overlay Orchestrator

The Fabric Overlay Orchestrator feature is an easy-to-use GUI wizard that simplifies the process of configuring a self-orchestrated SD-WAN overlay within a single Security Fabric. This feature is self-orchestrated since no additional tool or device, aside from the FortiGates themselves, is required to orchestrate this configuration. An SD-WAN overlay configuration consists of IPsec and BGP configuration settings.

Currently, the Fabric Overlay Orchestrator supports a single hub architecture and builds upon an existing Security Fabric configuration. This feature configures the root FortiGate as the SD-WAN overlay hub and the downstream first-level FortiGates as the spokes.

After configuring the Fabric Overlay, you can complete the SD-WAN deployment by configuring SD-WAN rules.

If you cannot view the *VPN > Fabric Overlay Orchestrator* tree menu, configure the FortiGate as a root or a downstream device in the Security Fabric. See Configuring the root FortiGate and downstream FortiGates on page 4302 for more details.

The Fabric Overlay Orchestrator does not work when VDOM mode is enabled.

This section contains the following topics:

- Prerequisites on page 3278
- Network topology on page 3278
- Using the Fabric Overlay Orchestrator on page 3279
- SPA easy configuration key for FortiSASE on page 3297
- Fabric Overlay Orchestrator Topology dashboard widget for hub FortiGates on page 3299

Prerequisites

Create a single Fortinet Security Fabric with the following components:

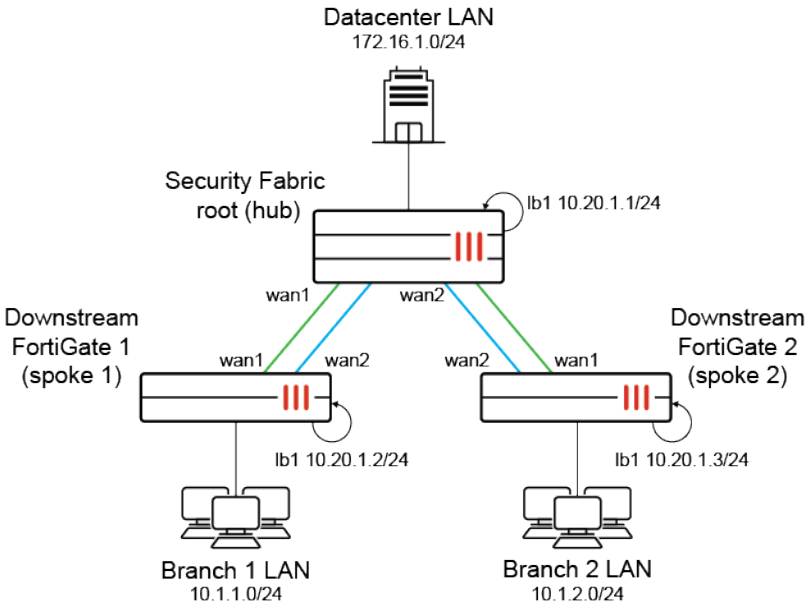
- A root FortiGate and one or more downstream FortiGates all running FortiOS 7.2.4 or later
- A FortiAnalyzer, or cloud logging using FortiAnalyzer Cloud or FortiGate Cloud
  - For FortiGate Cloud, all downstream devices must belong to the same FortiCloud account

  For more information about configuring these components, see Configuring the root FortiGate and downstream FortiGates on page 4302, Configuring FortiAnalyzer on page 4311, and Configuring cloud logging on page 4314 in the Security Fabric chapter.

The Fabric Overlay Orchestrator does not work when VDOM mode is enabled.

Network topology

The Fabric Overlay Orchestrator supports configuring an overlay for the following example hub and spoke topology using ADVPN and a single hub.



This topology corresponds to the single datacenter (active-passive gateway) design using the IPsec overlay design of one-to-one overlay mapping per underlay. For more details on these topics, see the SD-WAN Architectures for Enterprise guide.

In this topology, the datacenter FortiGate (Security Fabric root FortiGate) is the hub, and the branch FortiGates (Security Fabric downstream FortiGates) are the spokes. Each FortiGate has a distinctly defined LAN subnet and loopback interface (lb1) with an IP address within the 10.20.1.0/24 subnet.

The Fabric Overlay Orchestrator creates loopbacks to act as health check servers that are always up, and they can be accessed by adjacent Fabric devices. When configuring the policy creation option of either automatic or health check on the hub, the Fabric Overlay Orchestrator configures performance SLAs from the hub to the health check servers on 10.20.1.2 and 10.20.1.3 corresponding to the spoke 1 and spoke 2 FortiGates respectively. Likewise, when the Fabric Overlay Orchestrator runs on each spoke, it creates a performance SLA to the hub using its loopback address of 10.20.1.1.

Instead of using loopbacks, any business-critical applications and resources connected to the LAN of each device can be used as health check servers for performance SLAs.

Using the Fabric Overlay Orchestrator

If you cannot view the *VPN > Fabric Overlay Orchestrator* tree menu, configure the FortiGate as a root or a downstream device in the Security Fabric. See Configuring the root FortiGate and downstream FortiGates on page 4302 for more details.

The Fabric Overlay Orchestrator does not work when VDOM mode is enabled.

The following steps should be used to configure a self-orchestrated SD-WAN overlay within a single Security Fabric. These steps must be followed in order, and assume that the prerequisites and network topology are in place.

1. Configure the root FortiGate using the Fabric Overlay Orchestrator.
2. Configure one or more downstream FortiGates using the Fabric Overlay Orchestrator.
3. Configure an overlay on the spoke for an additional incoming interface on the hub (if applicable).
4. Verify the firewall policies on the hub FortiGate.
5. Verify the Fabric Overlay created by the Fabric Overlay Orchestrator:
   a. Verify the IPsec VPN tunnels on the hub FortiGate.
   b. Verify BGP routing on the hub FortiGate.
   c. Verify the performance SLAs on the hub FortiGate.
   d. Verify the firewall policies on a spoke FortiGate.
   e. Verify the IPsec VPN tunnels on a spoke FortiGate.
   f. Verify BGP routing on a spoke FortiGate.
   g. Verify the performance SLAs on a spoke FortiGate.
   h. Verify the spoke-to-spoke ADVPN communication.
6. Configure SD-WAN rules on the hub FortiGate.
7. Configure SD-WAN rules on the spoke FortiGates.

When configuring the root and downstream FortiGates, the Fabric Overlay Orchestrator configures the following settings in the background:

- IPsec overlay configuration (hub and spoke ADVPN tunnels)
- BGP configuration
- Policy routing
- SD-WAN zones
- SD-WAN performance SLAs

The FortiGate's role in the SD-WAN overlay is automatically determined by its role in the Security Fabric. The Fabric root will be the hub, and any first-level downstream devices from the Fabric root will be spokes.

After using the Fabric Overlay Orchestrator on all FortiGates and verifying the overlay settings, complete the SD-WAN deployment configuration using steps 3 (if applicable), and steps 6 and 7. See SD-WAN rules on page 1105 for more information.

Creating firewall policies

The Fabric Overlay Orchestrator can create firewall policies to allow all traffic through the SD-WAN overlay, or firewall policies to just allow health check traffic through it instead. When the Fabric Overlay Orchestrator is enabled on the root FortiGate, there are three *Policy creation* options:

- *Automatic*: automatically create policies for the loopback interface and tunnel overlays.
- *Health check*: automatically create a policy for the loopback interface so the SD-WAN health checks are functional.
- *Manual*: no policies are automatically created.

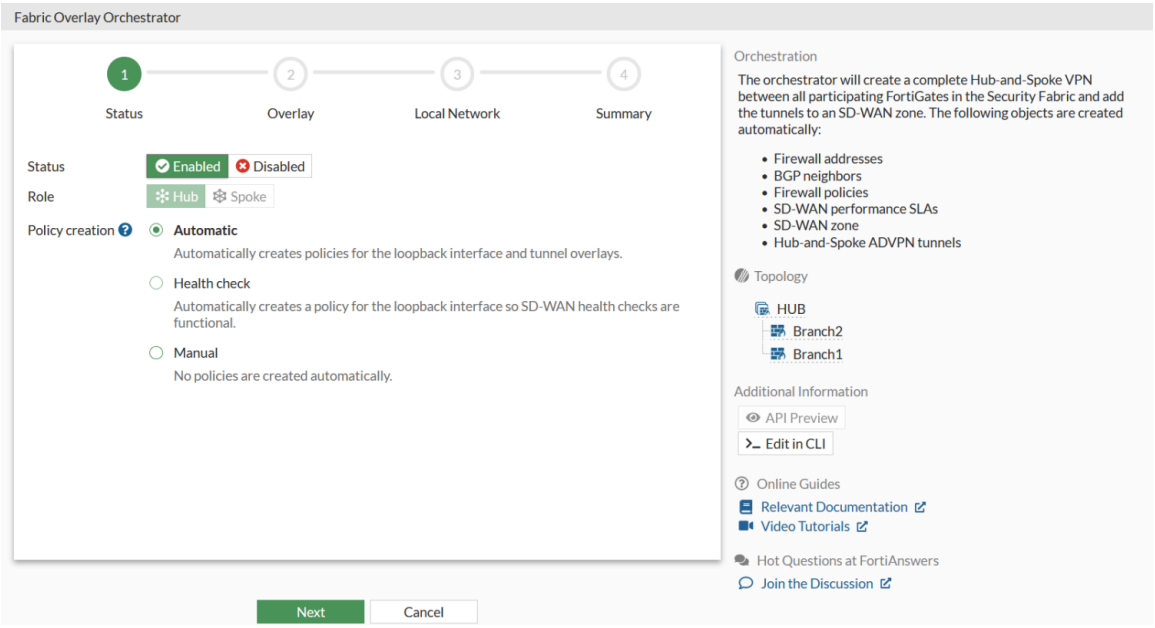
The *Automatic* policy creation option creates wildcard allow policies for the tunnel overlays. For some cases, these policies do not provide the necessary granularity to restrict overlay traffic to specific subnets or hosts.

When the Fabric Overlay Orchestrator is configured on a device, changing the policy creation rule will create new policies based on the rule, but it will not delete existing policies. Deleting existing policies must be performed manually.

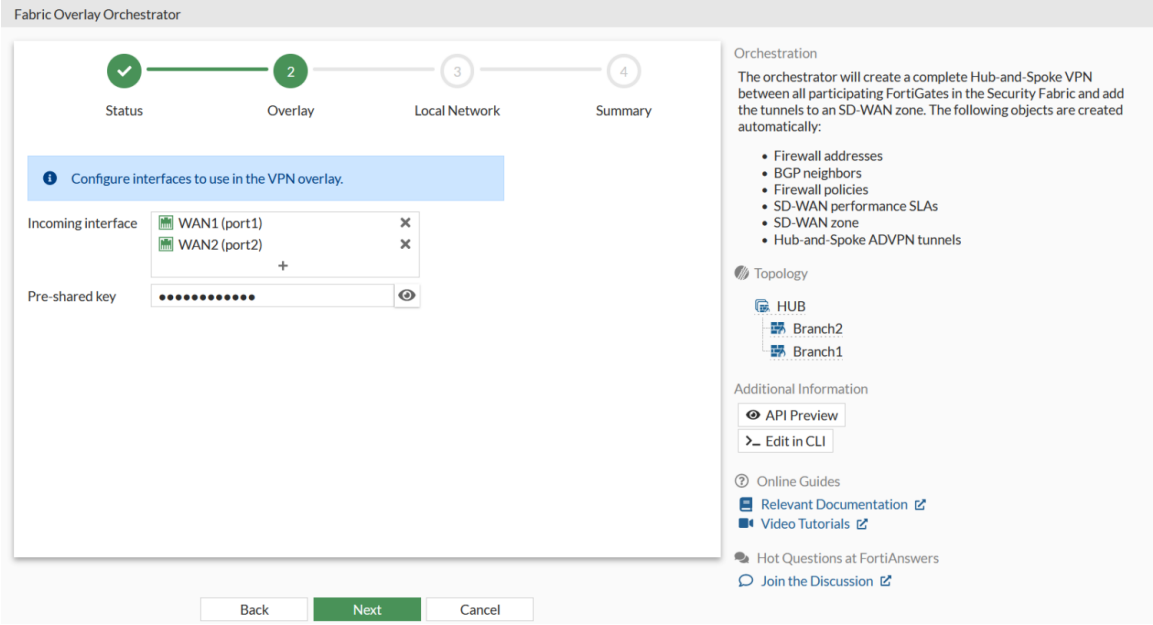
Configuring the root FortiGate using the Fabric Overlay Orchestrator

To configure the root FortiGate using the Fabric Overlay Orchestrator:

1. Go to *VPN* > *Fabric Overlay Orchestrator*.
2. Set the *Status* to *Enabled*. The *Role* is automatically selected based on the FortiGate's role in the Security Fabric. Ensure that *Hub* is selected. The Fabric root must always be the hub.
3. Set *Policy creation* to *Automatic*.

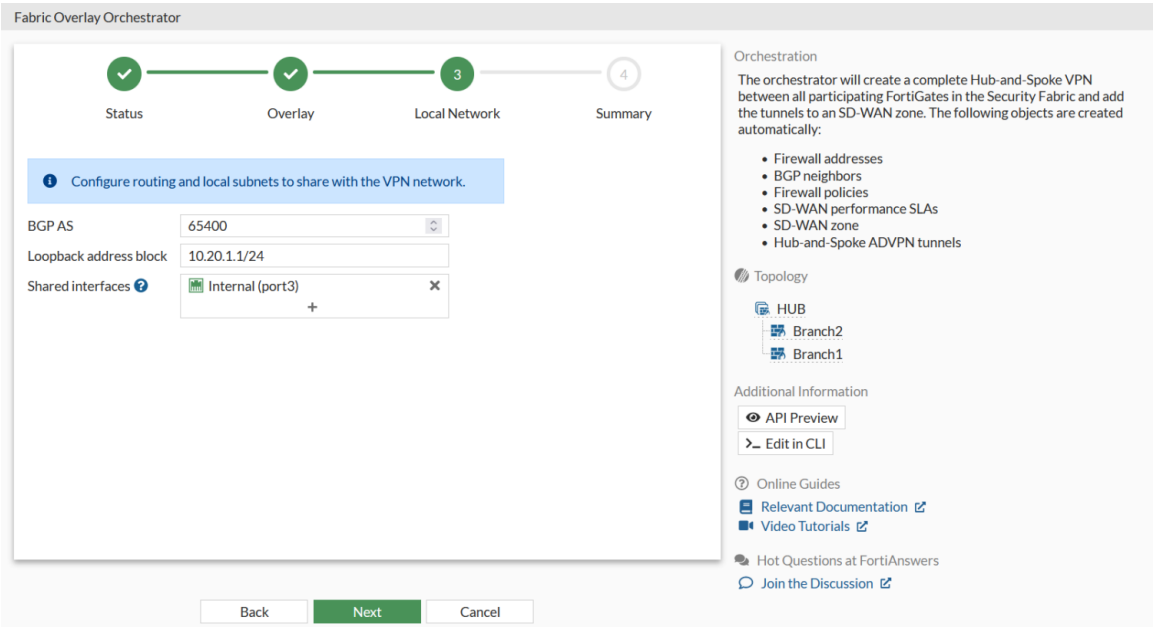


4. Click *Next*. The *Overlay* settings appear.
5. Select one or more interfaces as the *Incoming interface* or the underlay link over which the VPN overlay will be built (two incoming interfaces are selected in this example).
6. Enter the *Pre-shared key*.

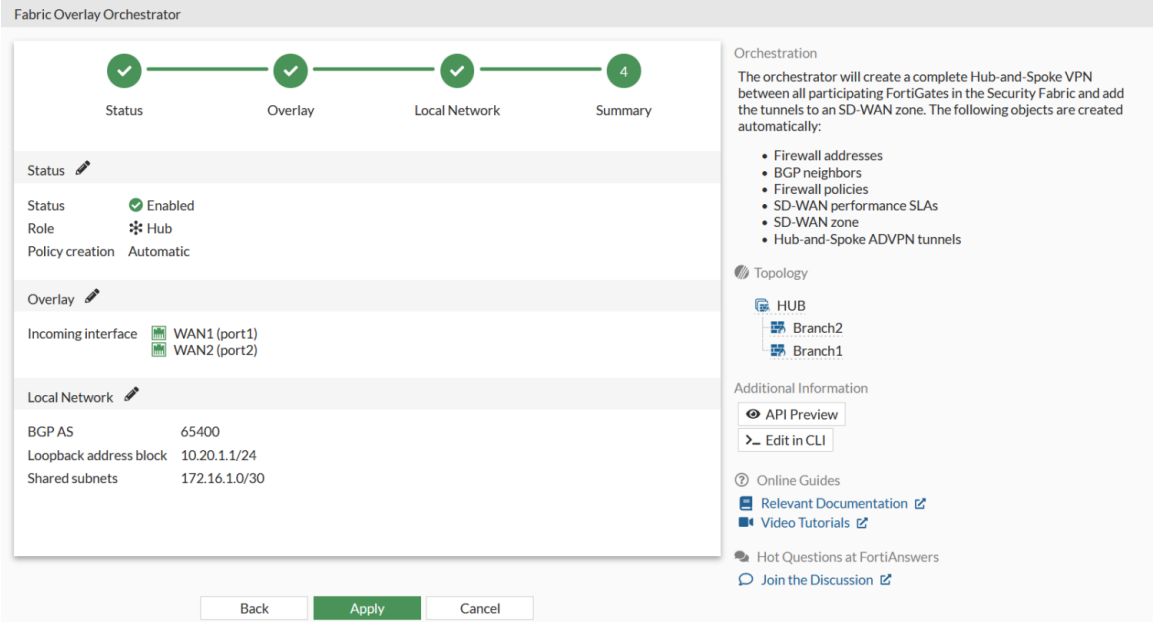


7. Click *Next*. The *Local Network* settings appear.
8. Configure routing and local subnets to share the following with the VPN network:

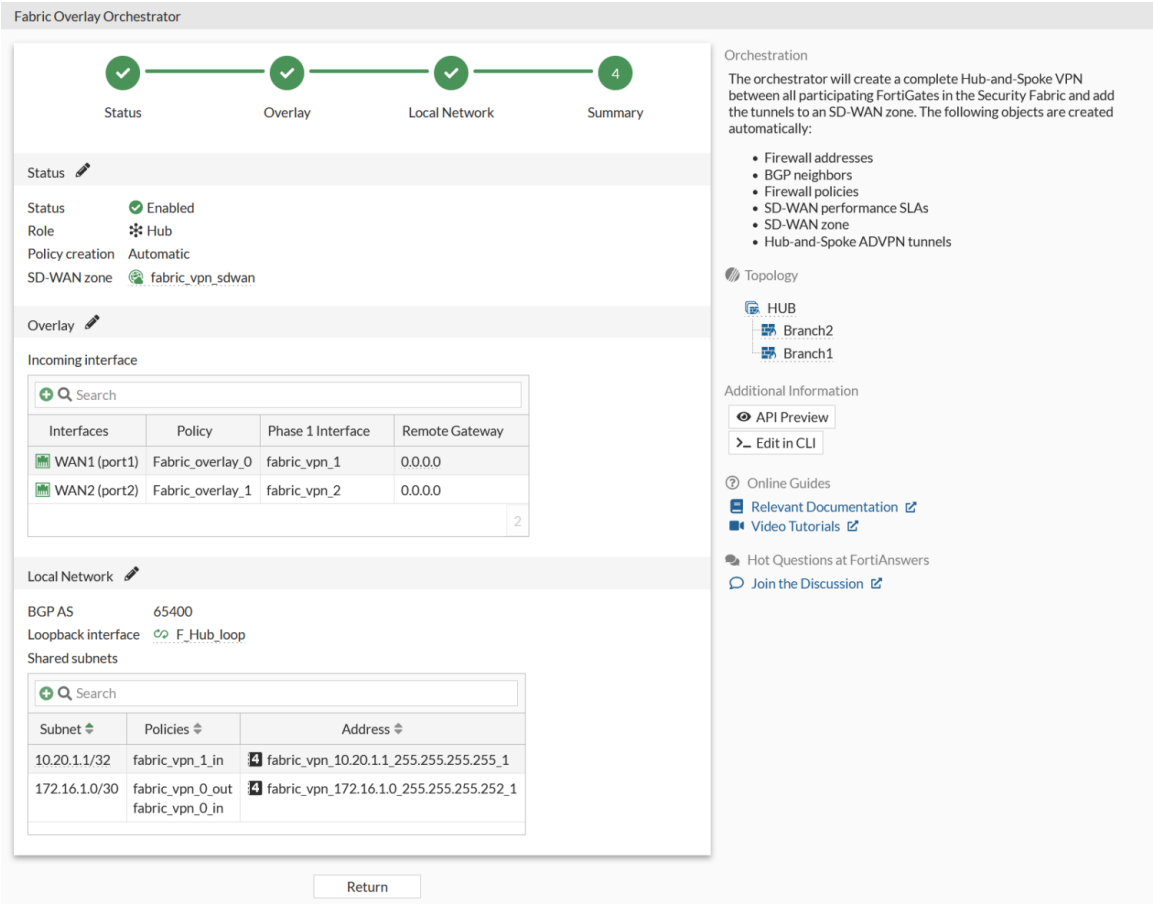
BGP AS	Optional setting to configure the BGP AS number. By default, this is set to *65400*.
Loopback address block	Optional setting to configure the loopback IP address. By default, this is set to *10.20.1.1/255.255.255.0*.
Shared interfaces	Select the interface of the local network to share with the VPN network.



9. Click *Next*. The *Summary* page appears.
10. Review the settings, then click *Apply*.



An updated *Summary* page appears with all the settings.



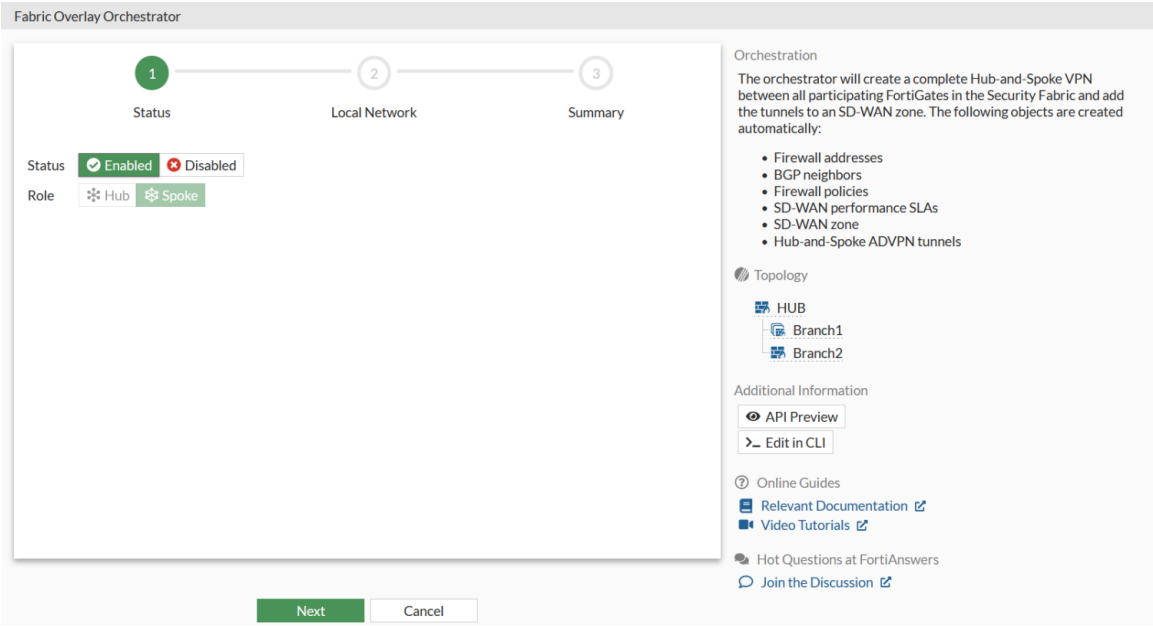
Note the following settings in this example:

SD-WAN zone	Located in the *Status* section: *fabric_vpn_sdwan*.
VPN tunnels	Located in the *Overlay* section in the *Incoming interface* table under the *Phase 1 Interface* column: *fabric_vpn1* and *fabric_vpn2*.
BGP	Located in the *Local Network* section. The *BGP AS* is *65400*. The *Shared subnets* are *10.20.1.1/32* and *172.16.1.0/30*.
Loopback interface	Located in the *Local Network* section: *F_Hub_loop*.
Firewall policies	Located in two sections: • *Overlay* section in the *Incoming interface* table under the *Policy* column: *Fabric_overlay_0* and *Fabric_overlay_1* • *Local Network* section in the *Shared subnets* table under the *Policies* column: *fabric_vpn_1_in*, *fabric_vpn_0_out*, and *fabric_vpn_0_in*

Configuring a downstream FortiGate using the Fabric Overlay Orchestrator

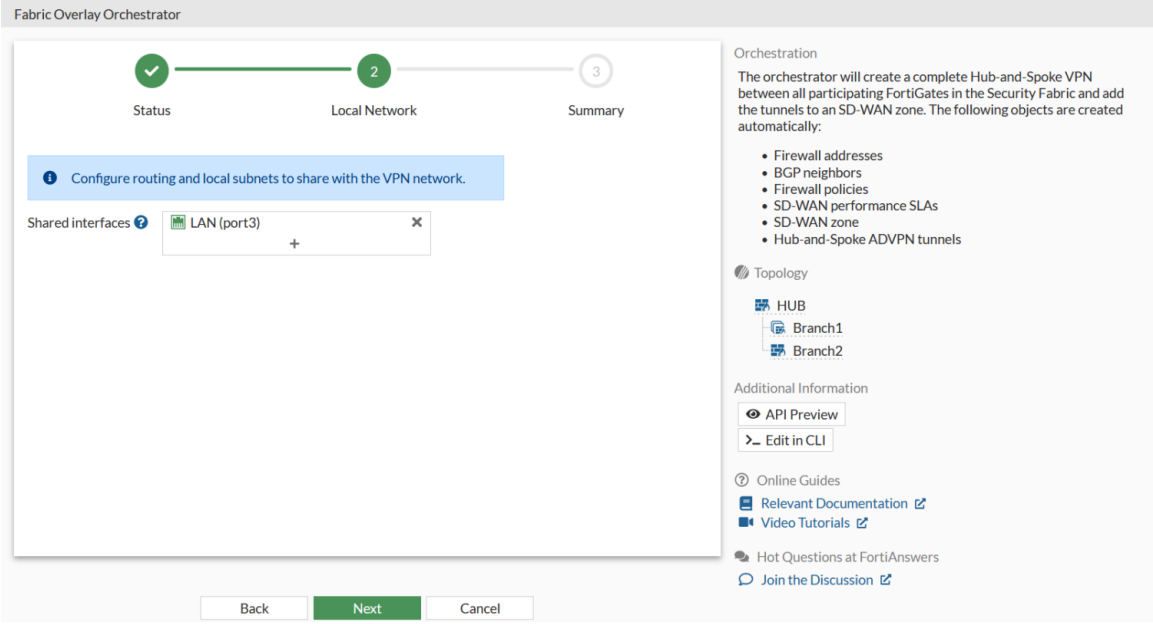
To configure a downstream FortiGate using the Fabric Overlay Orchestrator:

1. Go to *VPN* > *Fabric Overlay Orchestrator*.
2. Set the *Status* to *Enabled*. The *Role* is automatically selected based on the FortiGate's role in the Security Fabric. Ensure that *Spoke* is selected. Only downstream first-level FortiGates can be spokes.

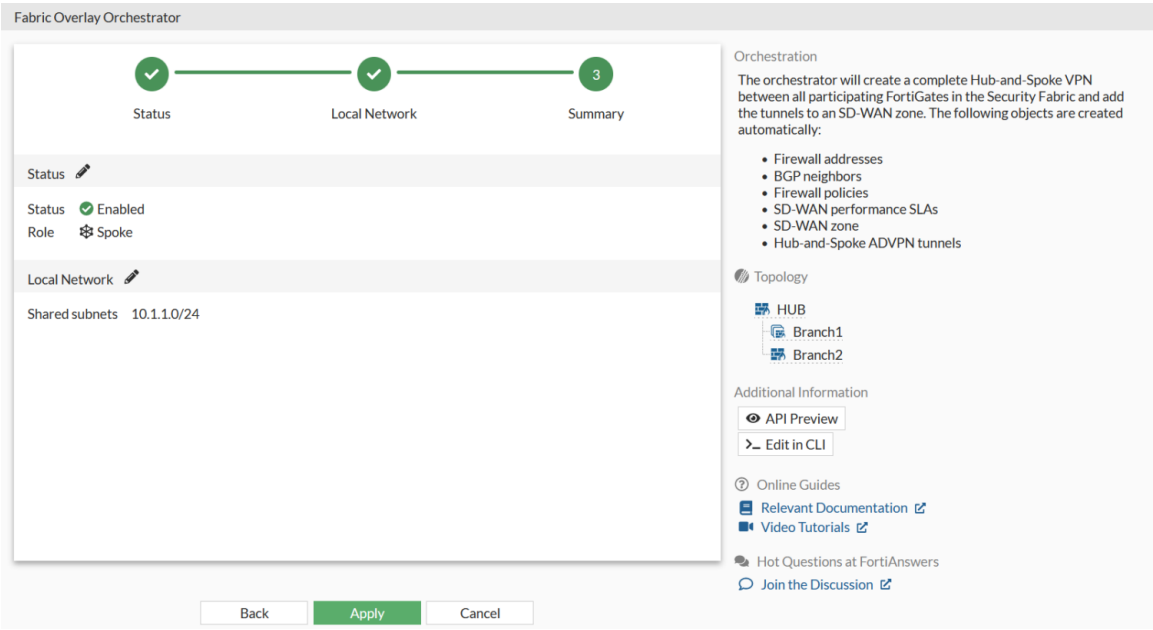


3. Click *Next*. The *Local Network* settings appear.
4. Configure the routing and local subnets to share with the VPN network in the following fields:

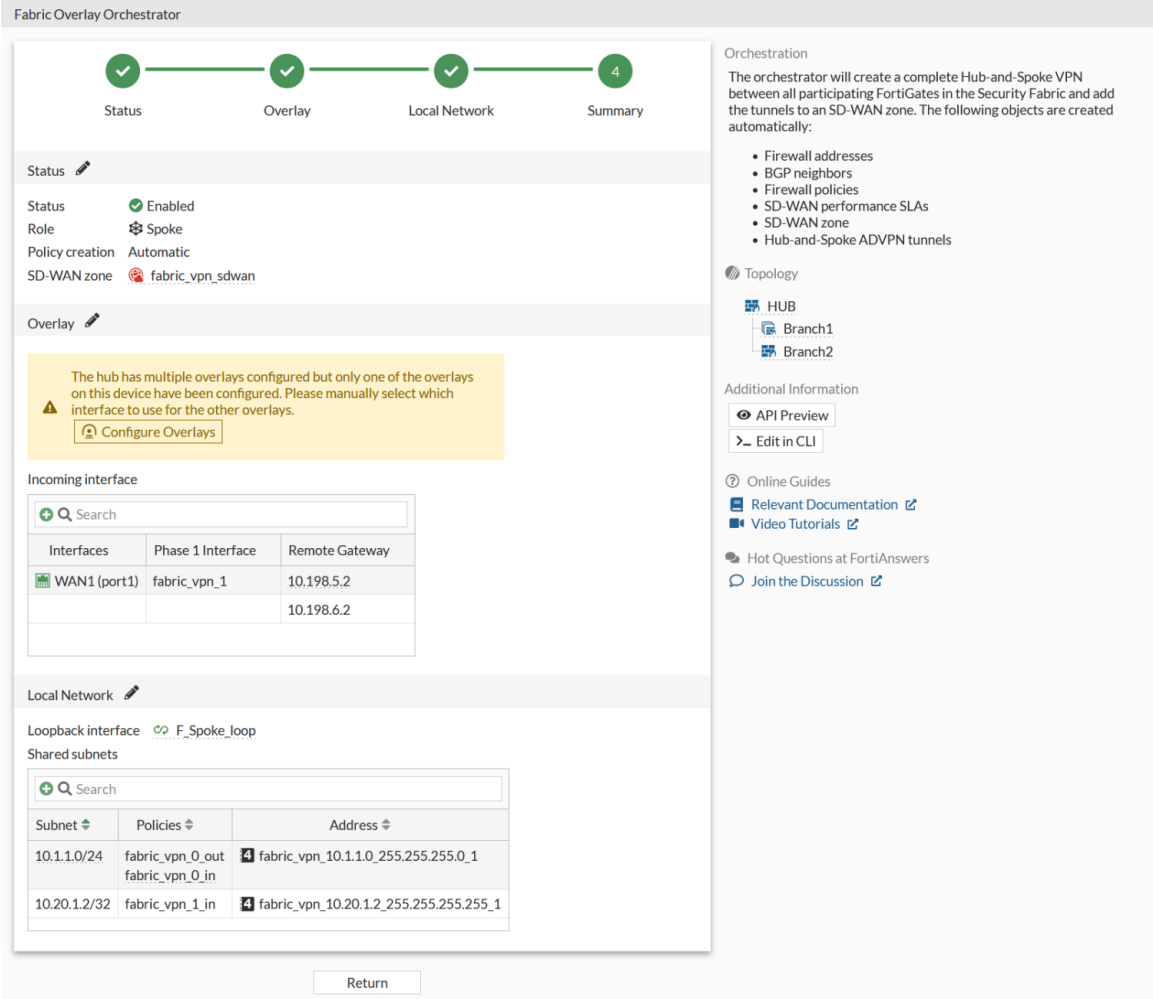
Shared interfaces	Select the interface of the local network to share with the VPN network.



5. Click *Next*. As the downstream FortiGate updates, a *Configuring spoke Fabric VPN from root FortiGate* message appears The *Summary* page appears once the update is complete.



6. Review the settings, then click *Apply*.
   An updated *Summary* page appears with all the settings.



Note the following settings in this example:

SD-WAN zone	Located in the *Status* section: *fabric_vpn_sdwan*.
VPN tunnels	Located in the *Overlay* section in the *Incoming interface* table under the *Phase 1 Interface* column: *fabric_vpn1*.
BGP	Located in the *Local Network* section. The *BGP AS* is *65400*. The *Shared subnets* are *10.1.1.0/24* and *10.20.1.2/32*.
Loopback interface	Located in the *Local Network* section: *F_Hub_loop*.
Firewall policies	Located in the *Local Network* section in the *Shared subnets* table under the *Policies* column: *fabric_vpn_0_out*, *fabric_vpn_0_in*, and *fabric_vpn_1_in*.

The loopback IP addresses for the branches are generated based on the index number of the trusted device in the root FortiGate's Security Fabric (HUB) configuration.

```
config system csf
    set status enable
    set group-name "fabric"
```

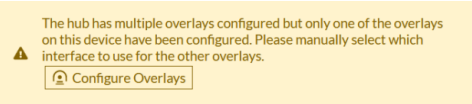
```
    config trusted-list
        edit "FGVM02TM22000001"
            set serial "FGVM02TM22000001"
            set index 1
        next
        edit "FGVM02TM22000002"
            set serial "FGVM02TM22000002"
            set index 2
        next
    end
end
```

For example, if Branch1 (index 1) is the first FortiGate and Branch2 (index 2) is the second FortiGate authorized on the root FortiGate, the loopback addresses are generated as follows:

- Branch1 loopback IP: 10.20.1.2
- Branch2 loopback IP: 10.20.1.3

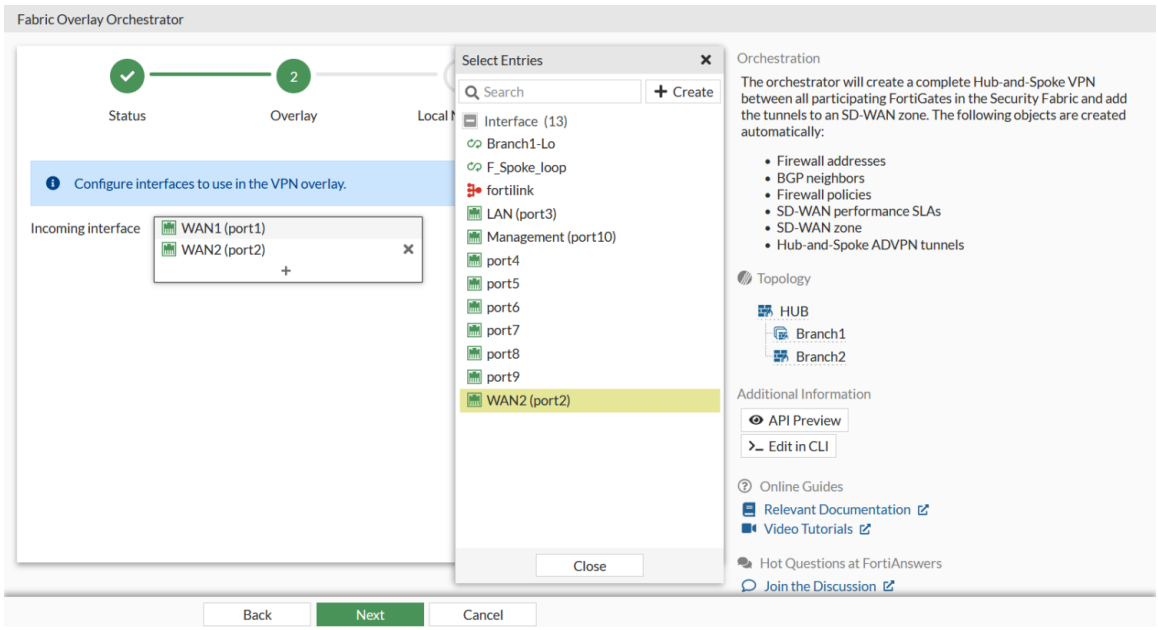
Configuring an overlay on the spoke for an additional incoming interface on the hub

A hub typically includes two incoming interfaces, but additional interfaces can be configured if needed. On downstream devices, the following warning is displayed on the *Fabric Overlay Orchestrator* page that *The hub has multiple overlays configured but only one of the overlays on this device have been configured. Please manually select which interface to use for the other overlays.*

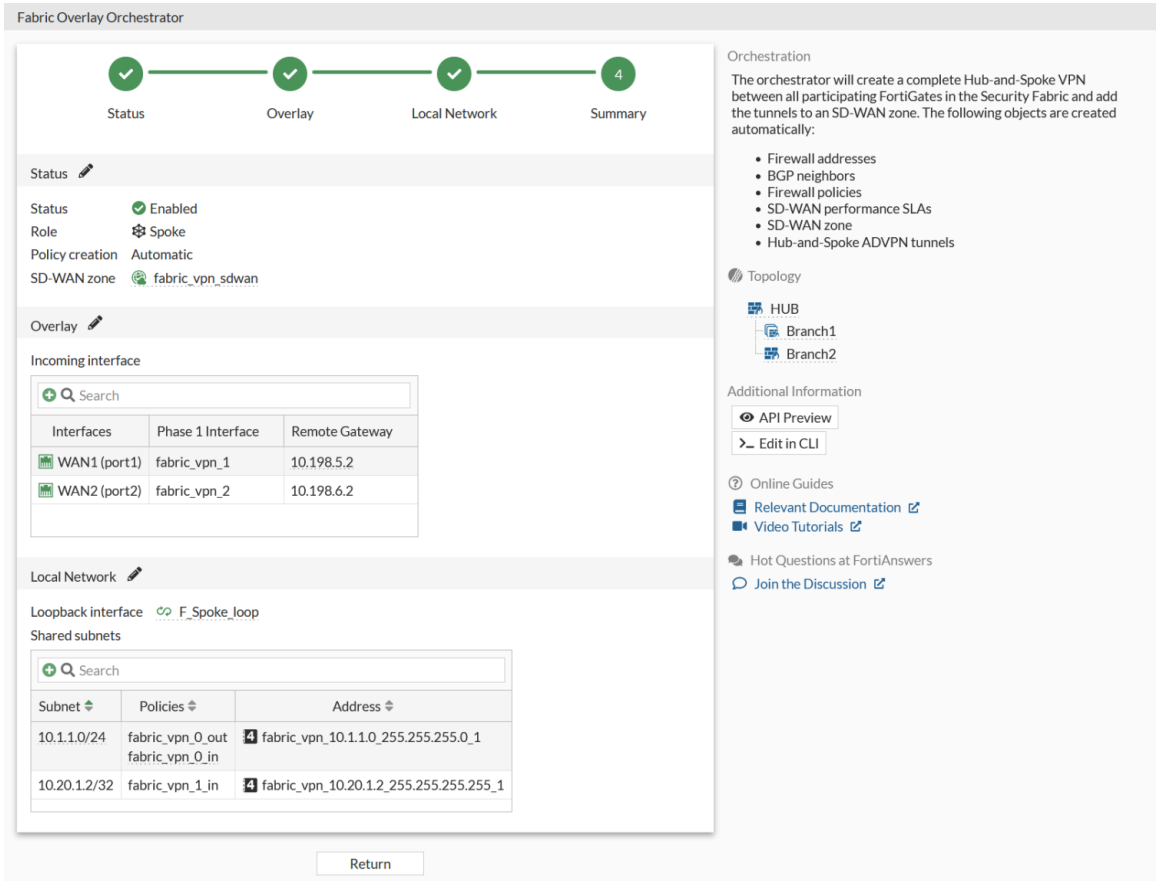


To configure an additional incoming interface on a spoke:

1. Go to *VPN > Fabric Overlay Orchestrator*.
2. Click *Configure Overlays* in the warning box.
3. Navigate to the *Overlay* section, click the *+* in the *Incoming interface* field, and select *WAN2 (port2)* to add it to the overlay.



4. Click *Next*, then complete the remaining steps in the GUI wizard. On the *Summary* page, the additional interface *WAN2 (port2)* appears in the *Incoming interfaces* table.

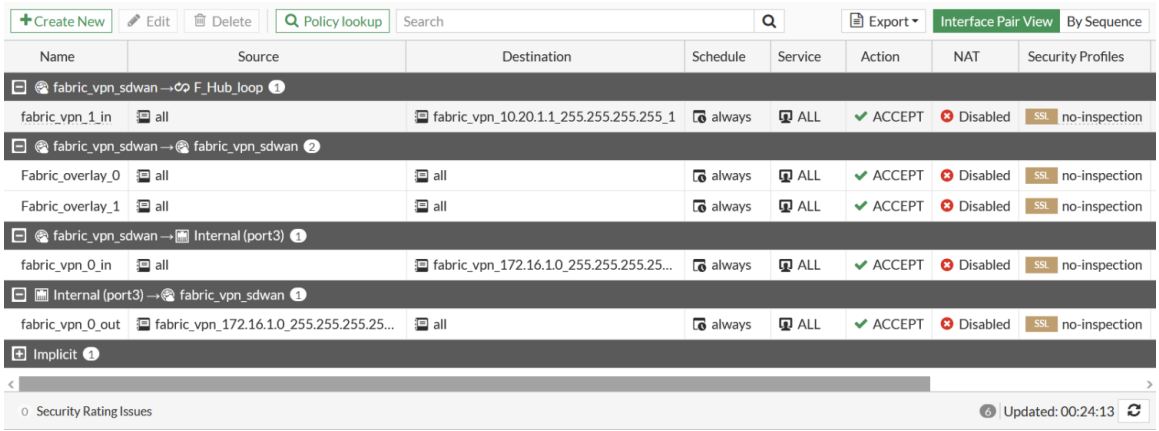


Verifying the firewall policies on the hub FortiGate

Different policies are created on the hub FortiGate based on the *Policy creation* setting in the Fabric Overlay Orchestrator configuration (*Automatic*, *Health check*, or *Manual*).

Automatic

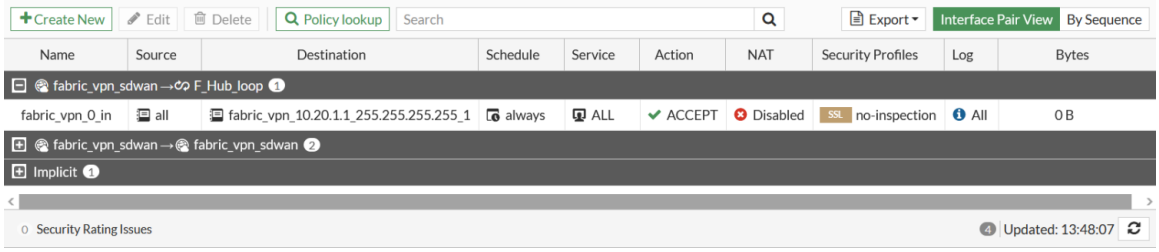
Go to *Policy & Objects* > *Firewall Policy* to verify that wildcard firewall policies have been configured on the hub FortiGate. This Fabric Overlay Orchestrator configuration example uses automatic policy creation, and the following firewall policies are configured:



> The *Automatic* policy creation option creates wildcard allow policies for the tunnel overlays. For some cases, these policies do not provide the necessary granularity to restrict overlay traffic to specific subnets or hosts.

Health check

Go to *Policy & Objects* > *Firewall Policy* to verify that a single firewall policy allowing health check traffic to the hub's loopback has been configured on the hub FortiGate. For example:



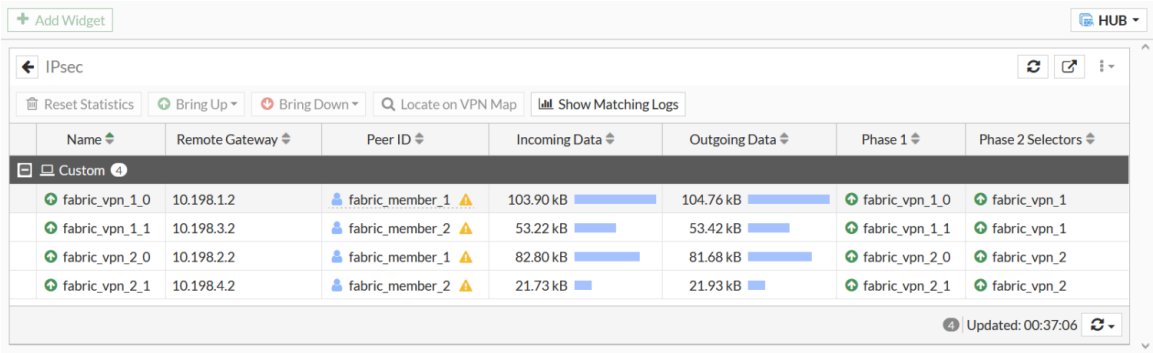
Manual

Go to *Policy & Objects* > *Firewall Policy* to verify that no firewall policies have been created by the Fabric Overlay Orchestrator. If desired, firewall policies must be manually configured on the hub FortiGate to allow traffic to the loopback interface for health checks and the overlays.

Verifying the Fabric Overlay created by the Fabric Overlay Orchestrator

To verify the IPsec VPN tunnels on the hub:

1. Go to *Dashboard* > *Network Monitor* > *VPN*.
2. Verify that there are two tunnels established for each phase 1 interface.



The naming convention <tunnel_name>_<number> indicates the relative order in which the tunnels were established:

fabric_vpn_1_0	VPN tunnel listening on the hub's WAN1 incoming interface; established with spoke 1 using its WAN1 interface
fabric_vpn_1_1	VPN tunnel listening on the hub's WAN1 incoming interface; established with spoke 2 using its WAN1 interface
fabric_vpn_2_0	VPN tunnel listening on the hub's WAN2 incoming interface; established with spoke 1 using its WAN2 interface
fabric_vpn_2_1	VPN tunnel listening on the hub's WAN2 incoming interface; established with spoke 2 using its WAN2 interface

Verify the BGP routing on the hub:

1. In the CLI, check the BGP peering status:

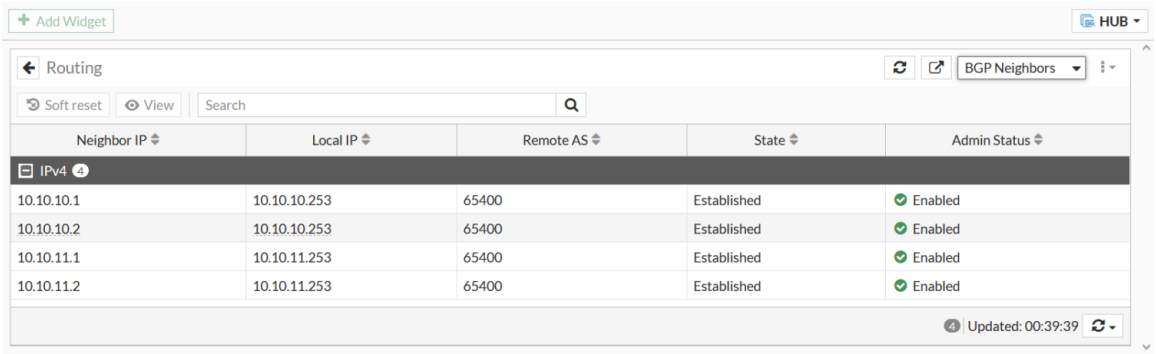
```
HUB # get router info bgp summary

VRF 0 BGP router identifier 10.20.1.1, local AS number 65400
BGP table version is 11
1 BGP AS-PATH entries
0 BGP community entries
Next peer check timer due in 43 seconds

Neighbor    V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.10.10.1  4      65400      23      27       11    0    0 00:09:28        2
10.10.10.2  4      65400      16      16       11    0    0 00:06:36        2
10.10.11.1  4      65400      14      20       11    0    0 00:09:22        2
10.10.11.2  4      65400       7      11       11    0    0 00:03:22        2
```

```
Total number of neighbors 4
```

2. Check the BGP advertised routes:

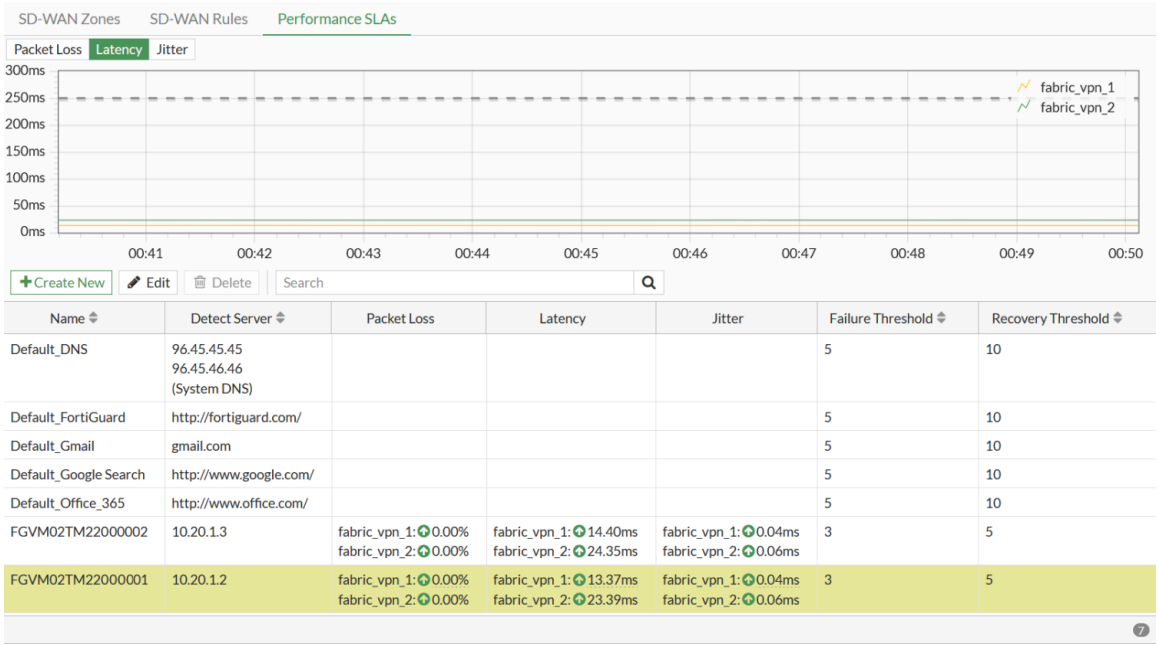
```
HUB # get router info bgp neighbors 10.10.10.1 advertised-routes
VRF 0 BGP table version is 11, local router ID is 10.20.1.1
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
*>i10.1.1.0/24      10.10.11.1                       100      0        0 i <-/2>
*>i10.1.2.0/24      10.10.11.2                       100      0        0 i <-/2>
*>i10.1.2.0/24      10.10.10.2                       100      0        0 i <-/1>
*>i10.10.10.0/24    10.10.10.253                     100  32768        0 i <-/1>
*>i10.10.11.0/24    10.10.10.253                     100  32768        0 i <-/1>
*>i10.20.1.1/32     10.10.10.253                     100  32768        0 i <-/1>
*>i10.20.1.2/32     10.10.11.1                       100      0        0 i <-/2>
*>i10.20.1.3/32     10.10.11.2                       100      0        0 i <-/2>
*>i10.20.1.3/32     10.10.10.2                       100      0        0 i <-/1>
*>i172.16.1.0/30    10.10.10.253                     100  32768        0 i <-/1>

Total number of prefixes 10
```

3. In the GUI, go to *Dashboard > Network Monitor > Routing*.
4. In the dropdown, select *BGP Neighbors*.



To verify the performance SLAs on the hub:

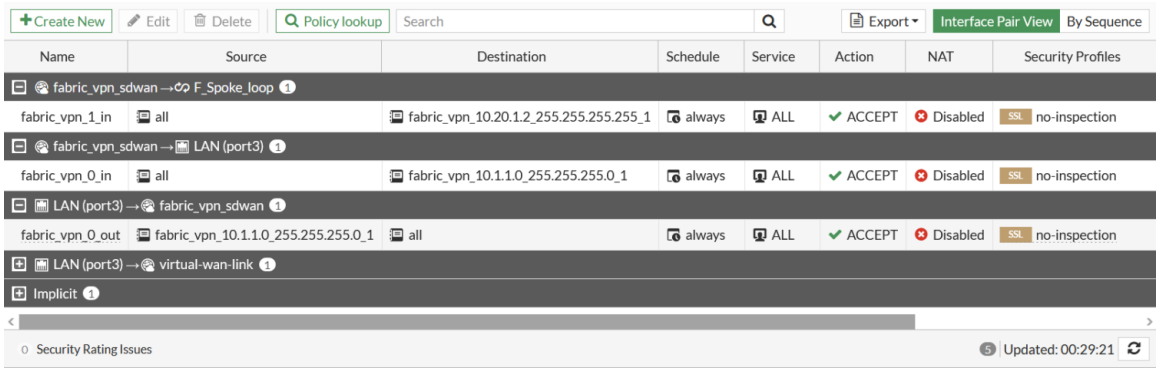
1. Go to *Network > SD-WAN* and select the *Performance SLAs* tab.
2. Verify that the performance SLAs are automatically created for each spoke. The performance SLA naming uses the serial number of the spoke FortiGate. There are two new entries.



To verify the firewall policies on a spoke FortiGate:

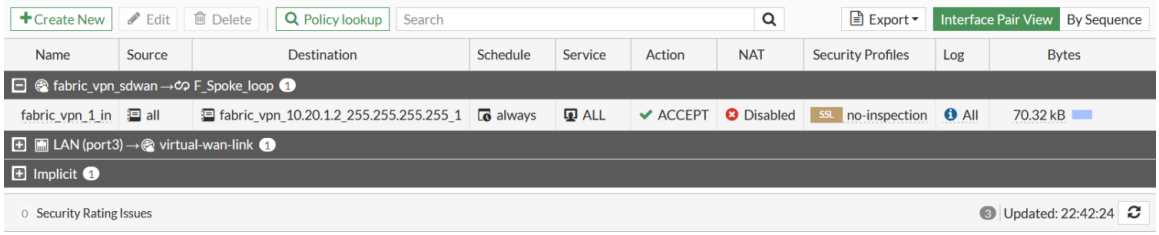
Different policies are created on the spoke FortiGates based on the hub's *Policy creation* setting in the Fabric Overlay Orchestrator configuration (*Automatic*, *Health check*, or *Manual*). The *Automatic* setting is used in this example.

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Verify that wildcard firewall policies have been configured.



The *Automatic* policy creation option creates wildcard allow policies for the tunnel overlays. For some cases, these policies do not provide the necessary granularity to restrict overlay traffic to specific subnets or hosts.

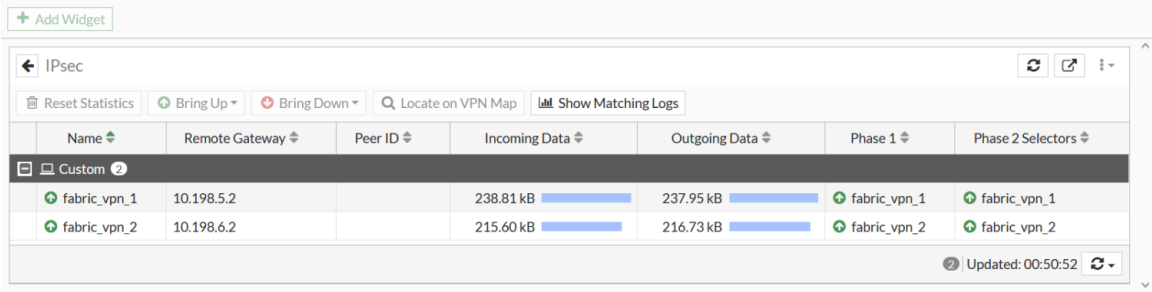
If the hub's *Policy creation*setting is *Health Check*, a single firewall policy that allows health check traffic to the spoke's loopback should be configured on the spoke FortiGates:



If the hub's *Policy creation*setting is *Manual*, there should be no new policies created by the Fabric Overlay Orchestrator. If desired, firewall policies must be manually configured on the spoke FortiGates to allow traffic to the loopback interface for health checks and the overlays.

To verify the IPsec VPN tunnels on a spoke:

1. Go to *Dashboard* > *Network Monitor* > *VPN*.
2. Verify the IPsec tunnels that go back to the hub.



To verify BGP routing on a spoke:

1. In the CLI, check the BGP peering status:

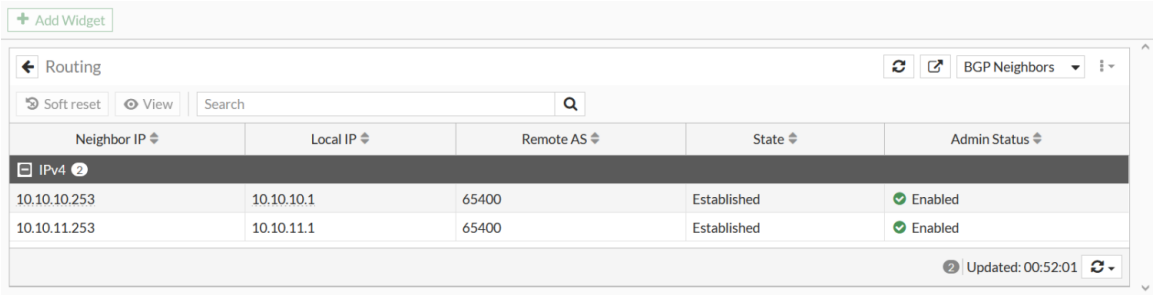
```
Branch1 # get router info bgp summary

VRF 0 BGP router identifier 10.20.1.2, local AS number 65400
BGP table version is 5
1 BGP AS-PATH entries
0 BGP community entries

Neighbor     V         AS MsgRcvd MsgSent   TblVer  InQ OutQ Up/Down  State/PfxRcd
10.10.10.253 4      65400      41      37        4    0    0 00:23:39        8
10.10.11.253 4      65400      38      34        4    0    0 00:23:33        8

Total number of neighbors 2
```

2. Check the BGP advertised routes:

```
Branch1 # get router info bgp neighbors 10.10.10.253 advertised-routes
VRF 0 BGP table version is 5, local router ID is 10.20.1.2
Status codes: s suppressed, d damped, h history, * valid, > best, i - internal
Origin codes: i - IGP, e - EGP, ? - incomplete

   Network          Next Hop            Metric     LocPrf Weight RouteTag Path
```

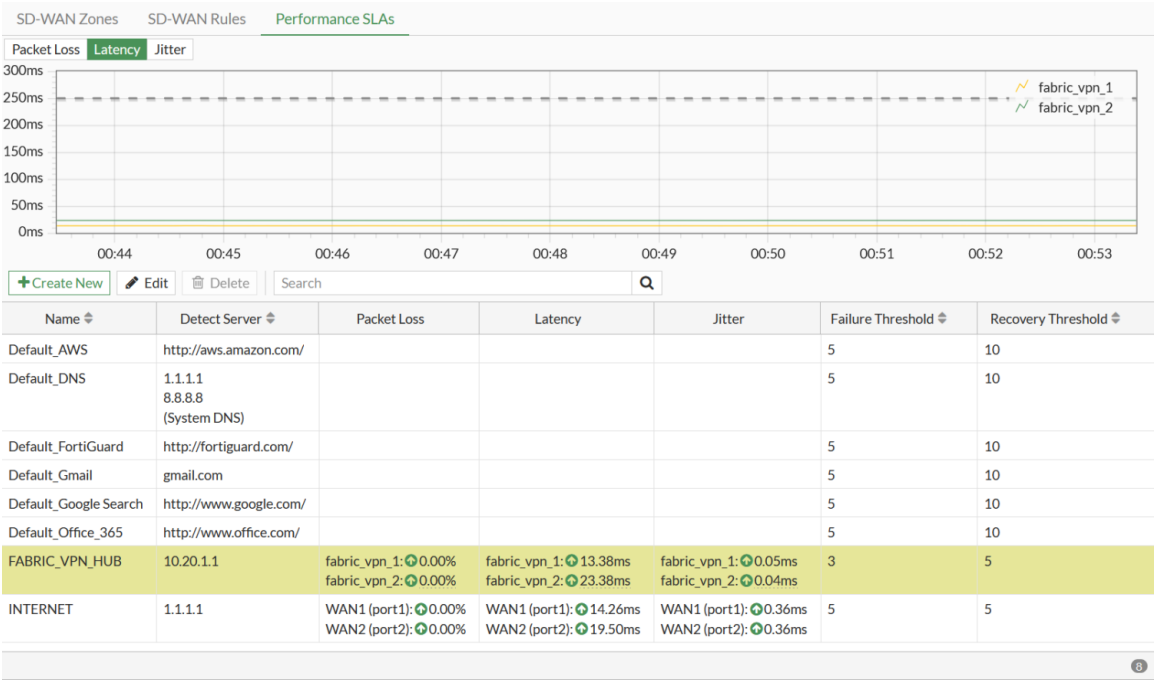
```
*>i10.1.1.0/24      10.10.10.1                    100  32768        0 i <0/->
*>i10.20.1.2/32     10.10.10.1                    100  32768        0 i <0/->

Total number of prefixes 2
```

3. In the GUI, go to *Dashboard* > *Network Monitor* > *Routing*.
4. In the dropdown, select *BGP Neighbors*.



To verify the performance SLAs on a spoke:

1. Go to *Network* > *SD-WAN* and select the *Performance SLAs* tab.
2. Verify that the performance SLA is automatically created for the hub FortiGate. There is a new entry (*FABRIC_VPN_HUB*).



To verify the spoke-to-spoke ADVPN communication:

1. From Branch1, ping Branch2 (10.20.1.3):

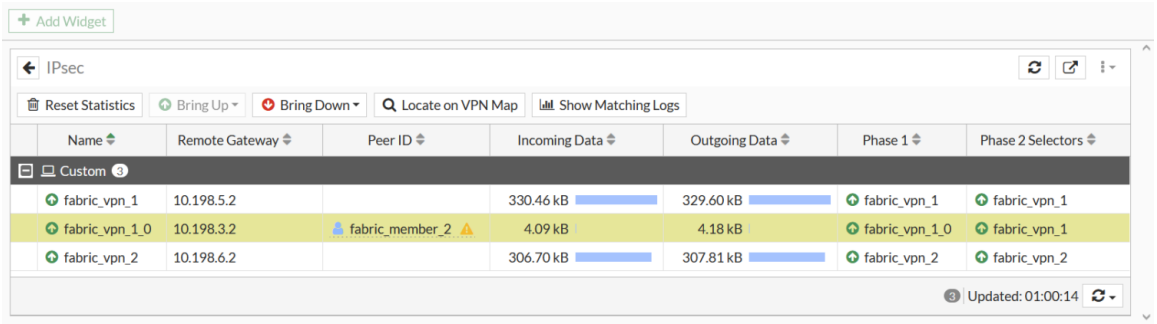
```
Branch1 # exec ping-options source 10.20.1.2
Branch1 # exec ping 10.20.1.3
```

```
PING 10.20.1.3 (10.20.1.3): 56 data bytes
64 bytes from 10.20.1.3: icmp_seq=0 ttl=254 time=27.7 ms
64 bytes from 10.20.1.3: icmp_seq=2 ttl=255 time=17.4 ms
64 bytes from 10.20.1.3: icmp_seq=3 ttl=255 time=17.5 ms
64 bytes from 10.20.1.3: icmp_seq=4 ttl=255 time=17.4 ms

--- 10.20.1.3 ping statistics ---
5 packets transmitted, 4 packets received, 20% packet loss
round-trip min/avg/max = 17.4/20.0/27.7 ms
```

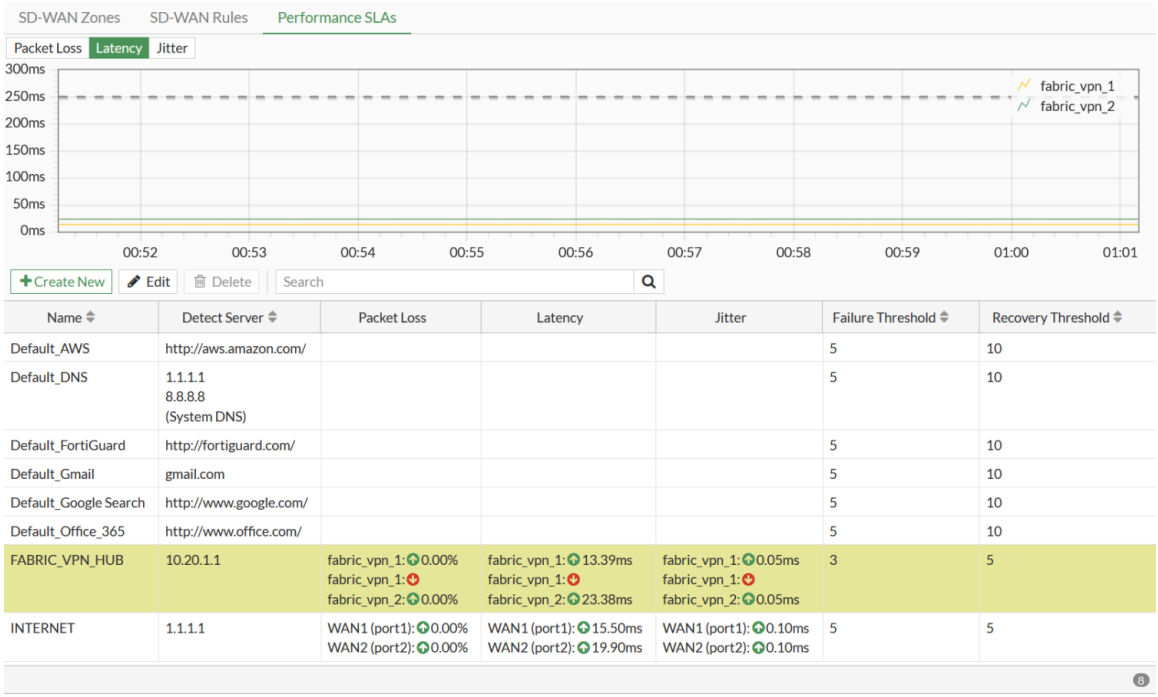
2. Verify the IPsec tunnel summary.
   - In the CLI, enter the following:

     ```
     Branch1 # get vpn ipsec tunnel summary
     'fabric_vpn_1_0' 10.198.3.2:0  selectors(total,up): 1/1  rx(pkt,err): 25/0  tx(pkt,err):
     26/2
     'fabric_vpn_1' 10.198.5.2:0  selectors(total,up): 1/1  rx(pkt,err): 8032/0  tx(pkt,err):
     8022/2
     'fabric_vpn_2' 10.198.6.2:0  selectors(total,up): 1/1  rx(pkt,err): 7462/0  tx(pkt,err):
     7478/1
     ```

     The `fabric_vpn_1_0` tunnel was created for spoke 1-to-spoke 2 communication.
   - In the GUI, go to *Dashboard > Network Monitor > VPN*.



3. Verify that the performance SLA was updated. Go to *Network > SD-WAN* and select the *Performance SLAs* tab.



Name	Detect Server	Packet Loss	Latency	Jitter	Failure Threshold	Recovery Threshold
Default_AWS	http://aws.amazon.com/				5	10
Default_DNS	1.1.1.1 8.8.8.8 (System DNS)				5	10
Default_FortiGuard	http://fortiguard.com/				5	10
Default_Gmail	gmail.com				5	10
Default_Google Search	http://www.google.com/				5	10
Default_Office_365	http://www.office.com/				5	10
FABRIC_VPN_HUB	10.20.1.1	fabric_vpn_1: 0.00% fabric_vpn_1: fabric_vpn_2: 0.00%	fabric_vpn_1: 13.39ms fabric_vpn_1: fabric_vpn_2: 23.38ms	fabric_vpn_1: 0.05ms fabric_vpn_1: fabric_vpn_2: 0.05ms	3	5
INTERNET	1.1.1.1	WAN1 (port1): 0.00% WAN2 (port2): 0.00%	WAN1 (port1): 15.50ms WAN2 (port2): 19.90ms	WAN1 (port1): 0.10ms WAN2 (port2): 0.10ms	5	5

The first performance SLA, *fabric_vpn_1*, that corresponds to the spoke-to-hub VPN tunnel is shown as up. The second one, *fabric_vpn_1* that corresponds to the spoke-to-spoke VPN tunnel (*fabric_vpn_1_0*) is shown as down since 10.20.1.1 is the IP address corresponding to the hub's loopback interface that is not present on another spoke.

Configuring SD-WAN rules on the hub FortiGate

On the hub, the Fabric Overlay Orchestrator automatically creates a performance SLA that corresponds to each spoke FortiGate using the serial number as the name of the performance SLA. SD-WAN rules must be configured on the hub FortiGate to direct traffic to each of the spokes using these performance SLAs.

To configure SD-WAN rules on the hub FortiGate:

1. Go to *Network* > *SD-WAN*, select the *SD-WAN Rules* tab, and click *Create New*.
2. Enter a name (such as *Hub-To-Br1*).
3. In the *Source* section, set the *Address* to the local subnet of the hub.
4. Configure the following in the *Destination* section:
   a. Set the *Address* to the local subnet of the spoke. If an address object does not exist yet, click *Create* in the slide-out pane and configure the address.
   b. Set the *Protocol number* as needed (default = *ANY*).
5. Configure the following in the *Outgoing Interfaces* section:
   a. Set the *Interface selection strategy* to *Lowest cost (SLA)*.
   b. Set the *Interface preference* to the SD-WAN members.
   c. Set *Required SLA target* to the corresponding performance SLA created by the Fabric Overlay Orchestrator for this the spoke. The name is based on the spoke FortiGate's serial number (*FGVM0XXX00000000 #1*).

6. Click *OK*.
7. Repeat these steps for the other spoke. Ensure the *Name* is unique, and that the *Destination* address corresponds to the local subnet behind the spoke.

If you need to disable the Fabric Overlay Orchestrator on the hub FortiGate by setting the *Status* to *Disabled*, you must first delete any SD-WAN rules on the hub FortiGate created using this procedure to ensure the added configuration does not block the clean-up process.

Configuring SD-WAN rules on the spoke FortiGates

On each spoke, the Fabric Overlay Orchestrator automatically creates a performance SLA that corresponds to the hub FortiGate. An SD-WAN rule must be configured on the spoke FortiGates to direct traffic to the hub FortiGate using this performance SLA.

To configure an SD-WAN rule on a spoke FortiGate:

1. Go to *Network* > *SD-WAN*, select the *SD-WAN Rules* tab, and click *Create New*.
2. Enter a name (such as *LAN-to-HUB*).
3. In the *Source* section, set the *Address* to the local subnet of the spoke.
4. Configure the following in the *Destination* section:
   a. Set the *Address* to the local subnet of the hub. If an address object does not exist yet, click *Create* in the slide-out pane and configure the address.
   b. Set the *Protocol number* as needed (default = *ANY*).
5. Configure the following in the *Outgoing Interfaces* section:
   a. Set the *Interface selection strategy* to *Lowest cost (SLA)*.
   b. Set the *Interface preference* to the SD-WAN members.
   c. Set *Required SLA target* to the corresponding performance SLA created by the Fabric Overlay Orchestrator, which is named *FABRIC_VPN_HUB#1* by default.
6. Click *OK*.

If you need to disable the Fabric Overlay Orchestrator on a spoke FortiGate by setting the *Status* to *Disabled*, you must first delete any SD-WAN rules on the spoke FortiGate created using this procedure to ensure the added configuration does not block the clean-up process.

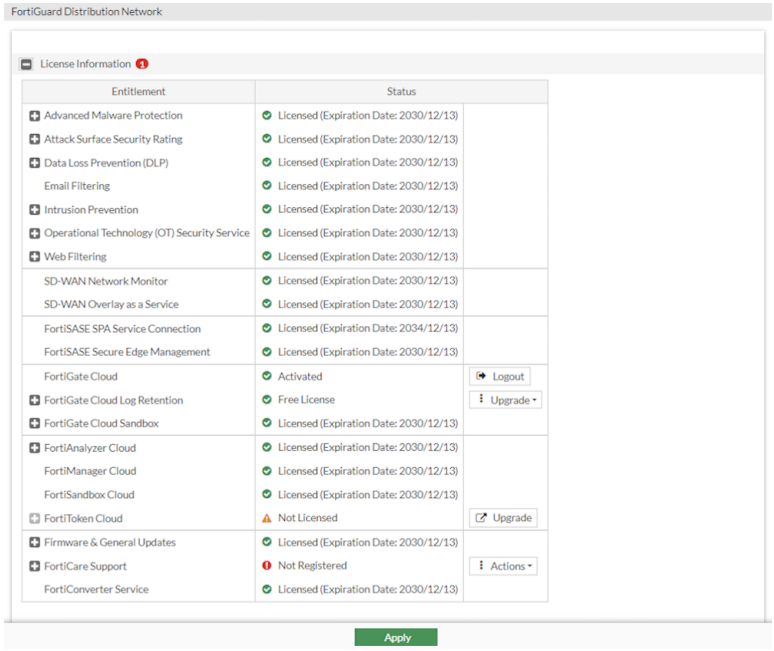
SPA easy configuration key for FortiSASE

A gutter section is available in the *Fabric Overlay Orchestrator* page if the FortiSASE SPA license is active. From this section, the user can open a pane that will generate a FortiSASE SPA easy configuration key based on the current Fabric Overlay Orchestrator configuration which can be used in the SPA setup of FortiSASE.

The easy configuration key is an encode of Base64 of a JSON object. It includes the FortiOS version, gateway, peer IP address, BGP Autonomous Systems (AS), BGP method, and the FortiSASE BGP router subnet.

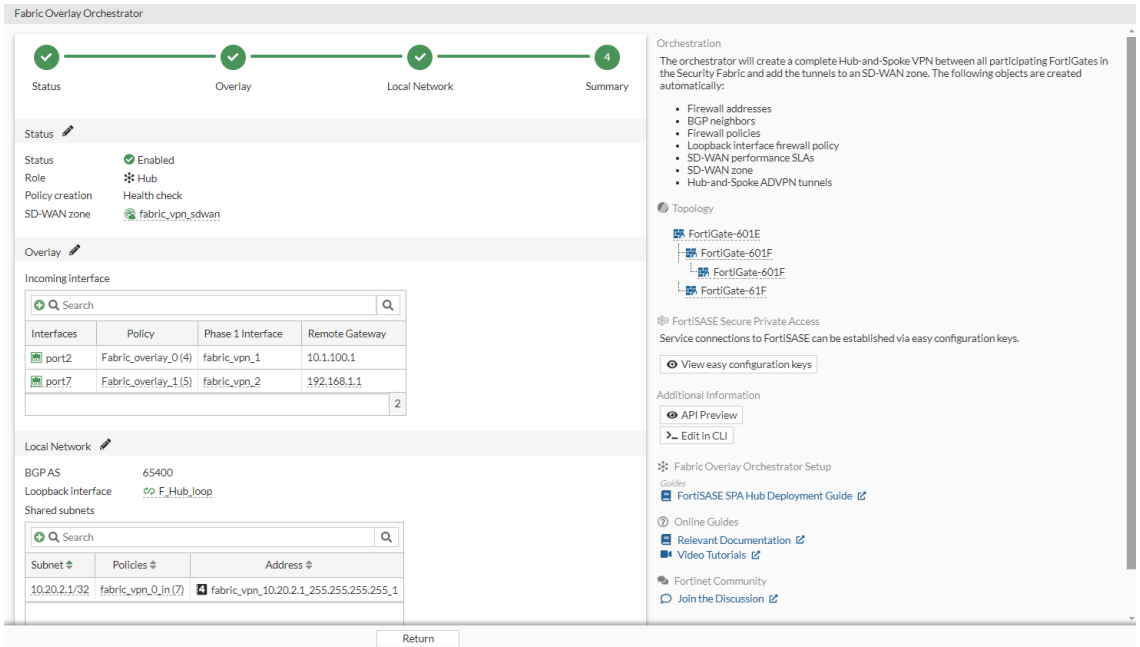
To access the easy configuration key:

1. Prepare the two FortiGates:
   a. Go to *System* > *FortiGuard* and confirm that the FortiGates have a FortiSASE SPA license.



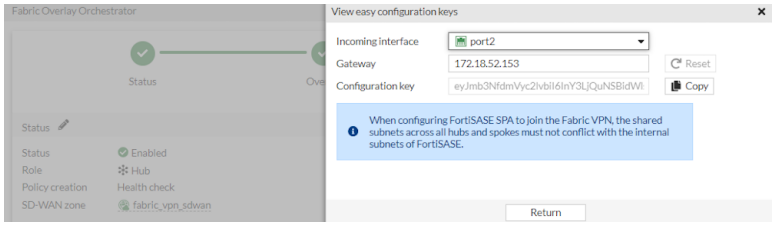
   b. Configure the Security Fabric in both FortiGates.
   c. Go to *VPN* > *Fabric Overlay Orchestrator* and enable it on both FortiGates. See Using the Fabric Overlay Orchestrator on page 3279.

   The *FortiSASE Secure Private Access* section will be included in the gutter.



2. Access the easy configuration key:

a. In *VPN* > *Fabric Overlay Orchestrator*, click *View easy configuration keys* in the gutter.
The *View easy configuration keys* pane is displayed.



b. Select the *Incoming interface*.
c. Enter the *Gateway*.

> Shared subnets cannot conflict with FortiSASE's internal subnets.

d. Click *Copy* beside the *Configuration key*.
This easy configuration key can be pasted into FortiSASE when setting up SPA.

> If the FortiOS administrator makes any changes to BGP or to their IPsec, the configuration key is auto-updated on FortiOS. However, the FortiSASE administrator needs to know to re-copy and paste the configuration into FortiSASE.
>
> Currently, there is no way to detect the above and throw a warning on either FortiSASE nor FortiOS GUI.

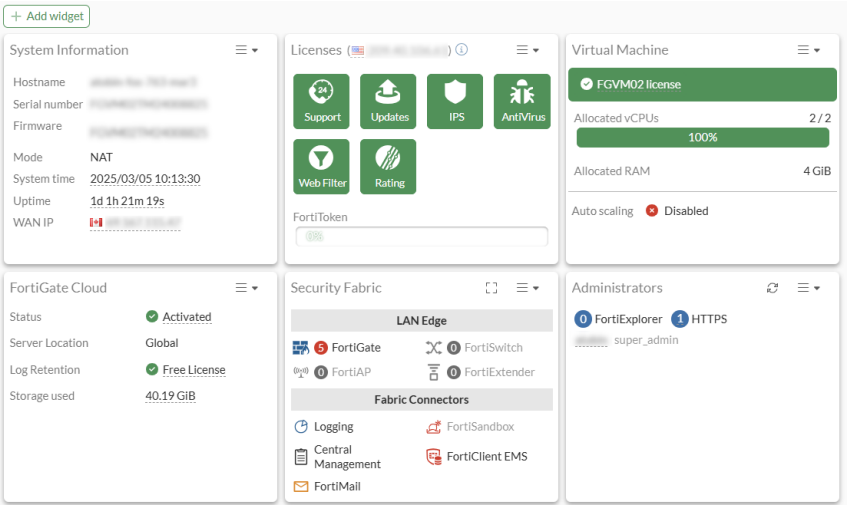
Fabric Overlay Orchestrator Topology dashboard widget for hub FortiGates

The *Fabric Overlay Orchestrator Topology* dashboard widget provides an interactive view of hub and spoke devices previously configured using the Fabric Overlay Orchestrator feature. It can display health-check status, multiple tunnels, resource usage, shortcuts, and so on.

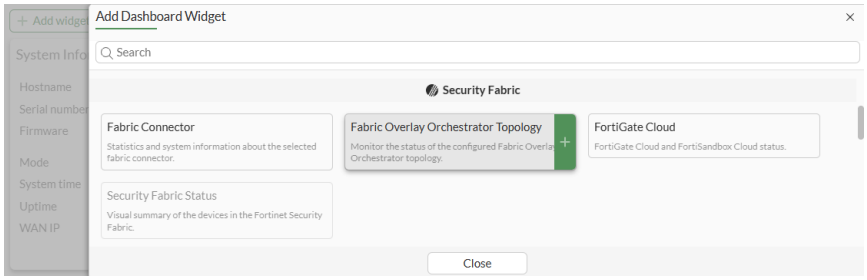
> This dashboard widget is only available on the hub or root FortiGate device.

To enable the Fabric Overlay Orchestrator Topology widget:

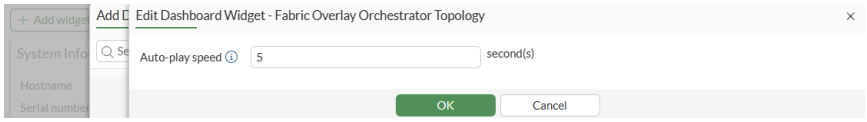
1. Go to *Dashboard* > *Status*.



2. Click *Add widget*.
3. Select *Fabric Overlay Orchestrator Topology* under *Security Fabric*.



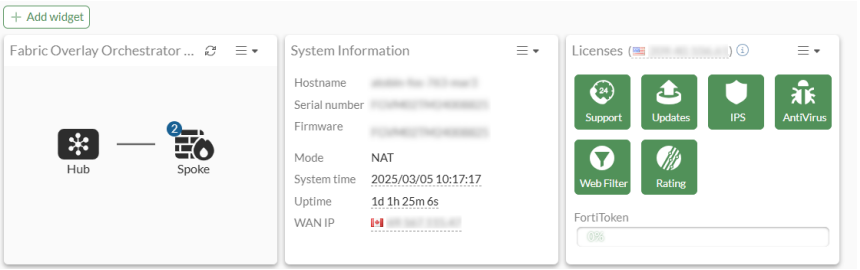
4. Enter the *Auto-play speed*.



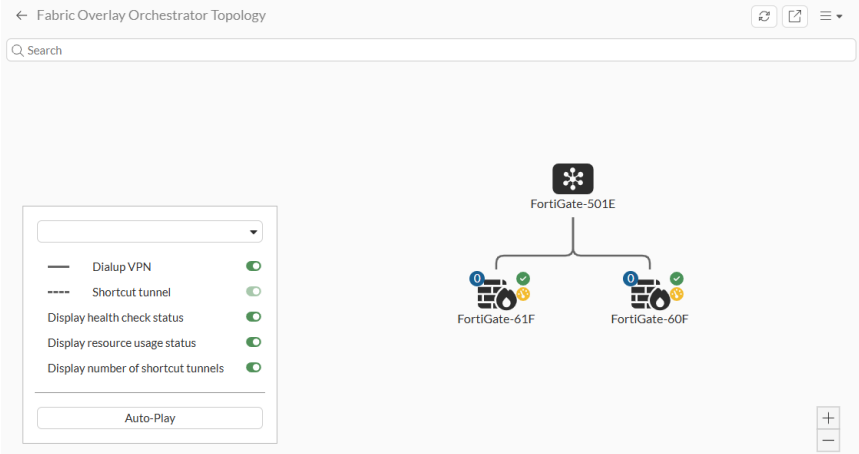
5. Click *OK*.
6. Click *Close*.

To review hub and spoke details in the Fabric Overlay Orchestrator Topology widget:

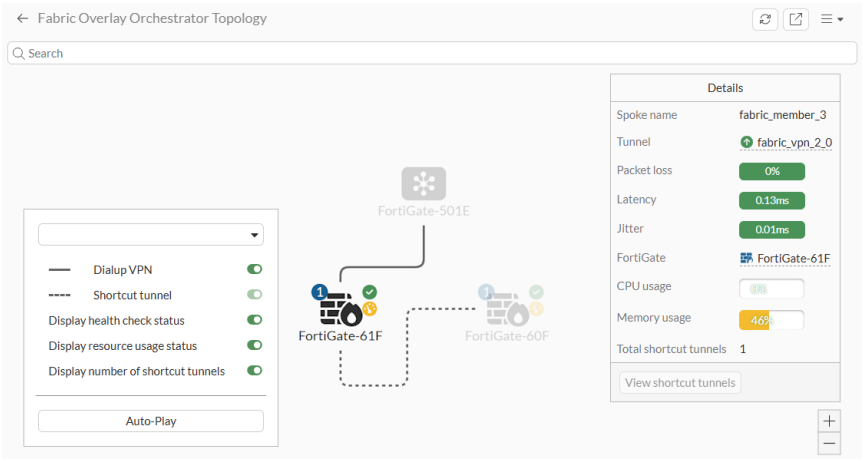
1. Go to *Dashboard* > *Status*.



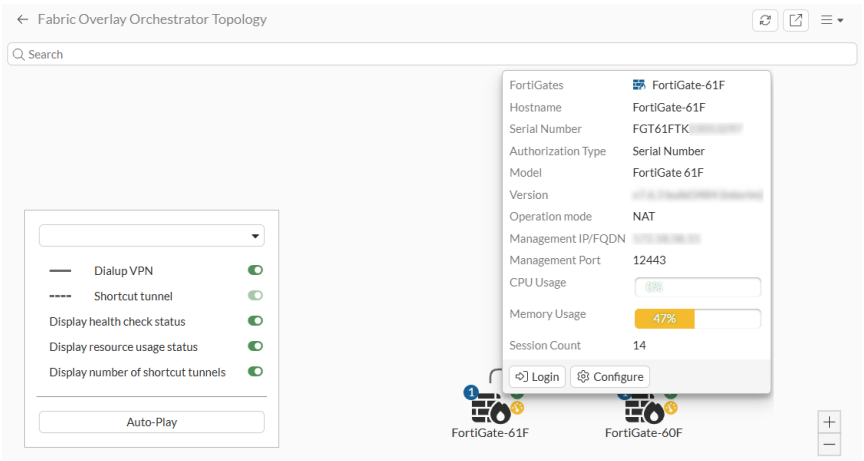
2. Click the *Fabric Overlay Orchestrator Topology* widget to expand it.



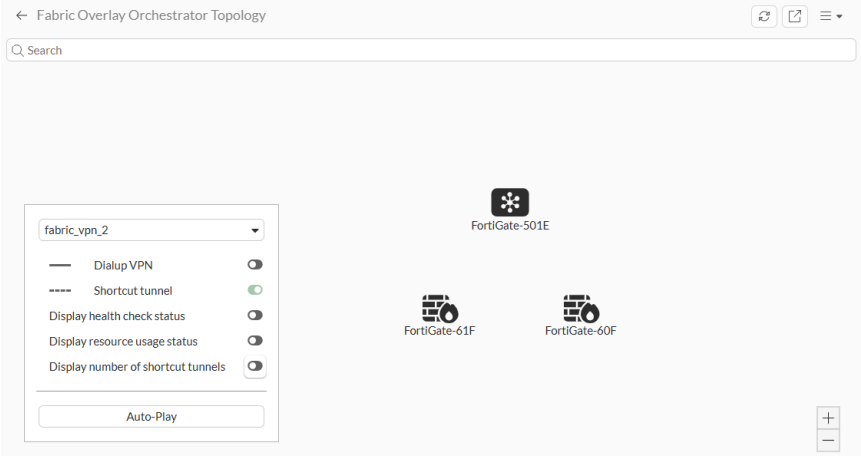
3. Review the topology:
   - Select a hub or spoke in the topology to view *Details*, such as *Latency*, *Jitter*, and usage information.



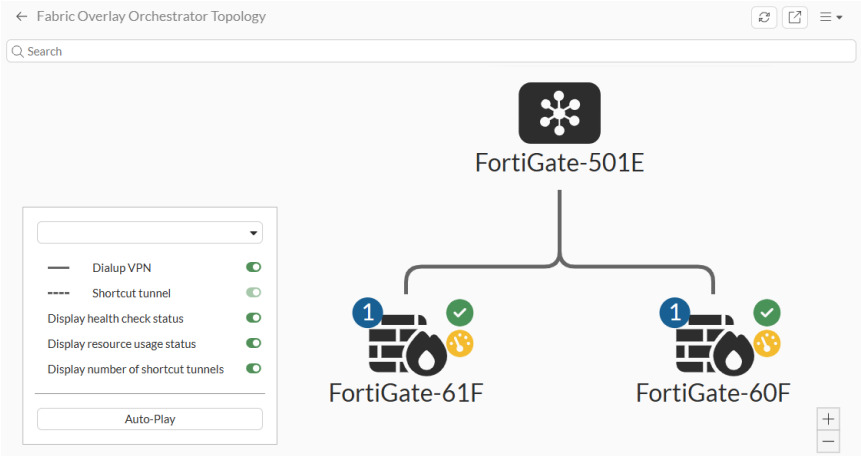
   - Hover over a hub or spoke in the topology to view device information, such as *Serial Number*, *Model*, and *Version* information.



   - Toggle the display options to define what is displayed.



- Use the + and - to zoom in and out of the topology.



Other VPN topics

The following topics provide instructions on configuring other VPN topics.

- VPN and ASIC offload on page 3302
- Encryption algorithms on page 2843
- Configure DSCP for IPsec tunnels on page 3313
- IPsec IKE load balancing based on FortiSASE account information on page 3315
- IPsec SA key retrieval from a KMS server using KMIP on page 3318
- Multiple interface monitoring for IPsec on page 3329

VPN and ASIC offload

This topic provides a brief introduction to VPN traffic offloading.

IPsec traffic processed by NPU

1. Check the device ASIC information. For example, a FortiGate 900D has an NP6 and a CP8.

```
# get hardware status
Model name: [[QualityAssurance62/FortiGate]]-900D
ASIC version: CP8
ASIC SRAM: 64M
CPU: Intel(R) Xeon(R) CPU E3-1225 v3 @ 3.20GHz
Number of CPUs: 4
RAM: 16065 MB
Compact Flash: 1925 MB /dev/sda
Hard disk: 244198 MB /dev/sdb
USB Flash: not available
Network Card chipset: [[QualityAssurance62/FortiASIC]] NP6 Adapter (rev.)
```

2. Check port to NPU mapping.

```
# diagnose npu np6 port-list
Chip   XAUI Ports            Max   Cross-chip
                             Speed offloading
----
np6_0  0
        1.    port17           1G    Yes
        1.    port18           1G    Yes
        1.    port19           1G    Yes
        1.    port20           1G    Yes
        1.    port21           1G    Yes
        1.    port22           1G    Yes
        1.    port23           1G    Yes
        1.    port24           1G    Yes
        1.    port27           1G    Yes
        1.    port28           1G    Yes
        1.    port25           1G    Yes
        1.    port26           1G    Yes
        1.    port31           1G    Yes
        1.    port32           1G    Yes
        1.    port29           1G    Yes
        1.    port30           1G    Yes
        1.    portB            10G   Yes
        1.
----
np6_1  0
        1.    port1            1G    Yes
        1.    port2            1G    Yes
        1.    port3            1G    Yes
        1.    port4            1G    Yes
        1.    port5            1G    Yes
        1.    port6            1G    Yes
        1.    port7            1G    Yes
        1.    port8            1G    Yes
        1.    port11           1G    Yes
```

```
        1.    port12          1G    Yes
        1.    port9           1G    Yes
        1.    port10          1G    Yes
        1.    port15          1G    Yes
        1.    port16          1G    Yes
        1.    port13          1G    Yes
        1.    port14          1G    Yes
        1.    portA           10G   Yes
        1.
----
```

3. Configure the option in IPsec phase1 settings to control NPU encrypt/decrypt IPsec packets (enabled by default).

```
config vpn ipsec phase1/phase1-interface
    edit "vpn_name"
        set npu-offload enable/disable
    next
end
```

4. Check NPU offloading. The NPU encrypted/decrypted counter should tick. The `npu_flag 03` flag means that the traffic processed by the NPU is bi-directional.

```
# diagnose vpn tunnel list
list all ipsec tunnel in vd 0
----
name=test ver=2 serial=1 173.1.1.1:0->11.101.1.1:0 tun_id=11.101.1.1
bound_if=42 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/8 options[0008]=npu
proxyid_num=1 child_num=0 refcnt=14 ilast=2 olast=2 ad=/0
stat: rxp=12231 txp=12617 rxb=1316052 txb=674314
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=test proto=0 sa=1 ref=4 serial=7
  src: 0:0.0.0.0/0.0.0.0:0
  dst: 0:0.0.0.0/0.0.0.0:0
  SA:  ref=6 options=10626 type=00 soft=0 mtu=1438 expire=42921/0B replaywin=2048
       seqno=802 esn=0 replaywin_lastseq=00000680 itn=0
  life: type=01 bytes=0/0 timeout=42930/43200
  dec: spi=e313ac46 esp=aes key=16 0dcb52642eed18b852b5c65a7dc62958
       ah=md5 key=16 c61d9fe60242b9a30e60b1d01da77660
  enc: spi=706ffe03 esp=aes key=16 6ad98c204fa70545dbf3d2e33fb7b529
       ah=md5 key=16 dcc3b866da155ef73c0aba15ec530e2e
  dec:pkts/bytes=1665/16352, enc:pkts/bytes=2051/16826
  npu_flag=03 npu_rgwy=11.101.1.1 npu_lgwy=173.1.1.1 npu_selid=6 dec_npuid=2 enc_npuid=2

FGT_900D # diagnose vpn ipsec st
All ipsec crypto devices in use:
NP6_0:
    Encryption (encrypted/decrypted)
        null             : 0                1.
        des              : 0                1.
        3des             : 0                1.
```

```
        aes              : 0                1.
        aes-gcm          : 0                1.
        aria             : 0                1.
        seed             : 0                1.
        chacha20poly1305 : 0                1.
    Integrity (generated/validated)
        null             : 0                1.
        md5              : 0                1.
        sha1             : 0                1.
        sha256           : 0                1.
        sha384           : 0                1.
        sha512           : 0                1.

NP6_1:
    Encryption (encrypted/decrypted)
        null             : 14976            15357
        des              : 0                1.
        3des             : 0                1.
        aes              : 1664             2047
        aes-gcm          : 0                1.
        aria             : 0                1.
        seed             : 0                1.
        chacha20poly1305 : 0                1.
    Integrity (generated/validated)
        null             : 0                1.
        md5              : 1664             2047
        sha1             : 14976            15357
        sha256           : 0                1.
        sha384           : 0                1.
        sha512           : 0                1.

NPU Host Offloading:
    Encryption (encrypted/decrypted)
        null             : 3                1.
        des              : 0                1.
        3des             : 0                1.
        aes              : 3                1.
        aes-gcm          : 0                1.
        aria             : 0                1.
        seed             : 0                1.
        chacha20poly1305 : 0                1.
    Integrity (generated/validated)
        null             : 0                1.
        md5              : 3                1.
        sha1             : 3                1.
        sha256           : 0                1.
        sha384           : 0                1.
        sha512           : 0                1.

CP8:
    Encryption (encrypted/decrypted)
        null             : 1                1.
```

```
        des              : 0                1.
        3des             : 0                1.
        aes              : 1                1.
        aes-gcm          : 0                1.
        aria             : 0                1.
        seed             : 0                1.
        chacha20poly1305 : 0                1.
    Integrity (generated/validated)
        null             : 0                1.
        md5              : 1                1.
        sha1             : 1                1.
        sha256           : 0                1.
        sha384           : 0                1.
        sha512           : 0                1.

SOFTWARE:
    Encryption (encrypted/decrypted)
        null             : 0                1.
        des              : 0                1.
        3des             : 0                1.
        aes              : 0                1.
        aes-gcm          : 29882            29882
        aria             : 21688            21688
        seed             : 153774           153774
        chacha20poly1305 : 29521            29521
    Integrity (generated/validated)
        null             : 59403            59403
        md5              : 0                1.
        sha1             : 175462           175462
        sha256           : 0                1.
        sha384           : 0                1.
        sha512           : 0                1.
```

5. If traffic cannot be offloaded by the NPU, the CP will try to encrypt/decrypt the IPsec packets.

IPsec traffic processed by CP

1. Check the NPU flag and CP counter.

```
# diagnose vpn tunnel list
list all ipsec tunnel in vd 0
----
name=test ver=2 serial=1 173.1.1.1:0->11.101.1.1:0 tun_id=11.101.1.1
bound_if=42 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/0
proxyid_num=1 child_num=0 refcnt=13 ilast=0 olast=0 ad=/0
stat: rxp=8418 txp=8418 rxb=1251248 txb=685896
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=test proto=0 sa=1 ref=3 serial=7
  src: 0:0.0.0.0/0.0.0.0:0
  dst: 0:0.0.0.0/0.0.0.0:0
```

```
  SA:  ref=3 options=10226 type=00 soft=0 mtu=1438 expire=42037/0B replaywin=2048
       seqno=20e3 esn=0 replaywin_lastseq=000020e3 itn=0
  life: type=01 bytes=0/0 timeout=42928/43200
  dec: spi=e313ac48 esp=aes key=16 393770842f926266530db6e43e21c4f8
       ah=md5 key=16 b2e4e025e8910e95c1745e7855479cca
  enc: spi=706ffe05 esp=aes key=16 7ef749610335f9f50e252023926de29e
       ah=md5 key=16 0b81e4d835919ab2b8ba8edbd01aec9d
  dec:pkts/bytes=8418/685896, enc:pkts/bytes=8418/1251248
  npu_flag=00 npu_rgwy=11.101.1.1 npu_lgwy=173.1.1.1 npu_selid=6 dec_npuid=0 enc_npuid=0

FGT-D # diagnose vpn ipsec status
All ipsec crypto devices in use:
NP6_0:
    Encryption (encrypted/decrypted)
        null             : 0                1.
        des              : 0                1.
        3des             : 0                1.
        aes              : 0                1.
        aes-gcm          : 0                1.
        aria             : 0                1.
        seed             : 0                1.
        chacha20poly1305 : 0                1.
    Integrity (generated/validated)
        null             : 0                1.
        md5              : 0                1.
        sha1             : 0                1.
        sha256           : 0                1.
        sha384           : 0                1.
        sha512           : 0                1.

NP6_1:
    Encryption (encrypted/decrypted)
        null             : 14976            15357
        des              : 0                1.
        3des             : 0                1.
        aes              : 1664             2047
        aes-gcm          : 0                1.
        aria             : 0                1.
        seed             : 0                1.
        chacha20poly1305 : 0                1.
    Integrity (generated/validated)
        null             : 0                1.
        md5              : 1664             2047
        sha1             : 14976            15357
        sha256           : 0                1.
        sha384           : 0                1.
        sha512           : 0                1.

NPU Host Offloading:
    Encryption (encrypted/decrypted)
        null             : 3                1.
        des              : 0                1.
```

```
        3des             : 0                1.
        aes              : 3                1.
        aes-gcm          : 0                1.
        aria             : 0                1.
        seed             : 0                1.
        chacha20poly1305 : 0                1.
    Integrity (generated/validated)
        null             : 0                1.
        md5              : 3                1.
        sha1             : 3                1.
        sha256           : 0                1.
        sha384           : 0                1.
        sha512           : 0                1.

CP8:
    Encryption (encrypted/decrypted)
        null             : 1                1.
        des              : 0                1.
        3des             : 0                1.
        aes              : 8499             8499
        aes-gcm          : 0                1.
        aria             : 0                1.
        seed             : 0                1.
        chacha20poly1305 : 0                1.
    Integrity (generated/validated)
        null             : 0                1.
        md5              : 8499             8499
        sha1             : 1                1.
        sha256           : 0                1.
        sha384           : 0                1.
        sha512           : 0                1.

SOFTWARE:
    Encryption (encrypted/decrypted)
        null             : 0                1.
        des              : 0                1.
        3des             : 0                1.
        aes              : 0                1.
        aes-gcm          : 29882            29882
        aria             : 21688            21688
        seed             : 153774           153774
        chacha20poly1305 : 29521            29521
    Integrity (generated/validated)
        null             : 59403            59403
        md5              : 0                1.
        sha1             : 175462           175462
        sha256           : 0                1.
        sha384           : 0                1.
        sha512           : 0                1.
```

2. Two options are used to control if the CP processes packets. If disabled, packets are processed by the CPU.

```
config system global
    set ipsec-asic-offload disable
    set ipsec-hmac-offload disable
end
```

IPsec traffic processed by CPU

IPsec traffic might be processed by the CPU for the following reasons:

- Some entry-level models do not have NPUs.
- NPU offloading and CP IPsec traffic processing manually disabled.
- Some types of proposals - SEED, ARIA, chacha20poly1305 - are not supported by the NPU or CP.
- NPU flag set to 00 and software encrypt/decrypt counter ticked.

```
# diagnose vpn tunnel list
list all ipsec tunnel in vd 0
----
name=test ver=2 serial=1 173.1.1.1:0->11.101.1.1:0 tun_id=11.101.1.1
bound_if=42 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/0
proxyid_num=1 child_num=0 refcnt=14 ilast=0 olast=0 ad=/0
stat: rxp=12162 txp=12162 rxb=1691412 txb=1008216
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=test proto=0 sa=1 ref=4 serial=8
  src: 0:0.0.0.0/0.0.0.0:0
  dst: 0:0.0.0.0/0.0.0.0:0
  SA:  ref=3 options=10602 type=00 soft=0 mtu=1453 expire=42903/0B replaywin=2048
       seqno=2d70 esn=0 replaywin_lastseq=00002d70 itn=0
  life: type=01 bytes=0/0 timeout=42931/43200
  dec: spi=e313ac4d esp=chacha20poly1305 key=36
812d1178784c1130d1586606e44e1b9ab157e31a09edbed583be1e9cc82e8c9f2655a2cf
       ah=null key=0
  enc: spi=706ffe0a esp=chacha20poly1305 key=36
f2727e001e2243549b140f1614ae3df82243adb070e60c33911f461b389b05a7a642e11a
       ah=null key=0
  dec:pkts/bytes=11631/976356, enc:pkts/bytes=11631/1627692
  npu_flag=00 npu_rgwy=11.101.1.1 npu_lgwy=173.1.1.1 npu_selid=7 dec_npuid=0 enc_npuid=0

FGT_900D # diagnose vpn ipsec status
All ipsec crypto devices in use:
NP6_0:
    Encryption (encrypted/decrypted)
        null             : 0                1.
        des              : 0                1.
        3des             : 0                1.
        aes              : 0                1.
        aes-gcm          : 0                1.
        aria             : 0                1.
        seed             : 0                1.
        chacha20poly1305 : 0                1.
    Integrity (generated/validated)
```

```
        null             : 0                1.
        md5              : 0                1.
        sha1             : 0                1.
        sha256           : 0                1.
        sha384           : 0                1.
        sha512           : 0                1.

NP6_1:
    Encryption (encrypted/decrypted)
        null             : 14976            15357
        des              : 0                1.
        3des             : 0                1.
        aes              : 1664             2047
        aes-gcm          : 0                1.
        aria             : 0                1.
        seed             : 0                1.
        chacha20poly1305 : 0                1.
    Integrity (generated/validated)
        null             : 0                1.
        md5              : 1664             2047
        sha1             : 14976            15357
        sha256           : 0                1.
        sha384           : 0                1.
        sha512           : 0                1.

NPU Host Offloading:
    Encryption (encrypted/decrypted)
        null             : 3                1.
        des              : 0                1.
        3des             : 0                1.
        aes              : 3                1.
        aes-gcm          : 0                1.
        aria             : 0                1.
        seed             : 0                1.
        chacha20poly1305 : 0                1.
    Integrity (generated/validated)
        null             : 0                1.
        md5              : 3                1.
        sha1             : 3                1.
        sha256           : 0                1.
        sha384           : 0                1.
        sha512           : 0                1.

CP8:
    Encryption (encrypted/decrypted)
        null             : 1                1.
        des              : 0                1.
        3des             : 0                1.
        aes              : 8865             8865
        aes-gcm          : 0                1.
        aria             : 0                1.
        seed             : 0                1.
```

```
        chacha20poly1305 : 0                1.
    Integrity (generated/validated)
        null             : 0                1.
        md5              : 8865             8865
        sha1             : 1                1.
        sha256           : 0                1.
        sha384           : 0                1.
        sha512           : 0                1.

SOFTWARE:
    Encryption (encrypted/decrypted)
        null             : 0                1.
        des              : 0                1.
        3des             : 0                1.
        aes              : 531              531
        aes-gcm          : 29882            29882
        aria             : 21688            21688
        seed             : 153774           153774
        chacha20poly1305 : 41156            41156
    Integrity (generated/validated)
        null             : 71038            71038
        md5              : 531              531
        sha1             : 175462           175462
        sha256           : 0                1.
        sha384           : 0                1.
        sha512           : 0                1.
```

Disable automatic ASIC offloading

When `auto-asic-offload` is set to `disable` in the firewall policy, traffic is not offloaded and the NPU hosting counter is ticked.

```
# diagnose vpn ipsec status
All ipsec crypto devices in use:
NP6_0:
    Encryption (encrypted/decrypted)
        null             : 0                1.
        des              : 0                1.
        3des             : 0                1.
        aes              : 0                1.
        aes-gcm          : 0                1.
        aria             : 0                1.
        seed             : 0                1.
        chacha20poly1305 : 0                1.
    Integrity (generated/validated)
        null             : 0                1.
        md5              : 0                1.
        sha1             : 0                1.
        sha256           : 0                1.
        sha384           : 0                1.
        sha512           : 0                1.
```

```
NP6_1:
    Encryption (encrypted/decrypted)
        null             : 14976           15357
        des              : 0                1.
        3des             : 0                1.
        aes              : 110080          2175
        aes-gcm          : 0                1.
        aria             : 0                1.
        seed             : 0                1.
        chacha20poly1305 : 0                1.
    Integrity (generated/validated)
        null             : 0                1.
        md5              : 110080          2175
        sha1             : 14976           15357
        sha256           : 0                1.
        sha384           : 0                1.
        sha512           : 0                1.

NPU Host Offloading:
    Encryption (encrypted/decrypted)
        null             : 3                1.
        des              : 0                1.
        3des             : 0                1.
        aes              : 111090           1.
        aes-gcm          : 0                1.
        aria             : 0                1.
        seed             : 0                1.
        chacha20poly1305 : 0                1.
    Integrity (generated/validated)
        null             : 0                1.
        md5              : 111090           1.
        sha1             : 3                1.
        sha256           : 0                1.
        sha384           : 0                1.
        sha512           : 0                1.

CP8:
    Encryption (encrypted/decrypted)
        null             : 1                1.
        des              : 0                1.
        3des             : 0                1.
        aes              : 8865            8865
        aes-gcm          : 0                1.
        aria             : 0                1.
        seed             : 0                1.
        chacha20poly1305 : 0                1.
    Integrity (generated/validated)
        null             : 0                1.
        md5              : 8865            8865
        sha1             : 1                1.
        sha256           : 0                1.
```

```
        sha384           : 0                1.
        sha512           : 0                1.

SOFTWARE:
    Encryption (encrypted/decrypted)
        null             : 0                1.
        des              : 0                1.
        3des             : 0                1.
        aes              : 539              539
        aes-gcm          : 29882            29882
        aria             : 21688            21688
        seed             : 153774           153774
        chacha20poly1305 : 41259            41259
    Integrity (generated/validated)
        null             : 71141            71141
        md5              : 539              539
        sha1             : 175462           175462
        sha256           : 0                1.
        sha384           : 0                1.
        sha512           : 0                1.
```

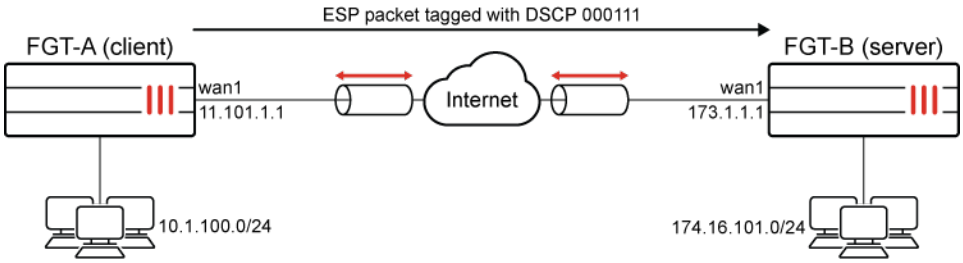
Configure DSCP for IPsec tunnels

Configuring the differentiated services (DiffServ) code in phase2 of an IPsec tunnel allows the tag to be applied to the outer IP header of the Encapsulating Security Payload (ESP) packet. This allows more control over the DSCP value of the ESP packet that can be used by the ISP for QoS controls.

This may be helpful when there is already a DSCP tag within the inner IP header of encapsulated payload. When the inner IP header has a DSCP tag:

- If `diffserv` is disabled in the IPsec phase2 configuration, then the ESP packets' DSCP value is copied from the DSCP value of the inner IP header.
- If `diffserv` is enabled in the IPsec phase2 configuration, then ESP packets' DSCP value is set to the configured value.

> Offloading traffic to the NPU must be disabled for the tunnel in order for the configured DSCP value to be applied to the outer IP header of the ESP packet. If NPU offloading is enabled, and diffserv is enabled on the IPsec phase2 settings, then the DSCP value is copied from the inner IP header to the outer IP header.



In this example, NPU offloading is disabled, diffserv is enabled, and the diffserv code is set to 000111 on FGT-A. ESP packets egressing from FGT-A will have the DSCP tag which can be used by the ISP for QoS purpose. Only one side of the tunnel needs to have diffserv enabled.

To configure IPsec on FGT-A:

1. Configure the phase1-interface:

```
config vpn ipsec phase1-interface
    edit "s2s"
        set interface "wan1"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set npu-offload disable
        set dhgrp 14 5
        set wizard-type static-fortigate
        set remote-gw 173.1.1.1
        set psksecret **********
    next
end
```

2. Configure the phase2-interface:

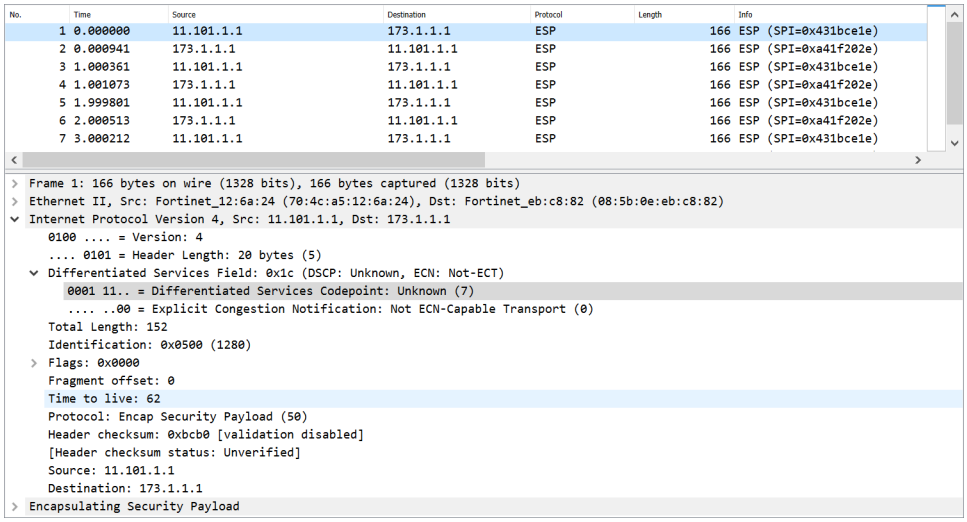
```
config vpn ipsec phase2-interface
    edit "s2s"
        set phase1name "s2s"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
        set dhgrp 14 5
        set diffserv enable
        set diffservcode 000111
        set src-addr-type name
        set dst-addr-type name
        set src-name "s2s_local"
        set dst-name "s2s_remote"
    next
end
```

3. Check the state of the IPsec tunnel:

```
FGT-A # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=s2s ver=1 serial=1 11.101.1.1:0->173.1.1.1:0 tun_id=173.1.1.1 dst_mtu=1500
bound_if=17 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/512 options[0200]=frag-rfc  run_
state=0 accept_traffic=1 overlay_id=0

proxyid_num=1 child_num=0 refcnt=11 ilast=12 olast=2978 ad=/0
stat: rxp=4 txp=4 rxb=608 txb=336
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=s2s proto=0 sa=1 ref=2 serial=2 dscp
```

```
  src: 0:10.1.100.0/255.255.255.0:0
  dst: 0:174.16.101.0/255.255.255.0:0
  SA:  ref=3 options=110226 type=00 soft=0 mtu=1438 expire=39916/0B replaywin=2048
       seqno=5 esn=0 replaywin_lastseq=00000005 itn=0 qat=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42899/43200
  dec: spi=a41f202e esp=aes key=16 8a02875b80b884d961af227fe8b5cdee
       ah=sha1 key=20 fc9760b79e79dbbeef630ec0c5dca74777976208
  enc: spi=431bce1e esp=aes key=16 851117af24212da89e466d8bea9632bb
       ah=sha1 key=20 0807cc0af2dc4ea049a6b1a4af410ccc71e2156d
  dec:pkts/bytes=4/336, enc:pkts/bytes=4/608
  npu_flag=00 npu_rgwy=173.1.1.1 npu_lgwy=11.101.1.1 npu_selid=1 dec_npuid=0 enc_npuid=0
run_tally=1
```

4. Use a packet analyzer, or sniffer, to check the ESP packets:



IPsec IKE load balancing based on FortiSASE account information

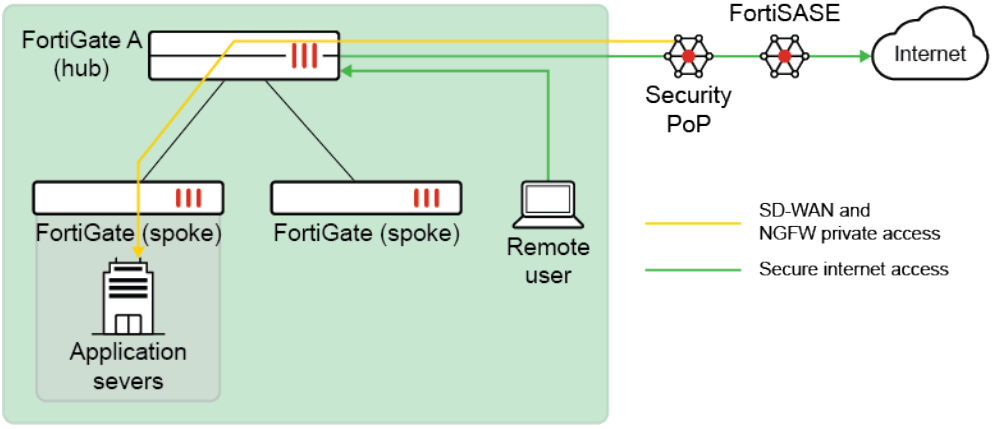
The FortiGate device ID is carried by the IKEv2 message NOTIFY payload when it is configured.

```
config vpn ipsec phase1-interface
    edit <name>
        set dev-id-notification enable
        set dev-id <string>
    next
end
```

This device ID configuration is required when the FortiGate is configured as a secure edge LAN extension for FortiSASE. It allows FortiSASE to distribute IKE/IPsec traffic according to the FortiGate device ID to achieve load balancing.

Example

In this example, a FortiGate SD-WAN is configured, which acts as a secure edge. FortiSASE ensures secure internet access for users in the local network behind the FortiGate and allows other FortiSASE remote users with secure private access to private resources behind the FortiGate.



To configure FortiGate A (FGT-A):

1. Configure the IPsec phase 1 settings:

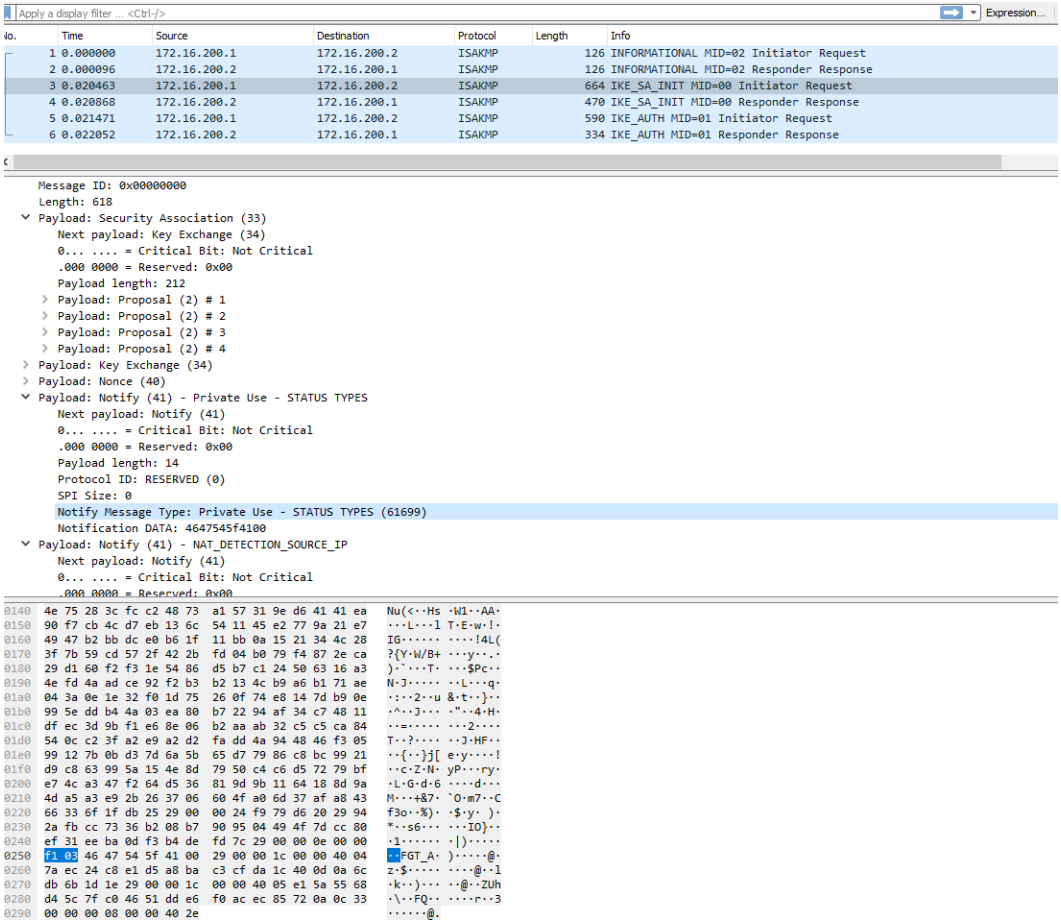
```
config vpn ipsec phase1-interface
    edit "ul-port1"
        set interface "port1"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set localid "peerid-UNshTWcLQ22UNWqk0UwYtCQNtVhujrxAdyMG0qRsGVkx9mM8ksdaRZOF"
        set dpd on-idle
        set comments "[FGCONN] Do NOT edit. Automatically generated by extension controller."
        set dev-id-notification enable
        set dev-id "FGT_A"
        set remote-gw 172.16.200.2
        set psksecret ********
    next
end
```

2. Verify that the IPsec tunnel is established:

```
# diagnose vpn tunnel list
list all ipsec tunnel in vd 3
------------------------------------------------------
name=ul-port1 ver=2 serial=3 172.16.200.1:0->172.16.200.2:0 tun_id=172.16.200.2 tun_
id6=::172.16.200.2 dst_mtu=1500 dpd-link=on weight=1
bound_if=19 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc  run_
state=0 role=primary accept_traffic=1 overlay_id=0
```

```
proxyid_num=1 child_num=0 refcnt=4 ilast=0 olast=0 ad=/0
stat: rxp=2689 txp=7115 rxb=278520 txb=617095
dpd: mode=on-idle on=1 idle=20000ms retry=3 count=0 seqno=1
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=ul-port1 proto=0 sa=1 ref=3 serial=1
  src: 0:10.252.0.2-10.252.0.2:0
  dst: 0:10.252.0.1-10.252.0.1:0
  SA:  ref=6 options=10226 type=00 soft=0 mtu=1438 expire=41281/0B replaywin=2048
       seqno=1bca esn=0 replaywin_lastseq=00000a80 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=42897/43200
  dec: spi=acf1f0fc esp=aes key=16 97d75ba10fbc904f14ce4a4caf8b4148
       ah=sha1 key=20 4ab706602068f9590314c4b16f53130a8011f410
  enc: spi=ca8de50b esp=aes key=16 8185ec9d2ecbb1d157663a6c199fc998
       ah=sha1 key=20 9430df55054152ab88e7372a322aad8f87688614
  dec:pkts/bytes=2690/278560, enc:pkts/bytes=14227/1632503
  npu_flag=03 npu_rgwy=172.16.200.2 npu_lgwy=172.16.200.1 npu_selid=2 dec_npuid=2 enc_npuid=2
run_tally=0
```

3. Perform a packet capture of IPsec traffic (Wireshark is used in this example) and locate the initiator request IKE packet's NOTIFY message (type 61699).



IPsec SA key retrieval from a KMS server using KMIP

In environments that require centralized management of cryptographic keys where no key derivations or algorithmic operations are allowed on edge devices (such as the FortiGate), they will deploy a Key Management Services (KMS) server cluster to generate and manage all cryptographic keys. Then, the Key Management Interoperability Protocol (KMIP) is used on the edge devices to locate the KMS server, create keys if they do not exist, and retrieve keys to be used for securing these edge devices.

FortiGates have a KMIP client that sends KMIP requests to locate the Key Management Services (KMS) server, creates keys if they do not exist on the KMS server, and retrieves keys from the KMS server to use as IPsec security association (SA) keys for IKEv2 only.

This feature allows the FortiGate to offload the task of generating IPsec SA keys to a KMS server, regardless of specific IPsec VPN topologies with a FortiGate, when the administrator has the requirement to centralize cryptographic keys management in a KMS server.

The FortiGate's integrated KMIP client also supports the following:

- If the KMS server is unavailable, then the FortiGate continues to use the previous keys to avoid a network blackout.
- ADVPN configurations for the hub and spoke, so that shortcuts between two spokes will use their own encryption keys retrieved from the KMS server.
- Multiple tunnels between the same tunnel endpoints using multiple VRFs.

To configure the KMIP server:

```
config vpn kmip-server
    edit <KMS_server_ID>
        config server-list
            edit <ID>
                set server <server_IP>
                set cert <string>
            next
        end
        set username <username_defined_on_KMS_server>
        set password <password>
    next
end
```

To apply the KMS server in the phase 1 interface settings:

```
config vpn ipsec phase1-interface
    edit <name>
        set kms <KMS_server_ID>
    next
end
```

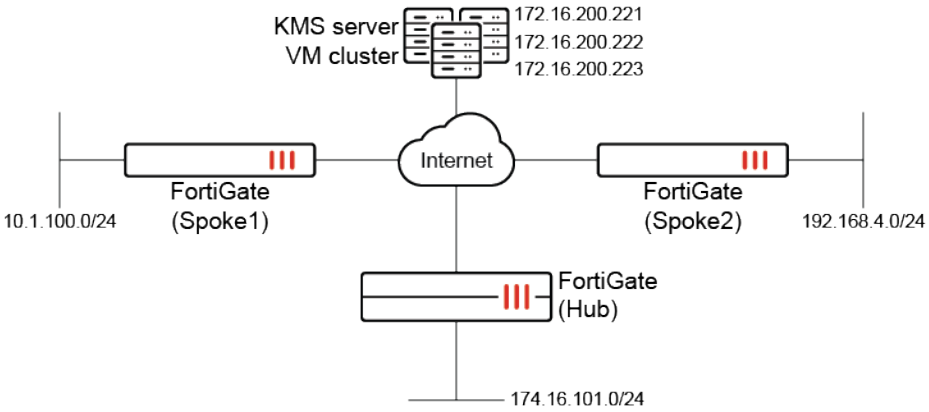
IPsec tunnels will not be established if a FortiGate VPN peer does not support KMS, or has not configured `kms <KMS_server_ID>` in `config vpn ipsec phase1-interface`.

The following diagnostic commands have been added:

- `get vpn ike kms-keys`
- `diagnose debug application kmipd -1`
- `execute kmip {create | destroy | get | locate | rekey} <parameter>`

Example

In this example, there is a topology with an ADVPN hub FortiGate and two spoke FortiGates. There is a cluster or three KMS server VMs (172.16.200.221, 172.16.200.222, and 172.16.200.223) that operates in round-robin mode. The `testuser1_Cert` certificate is issued by the KMS server, and the `testuser1` user is defined on the KMS server. Authentication to the KMS server by the KMIP client requires both a certificate and a password.



The Hub FortiGate acting as the responder will try to locate keys on the KMS server first. If they do not exist, the FortiGate requests to create new keys on KMS server. The responder sends the keys' names to the Spoke1 and Spoke2 FortiGates acting as the initiators using IKE messages, and these initiators locate and retrieve keys from KMS server using the keys' names. The `keylifeseconds` parameter in phase 2 defines how often the FortiGate will try to synchronize local keys to those on the KMS server.

The keys are retrieved from the KMS server and used as IPsec SA keys in IPsec tunnels. The key format used is: [IDi/r]-[IDr/i]-[phase2name]-ENC/AUTH-[keyalg]-[keylen].

First, this example focuses on the Hub FortiGate and the IPsec VPN connection between the Spoke1 and Hub FortiGate. Second, this example focuses on the spoke-to-spoke tunnel, also known as a shortcut tunnel or shortcut, which is established when traffic flows between the Spoke1 and Spoke2 FortiGates.

To configure IPsec SA key retrieval from a KMS server on the Hub FortiGate:

1. Configure the KMIP server:

```
config vpn kmip-server
    edit "KMS_server"
        config server-list
            edit 1
                set server "172.16.200.221"
                set cert "testuser1_Cert"
            next
            edit 2
```

```
                set server "172.16.200.222"
                set cert "testuser1_Cert"
            next
            edit 3
                set server "172.16.200.223"
                set cert "testuser1_Cert"
            next
        end
        set username "testuser1"
        set password **********
    next
end
```

2. Configure the IPsec VPN phase 1 settings:

```
config vpn ipsec phase1-interface
    edit "hub"
        set type dynamic
        set interface "port2"
        set ike-version 2
        set authmethod signature
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128gcm-prfsha256 aes256gcm-prfsha384
chacha20poly1305-prfsha256
        set add-route disable
        set dpd on-idle
        set auto-discovery-sender enable
        set kms "KMS_server"
        set certificate "Fortinet_Factory_Backup"
        set dpd-retryinterval 60
    next
end
```

This feature is only supported in IKEv2. The `localid` is required in the phase 1 settings when using the PSK authentication method.

3. Configure the IPsec VPN phase 2 settings:

```
config vpn ipsec phase2-interface
    edit "hub"
        set phase1name "hub"
        set proposal aes128-sha1 aes256-sha1 aes128-sha256 aes256-sha256 aes128gcm aes256gcm
chacha20poly1305
        set keylifeseconds 7200
    next
end
```

To verify the IPsec configuration and tunnel between the Hub and Spoke1 FortiGates:

1. Verify the tunnel state on the Hub:

```
Hub # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=hub ver=2 serial=1 172.16.200.4:0->0.0.0.0:0 tun_id=10.0.0.1 tun_id6=::10.0.0.1 dst_mtu=0
dpd-link=on weight=1
bound_if=10 lgwy=static/1 tun=intf mode=dialup/2 encap=none/552 options[0228]=npu frag-rfc
role=primary accept_traffic=1 overlay_id=0

proxyid_num=0 child_num=2 refcnt=4 ilast=42965007 olast=42965007 ad=/0
stat: rxp=980 txp=1980 rxb=125003 txb=123108
dpd: mode=on-idle on=0 idle=60000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
run_tally=0
------------------------------------------------------
name=hub_0 ver=2 serial=10 172.16.200.4:0->172.16.200.1:0 tun_id=10.10.10.2 tun_
id6=::10.0.0.16 dst_mtu=1500 dpd-link=on weight=1
bound_if=10 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/74408 options[122a8]=npu rgwy-
chg frag-rfc  run_state=0 role=primary accept_traffic=1 overlay_id=0

parent=hub index=0
proxyid_num=1 child_num=0 refcnt=5 ilast=6 olast=6 ad=s/1
stat: rxp=21 txp=39 rxb=2644 txb=2389
dpd: mode=on-idle on=1 idle=60000ms retry=3 count=0 seqno=1
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=hub proto=0 sa=1 ref=3 serial=1 ads
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=6 options=826 type=00 soft=0 mtu=1438 expire=6673/0B replaywin=2048
       seqno=15 esn=0 replaywin_lastseq=00000002 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=7191/7200
  dec: spi=628d1814 esp=aes key=16 5dad0d8d3568eab7c3f259349dc64039
       ah=sha1 key=20 e660f491b80b2cfdcdb0d737942bea2e853dac8d
  enc: spi=471dfe2e esp=aes key=16 1de4b8e8accaa792e0934fbd9f933a6a
       ah=sha1 key=20 1fa244d3971b4d4df59b8d7b3655a1b77f8e65af
  dec:pkts/bytes=22/2696, enc:pkts/bytes=59/4949
  npu_flag=03 npu_rgwy=172.16.200.1 npu_lgwy=172.16.200.4 npu_selid=e dec_npuid=1 enc_npuid=0
------------------------------------------------------
name=hub_1 ver=2 serial=f 172.16.200.4:0->172.16.200.3:0 tun_id=10.10.10.3 tun_id6=::10.0.0.15
dst_mtu=1500 dpd-link=on weight=1
bound_if=10 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/74408 options[122a8]=npu rgwy-
chg frag-rfc  run_state=0 role=primary accept_traffic=1 overlay_id=0

parent=hub index=1
proxyid_num=1 child_num=0 refcnt=5 ilast=2 olast=2 ad=s/1
stat: rxp=21 txp=43 rxb=2615 txb=2718
dpd: mode=on-idle on=1 idle=60000ms retry=3 count=0 seqno=1
natt: mode=none draft=0 interval=0 remote_port=0
```

```
fec: egress=0 ingress=0
proxyid=hub proto=0 sa=1 ref=3 serial=1 ads
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=6 options=826 type=00 soft=0 mtu=1438 expire=6665/0B replaywin=2048
       seqno=17 esn=0 replaywin_lastseq=00000002 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=7187/7200
  dec: spi=628d1813 esp=aes key=16 5fcca9194ced21b0a586a8fd7a27cbf7
       ah=sha1 key=20 6d6d9dc77d5af89f062927c4d4695d404df1ffe3
  enc: spi=8d568113 esp=aes key=16 2006f323b760238048fcd6f7783b0a04
       ah=sha1 key=20 bd6db68ee035088f35174b2b5c58a51fbbe3f5b5
  dec:pkts/bytes=22/2686, enc:pkts/bytes=65/5566
  npu_flag=03 npu_rgwy=172.16.200.3 npu_lgwy=172.16.200.4 npu_selid=d dec_npuid=1 enc_npuid=0
```

2. Verify the KMS keys for the VPN tunnel between the Hub and Spoke1:

```
Hub # get vpn ike  kms-keys

vd: root/0
name: hub_1
addr: 172.16.200.4:500 -> 172.16.200.3:500

  phase2
  name: hub
  server: "KMS_server"
  spi: 628d1813
    enc
      keyname: "Spoke2-hub-hub-ENC-AES-16"
      key: 5fcca9194ced21b0a586a8fd7a27cbf7
    auth
      keyname: "Spoke2-hub-hub-AUTH-SHA1-20"
      key: 6d6d9dc77d5af89f062927c4d4695d404df1ffe3
  spi: 8d568113
    enc
      keyname: "hub-Spoke2-hub-ENC-AES-16"
      key: 2006f323b760238048fcd6f7783b0a04
    auth
      keyname: "hub-Spoke2-hub-AUTH-SHA1-20"
      key: bd6db68ee035088f35174b2b5c58a51fbbe3f5b5


vd: root/0
name: hub_0
addr: 172.16.200.4:500 -> 172.16.200.1:500

  phase2
  name: hub
  server: "KMS_server"
  spi: 628d1814
    enc
      keyname: "Spoke1-hub-hub-ENC-AES-16"
      key: 5dad0d8d3568eab7c3f259349dc64039
```

```
    auth
      keyname: "Spoke1-hub-hub-AUTH-SHA1-20"
      key: e660f491b80b2cfdcdb0d737942bea2e853dac8d
  spi: 471dfe2e
    enc
      keyname: "hub-Spoke1-hub-ENC-AES-16"
      key: 1de4b8e8accaa792e0934fbd9f933a6a
    auth
      keyname: "hub-Spoke1-hub-AUTH-SHA1-20"
      key: 1fa244d3971b4d4df59b8d7b3655a1b77f8e65af
```

3. Verify the IKE and KMIP debug messages on both FortiGates:

```
# diagnose debug application ike -1
# diagnose debug application kmipd -1
```

   a. For the responder FortiGate, Hub:

```
ike 0: comes 172.16.200.1:500->172.16.200.4:500,ifindex=10,vrf=0....
ike 0: IKEv2 exchange=AUTH id=6e99ee7fd238b462/82e575f08b93f44c:00000001 len=708
ike 0:hub:537: encrypted fragment 3 of 3 queued
ike 0:hub:537: reassembled fragmented message
ike 0:hub:537: responder received AUTH msg
ike 0:hub:537: processing notify type INITIAL_CONTACT
ike 0:hub:537: processing notify type INTERFACE_ADDR4
ike 0:hub:537: INTERFACE-ADDR4 10.10.10.2
ike 0:hub:537: processing notify type MESSAGE_ID_SYNC_SUPPORTED
ike 0:hub:537: processing notify type KMS_SUPPORT
...
ike 0:hub:hub: sending kmip locate request: id=4321 keyname=Spoke1-hub-hub-ENC-AES-16
ike 0:hub:hub: sending kmip locate request: id=4322 keyname=hub-Spoke1-hub-ENC-AES-16
ike 0:hub:hub: sending kmip locate request: id=4323 keyname=Spoke1-hub-hub-AUTH-SHA1-20
ike 0:hub:hub: sending kmip locate request: id=4324 keyname=hub-Spoke1-hub-AUTH-SHA1-20
...
ike 0:hub:hub: sending kmip create request: id=4328 keyname=hub-Spoke1-hub-AUTH-SHA1-20
keyalg=7 keylen=160
kmip_tsk_resp_finalizer()-365: server-KMS_server, vfid-0, cur_total-4, batch_count-4
kmip_free_tsk()-144: Freeing tsk pid=6487, job_id=4321, seq=4321
kmip_free_tsk()-144: Freeing tsk pid=6487, job_id=4322, seq=4322
kmip_free_tsk()-144: Freeing tsk pid=6487, job_id=4323, seq=4323
kmip_free_tsk()-144: Freeing tsk pid=6487, job_id=4324, seq=4324
...
kmipd_op_create_req_check()-35: New tsk for 'KMS_server', op-create, vfid-0, pid-6487,
job_id-4325, name-'Spoke1-hub-hub-ENC-AES-16'
kmip_new_tsk()-131: New tsk pid=6487, job_id=4325, seq=4325
...
kmipd_op_create_req_check()-35: New tsk for 'KMS_server', op-create, vfid-0, pid-6487,
job_id-4326, name-'hub-Spoke1-hub-ENC-AES-16'
kmip_new_tsk()-131: New tsk pid=6487, job_id=4326, seq=4326
...
kmipd_op_create_req_check()-35: New tsk for 'KMS_server', op-create, vfid-0, pid-6487,
job_id-4327, name-'Spoke1-hub-hub-AUTH-SHA1-20'
```

```
kmip_new_tsk()-131: New tsk pid=6487, job_id=4327, seq=4327
...
kmipd_op_create_req_check()-35: New tsk for 'KMS_server', op-create, vfid-0, pid-6487,
job_id-4328, name-'hub-Spoke1-hub-AUTH-SHA1-20'
kmip_new_tsk()-131: New tsk pid=6487, job_id=4328, seq=4328
...
kmip_send_reply()-32: Sending 28 data. Job_id-4332 ret-0
ike KMIP response received: id=4332 ret=0
ike 0:hub:hub processing kmip get-response
ike 0:hub:hub recevied KMS keys 4/4
...
ike 0:hub: adding new dynamic tunnel for 172.16.200.1:500
ike 0:hub_0: tunnel created tun_id 10.10.10.2/::10.0.0.12 remote_location 0.0.0.0
ike 0:hub_0: added new dynamic tunnel for 172.16.200.1:500
ike 0:hub_0:539: established IKE SA 709d9a9eab5b5a48/01afbbcfa47c1459
ike 0:hub_0:539: auto-discovery sender
ike 0:hub_0:539: auto-discovery 1
ike 0:hub_0:539: check peer route: if_addr4_rcvd=1, if_addr6_rcvd=0, mode_cfg=0
ike 0:hub_0:539: update peer route 0.0.0.0 -> 10.10.10.2
ike 0:hub_0:539: processing INITIAL-CONTACT
ike 0:hub_0: flushing
ike 0:hub_0: flushed
ike 0:hub_0:539: processed INITIAL-CONTACT
ike 0:hub_0:539: local cert, subject='hub', issuer='support'
ike 0:hub_0:539: local CA cert, subject='support', issuer='support'
ike 0:hub_0:539: add INTERFACE-ADDR4 10.10.10.1
ike 0:hub_0:hub: added KMS_KEY payloads
ike 0:hub_0:539:hub:1085: replay protection enabled
ike 0:hub_0:539:hub:1085: set sa life soft seconds=7190.
ike 0:hub_0:539:hub:1085: set sa life hard seconds=7200.
ike 0:hub_0:539:hub:1085: IPsec SA selectors #src=1 #dst=1
ike 0:hub_0:539:hub:1085: src 0 7 0:0.0.0.0-255.255.255.255:0
ike 0:hub_0:539:hub:1085: dst 0 7 0:0.0.0.0-255.255.255.255:0
ike 0:hub_0:539:hub:1085: add dynamic IPsec SA selectors
ike 0:hub_0:539:hub:1085: added dynamic IPsec SA proxyids, new serial 1
ike 0:hub_0:539:hub:1085: tunnel 2 of VDOM limit 0/0
ike 0:hub_0:539:hub:1085: add IPsec SA: SPIs=628d180e/471dfe29
ike 0:hub_0:539:hub:1085: IPsec SA dec spi 628d180e key
16:5DAD0D8D3568EAB7C3F259349DC64039 auth 20:E660F491B80B2CFDCDB0D737942BEA2E853DAC8D
ike 0:hub_0:539:hub:1085: IPsec SA enc spi 471dfe29 key
16:1DE4B8E8ACCAA792E0934FBD9F933A6A auth 20:1FA244D3971B4D4DF59B8D7B3655A1B77F8E65AF
ike 0:hub_0:539:hub:1085: added IPsec SA: SPIs=628d180e/471dfe29
ike 0:hub_0: tunnel up event
ike 0:hub_0:539:hub:1085: sending SNMP tunnel UP trap
```

b. For the initiator FortiGate, Spoke1:

```
ike 0:spoke1: schedule auto-negotiate
ike 0:spoke1:spoke1: initiator received KMS_KEY: "Spoke1-hub-hub-ENC-AES-16" "hub-Spoke1-
hub-ENC-AES-16" "Spoke1-hub-hub-AUTH-SHA1-20" "hub-Spoke1-hub-AUTH-SHA1-20"
...
ike 0:spoke1:spoke1: sending kmip locate request: id=77 keyname=Spoke1-hub-hub-ENC-AES-16
```

```
ike 0:spoke1:spoke1: sending kmip locate request: id=78 keyname=hub-Spoke1-hub-ENC-AES-16
ike 0:spoke1:spoke1: sending kmip locate request: id=79 keyname=Spoke1-hub-hub-AUTH-SHA1-
20
ike 0:spoke1:spoke1: sending kmip locate request: id=80 keyname=hub-Spoke1-hub-AUTH-SHA1-
20
...
kmipd_op_locate_req_check()-48: New tsk for 'KMS_server', op-locate, vfid-0, pid-3341,
job_id-78, name-'hub-Spoke1-hub-ENC-AES-16'
kmip_new_tsk()-131: New tsk pid=3341, job_id=78, seq=78
...
kmipd_op_locate_req_check()-48: New tsk for 'KMS_server', op-locate, vfid-0, pid-3341,
job_id-79, name-'Spoke1-hub-hub-AUTH-SHA1-20'
kmip_new_tsk()-131: New tsk pid=3341, job_id=79, seq=79
...
kmipd_op_locate_req_check()-48: New tsk for 'KMS_server', op-locate, vfid-0, pid-3341,
job_id-80, name-'hub-Spoke1-hub-AUTH-SHA1-20'
kmip_new_tsk()-131: New tsk pid=3341, job_id=80, seq=80
...
kmipd_op_locate_req_check()-48: New tsk for 'KMS_server', op-locate, vfid-0, pid-3341,
job_id-77, name-'Spoke1-hub-hub-ENC-AES-16'
kmip_new_tsk()-131: New tsk pid=3341, job_id=77, seq=77
...
kmipd_op_get_req_check()-35: New tsk for 'KMS_server', op-get, vfid-0, pid-3341, job_id-
81, keyid-'a98f50b20bfe4037a7c47283eef578e61b474bf3829f45beb4a6c972c31a5d63'
kmip_new_tsk()-131: New tsk pid=3341, job_id=81, seq=81
...
kmipd_op_get_req_check()-35: New tsk for 'KMS_server', op-get, vfid-0, pid-3341, job_id-
82, keyid-'b4867ef7052b484faea2e7916b585bfc171e0981b843444097ee39d67fba30ea'
kmip_new_tsk()-131: New tsk pid=3341, job_id=82, seq=82
...
kmipd_op_get_req_check()-35: New tsk for 'KMS_server', op-get, vfid-0, pid-3341, job_id-
83, keyid-'41d4e37c4a014811a78cd1e1053d6370edc62a5a975e46c8a8aeda3bf4d76061'
kmip_new_tsk()-131: New tsk pid=3341, job_id=83, seq=83
...
kmipd_op_get_req_check()-35: New tsk for 'KMS_server', op-get, vfid-0, pid-3341, job_id-
84, keyid-'2ba130bff7174ba7a237d7ea53611121383b132cf18a4fd183890ca196296cb4'
kmip_new_tsk()-131: New tsk pid=3341, job_id=84, seq=84
...
ike 0:spoke1:spoke1 processing kmip get-response
ike 0:spoke1:spoke1 recevied KMS keys 4/4
ike 0:spoke1:536:spoke1:549: replay protection enabled
ike 0:spoke1:536:spoke1:549: set sa life soft seconds=6901.
ike 0:spoke1:536:spoke1:549: set sa life hard seconds=7200.
ike 0:spoke1:536:spoke1:549: IPsec SA selectors #src=1 #dst=1
ike 0:spoke1:536:spoke1:549: src 0 7 0:0.0.0.0-255.255.255.255:0
ike 0:spoke1:536:spoke1:549: dst 0 7 0:0.0.0.0-255.255.255.255:0
ike 0:spoke1:536:spoke1:549: add IPsec SA: SPIs=471dfe29/628d180e
ike 0:spoke1:536:spoke1:549: IPsec SA dec spi 471dfe29 key
16:1DE4B8E8ACCAA792E0934FBD9F933A6A auth 20:1FA244D3971B4D4DF59B8D7B3655A1B77F8E65AF
ike 0:spoke1:536:spoke1:549: IPsec SA enc spi 628d180e key
16:5DAD0D8D3568EAB7C3F259349DC64039 auth 20:E660F491B80B2CFDCDB0D737942BEA2E853DAC8D
```

```
ike 0:spoke1:536:spoke1:549: added IPsec SA: SPIs=471dfe29/628d180e
ike 0:spoke1:536:spoke1:549: sending SNMP tunnel UP trap
```

To verify the IPsec configuration and tunnel between the Spoke1 and Spoke2 FortiGates:

1. Verify the tunnel state on Spoke1:

```
Spoke1 # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=spoke1 ver=2 serial=1 172.16.200.1:0->172.16.200.4:0 tun_id=172.16.200.4 tun_
id6=::172.16.200.4 dst_mtu=1500 dpd-link=on weight=1
bound_if=19 lgwy=static/1 tun=intf mode=auto/1 encap=none/552 options[0228]=npu frag-rfc  run_
state=0 role=primary accept_traffic=1 overlay_id=0 proxyid_num=1 child_num=1 refcnt=5 ilast=35
olast=35 ad=r/2
stat: rxp=1 txp=11 rxb=71 txb=699
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=5
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=spoke1 proto=0 sa=1 ref=3 serial=2 adr
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=6 options=12026 type=00 soft=0 mtu=1438 expire=6621/0B replaywin=2048
       seqno=c esn=0 replaywin_lastseq=00000002 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=6903/7200
  dec: spi=471dfe2e esp=aes key=16 1de4b8e8accaa792e0934fbd9f933a6a
       ah=sha1 key=20 1fa244d3971b4d4df59b8d7b3655a1b77f8e65af
  enc: spi=628d1814 esp=aes key=16 5dad0d8d3568eab7c3f259349dc64039
       ah=sha1 key=20 e660f491b80b2cfdcdb0d737942bea2e853dac8d
  dec:pkts/bytes=2/142, enc:pkts/bytes=22/2131
  npu_flag=03 npu_rgwy=172.16.200.4 npu_lgwy=172.16.200.1 npu_selid=1 dec_npuid=2 enc_npuid=2
run_tally=0
------------------------------------------------------
name=spoke1_0 ver=2 serial=4 172.16.200.1:0->172.16.200.3:0 tun_id=172.16.200.3 tun_
id6=::172.16.200.3 dst_mtu=1500 dpd-link=on weight=1
bound_if=19 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/66216 options[102a8]=npu rgwy-
chg frag-rfc  run_state=0 role=primary accept_traffic=1 overlay_id=0 parent=spoke1 index=0
proxyid_num=1 child_num=0 refcnt=5 ilast=10 olast=10 ad=r/2
stat: rxp=1 txp=5 rxb=84 txb=420
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=1
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=spoke1 proto=0 sa=1 ref=3 serial=1 adr
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:0.0.0.0-255.255.255.255:0
  SA:  ref=6 options=12026 type=00 soft=0 mtu=1438 expire=6947/0B replaywin=2048
       seqno=6 esn=0 replaywin_lastseq=00000402 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=7190/7200
  dec: spi=471dfe2f esp=aes key=16 a6d6a25cd986860bcc502d58f32e99de
       ah=sha1 key=20 07d712156eaca28439fbe944e3a8c9af4c45166a
  enc: spi=8d568114 esp=aes key=16 b01c534b11792b856c1b95c78c4cad91
       ah=sha1 key=20 fe6a82177db6911b3203d1306969e5ddec8fd039
```

```
  dec:pkts/bytes=2/168, enc:pkts/bytes=10/1180
  npu_flag=03 npu_rgwy=172.16.200.3 npu_lgwy=172.16.200.1 npu_selid=4 dec_npuid=2 enc_npuid=2
```

2. Verify the KMS keys for the VPN tunnel between Spoke1 and Spoke2:

```
Spoke1 # get vpn ike kms-keys

vd: root/0
name: spoke1
addr: 172.16.200.1:500 -> 172.16.200.4:500

  phase2
  name: spoke1
  server: "KMS_server"
  spi: 628d1814
    enc
      keyname: "Spoke1-hub-hub-ENC-AES-16"
      key: 5dad0d8d3568eab7c3f259349dc64039
    auth
      keyname: "Spoke1-hub-hub-AUTH-SHA1-20"
      key: e660f491b80b2cfdcdb0d737942bea2e853dac8d
  spi: 471dfe2e
    enc
      keyname: "hub-Spoke1-hub-ENC-AES-16"
      key: 1de4b8e8accaa792e0934fbd9f933a6a
    auth
      keyname: "hub-Spoke1-hub-AUTH-SHA1-20"
      key: 1fa244d3971b4d4df59b8d7b3655a1b77f8e65af


vd: root/0
name: spoke1_0
addr: 172.16.200.1:500 -> 172.16.200.3:500

  phase2
  name: spoke1
  server: "KMS_server"
  spi: 8d568114
    enc
      keyname: "Spoke1-Spoke2-spoke2-ENC-AES-16"
      key: b01c534b11792b856c1b95c78c4cad91
    auth
      keyname: "Spoke1-Spoke2-spoke2-AUTH-SHA1-20"
      key: fe6a82177db6911b3203d1306969e5ddec8fd039
  spi: 471dfe2f
    enc
      keyname: "Spoke2-Spoke1-spoke2-ENC-AES-16"
      key: a6d6a25cd986860bcc502d58f32e99de
    auth
      keyname: "Spoke2-Spoke1-spoke2-AUTH-SHA1-20"
      key: 07d712156eaca28439fbe944e3a8c9af4c45166a
```

3. Verify the FortiGate (KMIP client) connection to the KMS server:

```
Spoke1 # execute kmip locate KMS_server hub-Spoke1-hub-AUTH-SHA1-20
Locating key 'hub-Spoke1-hub-AUTH-SHA1-20', jobid=1935521133
Ret=0, jobid=1935521133
        Key ID: 2ba130bff7174ba7a237d7ea53611121383b132cf18a4fd183890ca196296cb4
```

4. Verify the IKE and KMIP debug messages on Spoke1 to confirm that when the KMS server is down during IPsec rekey, IPsec tunnel does not go down:

```
Spoke1 # diagnose debug application ike -1
Spoke1 # diagnose debug application kmipd -1

ike 0:spoke1:543:580 rekey in progress for SPI 471dfe32
ike 0:spoke1:543: sent IKE msg (CREATE_CHILD): 172.16.200.1:500->172.16.200.4:500, len=416,
vrf=0, id=627aee1c2562d5e5/31d6fccbac9dae7b:00000003
ike 0:spoke1:543: sent IKE msg (RETRANSMIT_CREATE_CHILD): 172.16.200.1:500->172.16.200.4:500,
len=416, vrf=0, id=627aee1c2562d5e5/31d6fccbac9dae7b:00000003
ike 0:spoke1:543: sent IKE msg (RETRANSMIT_CREATE_CHILD): 172.16.200.1:500->172.16.200.4:500,
len=416, vrf=0, id=627aee1c2562d5e5/31d6fccbac9dae7b:00000003
ike 0: comes 172.16.200.4:500->172.16.200.1:500,ifindex=19,vrf=0....
ike 0: IKEv2 exchange=CREATE_CHILD_RESPONSE id=627aee1c2562d5e5/31d6fccbac9dae7b:00000003
len=192
ike 0:spoke1:543: received create-child response
ike 0:spoke1:543: initiator received CREATE_CHILD msg
ike 0:spoke1:543:spoke1:580: found child SA SPI 471dfe34 state=3
ike 0:spoke1:543: processing notify type KMS_KEYS_REUSE
...
ike 0:spoke1:543:spoke1:580: IPsec SA dec spi 471dfe34 key 16:1DE4B8E8ACCAA792E0934FBD9F933A6A
auth 20:1FA244D3971B4D4DF59B8D7B3655A1B77F8E65AF
ike 0:spoke1:543:spoke1:580: IPsec SA enc spi 628d181b key 16:5DAD0D8D3568EAB7C3F259349DC64039
auth 20:E660F491B80B2CFDCDB0D737942BEA2E853DAC8D
ike 0:spoke1:543:spoke1:580: added IPsec SA: SPIs=471dfe34/628d181b
ike 0:spoke1:543:spoke1:580: scheduling rekeyed SPI 471dfe32 for deletion
ike 0:spoke1:543:spoke1:580: rekey in progress, old SPI 471dfe32
...
ike 0:spoke1_0:spoke1: sending kmip locate request: id=166 keyname=FGT80FTK22056585-
FG200E4Q17904575-spoke2-ENC-AES-16
ike 0:spoke1_0:spoke1: sending kmip locate request: id=167 keyname=FG200E4Q17904575-
FGT80FTK22056585-spoke2-AUTH-SHA1-20
ike 0:spoke1_0:spoke1: sending kmip locate request: id=168 keyname=FGT80FTK22056585-
FG200E4Q17904575-spoke2-AUTH-SHA1-20
...
__kmip_conn_connect()-489: Failed to connect KMIP server 'KMS_server', vfid-0, addr-
172.16.200.221:5696
...
__kmip_conn_connect()-489: Failed to connect KMIP server 'KMS_server', vfid-0, addr-
172.16.200.222:5696
...
__kmip_conn_connect()-489: Failed to connect KMIP server 'KMS_server', vfid-0, addr-
172.16.200.223:5696
__kmip_conn_pick_one_addr()-212: No more host to try.
__kmip_conn_schedule_next_retry()-169: server-KMS_server, st=0, vfid-0
ike 0:spoke1_0:spoke1: kmip req expired: id=165
ike 0:spoke1_0:544:spoke1:581: KMS: rekey using old child_sa keys.
```

```
ike 0:spoke1: schedule auto-negotiate
ike 0:spoke1_0:544:spoke1:581: replay protection enabled
ike 0:spoke1_0:544:spoke1:581: set sa life soft seconds=111.
ike 0:spoke1_0:544:spoke1:581: set sa life hard seconds=120.
ike 0:spoke1_0:544:spoke1:581: IPsec SA selectors #src=1 #dst=1
ike 0:spoke1_0:544:spoke1:581: src 0 4 0:0.0.0.0/0.0.0.0:0
ike 0:spoke1_0:544:spoke1:581: dst 0 4 0:0.0.0.0/0.0.0.0:0
ike 0:spoke1_0:544:spoke1:581: add dynamic IPsec SA selectors
ike 0:spoke1_0:544:spoke1:581: added dynamic IPsec SA proxyids, existing serial 1
ike 0:spoke1_0:544:spoke1:581: add IPsec SA: SPIs=471dfe35/8d56811c
```

Multiple interface monitoring for IPsec

IPsec can monitor multiple interfaces per tunnel, and activate a backup link only when all of the primary links are down. This can be useful if you have multiple WAN links and want to optimize your WAN link selection and performance while limiting the use of more expensive and bandwidth intensive interfaces, like 5G or LTE.

In cases where multiple primary overlays are deployed and the backup overlay is on an LTE connection, avoiding IPsec keep alive messages, BGP hellos, and SD-WAN health checks on the backup connection is required when the primary overlays are working. The backup overlay can monitor all of the primary overlays, and is not activated until the number of unhealthy primary overlays equals or surpasses the predefined threshold.

```
config vpn ipsec phase1-interface
    edit <phase-1 name>
        set monitor <overlay> <overlay> ... <overlay>
        set monitor-min <integer>
    next
end
```

`monitor`	The IPsec interfaces to monitor.
`monitor-min`	The minimum number of monitored interfaces that must become degraded before this interface is activated (0 = all interfaces, default = 0).

In this example, four primary overlays are configured, T1 - T4, on fixed broadband connections and one backup overlay, T5, is configured on an LTE connection.

The backup overlay stays down as long as the primary overlays are working normally. When all four of the primary overlays go down, the backup overlay is activated and used to forward traffic. If any of the primary overlays recover, then the backup overlay goes down.

SD-WAN can also be configured to steer traffic.

To configure the overlays:

1. Configure the VPN remote gateways:

```
config vpn ipsec phase1-interface
    edit "T1"
        set interface "dmz"
```

```
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256
        set remote-gw 172.16.208.2
        set psksecret **********
    next
    edit "T2"
        set interface "agg1"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256
        set remote-gw 172.16.203.2
        set psksecret **********
    next
    edit "T3"
        set interface "vlan100"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256
        set remote-gw 172.16.206.2
        set psksecret **********
    next
    edit "T4"
        set interface "port15"
        set ike-version 2
        set peertype any
        set net-device disable
        set proposal aes128-sha256
        set remote-gw 172.16.209.2
        set psksecret **********
    next
    edit "T5"
        set interface "vlan200"
        set ike-version 2
        set peertype any
        set monitor "T1" "T2" "T3" "T4"
        set monitor-min 4
        set net-device disable
        set proposal aes128-sha256
        set remote-gw 172.16.210.2
        set psksecret **********
    next
end
```

2. Configure the VPN tunnels:

```
config vpn ipsec phase2-interface
    edit "T1_P2"
        set phase1name "T1"
```

```
        set proposal aes256-sha256
        set auto-negotiate enable
    next
    edit "T2_P2"
        set phase1name "T2"
        set proposal aes256-sha256
        set auto-negotiate enable
    next
    edit "T3_P2"
        set phase1name "T3"
        set proposal aes256-sha256
        set auto-negotiate enable
    next
    edit "T4_P2"
        set phase1name "T4"
        set proposal aes256-sha256
        set auto-negotiate enable
    next
    edit "T5_P2"
        set phase1name "T5"
        set proposal aes256-sha256
        set auto-negotiate enable
    next
end
```

3. Configure the interfaces:

```
config system interface
    edit "T1"
        set vdom "root"
        set ip 100.1.1.1 255.255.255.255
        set allowaccess ping
        set type tunnel
        set remote-ip 100.1.1.2 255.255.255.0
        set snmp-index 113
        set interface "dmz"
    next
    edit "T2"
        set vdom "root"
        set ip 100.1.2.1 255.255.255.255
        set allowaccess ping
        set type tunnel
        set remote-ip 100.1.2.2 255.255.255.0
        set snmp-index 114
        set interface "agg1"
    next
    edit "T3"
        set vdom "root"
        set ip 100.1.3.1 255.255.255.255
        set allowaccess ping
        set type tunnel
        set remote-ip 100.1.3.2 255.255.255.0
```

```
        set snmp-index 115
        set interface "vlan100"
    next
    edit "T4"
        set vdom "root"
        set ip 100.1.4.1 255.255.255.255
        set allowaccess ping
        set type tunnel
        set remote-ip 100.1.4.2 255.255.255.0
        set snmp-index 65
        set interface "port15"
    next
    edit "T5"
        set vdom "root"
        set ip 100.1.5.1 255.255.255.255
        set allowaccess ping
        set type tunnel
        set remote-ip 100.1.5.2 255.255.255.0
        set snmp-index 117
        set interface "vlan200"
    next
end
```

4. Check the IPsec tunnel summary:

```
# get vpn ipsec tunnel summary
'T2' 172.16.203.2:0  selectors(total,up): 1/1  rx(pkt,err): 0/0  tx(pkt,err): 0/4
'T3' 172.16.206.2:0  selectors(total,up): 1/1  rx(pkt,err): 0/0  tx(pkt,err): 0/4
'T4' 172.16.209.2:0  selectors(total,up): 1/1  rx(pkt,err): 0/0  tx(pkt,err): 0/4
'T5' 172.16.210.2:0  selectors(total,up): 1/0  rx(pkt,err): 0/0  tx(pkt,err): 0/4
'T1' 172.16.208.2:0  selectors(total,up): 1/1  rx(pkt,err): 0/0  tx(pkt,err): 0/4
```

The backup overlay, T5, is down.

To configure steering traffic with SD-WAN:

1. Configure the SD-WAN:

```
config system sdwan
    set status enable
    config zone
        edit "virtual-wan-link"
        next
    end
    config members
        edit 1
            set interface "T1"
        next
        edit 2
            set interface "T2"
        next
        edit 3
```

```
            set interface "T3"
        next
        edit 4
            set interface "T4"
        next
        edit 5
            set interface "T5"
        next
    end
    config service
        edit 1
            set name "1"
            set load-balance enable
            set dst "all"
            set src "172.16.205.0"
            set priority-members 1 2 3 4 5
        next
    end
end
```

2. Configure a static route:

```
config router static
    edit 5
        set dst 8.0.0.0 255.0.0.0
        set distance 1
        set sdwan-zone "virtual-wan-link"
    next
end
```

3. Check the routing table:

```
# get router info routing-table static
Routing table for VRF=0
S       8.0.0.0/8 [1/0] via T2 tunnel 172.16.203.2, [1/0]
                  [1/0] via T3 tunnel 172.16.206.2, [1/0]
                  [1/0] via T1 tunnel 172.16.208.2, [1/0]
                  [1/0] via T4 tunnel 172.16.209.2, [1/0]
```

Check the results:

- When both the T1 and T2 connections are down, T5 stays down as well, and traffic is load-balanced on T3 and T4 by the SD-WAN configuration:

```
 # get vpn ipsec tunnel summary
'T2' 172.16.203.2:0  selectors(total,up): 1/0  rx(pkt,err): 0/0  tx(pkt,err): 0/0
'T3' 172.16.206.2:0  selectors(total,up): 1/1  rx(pkt,err): 0/0  tx(pkt,err): 0/0
'T4' 172.16.209.2:0  selectors(total,up): 1/1  rx(pkt,err): 0/0  tx(pkt,err): 0/4
'T5' 172.16.210.2:0  selectors(total,up): 1/0  rx(pkt,err): 0/0  tx(pkt,err): 0/4
'T1' 172.16.208.2:0  selectors(total,up): 1/0  rx(pkt,err): 0/0  tx(pkt,err): 0/0
```

```
# get router info routing-table static
Routing table for VRF=0
S       8.0.0.0/8 [1/0] via T3 tunnel 172.16.206.2, [1/0]
                  [1/0] via T4 tunnel 172.16.209.2, [1/0]
```

Traffic is load-balanced between the remaining tunnels:

```
# diagnose sniffer packet any 'host 8.8.8.8' 4
interfaces=[any]
filters=[host 8.8.8.8]
3.027055 port5 in 172.16.205.100 -> 8.8.8.8: icmp: echo request
3.027154 T4 out 172.16.205.100 -> 8.8.8.8: icmp: echo request
3.031434 T4 in 8.8.8.8 -> 172.16.205.100: icmp: echo reply
3.031485 port5 out 8.8.8.8 -> 172.16.205.100: icmp: echo reply
3.612818 port5 in 172.16.205.100 -> 8.8.8.8: icmp: echo request
3.612902 T3 out 172.16.205.100 -> 8.8.8.8: icmp: echo request
3.617107 T3 in 8.8.8.8 -> 172.16.205.100: icmp: echo reply
3.617159 port5 out 8.8.8.8 -> 172.16.205.100: icmp: echo reply
4.168845 port5 in 172.16.205.100 -> 8.8.8.8: icmp: echo request
4.168907 T4 out 172.16.205.100 -> 8.8.8.8: icmp: echo request
4.173150 T4 in 8.8.8.8 -> 172.16.205.100: icmp: echo reply
4.173174 port5 out 8.8.8.8 -> 172.16.205.100: icmp: echo reply
4.710907 port5 in 172.16.205.100 -> 8.8.8.8: icmp: echo request
4.710991 T3 out 172.16.205.100 -> 8.8.8.8: icmp: echo request
4.715933 T3 in 8.8.8.8 -> 172.16.205.100: icmp: echo reply
4.715958 port5 out 8.8.8.8 -> 172.16.205.100: icmp: echo reply
```

- When all of the primary overlays are down, T5 is activated and used for traffic

```
# get vpn ipsec tunnel summary
'T2' 172.16.203.2:0  selectors(total,up): 1/0  rx(pkt,err): 0/0  tx(pkt,err): 0/0
'T3' 172.16.206.2:0  selectors(total,up): 1/0  rx(pkt,err): 0/0  tx(pkt,err): 0/0
'T4' 172.16.209.2:0  selectors(total,up): 1/0  rx(pkt,err): 0/0  tx(pkt,err): 0/0
'T5' 172.16.210.2:0  selectors(total,up): 1/1  rx(pkt,err): 0/0  tx(pkt,err): 0/4
'T1' 172.16.208.2:0  selectors(total,up): 1/0  rx(pkt,err): 0/0  tx(pkt,err): 0/0
```

```
# get router info routing-table static
Routing table for VRF=0
S       8.0.0.0/8 [1/0] via T5 tunnel 172.16.210.2, [1/0]
```

Traffic is using the backup overlay, T5:

```
# diagnose sniffer packet any 'host 8.8.8.8' 4
interfaces=[any]
filters=[host 8.8.8.8]
1.907944 port5 in 172.16.205.100 -> 8.8.8.8: icmp: echo request
1.908045 T5 out 172.16.205.100 -> 8.8.8.8: icmp: echo request
1.912283 T5 in 8.8.8.8 -> 172.16.205.100: icmp: echo reply
1.912351 port5 out 8.8.8.8 -> 172.16.205.100: icmp: echo reply
2.665921 port5 in 172.16.205.100 -> 8.8.8.8: icmp: echo request
2.665999 T5 out 172.16.205.100 -> 8.8.8.8: icmp: echo request
2.670209 T5 in 8.8.8.8 -> 172.16.205.100: icmp: echo reply
```

```
2.670235 port5 out 8.8.8.8 -> 172.16.205.100: icmp: echo reply
5.269997 port5 in 172.16.205.100 -> 8.8.8.8: icmp: echo request
5.270090 T5 out 172.16.205.100 -> 8.8.8.8: icmp: echo request
5.274275 T5 in 8.8.8.8 -> 172.16.205.100: icmp: echo reply
5.274300 port5 out 8.8.8.8 -> 172.16.205.100: icmp: echo reply
5.781848 port5 in 172.16.205.100 -> 8.8.8.8: icmp: echo request
5.781920 T5 out 172.16.205.100 -> 8.8.8.8: icmp: echo request
5.786334 T5 in 8.8.8.8 -> 172.16.205.100: icmp: echo reply
5.786363 port5 out 8.8.8.8 -> 172.16.205.100: icmp: echo reply
```

- If T4 recovers, T5 is deactivated and traffic switches to T4:

```
# get vpn ipsec tunnel summary
'T2' 172.16.203.2:0  selectors(total,up): 2/0  rx(pkt,err): 0/0  tx(pkt,err): 0/0
'T3' 172.16.206.2:0  selectors(total,up): 2/0  rx(pkt,err): 0/0  tx(pkt,err): 0/0
'T4' 172.16.209.2:0  selectors(total,up): 2/2  rx(pkt,err): 0/0  tx(pkt,err): 0/0
'T5' 172.16.210.2:0  selectors(total,up): 2/0  rx(pkt,err): 0/0  tx(pkt,err): 0/0
'T1' 172.16.208.2:0  selectors(total,up): 2/0  rx(pkt,err): 0/0  tx(pkt,err): 0/0
```

```
# get router info routing-table static
Routing table for VRF=0
S       8.0.0.0/8 [1/0] via T4 tunnel 172.16.209.2, [1/0]
```

The primary overlay T4 has recovered, and the backup overlay is down again:

```
# diagnose sniffer packet any 'host 8.8.8.8' 4
interfaces=[any]
filters=[host 8.8.8.8]
4.555685 port5 in 172.16.205.100 -> 8.8.8.8: icmp: echo request
4.555790 T4 out 172.16.205.100 -> 8.8.8.8: icmp: echo request
4.560428 T4 in 8.8.8.8 -> 172.16.205.100: icmp: echo reply
4.560478 port5 out 8.8.8.8 -> 172.16.205.100: icmp: echo reply
5.163223 port5 in 172.16.205.100 -> 8.8.8.8: icmp: echo request
5.163332 T4 out 172.16.205.100 -> 8.8.8.8: icmp: echo request
5.167590 T4 in 8.8.8.8 -> 172.16.205.100: icmp: echo reply
5.167620 port5 out 8.8.8.8 -> 172.16.205.100: icmp: echo reply
5.650089 port5 in 172.16.205.100 -> 8.8.8.8: icmp: echo request
5.650194 T4 out 172.16.205.100 -> 8.8.8.8: icmp: echo request
5.654352 T4 in 8.8.8.8 -> 172.16.205.100: icmp: echo reply
5.654387 port5 out 8.8.8.8 -> 172.16.205.100: icmp: echo reply
6.102181 port5 in 172.16.205.100 -> 8.8.8.8: icmp: echo request
6.102263 T4 out 172.16.205.100 -> 8.8.8.8: icmp: echo request
6.106411 T4 in 8.8.8.8 -> 172.16.205.100: icmp: echo reply
6.106445 port5 out 8.8.8.8 -> 172.16.205.100: icmp: echo reply
```

VPN IPsec troubleshooting

See the following IPsec troubleshooting examples:

- Understanding VPN related logs
- IPsec related diagnose commands on page 3338

Understanding VPN related logs

This section provides some IPsec log samples.

IPsec phase1 negotiating

```
logid="0101037127" type="event" subtype="vpn" level="notice" vd="root" eventtime=1544132571
logdesc="Progress IPsec phase 1" msg="progress IPsec phase 1" action="negotiate" remip=11.101.1.1
```

```
locip=173.1.1.1 remport=500 locport=500 outintf="port13"
cookies="e41eeecb2c92b337/0000000000000000" user="N/A" group="N/A" xauthuser="N/A"
xauthgroup="N/A" assignip=N/A vpntunnel="to_HQ" status="success" init="local"
mode="aggressive" dir="outbound" stage=1 role="initiator" result="OK"
```

IPsec phase1 negotiated

```
logid="0101037127" type="event" subtype="vpn" level="notice" vd="root" eventtime=1544132571
logdesc="Progress IPsec phase 1" msg="progress IPsec phase 1" action="negotiate" remip=11.101.1.1
```

```
locip=173.1.1.1 remport=500 locport=500 outintf="port13"
cookies="e41eeecb2c92b337/1230131a28eb4e73" user="N/A" group="N/A" xauthuser="N/A"
xauthgroup="N/A" assignip=N/A vpntunnel="to_HQ" status="success" init="local"
```

```
mode="aggressive" dir="outbound" stage=2 role="initiator" result="DONE"
```

IPsec phase1 tunnel up

```
logid="0101037138" type="event" subtype="vpn" level="notice" vd="root" eventtime=1544132604
logdesc="IPsec connection status changed" msg="IPsec connection status change" action="tunnel-up"
remip=11.101.1.1 locip=173.1.1.1 remport=500 locport=500 outintf="port13"
cookies="5b1c59fab2029e43/bf517e686d3943d2" user="N/A" group="N/A" xauthuser="N/A"
xauthgroup="N/A" assignip=11.11.11.1 vpntunnel="to_HQ" tunnelip=N/A tunnelid=1530910918
tunneltype="ipsec" duration=0 sentbyte=0 rcvdbyte=0 nextstat=0
```

IPsec phase2 negotiate

```
logid="0101037129" type="event" subtype="vpn" level="notice" vd="root" eventtime=1544132604
logdesc="Progress IPsec phase 2" msg="progress IPsec phase 2" action="negotiate" remip=11.101.1.1
```

```
locip=173.1.1.1 remport=500 locport=500 outintf="port13"
cookies="5b1c59fab2029e43/bf517e686d3943d2" user="N/A" group="N/A" xauthuser="N/A"
xauthgroup="N/A" assignip=11.11.11.1 vpntunnel="to_HQ" status="success" init="local"
```

```
mode="quick" dir="outbound" stage=1 role="initiator" result="OK"
```

IPsec phase2 tunnel up

```
logid="0101037139" type="event" subtype="vpn" level="notice" vd="root" eventtime=1544132604
logdesc="IPsec phase 2 status changed" msg="IPsec phase 2 status change" action="phase2-up"
```

```
remip=11.101.1.1 locip=173.1.1.1 remport=500 locport=500 outintf="port13"
cookies="5b1c59fab2029e43/bf517e686d3943d2" user="N/A" group="N/A" xauthuser="N/A"
xauthgroup="N/A" assignip=11.11.11.1 vpntunnel="to_HQ"
```

```
phase2_name="to_HQ"
```

IPsec phase2 sa install

```
logid="0101037133" type="event" subtype="vpn" level="notice" vd="root" eventtime=1544132604
logdesc="IPsec SA installed" msg="install IPsec SA" action="install_sa" remip=11.101.1.1
locip=173.1.1.1
```

```
remport=500 locport=500 outintf="port13" cookies="5b1c59fab2029e43/bf517e686d3943d2"
user="N/A" group="N/A" xauthuser="N/A" xauthgroup="N/A" assignip=11.11.11.1 vpntunnel="to_HQ"
role="initiator" in_spi="ca646448" out_spi="747c10c6"
```

IPsec tunnel statistics

```
logid="0101037141" type="event" subtype="vpn" level="notice" vd="root" eventtime=1544131118
logdesc="IPsec tunnel statistics" msg="IPsec tunnel statistics" action="tunnel-stats"
remip=10.1.100.15 locip=172.16.200.4 remport=500 locport=500 outintf="mgmt1"
cookies="3539884dbd8f3567/c32e4c1beca91b36"
```

```
user="N/A" group="N/A" xauthuser="N/A" xauthgroup="N/A" assignip=N/A vpntunnel="L2tpoIPsec_0"
tunnelip=10.1.100.15 tunnelid=1530910802 tunneltype="ipsec" duration=6231 sentbyte=57343
rcvdbyte=142640 nextstat=60
```

IPsec phase2 tunnel down

```
logid="0101037138" type="event" subtype="vpn" level="notice" vd="root" eventtime=1544132571
logdesc="IPsec connection status changed" msg="IPsec connection status change" action="tunnel-
down" remip=11.101.1.1 locip=173.1.1.1 remport=500 locport=500 outintf="port13"
cookies="30820aa390687e39/886e72bf5461fb8d" user="N/A" group="N/A" xauthuser="N/A"
xauthgroup="N/A" assignip=11.11.11.1 vpntunnel="to_HQ" tunnelip=N/A tunnelid=1530910786
tunneltype="ipsec" duration=6425 sentbyte=504 rcvdbyte=152 nextstat=0
```

IPsec phase1 sa deleted

```
logid="0101037134" type="event" subtype="vpn" level="notice" vd="root" eventtime=1544132571
logdesc="IPsec phase 1 SA deleted" msg="delete IPsec phase 1 SA" action="delete_phase1_sa"
remip=11.101.1.1 locip=173.1.1.1 remport=500 locport=500 outintf="port13"
cookies="30820aa390687e39/886e72bf5461fb8d" user="N/A" group="N/A" xauthuser="N/A"
xauthgroup="N/A" assignip=11.11.11.1 vpntunnel="to_HQ"
```

IPsec related diagnose commands

This section provides IPsec related diagnose commands.

- Daemon IKE summary information list: `diagnose vpn ike status`

```
connection: 2/50
IKE SA: created 2/51  established 2/9  times 0/13/40 ms
IPsec SA: created 1/13  established 1/7  times 0/8/30 ms
```

- IPsec phase1 interface status: `diagnose vpn ike gateway list`

```
vd: root/0
name: tofgtc
version: 1
interface: port13 42
addr: 173.1.1.1:500 -> 172.16.200.3:500
created: 4313s ago
IKE SA: created 1/1  established 1/1  time 10/10/10 ms
IPsec SA: created 0/0

  id/spi: 92 5639f7f8a5dc54c0/809a6c9bbd266a4b
  direction: initiator
  status: established 4313-4313s ago = 10ms
  proposal: aes128-sha256
  key: 74aa3d63d88e10ea-8a1c73b296b06578
  lifetime/rekey: 86400/81786
  DPD sent/recv: 00000000/00000000

vd: root/0
name: to_HQ
version: 1
interface: port13 42
addr: 173.1.1.1:500 -> 11.101.1.1:500
created: 1013s ago
assigned IPv4 address: 11.11.11.1/255.255.255.252
IKE SA: created 1/1  established 1/1  time 0/0/0 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 95 255791bd30c749f4/c2505db65210258b
  direction: initiator
  status: established 1013-1013s ago = 0ms
  proposal: aes128-sha256
  key: bb101b9127ed5844-1582fd614d5a8a33
  lifetime/rekey: 86400/85086
  DPD sent/recv: 00000000/00000010
```

- IPsec phase2 tunnel status: `diagnose vpn tunnel list`

```
list all ipsec tunnel in vd 0
----
nname=L2tpoIPsec ver=1 serial=6 172.16.200.4:0->0.0.0.0:0 tun_id=0.0.0.0
```

```
bound_if=4 lgwy=static/1 tun=intf/0 mode=dialup/2 encap=none/24 options[0018]=npu create_dev
proxyid_num=0 child_num=0 refcnt=10 ilast=13544 olast=13544 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=0 idle=60000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
run_tally=0
----
name=to_HQ ver=1 serial=7 173.1.1.1:0->11.101.1.1:0 tun_id=11.101.1.1
bound_if=42 lgwy=static/1 tun=intf/0 mode=auto/1 encap=none/8 options[0008]=npu
proxyid_num=1 child_num=0 refcnt=13 ilast=10 olast=1112 ad=/0
stat: rxp=1 txp=4 rxb=152 txb=336
dpd: mode=on-demand on=1 idle=20000ms retry=3 count=0 seqno=5
natt: mode=none draft=0 interval=0 remote_port=0
proxyid=to_HQ proto=0 sa=1 ref=2 serial=1
  src: 0:0.0.0.0/0.0.0.0:0
  dst: 0:0.0.0.0/0.0.0.0:0
  SA:  ref=6 options=10226 type=00 soft=0 mtu=1438 expire=41773/0B replaywin=2048
       seqno=5 esn=0 replaywin_lastseq=00000002 itn=0
  life: type=01 bytes=0/0 timeout=42900/43200
  dec: spi=ca64644a esp=aes key=16 6cc873fdef91337a6cf9b6948972c90f
       ah=sha1 key=20 e576dbe3ff92605931e5670ad57763c50c7dc73a
  enc: spi=747c10c8 esp=aes key=16 5060ad8d0da6824204e3596c0bd762f4
       ah=sha1 key=20 52965cbd5b6ad95212fc825929d26c0401948abe
  dec:pkts/bytes=1/84, enc:pkts/bytes=4/608
  npu_flag=03 npu_rgwy=11.101.1.1 npu_lgwy=173.1.1.1 npu_selid=5 dec_npuid=2 enc_npuid=2
```

- Packets encrypted/decrypted counter: `diagnose vpn ipsec status`

```
All ipsec crypto devices in use:
NP6_0:
    Encryption (encrypted/decrypted)
        null             : 0                1.
        des              : 0                1.
        3des             : 0                1.
        aes              : 0                1.
        aes-gcm          : 0                1.
        aria             : 0                1.
        seed             : 0                1.
        chacha20poly1305 : 0                1.
    Integrity (generated/validated)
        null             : 0                1.
        md5              : 0                1.
        sha1             : 0                1.
        sha256           : 0                1.
        sha384           : 0                1.
        sha512           : 0                1.

NP6_1:
    Encryption (encrypted/decrypted)
        null             : 0                1.
        des              : 0                1.
        3des             : 0                1.
```

```
        aes              : 337152           46069
        aes-gcm          : 0                 1.
        aria             : 0                 1.
        seed             : 0                 1.
        chacha20poly1305 : 0                 1.
    Integrity (generated/validated)
        null             : 0                 1.
        md5              : 0                 1.
        sha1             : 337152           46069
        sha256           : 0                 1.
        sha384           : 0                 1.
        sha512           : 0                 1.

NPU Host Offloading:
    Encryption (encrypted/decrypted)
        null             : 0                 1.
        des              : 0                 1.
        3des             : 0                 1.
        aes              : 38                1.
        aes-gcm          : 0                 1.
        aria             : 0                 1.
        seed             : 0                 1.
        chacha20poly1305 : 0                 1.
    Integrity (generated/validated)
        null             : 0                 1.
        md5              : 0                 1.
        sha1             : 38                1.
        sha256           : 0                 1.
        sha384           : 0                 1.
        sha512           : 0                 1.

CP8:
    Encryption (encrypted/decrypted)
        null             : 0                 1.
        des              : 0                 1.
        3des             : 1337             1582
        aes              : 71               11426
        aes-gcm          : 0                 1.
        aria             : 0                 1.
        seed             : 0                 1.
        chacha20poly1305 : 0                 1.
    Integrity (generated/validated)
        null             : 0                 1.
        md5              : 48               28
        sha1             : 1360             12980
        sha256           : 0                 1.
        sha384           : 0                 1.
        sha512           : 0                 1.

SOFTWARE:
    Encryption (encrypted/decrypted)
        null             : 0                 1.
```

```
        des              : 0                1.
        3des             : 0                1.
        aes              : 0                1.
        aes-gcm          : 0                1.
        aria             : 0                1.
        seed             : 0                1.
        chacha20poly1305 : 0                1.
    Integrity (generated/validated)
        null             : 0                1.
        md5              : 0                1.
        sha1             : 0                1.
        sha256           : 0                1.
        sha384           : 0                1.
        sha512           : 0                1.
```

- diagnose debug application ike -1
  - diagnose vpn ike log filter rem-addr4 11.101.1.1
  - diagnose vpn ike log filter loc-addr4 173.1.1.1

```
# ike 0:to_HQ:101: initiator: aggressive mode is sending 1st message...
ike 0:to_HQ:101: cookie dff03f1d4820222a/0000000000000000
ike 0:to_HQ:101: sent IKE msg (agg_i1send): 173.1.1.1:500->11.101.1.1:500, len=912,
id=dff03f1d4820222a/0000000000000000
ike 0: comes 11.101.1.1:500->173.1.1.1:500,ifindex=42....
ike 0: IKEv1 exchange=Aggressive id=dff03f1d4820222a/6c2caf4dcf5bab75 len=624
ike 0:to_HQ:101: initiator: aggressive mode get 1st response...
ike 0:to_HQ:101: VID RFC 3947 4A131C81070358455C5728F20E95452F
ike 0:to_HQ:101: VID DPD AFCAD71368A1F1C96B8696FC77570100
ike 0:to_HQ:101: DPD negotiated
ike 0:to_HQ:101: VID draft-ietf-ipsra-isakmp-xauth-06.txt 09002689DFD6B712
ike 0:to_HQ:101: VID CISCO-UNITY 12F5F28C457168A9702D9FE274CC0204
ike 0:to_HQ:101: peer supports UNITY
ike 0:to_HQ:101: VID FORTIGATE 8299031757A36082C6A621DE00000000
ike 0:to_HQ:101: peer is [[QualityAssurance62/FortiGate]]/FortiOS (v0 b0)
ike 0:to_HQ:101: VID FRAGMENTATION 4048B7D56EBCE88525E7DE7F00D6C2D3
ike 0:to_HQ:101: VID FRAGMENTATION 4048B7D56EBCE88525E7DE7F00D6C2D3C0000000
ike 0:to_HQ:101: peer identifier IPV4_ADDR 11.101.1.1
ike 0:to_HQ:101: negotiation result
ike 0:to_HQ:101: proposal id = 1:
ike 0:to_HQ:101:   protocol id = ISAKMP:
ike 0:to_HQ:101:      trans_id = KEY_IKE.
ike 0:to_HQ:101:      encapsulation = IKE/none
ike 0:to_HQ:101:         type=OAKLEY_ENCRYPT_ALG, val=AES_CBC, key-len=128
ike 0:to_HQ:101:         type=OAKLEY_HASH_ALG, val=SHA2_256.
ike 0:to_HQ:101:         type=AUTH_METHOD, val=PRESHARED_KEY_XAUTH_I.
ike 0:to_HQ:101:         type=OAKLEY_GROUP, val=MODP2048.
ike 0:to_HQ:101: ISAKMP SA lifetime=86400
ike 0:to_HQ:101: received NAT-D payload type 20
ike 0:to_HQ:101: received NAT-D payload type 20
ike 0:to_HQ:101: selected NAT-T version: RFC 3947
ike 0:to_HQ:101: NAT not detected
```

```
ike 0:to_HQ:101: ISAKMP SA dff03f1d4820222a/6c2caf4dcf5bab75 key
16:D81CAE6B2500435BFF195491E80148F3
ike 0:to_HQ:101: PSK authentication succeeded
ike 0:to_HQ:101: authentication OK
ike 0:to_HQ:101: add INITIAL-CONTACT
ike 0:to_HQ:101: sent IKE msg (agg_i2send): 173.1.1.1:500->11.101.1.1:500, len=172,
id=dff03f1d4820222a/6c2caf4dcf5bab75
ike 0:to_HQ:101: established IKE SA dff03f1d4820222a/6c2caf4dcf5bab75
ike 0: comes 11.101.1.1:500->173.1.1.1:500,ifindex=42....
ike 0: IKEv1 exchange=Mode config id=dff03f1d4820222a/6c2caf4dcf5bab75:97d88fb4 len=92
ike 0:to_HQ:101: mode-cfg type 16521 request 0:
ike 0:to_HQ:101: mode-cfg type 16522 request 0:
ike 0:to_HQ:101: sent IKE msg (cfg_send): 173.1.1.1:500->11.101.1.1:500, len=108,
id=dff03f1d4820222a/6c2caf4dcf5bab75:97d88fb4
ike 0: comes 11.101.1.1:500->173.1.1.1:500,ifindex=42....
ike 0: IKEv1 exchange=Mode config id=dff03f1d4820222a/6c2caf4dcf5bab75:3724f295 len=92
ike 0:to_HQ:101: sent IKE msg (cfg_send): 173.1.1.1:500->11.101.1.1:500, len=92,
id=dff03f1d4820222a/6c2caf4dcf5bab75:3724f295
ike 0:to_HQ:101: initiating mode-cfg pull from peer
ike 0:to_HQ:101: mode-cfg request APPLICATION_VERSION
ike 0:to_HQ:101: mode-cfg request INTERNAL_IP4_ADDRESS
ike 0:to_HQ:101: mode-cfg request INTERNAL_IP4_NETMASK
ike 0:to_HQ:101: mode-cfg request UNITY_SPLIT_INCLUDE
ike 0:to_HQ:101: mode-cfg request UNITY_PFS
ike 0:to_HQ:101: sent IKE msg (cfg_send): 173.1.1.1:500->11.101.1.1:500, len=140,
id=dff03f1d4820222a/6c2caf4dcf5bab75:3bca961f
ike 0: comes 11.101.1.1:500->173.1.1.1:500,ifindex=42....
ike 0: IKEv1 exchange=Mode config id=dff03f1d4820222a/6c2caf4dcf5bab75:3bca961f len=172
ike 0:to_HQ:101: mode-cfg type 1 response 4:0B0B0B01
ike 0:to_HQ:101: mode-cfg received INTERNAL_IP4_ADDRESS 11.11.11.1
ike 0:to_HQ:101: mode-cfg type 2 response 4:FFFFFFFC
ike 0:to_HQ:101: mode-cfg received INTERNAL_IP4_NETMASK 255.255.255.252
ike 0:to_HQ:101: mode-cfg received UNITY_PFS 1
ike 0:to_HQ:101: mode-cfg type 28676 response
28:0A016400FFFFFF00000000000000A016500FFFFFF00000000000000
ike 0:to_HQ:101: mode-cfg received UNITY_SPLIT_INCLUDE 0 10.1.100.0/255.255.255.0:0 local port
0
ike 0:to_HQ:101: mode-cfg received UNITY_SPLIT_INCLUDE 0 10.1.101.0/255.255.255.0:0 local port
0
ike 0:to_HQ:101: mode-cfg received APPLICATION_VERSION 'FortiGate-100D v6.0.3,build0200,181009
(GA)'
ike 0:to_HQ: mode-cfg add 11.11.11.1/255.255.255.252 to 'to_HQ'/58
ike 0:to_HQ: set oper up
ike 0:to_HQ: schedule auto-negotiate
ike 0:to_HQ:101: no pending Quick-Mode negotiations
ike shrank heap by 159744 bytes
ike 0:to_HQ:to_HQ: IPsec SA connect 42 173.1.1.1->11.101.1.1:0
ike 0:to_HQ:to_HQ: using existing connection
```

```
# ike 0:to_HQ:to_HQ: config found
ike 0:to_HQ:to_HQ: IPsec SA connect 42 173.1.1.1->11.101.1.1:500 negotiating
ike 0:to_HQ:101: cookie dff03f1d4820222a/6c2caf4dcf5bab75:32f4cc01
```

```
ike 0:to_HQ:101:to_HQ:259: initiator selectors 0 0:0.0.0.0/0.0.0.0:0:0->0:0.0.0.0/0.0.0.0:0:0
ike 0:to_HQ:101: sent IKE msg (quick_i1send): 173.1.1.1:500->11.101.1.1:500, len=620,
id=dff03f1d4820222a/6c2caf4dcf5bab75:32f4cc01
ike 0: comes 11.101.1.1:500->173.1.1.1:500,ifindex=42....
ike 0: IKEv1 exchange=Quick id=dff03f1d4820222a/6c2caf4dcf5bab75:32f4cc01 len=444
ike 0:to_HQ:101:to_HQ:259: responder selectors 0:0.0.0.0/0.0.0.0:0->0:0.0.0.0/0.0.0.0:0
ike 0:to_HQ:101:to_HQ:259: my proposal:
ike 0:to_HQ:101:to_HQ:259: proposal id = 1:
ike 0:to_HQ:101:to_HQ:259:   protocol id = IPSEC_ESP:
ike 0:to_HQ:101:to_HQ:259:   PFS DH group = 14
ike 0:to_HQ:101:to_HQ:259:      trans_id = ESP_AES_CBC (key_len = 128)
ike 0:to_HQ:101:to_HQ:259:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:to_HQ:101:to_HQ:259:         type = AUTH_ALG, val=SHA1
ike 0:to_HQ:101:to_HQ:259:      trans_id = ESP_AES_CBC (key_len = 256)
ike 0:to_HQ:101:to_HQ:259:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:to_HQ:101:to_HQ:259:         type = AUTH_ALG, val=SHA1
ike 0:to_HQ:101:to_HQ:259:      trans_id = ESP_AES_CBC (key_len = 128)
ike 0:to_HQ:101:to_HQ:259:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:to_HQ:101:to_HQ:259:         type = AUTH_ALG, val=SHA2_256
ike 0:to_HQ:101:to_HQ:259:      trans_id = ESP_AES_CBC (key_len = 256)
ike 0:to_HQ:101:to_HQ:259:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:to_HQ:101:to_HQ:259:         type = AUTH_ALG, val=SHA2_256
ike 0:to_HQ:101:to_HQ:259:      trans_id = ESP_AES_GCM_16 (key_len = 128)
ike 0:to_HQ:101:to_HQ:259:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:to_HQ:101:to_HQ:259:         type = AUTH_ALG, val=NULL
ike 0:to_HQ:101:to_HQ:259:      trans_id = ESP_AES_GCM_16 (key_len = 256)
ike 0:to_HQ:101:to_HQ:259:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:to_HQ:101:to_HQ:259:         type = AUTH_ALG, val=NULL
ike 0:to_HQ:101:to_HQ:259:      trans_id = ESP_CHACHA20_POLY1305 (key_len = 256)
ike 0:to_HQ:101:to_HQ:259:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:to_HQ:101:to_HQ:259:         type = AUTH_ALG, val=NULL
ike 0:to_HQ:101:to_HQ:259: incoming proposal:
ike 0:to_HQ:101:to_HQ:259: proposal id = 1:
ike 0:to_HQ:101:to_HQ:259:   protocol id = IPSEC_ESP:
ike 0:to_HQ:101:to_HQ:259:   PFS DH group = 14
ike 0:to_HQ:101:to_HQ:259:      trans_id = ESP_AES_CBC (key_len = 128)
ike 0:to_HQ:101:to_HQ:259:      encapsulation = ENCAPSULATION_MODE_TUNNEL
ike 0:to_HQ:101:to_HQ:259:         type = AUTH_ALG, val=SHA1
ike 0:to_HQ: schedule auto-negotiate
ike 0:to_HQ:101:to_HQ:259: replay protection enabled
ike 0:to_HQ:101:to_HQ:259: SA life soft seconds=42902.
ike 0:to_HQ:101:to_HQ:259: SA life hard seconds=43200.
ike 0:to_HQ:101:to_HQ:259: IPsec SA selectors #src=1 #dst=1
ike 0:to_HQ:101:to_HQ:259: src 0 4 0:0.0.0.0/0.0.0.0:0
ike 0:to_HQ:101:to_HQ:259: dst 0 4 0:0.0.0.0/0.0.0.0:0
ike 0:to_HQ:101:to_HQ:259: add IPsec SA: SPIs=ca64644b/747c10c9
ike 0:to_HQ:101:to_HQ:259: IPsec SA dec spi ca64644b key 16:D5C60F1A3951B288CE4DEC7E04D2119D
auth 20:F872A7A26964208A9AA368A31AEFA3DB3F3780BC
ike 0:to_HQ:101:to_HQ:259: IPsec SA enc spi 747c10c9 key 16:97952E1594F718128D9D7B09400856EA
auth 20:4D5E5BC45A9D5A9A4631E911932F5650A4639A37
ike 0:to_HQ:101:to_HQ:259: added IPsec SA: SPIs=ca64644b/747c10c9
```

```
ike 0:to_HQ:101:to_HQ:259: sending SNMP tunnel UP trap
ike 0:to_HQ:101: sent IKE msg (quick_i2send): 173.1.1.1:500->11.101.1.1:500, len=76,
id=dff03f1d4820222a/6c2caf4dcf5bab75:32f4cc01
```

- Flush a Phase2 tunnel security association (SA) by name:
  - List the tunnels: `diagnose vpn tunnel list`
  - Pick and flush the specific tunnel or tunnels: `diagnose vpn tunnel flush <name> [<name>] ... [<name>]`

```
# diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=ng-vpn-access_0 ver=2 serial=3 203.0.113.2:4500->198.51.100.2:57245
nexthop=203.0.113.3 tun_id=10.100.66.10 tun_id6=::10.0.0.3 status=up dst_mtu=1500 weight=1
bound_if=5 real_if=5 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/74664 options
[123a8]=npu rgwy-chg rport-chg frag-rfc  run_state=0 role=primary accept_traffic=1
 overlay_id=0

parent=ng-vpn-access index=0
proxyid_num=1 child_num=0 refcnt=6 ilast=2 olast=44301916 ad=/0
stat: rxp=212 txp=0 rxb=16883 txb=0
dpd: mode=on-demand on=1 status=ok idle=20000ms retry=3 count=0 seqno=0
natt: mode=keepalive draft=0 interval=10 remote_port=57245
fec: egress=0 ingress=0
proxyid=ng-vpn-access proto=0 sa=1 ref=2 serial=1 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.100.66.10-10.100.66.10:0
  SA:  ref=3 options=20682 type=00 soft=0 mtu=1422 expire=42922/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=000000d4 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43188/43200
  dec: spi=c595e5cb esp=aes key=16 25f56b4270d642a53b3286c776568ad3
       ah=sha1 key=20 ee4b56346a0ea6d307110281a4fbb245a610230c
  enc: spi=7c9076c5 esp=aes key=16 3b57c313c14532d7bea2a5bd4a0f4788
       ah=sha1 key=20 1f8cf4a88dd78a7bf2b0db25910d42b7a6a9b30a
  dec:pkts/bytes=212/16883, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=198.51.100.2 npu_lgwy=203.0.113.2 npu_selid=1 dec_npuid=0 enc_
npuid=0
------------------------------------------------------
name=ng-vpn-access_1 ver=2 serial=4 203.0.113.2:4500->198.51.100.3:49205
nexthop=203.0.113.3 tun_id=10.100.66.11 tun_id6=::10.0.0.4 status=up dst_mtu=0 weight=1
bound_if=5 real_if=0 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/74664 options
[123a8]=npu rgwy-chg rport-chg frag-rfc  run_state=0 role=primary accept_traffic=1
 overlay_id=0

parent=ng-vpn-access index=1
proxyid_num=1 child_num=0 refcnt=6 ilast=0 olast=44301916 ad=/0
stat: rxp=29 txp=0 rxb=2507 txb=0
dpd: mode=on-demand on=1 status=ok idle=20000ms retry=3 count=0 seqno=0
natt: mode=keepalive draft=0 interval=10 remote_port=49205
fec: egress=0 ingress=0
proxyid=ng-vpn-access proto=0 sa=1 ref=2 serial=1 add-route
```

```
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.100.66.11-10.100.66.11:0
  SA:  ref=3 options=20682 type=00 soft=0 mtu=1280 expire=43178/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=0000001d qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43188/43200
  dec: spi=c595e5cc esp=aes key=16 4a1c4602a540bb31f4a01f8bd5b9fc58
       ah=sha1 key=20 de95405ff86798ef1c9856a2b059ce04d8ceb3ce
  enc: spi=218a9591 esp=aes key=16 29038591b7a6d67ea92d4035609f3178
       ah=sha1 key=20 b11fb40a319f57a948c86d3bb30f9a69f9f0c9ea
  dec:pkts/bytes=29/2507, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=198.51.100.3 npu_lgwy=203.0.113.2 npu_selid=2 dec_npuid=0 enc_
npuid=0
------------------------------------------------------
name=ng-vpn-access ver=2 serial=1 203.0.113.2:0->0.0.0.0:0 nexthop=203.0.113.3 tun_
id=10.0.0.1 tun_id6=::10.0.0.1 status=up dst_mtu=0 weight=1
bound_if=5 real_if=0 lgwy=static/1 tun=intf mode=dialup/2 encap=none/552 options[0228]=npu
frag-rfc  role=primary accept_traffic=1 overlay_id=0

proxyid_num=0 child_num=2 refcnt=4 ilast=44301916 olast=44301916 ad=/0
stat: rxp=357 txp=0 rxb=28890 txb=0
dpd: mode=on-demand on=0 status=ok idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
```

```
# diagnose vpn tunnel flush ng-vpn-access_0
# diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=ng-vpn-access_1 ver=2 serial=4 203.0.113.2:4500->198.51.100.3:49205 nexthop= tun_
id=10.100.66.11 tun_id6=::10.0.0.4 status=up dst_mtu=0 weight=1
bound_if=5 real_if=0 lgwy=static/1 tun=intf mode=dial_inst/3 encap=none/74664 options
[123a8]=npu rgwy-chg rport-chg frag-rfc  run_state=0 role=primary accept_traffic=1
 overlay_id=0

parent=ng-vpn-access index=1
proxyid_num=1 child_num=0 refcnt=6 ilast=0 olast=44301940 ad=/0
stat: rxp=74 txp=0 rxb=5784 txb=0
dpd: mode=on-demand on=1 status=ok idle=20000ms retry=3 count=0 seqno=0
natt: mode=keepalive draft=0 interval=10 remote_port=49205
fec: egress=0 ingress=0
proxyid=ng-vpn-access proto=0 sa=1 ref=2 serial=1 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.100.66.11-10.100.66.11:0
  SA:  ref=3 options=20682 type=00 soft=0 mtu=1280 expire=43155/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=0000004a qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=43188/43200
  dec: spi=c595e5cc esp=aes key=16 4a1c4602a540bb31f4a01f8bd5b9fc58
       ah=sha1 key=20 de95405ff86798ef1c9856a2b059ce04d8ceb3ce
  enc: spi=218a9591 esp=aes key=16 29038591b7a6d67ea92d4035609f3178
       ah=sha1 key=20 b11fb40a319f57a948c86d3bb30f9a69f9f0c9ea
  dec:pkts/bytes=74/5784, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=198.51.100.3 npu_lgwy=203.0.113.2 npu_selid=2 dec_npuid=0 enc_
```

```
npuid=0
----------------------------------------------------
name=ng-vpn-access ver=2 serial=1 203.0.113.2:0->0.0.0.0:0 nexthop= tun_id=10.0.0.1 tun_
id6=::10.0.0.1 status=up dst_mtu=0 weight=1
bound_if=5 real_if=0 lgwy=static/1 tun=intf mode=dialup/2 encap=none/552 options[0228]=npu
frag-rfc  role=primary accept_traffic=1 overlay_id=0

proxyid_num=0 child_num=1 refcnt=3 ilast=44301940 olast=44301940 ad=/0
stat: rxp=412 txp=0 rxb=32977 txb=0
dpd: mode=on-demand on=0 status=ok idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
                                       fec: egress=0 ingress=0
```

- See debug messages while the tunnel is being flushed:
  - `diagnose debug application ike -1`
  - `diagnose debug enable`

```
# diagnose vpn tunnel flush ng-vpn-access_0
ike V=root:0:ng-vpn-access_0: admin flush IPsec SAs
ike V=root:0:ng-vpn-access_0: going to be deleted
ike V=root:0:ng-vpn-access_0: sent tunnel-down message to EMS: (fct-
uid=9A016B5A6E914B42AD4168C066EB04CA, intf=ng-vpn-access_0, addr=10.100.66.10, vdom=root)
ike V=root:0:ng-vpn-access_0: flushing
ike V=root:0:ng-vpn-access_0: deleting IPsec SA with SPI 4c04e46a
ike V=root:0:ng-vpn-access_0:ng-vpn-access: deleted IPsec SA with SPI 4c04e46a, SA count:
0
ike V=root:ng-vpn-access_0: ng-vpn-access:0: del route 10.100.66.10/255.255.255.255 tunnel
10.100.66.10 oif ng-vpn-access(16) metric 15 priority 1
ike V=root:0:ng-vpn-access_0: sending SNMP tunnel DOWN trap for ng-vpn-access
ike V=root:0:ng-vpn-access_0: remote selector down event 10.100.66.10 (devidx=16)
ike V=root:0:ng-vpn-access_0: user 'tsmith' 10.100.66.10 del
ike V=root:0:ng-vpn-access_0:ng-vpn-access: delete
ike V=root:0:ng-vpn-access_0: flushed
ike V=root:0:ng-vpn-access_0:0:1: send informational
ike 0:ng-vpn-access_0:0: enc 00000008010000000706050403020107
ike 0:ng-vpn-access_0:0: out
D2AC0DCA7ACF1905BD68AE1C9065DEF72E2025000000000000000502A00003408EDAD27D533547E17DB2EB3A9
DC7C5DF2151C08B8E99D56D78075056233ABBB6CCC2E1566
104FFF6C0996E5209103C5
ike V=root:0:ng-vpn-access_0:0: sent IKE msg (INFORMATIONAL): 203.0.113.2:4500-
>198.51.100.2:61368, len=80, vrf=0, id=d2ac0dca7acf1905/bd68ae1c9065def7, oif=5
ike V=root:0:ng-vpn-access_0: mode-cfg release 10.100.66.10/255.255.255.255
ike V=root:0:ng-vpn-access_0: delete dynamic
```

- Flush all Phase2 tunnel SAs without specifying a name: `diagnose vpn tunnel flush`

```
# diagnose vpn tunnel list
list all ipsec tunnel in vd 0
----------------------------------------------------
name=ng-vpn-access ver=2 serial=1 203.0.113.2:0->0.0.0.0:0 nexthop= tun_id=10.0.0.1 tun_
id6=::10.0.0.1 status=up dst_mtu=0 weight=1
bound_if=5 real_if=0 lgwy=static/1 tun=intf mode=dialup/2 encap=none/552 options[0228]=npu
```

```
frag-rfc  role=primary accept_traffic=1 overlay_id=0

proxyid_num=0 child_num=0 refcnt=2 ilast=44301952 olast=44301952 ad=/0
stat: rxp=430 txp=0 rxb=34189 txb=0
dpd: mode=on-demand on=0 status=ok idle=20000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
```

- View the Phase1 gateway: `diagnose vpn ike gateway list`

```
vd: root/0
name: ng-vpn-access_0
version: 2
interface: port3 5
addr: 203.0.113.2:4500 -> 198.51.100.2:52368
tun_id: 10.100.66.10/::10.0.0.5
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 188s ago
eap-user: tsmith
2FA: no
groups:
  LDAP-Administrator 4
peer-id: 198.51.100.2
peer-id-auth: no
FortiClient UID: 9A016B5A6E914B42AD4168C066EB04CA
assigned IPv4 address: 10.100.66.10/255.255.255.255
nat: me peer
pending-queue: 0
PPK: no
IKE SA: created 1/1  established 1/1  time 5130/5130/5130 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 7 e05dcbd4346e9e3f/cf853d169817be83
  direction: responder
  status: established 188-183s ago = 5130ms
  proposal: aes256-sha256
  child: no
  SK_ei: 41dd4ed657a5588d-8cd005291374945a-251931ff73ebba0f-738b3651faaa843a
  SK_er: 65bfcdf70d63f4e2-08c7f995aae55c0f-04c98ee186246de3-18c3ddfa159ba747
  SK_ai: 96bb92713a04f6c5-6371892639daa541-b07a706cc4b180f9-d3fa04772efc16fd
  SK_ar: d465e491b24362e0-454f3670cb8dc264-3bad9d1c7f890d90-e9047b00a5ebd7f1
  PPK: no
  message-id sent/recv: 0/29
  QKD: no
  PQC-KEM (IKE): no
  PQC-KEM (all IPsec): no
  lifetime/rekey: 86400/85946
  DPD sent/recv: 00000000/00000000
  peer-id: 198.51.100.2
```

```
vd: root/0
name: ng-vpn-access_1
version: 2
interface: port3 5
addr: 203.0.113.2:4500 -> 198.51.100.3:59702
tun_id: 10.100.66.11/::10.0.0.6
remote_location: 0.0.0.0
network-id: 0
transport: UDP
created: 138s ago
eap-user: tsmith
2FA: no
groups:
  LDAP-Administrator 4
peer-id: 198.51.100.3
peer-id-auth: no
FortiClient UID: 9A016B5A6E914B42AD4168C066EB04CA
assigned IPv4 address: 10.100.66.11/255.255.255.255
nat: me peer
pending-queue: 0
PPK: no
IKE SA: created 1/1  established 1/1  time 5130/5130/5130 ms
IPsec SA: created 1/1  established 1/1  time 0/0/0 ms

  id/spi: 8 6d9d6fc4a5f65a09/c02126815c1931cd
  direction: responder
  status: established 138-133s ago = 5130ms
  proposal: aes256-sha256
  child: no
  SK_ei: 888bb8b6ca8a387a-61487120a2243997-e6e7b227ae06600a-e77a9b89fe0158e0
  SK_er: 1099088559585d0f-f30e1ffe23a7ac03-3ae7d65a8b8f178f-fc17ad65c018dd99
  SK_ai: 55a06b7336ca937f-25d5cc8583e8688a-18b9cddbbdb60336-9f412fd9c5baafa5
  SK_ar: e30739bcfc070e6a-e7ef36a3857eb0c6-7d6b4d1acc6d960b-9c72889da9fa93bb
  PPK: no
  message-id sent/recv: 0/24
  QKD: no
  PQC-KEM (IKE): no
  PQC-KEM (all IPsec): no
  lifetime/rekey: 86400/85996
  DPD sent/recv: 00000000/00000000
  peer-id: 198.51.100.3
```

- Filter a gateway by its local address, and then flush it:
  - `diagnose vpn ike gateway filter loc-addr4 203.0.113.2 203.0.113.8`
  - `diagnose vpn ike gateway clear`

Agentless VPN

A Virtual Private Network (VPN) allows remote users to securely access private networks over the Internet, protecting data from unauthorized access. Employees working remotely can use a VPN to connect securely to their office network. VPNs also enable secure interconnectivity between multiple office locations.

Agentless VPN uses the Secure Socket Layer (SSL) protocol to establish a secure tunnel through a web browser, eliminating the need for a dedicated VPN client. Operating at the application layer of the OSI model, it provides clientless access to specific network resources without additional software installation.

Users authenticate to FortiGate's Agentless VPN web portal, which provides access to network services and resources, including HTTP/HTTPS, Telnet, FTP, SMB/CIFS, VNC, RDP, and SSH. When a user starts a connection to a server from the web portal, FortiOS proxies the communication with the server. All communication between the FortiGate and the user continues over HTTPS, regardless of the service being accessed.

Agentless VPN security restricts and validates HTTP messages sent from clients to FortiGate. With advanced checks and binary code verification, FortiGate automatically detects and blocks certain HTTP methods that could be exploited for unauthorized access attempts. By implementing this proactive defense, FortiGate enhances the security of its Agentless VPN feature, ensuring a safer environment for users.

Agentless VPN can be used when you require:

- A clientless solution in which all remote services are accessed through a web portal
- Tight control over the contents of the web portal
- Limited services to be provided to the remote users

Considerations to use Agentless VPN:

- Multiple applications and protocols are not supported.
- VNC and RDP access might have limitations, such as unsupported shortcut keys.
- In some configurations, RDP can consume a significant amount of memory and CPU resources.
- Firewall performance might decrease as remote usage increases.
- Highly customized web pages might render incorrectly.

Alternative remote-access solutions in FortiOS are IPsec VPN on page 2760 and Zero Trust Network Access on page 1590.

To overcome the limitations of Agentless VPN, consider using IPsec VPN, which requires the FortiClient VPN agent to be installed on the remote endpoint. The FortiClient VPN client is available in both free and paid tiers, offering enhanced security and performance. For supported operating systems, refer to the FortiClient Technical Specifications.

For the highest VPN throughput and reliability, configure a dial-up IPsec VPN, which provides superior performance compared to SSL-based VPN solutions. See FortiClient as dialup client on page 2973.

By default, the *VPN > Agentless VPN* and *VPN > Agentless VPN Settings* menus are hidden from the GUI and the CLI.

To enable Agentless VPN, enter:

```
config system global
    set sslvpn-web-mode enable
end
```

To enable Agentless VPN feature visibility in the CLI, enter:

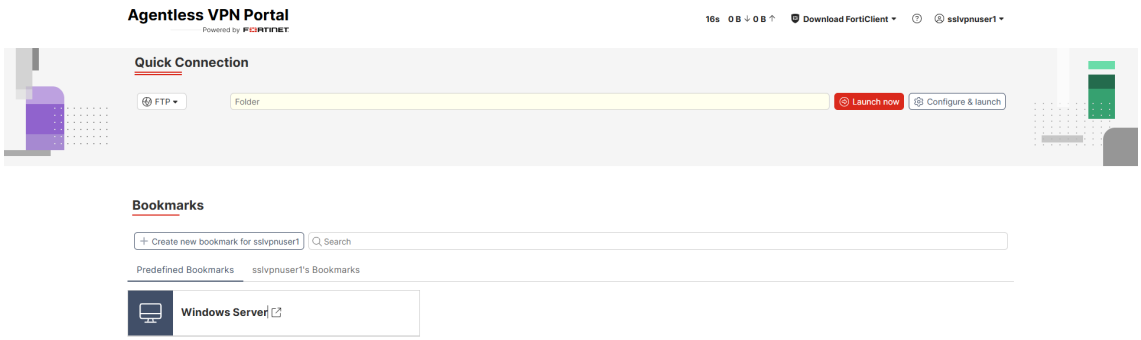
```
config system settings
    set gui-sslvpn enable
end
```

To enable Agentless VPN feature visibility in the GUI, go to *System > Feature Visibility*, enable *Agentless VPN*, and click *Apply*.

When `sslvpn-web-mode` is disabled, an error message appears when users access the Agentless VPN web portal by navigating to the listening IP address, domain, or port.

A user must have valid username and password credentials to log in to an Agentless VPN web portal. Other multi-factor authentication components may also be configured and required, such as FortiTokens.

After logging in, the Agentless VPN web portal page appears:



An Agentless VPN web portal includes the following features:

- The session information is displayed in the right corner of the top banner. This includes the elapsed time since logging in and the volume of inbound and outbound HTTP and HTTPS traffic.
- The *Launch FortiClient* button appears if FortiClient is installed. Click the button to open the *FortiClient Remote Access* tab. FortiClient does not automatically create a VPN connection based on the web-mode connection information.
- The *Download FortiClient* button downloads the FortiClient application for various operating systems.
- The *Bookmarks* widget includes links to network resources (administrator-defined bookmarks), and users can create their own bookmarks.
- The *Quick Connection* dropdown menu enables a connection to network resources without using or creating a bookmark.

The following topics provide information about Agentless VPN in FortiOS:

- Agentless VPN portal configurations on page 3351
- Quick Connection tool on page 3352

- Agentless VPN bookmarks on page 3354
- Agentless VPN for remote users on page 3357
- Customizing the RDP display size on page 3361
- Showing the Agentless VPN web portal login page in the browser's language on page 3365
- Agentless VPN protocols on page 3371
- Agentless VPN security best practices on page 3372
- Agentless VPN authentication on page 3379
- Disable the clipboard for Agentless VPN RDP connections on page 3472
- Post-Quantum Cryptography for Agentless VPN on page 3477
- Agentless VPN troubleshooting on page 3481

Agentless VPN portal configurations

An Agentless VPN portal enables users to access network resources through a secure channel using a web browser. System administrators can configure login privileges and available network resources for users. The portal configuration determines what displays for users logged in to the portal. Both system administrators and users can customize the Agentless VPN portal.

The following predefined, default Agentless VPN portal configurations are available with specific settings for Agentless VPN:

- full-access: Agentless VPN is enabled.
- tunnel-access: Agentless VPN is disabled.
- web-access: Agentless VPN is enabled.

Custom Agentless VPN portals can also be configured.

To configure a custom Agentless VPN portal:

1. Go to *VPN* > *Agentless VPN Portals* and click *Create New*.
2. Configure the following settings as needed:

GUI option	Description
Name	Enter the portal name.
Limit Users to One Agentless VPN Connection at a Time	This option is disabled by default. When enabled, once a user logs in to the portal, they cannot go to another system and log in with the same credentials again.
Web Mode	This option enables or disables Agentless VPN. By default, this option is enabled for new Agentless VPN portals and hidden from the GUI. The option is visible in the GUI after being disabled using the CLI.
Portal Message	Enter a message that appears at the top of the web portal screen (default = *Agentless VPN Portal*).

GUI option		Description
	Theme	Select a color theme from the dropdown.
	Default protocol	Select the default protocol to be visible under *Quick Connection*.
	Show Session Information	Enable to display session information in the top banner of the Agentless portal (username, amount of time logged in, and traffic statistics).
	Show Connection Launcher	Enable to display the *Quick Connection* button.
	Show Login History	Enable to display the user's login history (*History*).
	User Bookmarks	Enable to allow users to add their own bookmarks (*New Bookmark*).
	Predefined Bookmarks	Use the table to create and edit predefined bookmarks. See To create a predefined administrator bookmark in FortiOS: on page 3356 for more details.
	Rewrite Content IP/UI/	Enable contents rewrite for URIs containing `IP-address/ui/`.
	RDP/VNC clipboard	Enable to support RDP/VPC clipboard functionality.
FortiClient Download		Enable this option to display the *Download FortiClient* button.
	Download Method	Select either *Direct* or *Agentless VPN Proxy* as the method to download FortiClient.
	Customize Download Location	Enable to configure a custom download location for *Windows* or *Mac*.

3. Click *OK*.

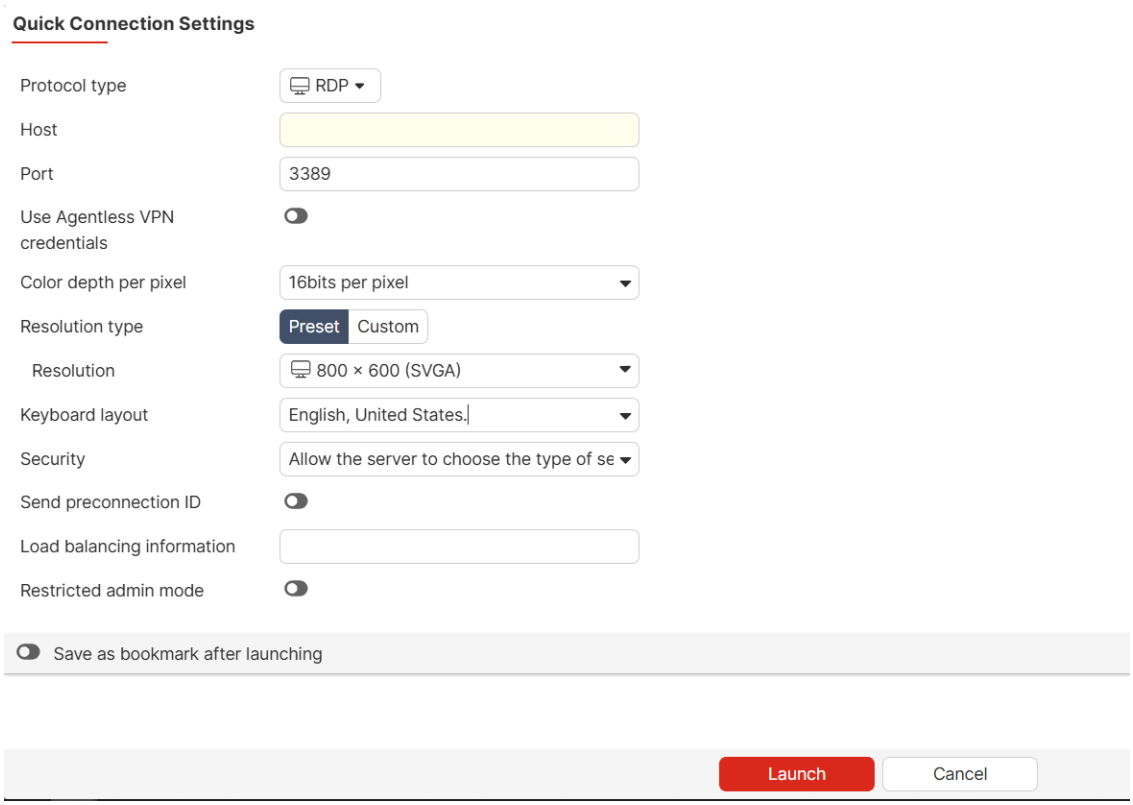
By default, the browser's language preference is automatically detected and used by the Agentless VPN web portal login page. The system language can still be used by changing the settings on the *Agentless VPN Settings* page of the GUI, or disabling browser-language detection in the CLI. See Showing the Agentless VPN web portal login page in the browser's language on page 3365 for more details.

Quick Connection tool

The *Quick Connection* tool allows a user to connect to a resource when it is not a predefined bookmark. The tool allows the user to specify the type of server and the URL or IP address of the host.

To connect to a resource:

1. After logging in to Agentless VPN web portal, under *Quick Connection*, click *Configure & launch*.
2. Select the desired *Protocol type*, and enter the required information, such as the IP address or URL of the host.
3. Click *Launch*.



In a VNC session, to send Ctrl+Alt+Del, press *F8* then select *Send Ctrl-Alt-Delete*.

RDP sessions

Some Windows servers require that a specific security be set for RDP sessions, as opposed to the standard RDP encryption security. For example, Windows 10 requires that TLS be used.

You can specify a location option if the remote computer does not use the same keyboard layout as your computer by appending it to the *Host* field using the following format: `<IP address> -m <locale>`

The available options are:

ar	Arabic	fr-be	Belgian French	no	Norwegian
da	Danish	fr-ca	Canadian French	pl	Polish

de	German	fr-ch	Swiss French	pt	Portuguese
de-ch	Swiss German	hr	Croatian	pt-br	Brazilian Portuguese
en-gb	British English	hu	Hungarian	ru	Russian
en-uk	UK English	it	Italian	sl	Slovenian
en-us	US English	ja	Japanese	sv	Sudanese
es	Spanish	lt	Lithuanian	tk	Turkmen
fi	Finnish	lv	Latvian	tr	Turkish
fr	French	mk	Macedonian		

Agentless VPN bookmarks

The *Bookmarks* widget displays bookmarks configured by administrators and users. Administrator bookmarks cannot be edited, and they are configured in FortiOS. Users can add, edit, and delete their own bookmarks within the web portal.

The FortiGate forwards client requests to servers on the internet or internal network. To use the web portal applications, add the URL, IP address, or name of the server application to the *Bookmarks* list. Once a bookmark is created, click the bookmark icon to initiate a session.

To access a destination without adding a bookmark to the *Your Bookmarks* list, use the Quick Connection tool. See Quick Connection tool on page 3352 for more details.

Configuring bookmarks

The following table summarizes which options can be configured based on the bookmark type in the Agentless VPN web portal:

Setting	HTTP/ HTTPS	FTP	SMB	SFTP	RDP	VNC	SSH	Telnet
URL	✓							
Folder		✓	✓	✓				
Host					✓	✓	✓	✓
Domain			✓					
Port					✓	✓		
Description	✓	✓	✓	✓	✓	✓	✓	✓
Password						✓		
SSO Credentials	✓	✓	✓	✓				
Agentless VPN Login	✓	✓	✓	✓				
SSO Form Data	✓							
Form Key	✓							
Form Value	✓							
Alternative	✓	✓	✓	✓				
Username	✓	✓	✓	✓				
Password	✓	✓	✓	✓				
Use Agentless VPN Credentials					✓			
Username					✓			
Password					✓			
*Color Depth Per Pixel**					✓			
*Screen Width**					✓			
*Screen Height**					✓			
Keyboard Layout					✓			
Security					✓			
Preconnection ID					✓			
Preconnection Blob					✓			
Load Balancing Information					✓			
Restricted Admin Mode					✓			

* = This setting can only be configured by an administrator.

To create a user bookmark in the web portal:

1. Log in to the Agentless VPN web portal.
2. Under *Bookmarks*, click *Create new bookmark for <username>*.
3. Enter a *Name*.
4. Select a *Protocol* type, and configure the type-based settings using the table above.
5. Click *OK*.

To create a predefined administrator bookmark in FortiOS:

1. Go to *VPN > Agentless VPN Portals*, and double-click a portal to edit it.
2. In the *Predefined Bookmarks* table, click *Create New*. The *New Bookmark* pane appears.
3. Enter a *Name*.
4. Select a protocol *Type*, and configure the settings based on its type.
5. Click *OK* to save the bookmark settings.
6. Click *OK* to save the portal settings.

Configuring group-based Agentless VPN bookmarks

Administrators can add bookmarks for users in the same user group. Agentless VPN will only output the matched group name entry to the client. This setting can only be configured in the CLI.

To add bookmarks for users in the same user group:

1. Enable group bookmarks in the web portal settings:

```
config vpn ssl web portal
    edit <name>
        set user-group-bookmark enable
    next
end
```

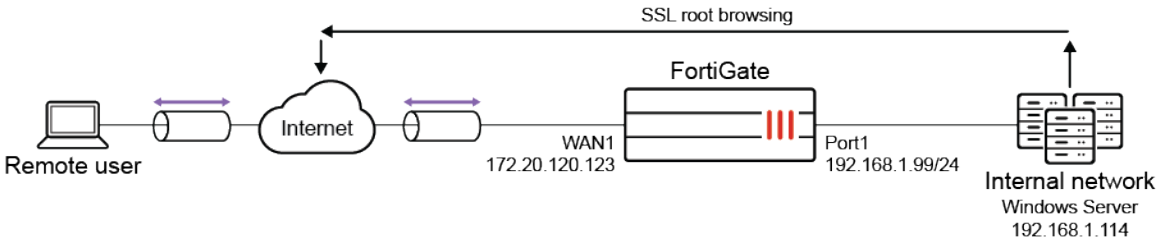
2. Configure the user group bookmark:

```
config vpn ssl web user-group-bookmark
    edit <name>
        config bookmarks
            edit <name>
                ...
            next
        end
    next
end
```

Agentless VPN for remote users

This is a sample configuration of remote users accessing the corporate network through an Agentless VPN web portal.

Sample topology



Sample configuration

WAN interface is the interface connected to ISP using a static IP. You can also use DHCP or PPPoE mode on WAN1 interface. The Agentless VPN's SSL tunnel is established over the WAN interface. By default, Agentless VPN settings are disabled and hidden from the GUI and CLI.

To enable Agentless VPN feature and Agentless VPN feature visibility in FortiOS:

1. Enable Agentless VPN using the CLI:

```
config system global
    set sslvpn-web-mode enable
end
```

2. Enable Agentless VPN feature visibility.
   a. In the GUI:
      i. Go to *System* > *Feature Visibility*.
      ii. In the *Core Features* section, enable *Agentless VPN*.
      iii. Click *Apply*.
   b. In the CLI:

```
config system settings
    set gui-sslvpn enable
end
```

To configure Agentless VPN using the GUI:

1. Configure the interface and firewall address. The port1 interface connects to the internal network.
   a. Go to *Network* > *Interfaces* and edit the *wan1* interface.
   b. Set *IP/Network Mask* to *172.20.120.123/255.255.255.0*.
   c. Edit *port1* interface and set *IP/Network Mask* to *192.168.1.99/255.255.255.0*.
   d. Click *OK*.
   e. Go to *Policy & Objects* > *Address* and create an address for internet subnet *192.168.1.0*.
2. Configure user and user group.
   a. Go to *User & Authentication* > *User Definition* to create a local user *agentlessvpnuser1*.
   b. Go to *User & Authentication* > *User Groups* to create a group *agentlessvpngroup* with the member *agentlessvpnuser1*.
3. Configure Agentless VPN portals.
   a. Go to *VPN* > *Agentless VPN Portals*, and click *Create New*.
   b. Enter a name, such as *agentless-portal*.
   c. Under *Predefined Bookmarks*, click *Create New*, and enter the following details:

Field	Value
Name	Windows Server
Type	RDP
Host	192.168.1.114
Port	3389
Single Sign-On	Disable
Username	Set the username to log in to Windows Server.
Password	Set the password to log in to Windows Server.
Port	3389

   d. Click *OK* to save the predefined bookmark.
   e. Click *OK* to save the portal settings.
   f. Create another Agentless VPN portal and name it *portal-access-disabled*.
   g. Click *OK* to save the portal settings.
   h. Disable Agentless VPN for the newly created portal (that is, *portal-access-disabled*) using CLI:

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
end
```

4. Configure Agentless VPN settings.
   a. Go to *VPN* > *Agentless VPN Settings*.
   b. Set *Agentless VPN* status to *Enable*.
   c. Set *Listen on Interface(s)* to *wan1*.

d. Set *Listen on Port* to *10443*.
e. Choose a certificate for *Server Certificate*. The default is *Fortinet_Factory*.
f. In *Authentication/Portal Mapping*, select *All Other Users/Groups*, and click *Edit*.
g. Use *Portal* dropdown to select *portal-access-disabled*.
h. Click *OK*.
i. In *Authentication/Portal Mapping*, click *Create New*.
    i. Set *Users/Groups* to *agentlessvpngroup*.
    ii. Set *Portal* to *agentless-portal*.
    iii. Click *OK*.
j. Click *Apply* to save the Agentless VPN settings.

5. Configure Agentless VPN firewall policy.
a. Go to *Policy & Objects > Firewall Policy*, and click *Create New*.
b. Set *Name* to *Agentless VPN firewall policy*.
c. Set *Schedule* to *always* and *Action* to *Accept*.
d. Set *Incoming interface* to *Agentless VPN interface (ssl.root)*.
e. Choose an *Outgoing Interface*. This example uses *port1*.
f. Set the *Source* to *all* and *User/group* to *agentlessvpngroup*.
g. In this example, the *Destination* is the internal, protected subnet *192.168.1.0*.
h. Set *Service* to *ALL*.
i. Click *OK*.

To configure Agentless VPN using the CLI:

1. Configure the interface and firewall address.

```
config system interface
    edit "wan1"
        set vdom "root"
        set ip 172.20.120.123 255.255.255.0
    next
end
```

2. Configure the internal interface and protected subnet, then connect the port1 interface to the internal network.

```
config system interface
    edit "port1"
        set vdom "root"
        set ip 192.168.1.99 255.255.255.0
    next
end
```

```
config firewall address
    edit "192.168.1.0"
        set subnet 192.168.1.0 255.255.255.0
    next
end
```

3. Configure user and user group.

```
config user local
    edit "agentlessvpnuser1"
        set type password
        set passwd your-password
    next
end
```

```
config user group
    edit "agentlessvpngroup"
        set member "vpnuser1"
    next
end
```

4. Configure Agentless VPN portal and predefine RDP bookmark for windows server.

```
config vpn ssl web portal
    edit "agentless-portal"
        set web-mode enable
        config bookmark-group
            edit "gui-bookmarks"
                config bookmarks
                    edit "Windows Server"
                        set apptype rdp
                        set host "192.168.1.114"
                        set port 3389
                        set logon-user "your-windows-server-user-name"
                        set logon-password your-windows-server-password
                    next
                end
            next
        end
    next
end
```

5. Configure another Agentless VPN portal and disable Agentless VPN on it.

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
end
```

6. Configure one Agentless VPN firewall policy to allow remote users to access the internal network. This policy does not allow traffic initiated from internal network to remote client.

```
config firewall policy
    edit 1
        set name "Agentless VPN firewall policy"
        set srcintf "ssl.root"
        set dstintf "port1"
        set action accept
```

```
        set srcaddr "all"
        set dstaddr "192.168.1.0"
        set service "ALL"
        set nat enable
        set groups "agentlessvpngroup"
    next
end
```

Do not set the virtual IP addresses as the destination address in a firewall policy when using Agentless VPN, as it will result in no destination address being accessible. Please note that the FortiOS Agentless VPN does not support mapping the virtual IP to the actual one.

To see the results:

1. In a web browser, log in to the portal *https://172.20.120.123:10443* using the credentials you've set up.
2. In the portal with the predefined bookmark, select the bookmark to begin an RDP session. If there are no predefined bookmarks, the Quick Connection tool can be used; see Quick Connection tool on page 3352 for more information.
3. On FortiGate, go to *Dashboard* > *Agentless VPN Monitor* to verify the list of Agentless VPN users.
   If the Agentless VPN Monitor is hidden, click *+* under the *Dashboard*, search for *Agentless VPN Monitor*, and add it to the display.
4. Go to *Log & Report* > *System Events*, and use the dropdown to select *VPN Events* to view VPN logs.

Customizing the RDP display size

The RDP display size (width and height settings) can be customized for Agentless VPN when creating a new connection or bookmark. Administrators can also specify the display size when preconfiguring bookmarks.

To configure the default window dimensions in an RDP Agentless VPN portal:

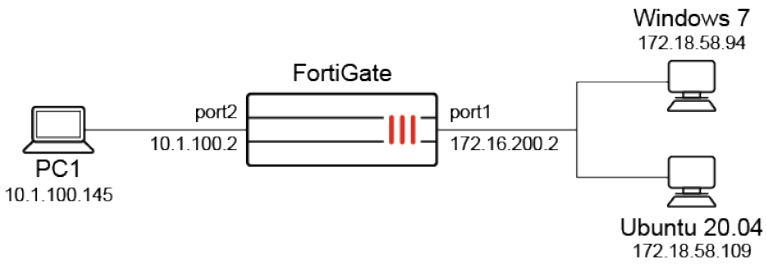
```
config vpn ssl web portal
    edit <name>
        set default-window-width <integer>
        set default-window-height <integer>
    next
end
```

`default-window-width <integer>`	Set the default RDP screen width, in pixels (0 - 65535, default = 1024).
`default-window-height <integer>`	Set the default RDP screen height, in pixels (0 - 65535, default = 768).

Example

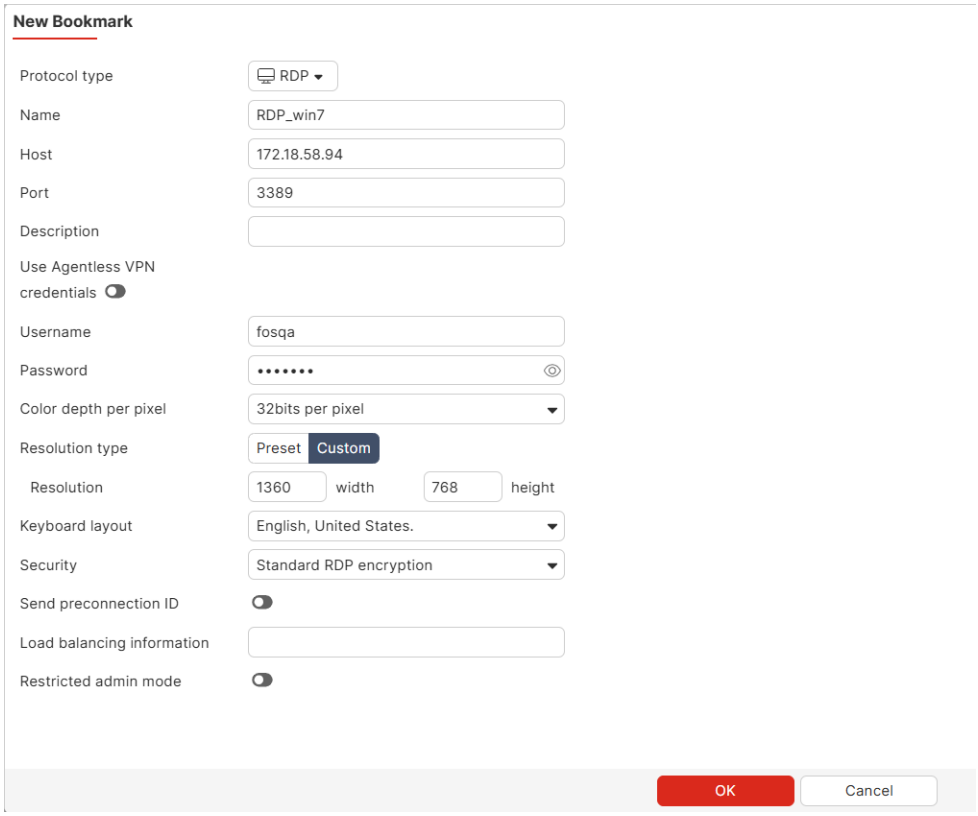
In this example, a user has a monitor with a resolution of 1920 × 1080. The user creates two bookmarks for RDP servers with different resolutions:

- Windows 7: 1360 × 768
- Ubuntu 20.04: 800 × 600

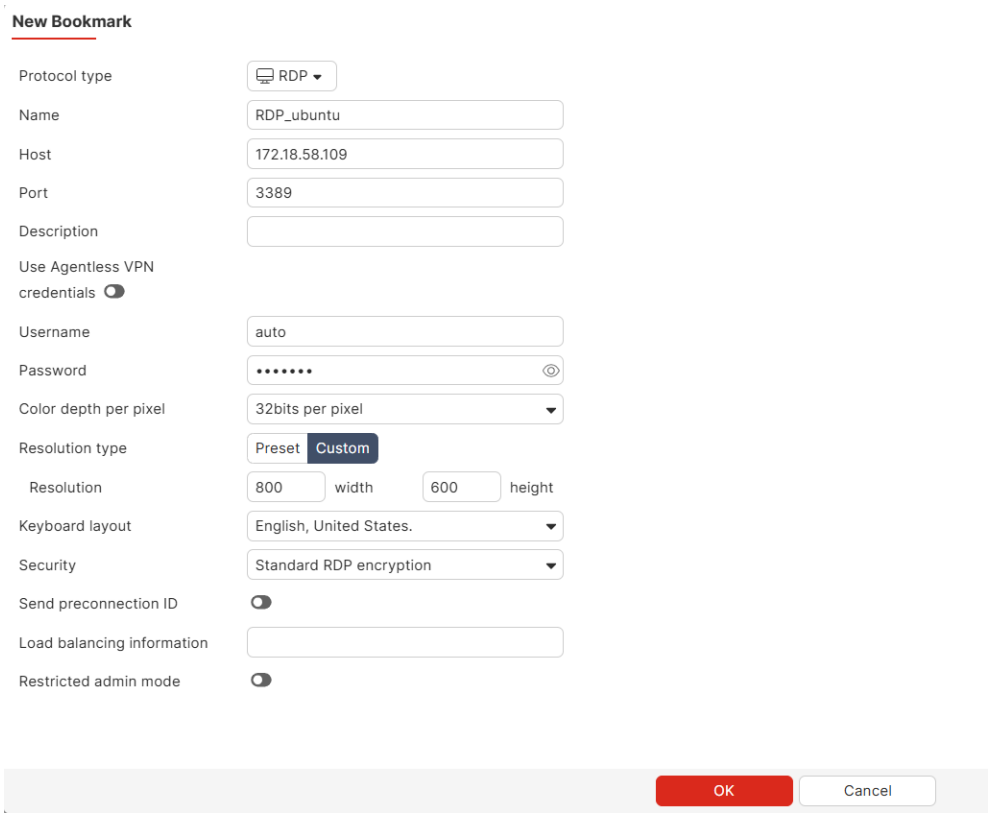


To customize the RDP bookmark display size:

1. Log in to the Agentless VPN web portal.
2. Under *Bookmarks*, click *Create new bookmark for <user>*.
3. Set *Protocol type* to *RDP*, and enter a suitable *Name*, *Host*, and *Port*.
4. Set the *Resolution width* and *height* fields as required.
   a. Windows 7: 1360 width and 768 height.



   b. Ubuntu 20.04: 800 width and 600 height.

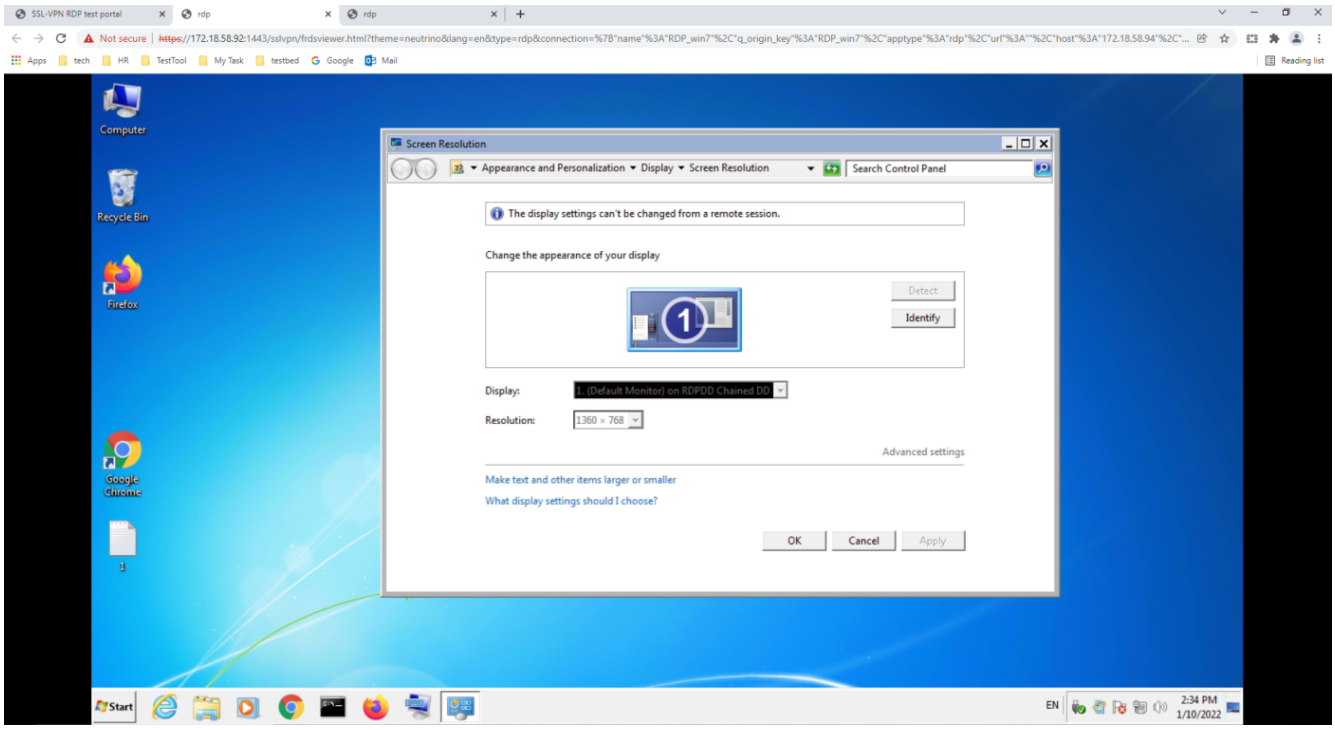


5. Click *OK*.

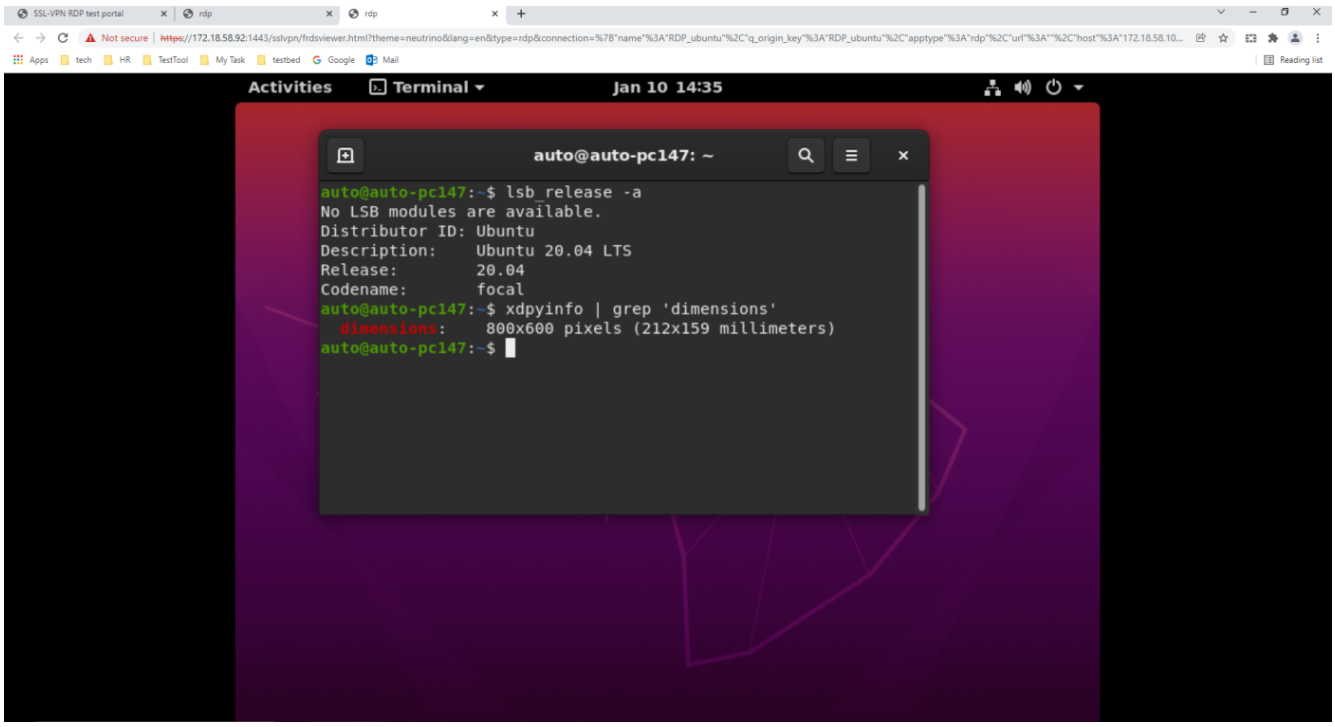
Verification:

When the user connects to the RDP servers using the bookmarks, the customized screen resolutions are applied regardless of the client PC's screen resolution (1920 × 1080).

Windows 7:



Ubuntu 20.04:



To view the bookmarks created by the user:

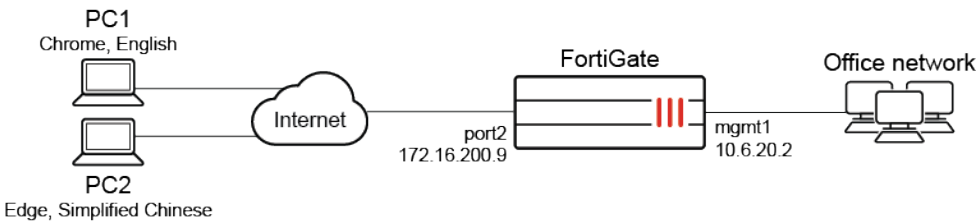
```
show vpn ssl web user-bookmark
config vpn ssl web user-bookmark
```

```
    edit "rdp_user#"
        config bookmarks
            edit "RDP_win7"
                set apptype rdp
                set host "172.18.58.94"
                set port 3389
                set logon-user "fosqa"
                set logon-password ********
                set color-depth 32
                set width 1360
                set height 768
            next
            edit "RDP_ubuntu"
                set apptype rdp
                set host "172.18.58.109"
                set port 3389
                set logon-user "auto"
                set logon-password ********
                set color-depth 32
                set width 800
                set height 600
            next
        end
    next
end
```

Showing the Agentless VPN web portal login page in the browser's language

By default, the browser's language preference is automatically detected and used by the Agentless VPN web portal login page. The system language can be used by changing the settings on the *Agentless VPN Settings* page of the GUI or disabling `browser-language-detection` in the CLI:

```
config vpn ssl settings
    set browser-language-detection disable
end
```



In this example, the *agentlessvpnadmin* user account is used for Agentless VPN connections on the *testportal1* Agentless VPN portal. The account is shared by users from different countries that use different browsers and different languages in their browsers. The user on PC1 uses Chrome in English, and the user on PC2 uses Edge

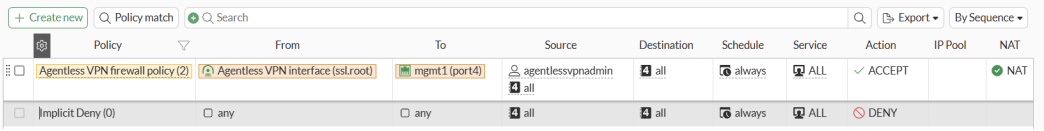
in Simplified Chinese. When a user logs in to the Agentless VPN portal, all pages are shown in the same language as their browser.

To configure the Agentless VPN portal to use the client's browser language:

1. Ensure Agentless VPN is enabled on *testportal1* portal.

```
config vpn ssl web portal
    edit "testportal1"
        set web-mode enable
    next
end
```

2. Set the language preference:
   a. Go to *VPN* > *Agentless VPN Settings*.
   b. Set *Language* to *Browser Preference*.
   c. Click *Apply*.
3. Add the *agentlessvpnadmin* user to the policy used by the Agentless VPN portal.



4. Confirm that the configuration works:
   - When the user on PC1 logs in to the Agentless VPN portal using Chrome in English, all of the pages are shown in English.
   - When the user on PC2 logs in to the Agentless VPN portal using Edge in Simplified Chinese, all of the pages are shown in Simplified Chinese.

Agentless VPN custom landing page

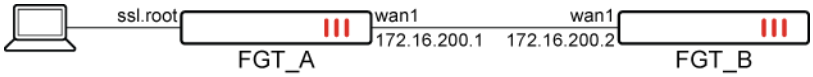
An Agentless VPN user can log in to the Agentless VPN portal and be redirected to a custom landing page. The custom landing page can be configured in *VPN* > *Agentless VPN Portals* by setting the portal *Landing page* to *Custom* or by using the command `config landing-page`.

The landing page can accept SSO credentials as well as SSO from form data. This allows administrators to streamline web application access for their users. The custom redirected portal has a logout button so that when users log out from the web application, they are also logged out from the Agentless VPN connection.

Example

In the following example, the Agentless VPN portal settings are configured so that the URL of the custom landing page of FGT_A is set to the FGT_B login page. Therefore, when a web user is logging into FGT_A's Agentless VPN portal, they will automatically be redirected to FGT_B, where the SSO username and password

are passed into the username and password input fields. This allows for single sign on of the connecting user into FGT_B through the Agentless VPN.



To configure a custom landing page from the CLI:

1. Configure the user and user group:

```
config user local
    edit "custom_landing_user"
        set type password
        set passwd ********
    next
end
config user group
    edit "ssl-web-group"
        set member "custom_landing_user"
    next
end
```

2. Configure the Agentless VPN portal:

```
config vpn ssl web portal
    edit "custom_landing"
        set web-mode enable
        set landing-page-mode enable
        config landing-page
            set url "https://172.16.200.2/login"
            set sso static
            config form-data
                edit "username"
                    set value "admin"
                next
                edit "secretkey"
                    set value "1"
                next
            end
            set sso-credential alternative
            set sso-username "admin"
            set sso-password ********
        end
    next
end
```

3. Configure the Agentless VPN settings:

```
config vpn ssl settings
    set banned-cipher SHA1 SHA256 SHA384
    set servercert "fgt_gui_automation"
    set port 1443
```

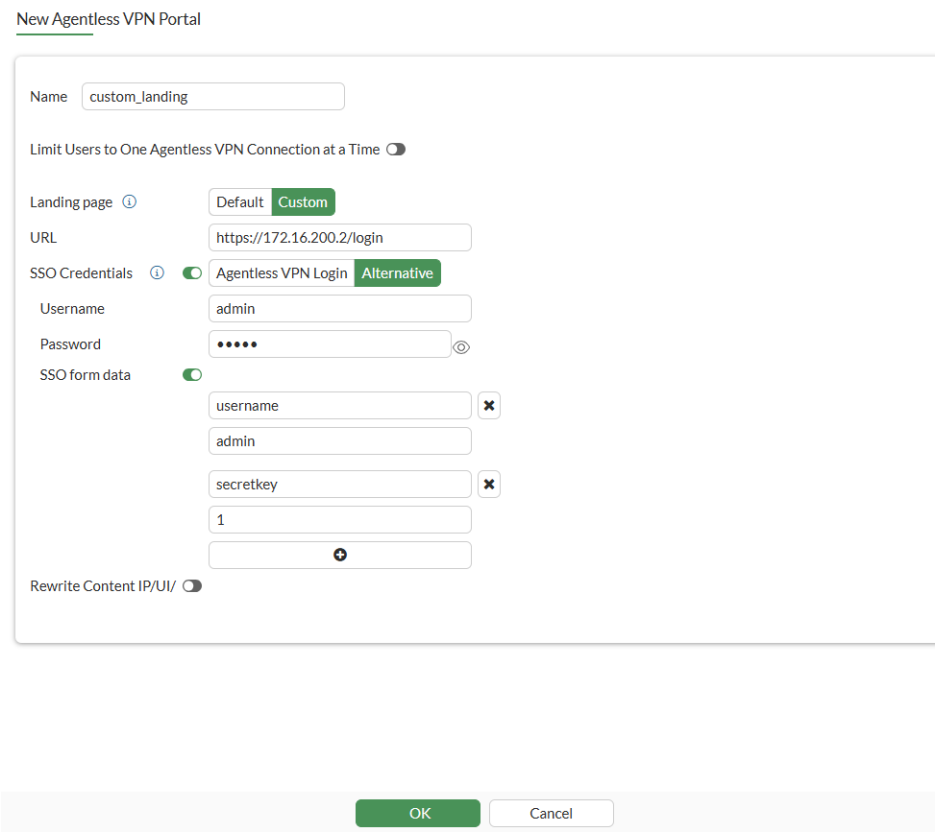
```
    set source-interface "port1"
    set source-address "all"
    set source-address6 "all"
    set default-portal "portal-access-disabled"
    config authentication-rule
        edit 1
            set groups "ssl-web-group"
            set portal "custom_landing"
        next
    end
    set encrypt-and-store-password enable
end
```

4. Configure the firewall policy:

```
config firewall policy
    edit 1
        set name "testpolicy"
        set srcintf "ssl.root"
        set dstintf "wan1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set nat enable
        set groups "ssl-web-group"
    next
end
```

To configure a custom landing page from the GUI:

1. Configure the user and user group:
   a. Go to *User & Authentication > User Definition* to create the `custom_landing_user` user.
   b. Go to *User & Authentication > User Groups* to create the `ssl-web-group` user group with the member `custom_landing_user`.
2. Configure the Agentless VPN portal:
   a. Go to *VPN > Agentless VPN Portals*.
   b. Click *Create New*.
   c. Enter `custom_landing` as the *Name*.
   d. Set *Landing Page* to *Custom*.
   e. Enter the FGT_B login page *URL*.
   f. Enable *SSO Credentials* and select *Alternative*.
   g. Enable *SSO form data* and enter the form keys and values.



   h. Click *OK*.

3. Configure the Agentless VPN settings:
   a. Go to *VPN > Agentless VPN Settings*.
   b. Set *Listen on Interface(s)* to *port1*.
   c. Set *Listen on Port* to *1443*.
   d. Set *Server Certificate* to *fgt_gui_automation*.
   e. Create a new *Authentication/Portal Mapping* for group *ssl-web-group* mapping the portal *custom-landing*.
   f. Click *Apply*.
4. Configure the firewall policy:
   a. Go to *Policy & Objects > Firewall Policy* and click *Create New*.
   b. Configure the following settings:

Name	testpolicy
Incoming Interface	ssl.root
Outgoing Interface	wan1
Source	all
User/group	ssl-web-group
Destination	all

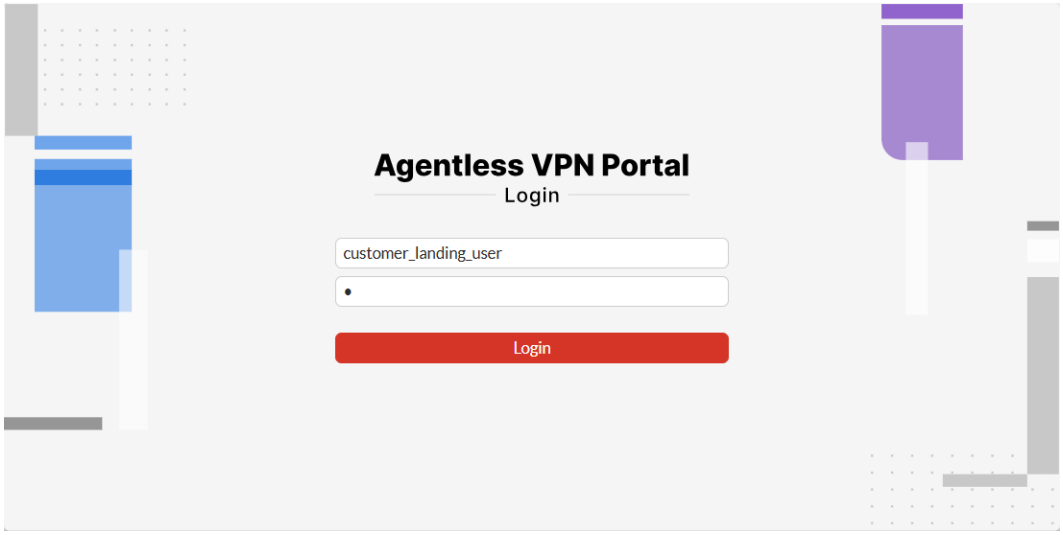
Schedule	always
Service	ALL
Action	ACCEPT

c. Enable *NAT*.
d. Enable *Log Allowed Traffic* and set it to *All Sessions*.
e. Click *OK*.

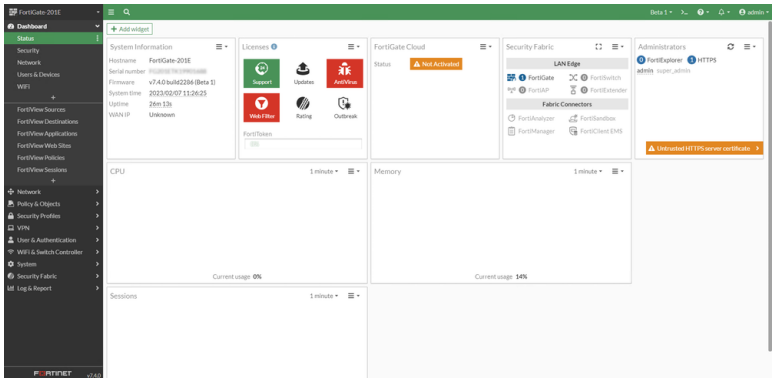
Once the Agentless VPN portal is configured, the connected user can access FGT_B through the FGT_A Agentless VPN portal.

To access FGT_B through the FGT_A Agentless VPN portal:

1. Enter your SSO credentials in the Agentless VPN login fields.



The landing page is redirected to the FGT_B GUI automatically.



Agentless VPN protocols

The following topics provide information about Agentless VPN protocols:

- TLS 1.3 support on page 3371
- SMBv2 support on page 3372

TLS 1.3 support

FortiOS supports TLS 1.3 for Agentless VPN.

TLS 1.3 support requires IPS engine 4.205 or later and browsers that support TLS 1.3.

To establish a TLS 1.3 connection to Agentless VPN:

1. Enable TLS 1.3 support using the CLI:

```
config vpn ssl setting
    set ssl-max-proto-ver tls1-3
    set ssl-min-proto-ver tls1-3
end
```

2. Configure the Agentless VPN settings. See Agentless VPN for remote users on page 3357.
3. Configure the firewall policy. See Firewall policy on page 1842.
4. For Linux clients, ensure OpenSSL 1.1.1a is installed:
   a. Run the following commands in the Linux client terminal:

```
root@PC1:~/tools# openssl
OpenSSL> version
```

   If OpenSSL 1.1.1a is installed, the system displays a response like the following:

```
OpenSSL 1.1.1a  20 Nov 2018
```

5. For Linux clients, use OpenSSL with the TLS 1.3 option to connect to Agentless VPN:
   a. Run the following command in the Linux client terminal:

```
#openssl s_client -connect 10.1.100.10:10443 -tls1_3
```

6. Ensure the Agentless VPN connection is established with TLS 1.3 using the CLI:

```
# diagnose debug application sslvpn -1
# diagnose debug enable
```

   The system displays a response like the following:

```
[207:root:1d]SSL established: TLSv1.3 TLS_AES_256_GCM_SHA384
```

SMBv2 support

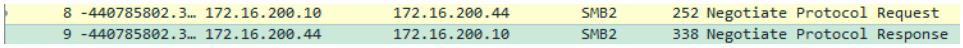
On all FortiGate models, SMBv2 is enabled by default for Agentless VPN.

To configure SMBv2:

1. Set the minimum and maximum SMB versions.

```
config vpn ssl web portal
    edit portal-name
       set smb-min-version smbv2
       set smb-max-version smbv3
    next
end
```

2. Configure the Agentless VPN settings. See Agentless VPN for remote users on page 3357.
3. Configure the firewall policy. See Firewall policy on page 1842.
4. Connect to the Agentless VPN web portal, and create an SMB bookmark for the SMBv2 server.
5. Click the bookmark to connect to the SMBv2 server.
6. On the FortiGate, use package capture to verify that SMBv2 works:



Agentless VPN security best practices

Securing remote access to network resources is a critical part of security operations. Agentless VPN allows administrators to configure, administer, and deploy a remote access strategy for their remote workers. Applying the proper levels of security are integral to providing optimal performance and user experience and keeping user data safe.

The guidelines in this topic outline how to employ security best practices to protect your data.

Information about Agentless VPN throughput and maximum concurrent users is available on your device's datasheet; see Next-Generation Firewalls Models and Specifications.

Do not set the virtual IP addresses as the destination address in a firewall policy when using Agentless VPN, as it will result in no destination address being accessible. Please note that the FortiOS Agentless VPN does not support mapping the virtual IP to the actual one.

Ensure you always upgrade your FortiGate to the latest FortiOS firmware version. This ensures you are running the latest Agentless VPN security enhancements to protect your VPN deployment.

Agentless VPN settings on page 3373

- Define your minimum supported TLS version and cipher suites on page 3373
- Limit log in attempts and block duration on page 3373

- Limit users to one Agentless VPN session at a time on page 3374
- Use a custom listening port for Agentless VPN on page 3374
- Disable Agentless VPN on page 3374
- Enable server hostname on page 3375

Authentication on page 3375

- Integrate with authentication servers on page 3375
- Use a non-factory SSL certificate for the Agentless VPN portal on page 3375
- Use multi-factor authentication on page 3375
- Use multi-factor authentication with authentication servers on page 3376
- Disable case sensitivity for remote users using authentication servers and MFA on page 3376
- Deploy user certificates for remote Agentless VPN users on page 3376

Authorization on page 3376

- Properly administer firewall policies and profiles against only the access level required for the remote user on page 3376
- Set the default portal to a custom VPN portal with Agentless VPN disabled on page 3377
- Limit incoming access to specific hosts or geography based addresses on page 3377
- Limit incoming access using local-in policies with specific hosts, geography based addresses, or schedules on page 3378
- Limit incoming access using a virtual IP, loopback interface, and firewall policy with Internet Services or an external feed or schedule on page 3379

Agentless VPN settings

Define your minimum supported TLS version and cipher suites

Minimum and maximum supported TLS version can be configured in the FortiGate CLI. The cipher algorithm can also be customized.

See How to control the SSL version and cipher suite for Agentless VPN for more information.

Limit log in attempts and block duration

To prevent brute force attacks, limit log in attempts and configure the block duration:

```
config vpn ssl settings
   set login-attempt-limit 2
   set login-block-time 60
   set login-timeout 30
end
```

These are the default values. The FortiGate will block attempts to connect to Agentless VPN for 60 seconds after two unsuccessful log in attempts.

This configuration enhances Agentless VPN security by limiting failed login attempts (`login-attempt-limit 2`), temporarily blocking an IP for 60 seconds after two consecutive failed attempts (`login-block-time 60`), and defining a 30-second window for consecutive login attempts (`login-timeout 30`). If two failed attempts occur within 30 seconds of each other, they are considered consecutive and trigger a block. However, if the second attempt occurs after 30 seconds, it resets the login attempt count. These settings help mitigate brute-force attacks by restricting repeated login attempts within a short period.

Modify the values as needed for your requirements.

Limit users to one Agentless VPN session at a time

To prevent attacks from a compromised user, you can limit a user to one Agentless VPN session at a time by going to *VPN > Agentless VPN Portals*, editing a portal, and enabling *Limit Users to One Agentless VPN Connection at a Time*. This option can also be configured in the CLI:

```
config vpn ssl web portal
    edit < portal name >
        set limit-user-logins enable
    end
end
```

Use a custom listening port for Agentless VPN

To prevent external attacks targeting the default Agentless VPN port 443, use a custom port for Agentless VPN.

The Agentless VPN listening port can be configured from the GUI on the *VPN > Agentless VPN Settings* page by changing the *Listen on Port* field from the default 443 to any other port. To change the listening port in the CLI:

```
config vpn ssl settings
    set port <port number>
end
```

After changing the Agentless VPN listening port, ensure that end users are informed of the new port. Users must update the Agentless VPN web portal URL in their browsers, replacing 443 with the newly assigned port.

Make sure that the selected port is not in use by other services on the firewall.

Disable Agentless VPN

After the Agentless VPN settings have been configured, Agentless VPN can be disabled when not in use.

To disable Agentless VPN in the GUI:

1. Go to *VPN > Agentless VPN Settings*.
2. Set *Agentless VPN status* to *Disable*.
3. Click *Apply*.

If the FortiGate has VDOMs configured, select the appropriate VDOM, and repeat the steps to disable Agentless VPN for that specific VDOM.

See How to disable Agentless VPN functionality on FortiGate for more information.

Enable server hostname

Enabling server hostname ensures that if a page redirection occurs, the FortiGate server hostname is used in the host field of the HTTP header instead of a client-provided host field.

```
config vpn ssl settings
    set server-hostname <redirect host name>
end
```

Authentication

Integrate with authentication servers

For networks with many users, integrate your user configuration with existing authentication servers through LDAP, RADIUS, or FortiAuthenticator. When integrating with existing authentication servers, these users are referred to as remote users.

By integrating with existing authentication servers, such as Windows AD, there is a lower chance of making mistakes when configuring remote users and remote user groups, reducing your administration effort. Also, credentials for remote users are kept on the authentication servers themselves and are not stored on the FortiGate, unlike credentials for local users.

See Agentless VPN with LDAP user authentication on page 3380 and Agentless VPN with RADIUS on Windows NPS on page 3435 for more information.

It is best practice to integrate with encrypted protocols on authentication servers such as LDAPS instead of LDAP, and RADSEC over TLS instead of RADIUS. See Configuring client certificate authentication on the LDAP server on page 3524 and Configuring a RADSEC client on page 3566 for more information.

Use a non-factory SSL certificate for the Agentless VPN portal

Your certificate should identify your domain so that a remote user can recognize the identity of the server or portal that they are accessing through a trusted CA.

The default Fortinet factory self-signed certificates are provided to simplify initial installation and testing. If you use these certificates you are vulnerable to man-in-the-middle attacks, where an attacker spoofs your certificate, compromises your connection, and steals your personal information. It is highly recommended that you purchase a server certificate from a trusted CA to allow remote users to connect to Agentless VPN with confidence. See Procuring and importing a signed SSL certificate on page 3974 for more information.

Use multi-factor authentication

Multi-factor authentication (MFA) ensures that the end-user is who they claim to be by requiring at least two factors - a piece of information that the user knows (password), and an asset that the user has (OTP). A third

factor, something a user is (fingerprint or face), may be enabled as well. FortiToken Mobile is typically used for MFA.

FortiGate comes with two free FortiTokens, and more can be purchased from the FortiToken Mobile iOS app or through Fortinet partners.

See Agentless VPN with FortiToken mobile push authentication on page 3413 for more information.

2FA, a subset of MFA, can also be set up with email tokens. See Email Two-Factor Authentication on FortiGate for information.

Use multi-factor authentication with authentication servers

Users configured with MFA, such as FortiToken Mobile tokens and email tokens on the FortiGate, are still essentially local users with user credentials stored on the FortiGate.

You should therefore consider using a combination of MFA and authentication servers for optimal security. See Agentless VPN with LDAP-integrated certificate authentication on page 3398, Agentless VPN with RADIUS and FortiToken mobile push on FortiAuthenticator on page 3425, and RADIUS integrated certificate authentication for Agentless VPN on page 3571 for more information.

Disable case sensitivity for remote users using authentication servers and MFA

When using remote users with authentication servers and MFA, it is possible for MFA to be bypassed if the case of the entered username is not an exact match. To prevent this, ensure case sensitivity is disabled for each remote user that has been configured on the FortiGate with authentication server and MFA settings. See Agentless VPN for remote users with MFA and user sensitivity on page 3405 for more information.

Deploy user certificates for remote Agentless VPN users

This method of 2FA uses a user certificate as the second authentication factor. This is more secure, as it identifies the end user using a certificate. The configuration and administration of this solution is significantly more complicated, and requires administrators with advanced knowledge of the FortiGate and certificate deployment.

See Agentless VPN with certificate authentication on page 3392 for more information.

Authorization

Properly administer firewall policies and profiles against only the access level required for the remote user

Users do not all require the same access. Access should only be granted after careful considerations. Typically, users are placed in groups, and each group is allowed access to limited resources.

Using Agentless VPN simplifies defining the control structure for mapping users and groups to the appropriate resources.

See Agentless VPN multi-realm on page 3458 for more information.

Set the default portal to a custom VPN portal with Agentless VPN disabled

In the Agentless VPN settings, it is mandatory in the *Authentication/Portal Mapping* section to configure a portal for *All Other Users/Groups* or what can be considered a default portal for other users who are not specifically mapped to access Agentless VPN portals.

When the *Authentication/Portal Mapping* does not match users or groups that you have specifically mapped to access Agentless VPN portals, you can create a custom Agentless VPN portal and disable `web-mode` using the CLI, and then set default portal to that custom portal. This configuration is analogous to the implicit deny policy in firewall policies, in that this custom portal can deny all other users and groups.

For a default portal with `web-mode` disabled, users deemed part of *All Other Users/Groups* can still access the Agentless VPN portal. Once these users try to log in to the portal, access will be denied. Users deemed part of *All Other Users/Groups* should be unable to successfully establish Agentless VPN connection in this case.

To disable web-mode in custom Agentless VPN portal and apply it in Agentless VPN settings:

1. Using CLI, edit the Agentless VPN web portal and disable web mode.

```
config vpn ssl web portal
    edit "custom-web-portal"
        set web-mode disable
    next
end
```

2. Apply the custom web portal to Agentless VPN:
   - Using the GUI:
     i. Go to *VPN* > *Agentless VPN Settings*.
     ii. Under *Authentication/Portal Mapping*, click *All Other Users/Groups* > *Edit*.
     iii. Under *Portal* dropdown, select *custom-web-portal*.
     iv. Click *OK*.
     v. Click *Apply*.
   - Using the CLI:

```
config vpn ssl settings
    set default-portal "custom-web-portal"
end
```

Limit incoming access to specific hosts or geography based addresses

The simplest method for limiting incoming access is specifying source addresses in the Agentless VPN settings.

- If you require schedules and geography based addresses, and are comfortable using the CLI, then consider configuring local-in policies as described in To limit incoming access to specific hosts based on their source IP addresses in the CLI.
- If you require Internet Services using ISDB or external feeds, or schedules and geography based addresses, and are more comfortable using the GUI, then consider configuring a VIP, loopback, and firewall policy as described in To limit incoming access to specific hosts based on their source IP addresses in the GUI.

To limit incoming access to specific hosts based on their source IP addresses in the GUI:

1. Create address objects or groups with the source IP addresses you need to allow access to Agentless VPN.
2. In *VPN > Agentless VPN Settings* under *Restrict Access*, select *Limit access to specific hosts* and in the *Hosts* field select address objects or groups corresponding to specific source IP addresses for hosts that you need to allow.
3. To block specific source IPs while allowing all others, enable *Negate Source*, and select the address objects or groups representing the restricted hosts.

To limit incoming access to specific hosts based on their source IP addresses in the CLI:

```
config vpn ssl settings
    set source-address <source address 1> ... <source address n>
    set source-address-negate {disable | enable}
end
```

With this CLI, you can limit incoming access to hosts from specific countries by specifying geography-based addresses as the source addresses. See Geography based addresses on page 2042 for more information.

Limit incoming access using local-in policies with specific hosts, geography based addresses, or schedules

As an alternative to configuring source addresses in the Agentless VPN settings, you can configure local-in policies to allow and deny specific source addresses.

The configuration workflow is:

1. Create a new custom service corresponding to the Agentless VPN listening TCP port.
2. (Optional) Create new geography based addresses that can be specified as the source addresses for the allow and deny local-in policies that are created.
3. (Optional) Configure new schedules that can be specified as the schedule for the allow and deny local-in policies that are created.
4. Create a local-in policy using the Agentless VPN custom service to allow specific source addresses.
5. Create a local-in policy using the Agentless VPN custom service to deny specific source addresses. This is required since there is no implicit deny local-in policy defined by default.

Note that extra care should be taken when configuring a local-in policy, as an incorrect configuration could inadvertently deny traffic for IPsec VPN, dynamic routing protocols, HA, and other FortiGate features.

The source IP address objects that are used can be address objects or groups, or geography based address objects. New schedules can be created and applied to the local-in policies to impose schedule-based access restrictions.

You can also deny all access to Agentless VPN by creating a deny local-in policy using source address `all` and Agentless VPN custom service without creating a corresponding local-in policy to allow the Agentless VPN custom service.

Limit incoming access using a virtual IP, loopback interface, and firewall policy with Internet Services or an external feed or schedule

Agentless VPN access can be moved to a secondary IP address or any other WAN IP address defined on a FortiGate interface by using a virtual IP (VIP), loopback interface, and WAN-to-loopback firewall policy.

The configuration workflow is as follows:

1. Configure the Agentless VPN firewall policy from the `ssl.<VDOM>` interface to the internal interface or interfaces, ensuring that you specify the users and groups (required), and addresses or address groups (optional) in the policy's *Source* field. See Agentless VPN for remote users on page 3357.
2. Create a new loopback interface and specify an IP address that is not being used by any other interface on the FortiGate. This IP address can be a private IP address within the RFC 1918 range.
3. Create a new VIP with the following settings:
   - *External IP address/range* configured as the secondary WAN IP address, or any other WAN IP address that is available for the WAN interface.
   - *IPv4 address/range* configured as the IP address assigned to the loopback interface.
   - *Port forwarding* enabled with *External service port* and *Map to IPv4 port* with the Agentless VPN listening port.
4. Create a new WAN interface to loopback interface firewall policy with the VIP as the destination address.
5. Under *VPN > Agentless VPN Settings*, set *Listen on Interface(s)* to the newly created loopback interface.
6. (Optional) Create a new deny firewall policy, configure Internet Services as source addresses in the new policy, and place it above the WAN-to-loopback firewall policy.
7. (Optional) Create a new deny firewall policy, configure an IP address external feed, configure the external feed as a source address in the new policy, and place it above the WAN-to-loopback firewall policy.
8. (Optional) Configure a new schedule and apply it to the WAN-to-loopback firewall policy.

This configuration gives you the option to deny access to Agentless VPN from specified hosts. You can create a new deny firewall policy, apply Internet Services using the ISDB or external feed objects as source addresses and place the deny policy above the WAN-to-loopback firewall policy.

You also have the option of creating a new schedule and applying it to the WAN-to-loopback firewall policy to allow Agentless VPN access during a specified time or days of the week.

Finally, you can disable all Agentless VPN access by disabling the WAN-to-loopback firewall policy.

Agentless VPN authentication

The following topics provide instructions on configuring Agentless VPN authentication:

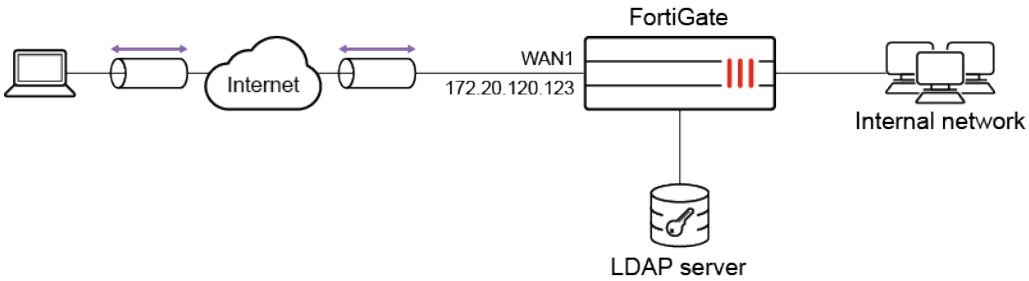
- Agentless VPN with LDAP user authentication on page 3380
- Agentless VPN with LDAP user password renew on page 3386
- Agentless VPN with certificate authentication on page 3392
- Agentless VPN with LDAP-integrated certificate authentication on page 3398
- Agentless VPN for remote users with MFA and user sensitivity on page 3405
- Agentless VPN with FortiToken mobile push authentication on page 3413
- Agentless VPN with RADIUS on FortiAuthenticator on page 3419
- Agentless VPN with RADIUS and FortiToken mobile push on FortiAuthenticator on page 3425
- Agentless VPN with RADIUS password renew on FortiAuthenticator on page 3430
- Agentless VPN with RADIUS on Windows NPS on page 3435
- Agentless VPN with multiple RADIUS servers on page 3441
- Agentless VPN with local user password policy on page 3452
- Agentless VPN multi-realm on page 3458
- NAS-IP support per Agentless VPN realm on page 3464
- Agentless VPN with Okta as SAML IdP on page 3466
- Agentless VPN with Microsoft Entra SSO integration on page 3472

Agentless VPN with LDAP user authentication

This is a sample configuration of Agentless VPN for LDAP users. In this example, the LDAP server is a Windows 2012 AD server. A user *ldu1* is configured on Windows 2012 AD server.

You must have generated and exported a CA certificate from the AD server and then have imported it as an external CA certificate into the FortiGate.

Sample topology



Sample configuration

WAN interface is the interface connected to ISP. This example shows static mode. You can also use DHCP or PPPoE mode. The Agentless VPN connection is established over the WAN interface.

To configure Agentless VPN using the GUI:

1. Configure the interface and firewall address. The port1 interface connects to the internal network:
   a. Go to *Network* > *Interfaces* and edit the *wan1* interface.
   b. Set *IP/Network Mask* to *172.20.120.123/255.255.255.0*.
   c. Edit *port1* interface and set *IP/Network Mask* to *192.168.1.99/255.255.255.0*.
   d. Click *OK*.
   e. Go to *Policy & Objects* > *Address* and create an address for internet subnet *192.168.1.0*.
2. Import CA certificate into FortiGate:
   a. Go to *System* > *Features Visibility* and ensure *Certificates* is enabled.
   b. Go to *System* > *Certificates* and select *Import* > *CA Certificate*.
   c. Select *Local PC* and then select the certificate file.
      The CA certificate now appears in the list of *External CA Certificates*. In this example, it is called *CA_Cert_1*.
   d. If you want, you can use CLI commands to rename the system-generated *CA_Cert_1* to be more descriptive:

```
config vpn certificate ca
    rename CA_Cert_1 to LDAPS-CA
end
```

3. Configure the LDAP user:
   a. Go to *User & Authentication* > *LDAP Servers* and click *Create New*.
   b. Specify *Name* and *Server IP/Name*.
   c. Specify *Common Name Identifier* and *Distinguished Name*.
   d. Set *Bind Type* to *Regular*.
   e. Specify *Username* and *Password*.
   f. Set *Protocol* to *LDAPS*.
   g. Enable *Certificate* and select *LDAPS-CA* from the list.
4. Configure user group:
   a. Go to *User & Authentication* > *User Groups* to create a user group.
   b. Enter a *Name*.
   c. In *Remote Groups*, click *Add* to add *ldaps-server*.
5. Configure Agentless VPN portals.
   a. Go to *VPN* > *Agentless VPN Portals*, and click *Create New*.
   b. Enter a name, such as *agentless-portal*.
   c. Under *Predefined Bookmarks*, click *Create New*, and enter the following details:

Field	Value
Name	Windows Server
Type	RDP
Host	192.168.1.114

Field	Value
Port	3389
Single Sign-On	Disable
Username	Set the username to log in to Windows Server.
Password	Set the password to log in to Windows Server.
Port	3389

d. Click *OK* to save the predefined bookmark.
e. Click *OK* to save the portal settings.
f. Create another Agentless VPN portal and name it *portal-access-disabled*.
g. Click *OK* to save the portal settings.
h. Disable Agentless VPN for the newly created portal (that is, *portal-access-disabled*) using CLI:

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
end
```

6. Configure Agentless VPN settings.
   a. Go to *VPN > Agentless VPN Settings*.
   b. Set *Agentless VPN* status to *Enable*.
   c. Set *Listen on Interface(s)* to *wan1*.
   d. Set *Listen on Port* to *10443*.
   e. Choose a certificate for *Server Certificate*. The default is *Fortinet_Factory*.
   f. In *Authentication/Portal Mapping*, select *All Other Users/Groups*, and click *Edit*.
   g. Use *Portal* dropdown to select *portal-access-disabled*.
   h. Click *OK*.
   i. In *Authentication/Portal Mapping*, click *Create New*.
      i. Set *Users/Groups* to *ldaps-server*.
      ii. Set *Portal* to *agentless-portal*.
      iii. Click *OK*.
   j. Click *Apply* to save the Agentless VPN settings.
7. Configure Agentless VPN firewall policy.
   a. Go to *Policy & Objects > Firewall Policy*, and click *Create New*.
   b. Set *Name* to *Agentless VPN firewall policy*.
   c. Set *Schedule* to *always* and *Action* to *Accept*.
   d. Set *Incoming interface* to *Agentless VPN interface (ssl.root)*.
   e. Choose an *Outgoing Interface*. This example uses *port1*.
   f. Set the *Source* to *all* and *User/group* to *ldaps-group*.
   g. In this example, the *Destination* is the internal, protected subnet *192.168.1.0*.
   h. Set *Service* to *ALL*.
   i. Enable *NAT*.

j. Configure any remaining firewall and security options as desired.
k. Click *OK*.

To configure Agentless VPN using the CLI:

1. Configure the interface and firewall address:

```
config system interface
    edit "wan1"
        set vdom "root"
        set ip 172.20.120.123 255.255.255.0
    next
end
```

2. Configure internal interface and protected subnet, then connect the port1 interface to the internal network:

```
config system interface
    edit "port1"
        set vdom "root"
        set ip 192.168.1.99 255.255.255.0
    next
end
```

```
config firewall address
    edit "192.168.1.0"
        set subnet 192.168.1.0 255.255.255.0
    next
end
```

3. Import CA certificate into FortiGate:
   a. Go to *System* > *Features Visibility* and ensure *Certificates* is enabled.
   b. Go to *System* > *Certificates* and select *Import* > *CA Certificate*.
   c. Select *Local PC* and then select the certificate file.
      The CA certificate now appears in the list of *External CA Certificates*. In the example, it is called *CA_Cert_1*.
   d. If you want, you can use CLI commands to rename the system-generated *CA_Cert_1* to be more descriptive:

```
config vpn certificate ca
    rename CA_Cert_1 to LDAPS-CA
end
```

4. Configure the LDAP server:

```
config user ldap
    edit "ldaps-server"
        set server "172.20.120.161"
        set cnid "cn"
        set dn "cn=Users,dc=qa,dc=fortinet,dc=com"
        set type regular
        set username "CN=Administrator,cn=users,DC=qa,DC=fortinet,DC=com"
```

```
        set password **********
        set group-member-check group-object
        set secure ldaps
        set ca-cert "LDAPS-CA"
        set port 636
    next
end
```

5. Configure user group:

```
config user group
    edit "ldaps-group"
        set member "ldaps-server"
    next
end
```

6. Configure Agentless VPN portal and predefine RDP bookmark for windows server.

```
config vpn ssl web portal
    edit "agentless-portal"
        set web-mode enable
        config bookmark-group
            edit "gui-bookmarks"
                config bookmarks
                    edit "Windows Server"
                        set apptype rdp
                        set host "192.168.1.114"
                        set port 3389
                        set logon-user "your-windows-server-user-name"
                        set logon-password your-windows-server-password
                    next
                end
            next
        end
    next
end
```

7. Configure another Agentless VPN portal and disable Agentless VPN on it.

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
end
```

8. Configure Agentless VPN settings:

```
config vpn ssl settings
    set banned-cipher SHA1 SHA256 SHA384
    set servercert "server_certificate"
    set port 1443
    set source-interface "wan1"
```

```
    set source-address "all"
    set default-portal "portal-access-disabled"
    config authentication-rule
        edit 1
            set groups "ldaps-group"
            set portal "agentless-portal"
        next
    end
end
```

9. Configure one Agentless VPN firewall policy to allow remote user to access the internal network. The policy does not allow traffic initiated from internal network to remote client:

```
config firewall policy
    edit 1
        set name "Agentless VPN firewall policy"
        set srcintf "ssl.root"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "192.168.1.0"
        set action accept
        set schedule "always"
        set service "ALL"
        set nat enable
        set groups "ldaps-group"
    next
end
```

To see the results of Agentless VPN portal access:

1. From a remote device, use a web browser to log into the Agentless VPN portal *http://172.20.120.123:10443*.
2. Enter the *ldu1* user credentials, then click *Login*.
3. On FortiGate, go to *Dashboard > Agentless VPN Monitor* to verify the list of Agentless VPN users.
   If the Agentless VPN Monitor is hidden, click *+* under the *Dashboard*, search for *Agentless VPN Monitor*, and add it to the display.
4. Go to *Log & Report > System Events*, and use the dropdown to select *VPN Events* to view VPN logs.

To check the Agentless VPN portal login using the FortiOS CLI:

```
# get vpn ssl monitor
Agentless VPN Login Users:
 Index   User    Auth Type    Timeout    From           HTTP in/out  HTTPS in/out
 0       ldu1    1(1)         229        10.1.100.254   0/0          0/0

Agentless VPN sessions:
 Index   User    Source IP      Duration        I/O Bytes       Tunnel/Dest IP
```

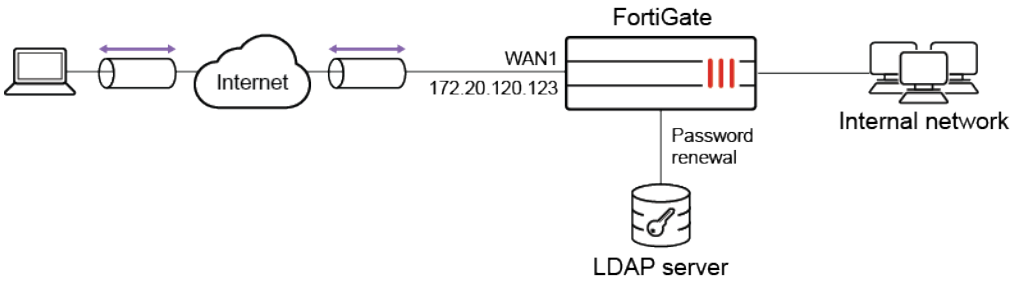
Agentless VPN with LDAP user password renew

This is a sample configuration of Agentless VPN for LDAP users with *Force Password Change on next logon*. In this example, the LDAP server is a Windows 2012 AD server. A user *ldu1* is configured on Windows 2012 AD server with *Force password change on next logon*.

> The LDAP renewal method is designed to replace (reset) the user password, meaning that the Active Directory password policy will not be enforced. For example, users can reuse the same password or use old ones.

You must have generated and exported a CA certificate from the AD server and then have imported it as an external CA certificate into the FortiGate.

Sample topology



Sample configuration

WAN interface is the interface connected to ISP. This example shows static mode. You can also use DHCP or PPPoE mode. The Agentless VPN connection is established over the WAN interface.

To configure Agentless VPN using the GUI:

1. Configure the interface and firewall address. The port1 interface connects to the internal network.
   a. Go to *Network* > *Interfaces* and edit the *wan1* interface.
   b. Set *IP/Network Mask* to *172.20.120.123/255.255.255.0*.
   c. Edit *port1* interface and set *IP/Network Mask* to *192.168.1.99/255.255.255.0*.
   d. Click *OK*.
   e. Go to *Policy & Objects* > *Address* and create an address for internet subnet *192.168.1.0*.
2. Import CA certificate into FortiGate:
   a. Go to *System* > *Features Visibility* and ensure *Certificates* is enabled.
   b. Go to *System* > *Certificates* and select *Import* > *CA Certificate*.
   c. Select *Local PC* and then select the certificate file.
      The CA certificate now appears in the list of *External CA Certificates*. In this example, it is called *CA_Cert_1*.

d. If you want, you can use CLI commands to rename the system-generated *CA_Cert_1* to be more descriptive:

```
config vpn certificate ca
    rename CA_Cert_1 to LDAPS-CA
end
```

3. Configure the LDAP user:

> The LDAP user must either be an administrator, or have the proper permissions delegated to it, to be able to change passwords of other registered users on the LDAP server.

a. Go to *User & Authentication* > *LDAP Servers* and click *Create New*.
b. Specify *Name* and *Server IP/Name*.
c. Specify *Common Name Identifier* and *Distinguished Name*.
d. Set *Bind Type* to *Regular*.
e. Specify *Username* and *Password*.
f. Set *Protocol* to *LDAPS*.
g. Enable *Certificate* and select *LDAPS-CA* from the list.
h. To enable the `password-renew` option, use these CLI commands.

```
config user ldap
    edit "ldaps-server"
        set password-expiry-warning enable
        set password-renewal enable
    next
end
```

4. Configure user group:
a. Go to *User & Authentication* > *User Groups* to create a user group.
b. Enter a *Name*.
c. In *Remote Groups*, click *Add* to add *ldaps-server*.
5. Configure Agentless VPN portals.
a. Go to *VPN* > *Agentless VPN Portals*, and click *Create New*.
b. Enter a name, such as *agentless-portal*.
c. Under *Predefined Bookmarks*, click *Create New*, and enter the following details:

Field	Value
Name	Windows Server
Type	RDP
Host	192.168.1.114
Port	3389
Single Sign-On	Disable
Username	Set the username to log in to Windows Server.

Field	Value
Password	Set the password to log in to Windows Server.
Port	3389

d. Click *OK* to save the predefined bookmark.
e. Click *OK* to save the portal settings.
f. Create another Agentless VPN portal and name it *portal-access-disabled*.
g. Click *OK* to save the portal settings.
h. Disable Agentless VPN for the newly created portal (that is, *portal-access-disabled*) using CLI:

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
end
```

6. Configure Agentless VPN settings.
   a. Go to *VPN > Agentless VPN Settings*.
   b. Set *Agentless VPN* status to *Enable*.
   c. Set *Listen on Interface(s)* to *wan1*.
   d. Set *Listen on Port* to *10443*.
   e. Choose a certificate for *Server Certificate*. The default is *Fortinet_Factory*.
   f. In *Authentication/Portal Mapping*, select *All Other Users/Groups*, and click *Edit*.
   g. Use *Portal* dropdown to select *portal-access-disabled*.
   h. Click *OK*.
   i. In *Authentication/Portal Mapping*, click *Create New*.
      i. Set *Users/Groups* to *ldaps-group*.
      ii. Set *Portal* to *agentless-portal*.
      iii. Click *OK*.
   j. Click *Apply* to save the Agentless VPN settings.
7. Configure Agentless VPN firewall policy:
   a. Go to *Policy & Objects > Firewall Policy*, and click *Create New*.
   b. Set *Name* to *Agentless VPN firewall policy*.
   c. Set *Schedule* to *always* and *Action* to *Accept*.
   d. Set *Incoming interface* to *Agentless VPN interface (ssl.root)*.
   e. Choose an *Outgoing Interface*. This example uses *port1*.
   f. Set the *Source* to *all* and *User/group* to *ldaps-group*.
   g. Set *Destination Address*. In this example, the destination is the internal, protected subnet *192.168.1.0*.
   h. Set *Service* to *ALL*.
   i. Click *OK*.

To configure Agentless VPN using the CLI:

1. Configure the interface and firewall address:

```
config system interface
    edit "wan1"
        set vdom "root"
        set ip 172.20.120.123 255.255.255.0
    next
end
```

2. Configure internal interface and protected subnet, then connect the port1 interface to the internal network:

```
config system interface
    edit "port1"
        set vdom "root"
        set ip 192.168.1.99 255.255.255.0
    next
end
```

```
config firewall address
    edit "192.168.1.0"
        set subnet 192.168.1.0 255.255.255.0
    next
end
```

3. Import CA certificate into FortiGate:
   a. Go to *System* > *Features Visibility* and ensure *Certificates* is enabled.
   b. Go to *System* > *Certificates* and select *Import* > *CA Certificate*.
   c. Select *Local PC* and then select the certificate file.
      The CA certificate now appears in the list of *External CA Certificates*. In the example, it is called *CA_Cert_1*.
   d. If you want, you can use CLI commands to rename the system-generated *CA_Cert_1* to be more descriptive:

```
config vpn certificate ca
    rename CA_Cert_1 to LDAPS-CA
end
```

4. Configure the LDAP server:

> The LDAP user must either be an administrator, or have the proper permissions delegated to it, to be able to change passwords of other registered users on the LDAP server.

```
config user ldap
    edit "ldaps-server"
        set server "172.20.120.161"
        set cnid "cn"
        set dn "cn=Users,dc=qa,dc=fortinet,dc=com"
        set type regular
```

```
        set username "CN=Administrator,cn=users,DC=qa,DC=fortinet,DC=com"
        set password **********
        set group-member-check group-object
        set secure ldaps
        set ca-cert "LDAPS-CA"
        set port 636
        set password-expiry-warning enable
        set password-renewal enable
    next
end
```

5. Configure user group:

```
config user group
    edit "ldaps-group"
        set member "ldaps-server"
    next
end
```

6. Configure Agentless VPN portal and predefine RDP bookmark for windows server.

```
config vpn ssl web portal
    edit "agentless-portal"
        set web-mode enable
        config bookmark-group
            edit "gui-bookmarks"
                config bookmarks
                    edit "Windows Server"
                        set apptype rdp
                        set host "192.168.1.114"
                        set port 3389
                        set logon-user "your-windows-server-user-name"
                        set logon-password your-windows-server-password
                    next
                end
            next
        end
    next
end
```

7. Configure another Agentless VPN portal and disable Agentless VPN on it.

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
end
```

8. Configure Agentless VPN settings:

```
config vpn ssl settings
    set servercert "server_certificate"
```

```
    set source-interface "wan1"
    set source-address "all"
    set default-portal "portal-access-disable"
    config authentication-rule
        edit 1
            set groups "ldaps-group"
            set portal "agentless-portal"
        next
    end
end
```

9. Configure one Agentless VPN firewall policy to allow remote user to access the internal network. This policy does not allow traffic initiated from internal network to remote client.

```
config firewall policy
    edit 1
        set name "Agentless VPN firewall policy"
        set srcintf "ssl.root"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "192.168.1.0"
        set action accept
        set schedule "always"
        set service "ALL"
        set nat enable
        set groups "ldaps-group"
    next
end
```

To see the results of web portal:

1. From a remote device, use a web browser to log into the Agentless VPN web portal *http://172.20.120.123:10443*.
2. Log in using the *ldu1* credentials.
   Use a user that is configured on FortiAuthenticator with *Force password change on next logon*.
3. Click *Login*. You are prompted to enter a new password. The prompt will timeout after 90 seconds.
4. On FortiGate, go to *Dashboard > Agentless VPN Monitor* to verify the list of Agentless VPN users.
   If the Agentless VPN Monitor is hidden, click *+* under the *Dashboard*, search for *Agentless VPN Monitor*, and add it to the display.
5. Go to *Log & Report > System Events*, and use the dropdown to select *VPN Events* to view VPN logs.

To check the web portal login using the FortiOS CLI:

```
# get vpn ssl monitor
Agentless VPN Login Users:
 Index   User    Auth Type    Timeout    From           HTTP in/out  HTTPS in/out
 0       ldu1    1(1)         229        10.1.100.254   0/0          0/0

Agentless VPN sessions:
 Index   User    Source IP      Duration        I/O Bytes       Tunnel/Dest IP
```

Agentless VPN with certificate authentication

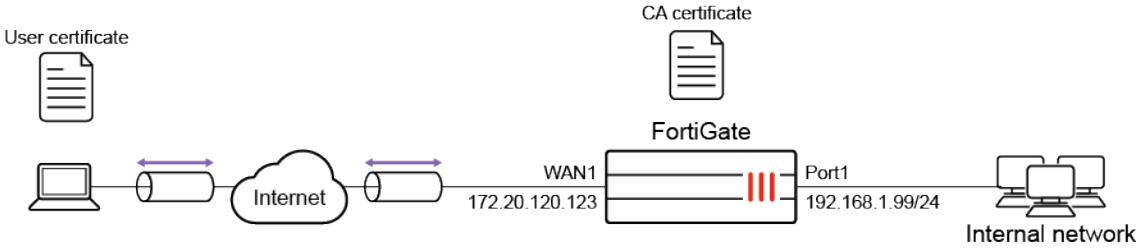
This is an example configuration of Agentless VPN that requires users to authenticate using a client certificate. The client certificate is issued by the company Certificate Authority (CA). Each user is issued a certificate with their username in the subject.

There are two ways to configure certificate authentication:

1. Using PKI users
2. Configuring Agentless VPN settings to require a client certificate

In this example, the server and client certificates are signed by the same Certificate Authority (CA).

> ⚠ Self-signed certificates are provided by default to simplify initial installation and testing. It is **HIGHLY** recommended that you acquire a signed certificate for your installation.
>
> Continuing to use these certificates can result in your connection being compromised, allowing attackers to steal your information, such as credit card details.
>
> For more information, please review the Use a non-factory SSL certificate for the Agentless VPN portal on page 3375 and learn how to Procuring and importing a signed SSL certificate on page 3974.



Using PKI users

When using PKI users, the FortiGate authenticates the user based on there identity in the subject or the common name on the certificate. The certificate must be signed by a CA that is known by the FortiGate, either through the default CA certificates or through importing a CA certificate.

The user can either match a static subject or common name defined in the PKI user settings, or match an LDAP user in the LDAP server defined in the PKI user settings. Multi-factor authentication can also be enabled with the password as the second factor.

Configuring Agentless VPN settings to require a client certificate

Using this method, the user is authenticated based on their regular username and password, but Agentless VPN will still require an additional certificate check. The client certificate only needs to be signed by a known CA in order to pass authentication.

This method can be configured by enabling *Require Client Certificate* (`reqclientcert`) in the Agentless VPN settings.

Configuration

In the following example, Agentless VPN users are authenticated using the first method. A PKI user is configured with multi-factor authentication

Pre-requisites:

- The CA has already issued a client certificate to the user.
- The CA has issued a server certificate for the FortiGate's Agentless VPN portal.
- The CA certificate is available to be imported on the FortiGate.

To configure Agentless VPN in the GUI:

1. Install the server certificate. The server certificate allows the clients to authenticate the server and to encrypt the Agentless VPN traffic.
   a. Go to *System > Feature Visibility* and ensure *Certificates* is enabled.
   b. Go to *System > Certificates* and select *Import > Local Certificate*.
      - Set *Type* to *Certificate*.
      - Choose the *Certificate file* and the *Key file* for your certificate, and enter the *Password*.
      - If required, you can change the *Certificate Name*.

   The server certificate now appears in the list of *Certificates*.
2. Install the CA certificate.
   The CA certificate is the certificate that signed both the server certificate and the user certificate. In this example, it is used to authenticate Agentless VPN users.
   a. Go to *System > Certificates* and select *Import > CA Certificate*.
   b. Select *Local PC* and then select the certificate file.

   The CA certificate now appears in the list of *External CA Certificates*. In this example, it is called *CA_Cert_1*.
3. Configure PKI users and a user group.
   To use certificate authentication, use the CLI to create PKI users.

```
config user peer
    edit pki01
        set ca CA_Cert_1
        set subject "CN=User01"
    next
end
```

   Ensure that the subject matches the name of the user certificate. In this example, *User01*.
4. After you have create a PKI user, a new menu is added to the GUI:
   a. Go to *User & Authentication > PKI* to see the new user.
   b. Edit the user account.
   c. Enable *Two-factor authentication* and set a password for the account.
   d. Go to *User & Authentication > User Groups* and create a group called *agentlessvpngroup*.
   e. Add the PKI user *pki01* to the group.
5. Configure Agentless VPN portals.
   a. Go to *VPN > Agentless VPN Portals*, and click *Create New*.
   b. Enter a name, such as *agentless-portal*.

c. Under *Predefined Bookmarks*, click *Create New*, and enter the following details:

Field	Value
Name	Windows Server
Type	RDP
Host	192.168.1.114
Port	3389
Single Sign-On	Disable
Username	Set the username to log in to Windows Server.
Password	Set the password to log in to Windows Server.
Port	3389

d. Click *OK* to save the predefined bookmark.
e. Click *OK* to save the portal settings.
f. Create another Agentless VPN portal and name it *portal-access-disabled*.
g. Click *OK* to save the portal settings.
h. Disable Agentless VPN for the newly created portal (that is, *portal-access-disabled*) using CLI:

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
end
```

6. Configure Agentless VPN settings.
   a. Go to *VPN > Agentless VPN Settings* and enable Agentless VPN.
   b. Set the *Listen on Interface(s)* to *wan1*.
   c. Set *Listen on Port* to *10443*.
   d. Set *Server Certificate* to the local certificate that was imported.
   e. Enable *Require Client Certificate*.
   f. Under *Authentication/Portal Mapping*, set select *All Other Users/Groups*, and click *Edit*.
   g. Use *Portal* dropdown to select *portal-access-disabled*.
   h. Click *OK*.
   i. In *Authentication/Portal Mapping*, click *Create New*.
      i. Set *Users/Groups* to *agentlessvpngroup*.
      ii. Set *Portal* to *agentless-portal*.
      iii. Click *OK*.
   j. Click *Apply* to save the Agentless VPN settings.
7. Configure Agentless VPN firewall policy.
   a. Go to *Policy & Objects > Firewall Policy*, and click *Create New*.
   b. Set *Name* to *Agentless VPN firewall policy*.
   c. Set *Schedule* to *always* and *Action* to *Accept*.
   d. Set *Incoming interface* to *Agentless VPN interface (ssl.root)*.

e. Choose an *Outgoing Interface*. This example uses *port1*.
f. Set the *Source* to *all* and *User/group* to *agentlessvpngroup*.
g. In this example, the *Destination* is the internal, protected subnet *192.168.1.0*.
h. Set *Service* to *ALL*.
i. Click *OK*.

To configure Agentless VPN in the CLI:

1. Configure the protected subnet:

```
config firewall address
    edit "192.168.1.0"
        set subnet 192.168.1.0 255.255.255.0
    next
end
```

2. Install the server certificate:

   The server certificate allows the clients to authenticate the server and to encrypt the Agentless VPN traffic. While it is easier to install the server certificate in the GUI, the CLI can be used to import a p12 certificate from a TFTP server.

   To import a p12 certificate, put the certificate *server_certificate.p12* on your TFTP server, then run following command on the FortiGate:

```
execute vpn certificate local import tftp server_certificate.p12 <your tftp_server> p12 <your
password for PKCS12 file>
```

   To check that the server certificate is installed:

```
show vpn certificate local server_certificate
```

3. Install the CA certificate:

   The CA certificate is the certificate that signed both the server certificate and the user certificate. In this example, it is used to authenticate Agentless VPN users. While it is easier to install the CA certificate from GUI, the CLI can be used to import a CA certificates from a TFTP server.

   To import a CA certificate, put the CA certificate on your TFTP server, then run following command on the FortiGate:

```
execute vpn certificate ca import tftp <your CA certificate name> <your tftp server>
```

   To check that a new CA certificate is installed:

```
show vpn certificate ca
```

4. Configure PKI users and a user group:

```
config user peer
    edit pki01
        set ca CA_Cert_1
        set subject "CN=User01"
        set two-factor enable
        set passwd **********
```

```
    next
end
```

```
config user group
    edit "agentlessvpngroup"
        set member "pki01"
    next
end
```

5. Configure Agentless VPN portal and predefine RDP bookmark for windows server.

```
config vpn ssl web portal
    edit "agentless-portal"
        set web-mode enable
        config bookmark-group
            edit "gui-bookmarks"
                config bookmarks
                    edit "Windows Server"
                        set apptype rdp
                        set host "192.168.1.114"
                        set port 3389
                        set logon-user "your-windows-server-user-name"
                        set logon-password your-windows-server-password
                    next
                end
            next
        end
    next
end
```

6. Configure another Agentless VPN portal and disable Agentless VPN on it.

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
end
```

7. Configure Agentless VPN settings:

```
config vpn ssl settings
    set reqclientcert enable
    set banned-cipher SHA1 SHA256 SHA384
    set servercert "server_certificate"
    set source-interface "wan1"
    set source-address "all"
    set default-portal "portal-access-disabled"
    config authentication-rule
        edit 1
            set groups "agentlessvpngroup"
            set portal "agentless-portal"
        next
```

```
    end
end
```

8. Configure one Agentless VPN firewall policy to allow remote user to access the internal network. This policy does not allow traffic initiated from internal network to remote client.

```
config firewall policy
    edit 1
        set name "Agentless VPN firewall policy"
        set srcintf "ssl.root"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "192.168.1.0"
        set action accept
        set schedule "always"
        set service "ALL"
        set nat enable
        set groups "agentlessvpngroup"
    next
end
```

Installation

To use the user certificate, you must first install it on the user's PC. When the user tries to authenticate, the user certificate is checked against the CA certificate to verify that they match.

Every user should have a unique user certificate. This allows you to distinguish each user and revoke a specific user's certificate, such as if a user no longer has VPN access.

To install the user certificate on Windows 7, 8, and 10:

1. Double-click the certificate file to open the *Import Wizard*.
2. Use the *Import Wizard* to import the certificate into the *Personal store* of the current user.

To install the user certificate on Mac OS X:

1. Open the certificate file, to open *Keychain Access*.
2. Double-click the certificate.
3. Expand *Trust* and select *Always Trust*.

To see the results of Agentless VPN web portal:

1. In a web browser, log into the portal *http://172.20.120.123:10443*.
   A message requests a certificate for authentication.
2. Select the user certificate.
3. Enter your user credentials.
   If the certificate is correct, you can connect to the Agentless VPN web portal.

To check the Agentless VPN connection using the GUI:

1. On FortiGate, go to *Dashboard* > *Agentless VPN Monitor* to verify the list of Agentless VPN users.
   If the Agentless VPN Monitor is hidden, click + under the *Dashboard*, search for *Agentless VPN Monitor*, and add it to the display.
2. Go to *Log & Report* > *System Events*, and use the dropdown to select *VPN Events* to view VPN logs.

To check the Agentless VPN connection using the CLI:

```
get vpn ssl monitor
Agentless VPN Login Users:
 Index   User    Auth Type      Timeout         From     HTTP in/out    HTTPS in/out
 0       pki01,cn=User01          1(1)            229     10.1.100.254   0/0     0/0
 1       pki01,cn=User01          1(1)            291     10.1.100.254   0/0     0/0

Agentless VPN sessions:
 Index   User    Source IP      Duration        I/O Bytes       Tunnel/Dest IP
 0       pki01,cn=User01          10.1.100.254    9       22099/43228    10.212.134.200
```

Agentless VPN with LDAP-integrated certificate authentication

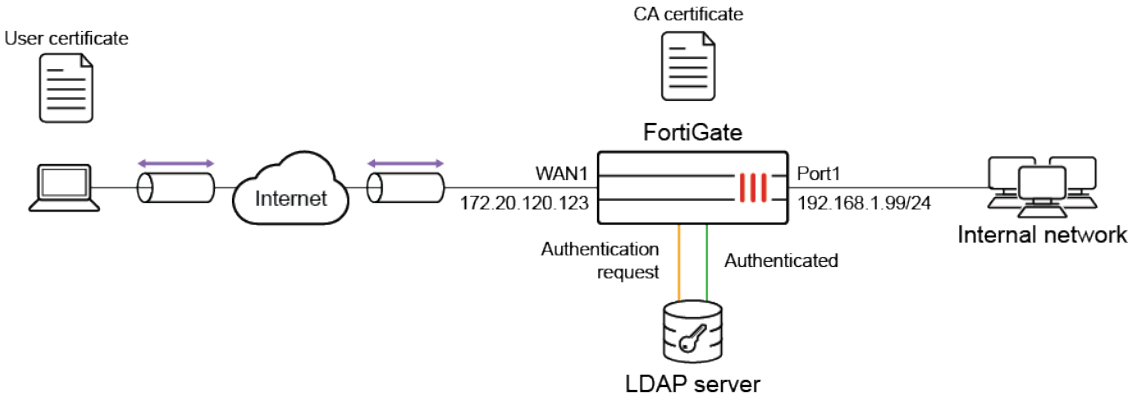
This is a sample configuration of Agentless VPN that requires users to authenticate using a certificate with LDAP `UserPrincipalName` checking.

This sample uses Windows 2012R2 Active Directory acting as both the user certificate issuer, the certificate authority, and the LDAP server.

⚠ When configuring an LDAP connection to an Active Directory server, an administrator must provide Active Directory user credentials.

- To secure this connection, use LDAPS on both the Active Directory server and FortiGate. See Configuring an LDAP server on page 3509 and Configuring client certificate authentication on the LDAP server on page 3524.
- Apply the principle of least privilege. For the LDAP regular bind operation, do not use credentials that provide full administrative access to the Windows server when using credentials. See Configuring least privileges for LDAP admin account authentication in Active Directory on page 3517.

Sample topology



Sample configuration

WAN interface is the interface connected to ISP. This example shows static mode. You can also use DHCP or PPPoE mode. The Agentless VPN connection is established over the WAN interface.

In this sample, the *User Principal Name* is included in the subject name of the issued certificate. This is the user field we use to search LDAP in the connection attempt.

To use the user certificate, you must first install it on the user's PC. When the user tries to authenticate, the user certificate is checked against the CA certificate to verify that they match.

Every user should have a unique user certificate. This allows you to distinguish each user and revoke a specific user's certificate, such as if a user no longer has VPN access.

To install the server certificate:

The server certificate is used for authentication and for encrypting Agentless VPN traffic.

1. Go to *System* > *Feature Visibility* and ensure *Certificates* is enabled.
2. Go to *System* > *Certificates* and select *Import* > *Local Certificate*.
3. Set *Type* to *Certificate*.
4. Choose the *Certificate file* and the *Key file* for your certificate, and enter the *Password*.
5. If required, change the *Certificate Name*.
   The server certificate now appears in the list of *Certificates*.

To install the CA certificate:

The CA certificate is the certificate that signed both the server certificate and the user certificate. In this example, it is used to authenticate Agentless VPN users.

1. Go to *System* > *Certificates* and select *Import* > *CA Certificate*.
2. Select *Local PC* and then select the certificate file.
   The CA certificate now appears in the list of *External CA Certificates*. In this example, it is called *CA_Cert_1*.

To configure Agentless VPN using the GUI:

1. Configure the interface and firewall address. The port1 interface connects to the internal network.
   a. Go to *Network* > *Interfaces* and edit the *wan1* interface.
   b. Set *IP/Network Mask* to *172.20.120.123/255.255.255.0*.
   c. Edit *port1* interface and set *IP/Network Mask* to *192.168.1.99/255.255.255.0*.
   d. Click *OK*.
   e. Go to *Policy & Objects* > *Address* and create an address for internet subnet *192.168.1.0*.
2. Configure the LDAP server:
   a. Go to *User & Authentication* > *LDAP Servers* and click *Create New*.
   b. Specify *Name* and *Server IP/Name*.
   c. Set *Distinguished Name* to *dc=fortinet-fsso,dc=com*.
   d. Set *Bind Type* to *Regular*.
   e. Set *Username* to *cn=admin,ou=testing,dc=fortinet-fsso,dc=com*.
   f. Set *Password*.
   g. Click *OK*.
3. Configure PKI users and a user group:
   To use certificate authentication, use the CLI to create PKI users.

```
config user peer
    edit user1
        set ca CA_Cert_1
        set mfa-server "ldap-AD"
        set mfa-mode subject-identity
    next
end
```

   When you have create a PKI user, a new menu is added to the GUI:
   a. Go to *User & Authentication* > *PKI* to see the new user.
   b. Go to *User & Authentication* > *User* > *User Groups* and create a group *agentlessvpn-group*.
   c. Add the PKI peer object you created as a local member of the group.
   d. Add a remote group on the LDAP server and select the group of interest.
      You need these users to be members using the LDAP browser window.
4. Configure Agentless VPN portals.
   a. Go to *VPN* > *Agentless VPN Portals*, and click *Create New*.
   b. Enter a name, such as *agentless-portal*.
   c. Under *Predefined Bookmarks*, click *Create New*, and enter the following details:

Field	Value
Name	Windows Server
Type	RDP
Host	192.168.1.114
Port	3389

Field	Value
Single Sign-On	Disable
Username	Set the username to log in to Windows Server.
Password	Set the password to log in to Windows Server.
Port	3389

d. Click *OK* to save the predefined bookmark.
e. Click *OK* to save the portal settings.
f. Create another Agentless VPN portal and name it *portal-access-disabled*.
g. Click *OK* to save the portal settings.
h. Disable Agentless VPN for the newly created portal (that is, *portal-access-disabled*) using CLI:

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
end
```

5. Configure Agentless VPN settings:
   a. Go to *VPN > Agentless VPN Settings*.
   b. Select the *Listen on Interface(s)*, in this example, *wan1*.
   c. Set *Listen on Port* to *10443*.
   d. Set *Server Certificate* to the authentication certificate.
      i. In *Authentication/Portal Mapping*, select *All Other Users/Groups*, and click *Edit*.
      ii. Use *Portal* dropdown to select *portal-access-disabled*.
      iii. Click *OK*.
   e. In *Authentication/Portal Mapping*, click *Create New*.
      i. Set *Users/Groups* to *agentlessvpn-group*.
      ii. Set *Portal* to *agentless-portal*.
      iii. Click *OK*.
   f. Click *Apply* to save the Agentless VPN settings.
6. Configure Agentless VPN firewall policy.
   a. Go to *Policy & Objects > Firewall Policy*, and click *Create New*.
   b. Set *Name* to *Agentless VPN firewall policy*.
   c. Set *Schedule* to *always* and *Action* to *Accept*.
   d. Set *Incoming interface* to *Agentless VPN interface (ssl.root)*.
   e. Choose an *Outgoing Interface*. This example uses *port1*.
   f. Set the *Source* to *all* and *User/group* to *agentlessvpngroup*.
   g. In this example, the *Destination* is the internal, protected subnet *192.168.1.0*.
   h. Set *Service* to *ALL*.
   i. Click *OK*.

To configure Agentless VPN using the CLI:

1. Configure the interface and firewall address:

```
config system interface
    edit "wan1"
        set vdom "root"
        set ip 172.20.120.123 255.255.255.0
    next
end
```

2. Configure internal interface and protected subnet, then connect the port1 interface to the internal network:

```
config system interface
    edit "port1"
        set vdom "root"
        set ip 192.168.1.99 255.255.255.0
    next
end
```

```
config firewall address
    edit "192.168.1.0"
        set subnet 192.168.1.0 255.255.255.0
    next
end
```

3. Configure the LDAP server:

```
config user ldap
    edit "ldap-AD"
        set server "172.18.60.206"
        set cnid "cn"
        set dn "dc=fortinet-fsso,dc=com"
        set type regular
        set username "cn=admin,ou=testing,dc=fortinet-fsso,dc=com"
        set password ldap-server-password
    next
end
```

4. Configure PKI users and a user group:

```
config user peer
    edit user1
        set ca CA_Cert_1
        set mfa-server "ldap-AD"
        set mfa-mode subject-identity
    next
end
```

```
config user group
    edit "agentlessvpn-group"
        set member "ldap-AD" "user1"
        config match
```

```
            edit 1
                set server-name "ldap-AD"
                set group-name "CN=group3,OU=Testing,DC=Fortinet-FSSO,DC=COM"
            next
        end
    next
end
```

5. Configure Agentless VPN portal and predefine RDP bookmark for windows server.

```
config vpn ssl web portal
    edit "agentless-portal"
        set web-mode enable
        config bookmark-group
            edit "gui-bookmarks"
                config bookmarks
                    edit "Windows Server"
                        set apptype rdp
                        set host "192.168.1.114"
                        set port 3389
                        set logon-user "your-windows-server-user-name"
                        set logon-password your-windows-server-password
                    next
                end
            next
        end
    next
end
```

6. Configure another Agentless VPN portal and disable Agentless VPN on it.

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
end
```

7. Configure Agentless VPN settings:

```
config vpn ssl settings
    set servercert "server_certificate"
    set source-interface "wan1"
    set source-address "all"
    set default-portal "agentless-portal"
    config authentication-rule
        edit 1
            set groups "agentlessvpn-group"
            set portal "agentless-portal"
        next
    end
end
```

8. Configure one Agentless VPN firewall policy to allow remote user to access the internal network. This policy does not allow traffic initiated from internal network to remote client.

```
config firewall policy
    edit 1
        set name "Agentless VPN firewall policy"
        set srcintf "ssl.root"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "192.168.1.0"
        set action accept
        set schedule "always"
        set service "ALL"
        set nat enable
        set groups "agentlessvpngroup"
    next
end
```

To see the results of Agentless VPN portal:

1. In a web browser, log into the portal *http://172.20.120.123:10443*.
   A message requests a certificate for authentication.
2. Select the user certificate. It does not require username and password.
   You can connect to the Agentless VPN web portal.

To check the Agentless VPN connection using the GUI:

1. On FortiGate, go to *Dashboard > Agentless VPN Monitor* to verify the list of Agentless VPN users.
   If the Agentless VPN Monitor is hidden, click *+* under the *Dashboard*, search for *Agentless VPN Monitor*, and add it to the display.
2. Go to *Log & Report > System Events*, and use the dropdown to select *VPN Events* to view VPN logs.

To check the Agentless VPN connection using the CLI:

```
get vpn ssl monitor
Agentless VPN Login Users:
 Index   User    Auth Type      Timeout         From     HTTP in/out    HTTPS in/out
 0       test3          1(1)            229     10.1.100.254   0/0     0/0

Agentless VPN sessions:
 Index   User    Source IP      Duration        I/O Bytes       Tunnel/Dest IP
```

To check the real-time authentication debug logs using the CLI:

Below is a sample output of `diagnose debug application fnbamd -1` while the user connects. This is a shortened output sample of a few locations to show the important parts. This sample shows lookups to find the group memberships (three groups total) of the user and that the correct group being found results in a match.

```
[1148] fnbamd_ldap_recv-Response len: 16, svr: 172.18.60.206
[829] fnbamd_ldap_parse_response-Got one MESSAGE. ID:4, type:search-result
```

```
[864] fnbamd_ldap_parse_response-ret=0
[1386] __fnbamd_ldap_primary_grp_next-Auth accepted
[910] __ldap_rxtx-Change state to 'Done'
[843] __ldap_rxtx-state 23(Done)
[925] fnbamd_ldap_send-sending 7 bytes to 172.18.60.206
[937] fnbamd_ldap_send-Request is sent. ID 5
[753] __ldap_stop-svr 'ldap-AD'
[53] ldap_dn_list_del_all-Del CN=test3,OU=Testing,DC=Fortinet-FSSO,DC=COM
[399] ldap_copy_grp_list-copied CN=group3,OU=Testing,DC=Fortinet-FSSO,DC=COM
[399] ldap_copy_grp_list-copied CN=Domain Users,CN=Users,DC=Fortinet-FSSO,DC=COM
[2088] fnbamd_auth_cert_check-Matching group 'agentlessvpn-group'
[2007] __match_ldap_group-Matching server 'ldap-AD' - 'ldap-AD'
[2015] __match_ldap_group-Matching group 'CN=group3,OU=Testing,DC=Fortinet-FSSO,DC=COM' -
'CN=group3,OU=Testing,DC=Fortinet-FSSO,DC=COM'
[2091] fnbamd_auth_cert_check-Group 'agentlessvpn-group' matched
[2120] fnbamd_auth_cert_result-Result for ldap svr[0] 'ldap-AD' is SUCCESS
[2126] fnbamd_auth_cert_result-matched user 'test3', matched group 'agentlessvpn-group'
```

You can also use `diagnose firewall auth list` to validate that a firewall user entry exists for the Agentless VPN user and is part of the right groups.

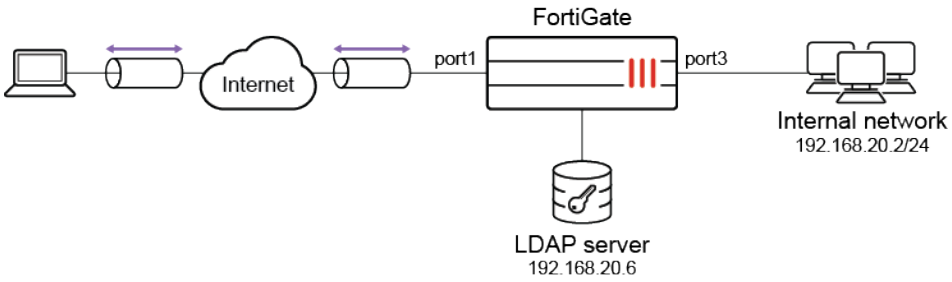
Agentless VPN for remote users with MFA and user sensitivity

By default, remote LDAP and RADIUS user names are case sensitive. When a remote user object is applied to Agentless VPN authentication, the user must type the exact case that is used in the user definition on the FortiGate.

Case sensitivity and accents can be ignored by disabling the `username-sensitivity` CLI command, allowing the remote user object to match any case or accents that the end user types in.

In this example, a remote user is configured with multi-factor authentication (MFA). The user group includes the LDAP user and server and is applied to Agentless VPN authentication and the policy.

Topology



Example configuration

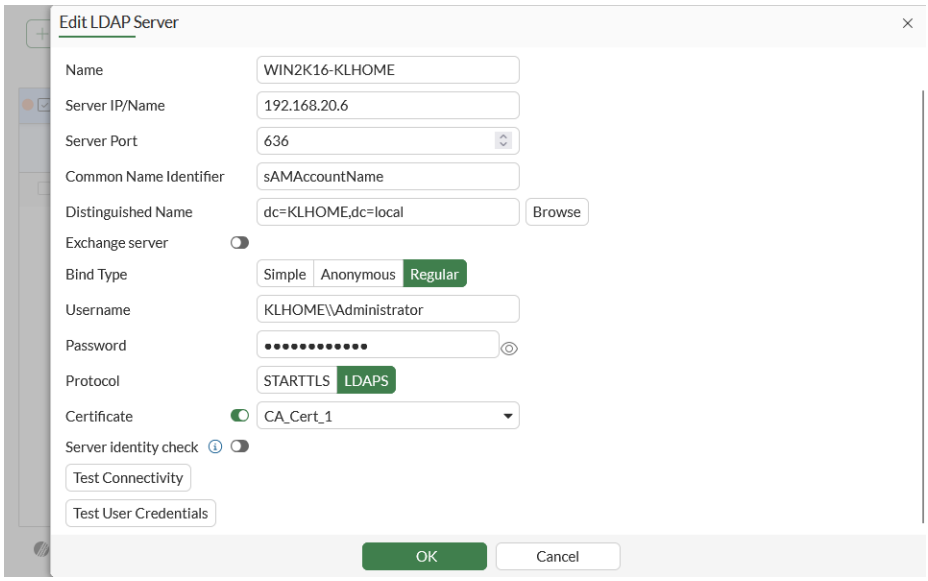
To configure the LDAP server:

1. Generate and export a CA certificate from the AD server .
2. Import the CA certificate into FortiGate:
   a. Go to *System* > *Features Visibility* and ensure *Certificates* is enabled.
   b. Go to *System* > *Certificates* and select *Import* > *CA Certificate*.
   c. Select *Local PC* and then select the certificate file.
      The CA certificate now appears in the list of *External CA Certificates*. In this example, it is called *CA_Cert_1*.
   d. If you want, you can use CLI commands to rename the system-generated *CA_Cert_1* to be more descriptive:

```
config vpn certificate ca
    rename CA_Cert_1 to LDAPS-CA
end
```

3. Configure the LDAP user:
   a. Go to *User & Authentication* > *LDAP Servers* and click *Create New*.
   b. Configure the following options for this example:

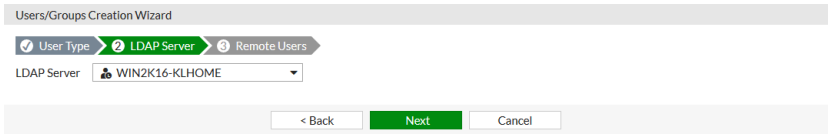
Name	WIN2K16-KLHOME
Server IP/Name	192.168.20.6
Server Port	636
Common Name Identifier	sAMAccountName
Distinguished Name	dc=KLHOME,dc=local
Bind Type	Regular
Username	KLHOME\\Administrator
Password	*********
Protocol	LDAPS
Certificate	CA_Cert_1 This is the CA certificate that you imported in step 2.



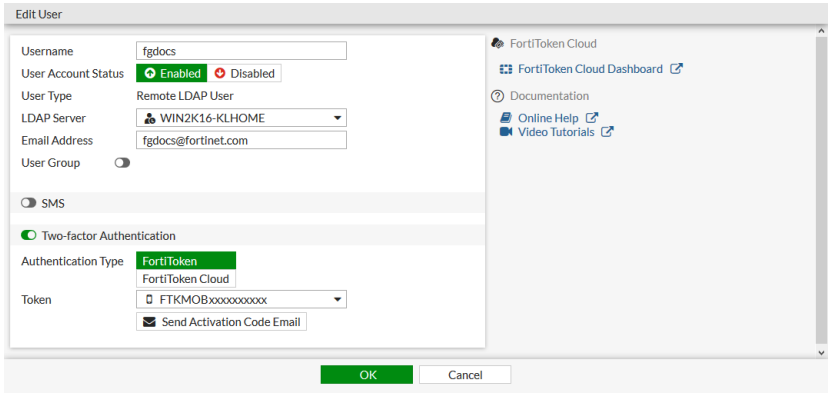
c. Click *OK*.

To configure an LDAP user with MFA:

1. Go to *User & Authentication* > *User Definition* and click *Create New*.
2. Select *Remote LDAP User*, then click *Next*.
3. Select the just created LDAP server, then click *Next*.



4. Right click to add the selected user, then click *Submit*.
5. Edit the user that you just created.
   The username will be pulled from the LDAP server with the same case as it has on the server.
6. Set the *Email Address* to the address that FortiGate will send the FortiToken to.
7. Enable *Two-factor Authentication*.
8. Set *Authentication Type* to *FortiToken*.
9. Set *Token* to a FortiToken device. See for more information.



10. Click *OK*.

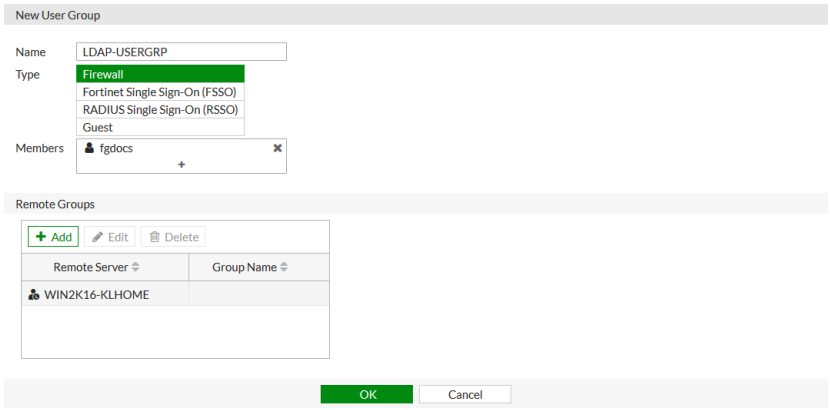
To disable case and accent sensitivity on the remote user:

This can only be configured in the CLI.

```
config user local
    edit "fgdocs"
        set type ldap
        set two-factor fortitoken
        set fortitoken "FTKMOBxxxxxxxxxx"
        set email-to "fgdocs@fortinet.com"
        set username-sensitivity disable
        set ldap-server "WIN2K16-KLHOME"
    next
end
```

To configure a user group with the remote user and the LDAP server:

1. Go to *User & Authentication* > *User Groups* and click *Create New*.
2. Set the *Name* to *LDAP-USERGRP*.
3. Set *Members* to the just created remote user.
4. In the *Remote Groups* table, click *Add*:
   a. Set *Remote Server* to the LDAP server.
   b. Set the group or groups that apply, and right click to add them.
   c. Click *OK*.



5. Click *OK*.

To configure Agentless VPN Portal:

1. Go to *VPN* > *Agentless VPN Portals*, and click *Create New*.
2. Enter a *Name*, such as *agentless-portal*.
3. Under *Predefined Bookmarks*, click *Create New*, and enter following details:

Field	Value
Name	Windows Server
Type	RDP

Field	Value
Host	192.168.1.114
Port	3389
Single Sign-On	Disable
Username	Set the username to login to Windows Server
Password	Set the password to login to Windows Server
Port	3389

4. Click *OK* to save the predefined bookmark.
5. Click *OK* to save the portal settings.
6. Create another Agentless VPN Portal and name it as *portal-access-disabled*.
7. Click *OK* to save the portal settings.
8. Disable Agentless VPN for the newly created portal (that is, `portal-access-disabled`) using CLI:

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
end
```

To configure Agentless VPN settings and apply the user group to the Agentless VPN portal:

1. Go to *VPN > Agentless VPN Settings*.
2. Set *Agentless VPN* status to *Enable*.
3. For *Listen on Interface(s)*, select *wan1*.
4. Set *Listen on Port* to *10443*.
5. Choose a certificate for *Server Certificate*. The default is *Fortinet_Factory*.
6. In *Authentication/Portal Mapping*, select *All Other Users/Groups*, and click *Edit*.
7. Use *Portal* dropdown to select *portal-access-disabled*.
8. Click *OK*.
9. In *Authentication/Portal Mapping*, click *Create New*.
   a. Set *Users/Groups* to *LDAP-USERGRP*.
   b. Set *Portal* to *agentless-portal*.
   c. Click *OK*.
10. Click *Apply* to save the Agentless VPN settings.

To apply the user group to Agentless VPN firewall policy:

1. Go to *Policy & Objects > Firewall Policy*, and click *Create New*.
2. Set *Name* as *Agentless VPN firewall policy*.
3. Set *Schedule* to *always* and *Action* to *Accept*.
4. Set *Incoming interface* to *Agentless VPN interface (ssl.root)*.
5. Choose an *Outgoing Interface*. This example uses *port1*.

6. Set the *Source* to *all* and *User/group* to *LDAP-USERGRP*.
7. In this example, the *Destination* is the internal, protected subnet 192.168.1.0.
8. Set *Service* to *ALL*.
9. Click *OK*.

To configure this example in the CLI:

1. Configure the LDAP server:

```
config user ldap
    edit "WIN2K16-KLHOME"
        set server "192.168.20.6"
        set cnid "sAMAccountName"
        set dn "dc=KLHOME,dc=local"
        set type regular
        set username "KLHOME\\Administrator"
        set password *********
        set secure ldaps
        set ca-cert "CA_Cert_1"
        set port 636
    next
end
```

2. Configure an LDAP user with MFA and disable case and accent sensitivity on the remote user:

```
config user local
    edit "fgdocs"
        set type ldap
        set two-factor fortitoken
        set fortitoken "FTKMOBxxxxxxxxxx"
        set email-to "fgdocs@fortinet.com"
        set username-sensitivity disable
        set ldap-server "WIN2K16-KLHOME"
    next
end
```

3. Configure a user group with the remote user and the LDAP server:

```
config user group
    edit "LDAP-USERGRP"
        set member "fgdocs" "WIN2K16-KLHOME"
    next
end
```

4. Configure Agentless VPN portal and predefine RDP bookmark for windows server.

```
config vpn ssl web portal
    edit "agentless-portal"
        set web-mode enable
        config bookmark-group
            edit "gui-bookmarks"
                config bookmarks
```

```
                    edit "Windows Server"
                        set apptype rdp
                        set host "192.168.1.114"
                        set port 3389
                        set logon-user "your-windows-server-user-name"
                        set logon-password your-windows-server-password
                    next
                end
            next
        end
    next
end
```

5. Configure another Agentless VPN portal and disable Agentless VPN on it.

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
end
```

6. Apply the user group to the Agentless VPN portal:

```
config vpn ssl settings
    set servercert "Fortinet_Factory"
    set source-interface "wan1"
    set source-address "all"
    set default-portal "portal-access-disabled"
    config authentication-rule
        edit 1
            set groups "LDAP-USERGRP"
            set portal "agentless-portal"
        next
    end
end
```

7. Configure one Agentless VPN firewall policy to allow the remote user to access the internal network. This policy does not allow traffic initiated from internal network to remote client:

```
config firewall policy
    edit 1
        set name "Agentless VPN firewall policy"
        set srcintf "ssl.root"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "192.168.1.0"
        set schedule "always"
        set service "ALL"
        set nat enable
        set groups "LDAP-USERGRP"
```

```
    next
end
```

Verification

To test the connection with case sensitivity disabled:

1. From a remote device, use a web browser to log in to the Agentless VPN web portal *https://172.20.120.123:10443*.
2. Connect to the Agentless VPN:
   a. Log in to the tunnel with the username, using the same case that it is on the FortiGate.
   b. When prompted, enter your FortiToken code.
   You should now be connected.
3. Check the web portal log in using the CLI:

```
# get vpn ssl monitor
Agentless VPN Login Users:
 Index   User    Group   Auth Type      Timeout         From     HTTP in/out   HTTPS in/out
 0       fgdocs          LDAP-USERGRP    16(1)           289     192.168.2.202 0/0      0/0

Agentless VPN sessions:
 Index   User    Group   Source IP      Duration        I/O Bytes       Tunnel/Dest IP
```

4. Log out from the Agentless VPN portal.
5. Reconnect to the Agentless VPN portal:
   a. Log in to the Agentless VPN portal with the username, using a different case than on the FortiGate.
   b. When prompted, enter your FortiToken code.
   You should now be connected.
6. Check the web portal log in using the CLI:

```
# get vpn ssl monitor
Agentless VPN Login Users:
 Index   User    Group   Auth Type      Timeout         From     HTTP in/out   HTTPS in/out
 0       FGDOCS          LDAP-USERGRP    16(1)           289     192.168.2.202 0/0      0/0

Agentless VPN sessions:
 Index   User    Group   Source IP      Duration        I/O Bytes       Tunnel/Dest IP
```

In both cases, the remote user is matched against the remote LDAP user object and prompted for multi-factor authentication.

To test the connection with case and accent sensitivity enabled:

1. Enable case and accent sensitivity for the user:

```
config user local
    edit "fgdocs"
        set username-sensitivity enable
```

```
    next
end
```

2. Connect to the Agentless VPN portal:
   a. Log in to the Agentless VPN portal with the username, using the same case that it is on the FortiGate.
   b. When prompted, enter your FortiToken code.
      You should now be connected.
3. Check the web portal log in using the CLI:

```
# get vpn ssl monitor
Agentless VPN Login Users:
 Index   User    Group   Auth Type      Timeout         From     HTTP in/out   HTTPS in/out
 0       fgdocs          LDAP-USERGRP    16(1)           289     192.168.2.202 0/0      0/0

Agentless VPN sessions:
 Index   User    Group   Source IP      Duration        I/O Bytes       Tunnel/Dest IP
 0       fgdocs          LDAP-USERGRP    192.168.2.202   45      99883/5572    10.212.134.200
```

4. Log out from Agentless VPN portal.
5. Reconnect to the Agentless VPN portal:
   a. Log in to the Agentless VPN portal with the username, using a different case than on the FortiGate.
      You will not be prompted for your FortiToken code. You should now be connected.
6. Check the web portal log in using the CLI:

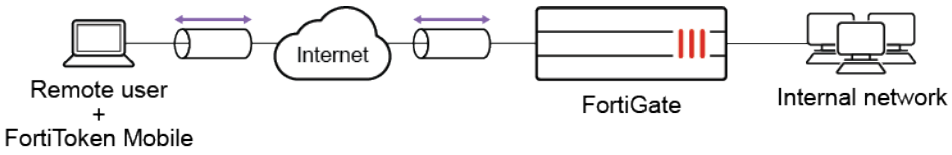
```
# get vpn ssl monitor
Agentless VPN Login Users:
 Index   User    Group   Auth Type      Timeout         From     HTTP in/out   HTTPS in/out
 0       FGdocs          LDAP-USERGRP    16(1)           289     192.168.2.202 0/0      0/0

Agentless VPN sessions:
 Index   User    Group   Source IP      Duration        I/O Bytes       Tunnel/Dest IP
```

In this case, the user is allowed to log in without a FortiToken code because the entered user name did not match the name defined on the remote LDAP user object. Authentication continues to be evaluated against the LDAP server though, which is not case sensitive.

Agentless VPN with FortiToken mobile push authentication

This is a sample configuration of Agentless VPN that uses FortiToken mobile push two-factor authentication. If you enable push notifications, users can accept or deny the authentication request.

Sample topology



Sample configuration

WAN interface is the interface connected to ISP. This example shows static mode. You can also use DHCP or PPPoE mode. The Agentless VPN connection is established over the WAN interface.

To configure Agentless VPN using the GUI:

1. Configure the interface and firewall address. The port1 interface connects to the internal network.
   a. Go to *Network* > *Interfaces* and edit the *wan1* interface.
   b. Set *IP/Network Mask* to *172.20.120.123/255.255.255.0*.
   c. Edit *port1* interface and set *IP/Network Mask* to *192.168.1.99/255.255.255.0*.
   d. Click *OK*.
   e. Go to *Policy & Objects* > *Address* and create an address for internet subnet *192.168.1.0*.
2. Register FortiGate for FortiCare Support:
   To add or download a mobile token on FortiGate, FortiGate must be registered for FortiCare Support. If your FortiGate is registered, skip this step.
   a. Go to *Dashboard* > *Licenses*.
   b. Hover the pointer on *Support* to check if FortiCare registered. If not, click it and select *Register*.
3. Add FortiToken mobile to FortiGate:
   If your FortiGate has FortiToken installed, skip this step.
   a. Go to *User & Authentication* > *FortiTokens* and click *Create New*.
   b. Select *Mobile Token* and type in *Activation Code*.
   c. Every FortiGate has two free mobile tokens. Go to *User & Authentication* > *FortiTokens* and click *Import Free Trial Tokens*.
4. Enable FortiToken mobile push:
   To use FTM-push authentication, use CLI to enable FTM-Push on the FortiGate.
   a. Ensure `server` is reachable from the Internet and enter the following CLI commands:

```
config system ftm-push
    set server 172.20.120.123
    set status enable
end
```

   b. Go to *Network* > *Interfaces*.
   c. Edit the *wan1* interface.
   d. Under *Administrative Access* > *IPv4*, select *FTM*.
   e. Click *OK*.
5. Configure user and user group:

a. Go to *User & Authentication > User Definition* to create a local user *agentlessvpnuser1*.
b. Enter the user's *Email Address*.
c. Enable *Two-factor Authentication* and select one mobile *Token* from the list,
d. Enable *Send Activation Code* and select *Email*.
e. Click *Next* and click *Submit*.
f. Go to *User & Authentication > User Groups* to create a group *agentlessvpngroup* with the member *agentlessvpnuser1*.

6. Activate the mobile token:
   a. When the user *agentlessvpnuser1* is created, an email is sent to the user's email address. Follow the instructions to install your FortiToken mobile application on your device and activate your token.
7. Configure Agentless VPN portals.
   a. Go to *VPN > Agentless VPN Portals*, and click *Create New*.
   b. Enter a name, such as *agentless-portal*.
   c. Under *Predefined Bookmarks*, click *Create New*, and enter the following details:

Field	Value
Name	Windows Server
Type	RDP
Host	192.168.1.114
Port	3389
Single Sign-On	Disable
Username	Set the username to log in to Windows Server.
Password	Set the password to log in to Windows Server.
Port	3389

   d. Click *OK* to save the predefined bookmark.
   e. Click *OK* to save the portal settings.
   f. Create another Agentless VPN portal and name it *portal-access-disabled*.
   g. Click *OK* to save the portal settings.
   h. Disable Agentless VPN for the newly created portal (that is, *portal-access-disabled*) using CLI:

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
end
```

8. Configure Agentless VPN settings.
   a. Go to *VPN > Agentless VPN Settings*.
   b. Set Agentless VPN to *Enable*.
   c. Set the *Listen on Interface(s)* to *wan1*.
   d. Set *Listen on Port* to *10443*.
   e. Choose a certificate for *Server Certificate*. The default is *Fortinet_Factory*.

f. Under *Authentication/Portal Mapping*, set select *All Other Users/Groups*, and click *Edit*.
g. Use *Portal* dropdown to select *portal-access-disabled*.
h. Click *OK*.
i. In *Authentication/Portal Mapping*, click *Create New*.
   i. Set *Users/Groups* to *agentlessvpngroup*.
   ii. Set *Portal* to *agentless-portal*.
   iii. Click *OK*.
j. Click *Apply* to save the Agentless VPN settings.

9. Configure Agentless VPN firewall policy:
   a. Go to *Policy & Objects > Firewall Policy*, and click *Create New*.
   b. Set *Name* to *Agentless VPN firewall policy*.
   c. Set *Schedule* to *always* and *Action* to *Accept*.
   d. Set *Incoming interface* to *Agentless VPN tunnel interface(ssl.root)*.
   e. Choose an *Outgoing Interface*. This example uses *port1*.
   f. Set the *Source* to *all* and *User/group* to *agentlessvpngroup*.
   g. Set *Destination Address*. In this example, the destination is the internal, protected subnet *192.168.1.0*.
   h. Set *Service* to *ALL*.
   i. Click *OK*.

To configure Agentless VPN using the CLI:

1. Configure the interface and firewall address.

```
config system interface
    edit "wan1"
        set vdom "root"
        set ip 172.20.120.123 255.255.255.0
    next
end
```

2. Configure internal interface and protected subnet, then connect the port1 interface to the internal network.

```
config system interface
    edit "port1"
        set vdom "root"
        set ip 192.168.1.99 255.255.255.0
    next
end
```

```
config firewall address
    edit "192.168.1.0"
        set subnet 192.168.1.0 255.255.255.0
    next
end
```

3. Register FortiGate for FortiCare Support.
   To add or download a mobile token on FortiGate, FortiGate must be registered for FortiCare Support. If your FortiGate is registered, skip this step.

```
diagnose forticare direct-registration product-registration -a "your account@xxx.com" -p "your
password" -T "Your Country/Region" -R "Your Reseller" -e 1
```

4. Add FortiToken mobile to FortiGate:

```
execute fortitoken-mobile import <your FTM code>
```

If your FortiGate has FortiToken installed, skip this step.
Every FortiGate has two free mobile Tokens. You can download the free token.

```
execute fortitoken-mobile import 0000-0000-0000-0000-0000
```

5. Enable FortiToken mobile push:
   a. To use FTM-push authentication, ensure `server` is reachable from the Internet and enable FTM-push in the FortiGate:

```
config system ftm-push
    set server 172.20.120.123
    set status enable
end
```

   b. Enable FTM service on WAN interface:

```
config system interface
    edit "wan1"
        append allowaccess ftm
    next
end
```

6. Configure user and user group:

```
config user local
    edit "agentlessvpnuser1"
        set type password
        set two-factor fortitoken
        set fortitoken <select mobile token for the option list>
        set email-to <user's email address>
        set passwd <user's password>
    next
end
config user group
    edit "agentlessvpngroup"
        set member "agentlessvpnuser1"
    next
end
```

7. Activate the mobile token.
   When the user *agentlessvpnuser1* is created, an email is sent to the user's email address. Follow the instructions to install your FortiToken mobile application on your device and activate your token.
8. Configure Agentless VPN portal and predefine RDP bookmark for windows server.

```
config vpn ssl web portal
    edit "agentless-portal"
        set web-mode enable
        config bookmark-group
            edit "gui-bookmarks"
                config bookmarks
                    edit "Windows Server"
                        set apptype rdp
                        set host "192.168.1.114"
                        set port 3389
                        set logon-user "your-windows-server-user-name"
                        set logon-password your-windows-server-password
                    next
                end
            next
        end
    next
end
```

9. Configure another Agentless VPN portal and disable Agentless VPN on it.

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
end
```

10. Configure Agentless VPN settings:

```
config vpn ssl settings
    set servercert "Fortinet_Factory"
    set source-interface "wan1"
    set source-address "all"
    set default-portal "portal-access-disabled"
    config authentication-rule
        edit 1
            set groups "agentlessvpngroup"
            set portal "agentless-portal"
        next
    end
end
```

11. Configure one Agentless VPN firewall policy to allow remote user to access the internal network. This policy does not allow traffic initiated from internal network to remote client.

```
config firewall policy
    edit 1
        set name "sslvpn web mode access"
        set srcintf "ssl.root"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "192.168.1.0"
```

```
        set groups “sslvpngroup”
        set action accept
        set schedule "always"
        set service "ALL"
        set nat enable
    next
end
```

To see the results of web portal:

1. From a remote device, use a web browser to log into the Agentless VPN web portal *http://172.20.120.123:10443*.
2. Log in using the *agentlessvpnuser1* credentials.
   The FortiGate pushes a login request notification through the FortiToken mobile application.
3. Check your mobile device and select *Approve*.
   When the authentication is approved, *agentlessvpnuser1* is logged into the Agentless VPN portal.

To check the Agentless VPN connection using the GUI:

1. On FortiGate, go to *Dashboard > Agentless VPN Monitor* to verify the list of Agentless VPN users.
   If the Agentless VPN Monitor is hidden, click *+* under the *Dashboard*, search for *Agentless VPN Monitor*, and add it to the display.
2. Go to *Log & Report > System Events*, and use the dropdown to select *VPN Events* to view VPN logs.

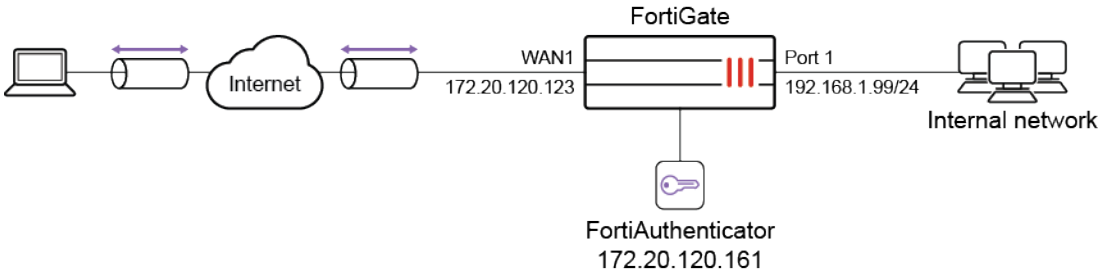
To check the web portal login using the CLI:

```
get vpn ssl monitor
Agentless VPN Login Users:
 Index   User           Auth Type    Timeout   From           HTTP in/out   HTTPS in/out
 0       agentlessvpnuser1    1(1)         229       10.1.100.254   0/0           0/0

Agentless VPN sessions:
 Index   User    Source IP      Duration        I/O Bytes       Tunnel/Dest IP
```

Agentless VPN with RADIUS on FortiAuthenticator

This is a sample configuration of Agentless VPN that uses FortiAuthenticator as a RADIUS authentication server.

Sample topology



Sample configuration

WAN interface is the interface connected to ISP. This example shows static mode. You can also use DHCP or PPPoE mode. The Agentless VPN connection is established over the WAN interface.

To configure FortiAuthenticator using the GUI:

1. Create a user on the FortiAuthenticator.
   a. On the FortiAuthenticator, go to *Authentication* > *User Management* > *Local Users* to create a user *agentlessvpnuser1*.
   b. Enable *Allow RADIUS authentication* and click *OK* to access additional settings.
   c. Go to *Authentication* > *User Management* > *User Groups* to create a group *agentlessvpngroup*.
   d. Add *agentlessvpnuser1* to the group by moving the user from *Available users* to *Selected users*.
2. Create the RADIUS client (FortiGate) on the FortiAuthenticator.
   a. On the FortiAuthenticator, go to *Authentication* > *RADIUS Service* > *Clients* to add the FortiGate as a RADIUS client *OfficeServer*).
   b. Enter the FortiGate IP address and set a *Secret*.
      The secret is a pre-shared secure password that the FortiGate uses to authenticate to the FortiAuthenticator.
   c. Set *Realms* to *local | Local users*.

To configure Agentless VPN using the GUI:

1. Configure the interface and firewall address. The port1 interface connects to the internal network.
   a. Go to *Network* > *Interfaces* and edit the *wan1* interface.
   b. Set *IP/Network Mask* to *172.20.120.123/255.255.255.0*.
   c. Edit *port1* interface and set *IP/Network Mask* to *192.168.1.99/255.255.255.0*.
   d. Click *OK*.
   e. Go to *Policy & Objects* > *Addresses* and create an address for internal subnet *192.168.1.0*.
2. Create a RADIUS user and user group .
   a. On the FortiGate, go to *User & Authentication* > *RADIUS Servers* to create a user to connect to the RADIUS server (FortiAuthenticator).
   b. For *Name*, use *FAC-RADIUS*.
   c. Enter the IP address of the FortiAuthenticator, and enter the *Secret* created above.

d. Click *Test Connectivity* to ensure you can connect to the RADIUS server.
e. Select *Test User Credentials* and enter the credentials for *sslvpnuser1*.
   The FortiGate can now connect to the FortiAuthenticator as the RADIUS client.
f. Go to *User & Authentication > User Groups* and click *Create New* to map authenticated remote users to a user group on the FortiGate.
g. For *Name*, use *AgentlessVPNGroup*.
h. In *Remote Groups*, click *Add*.
i. In the *Remote Server* dropdown list, select *FAC-RADIUS*.
j. Leave the *Groups* field blank.

3. Configure Agentless VPN portals.
   a. Go to *VPN > Agentless VPN Portals*, and click *Create New*.
   b. Enter a name, such as *agentless-portal*.
   c. Under *Predefined Bookmarks*, click *Create New*, and enter the following details:

Field	Value
Name	Windows Server
Type	RDP
Host	192.168.1.114
Port	3389
Single Sign-On	Disable
Username	Set the username to log in to Windows Server.
Password	Set the password to log in to Windows Server.
Port	3389

   d. Click *OK* to save the predefined bookmark.
   e. Click *OK* to save the portal settings.
   f. Create another Agentless VPN portal and name it *portal-access-disabled*.
   g. Click *OK* to save the portal settings.
   h. Disable Agentless VPN for the newly created portal (that is, *portal-access-disabled*) using CLI:

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
end
```

4. Configure Agentless VPN settings.
   a. Go to *VPN > Agentless VPN Settings*.
   b. Set *Agentless VPN* status to *Enable*.
   c. Set *Listen on Interface(s)* to *wan1*.
   d. Set *Listen on Port* to *10443*.
   e. Choose a certificate for *Server Certificate*. The default is *Fortinet_Factory*.
   f. In *Authentication/Portal Mapping*, select *All Other Users/Groups*, and click *Edit*.

g. Use *Portal* dropdown to select *portal-access-disabled*.
h. Click *OK*.
i. In *Authentication/Portal Mapping*, click *Create New*.
    i. Set *Users/Groups* to *AgentlessvpnGroup*.
    ii. Set *Portal* to *agentless-portal*.
    iii. Click *OK*.
j. Click *Apply* to save the Agentless VPN settings.

5. Configure Agentless VPN firewall policy.
a. Go to *Policy & Objects > Firewall Policy*, and click *Create New*.
b. Set *Name* to *Agentless VPN firewall policy*.
c. Set *Schedule* to *always* and *Action* to *Accept*.
d. Set *Incoming interface* to *Agentless VPN interface (ssl.root)*.
e. Choose an *Outgoing Interface*. This example uses *port1*.
f. Set the *Source* to *all* and *User/group* to *AgentlessVPNGroup*.
g. In this example, the *Destination* is the internal, protected subnet *192.168.1.0*.
h. Set *Service* to *ALL*.
i. Enable *NAT*.
j. Configure the remaining options as required.
k. Click *OK*.

To configure Agentless VPN using the CLI:

1. Configure the interface and firewall address.

```
config system interface
    edit "wan1"
        set vdom "root"
        set ip 172.20.120.123 255.255.255.0
    next
end
```

2. Configure internal interface and protected subnet, then connect the port1 interface to the internal network.

```
config system interface
    edit "port1"
        set vdom "root"
        set ip 192.168.1.99 255.255.255.0
    next
end
```

```
config firewall address
    edit "192.168.1.0"
        set subnet 192.168.1.0 255.255.255.0
    next
end
```

3. Create a RADIUS user and user group.

```
config user radius
    edit "FAC-RADIUS"
        set server "172.20.120.161"
        set secret <FAC client secret>
    next
end
```

```
config user group
    edit "sslvpngroup"
        set member "FAC-RADIUS"
    next
end
```

4. Configure Agentless VPN portal and predefine RDP bookmark for windows server.

```
config vpn ssl web portal
    edit "agentless-portal"
        set web-mode enable
        config bookmark-group
            edit "gui-bookmarks"
                config bookmarks
                    edit "Windows Server"
                        set apptype rdp
                        set host "192.168.1.114"
                        set port 3389
                        set logon-user "your-windows-server-user-name"
                        set logon-password your-windows-server-password
                    next
                end
            next
        end
    next
end
```

5. Configure another Agentless VPN portal and disable Agentless VPN on it.

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
end
```

6. Configure Agentless VPN settings.

```
config vpn ssl settings
    set banned-cipher SHA1 SHA256 SHA384
    set servercert "server_certificate"
    set port 1443
    set source-interface "wan1"
    set source-address "all"
    set default-portal "portal-access-disabled"
    config authentication-rule
```

```
        edit 1
            set groups "agentlessvpngroup"
            set portal "agentless-portal"
        next
    end
end
```

7. Configure one Agentless VPN firewall policy to allow remote users to access the internal network. This policy does not allow traffic initiated from internal network to remote client.

```
config firewall policy
    edit 1
        set name "Agentless VPN firewall policy"
        set srcintf "ssl.root"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "192.168.1.0"
        set action accept
        seet schedule "always"
        set service "ALL"
        set nat enable
        set groups "agentlessvpngroup"
    next
end
```

To see the results of web portal:

1. From a remote device, use a web browser to log into the Agentless VPN web portal *http://172.20.120.123:10443*.
2. Log in using the *agentlessvpnuser1* credentials.

To check the Agentless VPN connection using the GUI:

1. On FortiGate, go to *Dashboard > Agentless VPN Monitor* to verify the list of Agentless VPN users.
   If the Agentless VPN Monitor is hidden, click *+* under the *Dashboard*, search for *Agentless VPN Monitor*, and add it to the display.
2. Go to *Log & Report > System Events*, and use the dropdown to select *VPN Events* to view VPN logs.

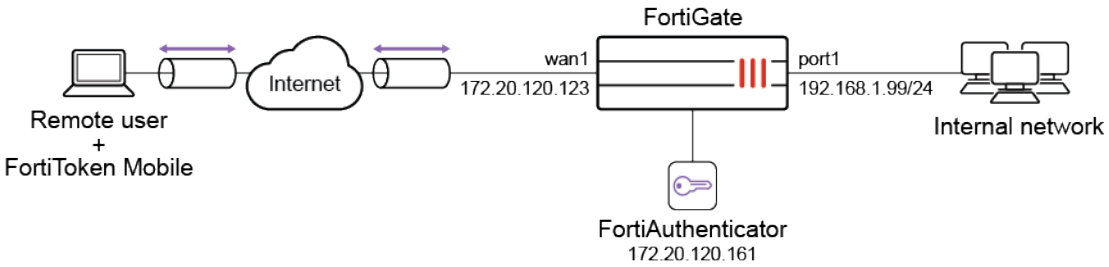
To check the web portal login using the CLI:

```
get vpn ssl monitor
Agentless VPN Login Users:
 Index   User          Auth Type    Timeout    From           HTTP in/out  HTTPS in/out
 0       agentlessvpnuser1   1(1)         229        10.1.100.254   0/0          0/0

Agentless VPN sessions:
 Index   User    Source IP      Duration        I/O Bytes       Tunnel/Dest IP
```

Agentless VPN with RADIUS and FortiToken mobile push on FortiAuthenticator

This is a sample configuration of Agentless VPN that uses FortiAuthenticator as a RADIUS authentication server and FortiToken mobile push two-factor authentication. If you enable push notifications, users can accept or deny the authentication request.

Sample topology



Sample configuration

WAN interface is the interface connected to ISP. This example shows static mode. You can also use DHCP or PPPoE mode. The Agentless VPN connection is established over the WAN interface.

To configure FortiAuthenticator using the GUI:

1. On the FortiAuthenticator, go to *System > Administration > System Access* and configure a *Public IP/FQDN for FortiToken Mobile*. If the FortiAuthenticator is behind a firewall, the public IP/FQDN will be an IP/port forwarding rule directed to one of the FortiAuthenticator interfaces. The interface that receives the approve/deny FTM push responses must have the *FortiToken Mobile API* service enabled.
2. Add a FortiToken mobile license on the FortiAuthenticator:
   a. Go to *Authentication > User Management > FortiTokens*.
   b. Click *Create New*.
   c. Set *Token type* to *FortiToken Mobile* and enter the *FortiToken Activation codes*.
3. Create the RADIUS client (FortiGate) on the FortiAuthenticator:
   a. Go to *Authentication > RADIUS Service > Clients* to add the FortiGate as a RADIUS client *OfficeServer*).
   b. Enter the FortiGate IP address and set a *Secret*.
      The secret is a pre-shared secure password that the FortiGate uses to authenticate to the FortiAuthenticator.
   c. Set *Authentication method* to *Enforce two-factor authentication*.
   d. Select *Enable FortiToken Mobile push notifications authentication*.
   e. Set *Realms* to *local | Local users*.
4. Create a user and assign FortiToken mobile to the user on the FortiAuthenticator:

a. Go to *Authentication > User Management > Local Users* to create a user *agentlessvpnuser1*.
b. Enable *Allow RADIUS authentication* and click *OK* to access additional settings.
c. Enable *Token-based authentication* and select to deliver the token code by *FortiToken*.
d. Select the FortiToken added from the FortiToken Mobile dropdown menu.
e. Set *Delivery method* to *Email* and fill in the *User Information* section.
f. Go to *Authentication > User Management > User Groups* to create a group *agentlessvpngroup*.
g. Add *agentlessvpnuser1* to the group by moving the user from *Available users* to *Selected users*.

5. Install the FortiToken mobile application on your Android or iOS smartphone.
   The FortiAuthenticator sends the FortiToken mobile activation to the user's email address.
6. Activate the FortiToken mobile through the FortiToken mobile application by entering the activation code or scanning the QR code.

To configure Agentless VPN using the GUI:

1. Configure the interface and firewall address. The port1 interface connects to the internal network.
   a. Go to *Network > Interfaces* and edit the *wan1* interface.
   b. Set *IP/Network Mask* to *172.20.120.123/255.255.255.0*.
   c. Edit *port1* interface and set *IP/Network Mask* to *192.168.1.99/255.255.255.0*.
   d. Click *OK*.
   e. Go to *Policy & Objects > Address* and create an address for internet subnet *192.168.1.0*.
2. Create a RADIUS user and user group:
   a. On the FortiGate, go to *User & Authentication > RADIUS Servers* to create a user to connect to the RADIUS server (FortiAuthenticator).
   b. For *Name*, use *FAC-RADIUS*.
   c. Enter the IP address of the FortiAuthenticator, and enter the *Secret* created above.
   d. Click *Test Connectivity* to ensure you can connect to the RADIUS server.
   e. Select *Test User Credentials* and enter the credentials for *agentlessvpnuser1*.
      The FortiGate can now connect to the FortiAuthenticator as the RADIUS client.
   f. Go to *User & Authentication > User Groups* and click *Create New* to map authenticated remote users to a user group on the FortiGate.
   g. For *Name*, use *AgentlessVPNGroup*.
   h. In *Remote Groups*, click *Add*.
   i. In the *Remote Server* dropdown list, select *FAC-RADIUS*.
   j. Leave the *Groups* field blank.
3. Configure Agentless VPN portals.
   a. Go to *VPN > Agentless VPN Portals*, and click *Create New*.
   b. Enter a name, such as *agentless-portal*.
   c. Under *Predefined Bookmarks*, click *Create New*, and enter the following details:

Field	Value
Name	Windows Server
Type	RDP

Field	Value
Host	192.168.1.114
Port	3389
Single Sign-On	Disable
Username	Set the username to log in to Windows Server.
Password	Set the password to log in to Windows Server.
Port	3389

d. Click *OK* to save the predefined bookmark.
e. Click *OK* to save the portal settings.
f. Create another Agentless VPN portal and name it *portal-access-disabled*.
g. Click *OK* to save the portal settings.
h. Disable Agentless VPN for the newly created portal (that is, *portal-access-disabled*) using CLI:

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
end
```

4. Configure Agentless VPN settings:
   a. Go to *VPN > Agentless VPN Settings*.
   b. Set *Agentless VPN status* to *Enable*.
   c. Set *Listen on Interface(s)* to *wan1*.
   d. Set *Listen on Port* to *10443*.
   e. Choose a certificate for *Server Certificate*. The default is *Fortinet_Factory*.
   f. In *Authentication/Portal Mapping*, select *All Other Users/Groups*, and click *Edit*.
   g. Use *Portal* dropdown to select *portal-access-disabled*.
   h. Click *OK*.
   i. In *Authentication/Portal Mapping*, click *Create New*.
      i. Set *Users/Groups* to *AgentlessVPNgroup*.
      ii. Set *Portal* to *agentless-portal*.
      iii. Click *OK*.
   j. Click *Apply* to save the Agentless VPN settings.
5. Configure Agentless VPN firewall policy.
   a. Go to *Policy & Objects > Firewall Policy*, and click *Create New*.
   b. Set *Name* to *Agentless VPN firewall policy*.
   c. Set *Schedule* to *always* and *Action* to *Accept*.
   d. Set *Incoming interface* to *Agentless VPN interface (ssl.root)*.
   e. Choose an *Outgoing Interface*. This example uses *port1*.
   f. Set the *Source* to *all* and *User/group* to *AgentlessVPNGroup*.
   g. In this example, the *Destination* is the internal, protected subnet *192.168.1.0*.

h. Set *Service* to *ALL*.
i. Enable *NAT*.
j. Configure any remaining firewall and security options as desired.
k. Click *OK*.

To configure Agentless VPN using the CLI:

1. Configure the interface and firewall address:

```
config system interface
    edit "wan1"
        set vdom "root"
        set ip 172.20.120.123 255.255.255.0
    next
end
```

2. Configure internal interface and protected subnet, then connect the port1 interface to the internal network:

```
config system interface
    edit "port1"
        set vdom "root"
        set ip 192.168.1.99 255.255.255.0
    next
end
```

```
config firewall address
    edit "192.168.1.0"
        set subnet 192.168.1.0 255.255.255.0
    next
end
```

3. Create a RADIUS user and user group:

```
config user radius
    edit "FAC-RADIUS"
        set server "172.20.120.161"
        set secret <FAC client secret>
    next
end
```

```
config user group
    edit "sslvpngroup"
        set member "FAC-RADIUS"
    next
end
```

4. Configure Agentless VPN portal and predefine RDP bookmark for windows server.

```
config vpn ssl web portal
    edit "agentless-portal"
        set web-mode enable
        config bookmark-group
```

```
            edit "gui-bookmarks"
                config bookmarks
                    edit "Windows Server"
                        set apptype rdp
                        set host "192.168.1.114"
                        set port 3389
                        set logon-user "your-windows-server-user-name"
                        set logon-password your-windows-server-password
                    next
                end
            next
        end
    next
end
```

5. Configure another Agentless VPN portal and disable Agentless VPN on it.

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
end
```

6. Configure Agentless VPN settings:

```
config vpn ssl settings
    set banned-cipher SHA1 SHA256 SHA384
    set servercert "server_certificate"
    set port 1443
    set source-interface "wan1"
    set source-address "all"
    set default-portal "portal-access-disabled"
    config authentication-rule
        edit 1
            set groups "agentlessvpngroup"
            set portal "agentless-portal"
        next
    end
end
```

7. Configure one Agentless VPN firewall policy to allow remote user to access the internal network:

```
config firewall policy
    edit 1
        set name "Agentless VPN firewall policy"
        set srcintf "ssl.root"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "192.168.1.0"
        set action accept
        set schedule "always"
        set service "ALL"
```

```
        set nat enable
        set groups “agentlessvpngroup”
    next
end
```

To see the results of web portal:

1. From a remote device, use a web browser to log into the Agentless VPN web portal *http://172.20.120.123:10443*.
2. Log in using the *agentlessvpnuser1* credentials.
   The FortiAuthenticator pushes a login request notification through the FortiToken Mobile application.
3. Check your mobile device and select *Approve*.
   When the authentication is approved, *agentlessvpnuser1* is logged into the Agentless VPN portal.

To check the Agentless VPN connection using the GUI:

1. On FortiGate, go to *Dashboard > Agentless VPN Monitor* to verify the list of Agentless VPN users.
   If the Agentless VPN Monitor is hidden, click *+* under the *Dashboard*, search for *Agentless VPN Monitor*, and add it to the display.
2. Go to *Log & Report > System Events*, and use the dropdown to select *VPN Events* to view VPN logs.

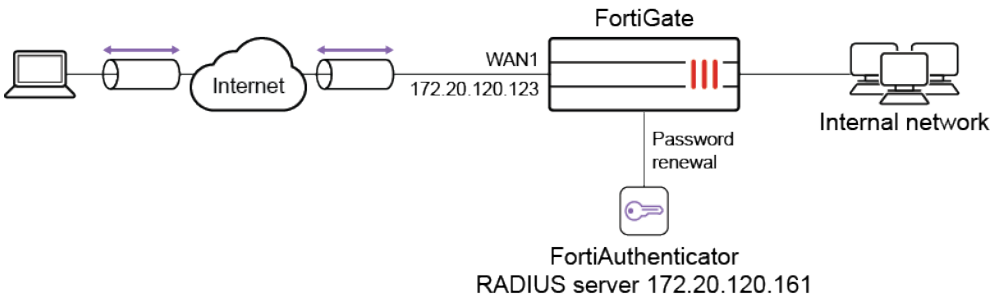
To check the Agentless VPN portal login using the CLI:

```
get vpn ssl monitor
Agentless VPN Login Users:
 Index   User          Auth Type   Timeout   From           HTTP in/out   HTTPS in/out
 0       agentlessvpnuser1   1(1)        229       10.1.100.254   0/0           0/0

Agentless VPN sessions:
 Index   User    Source IP      Duration        I/O Bytes       Tunnel/Dest IP
```

Agentless VPN with RADIUS password renew on FortiAuthenticator

This is a sample configuration of Agentless VPN for RADIUS users with *Force Password Change on next logon*. In this example, the RADIUS server is a FortiAuthenticator. A user *test1* is configured on FortiAuthenticator with *Force password change on next logon*.

Sample topology



Sample configuration

WAN interface is the interface connected to ISP. This example shows static mode. You can also use DHCP or PPPoE mode. The Agentless VPN connection is established over the WAN interface.

To configure Agentless VPN using the GUI:

1. Configure the interface and firewall address. The port1 interface connects to the internal network.
   a. Go to *Network* > *Interfaces* and edit the *wan1* interface.
   b. Set *IP/Network Mask* to *172.20.120.123/255.255.255.0*.
   c. Edit *port1* interface and set *IP/Network Mask* to *192.168.1.99/255.255.255.0*.
   d. Click *OK*.
   e. Go to *Policy & Objects* > *Address* and create an address for internet subnet *192.168.1.0*.
2. Create a RADIUS user.
   a. Go to *User & Authentication* > *RADIUS Servers* to create a user.
   b. Set *Authentication method* to *MS-CHAP-v2*.
   c. Enter the *IP/Name* and *Secret*.
   d. Click *Create*.
      Password renewal only works with the MS-CHAP-v2 authentication method.
   e. To enable the `password-renew` option, use these CLI commands.

```
config user radius
    edit "fac"
        set server "172.20.120.161"
        set secret <fac radius password>
        set auth-type ms_chap_v2
        set password-renewal enable
    next
end
```

3. Configure user group.
   a. Go to *User & Authentication* > *User Groups* to create a user group.
   b. For the *Name*, enter *fac-group*.
   c. In *Remote Groups*, click *Add* to add *Remote Server* you just created.

4. Configure Agentless VPN portal and predefine RDP bookmark for windows server.

```
config vpn ssl web portal
    edit "agentless-portal"
        set web-mode enable
        config bookmark-group
            edit "gui-bookmarks"
                config bookmarks
                    edit "Windows Server"
                        set apptype rdp
                        set host "192.168.1.114"
                        set port 3389
                        set logon-user "your-windows-server-user-name"
                        set logon-password your-windows-server-password
                    next
                end
            next
        end
    next
end
```

5. Configure another Agentless VPN portal and disable Agentless VPN on it.

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
end
```

6. Configure Agentless VPN settings.
   a. Go to *VPN > Agentless VPN Settings*.
   b. Set *Agentless VPN* status to *Enable*.
   c. Set *Listen on Interface(s)* to *wan1*.
   d. Set *Listen on Port* to *10443*.
   e. Choose a certificate for *Server Certificate*. The default is *Fortinet_Factory*.
   f. In *Authentication/Portal Mapping*, select *All Other Users/Groups*, and click *Edit*.
   g. Use *Portal* dropdown to select *portal-access-disabled*.
   h. Click *OK*.
   i. In *Authentication/Portal Mapping*, click *Create New*.
      i. Set *Users/Groups* to *fac-group*.
      ii. Set *Portal* to *agentless-portal*.
      iii. Click *OK*.
   j. Click *Apply* to save the Agentless VPN settings.
7. Configure Agentless VPN firewall policy.
   a. Go to *Policy & Objects > Firewall Policy*, and click *Create New*.
   b. Set *Name* to *Agentless VPN firewall policy*.
   c. Set *Schedule* to *always* and *Action* to *Accept*.
   d. Set *Incoming interface* to *Agentless VPN interface (ssl.root)*.
   e. Choose an *Outgoing Interface*. This example uses *port1*.

f. Set the *Source* to *all* and *User/group* to *fac-group*.
g. In this example, the *Destination* is the internal, protected subnet *192.168.1.0*.
h. Set *Service* to *ALL*.
i. Click *OK*.

To configure Agentless VPN using the CLI:

1. Configure the interface and firewall address.

```
config system interface
    edit "wan1"
        set vdom "root"
        set ip 172.20.120.123 255.255.255.0
    next
end
```

2. Configure internal interface and protected subnet, then connect the port1 interface to the internal network.

```
config system interface
    edit "port1"
        set vdom "root"
        set ip 192.168.1.99 255.255.255.0
    next
end
```

```
config firewall address
    edit "192.168.1.0"
        set subnet 192.168.1.0 255.255.255.0
    next
end
```

3. Configure the RADIUS server.

```
config user radius
    edit "fac"
        set server "172.18.58.107"
        set secret <fac radius password>
        set auth-type ms_chap_v2
        set password-renewal enable
    next
end
```

4. Configure user group.

```
config user group
    edit "fac-group"
        set member "fac"
    next
end
```

5. Configure Agentless VPN portal and predefine RDP bookmark for windows server.

```
config vpn ssl web portal
    edit "agentless-portal"
        set web-mode enable
        config bookmark-group
            edit "gui-bookmarks"
                config bookmarks
                    edit "Windows Server"
                        set apptype rdp
                        set host "192.168.1.114"
                        set port 3389
                        set logon-user "your-windows-server-user-name"
                        set logon-password your-windows-server-password
                    next
                end
            next
        end
    next
end
```

6. Configure another Agentless VPN portal and disable Agentless VPN on it.

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
end
```

7. Configure Agentless VPN settings.

```
config vpn ssl settings
    set servercert "server_certificate"
    set source-interface "wan1"
    set source-address "all"
    set default-portal "portal-access-disabled"
    config authentication-rule
        edit 1
            set groups "fac-group"
            set portal "agentless-portal"
        next
    end
end
```

8. Configure one Agentless VPN firewall policy to allow remote users to access the internal network. This policy does not allow traffic initiated from internal network to remote client.

```
config firewall policy
    edit 1
        set name "Agentless VPN firewall policy"
        set srcintf "ssl.root"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "192.168.1.0"
```

```
        set action accept
        set schedule "always"
        set service "ALL"
        set nat enable
        set groups “fac-group”
    next
end
```

To see the results of web portal:

1. From a remote device, use a web browser to log into the Agentless VPN web portal *http://172.20.120.123:10443*.
2. Log in using the *test1* credentials.
   Use a user which is configured on FortiAuthenticator with *Force password change on next logon*.
3. Click *Login*. You are prompted to enter a new password.
4. On the FortiGate, go to Agentless VPN dashboard to verify the user’s connection. See Adding an Agentless VPN dashboard on page 200.

To check the Agentless VPN connection using the GUI:

1. On FortiGate, go to *Dashboard > Agentless VPN Monitor* to verify the list of Agentless VPN users.
   If the Agentless VPN Monitor is hidden, click *+* under the *Dashboard*, search for *Agentless VPN Monitor*, and add it to the display.
2. Go to *Log & Report > System Events*, and use the dropdown to select *VPN Events* to view VPN logs.

To check the Agentless VPN portal login using the CLI:

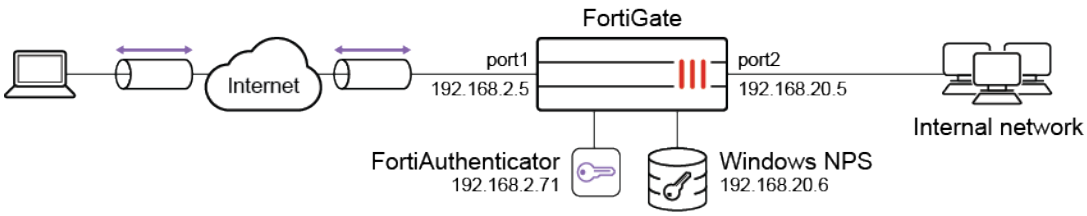
```
get vpn ssl monitor
Agentless VPN Login Users:
 Index   User    Auth Type    Timeout    From           HTTP in/out   HTTPS in/out
 0       test1   1(1)         229        10.1.100.254   0/0           0/0

Agentless VPN sessions:
 Index   User    Source IP      Duration        I/O Bytes       Tunnel/Dest IP
```

Agentless VPN with RADIUS on Windows NPS

This is an example configuration of Agentless VPN that uses Windows Network Policy Server (NPS) as a RADIUS authentication server.

The NPS must already be configured to accept the FortiGate as a RADIUS client and the choice of authentication method, such as MS-CHAPv2. A shared key must also have been created.

Example



The user is connecting from their PC to the FortiGate's port1 interface. RADIUS authentication occurs between the FortiGate and the Windows NPS, and the Agentless VPN connection is established once the authentication is successful.

Configure Agentless VPN with RADIUS on Windows NPS in the GUI

To configure the internal and external interfaces:

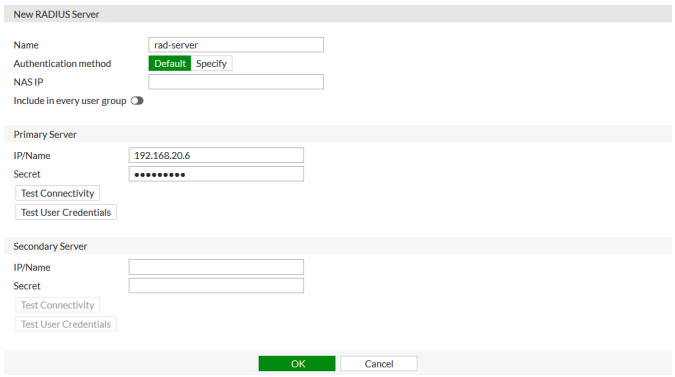
1. Go to *Network* > *Interfaces*
2. Edit the *port1* interface and set *IP/Network Mask* to *192.168.2.5/24*.
3. Edit the *port2* interface and set *IP/Network Mask* to *192.168.20.5/24*.
4. Click *OK*.

To create a firewall address:

1. Go to *Policy & Objects* > *Addresses* and select *Address*.
2. Click *Create new*.
3. Set *Name* to *192.168.20.0*.
4. Leave *Type* as *Subnet*
5. Set *IP/Netmask* to *192.168.20.0/24*.
6. Click *OK*.

To add the RADIUS server:

1. Go to *User & Authentication* > *RADIUS Servers* and click *Create New*.
2. Set *Name* to *rad-server*.
3. Leave *Authentication method* set to *Default*. The PAP, MS-CHAPv2, and CHAP methods will be tried in order.
4. Under *Primary Server*, set *IP/Name* to *192.168.20.6* and *Secret* to the shared secret configured on the RADIUS server.
5. Click *Test Connectivity* to test the connection to the server, and ensure that *Connection status* is *Successful*.
6. Optionally, click *Test User Credentials* to test user credentials. Testing from the GUI is limited to PAP.



7. Click *OK*.

To configure a user group:

1. Go to *User & Authentication* > *User Groups* and click *Create New*.
2. Set *Name* to *rad-group*.
3. Under *Remote Groups*, click *Add* and add the *rad-server*.
4. Click *OK*.

To configure Agentless VPN portals:

1. Configure Agentless VPN portals.
   a. Go to *VPN* > *Agentless VPN Portals*, and click *Create New*.
   b. Enter a name, such as *agentless-portal*.
   c. Under *Predefined Bookmarks*, click *Create New*, and enter the following details:

Field	Value
Name	Windows Server
Type	RDP
Host	192.168.1.114
Port	3389
Single Sign-On	Disable
Username	Set the username to log in to Windows Server.
Password	Set the password to log in to Windows Server.
Port	3389

   d. Click *OK* to save the predefined bookmark.
   e. Click *OK* to save the portal settings.
   f. Create another Agentless VPN portal and name it *portal-access-disabled*.
   g. Click *OK* to save the portal settings.
   h. Disable Agentless VPN for the newly created portal (that is, *portal-access-disabled*) using CLI:

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
end
```

To configure Agentless VPN settings:

1. Go to *VPN > Agentless VPN Settings*.
2. Set Agentless VPN to *Enable*.
3. Set the *Listen on Interface(s)* to *wan1*.
4. Set *Listen on Port* to *10443*.
5. Set *Server Certificate* to the authentication certificate.
6. In *Authentication/Portal Mapping*, select *All Other Users/Groups*, and click *Edit*.
7. Use *Portal* dropdown to select *portal-access-disabled*.
8. In *Authentication/Portal Mapping*, click *Create New*.
   a. Set *Users/Groups* to *rad-group*.
   b. Set *Portal* to *agentless-portal*.
   c. Click *OK*.
9. Click *Apply* to save the Agentless VPN settings.

To configure an Agentless VPN firewall policy:

1. Go to *Policy & Objects > Firewall Policy* and click *Create New*.
2. Set *Name* as *Agentless VPN firewall policy*.
3. Set *Schedule* to *always* and *Action* to *Accept*.
4. Set *Incoming Interface* to *Agentless VPN tunnel interface (ssl.root)*.
5. Choose an *Outgoing Interface*. This example uses *port1*.
6. Set *Source* to *all* and *User/group* to *rad-group*.
7. In this example, the *Destination* is the internal, protected subnet *192.168.20.0*.
8. Set *Service* to *ALL*.
9. Click *OK*.

Configure Agentless VPN with RADIUS on Windows NPS in the CLI

To configure Agentless VPN using the CLI:

1. Configure the internal and external interfaces:

```
config system interface
    edit "port1"
        set vdom "root"
        set ip 192.168.2.5 255.255.255.0
        set alias internal
    next
    edit "port2"
```

```
        set vdom "root"
        set ip 192.168.20.5 255.255.255.0
        set alias external
    next
end
```

2. Configure the firewall address:

```
config firewall address
    edit "192.168.20.0"
        set subnet 192.168.20.0 255.255.255.0
    next
end
```

3. Add the RADIUS server:

```
config user radius
    edit "rad-server"
        set server "192.168.20.6"
        set secret *********
    next
end
```

4. Create a user group and add the RADIUS server to it:.

```
config user group
    edit "rad-group"
        set member "rad-server"
    next
end
```

5. Configure Agentless VPN portal and predefine RDP bookmark for windows server.

```
config vpn ssl web portal
    edit "agentless-portal"
        set web-mode enable
        config bookmark-group
            edit "gui-bookmarks"
                config bookmarks
                    edit "Windows Server"
                        set apptype rdp
                        set host "192.168.1.114"
                        set port 3389
                        set logon-user "your-windows-server-user-name"
                        set logon-password your-windows-server-password
                    next
                end
            next
        end
    next
end
```

6. Configure another Agentless VPN portal and disable Agentless VPN on it.

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
end
```

7. Configure Agentless VPN settings:

```
config vpn ssl settings
    set servercert "server_certificate"
    set source-interface "wan1"
    set source-address "all"
    set default-portal "portal-access-disabled"
    config authentication-rule
        edit 1
            set groups "rad-group"
            set portal "agentless-portal"
        next
    end
end
```

8. Configure Agentless VPN firewall policy to allow remote user to access the internal network. . This policy does not allow traffic initiated from internal network to remote client.

```
config firewall policy
   edit 1
        set name "Agentless VPN firewall policy"
        set srcintf "ssl.root"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "192.168.1.0"
        set action accept
        set schedule "always"
        set service "ALL"
        set nat enable
        set groups “rad-group”
    next
end
```

Results

To check the Agentless VPN connection using the GUI:

1. On FortiGate, go to *Dashboard > Agentless VPN Monitor* to verify the list of Agentless VPN users.
   If the Agentless VPN Monitor is hidden, click *+* under the *Dashboard*, search for *Agentless VPN Monitor*, and add it to the display.
2. Go to *Log & Report > System Events*, and use the dropdown to select *VPN Events* to view VPN logs.

To check Agentless VPN connection using the CLI:

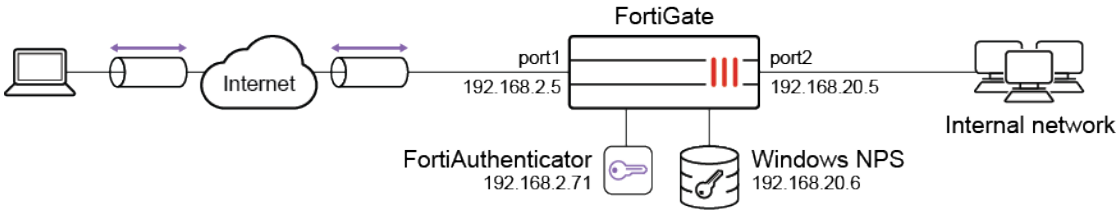
```
# get vpn ssl monitor
Agentless VPN Login Users:
 Index   User    Group   Auth Type      Timeout         From     HTTP in/out    HTTPS in/out
 0       radkeith        rad-group        2(1)            295     192.168.2.202  0/0     0/0

Agentless VPN sessions:
 Index   User    Group   Source IP      Duration        I/O Bytes       Tunnel/Dest IP
```

Agentless VPN with multiple RADIUS servers

When configuring two or more RADIUS servers, you can configure a Primary and Secondary server within the same RADIUS server configurations for backup purposes. You can also configure multiple RADIUS servers within the same User Group to service the access request at the same time.

A tertiary server can be configured in the CLI.

Sample topology



Sample configurations

- Configure a Primary and Secondary server for backup on page 3441
- Authenticating to two RADIUS servers concurrently on page 3447

Configure a Primary and Secondary server for backup

When you define a Primary and Secondary RADIUS server, the access request will always be sent to the Primary server first. If the request is denied with an Access-Reject, then the user authentication fails. However, if there is no response from the Primary server after another attempt, the access request will be sent to the Secondary server.

In this example, you will use a Windows NPS server as the Primary server and a FortiAuthenticator as the Secondary server. It is assumed that users are synchronized between the two servers.

To configure the internal and external interfaces:

1. Go to *Network* > *Interfaces*.
2. Edit the *port1* interface and set *IP/Network Mask* to *192.168.2.5/24*.
3. Edit the *port2* interface and set *IP/Network Mask* to *192.168.20.5/24*.
4. Click *OK*.

To create a firewall address:

1. Go to *Policy & Objects* > *Addresses* and select *Address*.
2. Click *Create new*.
3. Set *Name* to *192.168.20.0*.
4. Leave *Type* as *Subnet*
5. Set IP/*Netmask* to *192.168.20.0/24*.
6. Click *OK*.

To add the RADIUS servers:

1. Go to *User & Authentication* > *RADIUS Servers* and click *Create New*.
2. Set *Name* to *PrimarySecondary*.
3. Leave *Authentication method* set to *Default*. The PAP, MS-CHAPv2, and CHAP methods will be tried in order.
4. Under *Primary Server*, set *IP/Name* to *192.168.20.6* and *Secret* to the shared secret configured on the RADIUS server.
5. Click *Test Connectivity* to test the connection to the server, and ensure that *Connection status* is *Successful*.
6. Under *Secondary Server*, set *IP/Name* to *192.168.2.71* and *Secret* to the shared secret configured on the RADIUS server.
7. Click *Test Connectivity* to test the connection to the server, and ensure that *Connection status* is *Successful*.
8. *Click OK*.

To configure the user group:

1. Go to *User & Authentication* > *User Groups* and click *Create New*.
2. In the *Name* field, enter *PrimarySecondaryGroup*.
3. In the *Remote Groups* area, click *Add*, and from the *Remote Server* dropdown, select *PrimarySecondary*.
4. Click *OK*, and then click *OK* again.

To configure Agentless VPN portal:

1. Go to *VPN* > *Agentless VPN Portals*, and click *Create New*.
2. Enter a *Name*, such as *agentless-portal*.
3. Under *Predefined Bookmarks*, click *Create New*, and enter following details:

Field	Value
Name	Windows Server
Type	RDP
Host	192.168.1.114
Port	3389
Single Sign-On	Disable
Username	Set the username to login to Windows Server
Password	Set the password to login to Windows Server
Port	3389
Single Sign-On	Disable

4. Click *OK* to save the predefined bookmark.
5. Click *OK* to save the portal settings.
6. Create another Agentless VPN Portal and name it as *portal-access-disabled*.
7. Click *OK*.
8. Disable Agentless VPN for the newly created portal (that is, `portal-access-disabled`) using CLI:

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
end
```

To configure the Agentless VPN settings:

1. Go to *VPN > Agentless VPN Settings*.
2. Set Agentless VPN to *Enable*.
3. Set the *Listen on Interface(s)* to *wan1*.
4. Set *Listen on Port* to *10443*.
5. Set *Server Certificate* to the authentication certificate.
6. In *Authentication/Portal Mapping*, select *All Other Users/Groups*, and click *Edit*.
7. Use *Portal* dropdown to select *portal-access-disabled*.
8. In *Authentication/Portal Mapping*, click *Create New*.
   a. Set *Users/Groups* to *PrimarySecondaryGroup*.
   b. Set *Portal* to *agentless-portal*.
   c. Click *OK*.
9. Click *Apply* to save the Agentless VPN settings.

To configure Agentless VPN firewall policy:

1. Go to *Policy & Objects > Firewall Policy* and click *Create New*.
2. Set *Name* as *Agentless VPN firewall policy*.

3. Set *Schedule* to *always* and *Action* to *Accept*.
4. Set *Incoming interface* to *Agentless VPN interface (ssl.root)*.
5. Choose an *Outgoing Interface*. This example uses *port1*.
6. Set the *Source* to *all* and *User/group* to *PrimarySecondaryGroup*.
7. In this example, the *Destination* is the internal, protected subnet *192.168.1.0*.
8. Set *Service* to *ALL*.
9. Click *OK*.

To configure Agentless VPN using the CLI:

1. Configure the internal interface and firewall address:

```
config system interface
    edit "port3"
        set vdom "root"
        set ip 192.168.20.5 255.255.255.0
        set alias "internal"
    next
                <p class="CLI_0">end
                <p class="CLI_0">config firewall address
    edit "192.168.20.0"
        set uuid cc41eec2-9645-51ea-d481-5c5317f865d0
        set subnet 192.168.20.0 255.255.255.0
    next
end
```

2. Configure the RADIUS server:

```
config user radius
    edit "PrimarySecondary"
        set server "192.168.20.6"
        set secret <secret>
        set secondary-server "192.168.2.71"
        set secondary-secret <secret>
    next
end
```

3. Add the RADIUS user to the user group:

```
config user group
    edit "PrimarySecondaryGroup"
        set member "PrimarySecondary "
    next
end
```

4. Configure Agentless VPN portal and predefine RDP bookmark for windows server.

```
config vpn ssl web portal
    edit "agentless-portal"
        set web-mode enable
        config bookmark-group
```

```
            edit "gui-bookmarks"
                config bookmarks
                    edit "Windows Server"
                        set apptype rdp
                        set host "192.168.1.114"
                        set port 3389
                        set logon-user "your-windows-server-user-name"
                        set logon-password your-windows-server-password
                    next
                end
            next
        end
    next
end
```

5. Configure another Agentless VPN portal and disable Agentless VPN on it.

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
end
```

6. Configure Agentless VPN settings:

```
config vpn ssl settings
    set servercert "server_certificate"
    set source-interface "wan1"
    set source-address "all"
    set default-portal "portal-access-disabled"
    config authentication-rule
        edit 1
            set groups "PrimarySecondaryGroup"
            set portal "agentless-portal"
        next
    end
end
```

7. Configure one Agentless VPN firewall policy to allow remote users to access the internal network. This policy does not allow traffic initiated from internal network to remote client.

```
config firewall policy
    edit 1
        set name "Agentless VPN firewall policy"
        set srcintf "ssl.root"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "192.168.1.0"
        set action accept
        set schedule "always"
        set service "ALL"
        set nat enable
```

```
        set groups “PrimarySecondaryGroup”
    next
end
```

To verify the connection:

User *radkeith* is a member of both the NPS server and the FAC server.

When the Primary server is up, the user successfully connects to Agentless VPN.

```
# diagnose sniffer packet any 'port 1812' 4 0 l
interfaces=[any]
filters=[port 1812]
2020-05-15 16:26:50.838453 port3 out 192.168.20.5.2374 -> 192.168.20.6.1812: udp 118
2020-05-15 16:26:50.883166 port3 in 192.168.20.6.1812 -> 192.168.20.5.2374: udp 20
2020-05-15 16:26:50.883374 port3 out 192.168.20.5.2374 -> 192.168.20.6.1812: udp 182
2020-05-15 16:26:50.884683 port3 in 192.168.20.6.1812 -> 192.168.20.5.2374: udp 228
```

The access request is sent to the Primary NPS server 192.168.20.6, and the connection is successful.

```
# get vpn ssl monitor
Agentless VPN Login Users:
Index   User            Group                   Auth Type      Timeout         From     HTTP
in/out    HTTPS in/out
0       radkeith       PrimarySecondaryGroup   2(1)            285     192.168.2.202          0/0
   0/0
Agentless VPN sessions:
Index   User       Group                   Source IP      Duration        I/O Bytes
Tunnel/Dest IP
```

When the Primary server is down, and the Secondary server is up, the user successfully connects to Agentless VPN again:

```
# diagnose sniffer packet any 'port 1812' 4 0 l
interfaces=[any]
filters=[port 1812]
2020-05-15 16:31:23.016875 port3 out 192.168.20.5.7989 -> 192.168.20.6.1812: udp 118
2020-05-15 16:31:28.019470 port3 out 192.168.20.5.7989 -> 192.168.20.6.1812: udp 118
2020-05-15 16:31:30.011874 port1 out 192.168.2.5.23848 -> 192.168.2.71.1812: udp 118
2020-05-15 16:31:30.087564 port1 in 192.168.2.71.1812 -> 192.168.2.5.23848: udp 20
```

Access request is sent to the Primary NPS server 192.168.20.6, but there was no response. RADIUS authentication falls through to the Secondary FortiAuthenticator 192.168.2.71, and the authentication was accepted. The Agentless VPN connection is established.

```
# get vpn ssl monitor
Agentless VPN Login Users:
Index   User            Group                  Auth Type      Timeout         From     HTTP in/out
   HTTPS in/out
0       radkeith        PrimarySecondaryGroup   2(1)            287     192.168.2.202        0/0
     0/0
Agentless VPN sessions:
```

```
Index   User            Group                    Source IP      Duration        I/O Bytes
Tunnel/Dest IP
```

Authenticating to two RADIUS servers concurrently

There are times where users are located on separate RADIUS servers. This may be the case when migrating from an old server to a new one for example. In this scenario, a Windows NPS server and a FortiAuthenticator are configured in the same User Group. The access-request is sent to both servers concurrently. If FortiGate receives an access-accept from either server, authentication is successful.

To configure the internal and external interfaces:

1. Go to *Network* > *Interfaces*.
2. Edit the *port1* interface and set *IP/Network Mask* to *192.168.2.5/24*.
3. Edit the *port2* interface and set *IP/Network Mask* to *192.168.20.5/24*.
4. Click *OK*.

To create a firewall address:

1. Go to *Policy & Objects* > *Addresses* and select *Address*.
2. Click *Create new*.
3. Set *Name* to *192.168.20.0*.
4. Leave *Type* as *Subnet*
5. Set IP/*Netmask* to *192.168.20.0/24*.
6. Click *OK*.

To configure the first RADIUS server:

1. Go to *User & Authentication* > *RADIUS Servers* and click *Create New*.
2. Set *Name* to *win2k16*.
3. Leave *Authentication method* set to *Default*. The PAP, MS-CHAPv2, and CHAP methods will be tried in order.
4. Under *Primary Server*, set *IP/Name* to *192.168.20.6* and *Secret* to the shared secret configured on the RADIUS server.
5. Click *Test Connectivity* to test the connection to the server, and ensure that *Connection status* is *Successful*.
6. Click *OK*.

To configure the second RADIUS server:

1. Go to *User & Authentication* > *RADIUS Servers* and click *Create New*.
2. Set *Name* to *fac*.
3. Leave *Authentication method* set to *Default*. The PAP, MS-CHAPv2, and CHAP methods will be tried in order.

4. Under *Primary Server*, set *IP/Name* to *192.168.2.71* and *Secret* to the shared secret configured on the RADIUS server.
5. Click *Test Connectivity* to test the connection to the server, and ensure that *Connection status* is *Successful*.
6. Click *OK*.

To configure the user group:

1. Go to *User & Authentication > User Groups* and click *Create New*.
2. In the *Name* field, enter *dualPrimaryGroup.*.
3. In the *Remote Groups* area, click *Add*, and from the *Remote Server* dropdown, select *fac*.
4. Click *Add* again. From the *Remote Server* dropdown select *win2k16* and click *OK*.
5. Click *OK*, and then click *OK* again.

To configure Agentless VPN portals:

1. Go to *VPN > Agentless VPN Portals*, and click *Create New*.
2. Enter a *Name*, such as *agentless-portal*.
3. Under *Predefined Bookmarks*, click *Create New*, and enter following details:

Field	Value
Name	Windows Server
Type	RDP
Host	192.168.1.114
Port	3389
Single Sign-On	Disable
Username	Set the username to login to Windows Server
Password	Set the password to login to Windows Server
Port	3389
Single Sign-On	Disable

4. Click *OK* to save the predefined bookmark.
5. Click *OK* to save the portal settings.
6. Create another Agentless VPN Portal and name it as *portal-access-disabled*.
7. Click *OK*.
8. Disable Agentless VPN for the newly created portal (that is, `portal-access-disabled`) using CLI:

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
end
```

To configure the Agentless VPN settings:

1. Go to *VPN > Agentless VPN Settings*.
2. Set Agentless VPN to *Enable*.
3. Set the *Listen on Interface(s)* to *wan1*.
4. Set *Listen on Port* to *10443*.
5. Set *Server Certificate* to the authentication certificate.
6. In *Authentication/Portal Mapping*, select *All Other Users/Groups*, and click *Edit*.
7. Use *Portal* dropdown to select *portal-access-disabled*.
8. In *Authentication/Portal Mapping*, click *Create New*.
   a. Set *Users/Groups* to *dualPrimaryGroup*.
   b. Set *Portal* to *agentless-portal*.
   c. Click *OK*.
9. Click *Apply* to save the Agentless VPN settings.

To configure Agentless VPN firewall policy:

1. Go to *Policy & Objects > Firewall Policy* and click *Create New*.
2. Set *Name* as *Agentless VPN firewall policy*.
3. Set *Schedule* to *always* and *Action* to *Accept*.
4. Set *Incoming Interface* to *Agentless VPN tunnel interface(ssl.root)*.
5. Choose an *Outgoing Interface*. This example uses *port1*.
6. Set *Source* to *all* and *User/group* to *dualPrimaryGroup*.
7. In this example, the *Destination* is the internal, protected subnet *192.168.10.0*.
8. Set *Service* to *ALL*.
9. Click *OK*.

To configure Agentless VPN using the CLI:

1. Configure the internal interface and firewall address:

```
config system interface
    edit "port2"
        set vdom "root"
        set ip 192.168.20.5 255.255.255.0
        set alias "internal"
    next
end
config firewall address
    edit "192.168.20.0"
        set uuid cc41eec2-9645-51ea-d481-5c5317f865d0
        set subnet 192.168.20.0 255.255.255.0
    next
end
```

2. Configure the RADIUS server:

```
config user radius
    edit "win2k16"
        set server "192.168.20.6"
        set secret <secret>
    next
    edit "fac"
        set server "192.168.2.71"
        set secret <secret>
    next
end
```

3. Add the RADIUS user to the user group:

```
config user group
    edit "dualPrimaryGroup"
        set member "win2k16" “fac”
    next
end
```

4. Configure Agentless VPN portal and predefine RDP bookmark for windows server.

```
config vpn ssl web portal
    edit "agentless-portal"
        set web-mode enable
        config bookmark-group
            edit "gui-bookmarks"
                config bookmarks
                    edit "Windows Server"
                        set apptype rdp
                        set host "192.168.1.114"
                        set port 3389
                        set logon-user "your-windows-server-user-name"
                        set logon-password your-windows-server-password
                    next
                end
            next
        end
    next
end
```

5. Configure another Agentless VPN portal and disable Agentless VPN on it.

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
end
```

6. Configure Agentless VPN settings:

```
config vpn ssl settings
    set servercert "server_certificate"
```

```
    set source-interface "wan1"
    set source-address "all"
    set default-portal "portal-access-disabled"
    config authentication-rule
        edit 1
            set groups "dualPrimaryGroup"
            set portal "agentless-portal"
        next
    end
end
```

7. Configure one Agentless VPN firewall policy to allow remote users to access the internal network. This policy does not allow traffic initiated from internal network to remote client.

```
config firewall policy
   edit 1
        set name "Agentless VPN firewall policy"
        set srcintf "ssl.root"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "192.168.1.0"
        set action accept
        set schedule "always"
        set service "ALL"
        set nat enable
        set groups “dualPrimaryGroup”
    next
end
```

To verify the connection:

User *fackeith* is a member of the FortiAuthenticator server only.

User *radkeith* is a member of both the NPS server and the FortiAuthenticator server, but has different passwords on each server.

Case 1: Connect to the Agentless VPN web portal using a web browser and authentication with user FacAdmin:

```
# diagnose sniffer packet any 'port 1812' 4 0 l
interfaces=[any]
filters=[port 1812]
2020-05-15 17:21:31.217985 port3 out 192.168.20.5.11490 -> 192.168.20.6.1812: udp 118
2020-05-15 17:21:31.218091 port1 out 192.168.2.5.11490 -> 192.168.2.71.1812: udp 118
2020-05-15 17:21:31.219314 port3 in 192.168.20.6.1812 -> 192.168.20.5.11490: udp 20 <--
access-reject
2020-05-15 17:21:31.219519 port3 out 192.168.20.5.11490 -> 192.168.20.6.1812: udp 182
2020-05-15 17:21:31.220219 port3 in 192.168.20.6.1812 -> 192.168.20.5.11490: udp 42
2020-05-15 17:21:31.220325 port3 out 192.168.20.5.11490 -> 192.168.20.6.1812: udp 119
2020-05-15 17:21:31.220801 port3 in 192.168.20.6.1812 -> 192.168.20.5.11490: udp 20
```

```
2020-05-15 17:21:31.236009 port1 in 192.168.2.71.1812 -> 192.168.2.5.11490: udp 20 <--access-
accept
```

Access is denied by the NPS server because the user does not exist. However, access is accepted by FortiAuthenticator. The end result is the authentication is successful.

```
# get vpn ssl monitor
Agentless VPN Login Users:
Index   User       Group             Auth Type      Timeout         From       HTTP in/out
HTTPS in/out
0       fackeith  dualPrimaryGroup   2(1)            292        192.168.2.202      0/0
0/0
Agentless VPN sessions:
Index   User       Group                  Source IP      Duration        I/O Bytes
Tunnel/Dest IP
```

Case 2: Connect to the Agentless VPN web portal using a web browser and authenticate with user radkeith:

```
# diagnose sniffer packet any 'port 1812' 4 0 l
interfaces=[any]
filters=[port 1812]
2020-05-15 17:26:07.335791 port1 out 192.168.2.5.17988 -> 192.168.2.71.1812: udp 118
2020-05-15 17:26:07.335911 port3 out 192.168.20.5.17988 -> 192.168.20.6.1812: udp 118
2020-05-15 17:26:07.337659 port3 in 192.168.20.6.1812 -> 192.168.20.5.17988: udp 20 <--access-
accept
2020-05-15 17:26:07.337914 port3 out 192.168.20.5.17988 -> 192.168.20.6.1812: udp 182
2020-05-15 17:26:07.339451 port3 in 192.168.20.6.1812 -> 192.168.20.5.17988: udp 228
2020-05-15 17:26:08.352597 port1 in 192.168.2.71.1812 -> 192.168.2.5.17988: udp 20 <--access-
reject
```

There is a password mismatch for this user on the Secondary RADIUS server. However, even though the authentication was rejected by FortiAuthenticator, it was accepted by Windows NPS. Therefore, the end result is authentication successful.

```
# get vpn ssl monitor
Agentless VPN Login Users:
Index   User       Group            Auth Type      Timeout         From        HTTP in/out
HTTPS in/out
0       radkeith  dualPrimaryGroup   2(1)            290        192.168.2.202    0/0
0/0
Agentless VPN sessions:
Index   User      Group               Source IP      Duration        I/O Bytes
Tunnel/Dest IP
```

Agentless VPN with local user password policy

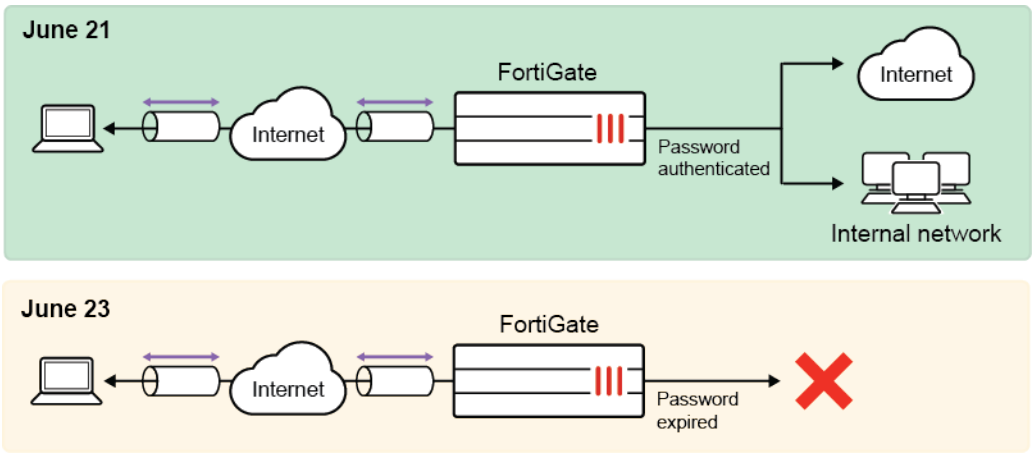
This is a sample configuration of Agentless VPN for users with passwords that expire after two days. Users are warned after one day about the password expiring. The password policy can be applied to any local user

password. The password policy cannot be applied to a user group or a local remote user such as LDAP/RADIUS/TACACS+.

In FortiOS 6.2, users are warned after one day about the password expiring and have one day to renew it. If the password expires, the user cannot renew the password and must contact the administrator for assistance.

In FortiOS 6.0/5.6, users are warned after one day about the password expiring and have to renew it. If the password expires, the user can still renew the password.

Sample topology



Sample configuration

WAN interface is the interface connected to ISP. This example shows static mode. You can also use DHCP or PPPoE mode. The Agentless VPN connection is established over the WAN interface.

To configure Agentless VPN using the GUI:

1. Configure the interface and firewall address. The port1 interface connects to the internal network.
   a. Go to *Network* > *Interfaces* and edit the *wan1* interface.
   b. Set *IP/Network Mask* to *172.20.120.123/255.255.255.0*.
   c. Edit *port1* interface and set *IP/Network Mask* to *192.168.1.99/255.255.255.0*.
   d. Click *OK*.
   e. Go to *Policy & Objects* > *Address* and create an address for internet subnet *192.168.1.0*.
2. Configure user and user group.
   a. Go to *User & Authentication* > *User Definition* to create a local user named agentlessvpnuser1.
   b. Go to *User & Authentication* > *User Groups* to create a user group named agentlessvpngroup, and add that local user to it.
3. Configure and assign the password policy using the CLI.
   a. Configure a password policy that includes an expiry date and warning time. The default start time for the password is the time the user was created.

```
config user password-policy
    edit "pwpolicy1"
        set expire-days 2
        set warn-days 1
    next
end
```

b. Assign the password policy to the user you just created.

```
config user local
    edit "agentlessvpnuser1"
        set type password
        set passwd-policy "pwpolicy1"
    next
end
```

4. Configure Agentless VPN portals.
   a. Go to *VPN > Agentless VPN Portals*, and click *Create New*.
   b. Enter a name, such as *agentless-portal*.
   c. Under *Predefined Bookmarks*, click *Create New*, and enter the following details:

Field	Value
Name	Windows Server
Type	RDP
Host	192.168.1.114
Port	3389
Single Sign-On	Disable
Username	Set the username to log in to Windows Server.
Password	Set the password to log in to Windows Server.
Port	3389

   d. Click *OK* to save the predefined bookmark.
   e. Click *OK* to save the portal settings.
   f. Create another Agentless VPN portal and name it *portal-access-disabled*.
   g. Click *OK* to save the portal settings.
   h. Disable Agentless VPN for the newly created portal (that is, *portal-access-disabled*) using CLI:

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
end
```

5. Configure Agentless VPN settings.
   a. Set Agentless VPN to *Enable*.
   b. Set the *Listen on Interface(s)* to *wan1*.

c. Set *Listen on Port* to *10443*.
d. Choose a certificate for *Server Certificate*. The default is *Fortinet_Factory*.
e. In *Authentication/Portal Mapping*, select *All Other Users/Groups*, and click *Edit*.
f. Use *Portal* dropdown to select *portal-access-disabled*.
g. Click *OK*.
h. In *Authentication/Portal Mapping*, click *Create New*.
   i. Set *Users/Groups* to *agentlessvpngroup*.
   ii. Set *Portal* to *agentless-portal*.
   iii. Click *OK*.
i. Click *Apply* to save the Agentless VPN settings.

6. Configure Agentless VPN firewall policy.
   a. Go to *Policy & Objects > Firewall Policy* and click *Create New*.
   b. Set *Name* as *Agentless VPN firewall policy*.
   c. Set *Schedule* to *always* and *Action* to *Accept*.
   d. Set *Incoming Interface* to *Agentless VPN tunnel interface(ssl.root)*.
   e. Choose an *Outgoing Interface*. This example uses *port1*.
   f. Set *Source* to *all* and *User/group* to *agentlessvpngroup*.
   g. In this example, the *Destination* is the internal, protected subnet *192.168.10.0*.
   h. Set *Service* to *ALL*.
   i. Click *OK*.

To configure Agentless VPN using the CLI:

1. Configure the interface and firewall address.

```
config system interface
    edit "wan1"
        set vdom "root"
        set ip 172.20.120.123 255.255.255.0
    next
end
```

2. Configure internal interface and protected subnet, then connect the port1 interface to the internal network.

```
config system interface
    edit "port1"
        set vdom "root"
        set ip 192.168.1.99 255.255.255.0
    next
end
```

```
config firewall address
    edit "192.168.1.0"
        set subnet 192.168.1.0 255.255.255.0
    next
end
```

3. Configure user and user group.

```
config user local
    edit "agentlessvpnuser1"
        set type password
        set passwd your-password
    next
end
config user group
    edit "agentlessvpngroup"
        set member "agentlessvpnuser1"
    next
end
```

4. Configure and assign the password policy.
   a. Configure a password policy that includes an expiry date and warning time. The default start time for the password is the time the user was created.

```
config user password-policy
    edit "pwpolicy1"
        set expire-days 2
        set warn-days 1
    next
end
```

   b. Assign the password policy to the user you just created.

```
config user local
    edit "sslvpnuser1"
        set type password
        set passwd-policy "pwpolicy1"
    next
end
```

5. Configure Agentless VPN portal and predefine RDP bookmark for windows server.

```
config vpn ssl web portal
    edit "agentless-portal"
        set web-mode enable
        config bookmark-group
            edit "gui-bookmarks"
                config bookmarks
                    edit "Windows Server"
                        set apptype rdp
                        set host "192.168.1.114"
                        set port 3389
                        set logon-user "your-windows-server-user-name"
                        set logon-password your-windows-server-password
                    next
                end
            next
        end
    next
end
```

6. Configure another Agentless VPN portal and disable Agentless VPN on it.

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
end
```

7. Apply the user group to the Agentless VPN portal.

```
config vpn ssl settings
    set servercert "Fortinet_Factory"
    set source-interface "wan1"
    set source-address "all"
    set default-portal "portal-access-disabled"
    config authentication-rule
        edit 1
            set groups "agentlessvpngroup"
            set portal "agentless-portal"
        next
    end
end
```

8. Configure one Agentless VPN firewall policy to allow remote user to access the internal network. This policy does not allow traffic initiated from internal network to remote client.

```
config firewall policy
    edit 1
        set name "Agentless VPN firewall policy"
        set srcintf "ssl.root"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "192.168.1.0"
        set schedule "always"
        set service "ALL"
        set nat enable
        set groups "LDAP-USERGRP"
    next
end
```

To see the results of web portal:

1. From a remote device, use a web browser to log into the Agentless VPN web portal *http://172.20.120.123:10443*.
2. Log in using the *agentlessvpnuser1* credentials.

   When the warning time is reached, the user is prompted to enter a new password.

   In FortiOS 6.2, when the password expires, the user cannot renew the password and must contact the administrator.

   In FortiOS 6.0/5.6, when the password expires, the user can still renew the password.

To check the Agentless VPN connection using the GUI:

1. On FortiGate, go to *Dashboard > Agentless VPN Monitor* to verify the list of Agentless VPN users.
   If the Agentless VPN Monitor is hidden, click *+* under the *Dashboard*, search for *Agentless VPN Monitor*, and add it to the display.
2. Go to *Log & Report > System Events*, and use the dropdown to select *VPN Events* to view VPN logs.

To check that login failed due to password expired on GUI:

1. Go to *Log & Report > System Events* and select the *VPN Events* card to see the Agentless VPN alert labeled `ssl-login-fail`.
2. Click *Details* to see the log details about the *Reason*`sslvpn_login_password_expired`.

To check the Agentless web portal login using the CLI:

```
get vpn ssl monitor
Agentless VPN Login Users:
 Index   User         Auth Type   Timeout    From           HTTP in/out   HTTPS in/out
 0       agentlessvpnuser1  1(1)        229        10.1.100.254   0/0           0/0

Agentless VPN sessions:
 Index   User    Source IP      Duration        I/O Bytes       Tunnel/Dest IP
```

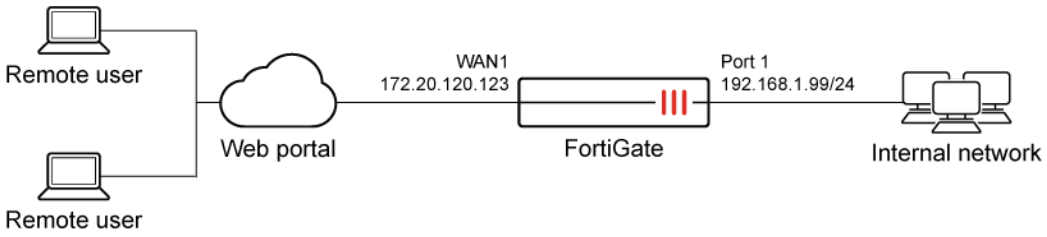
To check the FortiOS login password expired event log:

```
FG201E4Q17901354 # execute log  filter category event

FG201E4Q17901354 # execute log  filter field subtype vpn

FG201E4Q17901354 # execute log  filter field action ssl-login-fail

FG201E4Q17901354 # execute log  display
1: date=2019-02-15 time=10:57:56 logid="0101039426" type="event" subtype="vpn" level="alert"
vd="root" eventtime=1550257076 logdesc="SSL VPN login fail" action="ssl-login-fail"
tunneltype="ssl-web" tunnelid=0 remip=10.1.100.254 user="u1" group="g1" dst_host="N/A"
reason="sslvpn_login_password_expired" msg="SSL user failed to logged in"
```

Agentless VPN multi-realm

This sample shows how to create a multi-realm Agentless VPN that provides different portals for different user groups.

Sample topology



Sample configuration

WAN interface is the interface connected to ISP. This example shows static mode. You can also use DHCP or PPPoE mode. The Agentless VPN connection is established over the WAN interface.

To configure Agentless VPN using the GUI:

1. Configure the interface and firewall address. The port1 interface connects to the internal network.
   a. Go to *Network* > *Interfaces* and edit the *wan1* interface.
   b. Set *IP/Network Mask* to *172.20.120.123/255.255.255.0*.
   c. Edit *port1* interface and set *IP/Network Mask* to *192.168.1.99/255.255.255.0*.
   d. Click *OK*.
   e. Go to *Policy & Objects* > *Address* and create an address for internet *QA_subnet* with subnet *192.168.1.0/24* and *HR_subnet* with subnet *10.1.100.0/24*.
2. Configure user and user group.
   a. Go to *User & Authentication* > *User Definition* to create local users *qa-user1* and *hr-user1*.
   b. Go to *User & Authentication* > *User Groups* to create separate user groups for web-only and full-access portals:
      - *QA_group* with member *qa-user1*.
      - *HR_group* with the member *hr-user1*.
3. Configure Agentless VPN portal.
   a. Go to *VPN* > *Agentless VPN Portals* to create portal *qa-tunnel*, and complete other configurations, such as Bookmarks, as required.
   b. Create another portal named *hr-web* with its own bookmark configuration.
   c. Create another portal named *portal-access-disabled*, and disable Agentless VPN on it using CLI:

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
end
```

4. Configure Agentless VPN realms.
   a. Go to *System* > *Feature Visibility* to enable *Agentless VPN Realms*.
   b. Go to *VPN* > *Agentless VPN Realms* to create realms for *qa* and *hr*.

c. (Optional) To access each realm with FQDN instead of the default URLs *https://172.20.120.123:10443/hr* and *https://172.20.120.123:10443/qa*, you can configure a virtual-host for the realm in the CLI.

```
config vpn ssl web realm
    edit hr
        set virtual-host hr.mydomain.com
    next
    edit qa
        set virtual-host qa.mydomain.com
    next
end
```

Where `mydomain.com` is the name of your domain. Ensure FQDN resolves to the FortiGate wan1 interface and that your certificate is a wildcard certificate.

5. Configure Agentless VPN settings.
   a. Go to *VPN > Agentless VPN Settings*.
   b. Enable *Agentless VPN* status.
   c. For *Listen on Interface(s)*, select *wan1*.
   d. Set *Listen on Port* to *10443*.
   e. Choose a certificate for *Server Certificate*. The default is *Fortinet_Factory*.
   f. In *Authentication/Portal Mapping All Other Users/Groups*, set the *Portal* to *portal-access-disabled*.
   g. Create new *Authentication/Portal Mapping* for group *QA_group* mapping portal *qa-tunnel*.
   h. Specify the realm *qa*.
   i. Add another entry for group *HR_group* mapping portal *hr-web*.
   j. Specify the realm *hr*.
6. Configure Agentless VPN firewall policy.
   a. Go to *Policy & Objects > Firewall Policy*.
   b. Create a firewall policy for QA access.
   c. Fill in the firewall policy name. In this example, *QA agentlessvpn access*.
   d. Incoming interface must be *Agentless VPN tunnel interface (ssl.root)*.
   e. Choose an *Outgoing Interface*. In this example, *port1*.
   f. Set the *Source* to *all* and *User/group* to *QA_group*.
   g. In this example, the *Destination* is the internal protected subnet *QA_subnet*.
   h. Set *Schedule* to *always*, *Service* to *ALL*, and *Action* to *Accept*.
   i. Click *OK*.
   j. Create a firewall policy for HR access.
   k. Fill in the firewall policy name. In this example, *HR agentlessvpn access*.
   l. Incoming interface must be *Agentless VPN tunnel interface (ssl.root)*.
   m. Choose an *Outgoing Interface*. In this example, *port1*.
   n. Set the *Source* to *all* and *User/group* to *HR_group*.
   o. In this example, the *Destination* is the internal protected subnet *HR_subnet*.
   p. Set *Schedule* to *always*, *Service* to *ALL*, and *Action* to *Accept*.
   q. Click *OK*.

To configure Agentless VPN using the CLI:

1. Configure the interface and firewall address.

```
config system interface
    edit "wan1"
        set vdom "root"
        set ip 172.20.120.123 255.255.255.0
    next
end
```

2. Configure internal interface and protected subnet, then connect the port1 interface to the internal network.

```
config system interface
    edit "port1"
        set vdom "root"
        set ip 192.168.1.99 255.255.255.0
    next
end
```

```
config firewall address
    edit "QA_subnet"
        set subnet 192.168.1.0 255.255.255.0
    next
    edit "HR_subnet"
        set subnet 10.1.100.0 255.255.255.0
    next
end
```

3. Configure user and user group.

```
config user local
    edit "qa_user1"
        set type password
        set passwd your-password
    next
end
config user group
    edit "QA_group"
        set member "qa_user1"
    next
end
```

```
config user local
    edit "hr_user1"
        set type password
        set passwd your-password
    next
end
config user group
    edit "HR_group"
        set member "hr_user1"
```

```
    next
end
```

4. Configure Agentless VPN web portal each for HR and QA. Create another portal with Agentless VPN disabled.

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
    edit "qa-portal"
        set web-mode enable
        config bookmark-group
            edit "gui-bookmarks"
                        config bookmarks
                        ...
            next
        end
    next
    edit "hr-portal"
        set web-mode enable
        config bookmark-group
            edit "gui-bookmarks"
                        config bookmarks
                        ...
            next
        end
    next
end
```

5. Configure Agentless VPN realms.

```
config vpn ssl web realm
    edit hr
        set virtual-host hr.mydomain.com
    next
    edit qa
        set virtual-host qa.mydomain.com
    next
end
```

The set virtual-host setting is optional. For example:

```
config vpn ssl web realm
    edit hr
    next
    edit qa
    next
end
```

6. Configure Agentless VPN settings.

```
config vpn ssl settings
    set banned-cipher SHA1 SHA256 SHA384
    set servercert "Fortinet_Factory"
    set login-attempt-limit 0
    set port 10443
    set source-interface "port1"
    set source-address "all"
    set default-portal "portal-access-disabled"
    config authentication-rule
        edit 1
            set groups "HR_group"
            set portal "hr-portal"
            set realm "hr"
        next
        edit 2
            set groups "QA_group"
            set portal "qa-portal"
            set realm "qa"
        next
    end
end
```

7. Configure two Agentless VPN firewall policies to allow remote QA user to access internal QA bookmark network and HR user to access HR bookmark network.

```
config firewall policy
    edit 1
        set name "QA sslvnpn access"
        set srcintf "ssl.root"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "QA_subnet"
        set action accept
        set schedule "always"
        set service "ALL"
        set groups “QA_group”
    next
    edit 2
        set name "HR sslvpn access"
        set srcintf "ssl.root"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "HR_subnet"
        set action accept
        set schedule "always"
        set service "ALL"
        set groups “HR_group”
    next
end
```

To see the results for HR and QA users:

1. In a web browser, log in to the portal https://172.20.120.123:10443/hr and https://172.20.120.123:10443/qa using the credentials you've set up.
2. Alternatively, if a virtual-host is specified, use the FQDN defined for the realm (hr.mydomain.com or qa.mydomain.com).

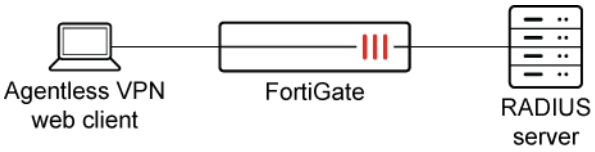
To check the Agentless VPN connection using the GUI:

1. On FortiGate, go to *Dashboard* > *Agentless VPN Monitor* to verify the list of Agentless VPN users.
   If the Agentless VPN Monitor is hidden, click *+* under the *Dashboard*, search for *Agentless VPN Monitor*, and add it to the display.
2. Go to *Log & Report* > *System Events*, and use the dropdown to select *VPN Events* to view VPN logs.

NAS-IP support per Agentless VPN realm

For RADIUS authentication and authorization, the RADIUS client (the FortiGate) passes the username, password, and NAS-IP to the RADIUS server in its access request. The RADIUS server authenticates and authorizes based on this information. Each RADIUS server can be configured with multiple NAS-IPs for authenticating different groups and NAS clients.

On the FortiGate, configuring the NAS-IP in the realm settings overrides the RADIUS server setting, allowing multiple NAS-IPs to be mapped to the same RADIUS server.



In this example, the user wants to present one FortiGate VDOM with different NAS-IPs to a single RADIUS server based on specific rules.

To configure Agentless VPN to use NAS-IP in the realm settings:

1. Configure a RADIUS user and add it to a group:

```
config user radius
    edit "fac150"
        set server "172.16.200.150"
        set secret ********
        set nas-ip 172.16.200.2
        config accounting-server
            edit 1
                set status enable
                set server "172.16.200.150"
                set secret ********
            next
        end
    next
```

```
end
config user group
    edit "radgrp"
        set member "fac150"
    next
end
```

2. Configure a realm for the user with a different NAS-IP:

```
config vpn ssl web realm
    edit "realm1"
        set login-page '.......'
        set radius-server "fac150"
        set nas-ip 10.1.100.2
    next
end
```

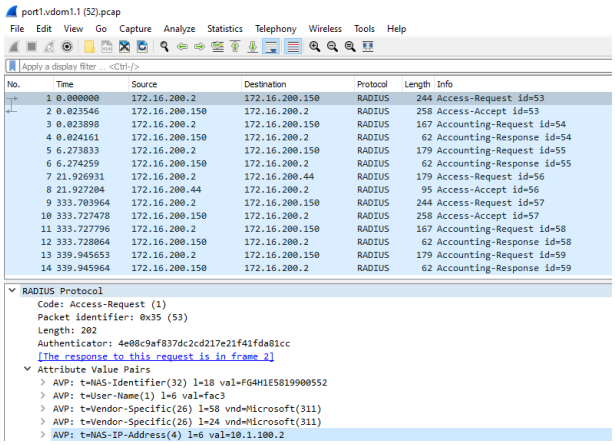
3. Configure Agentless VPN with an authentication rule that includes the user group and the realm:

```
config vpn ssl settings
    ...
    config authentication-rule
        edit 1
            set groupd "radgrp"
            set portal "testportal1"
            set realm "realm1"
        next
    end
end
```

4. Create a firewall policy:

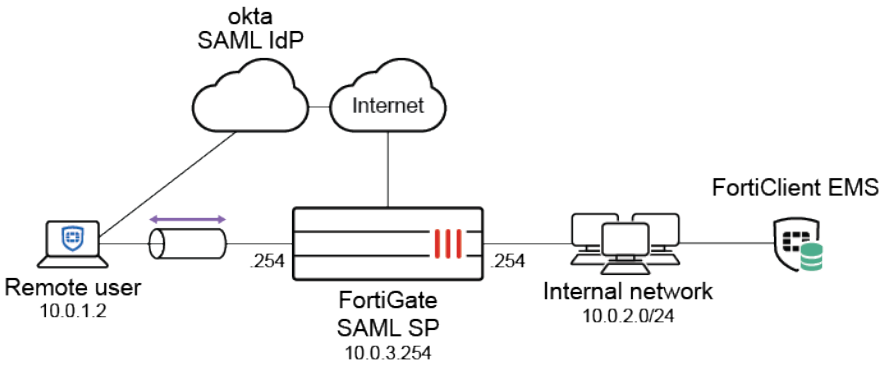
```
config firewall policy
    edit 1
        set name "sslvpn1"
        ...
        set srcintf "ssl.vdom1"
        set groups "radgrp"
    next
end
```

Because the RADIUS server and NAS-IP are specified in realm1, its NAS-IP is used for authentication.



Agentless VPN with Okta as SAML IdP

In this configuration, the FortiGate acts as a SAML service provider (SP) requesting authentication from Okta, which acts as a SAML identity provider (IdP). The following shows the topology in this configuration:



The authentication process is as follows in this deployment:

1. The user initiates an Agentless VPN request to the FortiGate using its web browser.
2. The FortiGate sends the browser POST redirect to web browser.
3. User's web browser redirects the SAML authentication request to Okta.
4. The user authenticates with Okta using their credentials.
5. Okta sends a SAML assertion that contains the user and group authentication in a POST redirect to the Agentless VPN login page.
6. User's browser sends the redirected Okta request that contains the SAML assertion to the FortiGate.
7. The FortiGate consumes the assertion and provides the user with access to resources based on the defined firewall security policy.

The example assumes that you already have an Okta account. This example uses users locally defined within the Okta directory and does not include LDAP mapping. The instructions describe the steps that you take if using the free Okta developer edition.

The following certificates are used:

Certificate	Description

SP certificate	This is the certificate used to sign the SAML messages originating from the SP to the IdP. This is configured on the FortiGate by the following CLI setting:
	
	`config user saml`
	`edit <profile>`
	`set cert <local certificate>`
	`next`
	`end`
	
	The certificate will be used by the SAML IdP (Okta) to verify the SP connection, so the certificate must be uploaded into Okta.

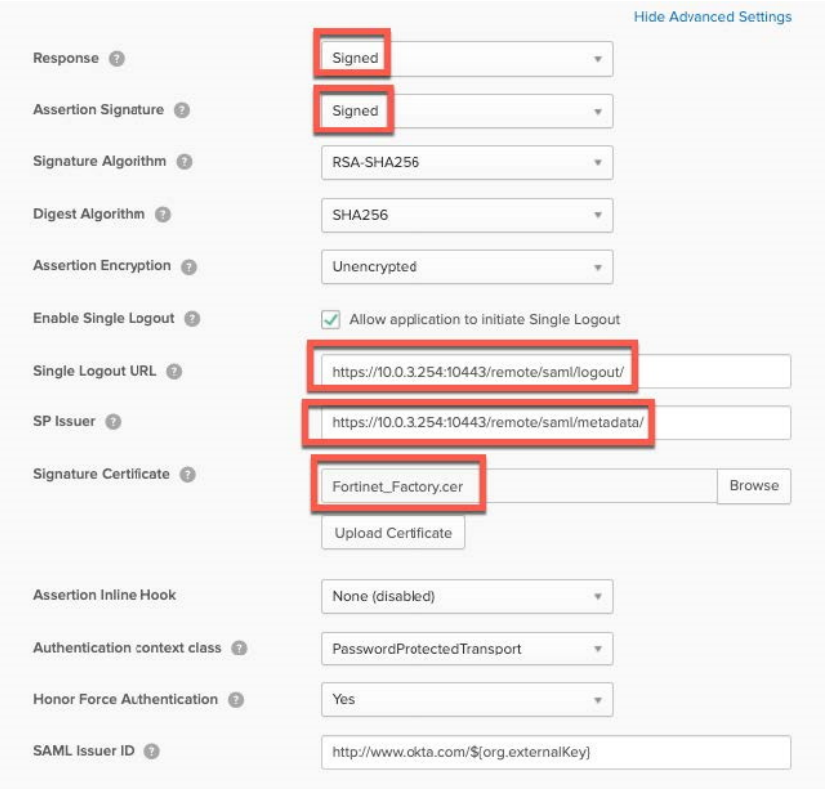
IdP certificate	This is the certificate used to sign the SAML response originating from the IdP. This must be trusted by the SP in order to verify the identity of the messages from the IdP.
	In this example, the certificate is provided by Okta. To upload this remote certificate into the FortiGate, follow the instructions in Remote certificate on page 3968.
	To use the certificate in your SAML SSO settings:
	
	`config user saml`
	`edit <profile>`
	`set idp-cert <Okta certificate>`
	`next`
	`end`

Agentless VPN server certificate	This certificate identifies the Agentless VPN portal when a remote client connects to the FortiGate using its browser.
	This is configured in the CLI as follows:
	
	`config vpn ssl settings`
	`set servercert <server certificate>`
	`end`
	
	When this is not specified, then the Fortinet factory self-signed certificate is used.

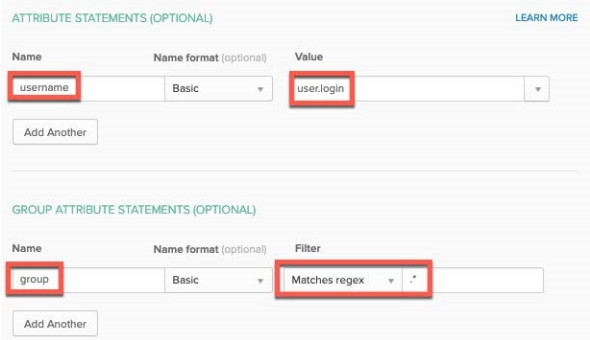
To configure Okta for Agentless VPN with FortiOS:

1. Log in to the Okta portal as the registered admin user.
2. Add the FortiGate application:
   a. Go to *Applications*.
   b. Click *Applications*, then click *Create App Integration*.
   c. Click *SAML 2.0*, then *Next*.
   d. Configure SAML settings:

i. Proceed through the application creation wizard. In the *Single sign on URL* field, enter https://<FortiGate IP address>:<port>/remote/saml/login/. In this example, it is https://10.0.3.254:10443/remote/saml/login/.
ii. Enable *Use this for Recipient URL and Destination URL*.
iii. In the *Audience URI (SP Entity ID)* field, enter the https://<FortiGate IP address>:<port>/remote/saml/metadata/. In this example, it is https://10.0.3.254:10443/remote/saml/metadata/.
iv. Click *Download Okta Certificate* to download the Okta certificate to your machine. You will provide this certificate to the FortiGate.
v. Click *Show Advanced Settings*. From the *Response* dropdown list, select *Signed*.
vi. From the *Assertion Signature* dropdown list, select *Signed*.
vii. In the *Single Logout URL* field, enter https://<FortiGate IP address>:<port>/remote/saml/logout/. In this example, it is https://10.0.3.254:10443/remote/saml/logout/.
viii. In the *SP Issuer* field, enter https://<FortiGate IP address>:<port>/remote/saml/metadata/. In this example, it is https://10.0.3.254:10443/remote/saml/metadata/.
ix. In the *Signature Certificate* field, first download the Fortinet_Factory certificate by logging into FortiOS, going to *System > Local Certificate*, then browsing to and uploading the FortiGate certificate. Okta uses this to authenticate the SAML SP.

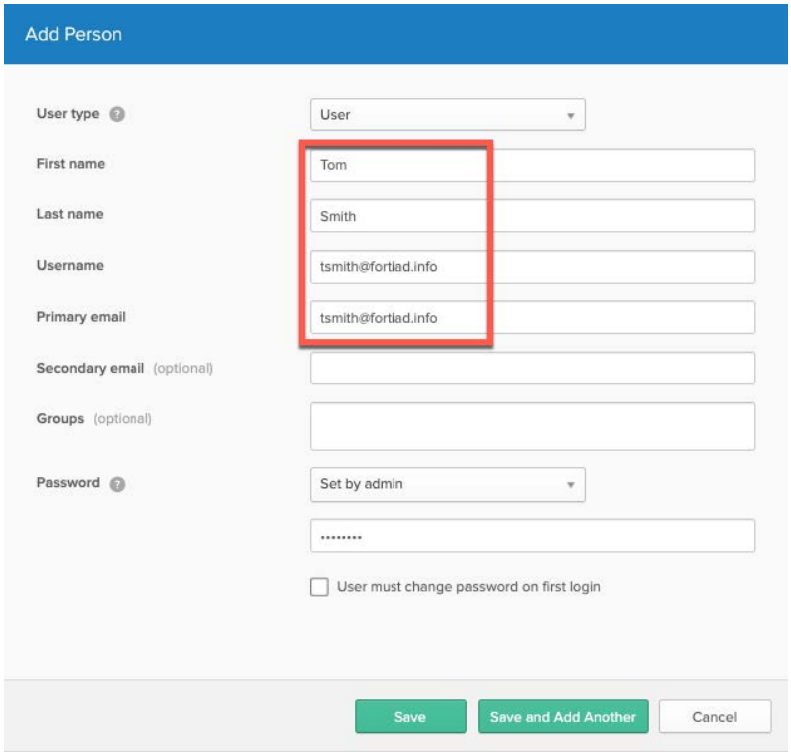


e. Under *ATTRIBUTE STATEMENTS* and *GROUP ATTRIBUTE STATEMENTS*, define attribute mappings for Okta to use in SAML assertion. In this example, the following is entered as a attribute statement and a group attribute statement, respectively:
   - username, with value user.login
   - group, with *Matches regex* filter



f. On the *Feedback* step, select *I'm an Okta customer adding an internal app*.
g. Select *This is an internal app that we have created*.
h. Click *Finish*.

3. Go to *Directory* > *People*.
4. Click *Add Person*.
5. Enter the person's details as desired. Click *Save*.



6. Add a group:
   a. Go to *Directory* > *Groups*.
   b. Click *Add Group*.
   c. Enter the desired name, then click *Add Group*. In this example, the name is corporate-saml.
   d. Select the newly added group, then click *Assign People*.
   e. Add the person that you created as a member of the new group. Click *Save*.
7. Assign the group to the FortiGate application:

   a. Go to *Applications* > *FortiGate application* > *Assignments*.
   b. From the *Assign* dropdown list, select *Assign to Groups*.
   c. Assign the group that you created to the FortiGate application.
8. To view the SAML setup instructions, do the following:
   a. Click the newly created application's name.
   b. Click *Sign On*.
   c. Go to *View SAML Setup Instructions*. Note down the *Identity Provider Single Sign-On URL*, *Identity Provider Single Logout URL*, and *Identity Provider Issuer* values.
9. Download the Okta certificate and upload it to FortiOS:
   a. From *View SAML Setup Instructions*, download the certificate.
   b. In FortiOS, go to *System* > *Certificates*.
   c. From the *Create/Import* dropdown list, select *Remote Certificate*.
   d. Click *Upload* and upload the downloaded Okta certificate.

To configure the FortiGate:

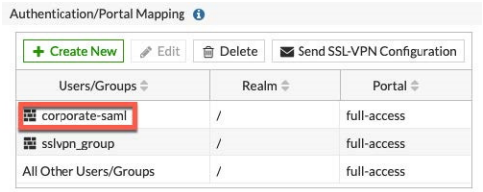
1. Configure the FortiGate SP to be a SAML user:

```
config user saml
    edit "okta-idp"
        set cert "Fortinet_Factory"
        set entity-id "https://10.0.3.254:10443/remote/saml/metadata/"
        set single-sign-on-url "https://10.0.3.254:10443/remote/saml/login"
        set single-logout-url "https://10.0.3.254:10443/remote/saml/logout"
        set idp-entity-id "http://www.okta.com/exk103foxaa8gk5qy4x7"
        set idp-single-sign-on-url "https://fortinet01.okta.com/app/fortinetorg878484_
fortigate_1/exk103foxaa8gk5qy4x7/sso/saml"
        set idp-single-logout-url "https://fortinet01.okta.com/app/fortinetorg878484_
fortigate_1/exk103foxaa8gk5qy4x7/slo/saml"
        set idp-cert "Okta-IDP_Certificate"
        set user-name "username"
        set group-name "group"
    next
end
```

2. Configure user group assertion on Okta as part of the SAML assertion attributes. It is important that the group attribute value received is locally matched with the `group-name` value:

```
config user group
    edit "corporate-saml"
        set member "okta-idp"
        config match
            edit 1
                set server-name "okta-idp"
                set group-name "corporate-saml"
            next
        end
    next
end
```

3. Go to *VPN* > *Agentless VPN Settings*. Configure VPN settings as desired.

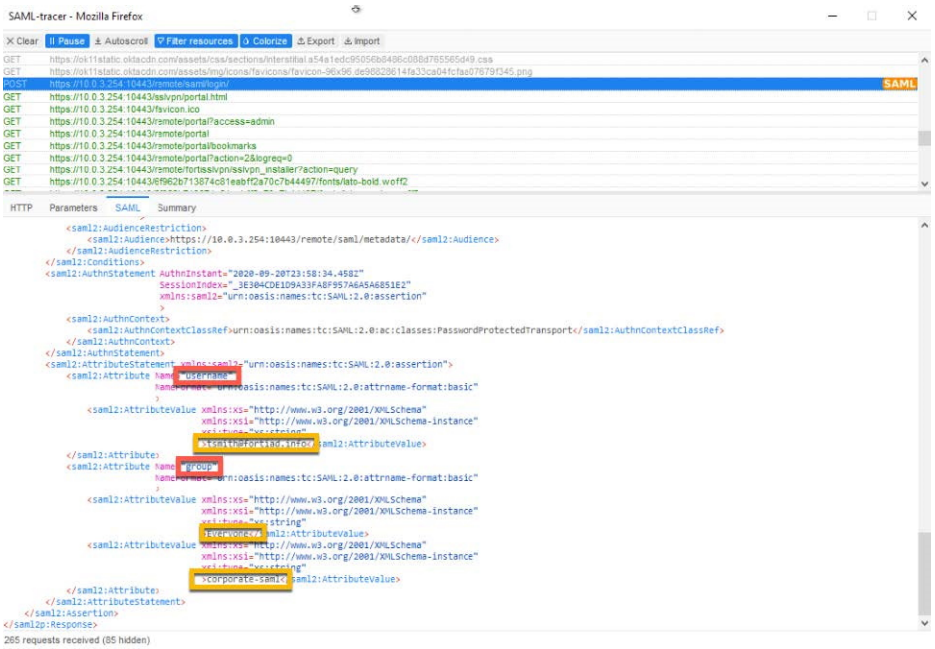


4. Configure a local or RADIUS user as a backup. This setting also provides a login web user with a choice of local or SSO login.
5. Go to *Policy and Objects* > *Firewall Policies*. Configure a policy as desired.
6. Increase the global authentication timeout period to allow users to fill in their credentials in time. The default timeout is five seconds:

```
config system global
    set remoteauthtimeout 60
end
```

To test the configuration:

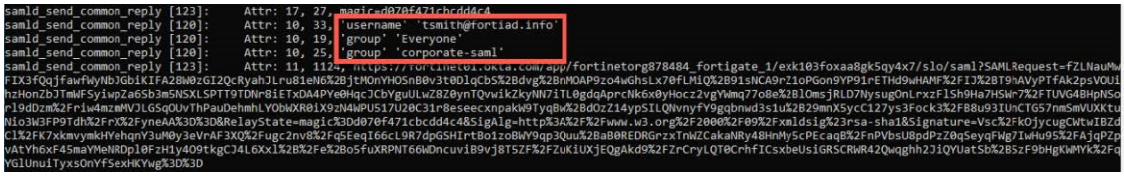
1. Install the SAML-tracer plugin to Firefox browser.
2. In Firefox, go to the FortiOS Agentless VPN login page. In this example, this is https://10.0.3.254:10443.
3. Open the SAML tracer.
4. The browser redirects to the Okta SAML login page. Enter the Okta credentials, then click *Sign in*.
5. Upon successful authentication, the browser redirects to the authenticated Agentless VPN page. If authentication does not succeed, review the SAML tracer to confirm the SAML assertion attributes that are passed during the authentication session. Select the POST message with the SAML information. On the SAML tab, confirm the username and group attributes.



To troubleshoot the configuration:

You can view FortiOS event logs in *Log & Report* > *Events* to verify successful authentication and user group allocation.

You can also run the `diagnose debug application samld -1` command to verify that the SAML IdP sent the correct information. The following shows example output for this scenario:



Agentless VPN with Microsoft Entra SSO integration

You can use SAML single sign-on to authenticate Agentless VPN users against Microsoft Entra ID, using similar SSL VPN web mode configuration. See:

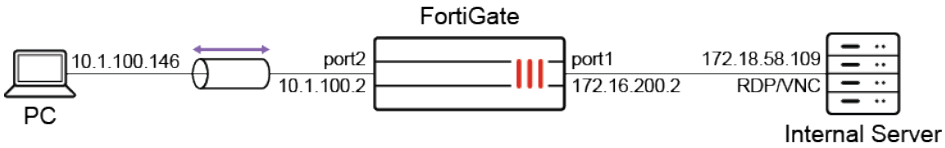
- Configuring SAML SSO login for SSL VPN with Entra ID acting as SAML IdP
- Tutorial: Microsoft Entra SSO integration with FortiGate SSL VPN

Disable the clipboard for Agentless VPN RDP connections

In Agentless VPN portals, the clipboard can be disabled for RDP/VNC connections, and the user will be unable to copy and paste content to or from the internal server.

Example

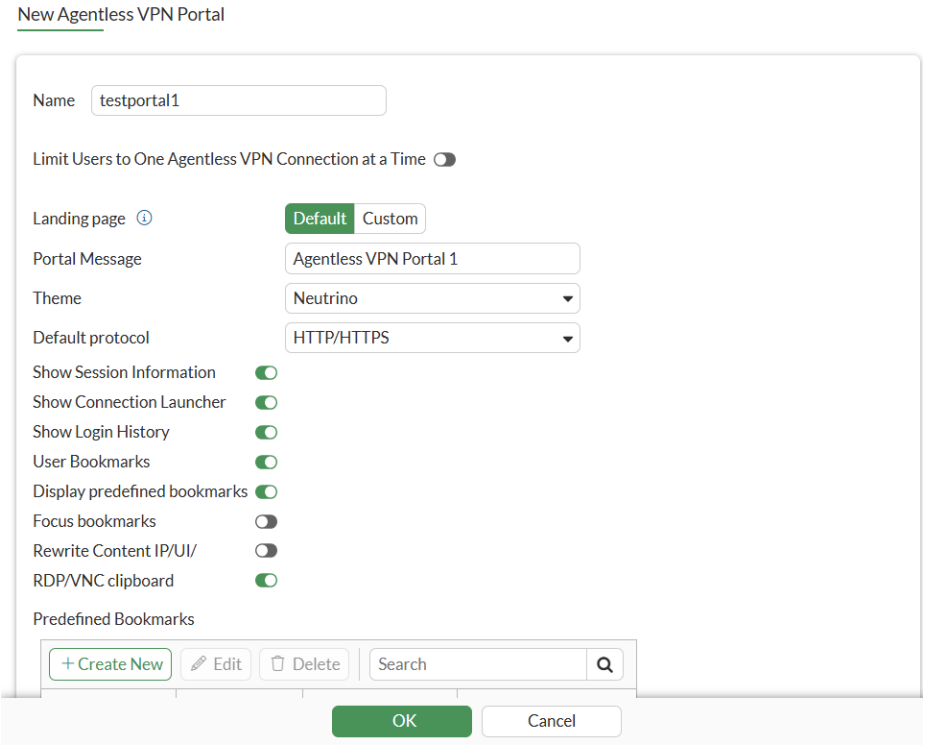
In this example, two groups of users are using Agentless VPN to access internal servers with RDP/VNC. One group is allowed to copy and paste content to and from the internal server using the clipboard, while the other is not.



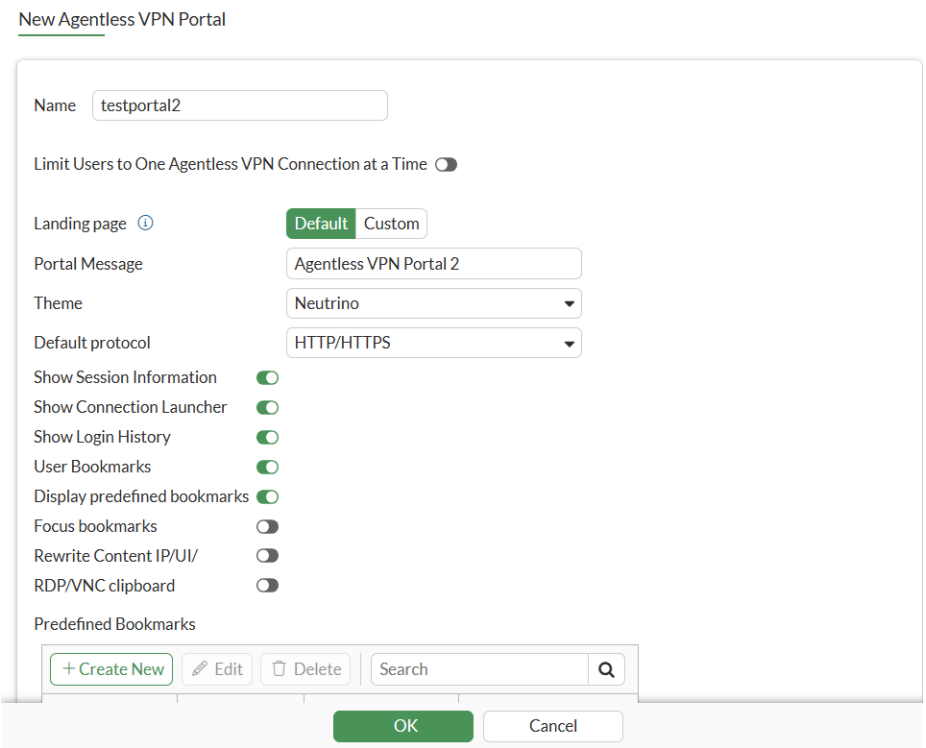
To configure Agentless VPN portals in the GUI:

1. Go to *VPN* > *Agentless VPN Portals* and click *Create New*.
2. Enter a name for the portal, such as *testportal1*.

3. Enable *RDP/VNC clipboard* to allow copying and pasting.
4. Configure the remaining settings as needed.



5. Click *OK*.
6. Click *Create New* again.
7. Enter a name for the portal, such as *testportal2*.
8. Disable *RDP/VNC clipboard* to prevent copying and pasting.
9. Configure the remaining settings as needed.



10. Click *OK*.
11. Click *Create New* again, and enter *Name* as *portal-access-disabled*.
12. Disable Agentless VPN on the portal using CLI.

```
config vpn ssl web portal
    edit "portal-access-disabled"
        set web-mode disable
    next
end
```

To configure the Agentless VPN settings in the GUI:

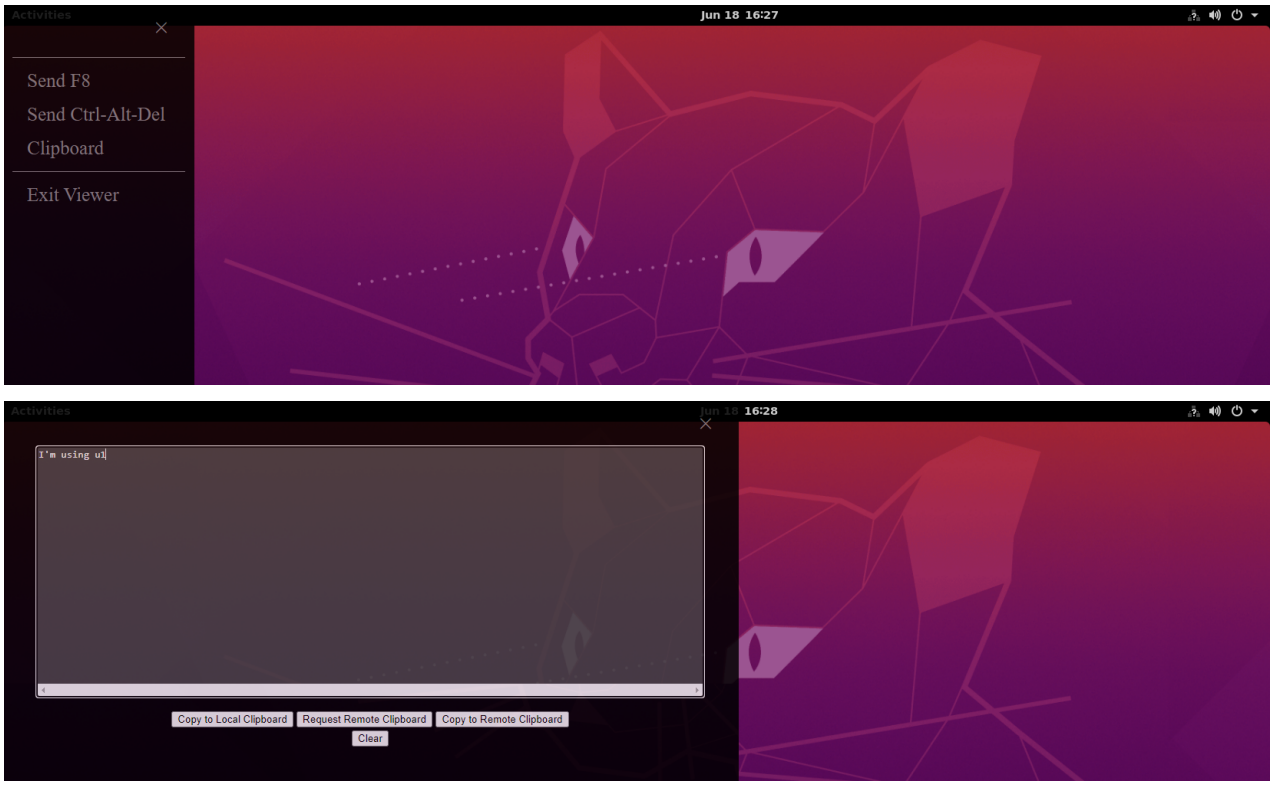
1. Go to *VPN* > *Agentless VPN Settings*.
2. Set Agentless VPN status to *Enable*.
3. Set *Listen on Interface* to port2.
4. In the *Authentication/Portal Mapping* table, add the users to each of the portals:
   a. Click *Create New*.
   b. Set *Users/Groups* to *u1* and *Portal* to *testportal1*.
   c. Click *OK*, then click *Create New* again.
   d. Set *Users/Groups* to *u2* and *Portal* to *testportal2*.
   e. Click *OK*.
   f. Set all *Other User/Groups* to *portal-access-disabled*.
5. Configure the remaining settings as needed.
6. Click *Apply*.

To configure a firewall policy for Agentless VPN in the GUI:

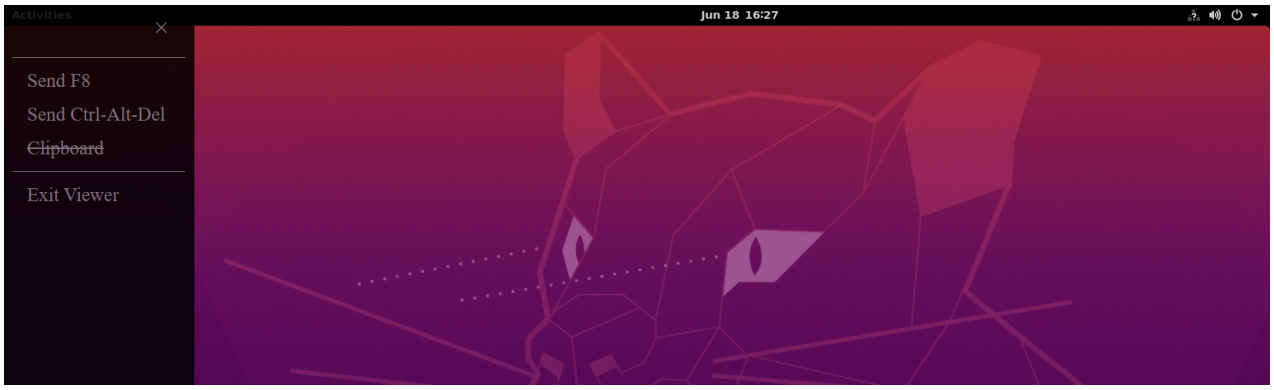
1. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
2. Set a name for the policy, such as *policy_to_agentlessvpn*.
3. Set *Incoming Interface* to the *Agentless VPN tunnel interface (ss.root)* and *Outgoing Interface* to port1.
4. Set *Source* to *all* and *Users/groups* to *u1* and *u2*.
5. Set *Destination* to all addresses.
6. Set *Schedule* to *always*, *Service* to *All*, and *Action* to *Accept*.
7. Configure the remaining settings as needed.
8. Click *OK*.

To test the if the users can use the clipboard:

1. On the PC, open a web browser and log in to the web portal as user *u1*.
2. Access the internal server using RDP/VNC.
3. The clipboard is available and you can copy and paste content to and from the remote server.



4. Log out of the web portal, then log back in as user *u2* and access the internal server using RDP/VNC. The clipboard is disabled.



To configure the Agentless VPN portals and settings in the CLI:

1. Configure the Agentless VPN portals:

```
config vpn ssl web portal
    edit "testportal1"
        set web-mode enable
        set clipboard enable
        ...
    next
    edit "testportal2"
        set web-mode enable
        set clipboard disable
        ...
    next
end
```

2. Configure the Agentless VPN settings:

```
config vpn ssl settings
    set banned-cipher SHA1 SHA256 SHA384
    set servercert "Server_Certificate "
    set login-attempt-limit 0
    set port 1443
    set source-interface "port2"
    set source-address "all"
    set source-address6 "all"
    set default-portal "portal-access-disabled"
    config authentication-rule
        edit 1
            set users "u1"
            set portal "testportal1"
        next
        edit 2
            set users "u2"
            set portal "testportal2"
        next
    end
end
```

3. Configure a firewall policy for Agentless VPN:

```
config firewall policy
    edit 1
        set name "policy_to_Agentless_VPN"
        set srcintf "ssl.root"
        set dstintf "port1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set nat enable
        set users "u1" "u2"
    next
end
```

4. On the PC, open a web browser, log in to the web portal as user *u1*, access the internal server using RDP/VNC, and use the clipboard.
5. Check the Agentless VPN session monitor using FortiOS CLI:

```
# get vpn ssl monitor
Agentless VPN Login Users:
 Index   User    Group   Auth Type      Timeout         Auth-Timeout    From     HTTP in/out
 HTTPS in/out    Two-factor Auth
 0       u1             1(1)             N/A     10.1.100.146   0/0     0/364   0

Agentless VPN sessions:
 Index   User    Group   Source IP      Duration        I/O Bytes       Tunnel/Dest IP
```

6. On the PC, open a web browser, log in to the web portal as user *u2*, access the internal server using RDP/VNC, and note that the clipboard is not available.
7. Check the Agentless VPN session monitor:

```
# get vpn ssl monitor
Agentless VPN Login Users:
 Index   User    Group   Auth Type      Timeout         Auth-Timeout    From     HTTP in/out
 HTTPS in/out    Two-factor Auth
 0       u2             1(1)             N/A     10.1.100.146   0/0     0/2681  0

Agentless VPN sessions:
 Index   User    Group   Source IP      Duration        I/O Bytes       Tunnel/Dest IP
```

Post-Quantum Cryptography for Agentless VPN

FortiOS supports Post-Quantum Cryptography (PQC) for Agentless VPN. Pure and hybrid PQC algorithms to prepare for future quantum computing threats can be selected in the CLI. Using OpenSSL library version 3.5 enables initial PQC algorithm support and establishes a foundation for broader quantum-resistant capabilities across FortiGate platforms.

In FIPS mode, Known Answer Tests (KATs) are available for PQC algorithms, including OSSL-ML-DSA-KAT_Signature, OSSL-SLH-DSA-KAT_Signature, and OSSL-ML-KEM, along with key generation and encapsulation tests. Legacy RSA encrypt/decrypt tests are not available to align with modern cryptographic standards.

Additionally, SSL VPN supports TLS group configuration in the CLI:

```
config vpn ssl settings
    set tls-groups <groups>
end
```

Supported groups include traditional elliptic curves (P-256, P-384, P-521), finite field Diffie-Hellman (FFDHE2048–8192), pure PQC (ML-KEM512, ML-KEM768, ML-KEM1024), and hybrid combinations (such as P-384-MLKEM1024, X25519-MLKEM768). PQC groups are currently unavailable in FIPS-CC mode pending FIPS 140-3 approval.

Use Case 1: FIPS-CC compliance customer

Strict FIPS-CC compliance for all cryptographic operations is required in a FortiGate deployment supporting employees using Agentless VPN with only FIPS-compliant algorithms.

1. The FortiGate in FIPS-CC mode can boot up successfully with new KATs added:

```
[smit] Installing and loading FIPS provider:
HMAC               : KAT_Integrity               : Pass
HMAC               : Module_Integrity            : Pass
SHA2               : KAT_Digest                  : Pass
SHA3               : KAT_Digest                  : Pass
AES_GCM            : KAT_Cipher                  : Pass
AES_ECB_Decrypt    : KAT_Cipher                  : Pass
TDES               : KAT_Cipher                  : Pass
RSA                : KAT_Signature               : Pass
ECDSA              : KAT_Signature               : Pass
ECDSA              : KAT_Signature               : Pass
EDDSA              : KAT_Signature               : Pass
EDDSA              : KAT_Signature               : Pass
DSA                : KAT_Signature               : Pass
ML-DSA             : KAT_Signature               : Pass
SLH-DSA            : KAT_Signature               : Pass
SLH-DSA            : KAT_Signature               : Pass
TLS13_KDF_EXTRACT  : KAT_KDF                     : Pass
TLS13_KDF_EXPAND   : KAT_KDF                     : Pass
TLS12_PRF          : KAT_KDF                     : Pass
PBKDF2             : KAT_KDF                     : Pass
KBKDF              : KAT_KDF                     : Pass
KBKDF_KMAC         : KAT_KDF                     : Pass
HKDF               : KAT_KDF                     : Pass
SSKDF              : KAT_KDF                     : Pass
X963KDF            : KAT_KDF                     : Pass
X942KDF            : KAT_KDF                     : Pass
HASH               : DRBG                        : Pass
CTR                : DRBG                        : Pass
```

```
HMAC              : DRBG                         : Pass
DH                : KAT_KA                       : Pass
ECDH              : KAT_KA                       : Pass
ML-KEM            : KAT_AsymmetricKeyGeneration  : Pass
ML-DSA            : KAT_AsymmetricKeyGeneration  : Pass
SLH-DSA           : KAT_AsymmetricKeyGeneration  : Pass
KEM_Encap         : KAT_KEM                      : Pass
KEM_Decap         : KAT_KEM                      : Pass
KEM_Decap_Reject  : KAT_KEM                      : Pass
[smit] FIPS provider installed.
System file integrity init check failed!

FIPS-CC mode: Starting self-tests.
Running Configuration/VPN Bypass test...      passed
Running AES test...                           passed
Running SHA1-HMAC test...                     passed
Running SHA256-HMAC test...                   passed
Running SHA384/512-HMAC test...               passed
Running IKEv1-KDF test...                     passed
Running IKEv2-KDF test...                     passed
Running Primitive-Z test...                   passed
Running Firmware integrity test...            passed
Running RBG-instantiate test...               passed
Running RBG-reseed test...                    passed
Running RBG-generate test...                  passed
Self-tests passed

[ipshelper] Installing and loading FIPS provider:
HMAC              : KAT_Integrity              : Pass
HMAC              : Module_Integrity           : Pass
SHA2              : KAT_Digest                 : Pass
SHA3              : KAT_Digest                 : Pass
AES_GCM           : KAT_Cipher                 : Pass
AES_ECB_Decrypt   : KAT_Cipher                 : Pass
TDES              : KAT_Cipher                 : Pass
RSA               : KAT_Signature              : Pass
ECDSA             : KAT_Signature              : Pass
ECDSA             : KAT_Signature              : Pass
EDDSA             : KAT_Signature              : Pass
EDDSA             : KAT_Signature              : Pass
DSA               : KAT_Signature              : Pass
ML-DSA            : KAT_Signature              : Pass
SLH-DSA           : KAT_Signature              : Pass
SLH-DSA           : KAT_Signature              : Pass
TLS13_KDF_EXTRACT : KAT_KDF                    : Pass
TLS13_KDF_EXPAND  : KAT_KDF                    : Pass
TLS12_PRF         : KAT_KDF                    : Pass
PBKDF2            : KAT_KDF                    : Pass
KBKDF             : KAT_KDF                    : Pass
KBKDF_KMAC        : KAT_KDF                    : Pass
HKDF              : KAT_KDF                    : Pass
```

```
SSKDF             : KAT_KDF                      : Pass
X963KDF           : KAT_KDF                      : Pass
X942KDF           : KAT_KDF                      : Pass
HASH              : DRBG                         : Pass
CTR               : DRBG                         : Pass
HMAC              : DRBG                         : Pass
DH                : KAT_KA                       : Pass
ECDH              : KAT_KA                       : Pass
ML-KEM            : KAT_AsymmetricKeyGeneration  : Pass
ML-DSA            : KAT_AsymmetricKeyGeneration  : Pass
SLH-DSA           : KAT_AsymmetricKeyGeneration  : Pass
KEM_Encap         : KAT_KEM                      : Pass
KEM_Decap         : KAT_KEM                      : Pass
KEM_Decap_Reject  : KAT_KEM                      : Pass
[ipshelper] FIPS provider installed.
```

```
# get sys status
Version: FortiGate-121G v8.0.0,buildXXXX
...
Virtual domains status: 1 in NAT mode, 0 in TP mode
Virtual domain configuration: disable
FIPS-CC mode: enable
Current HA mode: standalone
...
System time: Mon Sep 22 15:59:24 2025
                      Last reboot reason: warm reboot
```

2. TLS groups can be set when configuring Agentless VPN:

```
config vpn ssl settings
    set tls-groups P-521 P-384 FFDHE4096 FFDHE8192
end
```

Use Case 2: Normal mode customer testing PQC

An enterprise is testing PQC capabilities in preparation for future regulatory requirements and to evaluate quantum-resistant security options without current compliance restrictions in a FortiGate deployment supporting remote workers that use Agentless VPN.

In this example TLSv1.3 is used and the client is a Chrome web browser.

1. TLS groups can be set when configuring Agentless VPN:

```
config vpn ssl settings
    set tls-groups ML-KEM512 ML-KEM768 ML-KEM1024 X25519 P-384-MLKEM1024 P-256-MLKEM768
X25519-MLKEM768
end
```

2. The **Client Hello** message shows groups supported by the Chrome browser:

```
Supported Groups (5 groups)
    Supported Group: Reserved (GREASE) (0xcaca)
    Supported Group: X25519MLKEM768 (0x11ec)
    Supported Group: x25519 (0x001d)
    Supported Group: secp256r1 (0x0017)
    Supported Group: secp384r1 (0x0018)
```

3. The **Server Hello** message shows the group that the FortiGate picked:

```
Extension: key_share (len=1124) X25519MLKEM768
    Type: key_share (51)
    Length: 1124
    Key Share extension
```

Agentless VPN troubleshooting

The following topics provide information about Agentless VPN troubleshooting:

- Debug commands on page 3481
- Troubleshooting common issues on page 3482

Debug commands

Agentless VPN debug command

Use the following diagnose commands to identify Agentless VPN issues. These commands enable debugging of Agentless VPN with a debug level of -1 for detailed results.

```
diagnose debug application sslvpn -1
diagnose debug enable
```

The CLI displays debug output similar to the following:

```
FGT60C3G10002814 # [282:root]SSL state:before/accept initialization (172.20.120.12)
[282:root]SSL state:SSLv3 read client hello A (172.20.120.12)
[282:root]SSL state:SSLv3 write server hello A (172.20.120.12)
[282:root]SSL state:SSLv3 write change cipher spec A (172.20.120.12)
[282:root]SSL state:SSLv3 write finished B (172.20.120.12)
[282:root]SSL state:SSLv3 flush data (172.20.120.12)
[282:root]SSL state:SSLv3 read finished A:system lib(172.20.120.12)
[282:root]SSL state:SSLv3 read finished A (172.20.120.12)
[282:root]SSL state:SSL negotiation finished successfully (172.20.120.12)
[282:root]SSL established: DHE-RSA-AES256-SHA SSLv3 Kx=DH Au=RSA Enc=AES(256) Mac=SHA1
```

To disable the debug:

```
diagnose debug disable
diagnose debug reset
```

Remote user authentication debug command

Use the following diagnose commands to identify remote user authentication issues.

```
diagnose debug application fnbamd -1
diagnose debug reset
```

Troubleshooting common issues

To troubleshoot no visible Agentless VPN menus in the GUI:

Enable the feature visibility in the GUI or CLI.

In the GUI:

1. Go to *System* > *Feature Visibility*.
2. In the *Core Features* section, enable *Agentless VPN*.
3. Click *Apply*.

In the CLI:

```
config system settings
    set gui-sslvpn enable
end
```

To troubleshoot no visible Agentless VPN options in the GUI or CLI:

```
config system global
    set sslvpn-web-mode enable
end
```

To troubleshoot getting no response from the Agentless VPN URL:

1. Go to *VPN* > *Agentless VPN Settings*.
   a. Confirm that Agentless VPN is enabled.
   b. Check the Agentless VPN port assignment.
   c. Check the *Restrict Access* setting to ensure the host you are connecting from is allowed.
2. Go to *Policy* > *Firewall Policy*.
   a. Check that the policy for Agentless VPN traffic is configured correctly.
   b. Check the URL you are attempting to connect to. It should follow this pattern:

```
https://<FortiGate IP>:<Port>
```

c. Check that you are using the correct port number in the URL. Ensure FortiGate is reachable from the computer.

```
ping <FortiGate IP>
```

d. Check the browser has *TLS 1.1*, *TLS 1.2*, and *TLS 1.3* enabled.

To troubleshoot FortiGate connection issues:

1. Check the Release Notes to ensure that the web browser version is compatible with your version of FortiOS.
2. Check that your browser has TLS 1.1 and TLS 1.2 enabled.

To troubleshoot Agentless VPN connections shutting down after a few seconds:

This might occur if there are multiple interfaces connected to the Internet, for example, SD-WAN. This can cause the session to become “dirty”. To allow multiple interfaces to connect, use the following CLI commands.

If you are using a FortiOS 6.0.1 or later:

```
config system interface
  edit <name>
    set preserve-session-route enable
  next
end
```

User & Authentication

In *User & Authentication*, you can control network access for different users and devices in your network. FortiGate authentication controls system access by user group. By assigning individual users to the appropriate user groups you can control each user's access to network resources. You can define local users and peer users on the FortiGate unit. You can also define user accounts on remote authentication servers and connect them to FortiOS.

⚠ When configuring an LDAP connection to an Active Directory server, an administrator must provide Active Directory user credentials.

- To secure this connection, use LDAPS on both the Active Directory server and FortiGate. See Configuring an LDAP server on page 3509 and Configuring client certificate authentication on the LDAP server on page 3524.
- Apply the principle of least privilege. For the LDAP regular bind operation, do not use credentials that provide full administrative access to the Windows server when using credentials. See Configuring least privileges for LDAP admin account authentication in Active Directory on page 3517.

To secure RADIUS connections, consider using RADSEC over TLS instead. See Configuring a RADSEC client on page 3566.

You can control network access for different device types in your network by doing the following:

- Identifying and monitoring the types of devices connecting to your network
- Using MAC address based access control to allow or deny individual devices
- Using Telemetry data received from FortiClient endpoints to construct a policy to deny access to endpoints with known vulnerabilities or to quarantine compromised endpoints

The following sections provide information about users and devices:

- User definition, groups, and settings on page 3485
- Basic authentication with cached client certificates on page 3507
- LDAP servers on page 3509
- RADIUS servers on page 3527
- SAML on page 3573
- TACACS+ servers on page 3620
- SCIM servers on page 3622
- FortiTokens on page 3633
- PKI on page 3661
- FSSO on page 3669
- Include usernames in logs on page 3687

User definition, groups, and settings

FortiGate authentication controls system access by user groups. By assigning individual users to the appropriate user groups, this controls each user's access to network resources. The user groups members are user accounts, of which there are several types. Local and peer users are defined in FortiOS. User accounts can also be defined on remote authentication servers.

This section contains information about configuring the following:

- Users on page 3485
- User groups on page 3487
- Authentication settings on page 3495
- Retail environment guest access on page 3500
- Customizing complexity options for the local user password policy on page 3502

For information about configuring authentication servers, see the LDAP servers on page 3509, RADIUS servers on page 3527, TACACS+ servers on page 3620, and SAML on page 3573 sections.

Users

A user is a user account consisting of a username, password, and sometimes other information, that is configured in FortiOS or on an external authentication server. There are several types of user accounts with slightly different methods of authentication.

User type	Authentication method
Local	The username and password must match a user account stored in FortiOS. Authentication is done by a firewall policy.
Remote	Remote users consist of usernames defined in FortiOS that are authenticated by a remote server. For example, RADIUS, TACACS+, LDAP, or FortiNAC. The server must be configured in FortiOS before creating a user.
FSSO	Users on a Microsoft Windows, Citrix, or Novell network can use their network authentication to access resources through the FortiGate. Access is controlled through FSSO user groups, which contain Windows, Citrix, or Novell user groups as members. The FSSO agent must be configured in FortiOS before creating a user (see FSSO on page 3669).
PKI or peer	A PKI or peer user is a digital certificate holder that authenticates using a client certificate. No password is required, unless two-factor authentication is enabled. In the GUI, the *User & Authentication > PKI* menu is only available after a PKI user is configured in the CLI (see Configuring a PKI user on page 3662).
SAML	Local entries for SAML users, usable in firewall policies and agentless VPNs. See Local entries for SAML users on page 3606 for more information. A working SAML IdP connection must already be configured. See SAML on page 3573 for more information.

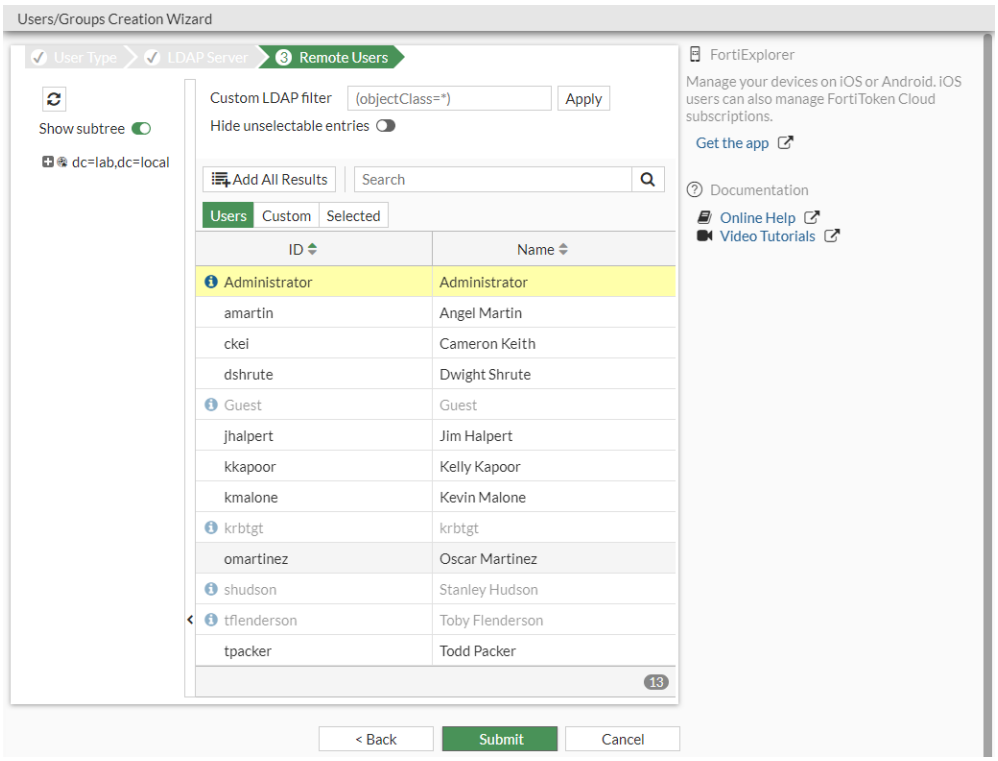
Some user types have an option to enable multi-factor authentication using FortiToken or FortiIdentity Cloud. In some cases, the user must be defined first, and then can be edited to add multi-factor authentication. See FortiTokens on page 3633 for more information.

To create a user:

1. Go to User *Authentication* > *User Definition* and click *Create New*. The *Users/Groups Creation Wizard* appears.
2. Select a *User Type* and click *Next*.
3. The remaining wizard steps depend on the user type:
   - *Local User*:
     i. Enter a *Username* and *Password*, then click *Next*.
     ii. Optionally, enable *Two-factor Authentication* and configure the following:

Authentication Type	Select *FortiToken Cloud* or *FortiToken*.
Token	If using FortiToken to authenticate, select a token.
Email Address	Enter an email address.
SMS	Enable to send an SMS message to activate the token.
Country Dial Code	Select the country code.
Phone Number	Enter a phone number.

     iii. Click *Next*
     iv. Set the *User Account Status* and, optionally, add the user to one or more user groups.
     v. Click *Submit*.
   - *Remote RADIUS User*, *Remote TACACS+ User*, and *SAML User*:
     i. Enter a *Username* and select the server.
     ii. Click *Next*.
     iii. Optionally, enable *Two-factor Authentication* and configure the settings as needed.
     iv. Click *Next*
     v. Set the *User Account Status* and, optionally, add the user to one or more user groups.
     vi. Click *Submit*.
   - *Remote LDAP User*:
     i. Select an *LDAP Server*, then click *Next*.
     ii. Select the users to add from the LDAP server. If the user ID matches an existing configured username, it cannot be added.



iii. Click *Submit*.

- *FSSO*:
  i. Select an *FSSO Agent*, click the *+* to add *AD Groups*, then click *Next*.
  ii. Select an FSSO group to add the *AD Groups* to. If an FSSO group already exists (see Configuring FSSO user groups on page 3493), click *Choose Existing* and select the group. Otherwise, click *Create New*, enter a name, and click *OK*.
  iii. Click *Next*
  iv. Set the *User Account Status* and, optionally, add the user to one or more user groups.
  v. Click *Submit*.

User groups

A user group is a list of user identities. A user identity can be a:

- Local user account (username/password) stored on the FortiGate
- Remote user account (password stored on a RADIUS, LDAP, or TACACS+ server)
- PKI user account with a digital client authentication certificate stored on the FortiGate
- RADIUS, LDAP, or TACACS+ server, optionally specifying particular user groups on that server
- User group defined on an FSSO server

User groups provide the ability to combine users that require the same permissions so they can be referenced at once, which enables consistency in configurations. User groups allow for remote servers to be referenced by leveraging the pre-existing user accounts, instead of redefining them on the FortiGate.

For example, when a new employee joins a department, they can be added to their respective group, whether in the remote authentication server or local group, and be subject to the same access as their colleagues in the same department. In FortiOS, user groups can be used when configuring firewall policies, traffic shaping policies, proxy policies, Agentless VPN portals, IPsec VPN XAUTH, ZTNA, wireless networks (SSIDs), web filtering profiles, identity-based routing, and system administrators with remote authentication.

In most cases, the FortiGate authenticates users by requesting their username and password. The FortiGate checks local user accounts first. If a match is not found, the FortiGate checks the RADIUS, LDAP, or TACACS+ servers that belong to the user group. Authentication succeeds when a matching username and password are found. If the user belongs to multiple groups on a server, those groups will also be matched.

Four types of user groups can be configured:

- Firewall
- FSSO
- RSSO
- Guest

Configuring firewall user groups

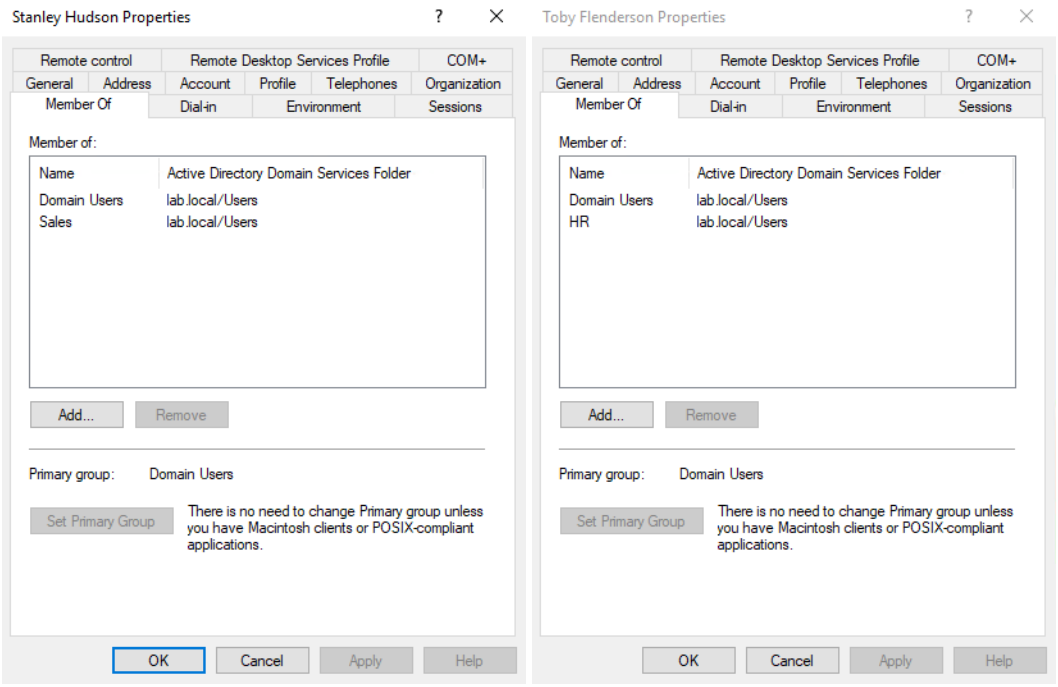
Firewall user groups are used locally as part of authentication. For example, when a firewall policy allows access only to specified user groups, users must authenticate before matching the policy. If the user authenticates successfully and is a member of one of the permitted groups, the policy is applied to the user. A firewall user group may contain local users (defined locally or authenticated remotely), PKI users, or authentication servers.

There are two options to add users in a firewall group configuration: members or remote groups. Members are the individual users who have been defined in FortiOS. Remote groups are remote server that users may authenticate to. One or more user groups can be specified within that server to limit which users can authenticate to the firewall user group. Both options may be used at the same time. The FortiGate attempts to authenticate users in the members list first, and then the remote groups if the initial authentication does not succeed.

When adding remote groups to user groups, FortiTokens cannot be applied to the users. To use remote authentication servers and FortiToken for multi-factor authentication, a remote user type must be created and then added as a user group member.

The following user group configuration examples have local members and a remote authentication server user group. There are two LDAP users, but the principle applies to other remote authentication server types.

Both LDAP users (shudson and tflenderson) belong to the primary group, Domain Users. The user, shudson belongs to the Sales group; tflenderson belongs to the HR group.

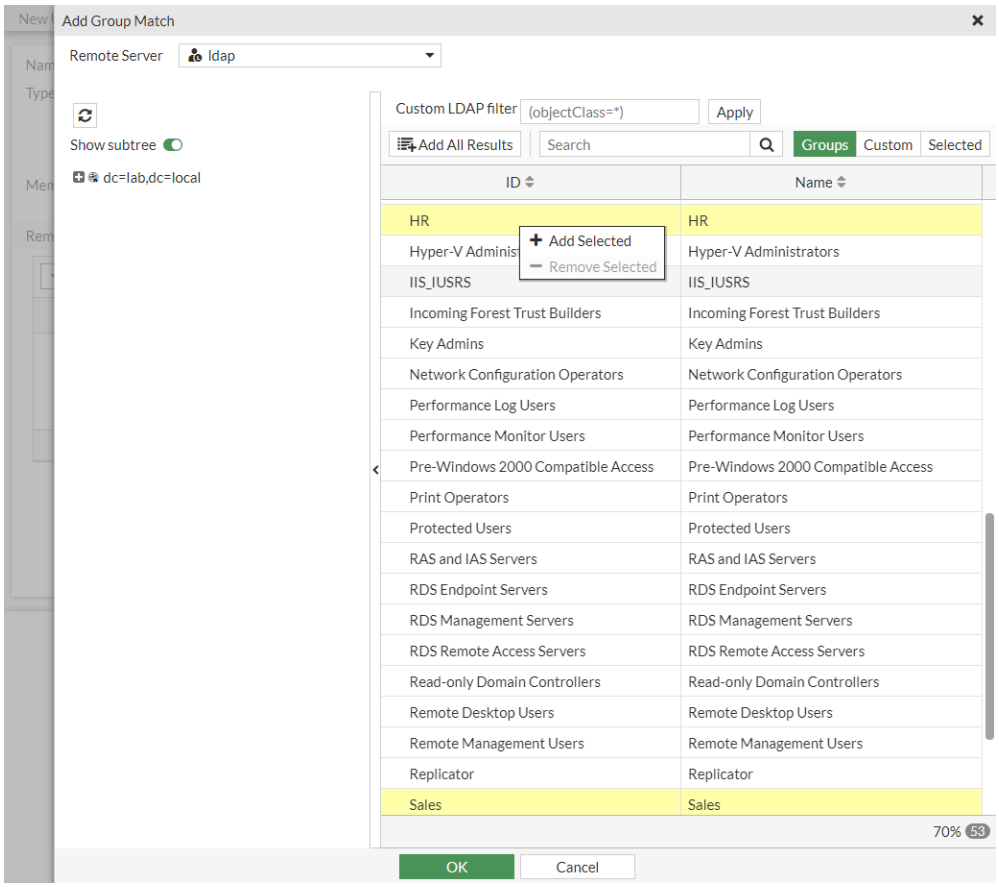


Example 1: Adding multiple remote groups to a user group

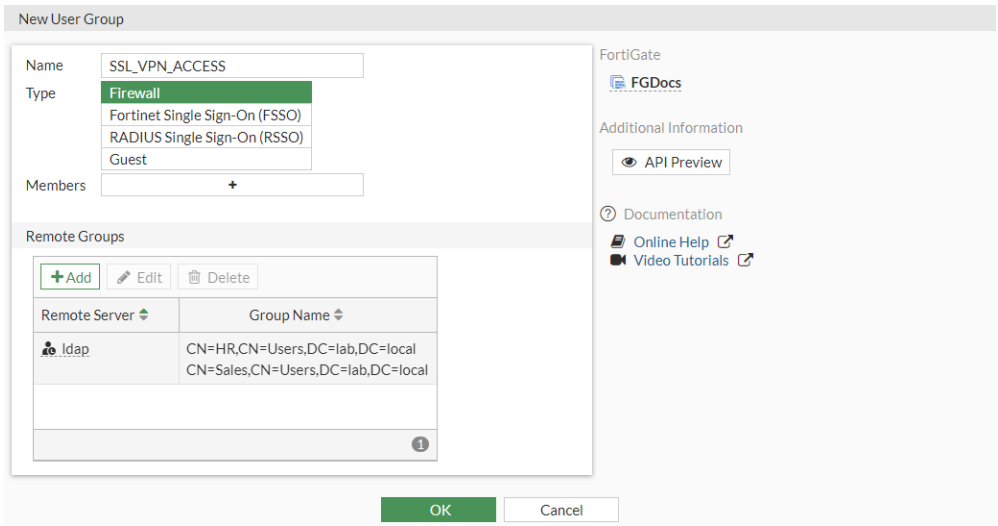
In this example, two remote groups (HR and Sales) are added to a firewall group called SSL_VPN_ACCESS.

To add multiple remote groups to a user group:

1. Go to *User & Authentication* > *User Groups* and click *Create New*. *Firewall* is selected as the default *Type*.
2. Enter the group name, *SSL_VPN_ACCESS*.
3. In the *Remote Groups* section, click *Add*.
4. Set *Remote Server* to the LDAP server (*ldap*).
5. In the *Groups* table, select *Sales*, then right-click and select *Add Selected*.
6. Select *HR*, then right-click and select *Add Selected*.



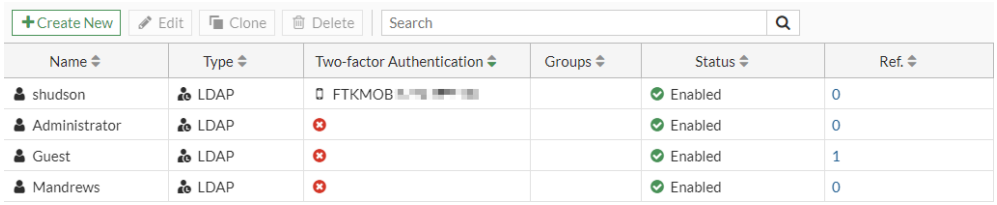
7. Click *OK*.
   Both user group paths are specified under the *Group Name*.



8. Click *OK*.
   In this configuration, shudson and tflenderson would be able to authenticate to this group.

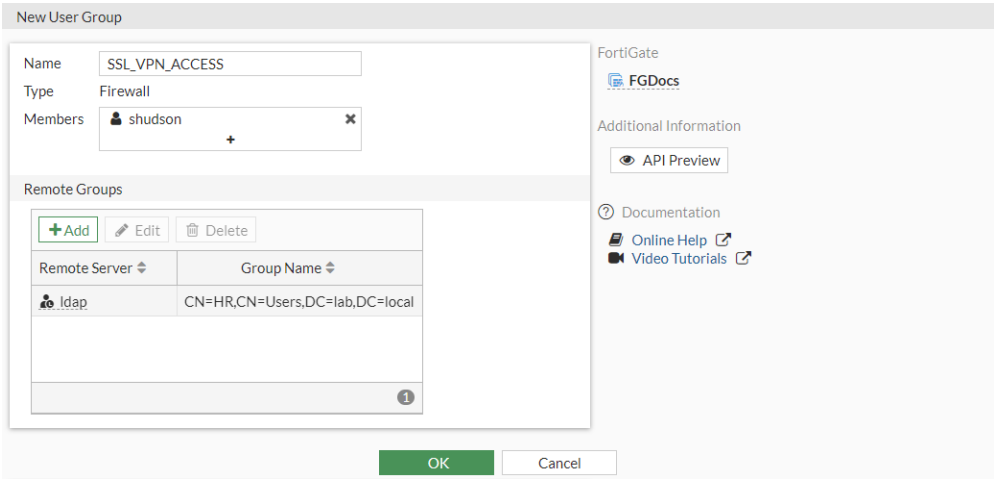
Example 2: combining remote groups and local users in a user group

In this example, the firewall group (SSL_VPN_ACCESS) is configured to contain the HR remote group and a local LDAP user (shudson) with multi-factor authentication.



To combine remote groups and local users in a user group:

1. Go to *User & Authentication* > *User Groups* and click *Create New*. *Firewall* is selected as the default *Type*.
2. Enter the group name, *SSL_VPN_ACCESS*.
3. In the *Remote Groups Section*, click *Add*.
4. Set *Remote Server* to the LDAP server (*ldap*).
5. In the *Groups* table, select *HR*, then right-click and select *Add Selected*.
6. Click *OK*.
7. In the *Members* field, click the *+* and add *shudson*.



8. Click *OK*.

   In this configuration, shudson, tflenderson, and any members of the HR LDAP group would be able to authenticate to the user group. Other users in the Sales group are not allowed.

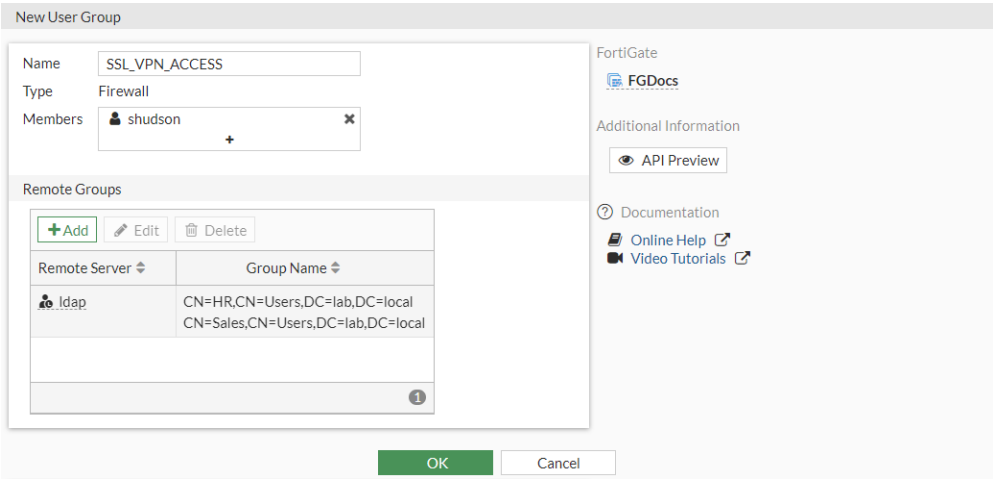
Example 3: adding a user as a member and their group as a remote groups

This example uses a combination of the previous examples. The HR and Sales groups are added as remote groups similar to example 1. The local LDAP user, shudson (using a FortiToken), from example 2 is added as a group member.

> This example is for demonstration only. It may cause unwanted results, so this configuration is not advised.

To add a user as a member and their group as a remote groups:

1. Refer to example 1 to configure the two remote groups.
2. In the *Members* field, click the *+* and add *shudson*.



3. Click *OK*.

One unwanted scenario from this configuration is that a user might be able to bypass multi-factor authentication on LDAP by changing the username case (see the related PSIRT advisory). By default, the username of the remote LDAP user is case sensitive. This means the username has to match what is configured (shudson). If a user types `sHudson`, for example, this will not match the user shudson, so it falls through to remote group authentication. It will match the Sales group in this example. To prevent this, disable username case sensitivity (see Agentless VPN for remote users with MFA and user sensitivity on page 3405 for more details).

To disable case sensitivity on the remote user:

```
config user local
    edit <name>
        set type ldap
        set two-factor fortitoken
        set fortitoken "FTKMOBxxxxxxxxxx"
        set email-to <email_address>
        set username-sensitivity disable
        set ldap-server <server_name>
    next
end
```

There is another unwanted scenario from this configuration than can occur to bypass multi-factor authentication. The LDAP server, ldap, has a user named shudson. Another LDAP server, ldap2, also has a user named shudson, but with a different password. If the ldap and ldap2 servers are added to the user group in addition to the remote shudson user, if a user tries to log in using `shudson` and the password on the ldap2 server, they would be able to bypass multi-factor authentication.

Configuring FSSO user groups

FSSO user groups contain only Windows, Citrix, and Novell network users. Information about these user groups and their member logon activities are provided by the corresponding FSSO connector. See the FSSO on page 3669 section for more information.

Configuring RSSO user groups

RADIUS single sign-on user groups leverage a RADIUS server to authenticate connecting users. This requires users to log in to their computer using their RADIUS account. The FortiGate does not interact with the remote RADIUS server. It only monitors RADIUS accounting records that the server forwards (originating from the RADIUS client). These records include the user IP address and user group. See RADIUS single sign-on agent on page 4688 for more information.

Configuring guest user groups

In some scenarios, an administrator might need to create temporary user accounts with a defined expiry time to access network resources. For example, if there is a large conference and may attendees require temporary network access for a few days. *Guest Management* can be used to combine many guest users into a group. Many guest accounts can be created at once using randomly-generated user IDs and passwords.

A guest group must be configured first. The guest user account user ID can be an email address, a randomly generated string, or an ID that the assigned by the administrator. The password can be assigned by the administrator or randomly generated. The guest group configuration determines the fields that are provided when creating guest user accounts in *Guest Management*.

To create a guest user group:

1. Go to *User & User & Authentication* > *User Groups* and click *Create New*.
2. Enter a name, and set the *Type* to *Guest*.
3. Configure the following:

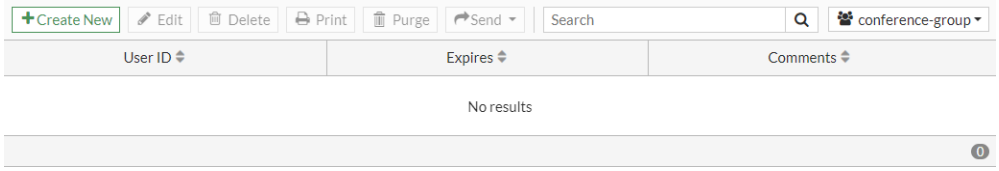
Batch Guest Account Creation	Create multiple accounts automatically. When enabled: • The user ID and password are automatically generated. • The accounts only have user ID, password, and expiration fields. The expiration field is editable in the GUI in the *Start Countdown* and *Time* settings. • An administrator can print the account information. • Users do not receive an email or SMS notification.
User ID	Select one of the following: • *Email*: use the user's email address • *Auto Generate*: FortiOS creates a random user ID • *Specify*: the administrator assigns a user ID

Maximum Accounts	Enable to set a maximum number of guest accounts that can be created for this group (disabled = unlimited).
Guest Details	
Enable Name	If enabled, the user form has a field to enter a name.
Enable Email	If enabled, the user is notified by email.
Enable SMS	If enabled, the user is notified by SMS.
Password	Select one of the following: • *Auto Generate*: FortiOS creates a random password • *Specify*: the administrator assigns a password If the setting is disabled, no password is used.
Sponsor	If enabled, the user form has a field to enter a sponsor (*Optional*). Select *Required* if the sponsor field is mandatory.
Company	If enabled, the user form has a field to enter a company (*Optional*). Select *Required* if the company field is mandatory.
Expiration	
Start Countdown	Select one of the following: • *On Account Creation*: the countdown starts from the time the account is created • *After First Login*: the countdown starts from the time the first time the user logs in
Time	Set the expiry time. There are fields to enter values for *Days*, *Hours*, *Minutes*, and *Seconds*.

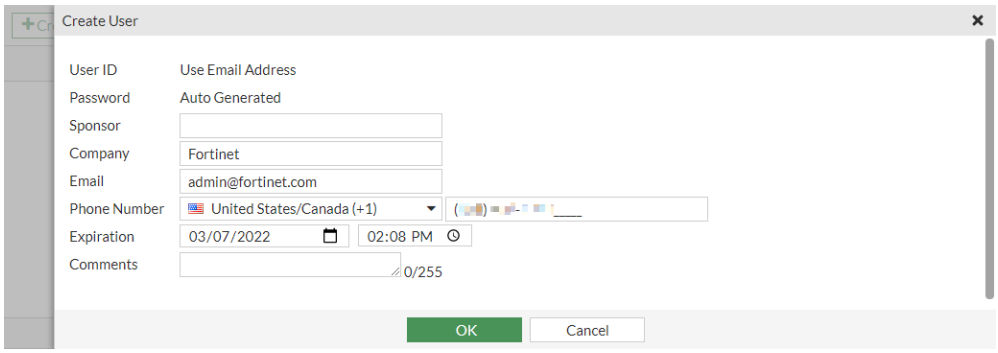
4. Click *OK*.

To manually create a guest user account:

1. Go to *User & User & Authentication > Guest Management*.
2. If more than one guest user group is configured, select the group from the dropdown beside the search box.



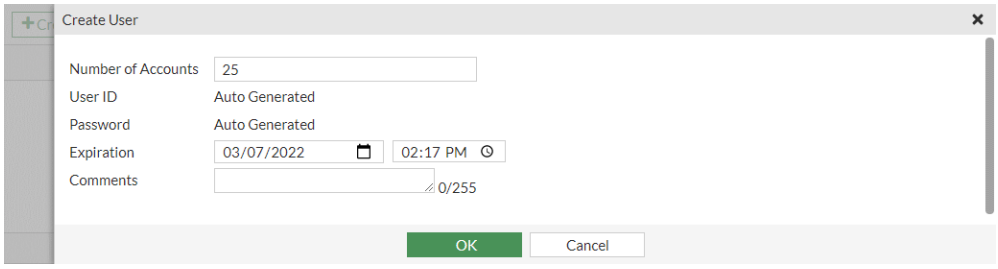
3. Click *Create New* and enter the information in the *Create User* pane. The fields are based on the guest group configuration. Optional fields can be left blank, such as *Sponsor* in this example.



4. Click *OK*.

To automatically create multiple guest user accounts:

1. Go to *User & User & Authentication* > *Guest Management*.
2. If more than one guest user group is configured, select the group from the dropdown beside the search box. The group must have *Batch Guest Account Creation* enabled.
3. Click *Create New* > *Multiple Users* and enter the *Number of Accounts*.
4. Optionally, edit the *Expiration* date and time.



5. Click *OK*.

Authentication settings

General authentication settings include:

- Timeout on page 3496
- Preserve authentication sessions after reboot on page 3497
- Protocols on page 3497
- Certificates on page 3498
- Lockouts on page 3498
- Authentication policy extensions on page 3499

Only some of the settings can be configured in the GUI.

To configure authentication settings in the GUI:

1. Go to *User & Authentication* > *Authentication Settings*.
2. Configure the following settings:

Setting	Description
Authentication Timeout	Enter the desired timeout, in minutes, from 1 to 1440 (24 hours). The default time is 5 minutes. Only idle timeout can be configured in the GUI.
Protocol support	Select the protocols to challenge during firewall user authentication.
HTTP redirect	Redirect HTTP challenge to a secure channel (HTTPS). This option is only available if *HTTP* is selected in the *Protocol Support* options.
Certificate	Select the local certificate to use for authentication.

3. Click *OK*.

Timeout

Authenticated users and user groups can have timeout values per user or group, in addition to FortiGate-wide timeouts. Three types of user timeouts can be configured:

Timeout type	Description
Idle	The idle timer starts when a user initiates a session. As long as data are transferred in this session, the timer continually resets. If the data flow stops, the timer is allowed to advance until it reaches its limit. When the user has been idle for too long, they must re-authenticate before traffic is allowed to continue in that session. This is the default setting. It can be configured in the GUI and CLI.
Hard	The hard timer starts when a user initiates a session. When the timeout is reached, all the sessions for that user must be re-authenticated. This timeout is not affected by any events. This setting can be configured in the CLI.
Session	The session timer starts when a user initiates a session. When the timeout is reached, existing sessions may continue. New sessions are not allowed until the user re-authenticates. This timeout is not affected by any events. This setting can be configured in the CLI.

The authentication timeout time is configured in minutes. The default is five minutes. If VDOMs are enabled, the global level `auth-timeout` user setting is the default that all VDOMs inherit. If the timeout time is set to zero,

To configure timeout for authenticated users:

```
config user setting
    set auth-timeout-type {idle-timeout | hard-timeout | new-session}
    set auth-timeout <integer>
end
```

To configure the authentication timeout for a user group:

```
config user group
    edit <name>
        set authtimeout <integer>
    next
end
```

If the group timeout time is zero (the default) or the user belongs to multiple RADIUS groups, then the user group timeout values are ignored and the global user timeout value is used.

Preserve authentication sessions after reboot

FortiGate models with a log disk can preserve authentication sessions a firewall reboot. This eliminates the need to reauthenticate after rebooting.

When session authentication backup is enabled, authenticated sessions are backed up at the configured interval. By default, session authentication backup is disabled.

```
config system global
    set auth-session-auto-backup {enable | disable}
    set auth-session-auto-backup-interval {1min | 5min | 15min | 30min | 1hr}
end
```

To configure authentication session backup:

1. Enable authentication backup and specify a backup interval.
   In this example, authentication backup runs every minute.

   ```
   config system global
       set auth-session-auto-backup enable
       set auth-session-auto-backup-interval 1min
   end
   ```

Protocols

When you enable user authentication within a security policy, the authentication challenge is normally issued for any of four protocols, depending on the connection protocol:

- HTTP (you can set this to redirect to HTTPS)
- HTTPS
- FTP
- Telnet

The selected protocols control which protocols support the authentication challenge. Users must connect with a supported protocol first so that they can subsequently connect with other protocols. If HTTPS is selected as a protocol support method, it allows the user to authenticate with a customized local certificate.

When you enable user authentication within a security policy, FortiOS challenges the security policy user to authenticate. For user ID and password authentication, the user must provide their username and password. For

certificate authentication (HTTPS, or HTTP redirected to HTTPS only), you can install customized certificates on the unit and the user can also install customized certificates on their browser. Otherwise, users see a warning message and must accept a default Fortinet certificate. The network user's web browser may deem the default certificate invalid.

Enable `auth-secure-http` to redirect HTTP challenges to a secure channel. Enable `auth-ssl-allow-renegotiation` to allow SSL re-negotiation for HTTPS authentication.

Enable `auth-http-basic` to use HTTP basic authentication for identity-based firewall policies. HTTP basic authentication usually causes a browser to display a pop-up authentication window instead of an authentication web page. Some basic web browsers, such as those on older mobile devices, may only support HTTP basic authentication.

FTP and Telnet authentication replacement messages cannot be customized.

To configure the protocols to challenge during firewall user authentication:

```
config user setting
    set auth-type {http https ftp telnet}
    set auth-secure-http {enable | disable}
    set auth-http-basic {enable | disable}
    set auth-ssl-allow-renegotiation  {enable | disable}
end
```

Certificates

Configure the HTTPS certificate and CA certificate to use for policy authentication.

To configure certificates for policy authentication:

```
config user setting
    set auth-cert <certificate>
    set auth-ca-cert <CA certificate>
end
```

Lockouts

Failed log in attempts can indicate malicious attempts to gain access to your network. To prevent this security risk, you can limit the number of failed log in attempts. After the configured maximum number of failed log in attempts is reached (1 - 10, default = 3), access to the account is blocked for the configured lockout duration (0 - 4294967295 seconds, default = 0)

To configure the maximum failed log in attempts and the lockout duration:

```
config user setting
    set auth-lockout-threshold <integer>
    set auth-lockout-duration <integer>
end
```

Authentication policy extensions

By default, unauthenticated traffic is permitted to fall to the next policy. This means that unauthenticated users are only forced to authenticate against a policy when there are no other matching policies. To avoid this, you can force authentication to always take place.

To set that authentication requirement:

```
config user setting
    set auth-on-demand {always | implicitly}
end
```

Where:

`always`	Always trigger firewall authentication on demand.
`implicitly`	Implicitly trigger firewall authentication on demand. This is the default setting (and the behavior in FortiOS 6.0 and earlier).

In the following example, authentication is required; traffic that would otherwise be allowed by the second policy is instead blocked by the first policy.

To use forced authentication:

```
config user setting
    set auth-on-demand always
end
```

```
config firewall policy
    edit 1
        set name "QA to Database"
        set srcintf "port10"
        set dstintf "port9"
        set srcaddr "QA_subnet"
        set dstaddr "Database"
        set action accept
        set schedule "always"
        set service "ALL"
        set fsso disable
        set groups "qa_group"
        set nat enable
    next
    edit 2
        set name "QA to Internet"
        set srcintf "port10"
        set dstintf "port9"
        set srcaddr "QA_subnet"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
```

```
        set fsso disable
        set nat enable
    next
end
```

Retail environment guest access

Businesses such as coffee shops provide free Internet access for customers. In this scenario, you do not need to configure guest management, as customers can access the WiFi access point without logon credentials.

However, consider that the business wants to contact customers with promotional offers to encourage future patronage. You can configure an email collection portal to collect customer email addresses for this purpose. You can configure a firewall policy to grant network access only to users who provide a valid email address. The first time a customer's device attempts WiFi connection, FortiOS requests an email address, which it validates. The customers' subsequent connections go directly to the Internet without interruption.

This configuration consists of the following steps:

1. Creating an email collection portal on page 3500
2. Creating a firewall policy on page 3501
3. Checking for collected emails on page 3501

Creating an email collection portal

The customer's first contact with your network is a captive portal that presents a webpage requesting an email address. When FortiOS has validated the email address, the customer's device MAC address is added to the collected emails device group.

This example modifies the freewifi WiFi interface to present an email collection captive portal.

To configure the freewifi SSID to use an email collection portal in the GUI:

1. Enable email collection:
   a. Go to *System* > *Feature Visibility*.
   b. In the *Additional Features* section, enable *Email Collection*.
   c. Click *Apply*.
2. Edit the freewifi SSID:
   a. Go to *WiFi & Switch Controller* > *SSIDs* and edit the *freewifi* SSID.
   b. In the *Security Mode Settings* section, set the *Security mode* to *Captive Portal*.
   c. Set the *Portal type* to *Email Collection*.
   d. Click *OK*.

To configure the freewifi SSID to use an email collection portal in the CLI:

```
config wireless-controller vap
    edit freewifi
        set security captive-portal
        set portal-type email-collect
```

```
    next
end
```

Creating a firewall policy

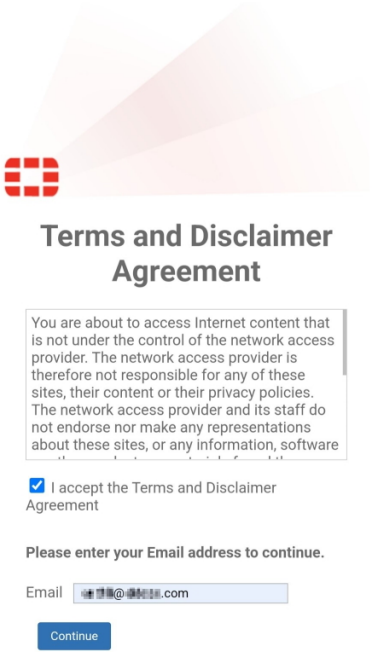
You must configure a firewall policy that allows traffic to flow from the WiFi SSID to the internet interface only for members of the collected emails device group. This policy must be listed first. Unknown devices are not members of the collected emails device group, so they do not match the policy.

To create a firewall policy:

```
config firewall policy
    edit 3
        set srcintf "freewifi"
        set dstintf "wan1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set nat enable
        set email-collect enable
    next
end
```

Checking for collected emails

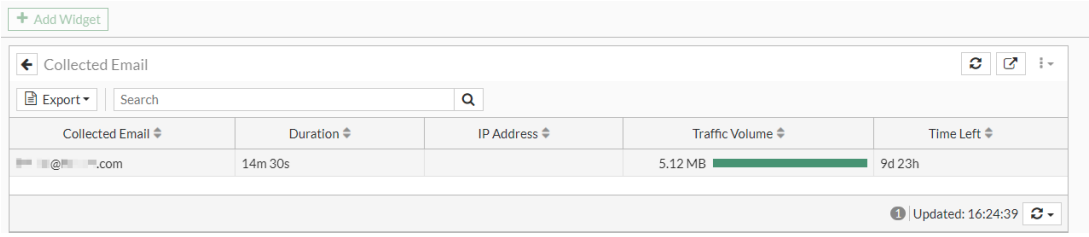
When a WiFi user connects to the freewifi SSID, they are presented with a captive portal to enter their email address.



Once the user enters their email and clicks *Continue*, they will have access to the Internet. The collected emails can be verified in FortiOS.

To check for collected emails in the GUI:

1. Go to *Dashboard* > *Assets & Identities* and click *Add Widget*.
2. In the *User & Authentication* section, select *Collected Email* and click *Add Widget*.
3. Click *Close*.
4. Click the *Collected Email* to expand to full view. The list of emails is displayed.



5. Optionally, click *Export* to export the data as a CSV or JSON file.

To check for collected emails in the CLI:

```
# diagnose firewall auth mac list

72:4d:e1:**:**:**, admin@fortinet.com
        type: email, id: 0, duration: 937, idled: 19
        expire: 863980, allow-idle: 864000
        flag(1000): src_idle
        packets: in 4753 out 4592, bytes: in 2662403 out 2458644

----- 1 listed, 0 filtered ------
```

Customizing complexity options for the local user password policy

The local firewall user password policy can be customized with various settings, such as minimum length, character types, and password reuse. These settings are similar to the ones available for the system administrator password policy, which offer more security and flexibility than the previous local user password policy.

```
config user password-policy
    edit <name>
        set minimum-length <integer>
        set min-lower-case-letter <integer>
        set min-upper-case-letter <integer>
        set min-non-alphanumeric <integer>
        set min-number <integer>
        set min-change-characters <integer>
        set expire-status {enable | disable}
```

```
        set reuse-password {enable | disable}
    next
end
```

Command	Description
`minimum-length <integer>`	Set the minimum password length (8 - 128, default = 8).
`min-lower-case-letter <integer>`	Set the minimum number of lowercase characters in the password (0 - 128, default = 0).
`min-upper-case-letter <integer>`	Set the minimum number of uppercase characters in the password (0 - 128, default = 0).
`min-non-alphanumeric <integer>`	Set the minimum number of non-alphanumeric in the password (0 - 128, default = 0).
`min-number <integer>`	Set the minimum number of numeric characters in the password (0 - 128, default = 0).
`min-change-characters <integer>`	Set the minimum number of unique characters in new password, which do not exist in the old password (0 - 128, default = 0). This attribute overrides `reuse-password` if both are enabled.
`expire-status {enable | disable}`	Enable/disable password expiration (default = disable).
`reuse-password {enable | disable}`	Enable/disable password reuse (default = enable. If both `reuse-password` and `min-change-characters` are enabled, `min-change-characters` overrides it.

After upgrading to 7.4.1 from 7.4.0, users must activate the user password policy using the CLI. The previous password policy settings will remain valid, but they will not be effective unless the password policy password expiration is enabled (`expire-status`). If the password policy password expiration is not enabled, the `expire-days <integer>` option will not force users to change their password after number of specified days.

Example

The following user password policy is configured before upgrading to 7.4.1:

```
config user password-policy
    edit "1"
        set expire-days 1
        set warn-days 1
        set expired-password-renewal enable
    next
end
```

To configure the user password policy options:

1. Check the user password policy settings after the upgrade:

```
config user password-policy
    edit 1
        get
            name                : 1
            expire-days         : 1
            warn-days           : 1
            expired-password-renewal: enable
            minimum-length      : 8
            min-lower-case-letter: 0
            min-upper-case-letter: 0
            min-non-alphanumeric: 0
            min-number          : 0
            min-change-characters: 0
            expire-status       : disable
            reuse-password      : enable
    next
end
```

2. Edit the user password policy settings, including enabling password expiration:

```
config user password-policy
    edit "1"
        set expire-days 1
        set warn-days 1
        set expired-password-renewal enable
        set min-lower-case-letter 1
        set min-upper-case-letter 1
        set min-non-alphanumeric 3
        set min-number 3
        set min-change-characters 2
        set expire-status enable
        set reuse-password disable
    next
end
```

3. Change a password for a local user.
   a. In the CLI when the password meets the criteria:

```
config user local
    edit pwd-test1
        set passwd CCbcset123!!!
    next
end
```

   b. In the CLI when the password does not meet the criteria (only two numbers, so an error message appears):

```
config user local
    edit pwd-test1
```

```
        set passwd CCbXsetp23!!!
New password must conform to the password policy enforced on this user:
Password must:
        Be a minimum length of 8
        Include at least 1 lower case letter(s) (a-z)
        Include at least 1 upper case letter(s) (A-Z)
        Include at least 3 non-alphanumeric character(s)
        Include at least 3 number(s) (0-9)
        Have at least 2 unique character(s) which don't exist in the old password
        Not be same as last two passwords

node_check_object fail! for passwd CCbXsetp23!!!

value parse error before 'CCbXsetp23!!!'
Command fail. Return code -49
```

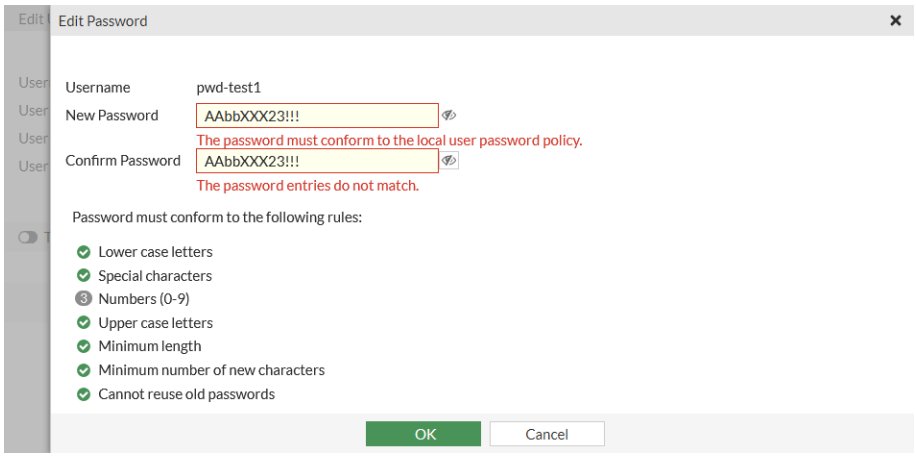
c. In the GUI:

i. Go to *User & Authentication* > *User Definition* and edit a local user.

ii. Click *Change Password*.

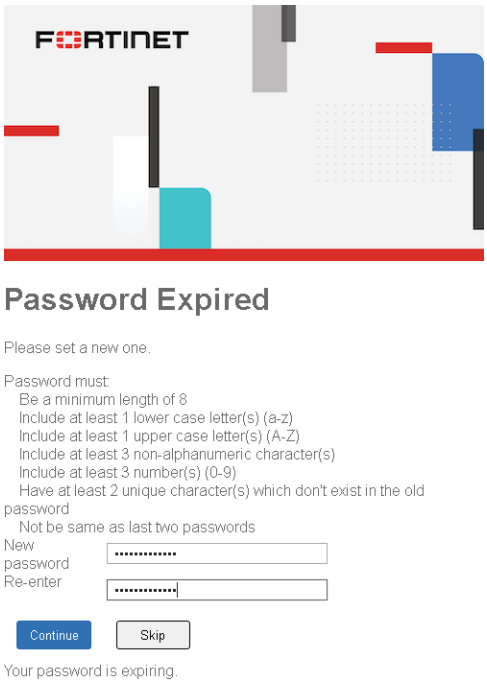
iii. Enter the *New Password*.

iv. Enter the password again (*Confirm Password*). A warning will appear when the password does not match the criteria and indicates which parameters must be fixed. In this example, there are less than three numbers used.

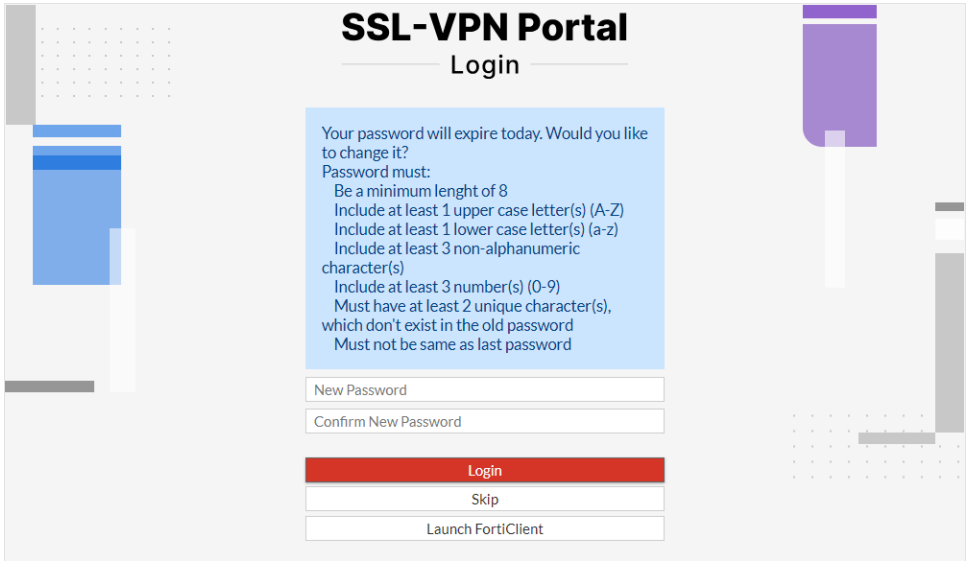


v. Click *OK*.

Sample prompt when a local user needs to update their password for firewall authentication:



Sample prompt when a local user needs to update their password for SSL VPN portal access:



Basic authentication with cached client certificates

With basic authentication, client certificates can be cached and used as authentication cookies, eliminating the need for repeated user authentication.

In this example, a CA signs a client certificate. The client certificate is installed on two endpoints, and the root CA certificate is imported to FortiGate.

During the authentication process, the client certificate from the endpoint is verified against the CA certificate. Once this verification is successful, the user is prompted to enter login credentials for user authentication. Once authenticated, the client certificate is stored as an authentication cookie so that subsequent access does not require any user authentication as long as the client certificate remains present on the endpoint.

To configure client certificates as authentication cookies:

1. Prepare the certificate:
   a. Use a CA to sign the client certificate.
   b. Import the root CA certificate that signed the client certificate to FortiGate.
   c. Install the client certificate on all endpoints.
2. In FortiOS, configure an authentication scheme to apply authentication against the local user database.

```
config authentication scheme
    edit "test"
        set method basic
        set user-database "local-user-db"
    next
end
```

3. Configure an authentication rule to enable the client certificate to be cached.

```
config authentication rule
    edit "test"
        set srcaddr "all"
        set ip-based disable
        set active-auth-method "test"
        set cert-auth-cookie enable
    next
end
```

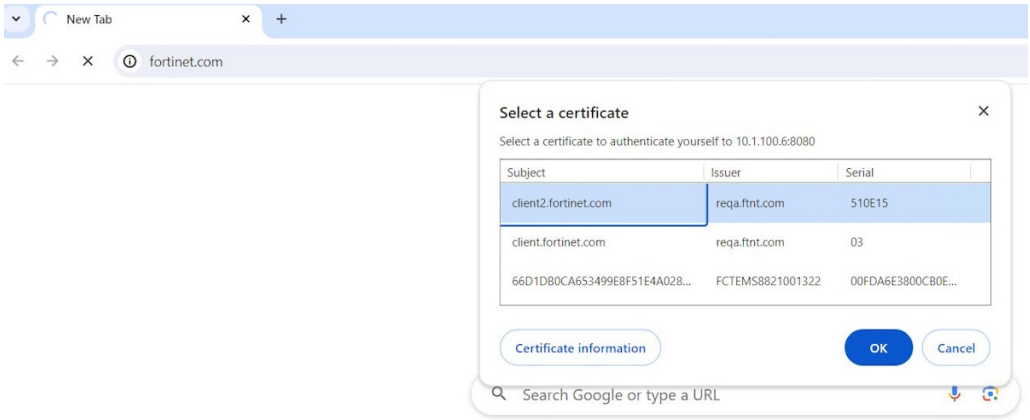
4. Configure verification of the client certificate with the root CA.

```
config authentication setting
    set user-cert-ca "root_ca"
end
```

When the user accesses a resource, such as a web site, for the first time:

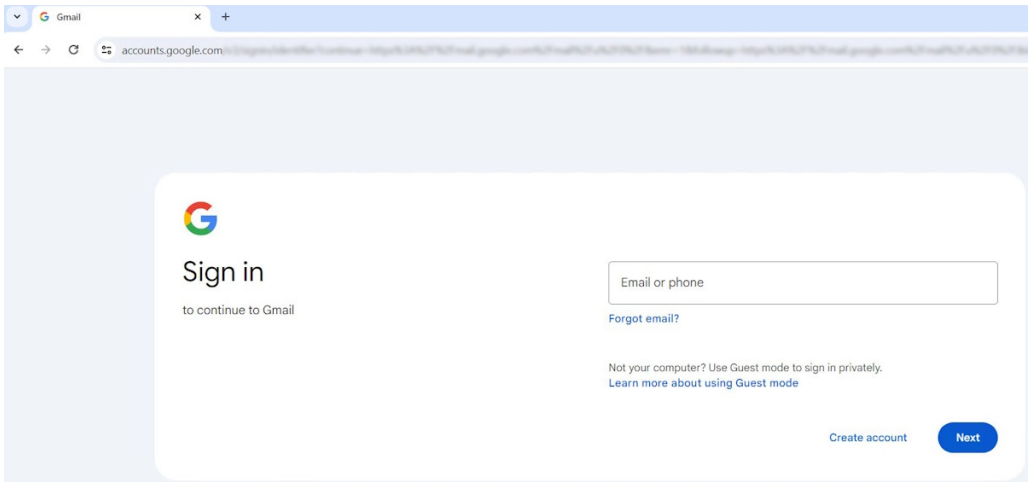
1. The browser prompts the user for a client certificate. The user selects the certificate (client2.fortinet.com), and clicks *OK*. Then the endpoint device (IP address 10.1.100.59) presents the client certificate to FortiGate

for verification.



2. Once the certificate verification passes, an authentication dialog box is displayed.

3. The user enters their username and password to authenticate with FortiGate and successfully access the web site.



FortiGate also logs the first access in the traffic log:

```
9: date=2024-04-24 time=12:28:51 eventtime=1713918531092354265 tz="+1200" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="vdom1" srcip=10.1.100.59 srcport=63615
srcintf="port2" srcintfrole="undefined" dstcountry="United States" srccountry="Reserved"
dstip=142.251.33.69 dstport=443 dstintf="port3" dstintfrole="undefined" sessionid=51442
service="web" proxyapptype="http" proto=6 action="accept" policyid=10 policytype="proxy-
```

```
policy" poluuid="e272fe7e-00d2-51ef-5fe0-09d157495e71" duration=73 user="localuser"
group="localgroup" authserver="localuser" gatewayid=1 realserverid=1 vip="ztna"
accessproxy="ztna" clientdevicemanageable="manageable" clientcert="yes" wanin=5734
rcvdbyte=5734 wanout=1505 lanin=3226 sentbyte=3226 lanout=42588 appcat="unscanned"
```

When the user accesses the resource from the same endpoint device for the second and subsequent times, FortiGate uses the cached authentication cookie to grant access, as long as the client certificate remains present on the endpoint.

When the user has multiple endpoint devices with the same certificate installed, the certificate will match the cached authentication cookie on the FortiGate, and the user can access resources without additional authentication.

This log shows a user accessing a website from a different PC (IP address 10.1.100.78) without needing to provide user credentials.

```
2: date=2024-04-24 time=12:30:42 eventtime=1713918642320943415 tz="+1200" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="vdom1" srcip=10.1.100.78 srcport=63799
srcintf="port2" srcintfrole="undefined" dstcountry="United States" srccountry="Reserved"
dstip=142.251.33.69 dstport=443 dstintf="port3" dstintfrole="undefined" sessionid=51819
service="web" proxyapptype="http" proto=6 action="accept" policyid=10 policytype="proxy-policy"
poluuid="e272fe7e-00d2-51ef-5fe0-09d157495e71" duration=9 user="localuser" group="localgroup"
gatewayid=1 realserverid=1 vip="ztna" accessproxy="ztna" clientdevicemanageable="manageable"
clientcert="yes" wanin=5737 rcvdbyte=5737 wanout=1295 lanin=3102 sentbyte=3102 lanout=7651
appcat="unscanned"
```

LDAP servers

The following topics provide information about LDAP servers:

- Configuring an LDAP server on page 3509
- Enabling Active Directory recursive search on page 3513
- Configuring LDAP dial-in using a member attribute on page 3514
- Configuring wildcard admin accounts on page 3515
- Configuring least privileges for LDAP admin account authentication in Active Directory on page 3517
- Tracking users in each Active Directory LDAP group on page 3518
- Tracking rolling historical records of LDAP user logins on page 3521
- Configuring client certificate authentication on the LDAP server on page 3524

Configuring an LDAP server

FortiOS can be configured to use an LDAP server for authentication.

When configuring an LDAP connection to an Active Directory server, an administrator must provide Active Directory user credentials.

- To secure this connection, use LDAPS on both the Active Directory server and FortiGate. See relevant LDAPS information in this topic and Configuring client certificate authentication on the LDAP server on page 3524.
- Apply the principle of least privilege. For the LDAP regular bind operation, do not use credentials that provide full administrative access to the Windows server when using credentials. See Configuring least privileges for LDAP admin account authentication in Active Directory on page 3517.

To configure an LDAP server on the FortiGate:

1. Go to *User & Authentication* > *LDAP Servers*.
2. Click *Create New*.
3. Configure the following:

Name	This connection name is for reference within the FortiGate only.
Server IP/Name	LDAP server IP address or FQDN resolvable by the FortiGate.
Server Port	By default, LDAP uses port 389 and LDAPS uses 636. Use this field to specify a custom port if necessary.
Common Name Identifier	Attribute field of the object in LDAP that the FortiGate uses to identify the connecting user. The identifier is case sensitive. Common attributes are: • *cn* (Common Name) • *sAMAccountName* (SAMAccountName) • *uid* (User ID)
Distinguished Name	Used to look up user account entries on the LDAP server. It reflects the hierarchy of LDAP database object classes above the CN identifier in which you are doing the lookup. Enter *dc=COMPANY,dc=com* to specify the root of the domain to include all objects. Enter *ou=VPN-Users,dc=COMPANY,dc=com* to look up users under a specific organization unit.
Exchange server	Enable to specify the exchange server connector to collect information about authenticated users from a corporate exchange server. See Exchange Server connector on page 4691 for more details.
Bind Type	Select one of the following options: • *Simple*: bind using simple password authentication using the client name. The LDAP server only looks up against the distinguished name (DN), but does not search on the subtree. • *Anonymous*: bind using an anonymous user, and search starting from the DN and recurse over the subtrees. Many LDAP servers do not allow this.

	• *Regular*: bind using the username and password provided, and search starting from the DN and recurse over the subtrees.
Username	If using regular bind, enter a username with sufficient privileges to access the LDAP server. The following formats are supported: • *username\administrator* • *administrator@domain* • *cn=administrator,cn=users,dc=domain,dc=com*
Password	If using regular bind, enter the password associated with the username.
Protocol	Select *STARTTLS* (default) or *LDAPS*. Selecting *STARTTLS* sets the port to 389 and selecting *LDAPS* sets the port to 636. Secure connection can be disabled in the CLI.
Certificate	Enable and select the root CA certificate so that the FortiGate will only accept a certificate from the LDAP server that is signed by this CA. The root CA certificate should be in the Remote CA Certificate store on the FortiGate. If this setting is not enabled (meaning that no certificate is chosen), the server certificate validation will not be performed. If the wrong certificate is chosen, which is not the issuing CA for the server certificate, then the LDAP connection will fail.
Server identity check	This check verifies the server domain or IP address against the server certificate. This option is enabled by default when *Certificate* is chosen and it is recommended to leave it enabled for a secure configuration.

When specifying a secure connection, there are some considerations for the certificate used by LDAP to secure the connection. The FortiGate checks the certificate presented by the LDAP server for the IP address or FQDN as specified in the *Server IP/Name* field with the following logic:

- If there is a Subject Alternative Name (SAN), it will ignore any Common Name (CN) value and look for a match in any of the SAN fields.
- If there is no SAN, it will check the CN for a match.

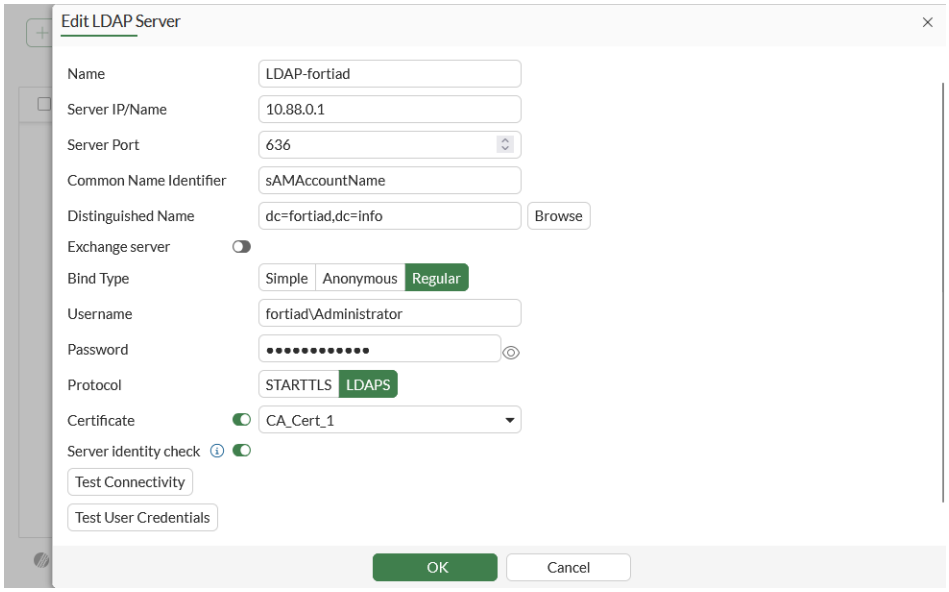
4. Optionally, click *Test User Credentials* to ensure that the account has sufficient access rights.
5. Click *OK*.
   The FortiGate checks the connection and updates the *Connection Status*.

To configure a secure connection to the LDAP server in the GUI:

1. Go to *User & Authentication > LDAP Servers*.
2. Click *Create New*.
3. Configure the following:

Name	LDAP-fortiad
Server IP/Name	10.88.0.1

Server Port	636
Common Name Identifier	sAMAccountName
Distinguished Name	dc=fortiad,dc=info
Exchange server	Disabled
Bind Type	Regular Enter the *Username* and *Password* for LDAP binding and lookup.
Protocol	LDAPS
Certificate	Enable and select the CA certificate to validate the server certificate.
Server identity check	Enable to verify the domain name or IP address against the server certificate.



4. Click *Test Connectivity* to verify the connection to the server.
5. Click *OK*.

To configure a secure connection to the LDAP server in the CLI:

```
config user ldap
    edit "LDAP-fortiad"
        set server "10.88.0.1"
        set cnid "sAMAccountName"
        set dn "dc=fortiad,dc=info"
        set type regular
        set username "fortiad\\Administrator"
        set password <password>
        set secure ldaps
        set ca-cert "CA_Cert_1"
        set port 636
```

```
    next
end
```

To account for dynamic IP address changes, such as those governed by SD-WAN rules, interface names can be used to define the source IP addresses in RADIUS, LDAP, and DNS configurations using the `source-ip-interface` command. See Implement the interface name as the source IP address in RADIUS, LDAP, and DNS configurations on page 378.

Enabling Active Directory recursive search

By default, nested groups (groups that are members or other groups) are not searched in Windows Active Directory (AD) LDAP servers because this can slow down the group membership search. There is an option in FortiOS to enable the searching of nested groups for user group memberships on AD LDAP servers.

This option is not available for other LDAP servers, such as OpenLDAP-based servers.

The default behavior does not include nested groups:

```
config user ldap
    edit "ldap-ad"
        set server "10.1.100.131"
        set cnid "cn"
        set dn "dc=fortinet-fsso,dc=com"
        set type regular
        set username "cn=Administrator,cn=users,dc=fortinet-fsso,dc=com"
        set password XXXXXXXXXXXXXXXXXXXXXXXX
    next
end
```

The default search results only show groups that have the user as member, and no groups that have groups as members:

```
diagnose test authserver ldap ldap-ad nuser nuser
    authenticate 'nuser' against 'ldap-ad' succeeded!
    Group membership(s) - CN=nested3,OU=Testing,DC=Fortinet-FSSO,DC=COM
                          CN=Domain Users,CN=Users,DC=Fortinet-FSSO,DC=COM
```

To enable recursive search to include nested groups in the results:

```
config user ldap
    edit "ldap-ad"
        set server "10.1.100.131"
        set cnid "cn"
        set dn "dc=fortinet-fsso,dc=com"
        set type regular
        set username "cn=Administrator,cn=users,dc=fortinet-fsso,dc=com"
        set password XXXXXXXXXXXXXXXXXXXXXXXX
```

```
        set search-type recursive
    next
end
```

The search results now include groups that have other groups as members:

```
diagnose test authserver ldap ldap-ad nuser nuser
    authenticate 'nuser' against 'ldap-ad' succeeded!
    Group membership(s) - CN=nested3,OU=Testing,DC=Fortinet-FSSO,DC=COM
                          CN=Domain Users,CN=Users,DC=Fortinet-FSSO,DC=COM
                          CN=nested2,OU=Testing,DC=Fortinet-FSSO,DC=COM
                          CN=nested1,OU=Testing,DC=Fortinet-FSSO,DC=COM
```

The group nested3 is a member of the group nested2, which is a member of the group nested1.

Configuring LDAP dial-in using a member attribute

In this configuration, users defined in Microsoft AD can set up a VPN connection based on an attribute that is set to `TRUE`, instead of their user group. You can activate the *Allow Dialin* property in AD user properties, which sets the `msNPAllowDialin` attribute to *TRUE*. You can use this procedure for other member attributes as your system requires.

This configuration consists of the following steps:

1. Ensure that the AD server has the `msNPAllowDialin` attribute set to *TRUE* for the desired users.
2. Configure user LDAP member attribute settings.
3. Configure LDAP group settings.
4. Ensure that you configured the settings correctly.

To configure user LDAP member attribute settings:

```
config user ldap
    edit "ldap_server"
        set server "192.168.201.3"
        set cnid "sAMAccountName"
        set dn "DC=fortilabanz,DC=com,DC=au"
        set type regular
        set username "fortigate@sample.com"
        set password ******
        set member-attr "msNPAllowDialin"
    next
end
```

To configure LDAP group settings:

```
config user group
    edit "ldap_grp"
        set member "ldap_server"
        config match
```

```
            edit 1
                set server-name "ldap_server"
                set group-name "TRUE"
            next
        end
    next
end
```

To ensure that you configured the settings correctly:

Users that are members of the `ldap_grp` user group should be able to authenticate. The following shows sample `diagnose debug` output when the `Allow Dial-in` attribute is set to `TRUE`:

```
get_member_of_groups-Get the memberOf groups.
get_member_of_groups- attr='msNPAllowDialin', found 1 values
get_member_of_groups-val[0]='TRUE'
fnbamd_ldap_get_result-Auth accepted
fnbamd_ldap_get_result-Going to DONE state res=0
fnbamd_auth_poll_ldap-Result for ldap svr 192.168.201.3 is SUCCESS
fnbamd_auth_poll_ldap-Passed group matching
```

If the attribute is not set to `TRUE` but is expected, you may see the following output:

```
get_member_of_groups-Get the memberOf groups.
get_member_of_groups- attr='msNPAllowDialin', found 1 values
get_member_of_groups-val[0]='FALSE'
fnbamd_ldap_get_result-Auth accepted
fnbamd_ldap_get_result-Going to DONE state res=0
fnbamd_auth_poll_ldap-Result for ldap svr 192.168.201.3 is SUCCESS
fnbamd_auth_poll_ldap-Failed group matching
```

The difference between the two outputs is the last line, which shows passed or failed depending on whether the member attribute is set to the expected value.

Configuring wildcard admin accounts

To avoid setting up individual admin accounts in FortiOS, you can configure an admin account with the wildcard option enabled, allowing multiple remote admin accounts to match one local admin account. This way, multiple LDAP admin accounts can use one FortiOS admin account.

You can create one global wildcard system admin account or one wildcard system admin account for each VDOM but not both.

Benefits include:

- Fast configuration of the FortiOS admin account to work with your LDAP network, saving effort and avoiding potential errors incurred when setting up multiple admin accounts
- Reduced ongoing maintenance. As long as LDAP users belong to the same group and you do not modify the wildcard admin account in FortiOS, you do not need to configure changes on the LDAP accounts. If you add

or remove a user from the LDAP group, you do not need to perform changes in FortiOS.

Potential issues include:

- Multiple users may be logged in to the same account simultaneously. This may cause issues if both users make changes simultaneously.
- Security is reduced since multiple users have login access to the same account, as opposed to an account for each user.

Wildcard admin configuration also applies to RADIUS. If configuring for RADIUS, configure the RADIUS server and RADIUS user group instead of LDAP. When using the GUI, wildcard admin is the only remote admin account that does not require you to enter a password on account creation. That password is normally used when the remote authentication server is unavailable during authentication.

This example uses default values where possible. If a specific value is not mentioned, the example sets it to its default value.

When configuring an LDAP connection to an Active Directory server, an administrator must provide Active Directory user credentials. To secure this connection, use LDAPS on both the Active Directory server and FortiGate. See Configuring an LDAP server on page 3509 and Configuring client certificate authentication on the LDAP server on page 3524.

To secure RADIUS connections, consider using RADSEC over TLS. See Configuring a RADSEC client on page 3566.

You can configure an admin account in Active Directory for LDAP authentication to allow an admin to perform lookups and reset passwords without being a member of the Account Operators or Domain Administrators built-in groups. See Configuring least privileges for LDAP admin account authentication in Active Directory on page 3517.

To configure the LDAP server:

The important parts of this configuration are the username and group lines. The username is the domain administrator account. The group binding allows only the GRP group access.

This example uses an example domain name. Configure as appropriate for your own network.

```
config user ldap
    edit "ldap_server"
        set server "192.168.201.3"
        set cnid "sAMAccountName"
        set dn "DC=example,DC=com,DC=au"
        set type regular
        set username "CN=Administrator,CN=Users,DC=example,DC=COM”
        set password *
        set group-member-check group-object
        set group-object-filter (&amp;
(objectcategory=group)member="CN=GRP,OU=training,DC=example,DC=COM"))
    next
end
```

To configure the user group and add the LDAP server:

```
config user group
    edit "ldap_grp"
        set member "ldap_server"
            config match
                edit 1
                    set server-name "ldap_server"
                    set group-name "CN=GRP,OU=training,DC=example,DC=COM"
                next
            end
        end
    end
end
```

To configure the wildcard admin account:

```
config system admin
    edit "test"
        set remote-auth enable
        set accprofile "super_admin"
        set wildcard enable
        set remote-group "ldap_grp"
    next
end
```

Configuring least privileges for LDAP admin account authentication in Active Directory

An administrator should only have sufficient privileges for their role. In the case of LDAP admin bind, you can configure an admin account in Active Directory for LDAP authentication to allow an admin to perform lookups and reset passwords without being a member of the Account Operators or Domain Administrators built-in groups.

For information about Active Directory, see the product documentation.

To configure account privileges for LDAP authentication in Active Directory:

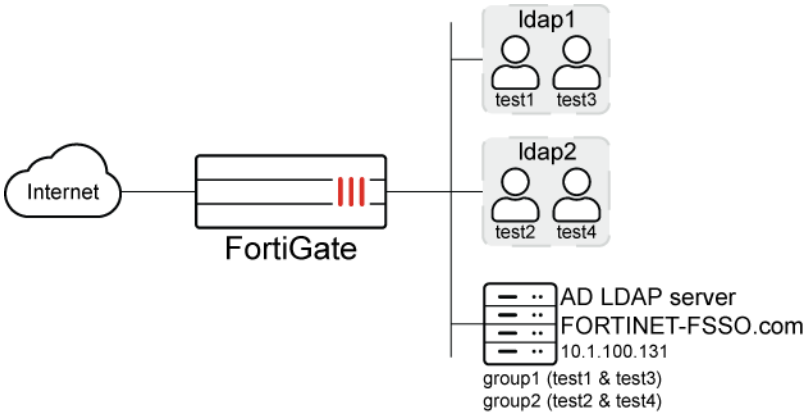
1. In the *Active Directory Users and Computers* administrative console, right-click the Organizational Unit (OU) or the top-level domain you want to configure and select *Delegate Control*.
2. In the *Delegation of Control Wizard* dialog, click *Next*.
3. In the *Users or Groups* dialog, click *Add...* and search Active Directory for the users or groups.
4. Click *OK* and then click *Next*.
5. In the *Tasks to Delegate* dialog, select *Create a custom task to delegate* and click *Next*.
6. Select *Only the following objects in the folder* and scroll to the bottom of the list. Select *User objects* and click *Next*.
7. In the *Permissions* dialog, select *General*.

8. From the *Permissions* list, select the following:
   - *Change password*
   - *Reset password*
9. Clear the *General* checkbox and select *Property-specific*.
10. From the *Permissions* list, select the following:
    - *Write lockoutTime*
    - *Read lockoutTime*
    - *Write pwdLastSet*
    - *Read pwdLastSet*
    - *Write UserAccountControl*
    - *Read UserAccountControl*
11. Click *Next* then click *Finish*.

Tracking users in each Active Directory LDAP group

When LDAP users log on through firewall authentication, the active users per Active Directory LDAP group is counted and displayed in the *Firewall Users* widget and the CLI.

Example



The Active Directory LDAP server, FORTINET-FSSO.com, is configured with two groups that contain two users each: group1 consists of users test1 and test3; group2 consists of users test2 and test4.

To configure AD LDAP user groups in the GUI:

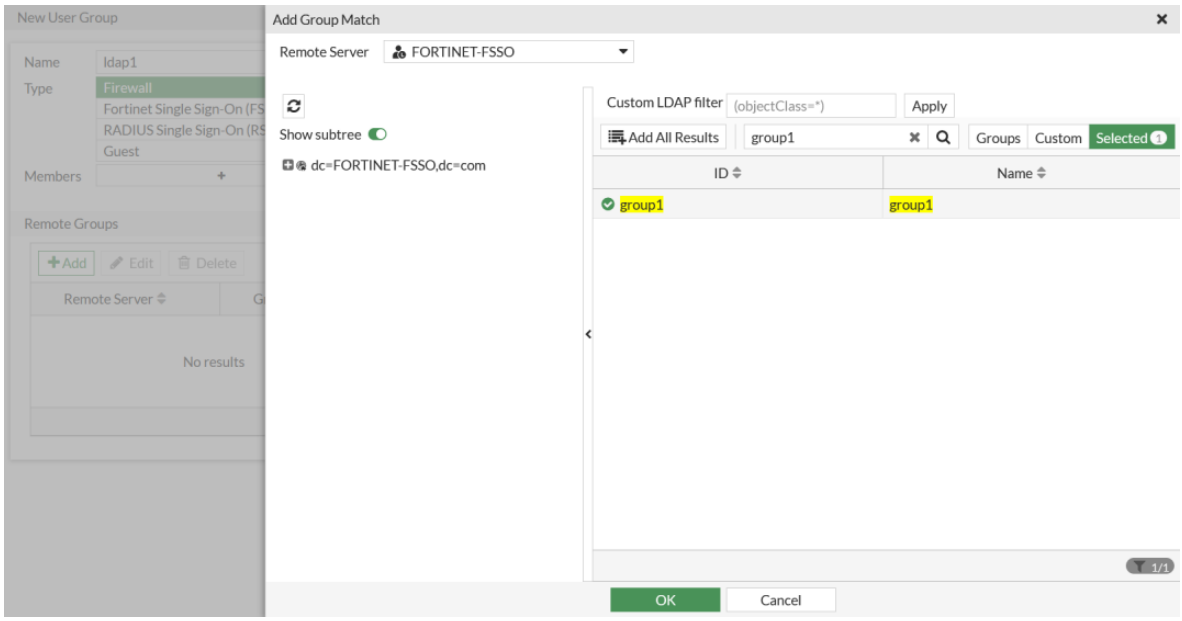
1. Configure the Active Directory LDAP server, FORTINET-FSSO:
   a. Go to *User & Authentication* > *LDAP Servers* and click *Create New*.
   b. Enter the following:

Name	*FORTINET-FSSO*

Server IP/Name	*10.1.100.131*
Distinguished Name	*dc=FORTINET-FSSO,dc=com*
Bind Type	*Regular*
Username	*cn=administrator,cn=users,dc=FORTINET-FSSO,dc=com*
Password	Enter the password.

c. Click *OK*.

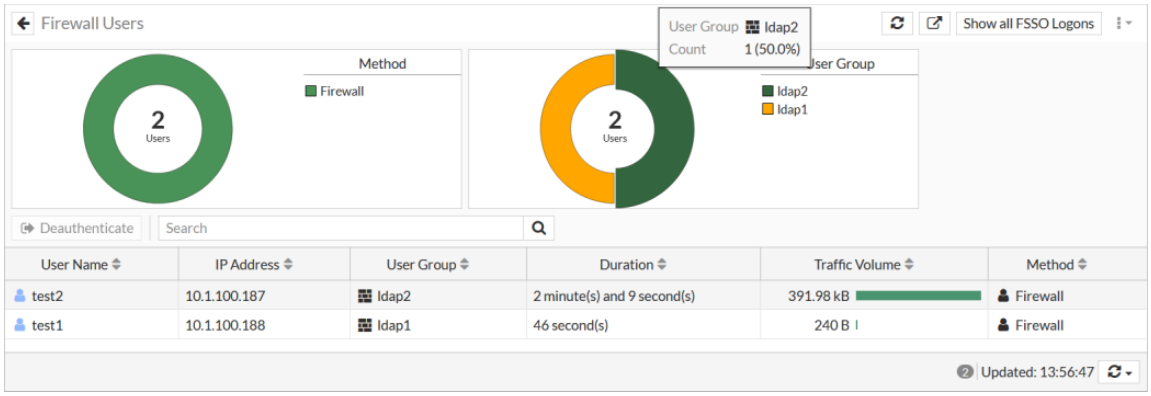
2. Configure the LDAP user groups:
   a. Go to *User & Authentication* > *User Groups* and click *Create New*.
   b. Enter the name, *ldap1*.
   c. In the *Remote Groups* table, click *Add*. The *Add Group Match* pane opens.
   d. For *Remote Server*, select *FORTINET-FSSO*.
   e. In the search box, enter *group1*, and select the result in the table.
   f. Click *OK*.



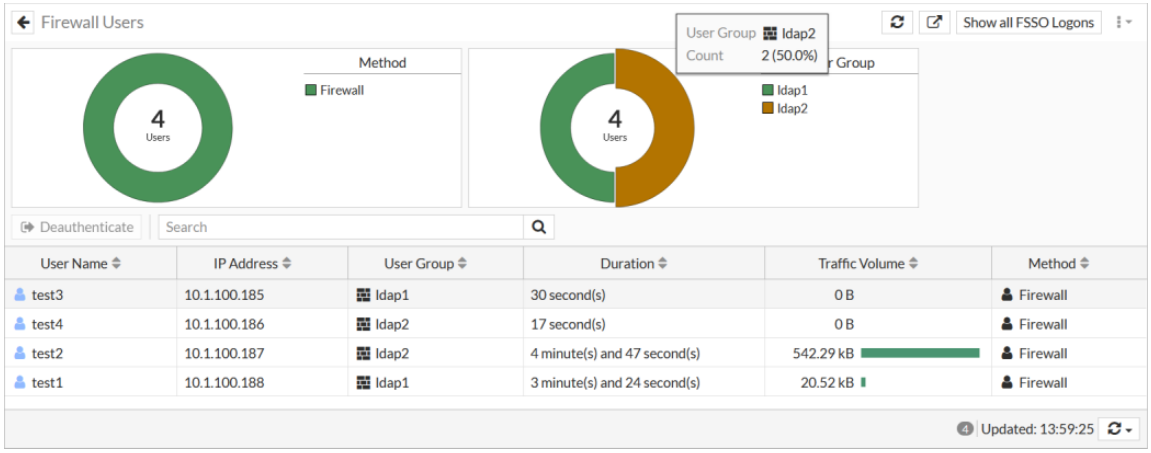
   g. Repeat these steps to configure *ldap2* with the FORTINET-FSSO *group2*.
   h. Click *OK*.
3. Configure a firewall policy with both LDAP groups:
   a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
   b. For *Source*, select *ldap1* and *ldap2*.
   c. Configure the other settings as needed.
   d. Click *OK*.
4. Get users test1 and test2 to log in.
5. In FortiOS, go to *Dashboard* > *Assets & Identities* and click the *Firewall Users* widget to expand to full screen view.

   Hover over a group in the *User Group* donut chart to view how many users are logged on from that group, and the number of users as a percentage of all logged on users. The chart shows that two users are logged

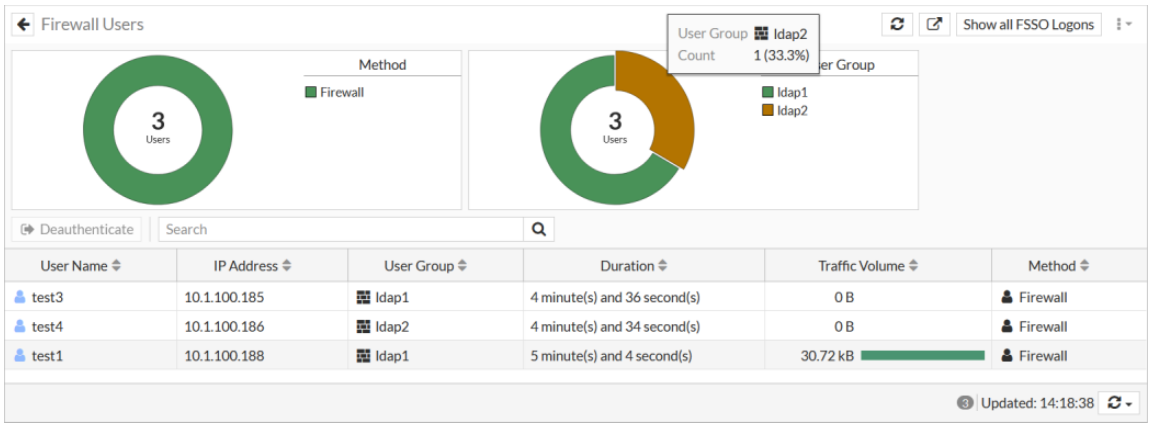
in.



6. Get users test3 and test4 to log in, and refresh the *Firewall Users* widget. Each LDAP group has two users logged in, with a total of four active users.



7. Get user test2 to log out, and refresh the *Firewall Users* widget. There is a total of three active users, and the ldap2 group only has one user that is logged in.



To verify the user group count in the CLI:

```
# diagnose user-device-store user-count list <integer>
# diagnose user-device-store user-count query <FQDN of AD group>
```

Tracking rolling historical records of LDAP user logins

Authenticated LDAP users can be tracked by logging the users' group memberships, logon timestamps, and logout timestamps into local files on a log disk over a rolling four-week period. The historical records can be queried from the CLI. This feature is only enabled on FortiGate models with a log disk.

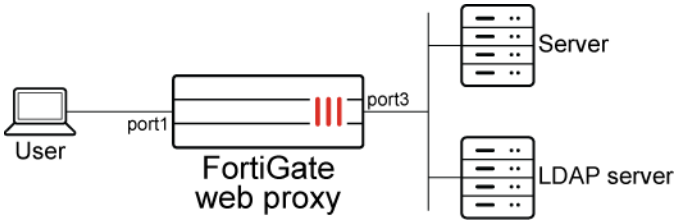
To view active user logged information:

```
# diagnose user-device-store user-stats query <yyyy-mm-dd> <range_in_days>
```

Example

In this example, the FortiGate is configured with an explicit web proxy and an LDAP server. When an LDAP user is authenticated by an IP-based authentication method in WAD, the WAD user is considered to be in an active logon status. This WAD user is listed in the `diagnose wad user list` output. If the user is removed from WAD as an authenticated, such as when the IP-based authentication expires, then the user is considered to become inactive (logout status). The user is no longer listed in the `diagnose wad user list` output.

The WAD user's group membership information and their logon and logout timestamps are written into local files on the FortiGate's disk. There is one log file for each day, and the FortiGate can maintain up to 28 log files over a rolling period of 28 days (four weeks). This means after 28 days with 28 files stored, on the 29th day, the first file will be removed and a new file will be created for the 29th day.



This feature works on other configurations such as firewall authentication, transparent web proxy, ZTNA, and Agentless VPN where an LDAP server is used.

To configure the FortiGate:

1. Enable the explicit web proxy on port1:

```
config system interface
    edit "port1"
        set explicit-web-proxy enable
        set explicit-ftp-proxy enable
        set snmp-index 3
    next
end
```

2. Configure the LDAP server:

```
config user ldap
    edit "ldap-test"
        set server "172.16.200.98"
        set cnid "cn"
        set dn "dc=fortinetqa,dc=local"
        set type regular
        set username "CN=root,CN=Users,DC=fortinetqa,DC=local"
        set password **********
    next
end
```

3. Configure the authentication scheme:

```
config authentication scheme
    edit "basic-ldap"
        set method basic
        set user-database "ldap-test"
    next
end
```

4. Configure the authentication rule:

```
config authentication rule
    edit "basic-ldap"
        set srcaddr "all"
        set active-auth-method "basic-ldap"
        set web-portal disable
    next
end
```

5. Configure the user group:

```
config user group
    edit "ldap-group"
        set member "ldap" "ldap-test"
    next
end
```

6. Configure the proxy policy:

```
config firewall proxy-policy
    edit 1
        set proxy explicit-web
        set dstintf "port3"
        set srcaddr "all"
        set dstaddr "all"
        set service "web"
        set action accept
        set schedule "always"
        set groups "ldap-group"
        set utm-status enable
        set ssl-ssh-profile "deep-custom"
```

```
        set av-profile "av"
    next
end
```

When users pass through the explicit proxy and log in and out through LDAP, their login and logout records will be logged to the disk.

In this example, there are two LDAP users, test1 and test3, with the following activity:

1. test3 logs on at 22:30:22 on February 23, 2022, then logs out at 22:31:09 on the same day.
2. test1 logs on at 23:55:02 on February 23, 2022, then logs out at 00:05:02 on February 24, 2022.
3. test3 logs on at 16:29:44 on February 24, 2022, then logs out at 16:39:44 on the same day.

The logon and logout timestamp information, and the group membership information for users test1 and test3 will be logged into two local files on the log disk.

To view the active user logged information for two days back from February 24, 2022:

```
# diagnose user-device-store user-stats query 2022-02-24 2

Record #0:
        'username' = 'test3'
        'groupname' = 'CN=Domain Admins,CN=Users,DC=FORTINETQA,DC=local'
        'groupname' = 'CN=FSSO,OU=QA,DC=FORTINETQA,DC=local'
        'logon' = '2022-02-23 22:30:22'
        'logout' = '2022-02-23 22:31:09'

Record #1:
        'username' = 'test1'
        'groupname' = 'CN=Domain Admins,CN=Users,DC=FORTINETQA,DC=local'
        'groupname' = 'CN=FSSO,OU=QA,DC=FORTINETQA,DC=local'
        'groupname' = 'CN=mytest-grp,OU=QA,DC=FORTINETQA,DC=local'
        'logon' = '2022-02-23 23:55:02'

Record #2:
        'username' = 'test1'
        'groupname' = 'CN=Domain Admins,CN=Users,DC=FORTINETQA,DC=local'
        'groupname' = 'CN=FSSO,OU=QA,DC=FORTINETQA,DC=local'
        'groupname' = 'CN=mytest-grp,OU=QA,DC=FORTINETQA,DC=local'
        'logon' = '2022-02-23 23:55:02'
        'logout' = '2022-02-24 00:05:02'

Record #3:
        'username' = 'test3'
        'groupname' = 'CN=Domain Admins,CN=Users,DC=FORTINETQA,DC=local'
        'groupname' = 'CN=FSSO,OU=QA,DC=FORTINETQA,DC=local'
        'logon' = '2022-02-24 16:29:44'
        'logout' = '2022-02-24 16:39:44'

Returned 4 records.
```

There is one record (`logon`) for test1 on `2022-02-23` because they remained active after midnight (until 00:05:02). There is another record for `2022-02-24` with logon and logout timestamps for test1.

Configuring client certificate authentication on the LDAP server

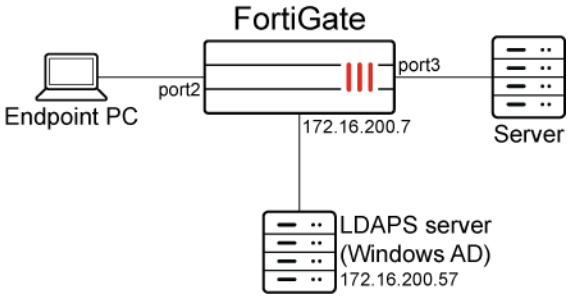
Administrators can configure a FortiGate client certificate in the LDAP server configuration when the FortiGate connects to an LDAPS server that requires client certificate authentication.

```
config user ldap
    edit <ldap_server>
        set client-cert-auth {enable | disable}
        set client-cert <source>
    next
end
```

Example

In this example, the FortiGate is configured as an explicit web proxy. It connects to the Windows AD server through LDAPS, where the Windows server requires a client certificate to connect. The client certificate is configured in the CLI.

The endpoint PC connecting to the web server will first need to authenticate to the explicit web proxy before accessing the server.



While this example demonstrates an LDAP client certificate for an explicit proxy configuration, LDAP client certificates can be used in firewall authentication, transparent proxy, ZTNA, and where ever LDAP configurations are used on the FortiGate.

To configure a client certificate on the LDAP server:

1. Enable the explicit web proxy on port2:

```
config system interface
    edit "port2"
        set explicit-web-proxy enable
    next
end
```

2. Upload the client certificate to the FortiGate:

```
config vpn certificate local
    edit "Zach"
        set password **********
        set private-key <private key>
        set certificate <certificate>
    next
end
```

3. Configure the LDAP server settings:

```
config user ldap
    edit "ldaps"
        set server "172.16.200.57"
        set server-identity-check disable
        set cnid "CN"
        set dn "CN=Users,DC=ftnt,DC=com"
        set secure ldaps
        set port 636
        set client-cert-auth enable
        set client-cert "Zach"
    next
end
```

To ensure the proper operation of LDAPS authentication, make sure that the CA that signed the server certificate for the LDAPS server, typically a private CA, is trusted by the FortiGate by uploading the private CA certificate. This is especially important for an existing configuration where this was not previously done to ensure proper operation after upgrades. See CA certificate on page 3967.

4. Configure the authentication scheme:

```
config authentication scheme
    edit "1"
        set method basic
        set user-database "ldaps"
    next
end
```

5. Configure the authentication rule:

```
config authentication rule
    edit "1"
        set srcintf "port2"
        set srcaddr "all"
        set dstaddr "all"
        set active-auth-method "1"
    next
end
```

6. Configure the user group:

```
config user group
    edit "test"
        set member "ldaps"
    next
end
```

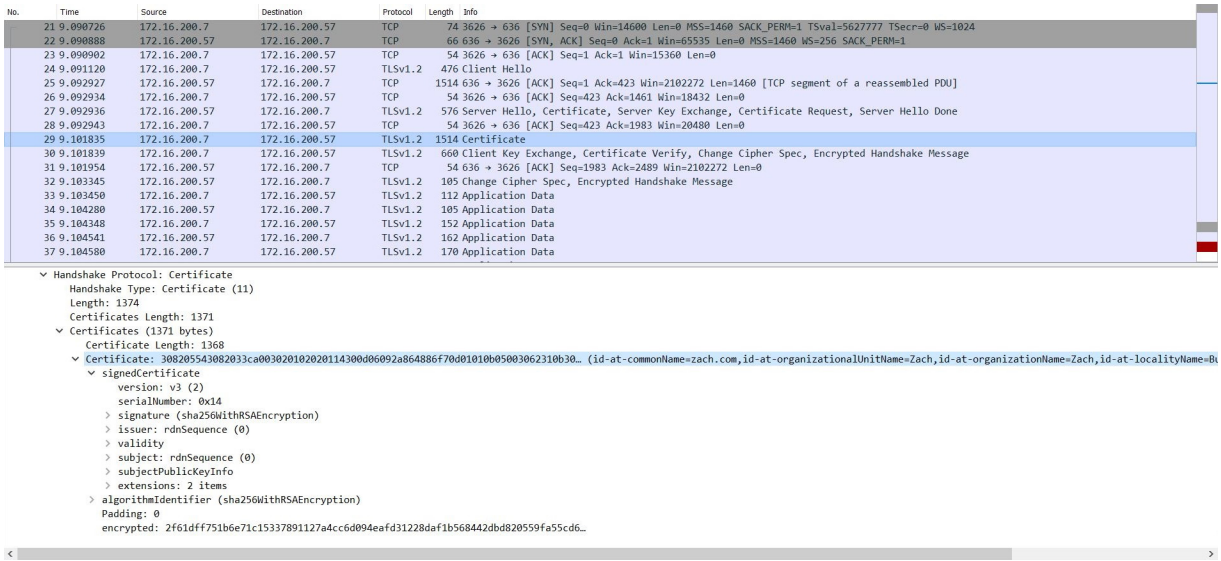
7. Configure the proxy policy with the user group:

```
config firewall proxy-policy
    edit 1
        set proxy explicit-web
        set dstintf "port3"
        set srcaddr "all"
        set dstaddr "all"
        set service "webproxy"
        set action accept
        set schedule "always"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set groups "test"
        set utm-status enable
        set ssl-ssh-profile "deep-inspection-clone"
        set av-profile "av"
    next
end
```

Testing and verification

When traffic from the endpoint PC matches a policy and triggers authentication, the FortiGate starts the LDAPS TLS connection handshake with the Windows AD. The LDAPS server requests a client certificate to identify the FortiGate as a client. The FortiGate provides a configured client certificate, issued to zach.com, to the LDAPS server.

The following communication between the FortiGate and the LDAPS server shows the client certificate is sent by the FortiGate:



RADIUS servers

Remote Authentication and Dial-In User Service (RADIUS) is a broadly supported client-server protocol that provides centralized authentication, authorization, and accounting functions. RADIUS clients are built into gateways that allow access to networks such a VPN server, network access server (NAS), and a network switch or firewall that uses authentication.

RADIUS servers use UDP packets to communicate with the RADIUS clients on the network to authenticate users before allowing them access to the network, authorize access to resources by appropriate users, and account or bill for those resources that are used. RADIUS servers are currently defined by RFC 2865 (RADIUS) and RFC 2866 (RADIUS Accounting), and listen on either UDP ports 1812 (authentication) and 1813 (accounting), or ports 1645 (authentication) and 1646 (accounting) requests. RADIUS servers exist for all major operating systems.

The RADIUS server must be configured to accept the FortiGate as a client so it can use the authentication and accounting functions of the RADIUS server.

> ⚠ To secure RADIUS connections, consider using RADSEC over TLS. See Configuring a RADSEC client on page 3566.

RADIUS authentication with a FortiGate requires the following:

- Configuring one or more RADIUS server profiles on the FortiGate.
- Assigning the RADIUS server profile to a user or user group.
- Applying the user or user group to a firewall policy.

RADIUS authentication can be applied to many FortiGate functions, such as firewall authentication, SSL and IPsec VPNs, administrator profiles, ZTNA, explicit proxy, wireless, 802.1X, and more.

The RADIUS server uses a shared secret key with MD5 hashing to encrypt information passed between RADIUS servers and clients. Typically, only user credentials are encrypted. Additional security can be configured through IPsec tunnels by placing the RADIUS server behind another VPN gateway.

The following topics provide more information about RADIUS servers:

- Configuring a RADIUS server on page 3528
- Using multiple RADIUS servers on page 3531
- RADIUS AVPs and VSAs on page 3534
- RADIUS VSAs for captive portal redirects on page 3536
- Restricting RADIUS user groups to match selective users on the RADIUS server on page 3538
- Configuring RADIUS SSO authentication on page 3539
- Trigger RADIUS authentication with DNS and ICMP queries on page 3545
- RSA ACE (SecurID) servers on page 3548
- Support for Okta RADIUS attributes filter-Id and class on page 3552
- Sending multiple RADIUS attribute values in a single RADIUS Access-Request on page 3554
- Traffic shaping based on dynamic RADIUS VSAs on page 3555
- RADIUS Termination-Action AVP in wired and wireless scenarios on page 3562
- Configuring a RADSEC client on page 3566
- RADIUS integrated certificate authentication for Agentless VPN on page 3571

Configuring a RADIUS server

A RADIUS server can be configured in the GUI by going to *User & Authentication* > *RADIUS Servers*, or in the CLI under `config user radius`.

Basic configuration

The following table summarizes the common RADIUS settings that can be configured in the GUI and CLI.

GUI field	CLI setting	Description
Name	`edit <name>`	Define the RADIUS server object within FortiOS.
Authentication method	`set auth-type {auto | ms_chap_v2 | ms_chap | chap | pap}`	Specify the authentication method, or select *Default*/`auto` to negotiate PAP, MSCHAP_v2, and CHAP in that order.
NAS IP	`set nas-ip <IPv4_address>`	Optional setting, also known as Calling-Station-Id. Specify the IP address the FortiGate uses to communicate with the RADIUS server. If left unconfigured, the FortiGate will use the IP address of the interface that communicates with the RADIUS server.
Include in every user group	`set all-usergroup {enable | disable}`	Optional setting to add the RADIUS server to each user group. This allows each user group to try and authenticate users against the RADIUS server if local authentication fails.
Primary Server		
IP/Name	`set server <string>`	Enter the IP address or resolvable FQDN of the RADIUS server.
Secret	`set secret <password>`	Enter the password used to connect to the RADIUS server.

There is an option in the GUI to configure a second server, and a third server can be configured in the CLI (see Using multiple RADIUS servers on page 3531).

Advanced settings

Advanced settings for RADIUS servers can be configured in the CLI. The following are some commonly used settings.

To edit the port used to connect with the RADIUS server:

```
config system global
    set radius-port <integer>
end
```

To edit the default setting for password encoding and username case sensitivity:

```
config user radius
    edit <name>
        set password-encoding {auto | ISO-8859-1}
        set username-case-sensitive {enable | disable}
    next
end
```

Command	Description
`password-encoding {auto | ISO-8859-1}`	Set the password encoding to use the original encoding or ISO-8859-1 (default = auto). The `auth-type` must be `auto` or `pap` to change this setting.
`username-case-sensitive {enable | disable}`	Enable/disable case sensitive usernames (default = disable).

To configure different transport protocols:

```
config user radius
    edit <name>
        set transport-protocol {udp | tcp | tls}
    next
end
```

Command	Description
`transport-protocol {udp | tcp | tls}`	Set the type of transport protocol to use: • `udp`: use UDP (default) • `tcp`: use TCP, but no TLS security • `tls`: use TLS over TCP

To configure a RADSEC client with TLS:

```
config user radius
    edit <name>
        set transport-protocol tls
        set ca-cert <string>
        set client-cert <string>
        set tls-min-proto-version {default | SSLv3 | TLSv1 | TLSv1-1 | TLSv1-2}
        set server-identity-check {enable | disable}
    next
end
```

Command	Description
`ca-cert <string>`	Set the CA certificate of server to trust under TLS.
`client-cert <string>`	Set the client certificate to use under TLS.
`tls-min-proto-version {default | SSLv3 | TLSv1 | TLSv1-1 | TLSv1-2}`	Set the minimum supported protocol version for TLS connections: • `default`: follow the system global setting • `SSLv3`: use SSLv3 • `TLSv1`: use TLSv1 • `TLSv1-1`: use TLSv1.1 • `TLSv1-2`: use TLSv1.2
`server-identity-check {enable | disable}`	Enable/disable RADIUS server identity check, which verifies the server domain name/IP address against the server certificate (default = enable).

For RADSEC over TLS example configuration, see Configuring a RADSEC client on page 3566.

It is best practice to enable RadSec over TLS whenever the FortiGate and RADIUS connection must pass through unencrypted transport. When using TCP and UDP transport modes, it is recommended to ensure the FortiGate and RADIUS connection passes through a trusted network or the connection passes through an encrypted tunnel over untrusted networks.

To account for dynamic IP address changes, such as those governed by SD-WAN rules, interface names can be used to define the source IP addresses in RADIUS, LDAP, and DNS configurations using the `source-ip-interface` command. See Implement the interface name as the source IP address in RADIUS, LDAP, and DNS configurations on page 378.

RADIUS Connection

When using TCP or UDP as transport, it is possible for the RADIUS protocol to be compromised by the vulnerability described in CVE-2024-3596. In order to protect against this RADIUS vulnerability, as a RADIUS client, FortiGate will:

1. Force the validation of message authenticator.
2. Reject RADIUS response with unrecognized `proxy-state` attribute.

Message authenticator checking is made mandatory under UDP/TCP. It is not mandatory when using TLS.

Therefore, if FortiGate is using UDP/TCP mode without RADSEC, the RADIUS server should be patched to ensure the message authenticator attribute is used in its RADIUS messages. Check with your RADIUS server vendor for information about support for the message authenticator attribute.

In the case that the RADIUS server cannot be immediately patched, you may optionally disable message authenticator checking from the CLI under your RADIUS server configurations:

```
config user radius
    edit <server>
        set require-message-authenticator disable
```

```
    next
end
```

The `require-message-authenticator` setting is enable by default.

> ⚠ It is best practice to enable RadSec over TLS whenever the FortiGate and RADIUS connection must pass through unencrypted transport. When using TCP and UDP transport modes, it is recommended to ensure the FortiGate and RADIUS connection passes through a trusted network or the connection passes through an encrypted tunnel over untrusted networks.

Using multiple RADIUS servers

There are several ways to implement multiple RADIUS servers, and each has a different effect on user authentication. The three main options available are:

- Add a second (or third) RADIUS server in the same profile.
- Add a second RADIUS server profile, and add both to the same user group.
- Use two RADIUS server profiles for two user groups (one for each).

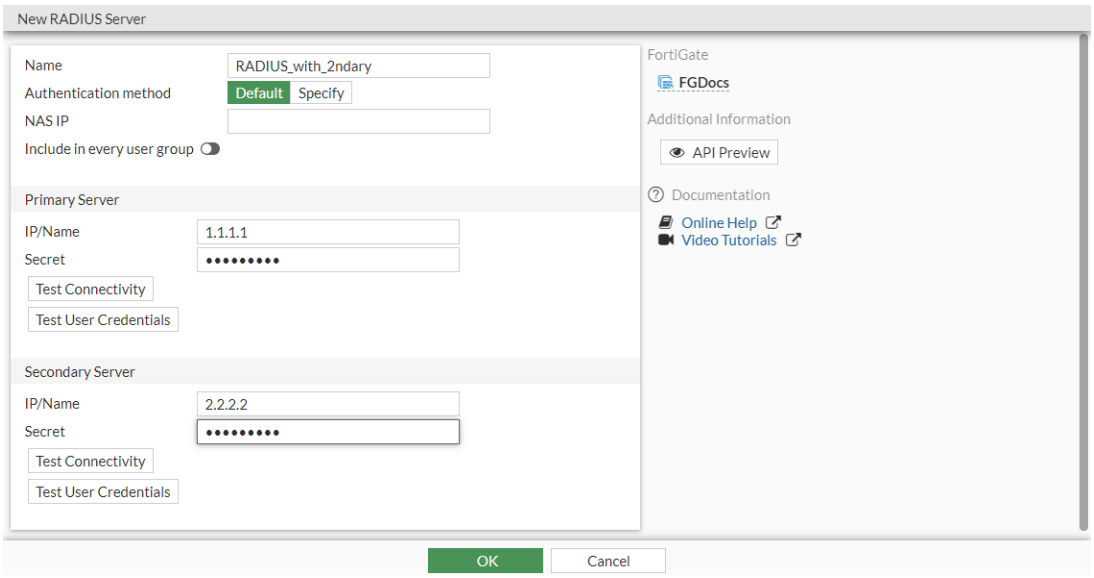
Adding a second server in a RADIUS profile

A second RADIUS server can be configured in the same RADIUS profile so in the event the first RADIUS server does not respond, the second server can be checked. If the first RADIUS server responds with an Access-Reject, no further servers are queried.

To add a second server in a RADIUS profile:

1. Go to *User & Authentication* > *RADIUS Servers* and click *Create New*.
2. Enter the following:

Name	*RADIUS_with_2ndary*
Authentication method	*Default*
Primary Server	
IP/Name	*1.1.1.1*
Secret	Enter the password used to connect to the RADIUS server.
Secondary Server	
IP/Name	*2.2.2.2*
Secret	Enter the password used to connect to the RADIUS server.



3. Click *OK*.

Adding two RADIUS server profiles in the same user group

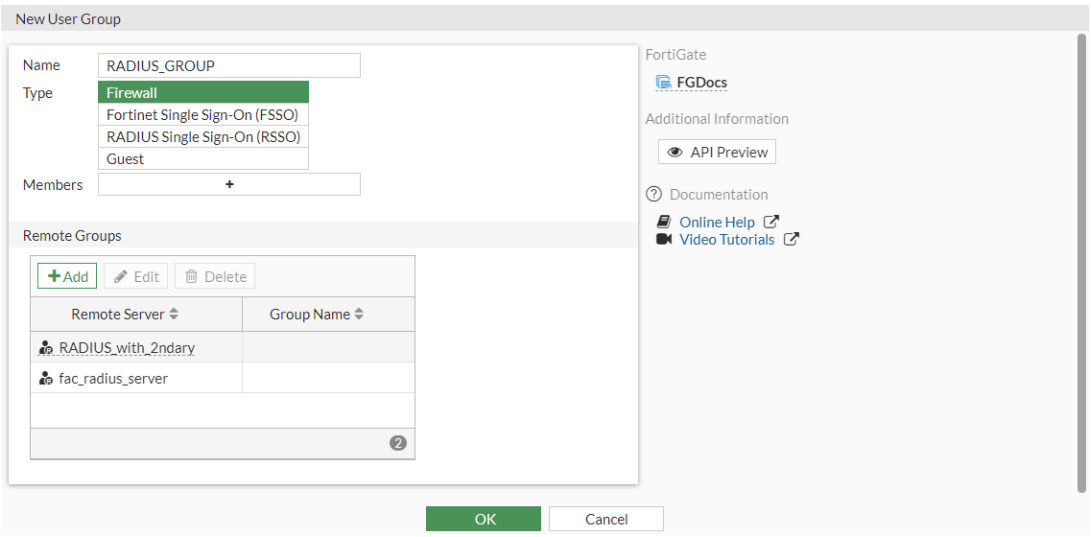
When two separate RADIUS profiles are added to a user group, the FortiGate sends an Access-Request simultaneously to both RADIUS servers, and authentication succeeds if either server sends back an Access-Accept. This example includes the settings from the previous example where one or more of the RADIUS server profiles has a secondary server configured.In this case, the secondary server in the *RADIUS_with_2ndary* profile, 2.2.2.2, is only checked if the primary server of this profile times out and the *fac_radius_server* profile does not return an Access-Accept.

To add two RADIUS server profiles in the same user group:

1. Go to *User & Authentication* > *RADIUS Servers*, click *Create New*, and configure the RADIUS servers as needed (refer to the previous example).
2. Go to *User & Authentication* > *User Groups* and click *Create New*.
3. Enter the following:

Name	*RADIUS_GROUP*
Type	*Firewall*

4. In the *Remote Groups* table, click *Add*.
5. Select *RADIUS_with_2ndary* and click *OK*.
6. Click *Add*, select *fac_radius_server*, then click *OK*.



7. Click *OK*.

Using separate RADIUS server profiles for separate user groups

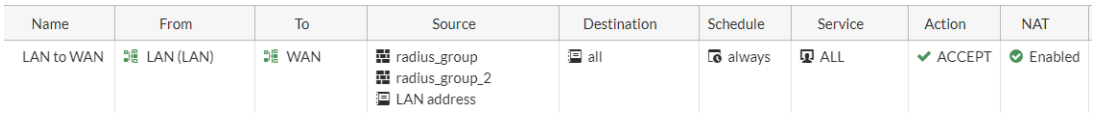
In this example, the FortiGate first evaluates if the user belongs to the first listed group (*radius_group*) in the policy. If the user fails to authenticate to this group, then the FortiGate checks if the user can successfully authenticate to the second user group (*radius_group_2*). Refer to the first and second examples for detailed instructions.

To use separate RADIUS server profiles for separate user groups:

1. Configure the RADIUS server profiles:
   a. Go to *User & Authentication* > *RADIUS Servers* and click *Create New*.
   b. Configure two RADIUS servers, *fac_radius_server* and *RADIUS_with_2ndary*, as needed (refer to the previous example).
2. Configure the firewall groups:
   a. Go to *User & Authentication* > *User Groups* and click *Create New*.
   b. Configure two firewall groups, one named *radius_group* with remote server member *fac_radius_server*, and one named *radius_group_2* with remote server member *RADIUS_with_2ndary* (refer to the previous example).

Group Name	Group Type	Members	Ref.
radius_group	Firewall	fac_radius_server	1
radius_group_2	Firewall	RADIUS_with_2ndary	1

3. Configure the firewall policy:
   a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
   b. For *Source*, click *User* then select *radius_group* and *radius_group_2*. Click *Address* and select *LAN address*.
   c. Configure the other settings as needed.
   d. Click *OK*.



Name	From	To	Source	Destination	Schedule	Service	Action	NAT
LAN to WAN	LAN (LAN)	WAN	radius_group radius_group_2 LAN address	all	always	ALL	ACCEPT	Enabled

RADIUS AVPs and VSAs

This topic describes RADIUS Attribute Value Pairs (AVPs) and Vendor-Specific Attributes (VSAs).

AVPs

RADIUS packets include a set of AVPs to identify information about the user, their location, and other information. The IETF defined a set of 255 standard attributes, which are well known and come in the form of Type, Length, Value (for more details, refer to RFC 2865). Of the standard 255, the FortiGate sends the following RADIUS attributes:

RADIUS attribute number	Name	Description
1	User-Name	Name of the user being authenticated by the RADIUS server.
4	NAS-IP-Address	IP address of the network access server (NAS) that is requesting authentication. The NAS is the FortiGate.
8	Framed-IP-Address	IP address to be configured for the user, by sending the IP address of a user to the RADIUS server in the Access-Request packet.
25	Class	Used in accounting packets and requests for firewall, WiFi, and proxy authentication. The attribute is returned in the Access-Accept message and is added to all accounting packets.
26	Fortinet-VSA	See VSAs.
32	NAS-Identifier	Identifier or IP address of the NAS that is requesting authentication. The NAS is the FortiGate.
42	Acct-Input-Octets	Number of octets received from the port over the course of this service being provided. Used to charge the user for the amount of traffic they used.
43	Acct-Output-Octets	Number of octets sent to the port while delivering this service. Used to charge the user for the amount of traffic they used.
44	Acct-Session-Id	Unique number assigned to each start and stop record to make it easy to match them, and to eliminate duplicate records.
55	Event-Timestamp	Records the time that the event occurred on the NAS. The timestamp is measured in seconds since January 1, 1970 00:00 UTC. Before the Event-Timestamp attribute can be sent in a packet, make sure that the correct time is set on the FortiGate.

VSAs

Some vendors want or need to send attributes that do not match any of the defined IETF attributes. This can be accomplished by using RADIUS attribute type 26, which allows a vendor to encapsulate their own specific attributes in this standard AVP.

In order to support VSAs, the RADIUS server requires a dictionary to define the VSAs. This dictionary is typically supplied by the client or server vendor.

The Fortinet RADIUS vendor ID is 12356 and contains the following attributes:

Attribute name	Attribute number	Attribute value format
Fortinet-Group-Name	1	String
Fortinet-Client-IP-Address	2	IP address
Fortinet-Vdom-Name*	3	String
Fortinet-Client-IPv6-Address	4	Octets
Fortinet-Interface-Name	5	String
Fortinet-Access-Profile	6	String
Fortinet-SSID	7	String
Fortinet-AP-Name	8	String
Fortinet-FAC-Auth-Status	11	String
Fortinet-FAC-Token-ID	12	String
Fortinet-FAC-Challenge-Code	15	String
Fortinet-Webfilter-Category-Allow	16	String
Fortinet-Webfilter-Category-Block	17	Octets
Fortinet-Webfilter-Category-Monitor	18	Octets
Fortinet-AppCtrl-Category-Allow	19	Octets
Fortinet-AppCtrl-Category-Block	20	Octets
Fortinet-AppCtrl-Risk-Allow	21	Octets
Fortinet-AppCtrl-Risk-Block	22	Octets
Fortinet-WirelessController-Device-MAC	23	Ether
Fortinet-WirelessController-WTP-ID	24	String
Fortinet-WirelessController-Assoc-Time	25	Date
Fortinet-FortiWAN-AVPair	26	String
Fortinet-Captive-Portal-URL	27	String
Fortinet-FDD-Access-Profile	30	String

Attribute name	Attribute number	Attribute value format
Fortinet-FDD-Trusted-Hosts	31	String
Fortinet-FDD-SPP-Name	32	String
Fortinet-FDD-Is-System-Admin	33	String
Fortinet-FDD-Is-SPP-Admin	34	String
Fortinet-FDD-SPP-Policy-Group	35	String
Fortinet-FDD-Allow-API-Access	36	String
Fortinet-Fpc-User-Role	40	String
Fortinet-Tenant-Identification	41	String
Fortinet-Host-Port-AVPair	42	String
Fortinet-PreShared-Key-Name	43	String
Fortinet-AS-Credential-Hash	44	String

* For Fortinet-Vdom-Name, users can be tied to a specific VDOM on the FortiGate. Refer to the documentation provided by your RADIUS server for configuration details.

RADIUS VSAs for captive portal redirects

RADIUS Vendor-Specific Attributes (VSA) for captive portal redirects provide a smoother user experience during captive portal redirects, especially in environments where vendor-specific attributes are heavily used, such as corporate networks or public Wi-Fi hotspots.

To configure RADIUS VSA for captive portal redirects:

1. Configure a RADIUS user:

```
config user radius
    edit "pc05"
        set server "172.16.200.55"
        set secret ***************
    next
end
```

2. Add the user to a group:

```
config user group
    edit "radius-group"
        set member "pc05"
    next
end
```

3. Configure the interface to use captive portal authentication and the group:

```
config system interface
    edit "port2"
        set security-mode captive-portal
        set security-groups "radius-group"
    next
end
```

4. Configure the firewall policy:

```
config firewall policy
    edit 1
        set name "1"
        set srcintf "port2"
        set dstintf "mgmt"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set logtraffic all
        set nat enable
    next
end
```

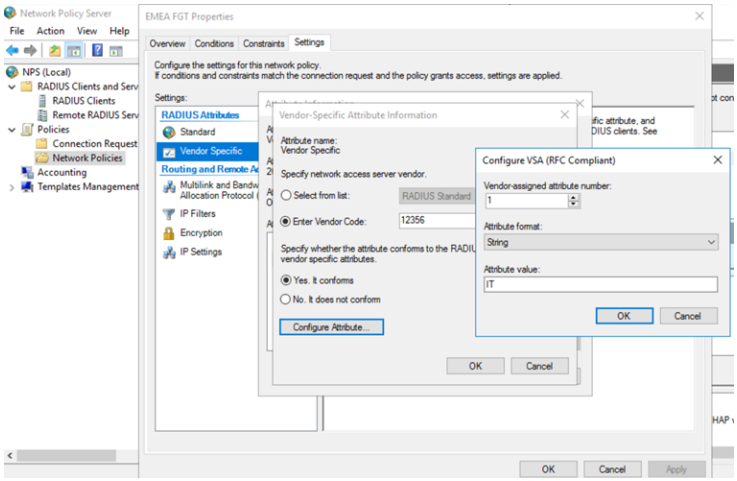
5. To check the configuration, on a client PC:
   a. Use a browser to access a web server.
   b. Authenticate using RADIUS.
   c. Browse the redirect to https://www.fortinet.com.
   d. Check the list of authenticated users:

```
# diagnose firewall auth list

10.1.100.11, 962341
        src_mac: 00:0c:29:61:d4:13
        type: fw, id: 0, duration: 752, idled: 2
        expire: 298, allow-idle: 300
        flag(10): radius
        server: pc05
        packets: in 8531 out 7654, bytes: in 7972540 out 1104574
        group_id: 1
        group_name: radius-group

----- 1 listed, 0 filtered ------
```

Restricting RADIUS user groups to match selective users on the RADIUS server

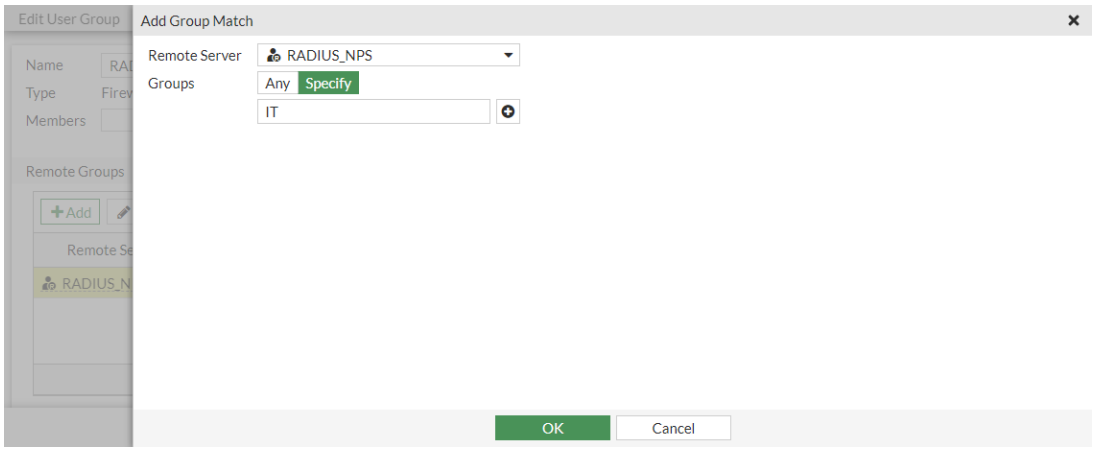
When a user group is configured in FortiOS to authenticate against a RADIUS server, it will allow any valid user account on the RADIUS server to match that user group. Sometimes you might want to specify which users on the RADIUS server should match a particular user group on the FortiGate. This can be accomplished using the RADIUS attribute value pair (AVP) 26, known as a Vendor-Specific Attribute (VSA). This attribute allows the Fortinet-Group-Name VSA to be included in the RADIUS response. In FortiOS, the user group must be configured to specifically match this group.

In the following example, a RADIUS Network Policy Server (NPS) has been configured to have the Fortinet-Group-Name be *IT*, and assumes that the user group, *RADIUS_IT* has been created, which authenticates to the *RADIUS_NPS* server.



To configure specific group matching in the GUI:

1. Go to *User & Authentication* > *User Groups* and edit the *RADIUS_IT* group.
2. In the *Remote Groups* table, select the *RADIUS_NPS* server and click *Edit*. The *Add Group Match* pane opens.
3. For *Groups*, select *Specify* and enter the group name configured on the RADIUS server (*IT*).
4. Click *OK*.



5. Click *OK*.

To configure specific group matching in the CLI:

```
config user group
    edit "RADIUS_IT"
        set member "RADIUS_NPS"
        config match
            edit 1
                set server-name "RADIUS_NPS"
                set group-name "IT"
            next
        end
    next
end
```

To change the matching back to any group, under `config match`, enter `delete 1`. Changing the `group-name` to `"Any"` will cause the FortiGate to match the Fortinet-Group-Name with the literal string, `Any`.

Configuring RADIUS SSO authentication

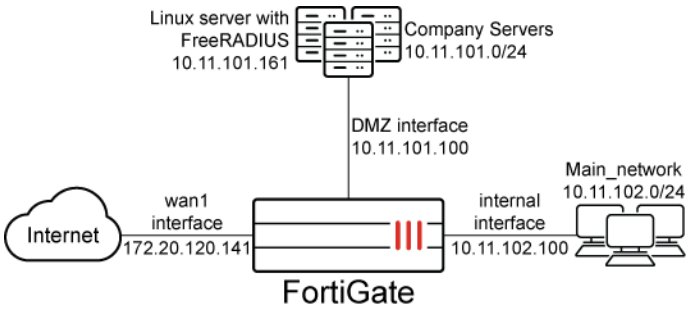
A common RADIUS SSO (RSSO) topology involves a medium-sized company network of users connecting to the Internet through the FortiGate and authenticating with a RADIUS server. The following describes how to configure FortiOS for this scenario. The example makes the following assumptions:

- VDOMs are not enabled.
- The super_admin account is used for all FortiGate configuration.
- A RADIUS server is installed on a server or FortiAuthenticator and uses default attributes.
- BGP is used for any dynamic routing.
- You have configured authentication event logging under *Log & Report*.

Example.com has an office with 20 users on the internal network who need access to the Internet. The office network is protected by a FortiGate-60C with access to the Internet through the wan1 interface, the user

network on the internal interface, and all servers are on the DMZ interface. This includes an Ubuntu sever running FreeRADIUS. This example configures two users:

User	Account
Pat Lee	plee@example.com
Kelly Green	kgreen@example.com



Configuring this example consists of the following steps:

1. Configure RADIUS.
2. Configure FortiGate interfaces.
3. Configure a RSSO agent.
4. Create a RSSO user group.
5. Configure security policies.
6. Test the configuration.

To configure RADIUS:

Configuring RADIUS includes configuring a RADIUS server such as FreeRADIUS on user's computers and configuring users in the system. In this example, Pat and Kelly belong to the exampledotcom_employees group. After completing the configuration, you must start the RADIUS daemon. The users have a RADIUS client installed on their PCs that allow them to authenticate through the RADIUS server.

For any problems installing FreeRADIUS, see the FreeRADIUS documentation.

To configure FortiGate interfaces:

You must define a DHCP server for the internal network, as this network type typically uses DHCP. The wan1 and dmz interfaces are assigned static IP addresses and do not need a DHCP server. The following table shows the FortiGate interfaces used in this example:

Interface	Subnet	Act as DHCP server	Devices
wan1	172.20.120.141	No	Internet service provider
dmz	10.11.101.100	No	Servers including RADIUS server
internal	10.11.102.100	Yes: x.x.x.110-250	Internal user network

1. Go to *Network* > *Interfaces*.
2. Edit wan1:

Alias	Internet
Addressing Mode	Manual
IP/Network Mask	172.20.120.141/255.255.255.0
Administrative Access	HTTPS, SSH
Enable DHCP Server	Not selected
Comments	Internet
Administrative Status	Up

3. Click *OK*.
4. Edit dmz:

Alias	Servers
Addressing Mode	Manual
IP/Network Mask	10.11.101.100/255.255.255.0
Administrative Access	HTTPS, SSH, PING, SNMP
Enable DHCP Server	Not selected
Listen for RADIUS Accounting Messages	Select
Comments	Servers
Administrative Status	Up

5. Click *OK*.
6. Edit internal:

Alias	Internal network
Addressing Mode	Manual
IP/Network Mask	10.11.102.100/255.255.255.0
Administrative Access	HTTPS, SSH, PING
Enable DHCP Server	Select
Address Range	10.11.102.110 - 10.11.102.250
Netmask	255.255.255.0
Default Gateway	Same as Interface IP
Comments	Internal network
Administrative Status	Up

To create a RADIUS SSO agent:

1. Go to *Security Fabric* > *External Connectors*.
2. Click *Create New*.
3. Under *Endpoint/Identity*, select *RADIUS Single Sign-On Agent*.
4. Enable *Use RADIUS Shared Secret*. Enter the RADIUS server's shared secret.
5. Enable *Send RADIUS Responses*. Click *OK*.

To create a RADIUS SSO user group:

1. Go to *User & Authentication* > *User Groups*.
2. Click *Create New*.
3. For *Type*, select *RADIUS Single Sign-On (RSSO)*.
4. In *RADIUS Attribute Value*, enter the name of the RADIUS user group that this local user group represents.
5. Click *OK*.

Configuring security policies

The following security policies are required for RADIUS SSO:

Sequence Number	From	To	Type	Schedule	Description
1	internal	wan1	RADIUS SSO	Business hours	Authenticate outgoing user traffic
2	internal	wan1	Regular	Always	Allow essential network services and VoIP
3	dmz	wan1	Regular	Always	Allow servers to access the Internet
4	internal	dmz	Regular	Always	Allow users to access servers
5	any	any	Deny	Always	Implicit policy denying all traffic that has not been matched

You must place the RADIUS SSO policy at the top of the policy list so that it is matched first. The only exception to this is if you have a policy to deny access to a list of banned users. In this case, you must put that policy at the top so that the RADIUS SSO does not mistakenly match a banned user or IP address.

You must configure lists before creating security policies.

Schedule

You must configure a business_hours schedule. You can configure a standard Monday to Friday 8 AM to 5 PM schedule, or whatever days and hours covers standard work hours at the company.

Address groups

You must configure the following address groups:

Name	Interface	Address range included
internal_network	internal	10.11.102.110 to 10.11.102.250
company_servers	dmz	10.11.101.110 to 10.11.101.250

Service groups

You must configure the service groups. The services listed are suggestions and you may include more or less as required:

Name	Interface	Description of services to be included
essential_network_services	internal	Any network protocols required for normal network operation such as DNS, NTP, BGP
essential_server_services	dmz	All the protocols required by the company servers such as BGP, HTTP, HTTPS, FTP, IMAP, POP3, SMTP, IKE, SQL, MYSQL, NTP, TRACEROUTE, SOCKs, and SNMP
user_services	internal	Any protocols required by users such as HTTP, HTTPS, FTP

The following security policy configurations are basic and only include logging and default AV and IPS. These policies allow or deny access to non-RADIUS SSO traffic. These are essential as network services including DNS, NTP, and FortiGuard require access to the Internet.

To configure security policies:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Click *Create New*.
3. Configure the policy as follows, then click *OK*:

Incoming Interface	Internal
Source Address	internal_network
Outgoing Interface	wan1
Destination Address	all
Schedule	always
Service	essential_network_services

Action	ACCEPT
NAT	ON
Security Profiles	ON: AntiVirus, IPS
Log Allowed Traffic	ON
Comments	Essential network services

4. Click *Create New*, and configure the new policy as follows, then click *OK*:

Incoming Interface	dmz
Source Address	company_servers
Outgoing Interface	wan1
Destination Address	all
Schedule	always
Service	essential_server_services
Action	ACCEPT
NAT	ON
Security Profiles	ON: AntiVirus, IPS
Log Allowed Traffic	enable
Comments	Company servers accessing the Internet

5. Click *Create New*, and configure the new policy as follows, then click *OK*:

Incoming Interface	Internal
Source Address	internal_network
Outgoing Interface	dmz
Destination Address	company_servers
Schedule	always
Service	all
Action	ACCEPT
NAT	ON
Security Profiles	ON: AntiVirus, IPS
Log Allowed Traffic	enable
Comments	Access company servers

6. Click *Create New*, and configure the RADIUS SSO policy as follows, then click *OK*. This policy allows access for members of specific RADIUS groups.

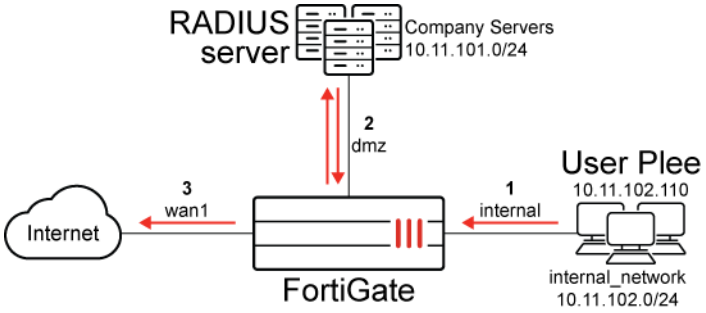
Incoming Interface	Internal
Source Address	internal_network
Source User(s)	Select the user groups that you created for RSSO.
Outgoing Interface	wan1
Destination Address	all
Schedule	business_hours
Service	ALL
Action	ACCEPT
NAT	ON
Security Profiles	ON: AntiVirus, Web Filter, IPS, and Email Filter. In each case, select the default profile.

7. Place the RSSO policy higher in the security policy list than more general policies for the same interfaces. Click *OK*.

To test the configuration:

Once configured, a user only needs to log in to their PC using their RADIUS account. After that, when they attempt to access the Internet, the FortiGate uses their session information to get their RADIUS information. Once the user is verified, they can access the website.

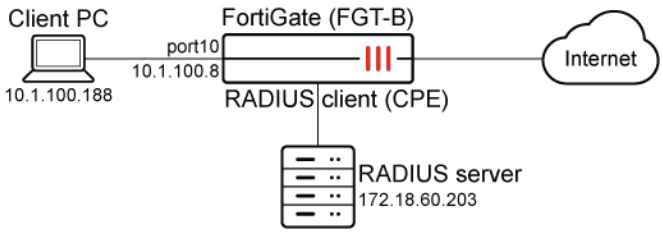
1. The user logs on to their PC and tries to access the Internet.
2. The FortiGate contacts the RADIUS server for the user's information. Once confirmed, the user can access the Internet. Each step generates logs that enable you to verify that each step succeeded.
3. If a step does not succeed, confirm that your configuration is correct.



Trigger RADIUS authentication with DNS and ICMP queries

DNS and ICMP queries can trigger RADIUS authentication. In some situations, a RADIUS client cannot trigger authentication with only HTTP, HTTPS, or Telnet traffic, such as VoIP gateways or servers. Without RADIUS authentication there is no RADIUS accounting, which can be required in some circumstances.

In this example, a Client PC is used as a VoIP gateway.



To configure FGT-B:

1. Configure a RADIUS user:

```
config user radius
    edit "FreeRADIUS"
        set server "172.18.60.203"
        set secret **********
        set acct-interim-interval 600
        set auth-type pap
        set password-renewal disable
        config accounting-server
            edit 1
                set status enable
                set server "172.18.60.203"
                set secret **********
            next
        end
    next
end
```

2. Find the MAC address on the interface. In the RADIUS accounting start message it will be the *Called-Station-Id*.

```
# get hardware nic port10 | grep HW
Current_HWaddr   80:80:2c:a3:50:f3
Permanent_HWaddr 80:80:2c:a3:50:f3
```

3. Add the RADIUS user to a group:

```
config user group
    edit "remote-radius"
        set member "FreeRADIUS"
    next
end
```

4. Configure the interface:

```
config system interface
    edit "port10"
        set vdom "root"
        set ip 10.1.100.8 255.255.255.0
        set allowaccess ping https ssh http telnet
        set type physical
```

```
        set security-mode captive-portal
        set security-ip-auth-bypass enable
        set security-groups "remote-radius"
        set device-identification enable
        set snmp-index 12
        config ipv6
            set ip6-address 2008::8/64
            set ip6-allowaccess ping https http
        end
    next
end
```

5. Configure a policy:

```
config firewall policy
    edit 1
        set name "pol1"
        set srcintf "port10"
        set dstintf "port9"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "custom-deep-inspection"
        set av-profile "default"
        set logtraffic all
        set nat enable
        set groups "remote-radius"
        set radius-ip-auth-bypass enable
    next
end
```

To test the configuration:

1. Trigger RADIUS authentication on the Client PC using the `ip-auth-bypass` feature configured on the FortiGate. This can be done by either:
   - Pinging any external resource (ICMP query).
   - Connecting to an external webpage using an FQDN URL in a browser with an empty cache, which requires a new DNS query.

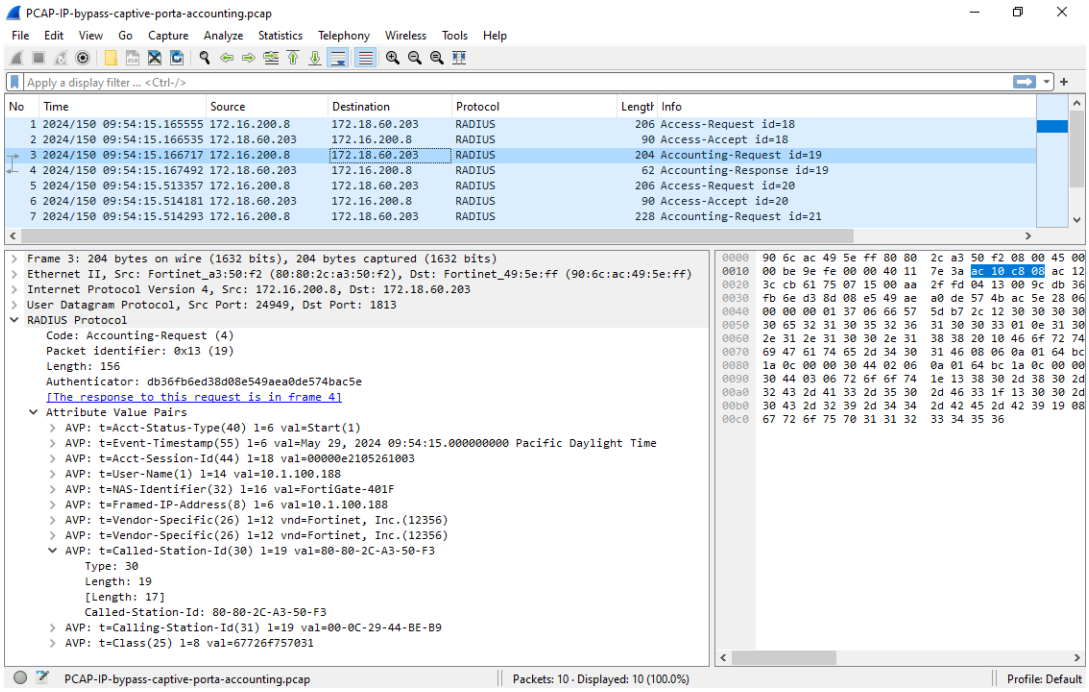
   The FortiGate will use the Client PC IP address as the credentials for authentication. If successful, this will be followed by a RADIUS accounting start message.
2. On FGT-B, check the local firewall authentication list.

   The RADIUS user that is used for `ip-auth-bypass` is the IP address of the Client PC. The source MAC address is the MAC address of the Client PC.

```
# diagnose firewall auth list

10.1.100.188, 10.1.100.188
```

```
        src_mac: 00:0c:29:44:be:b9
        type: fw, id: 0, duration: 16, idled: 10
        expire: 284, allow-idle: 300, max-life: 2384
        flag(14): hard radius
        server: FreeRADIUS
        packets: in 4 out 5, bytes: in 1105 out 1044
        group_id: 5
        group_name: remote-radius

----- 1 listed, 0 filtered ------
```

3. Check the RADIUS accounting start messages in the traffic between FGT-B and the RADIUS server. The MAC address of FGT-B is the *Called-Station-Id*, and the IP address of the Client PC is the *User-Name*.



RSA ACE (SecurID) servers

SecurID is a two-factor system produced by the company RSA that uses one-time password (OTP) authentication. This system consists of the following:

- Portable tokens that users carry
- RSA ACE/Server
- Agent host (the FortiGate)

When using SecurID, users carry a small device or "token" that generates and displays a pseudo-random password. According to RSA, each SecurID authenticator token has a unique 64-bit symmetric key that is combined with a powerful algorithm to generate a new code every 60 seconds. The token is time-synchronized with the SecurID RSA ACE/Server.

The RSA ACE/Server is the SecurID system's management component. It stores and validates the information about the SecurID tokens allowed on your network. Alternately, the server can be an RSA SecurID 130 appliance.

The agent host is the server on your network. In this case, this is the FortiGate, which intercepts user logon attempts. The agent host gathers the user ID and password entered from the SecurID token and sends the information to the RSA ACE/Server for validation. If valid, the RSA ACE/Server returns a reply indicating that it is a valid logon and FortiOS allows the user access to the network resources specified in the associated security policy.

Configuring SecurID with FortiOS consists of the following:

1. Configure the RSA and RADIUS servers to work with each other. See RSA server documentation.
2. Do one of the following:
   a. Configure the RSA SecurID 130 appliance.
   b. Configure the FortiGate as an agent host on the RSA ACE/Server.
3. Configure the RADIUS server in FortiOS.
4. Create a SecurID user group.
5. Create a SecurID user.
6. Configure authentication with SecurID.

The following instructions are based on RSA ACE/Server 5.1 and RSA SecurID 130 appliance. They assume that you have successfully completed all external RSA and RADIUS server configuration.

In this example, the RSA server is on the internal network and has an IP address of 192.168.100.102. The FortiOS internal interface address is 192.168.100.3. The RADIUS shared secret is fortinet123, and the RADIUS server is at IP address 192.168.100.202.

To configure the RSA SecurID 130 appliance:

1. Log on to the SecurID IMS console.
2. Go to *RADIUS* > *RADIUS clients*, then select *Add New*.

RADIUS Client Basics		
	Client Name	FortiGate
	Associated RSA Agent	FortiGate
RADIUS Client Settings		
	IP Address	Enter the FortiOS internal interface. In this example, it is 192.168.100.3.
	Make / Model	Select *Standard Radius*.
	Shared Secret	Enter the RADIUS shared secret. In this example, it is fortinet123.
	Accounting	Leave unselected.
	Client Status	Leave unselected.

3. Configure your FortiGate as a SecurID client:
4. Click *Save*.

To configure the FortiGate as an agent host on the RSA ACE/Server:

1. On the RSA ACE/Server, go to *Start > Programs > RSA ACE/Server*, then *Database Administration - Host Mode*.
2. From the *Agent Host* menu, select *Add Agent Host*.
3. Configure the following:

Name	FortiGate
Network Address	Enter the FortiOS internal interface. In this example, it is 192.168.100.3.
Secondary Nodes	You can optionally enter other IP addresses that resolve to the FortiGate.

For more information, see the RSA ACE/Server documentation.

To configure the RADIUS server in FortiOS:

1. Go to *User & Authentication > RADIUS Servers*, then click *Create New*.
2. Configure the following:

Name	RSA
Authentication method	Select *Default*.
Primary Server	
IP/Name	192.168.100.102. You can click *Test* to ensure the IP address is correct and that FortiOS can contact the RADIUS server.
Secret	fortinet123

3. Click *OK*.

To create a SecurID user group:

1. Go to *User & Authentication > User Groups*. Click *Create New*.
2. Configure the following:

Name	RSA_group
Type	Firewall

3. In *Remote Groups*, click *Add*, then select the RSA server.
4. Click *OK*.

To create a SecurID user:

1. Go to *User & Authentication > User Definition*. Click *Create New*.
2. Configure the following:

User Type	Remote RADIUS User
Type	wloman

RADIUS Server	RSA
Contact Info	(Optional) Enter email or SMS information.
User Group	RSA_group

3. Click *Create*.

You can test the configuration by entering the `diagnose test authserver radius RSA auto wloman 111111111` command. The series of 1s is the OTP that your RSA SecurID token generates that you enter for access.

Configuring authentication with SecurID

You can use the SecurID user group in several FortiOS features that authenticate by user group:

- Security policy on page 3551
- IPsec VPN XAuth on page 3552
- PPTP VPN on page 3552
- Agentless VPN

Unless stated otherwise, the following examples use default values.

Security policy

The example creates a security policy that allows HTTP, FTP, and POP3 traffic from the internal interface to WAN1. If these interfaces are not available in FortiOS, substitute other similar interfaces.

To configure a security policy with SecurID authentication:

1. Go to *Policy & Objects > Firewall Policy*.
2. Click *Create New*.
3. Configure the following:

Incoming Interface	internal
Source Address	all
Source User(s)	RSA_group
Outgoing Interface	wan1
Destination Address	all
Schedule	always
Service	HTTP, FTP, POP3
Action	ACCEPT
NAT	On
Shared Shaper	If you want to limit traffic or guarantee minimum bandwidth for traffic that uses the SecurID security policy, enable and use the default shaper, guarantee-100kbps.

Log Allowed Traffic	Enable if you want to generate usage reports on traffic that this policy has authenticated.

4. Click *OK*.

IPsec VPN XAuth

In *VPN > IPsec Wizard*, select the SecurID user group on the *Authentication* page. The SecurID user group members must enter their SecurID code to authenticate.

PPTP VPN

When configuring PPTP in the CLI, set `usrgrp` to the SecurID user group.

Agentless VPN

You must map the SecurID user group to the portal that will serve SecurID users and include the SecurID user group in the security policy's *Source User(s)* field.

To map the SecurID group to an Agentless VPN portal:

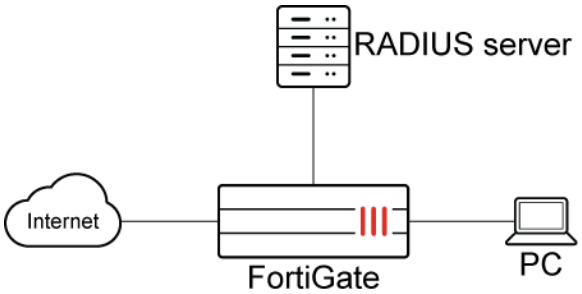
1. Go to *VPN > Agentless VPN Settings*.
2. Under *Authentication/Portal Mapping*, click *Create New*.
3. Configure the following:

Users/Groups	RSA_group
Portal	Select the desired portal.

4. Click *OK*.

Support for Okta RADIUS attributes filter-Id and class

RADIUS user group membership information can be returned in the filter-Id (11) and class (25) attributes in RADIUS Access-Accept messages. The group membership information can be used for group matching in FortiGate user groups in firewall policies and for FortiGate wildcard administrators with remote RADIUS authentication.



In this example, a FortiAuthenticator is used as the RADIUS server. A local RADIUS user on the FortiAuthenticator is configure with two groups in the filter-Id attribute: *okta-group1* and *okta-group2*.

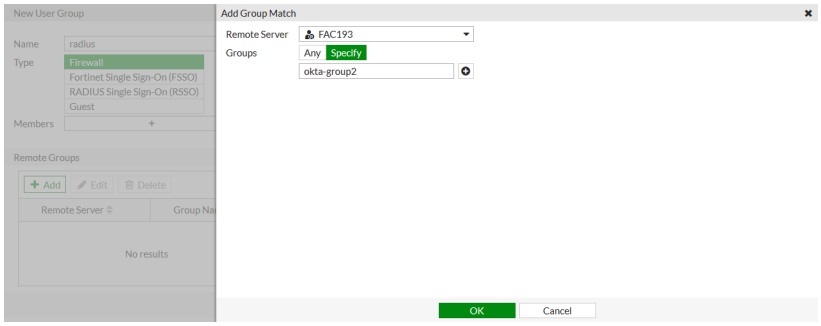
To create the RADIUS user and set the attribute type to override group information:

```
config user radius
    edit "FAC193"
        set server "10.1.100.189"
        set secret **********
        set group-override-attr-type filter-Id
    next
end
```

FortiOS will only use the configured filter-Id attribute, even if the RADIUS server sends group names in both class and filter-id attributes. To return group membership information from the class attribute instead, set `group-override-attr-type` to `class`.

To configure group match in the user group:

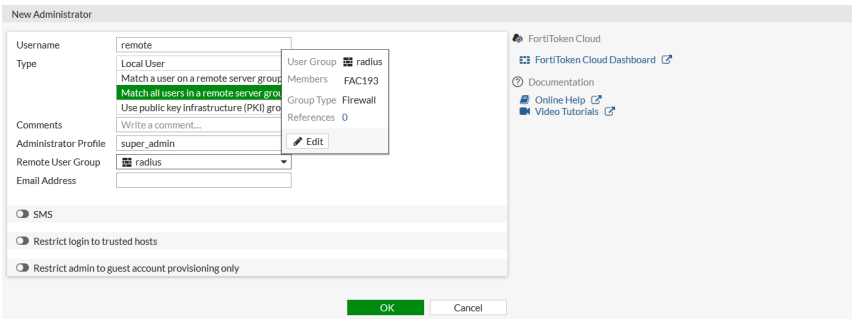
1. Go to *User & Authentication* > *User Groups*.
2. Click *Create New*.
3. Enter a name for the group, and set *Type* to *Firewall*.
4. In the *Remote Groups* table, click *Add*.
5. Set *Remote Server* to the just created RADIUS server, *FAC193*.
6. Set *Groups* to *Specify*, and enter the group name, `okta-group2`. The string must match the group name configured on the RADIUS server for the filter-Id attribute.



7. Click *OK*.
   The remote server is added to the *Remote Groups* table.
8. Click *OK*.
9. Add the new user group to a firewall policy and generate traffic on the client PC that requires firewall authentication, such as connecting to an external web server.
10. After authentication, on the FortiGate, verify that traffic is authorized in the traffic log:
    a. Go to *Log & Report* > *Forward Traffic*.
    b. Verify that the traffic was authorized.

To use the remote user group with group match in a system wildcard administrator configuration:

1. Go to *System* > *Administrators*.
2. Edit an existing administrator, or create a new one.
3. Set *Type* to *Match all users in a remote server group*.
4. Set *Remote User Group* to the remote server.



5. Configure the remaining settings as required.
6. Click *OK*.
7. Log in to the FortiGate using the remote user credentials on the RADIUS server.
   If the correct group name is returned in the filter-Id attribute, administrative access is allowed.

Sending multiple RADIUS attribute values in a single RADIUS Access-Request

A managed FortiSwitch can be configured to send multiple RADIUS attribute values in a single RADIUS Access-Request. This option is configured per RADIUS user, and is set to none by default.

The available service type options are:

login	User should be connected to a host.
framed	User use Framed Protocol.
callback-login	User disconnected and called back.
callback-framed	User disconnected and called back, then a Framed Protocol.
outbound	User granted access to outgoing devices.
administrative	User granted access to the administrative unsigned interface.
nas-prompt	User provided a command prompt on the NAS.
authenticate-only	Authentication requested, and no authentication information needs to be returned.
callback-nas-prompt	User disconnected and called back, then provided a command prompt.
call-check	Used by the NAS in an Access-Request packet, Access-Accept to answer the call.
callback-administrative	User disconnected and called back, granted access to the admin unsigned interface.

To configure a managed FortiSwitch to the RADIUS attributes login, framed, and authenticate-only all at the same time:

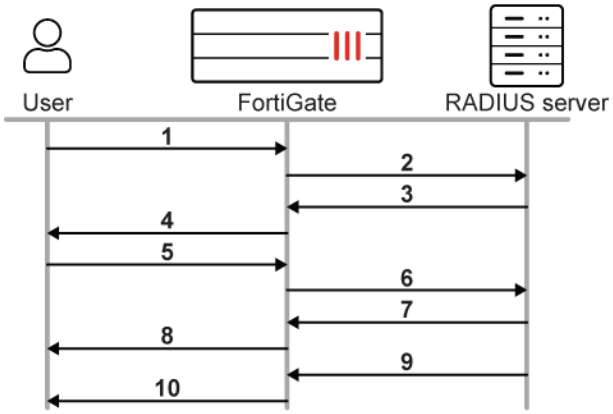
```
config user radius
    edit "Radius_Server"
        set switch-controller-service-type login framed authenticate-only
        ....
    next
end
```

Traffic shaping based on dynamic RADIUS VSAs

A FortiGate can use the WISPr-Bandwidth-Max-Down and WISPr-Bandwidth-Max-Up dynamic RADIUS VSAs (vendor-specific attributes) to control the traffic rates permitted for a certain device. The FortiGate can apply different traffic shaping to different users who authenticate with RADIUS based on the returned RADIUS VSA values. When the same user logs in from an additional device, the RADIUS server will send a CoA (change of authorization) message to update the bandwidth values to 1/*N* of the total values, where *N* is the number of logged in devices from the same user.

This feature is not supported by FortiGates with NP6, NP6Lite (SoC3), or NP6XLite (SoC4) processors. NP6 offloading is automatically disabled on the policy if this feature is enabled. This feature is supported by FortiGates with NP7 and NP7Lite (SoC5) processors.

When a user logs in to two devices through RADIUS authentication. The authentication and authorization flow is as follows:



1. The user logs in to a device and the authentication is sent to the FortiGate.
2. The FortiGate sends the Access-Request message to the RADIUS server.
3. The RADIUS server sends the Access-Accept message to the FortiGate. The server also returns the WISPr-Bandwidth-Max-Up and WISPr-Bandwidth-Max-Down VSAs.
4. Based on the VSA values, the FortiGate applies traffic shaping for the upload and download speeds based on its IP.
5. The user logs in to a second device and the authentication is sent to the FortiGate.
6. The FortiGate sends the Access-Request message to the RADIUS server.

7. The RADIUS server sends the Access-Accept message to the FortiGate. The server also returns the WISPr-Bandwidth-Max-Up and WISPr-Bandwidth-Max-Down VSAs at half the value from the first device.
8. Based on the VSA values, the FortiGate applies traffic shaping for the upload and download speeds on the second device based on its IP.
9. The RADIUS server sends a CoA message and returns WISPr-Bandwidth-Max-Up and WISPr-Bandwidth-Max-Down VSAs for the first device at half the value.
10. Based on the VSA values, the FortiGate updates traffic shaping for the upload and download speeds on the first device based on its IP.

Example

In this example, the FortiGate is configured to dynamically shape user traffic based on the WISPr-Bandwidth-Max-Up and WISPr-Bandwidth-Max-Down VSAs returned by the RADIUS server when the user logs in through firewall authentication.

To configure traffic shaping based on dynamic RADIUS VSAs:

1. Configure the RADIUS server users file to identify WISPr-Bandwidth-Max-Up and WISPr-Bandwidth-Max-Down:

   The WISPr-Bandwidth is measured in bps, and the FortiOS dynamic shaper is measured in Bps.

   ```
   WISPr-Bandwidth-Max-Up = 1004857,
   WISPr-Bandwidth-Max-Down = 504857,
   ```

2. In FortiOS, configure the RADIUS server:

   ```
   config user radius
       edit "rad1"
           set server "172.16.200.44"
           set secret ************
           set radius-coa enable
           set acct-all-servers enable
           config accounting-server
               edit 1
                   set status enable
                   set server "172.16.200.44"
                   set secret ************
               next
           end
       next
   end
   ```

3. Configure the RADIUS user group:

```
config user group
    edit "group_radius"
        set member "rad1"
    next
end
```

4. Configure the firewall policy with dynamic shaping and the RADIUS group:

```
config firewall policy
    edit 2
        set srcintf "port2"
        set dstintf "wan1"
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all6"
        set dstaddr6 "all6"
        set action accept
        set schedule "always"
        set service "ALL"
        set dynamic-shaping enable
        set groups "group_radius"
        set nat enable
    next
end
```

Verification

After a client PC is authenticated by the RADIUS server, dynamic shaping is applied to the client based on the IP address.

Use the following commands to monitor the dynamic shaper:

```
# diagnose firewall shaper dynamic-shaper stats
```

```
# diagnose firewall shaper dynamic-shaper list {ip | ipv6 | user} <address or username>
```

Use case 1

User1 is paying for rate plan A that limits their maximum bandwidth to 10 Mbps download and 5 Mbps upload. User2 is paying for rate plan B that limits their maximum bandwidth to 5 Mbps download and 5 Mbps upload. The speeds in both plans are provided by best effort, so there is no guaranteed minimum bandwidth.

User1 logs in to pc1 with RADIUS authentication and IP-based dynamic shaping is applied. User2 logs in to pc2 with RADIUS authentication and IP-based dynamic shaping is applied.

To verify the dynamic shaping:

1. On pc1, verify the bandwidth and transfer speed:

```
root@pc1:~# iperf -c 172.16.200.44 -u -t 25 -b 20M
------------------------------------------------------------
Client connecting to 172.16.200.44, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 10.1.100.11 port 50510 connected with 172.16.200.44 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-25.0 sec  59.6 MBytes  20.0 Mbits/sec
[  3] Sent 42518 datagrams
[  3] Server Report:
[  3]  0.0-25.3 sec  30.1 MBytes  9.99 Mbits/sec  15.651 ms 21058/42518 (50%)
```

2. On pc2, verify the bandwidth and transfer speed:

```
root@pc2:~# iperf -c 172.16.200.44 -u -t 25 -b 20M
------------------------------------------------------------
Client connecting to 172.16.200.44, UDP port 5001
Sending 1470 byte datagrams
UDP buffer size:  208 KByte (default)
------------------------------------------------------------
[  3] local 10.1.100.22 port 52814 connected with 172.16.200.44 port 5001
[ ID] Interval       Transfer     Bandwidth
[  3]  0.0-25.0 sec  59.6 MBytes  20.0 Mbits/sec
[  3] Sent 42518 datagrams
[  3] Server Report:
[  3]  0.0-25.3 sec  15.1 MBytes  5.03 Mbits/sec  15.652 ms 31710/42514 (75%)
```

3. In FortiOS, check the authentication list:

```
# diagnose firewall auth  list
10.1.100.11, test-shaper1
        src_mac: **:**:**:**:**:**
        type: fw, id: 0, duration: 38, idled: 16
        expire: 562
        flag(814): hard radius no_idle
        server: rad1
        packets: in 8207 out 3999, bytes: in 12306164 out 226963
        group_id: 3
        group_name: group_radius
10.1.100.22, test-shaper2
        src_mac: **:**:**:**:**:**
        type: fw, id: 0, duration: 24, idled: 24
        expire: 156, max-life: 35976
        flag(814): hard radius no_idle
        server: rad1
        packets: in 0 out 5, bytes: in 0 out 300
        group_id: 3
```

```
       group_name: group_radius
----- 2 listed, 0 filtered ------
```

4. Check the dynamic shaper list:

```
# diagnose firewall shaper dynamic-shaper list
addr: 10.1.100.11
bandwidth(original/reply): 1250000 Bps/625000 Bps
current bandwidth(original/reply): 1237072 Bps/0 Bps
allow packets(original/reply): 38524/14
allow bytes(original/reply): 55270378/11285
drop packets(original/reply): 10136/0
drop bytes(original/reply): 13516198/0
life: 441
idle: 0/40
idle time limit: 600 s

addr: 10.1.100.22
bandwidth(original/reply): 625000 Bps/625000 Bps
current bandwidth(original/reply): 622909 Bps/0 Bps
allow packets(original/reply): 3232/3
allow bytes(original/reply): 4841536/243
drop packets(original/reply): 2753/0
drop bytes(original/reply): 4123994/0
life: 10
idle: 0/10
idle time limit: 36000 s
```

5. Check the session list:

```
# diagnose sys session list
session info: proto=6 proto_state=05 duration=3 expire=116 timeout=3600 flags=00000004
socktype=4 sockport=10001 av_idx=0 use=4
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/0
state=redir log local may_dirty auth dst-vis f00 dynamic_shaping
statistic(bytes/packets/allow_err): org=0/0/0 reply=638/4/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 185/1
orgin->sink: org pre->post, reply pre->post dev=20->17/17->20 gwy=172.16.200.44/0.0.0.0
hook=pre dir=org act=noop 10.1.100.22:35561->172.16.200.44:80(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.200.44:80->10.1.100.22:35561(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
src_mac=**:**:**:**:**:**  dst_mac=**:**:**:**:**:**
misc=0 policy_id=2 auth_info=0 chk_client_info=0 vd=1
serial=0005994d tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id=00000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  redir-to-av auth disabled-by-policy
```

```
session info: proto=6 proto_state=05 duration=122 expire=38 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
user=test-shaper1 auth_server=rad1 state=log may_dirty authed f00 dynamic_shaping acct-ext
statistic(bytes/packets/allow_err): org=383611/6604/1 reply=26382470/17592/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=20->17/17->20 gwy=172.16.200.44/10.2.2.1
hook=post dir=org act=snat 10.1.100.11:54140->172.16.200.44:80(172.16.200.2:54140)
hook=pre dir=reply act=dnat 172.16.200.44:80->172.16.200.2:54140(10.1.100.11:54140)
pos/(before,after) 0/(0,0), 0/(0,0)
src_mac=**:**:**:**:**:**  dst_mac=**:**:**:**:**:**
misc=0 policy_id=2 auth_info=3 chk_client_info=0 vd=1
serial=000598c5 tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id=00000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
total session 2
```

6. Check the policy traffic:

```
# diagnose firewall iprope list 100004
policy index=2 uuid_idx=60 action=accept
flag (8052128): redir auth nat nids_raw master use_src pol_stats
flag2 (4030): fw wsso resolve_sso
flag3 (200000b0): !sp link-local best-route dynamic-shaping
schedule(always)
cos_fwd=255  cos_rev=255
group=00100004 av=00004e20 au=00000003 split=00000000
host=1 chk_client_info=0x1 app_list=0 ips_view=0
misc=0
zone(1): 20 -> zone(1): 17
source(1): 0.0.0.0-255.255.255.255, uuid_idx=32,
dest(1): 0.0.0.0-255.255.255.255, uuid_idx=32,
user group(1): 3
service(1):
        [0:0x0:0/(0,65535)->(0,65535)] helper:auto
```

Use case 2

A user logs in to a device (pc1, 10.1.100.11) and has a maximum bandwidth of 10 Mbps download and 5 Mbps upload. The same user logs in to a second device (pc2, 10.1.100.22) and the RADIUS server sends a CoA request with the WISPr-Bandwidth-Max to pc1. The maximum bandwidth on pc1 changes to 5 Mbps download and 2.5Mbps upload. On pc2, the maximum bandwidth is also 5 Mbps download and 2.5Mbps upload.

When the user logs out from pc1, the RADIUS server sends CoA request with the new WISPr-Bandwidth-Max for pc2. The FortiGate updates the authentication user list and dynamic shaper for pc2. The maximum bandwidth on pc2 changes to 10 Mbps download and 5 Mbps upload.

To verify the dynamic shaping:

1. Check the dynamic shaper list after the user logs in to pc1:

```
# diagnose firewall shaper dynamic-shaper list
addr: 10.1.100.11
bandwidth(original/reply): 1250000 Bps/625000 Bps
current bandwidth(original/reply): 0 Bps/0 Bps
allow packets(original/reply): 0/3
allow bytes(original/reply): 0/243
drop packets(original/reply): 0/0
drop bytes(original/reply): 0/0
life: 491
idle: 4/4
idle time limit: 86400 s
```

2. Check the dynamic shaper list after the user logs in to pc2:

```
# diagnose firewall shaper dynamic-shaper list
addr: 10.1.100.11
bandwidth(original/reply): 625000 Bps/312500 Bps
current bandwidth(original/reply): 0 Bps/0 Bps
allow packets(original/reply): 0/0
allow bytes(original/reply): 0/0
drop packets(original/reply): 0/0
drop bytes(original/reply): 0/0
life: 652
idle: 5/5
idle time limit: 600 s

addr: 10.1.100.22
bandwidth(original/reply): 625000 Bps/312500 Bps
current bandwidth(original/reply): 0 Bps/0 Bps
allow packets(original/reply): 0/3
allow bytes(original/reply): 0/243
drop packets(original/reply): 0/0
drop bytes(original/reply): 0/0
life: 3
idle: 3/3
idle time limit: 86400 s
```

3. Check the authentication list:

```
# diagnose firewall auth list
10.1.100.11, test
        src_mac: **:**:**:**:**:**
        type: fw, id: 0, duration: 171, idled: 11
        expire: 589, max-life: 589
        flag(814): hard radius no_idle
        server: rad1
        packets: in 0 out 0, bytes: in 0 out 0
        group_id: 15
        group_name: group_radius
```

```
10.1.100.22, test
        src_mac: **:**:**:**:**:**
        type: fw, id: 0, duration: 9, idled: 9
        expire: 86391
        flag(814): hard radius no_idle
        server: rad1
        packets: in 0 out 0, bytes: in 0 out 0
        group_id: 15
        group_name: group_radius
----- 2 listed, 0 filtered ------
```

4. Check the dynamic shaper list after the user logs out from pc1:

```
# diagnose firewall shaper dynamic-shaper list
addr: 10.1.100.22
bandwidth(original/reply): 1250000 Bps/625000 Bps
current bandwidth(original/reply): 0 Bps/0 Bps
allow packets(original/reply): 0/0
allow bytes(original/reply): 0/0
drop packets(original/reply): 0/0
drop bytes(original/reply): 0/0
life: 414
idle: 9/9
idle time limit: 600 s
```

5. Check the authentication list again:

```
# diagnose firewall auth  list
10.1.100.22, test
        src_mac: **:**:**:**:**:**
        type: fw, id: 0, duration: 453, idled: 49
        expire: 551, max-life: 551
        flag(814): hard radius no_idle
        server: rad1
        packets: in 0 out 0, bytes: in 0 out 0
        group_id: 15
        group_name: group_radius
----- 1 listed, 0 filtered ------
```

RADIUS Termination-Action AVP in wired and wireless scenarios

When authenticating with RADIUS in a wired or wireless scenario, the FortiGate can support proper handling of the Termination-Action AVP.

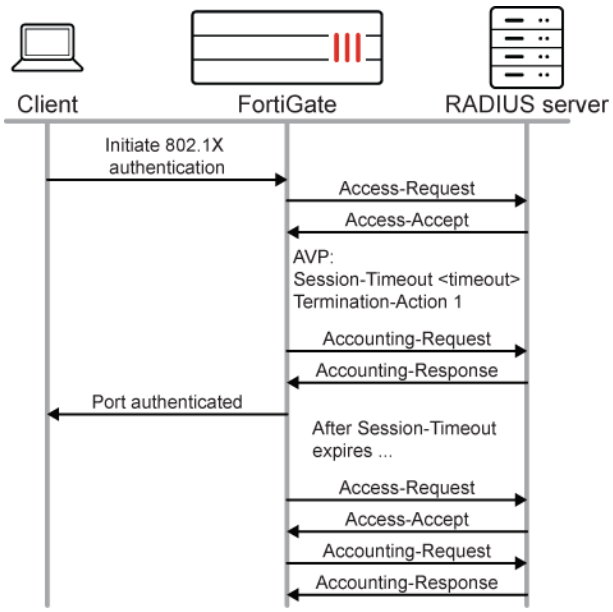
In a wired scenario, a hardware switch configured with 802.1X security authentication can read the Termination-Action attribute value from the RADIUS Access-Accept response. If the Termination-Action is 1, the FortiGate will initiate re-authentication when the session time has expired. During re-authentication, the port stays authorized. If the Termination-Action is 0, the session will be terminated.

In a wireless scenario, when a virtual AP is configured with WPA2-Enterprise security with RADIUS and has CoA enabled, it processes the RADIUS CoA request immediately upon receiving it and re-authenticates when the Termination-Action is 1.

Wired example

This example has a FortiGate configured with a hardware switch with two ports: port3 and port5. The hardware switch is enabled with 802.1X security and assigned to a RADIUS user group. Upon a successful authentication, the RADIUS server responds with an Access-Accept containing the authentication Session-Timeout and Termination-Action attributes. In this example, the Termination-Action value is 1, which informs the client to re-authenticate when the session time expires. During this time, the FortiGate keeps the client/port authorized while it initiates the re-authentication with the RADIUS server.

The message exchange is as follows:



To configure the RADIUS server and the FortiGate to handle the Termination-Action AVP:

1. On the RADIUS server, configure the Termination-Action AVP with the value `RADIUS-Request (1)` to indicate that re-authentication should occur upon expiration of the Session-Time.
2. On the FortiGate, configure the RADIUS server:

```
config user radius
    edit "rad1"
        set server "172.18.60.203"
        set secret ENC **********
        set radius-coa enable
        config accounting-server
            edit 1
                set status enable
                set server "172.18.60.203"
                set secret ENC **********
```

```
            next
        end
    next
end
```

3. Configure the RADIUS user group:

```
config user group
    edit "group_radius"
        set member "rad1"
    next
end
```

4. Configure the hardware switch with 802.1X enabled.
   a. Configure the virtual switch settings:

```
config system virtual-switch
    edit hw2
        set physical-switch "sw0"
        config port
            edit port3
            next
            edit port5
            next
        end
    next
end
```

   b. Configure the interface settings:

```
config system interface
    edit hw2
        set vdom vdom1
        set ip 6.6.6.1 255.255.255.0
        set allowaccess ping https ssh
        set stp enable
        set security-mode 802.1X
        set security-groups "group_radius"
    next
end

WARNING: Changing 802.1X could interrupt network connectivity on affected interfaces.
Do you want to continue? (y/n)y
```

5. On the client device, initiate 802.1X authentication, then verify that the switch port shows as authorized:

```
# diagnose sys 802-1x status
Virtual switch 'hw2' (default mode) 802.1x member status:
  port3: Link up, 802.1X state: unauthorized
  port5: Link up, 802.1X state: authorized
```

6. After successful authentication, wait for the session to timeout.

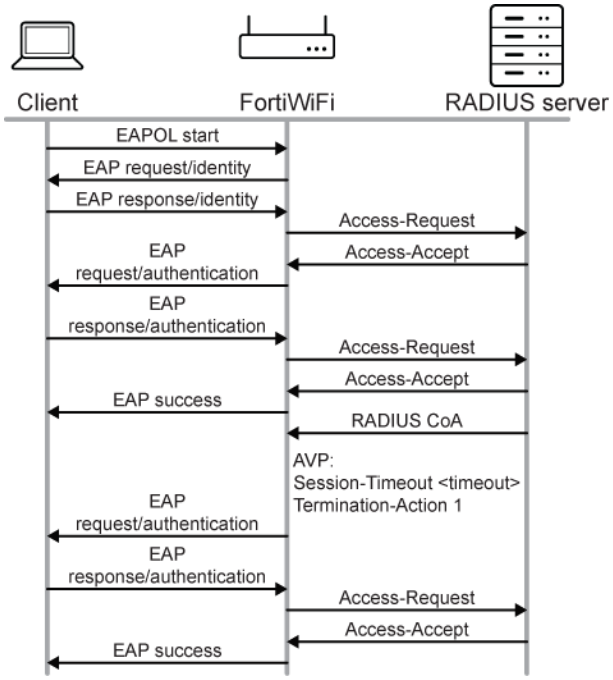
7. The FortiGate will keep the 802.1X port authenticated, and initiate re-authentication with the same Acct-Session-Id to the RADIUS server. The 802.1X status of the port remains unchanged:

```
# diagnose sys 802-1x status
Virtual switch 'hw2' (default mode) 802.1x member status:
  port3: Link up, 802.1X state: unauthorized
  port5: Link up, 802.1X state: authorized
```

Wireless example

In this example, a virtual AP is configured with WPA2-Enterprise security with RADIUS and has CoA enabled. After a wireless user authenticates and connects to the wireless SSID, the RADIUS server triggers a CoA event with AVPs Session-timeout and a Termination-Action of 1. This signals the FortiGate to trigger re-authentication of the client, which the client immediately performs to stay connected to the wireless SSID.

The message exchange is as follows:



To configure the FortiGate to handle the Termination-Action AVP:

1. Configure the RADIUS server:

```
config user radius
    edit "peap"
        set server "172.16.200.55"
        set secret **********
        set radius-coa enable
    next
end
```

2. Configure the VAP:

```
config wireless-controller vap
    edit "wifi"
        set ssid "FWF-60F-coa"
        set security wpa2-only-enterprise
        set auth radius
        set radius-server "peap"
        set schedule "always"
    next
end
```

3. Verify that the wireless station connects to the SSID:

```
# diagnose wireless-controller wlac -d sta online
   vf=0 wtp=1 rId=1 wlan=wifi vlan_id=0 ip=10.10.80.2 ip6=:: mac=**:**:**:**:**:** vci=
host=wifi-qa-01 user=test1 group=group1 signal=-28 noise=-95 idle=1 bw=0 use=6 chan=149 radio_
type=11AC security=wpa2_only_enterprise mpsk= encrypt=aes cp_authed=no online=yes mimo=2
```

4. From the RADIUS server, manually trigger a RADIUS CoA event.
   a. RADIUS CoA sent to the FortiGate:

```
Sent CoA-Request Id 7 from 0.0.0.0:54158 to 172.16.200.201:3799 length 39
    User-Name = "test1"
    Session-Timeout = 120
    Termination-Action = RADIUS-Request
```

   b. RADIUS CoA-ACK received from the FortiGate:

```
Received CoA-ACK Id 7 from 172.16.200.201:3799 to 0.0.0.0:0 length 44
    Event-Timestamp = "Jan  5 2022 14:43:12 PST"
    Message-Authenticator = 0x3311ba3b763d68da653ab34351b0308
```

5. On the wireless station console, verify that the re-authentication happens immediately:

```
root@wifi-qa-01:/home/wpa-test# wlan1: CTRL-EVENT-EAP-STARTED EAP authentication started
wlan1: CTRL-EVENT-EAP-PROPOSED-METHOD vendor=0 method=25
wlan1: CTRL-EVENT-EAP-METHOD EAP vendor 0 method 25 (PEAP) selected
EAP-TLV: TLV Result - Success - EAP-TLV/Phase2 Completed
wlan1: CTRL-EVENT-EAP-SUCCESS EAP authentication completed successfully
wlan1: PMKSA-CACHE-REMOVED **:**:**:**:**:** 0
wlan1: PMKSA-CACHE-ADDED **:**:**:**:**:** 0
wlan1: WPA: Key negotiation completed with **:**:**:**:**:** [PTK=CCMP GTK=CCMP]
```

Configuring a RADSEC client

FortiOS supports RADSEC clients in order to secure the communication channel over TLS for all RADIUS traffic, including RADIUS authentication and RADIUS accounting over port 2083. A FortiGate acting as a TLS client can initiate the TLS handshake with a remote RADIUS server. Administrators can specify a client certificate, perform a server identity check (enabled by default), and verify against a particular trust anchor (CA certificate). During a TLS handshake, the SNI check will use the RADIUS server FQDN if configured.

TCP connections are also supported, which use port 1812 for authentication and port 1813 for accounting.

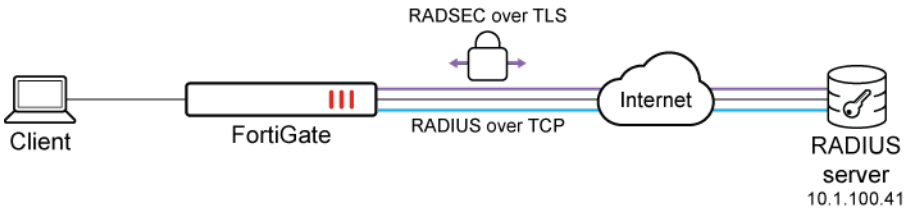
```
config user radius
    edit <name>
        set transport-protocol {udp | tcp | tls}
        set ca-cert <string>
        set client-cert <string>
        set tls-min-proto-version {default | SSLv3 | TLSv1 | TLSv1-1 | TLSv1-2}
        set server-identity-check {enable | disable}
    next
end
```

`transport-protocol {udp | tcp | tls}`	Set the type of transport protocol to use: • `udp`: use UDP (default) • `tcp`: use TCP, but no TLS security • `tls`: use TLS over TCP
`ca-cert <string>`	Set the CA certificate of server to trust under TLS.
`client-cert <string>`	Set the client certificate to use under TLS.
`tls-min-proto-version {default | SSLv3 | TLSv1 | TLSv1-1 | TLSv1-2}`	Set the minimum supported protocol version for TLS connections: • `default`: follow the system global setting • `SSLv3`: use SSLv3 • `TLSv1`: use TLSv1 • `TLSv1-1`: use TLSv1.1 • `TLSv1-2`: use TLSv1.2
`server-identity-check {enable | disable}`	Enable/disable RADIUS server identity check, which verifies the server domain name/IP address against the server certificate (default = enable).

⚠ It is best practice to enable RadSec over TLS whenever the FortiGate and RADIUS connection must pass through unencrypted transport. When using TCP and UDP transport modes, it is recommended to ensure the FortiGate and RADIUS connection passes through a trusted network or the connection passes through an encrypted tunnel over untrusted networks.

Examples

The following topology is used to demonstrate configurations using RADSEC over TLS and RADIUS over TCP.



Example 1: RADSEC over TLS

When using TLS, FortiOS uses port 2083 for RADIUS authentication and RADIUS accounting. There is no need to configure the RADIUS accounting separately.

Before configuring RADSEC over TLS, make sure that the CA certificate (which issues the remote RADIUS server certificate) is imported into the FortiGate trusted root store. If a customized local FortiGate client certificate is used, both the certificate and private key are imported into local FortiGate certificate store.

To configure RADSEC over TLS:

1. Configure the RADIUS server:

```
config user radius
    edit "radius-tls"
        set server "10.1.100.41"
        set secret ********
        set acct-interim-interval 600
        set radius-port 2083
        set auth-type pap
        set transport-protocol tls
        set ca-cert "CA_Cert_2"
        set client-cert "portal.fortinet-fsso"
        config accounting-server
            edit 1
                set status enable
                set server "10.1.100.41"
                set secret ********
            next
        end
    next
end
```

2. Enable fnbamd debug messages on the FortiGate to verify the RADIUS authentication triggered by client traffic requesting access to external networks, which requires user authentication by the firewall policy. Note the highlighted initial RADSEC TLS authentication, successfully completed TLS handshake, and RADIUS accounting using TLS over port 2083:

```
# diagnose debug application fnbamd -1
Debug messages will be on for 30 minutes.

# diagnose debug enable
...
[629] __fnbamd_cfg_add_radius_by_user-
[1726] fnbamd_match_and_update_auth_user-Found a matching user in CMDB 'test1'
[462] fnbamd_rad_get-vfid=0, name='radius-tls'
[635] __fnbamd_cfg_add_radius_by_user-Loaded RADIUS server 'radius-tls' for user 'test1'
(16777236)
[905] fnbamd_cfg_get_radius_list-Total rad servers to try: 1
...
[806] __fnbamd_rad_get_next_addr-Next available address of rad 'radius-tls': 10.1.100.41:2083.
[981] __auth_ctx_start-Connection starts radius-tls:10.1.100.41, addr 10.1.100.41:2083 proto:
```

```
TCP over TLS
[449] __rad_tcps_open-vfid 0, addr 10.1.100.41, src_ip (null), ssl_opt 1284
...
[618] create_auth_session-Total 1 server(s) to try
[1772] handle_req-r=4
[418] __rad_tcps_connect-tcps_connect(10.1.100.41) is established.
[716] __rad_rxtx-fd 10, state 1(Auth)
...
[565] fnbamd_rad_make_access_request-
[329] __create_access_request-Compose RADIUS request
[549] __create_access_request-Created RADIUS Access-Request. Len: 139.
...
[963] __auth_ctx_svr_push-Added addr 10.1.100.41:2083 from rad 'radius-tls'
[806] __fnbamd_rad_get_next_addr-Next available address of rad 'radius-tls': 10.1.100.41:2083.
[981] __auth_ctx_start-Connection starts radius-tls:10.1.100.41, addr 10.1.100.41:2083 proto:
TCP over TLS
[449] __rad_tcps_open-vfid 0, addr 10.1.100.41, src_ip (null), ssl_opt 1284
[481] __rad_tcps_open-Server identity check is enabled.
[495] __rad_tcps_open-Still connecting 10.1.100.41.
...
[1393] create_acct_session-Acct type 6 session created, 0x9827960
[418] __rad_tcps_connect-tcps_connect(10.1.100.41) is established.
[716] __rad_rxtx-fd 10, state 4(Acct)
...
[956] fnbamd_rad_make_acct_request-
[905] __create_acct_request-Compose RADIUS request
[944] __create_acct_request-Created RADIUS Acct-Request. Len: 129.
[572] __rad_tcps_send-Sent 129/129.
[574] __rad_tcps_send-Sent all. Total 129.
[749] __rad_rxtx-Sent radius req to server 'radius-tls': fd=10, IP=10.1.100.41
(10.1.100.41:2083) code=4 id=33 len=123
[758] __rad_rxtx-Start rad conn timer.
...
```

Example 2: RADIUS over TCP

When using TCP, the default RADIUS ports remain same as with UDP: 1812 for authentication and 1813 for accounting.

To configure RADIUS over TCP:

1. Configure the RADIUS server:

```
config user radius
    edit "radius-tcp"
        set server "10.1.100.41"
        set secret ********
        set acct-interim-interval 600
        set transport-protocol tcp
        config accounting-server
```

```
            edit 1
                set status enable
                set server "10.1.100.41"
                set secret ********
            next
        end
    next
end
```

2. Enable fnbamd debug messages on the FortiGate to verify the RADIUS authentication triggered by client traffic requesting access to external networks, which requires user authentication by the firewall policy. Note the highlighted initial RADIUS authentication over TCP: 1812 and initial RADIUS accounting over TCP: 1813:

```
# diagnose debug application fnbamd -1
Debug messages will be on for 30 minutes.

# diagnose debug enable
...

[806] __fnbamd_rad_get_next_addr-Next available address of rad 'radius-tcp': 10.1.100.41:1812.
[981] __auth_ctx_start-Connection starts radius-tcp:10.1.100.41, addr 10.1.100.41:1812 proto:
TCP
[449] __rad_tcps_open-vfid 0, addr 10.1.100.41, src_ip (null), ssl_opt 0
...
[1772] handle_req-r=4
[418] __rad_tcps_connect-tcps_connect(10.1.100.41) is established.
[716] __rad_rxtx-fd 10, state 1(Auth)
...
[565] fnbamd_rad_make_access_request-
[329] __create_access_request-Compose RADIUS request
[549] __create_access_request-Created RADIUS Access-Request. Len: 139.
[572] __rad_tcps_send-Sent 139/139.
[574] __rad_tcps_send-Sent all. Total 139.
[749] __rad_rxtx-Sent radius req to server 'radius-tcp': fd=10, IP=10.1.100.41
(10.1.100.41:1812) code=1 id=40 len=139
[758] __rad_rxtx-Start rad conn timer.
...
[806] __fnbamd_rad_get_next_addr-Next available address of rad 'radius-tcp': 10.1.100.41:1813.
[981] __auth_ctx_start-Connection starts radius-tcp:10.1.100.41, addr 10.1.100.41:1813 proto:
TCP
[449] __rad_tcps_open-vfid 0, addr 10.1.100.41, src_ip (null), ssl_opt 0
...
[1393] create_acct_session-Acct type 6 session created, 0x982b280
[418] __rad_tcps_connect-tcps_connect(10.1.100.41) is established.
[716] __rad_rxtx-fd 10, state 4(Acct)
...
[574] __rad_tcps_send-Sent all. Total 129.
[749] __rad_rxtx-Sent radius req to server 'radius-tcp': fd=10, IP=10.1.100.41
(10.1.100.41:1813) code=4 id=41 len=123
[758] __rad_rxtx-Start rad conn timer.
...
```

RADIUS integrated certificate authentication for Agentless VPN

Secure connections to Agentless VPNs can be established using certificate-based authentication. Access can be granted to the user by using the content inside the Subject Alternative Name (SAN) of the user certificate to authenticate to the RADIUS server. An extra layer of security is added by ensuring that only users with valid certificates can access the VPN.

Certificate-based authentication with RADIUS supports UserPrincipalName (UPN), RFC822 Name (corporate email address) defined in the SAN extension of the certificate, the DNS defined in the user certificate as the unique identifier in the SAN field for peer user certificates, and the Subject Common Name (CN) defined in the certificate.

```
config user radius
    edit <name>
        set account-key-processing {same | strip}
        set account-key-cert-field {othername | rfc822name | dnsname | cn}
    next
end
```

`account-key-processing {same | strip}`	Account key processing operation. The FortiGate will keep either the whole domain or strip the domain from the subject identity. • `same`: Same as subject identity field (default). • `strip`: Strip domain string from subject identity field.
`account-key-cert-field {othername | rfc822name | dnsname}`	Define subject identity field in certificate for user access right checking. • `othername`: match to UPN in SAN (default). • `rfc822name`: match to RFC822 email address in SAN. • `dnsname`: match to DNS name in SAN. • `cn`: match to CN in subject.

The RADIUS server configurations are applied to the user peer configuration when the PKI user is configured.

```
config user peer
    edit <name>
        set ca <string>
        set subject <string>
        set cn <string>
        set mfa-mode subject-identity
        set mfa-server <string>
    next
end
```

When a user authenticates to FortiGate over Agentless VPN, the user presents a user certificate signed by a trusted CA to FortiGate. This CA should also be trusted by the FortiGate. See CA certificate on page 3967 for more information about importing a CA certificate to FortiGate trusted CA store. The following sequence of events occurs as the FortiGate processes the certificate for authentication:

1. The FortiGate checks whether the certificate is issued by a trusted CA. If the CA is not a public CA, FortiGate ensures that the CA certificate is uploaded and trusted by the FortiGate, and applies it to the user

peer configurations (`set ca <string>`).

2. The FortiGate verifies that the CN field of the certificate matches the CN specified in the user peer configurations (`set cn <string>`).
3. If the user peer configuration has `mfa-mode` set to `subject-identity` and the `mfa-server` is configured, then the FortiGate uses a unique identifier in the certificate to authenticate against the RADIUS server.
   - If `account-key-cert-field` is set to `othername` (the default setting), then the FortiGate uses the UPN in the certificate's SAN field to authenticate against RADIUS.
   - If `account-key-cert-field` is set to `rfc822name`, then the FortiGate uses the RFC822 Name in the certificate's SAN field to authenticate against RADIUS.
   - If `account-key-cert-field` is set to `dnsname`, then the FortiGate uses the DNS name in the certificate's SAN field to authenticate against RADIUS.
   - If `account-key-cert-field` is set to `cn`, then the FortiGate uses the CN in the certificate's subject to authenticate against RADIUS.

Some RADIUS servers do not require a password in an Access Request, while others need a valid password to return an ACCESS ACCEPT. If your RADIUS server requires a valid password to return an ACCESS ACCEPT, then you can configure an MFA password for each peer user using the `set mfa-password` command.

When you configure a user MFA password in a user peer, you must need to have a user peer configuration on the FortiGate for each user with `cn=USER`.

Example

In this example, a user certificate is issued to a user by a customer's CA. The certificate is used to authenticate the user to the Agentless VPN web portal. The administrator uses the RFC822 Name in the SAN field to authenticate against their corporate RADIUS. The Active Directory mail attribute is used to check against the RFC822 Name field.

The configuration used in this example assumes the following:

- The CA certificate has already been uploaded to the FortiGate.
- Agentless VPN has already been configured, pending the assignment of the PKI user group.

To configure the authentication settings:

1. Configure the RADIUS server:

```
config user radius
    edit "NPS-MFA"
        set server "172.18.60.214"
        set secret  XXXXXXXXXX
        set auth-type pap
        set password-encoding ISO-8859-1
        set account-key-processing strip
        set account-key-cert-field rfc822name
    next
end
```

2. Configure the local peer user:

```
config user peer
    edit "peer2"
        set ca "CA_Cert_1"
        set subject "L = Burnaby"
        set cn "test2"
        set mfa-mode subject-identity
        set mfa-server "NPS-MFA"
    next
end
```

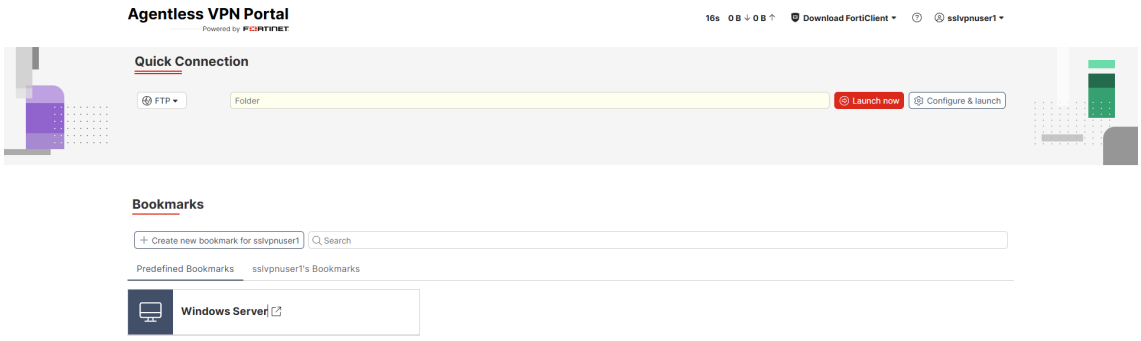
3. Configure the firewall user group for Agentless VPN authentication:

```
config user group
    edit "sslvpn-mfa"
        set member  "peer2"
    next
end
```

4. Apply the user group to the Agentless VPN configuration and firewall policy.

To verify the configuration:

When a user authenticates to Agentless VPN using their browser, the FortiOS fnbamd daemon first validates the certificate supplied by the user. If the certificate check is successful, the information in the SAN field of the user certificate is used to find a matching user record on the RADIUS server. See Agentless VPN on page 3349 for information about configuring Agentless VPN.



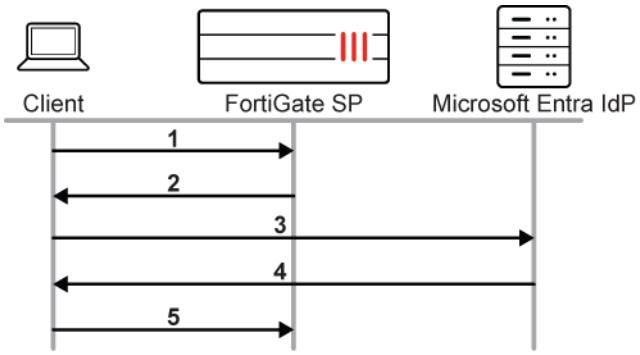
SAML

SAML authentication allows a user to complete authentication and authorization between a Service Provider (SP) and an Identity Provider (IdP). It enables Single Sign-On (SSO) by allowing users who have been authenticated against an IdP to be allowed access to different applications (the SPs providing a service such as a web application).

A FortiGate (SP) can provide a web service, such as an Agentless VPN connection, that requires users to be authenticated through SAML. The user identities for the company can be stored remotely in an IdP, such as Microsoft Entra ID. Other applications might also require authentication from the same IdP, so after a user is

authenticated once, the same SAML assertion carrying user and group information can be used to authorize the user access to the FortiGate Agentless VPN, as well as other applications.

To illustrate the communication between the user or browser, SP, and IdP:



1. The user initiates an Agentless VPN connection to the FortiGate.
2. The FortiGate SP redirects the user to the SAML IdP.
3. The user connects to the Microsoft log in page for the SAML authentication request.
4. The SAML IdP authenticates the user and sends the SAML assertion containing the user and group within the SAML response.
5. The browser forwards the SAML response to the FortiGate SP. If the user and group are allowed by the FortiGate, the user is allowed to access the application, in this case, connecting to Agentless VPN.

Usage

There are many practical uses and applications for SAML authentication on the FortiGate. For example:

- Authentication for Agentless VPN
- Authentication for IPsec VPN (with compatible FortiClient endpoint 7.2.4 and above)
- Firewall authentication for firewall policy access
- Authentication for ZTNA
- Authentication for Explicit Proxy
- Authentication for Administrative Access

Identity providers

FortiGate's SAML SSO configurations can be integrated with any common Identity providers, such as Microsoft Entra ID, Okta, Google Workspace, Onelogin, and others. You can also use FortiAuthenticator as an identity provider with local or remote user integration, or as an IdP Proxy to other IdP providers.

For more information on using FortiAuthenticator, see the FortiAuthenticator Administration Guide and FortiAuthenticator Examples Guide.

Response and assertion signing

For security purposes, FortiGate can perform signature verification for both the SAML response message and the SAML assertion carried inside the SAML response. This means that the SAML response must have a valid signature, and the SAML assertion must also have a valid signature. If the Identity Provider (IdP) provides an invalid signature, or fails to sign one of these, the FortiGate will reject the SAML response.

This check is disabled by default but can be enabled using the following configuration:

```
config user saml
    edit <name>
        set require-signed-resp-and-asrt {enable | disable}
    next
end
```

Option	Description
`enable`	Both response and assertion must be signed and valid.
`disable`	At least one of response or assertion must be signed and valid (default).

When disabled, the FortiGate performs a signature check using the following logic:

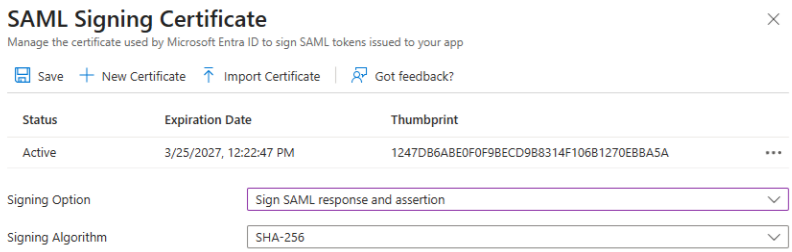
1. If the SAML response is missing signature, each SAML assertion must have a valid signature.
2. If the SAML response signature is invalid, check fails.
3. If the SAML response signature is valid, check the following:
   a. If the SAML assertion signature is missing, check passes.
   b. If the SAML assertion signature is invalid, check fails.
   c. If the SAML assertion signature is valid, check passes.

Example

Some SAML IdPs defaults to signing the SAML assertions only. To increase security, you can enable signing both the SAML response and assertion to work natively with the FortiGate.

To enable Sign SAML response and assertion in Microsoft Entra ID:

1. Go to the *Microsoft Entra admin center*.
2. Navigate to *Enterprise Applications*.
3. Select your application.
4. Go to *Single sign-on* > *SAML*.
5. Scroll to *SAML Signing Certificate* and click *Edit* (the pencil icon).
6. For *Signing Option*, choose *Sign SAML response and assertion*.



7. Click *Save*.
   For details, see Certificate signing options in a SAML token - Microsoft Entra ID | Microsoft Learn.

For other SAML IdP vendors, check the setting to enable signing both the SAML response and assertion.

If the SAML IdP vendor does not have a setting to sign both the SAML response and assertion, or fails to sign both even when configured, you can considering loosening up the requirement on the FortiGate per Response and assertion signing on page 3575.

Configuring SAML SSO

SAML Single Sign-On (SSO) can be configured from the GUI or CLI. The configurations allow administrators to set up the FortiGate as a SAML Service Provider (SP) while inputting the necessary settings for the Identity Provider (IdP).

There are many use cases for applying SAML authentication, as explained in the SAML introduction. For each use case, the configuration steps vary slightly. In general, to successfully configure SAML authentication for an application, you will need to perform the following:

1. Obtain IdP configurations from the Identity Provider. This is outside the scope of the FortiGate.
2. Create a Single Sign-On object in *User & Authentication* > *Single Sign-On*.
3. Apply the FortiGate SP URLs to the IdP.
4. Install appropriate IdP and SP certificates.
5. Configure user group with the SSO object as member.

After these steps are completed, the user group object can be applied to whatever type of policy is applicable to the use case.

Common SAML SSO settings

Configuring the IdP is outside the scope of this topic, but to successfully configure SAML on the FortiGate the following information must be obtained from the Identity Provider:

From IdP	Description
Entity ID	The identifier URL for the IdP used to identify the issuer of the SAML response or assertion.
Assertion consumer service (ACS) URL	The ACS URL, sometimes called the Login URL, informs the SP and end user where to send the Login request to the IdP.

From IdP	Description
Single logout service URL	The Single logout service URL, sometimes called the Logout URL, informs the SP and end user where to send the Logout request to the IdP.
SAML Signing Certificate	The certificate used to sign the SAML response originating from the IdP. This must be trusted by the SP in order to verify the identity of the messages from the IdP. To upload a remote certificate from the IdP, follow the instructions in Remote certificate on page 3968. Starting in FortiOS 7.6.4, the SAML IdP must sign both the SAML assertion and response. See Identity providers on page 3574.

At the same time, to complete the configurations on the IdP, it will require information about the SP from the FortiGate. The following describes the settings configured on the FortiGate, including the information needed for the IdP configuration.

To configure the FortiGate SP settings for SSO in the GUI:

1. Go to *User & Authentication* > *Single Sign-On* and click *Create new*.
2. Configure the SP settings:

Setting	Description
Name	Name of the SSO object.
Address	FQDN or IP address that clients will be connecting to. If this requires a non-standard port (eg. 443), specify the port in this format *<address>:<port>*.
Entity ID	The identifier URL for the SP used to identify the issuer of the SAML request. This URL must be provided to the IdP. Modifying the URL must be done in CLI.
Assertion consumer service URL	The ACS URL, sometimes referred to as the reply URL or the single sign-on URL, informs the IdP and end user the URL to send the SAML Assertion for login to. This URL must be provided to the IdP. Modifying the URL must be done in CLI.
Single logout service URL	The logout URL informs the IdP and end user the URL to send the request to logout to. This URL must be provided to the IdP. Modifying the URL must be done in CLI.
Certificate	The certificate used to sign the SAML messages originating from the SP to the IdP. This is typically an optional configuration.

3. Click *Next*.
4. Configure the IdP settings:

Setting	Description
Type	• *Fortinet Product*: If the IdP is a FortiAuthenticator or FortiTrust-ID,

Setting	Description
	IdP configurations are simplified. See FortiAuthenticator Admin Guide > Authentication > SAML IdP for more information • *Custom*: If the IdP is any other vendor, or you want to configure each field manually, select this option.
Fortinet Product setup	
Address	Enter the address of the FortiAuthenticator or FortiTrust-ID that users will access to authenticate to the IdP.
Prefix	Enter the prefix specified by the FortiAuthenticator or FortiTrust-ID.
Certificate	Select the SAML Signing certificate from the IdP. If this is not yet uploaded, use the *Import* option to import the remote certificate.
Custom setup	
Entity ID	Input the Entity ID URL from the IdP. See Entity ID on page 3576.
Assertion consumer service URL	Input the ACS URL from the IdP. See Assertion consumer service (ACS) URL on page 3576.
Single logout service URL	Input the Single logout service URL from the IdP. See Single logout service URL on page 3577.
Certificate	Select the SAML Signing certificate from the IdP. If this is not yet uploaded, use the *Import* option to import the remote certificate. See SAML Signing Certificate on page 3577.
Additional SAML Attributes	
AD FS claim	This setting is only available after the initial SSO object has been configured. Enable this setting to select the attribute names based on Active Directory Federated Services (AD FS) claim types.
User claim type	Select the AD FS claim type that will be used to match the user within the SAML assertion statement.
Group claim type	Select the AD FS claim type that will be used to match the group within the SAML assertion statement.
Attribute used to identify users	Specify the name of the attribute for a user within the SAML assertion statement. This value is case sensitive. If AD FS claim is enabled, this field will be auto-populated to reflect the claim type.
Attribute used to identify groups	Specify the name of the attribute for a group within the SAML assertion statement. This value is case-sensitive. If AD FS claim is enabled, this field will be auto-populated to reflect the claim type.

5. Click *Submit*.

To configure the FortiGate SP settings for SSO in the CLI:

```
config user saml
    edit <name>
        set adfs-claim [enable|disable]
        set cert {string}
        set clock-tolerance {integer}
        set digest-method [sha1|sha256]
        set entity-id {string}
        set group-claim-type [email|given-name|...]
        set group-name {string}
        set idp-cert {string}
        set idp-entity-id {string}
        set idp-single-logout-url {string}
        set idp-single-sign-on-url {string}
        set limit-relaystate [enable|disable]
        set reauth [enable|disable]
        set single-logout-url {string}
        set single-sign-on-url {string}
        set user-claim-type [email|given-name|...]
        set user-name {string}
    next
end
```

Setting	Description
`adfs-claim`	See AD FS claim on page 3578.
`cert`	The SP certificate used to sign SAML messages.
`clock-tolerance`	A SAML assertion is only valid for a specific duration. When the FortiGate SP and the SAML IdP clocks are not in synchronization, use clock-tolerance to define the number of seconds that the skew in time is tolerated. The setting is only available in the CLI.
`digest-method`	The type of hash used to compute the hash value of the content of the SAML assertion. The setting is only available in the CLI.
`entity-id`	The SP Entity ID.
`group-claim-type`	Specify the group claim type when `adfs-claim` is enabled.
`group-name`	The attribute used to identify a group within the SAML assertion statement.
`idp-cert`	The SAML Signing certificate from the IdP.
`idp-entity-id`	The Entity ID from the IdP.
`idp-single-logout-url`	The Single logout service URL from the IdP.
`idp-single-sign-on-url`	The ACS URL, sometimes called the Login URL, from the IdP.
`limit-relaystate`	Enable/disable limiting the relay-state parameter when it exceeds SAML 2.0 specification limits (80 bytes).

Setting	Description
	The setting is only available in the CLI.
`reauth`	Enable/disable signaling the IdP to force user re-authentication. The SAML AuthnRequest is set to ForceAuthn=”true”. The setting is only available in the CLI.
`single-logout-url`	The Single logout service URL from the SP.
`single-sign-on-url`	The ACS URL, sometimes referred to as the reply URL or the single sign-on URL, from the SP.
`user-claim-type`	Specify the user claim type when `adfs-claim` is enabled.
`user-name`	The attribute used to identify a user within the SAML assertion statement.

Other SAML related global settings

Authentication port

By default, the FortiGate listens on port 1003 for incoming authentication requests when traffic matches an identity based firewall policy. As a SAML SP with an identity based firewall policy configured for the SAML user group, the FortiGate will use the same port to listen for SAML authentication requests and redirect them to the IdP.

To change the default port:

```
config system global
    set auth-https-port <port>
end
```

Configuring the user authentication setting

When the FortiGate receives an authentication request in an identity based firewall policy, the authentication daemon uses a local server certificate to secure the connection. The client making the authentication request must trust the certificate presented by the FortiGate that is acting as the TLS server.

In SAML authentication, when a user initiates traffic to the SP, the traffic matches the identity based firewall policy which triggers the authentication request to hit the authentication daemon. The server certificate used by the authentication daemon must be trusted by the user, otherwise they will receive a certificate warning. To avoid a certificate warning, use a custom certificate that the user trusts.

To configure a custom certificate in the GUI:

1. Go to *User & Authentication* > *Authentication Settings*.
2. Set *Certificate* to the custom certificate.
   If the certificate is not available, click *Create* to create or import a new custom certificate.
   The custom certificate's SAN field should have the FQDN or IP address from the SP URL.

To configure a custom certificate in the CLI:

```
config user setting
    set auth-cert <custom certificate name>
end
```

Alternatively, assigning a CA certificate allows the FortiGate to automatically generate and sign a certificate for the authentication daemon. This will override any assigned server certificate.

To assign a CA certificate:

1. Edit the user setting :

```
config user setting
    set auth-ca-cert <CA certificate name>
end
```

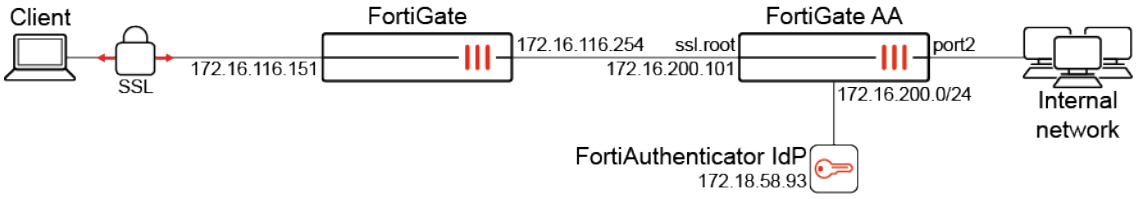
2. Go to *System* > *Certificates* and download the certificate.
3. Install the certificate into the client's certificate store.

Agentless VPN with FortiAuthenticator as a SAML IdP

A FortiGate can act as a SAML service provider (SP) for Agentless VPN that requests authentication from a SAML identity provider (IdP), such as Entra ID, Okta, Fortinet's FortiAuthenticator, or others. The following example shows the use of FortiAuthenticator as the IdP.

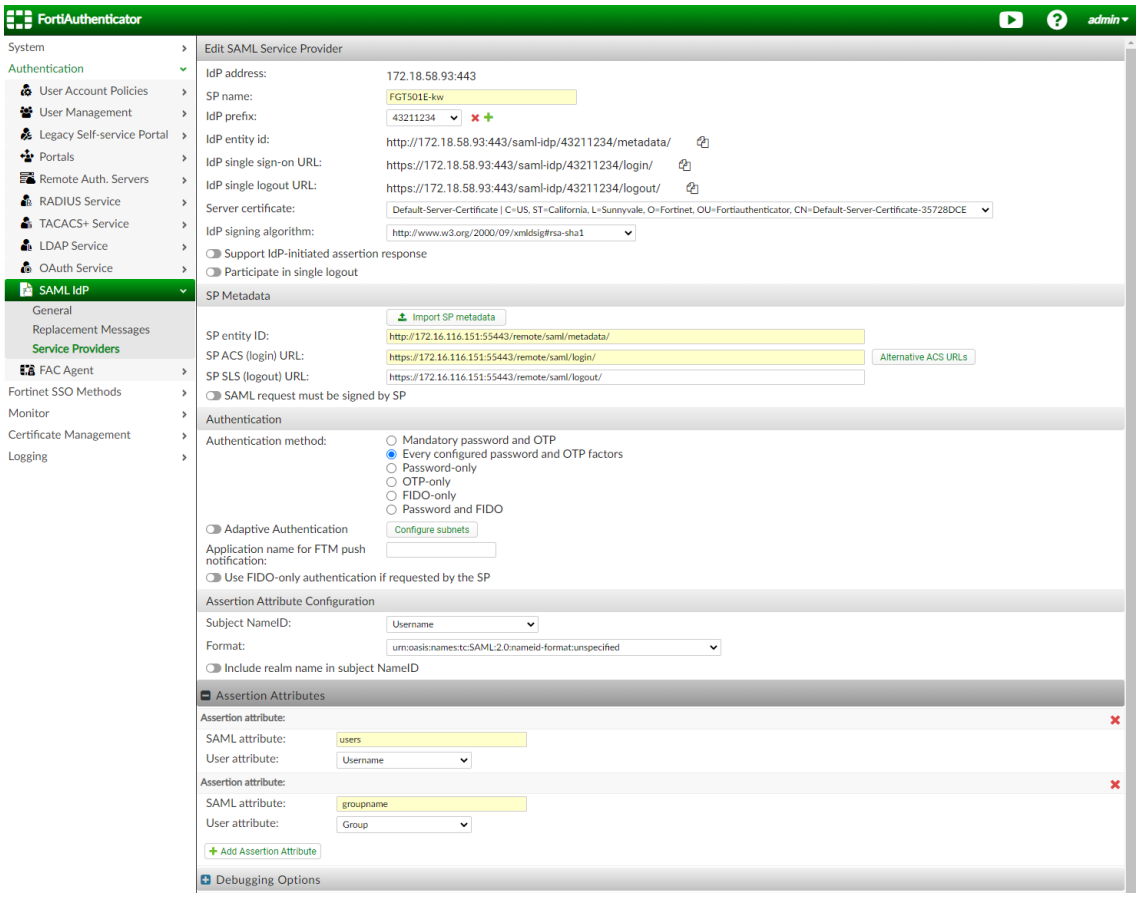
Using FortiAuthenticator as an IdP

In this example, FortiGate AA is the inside firewall (172.16.200.101). The other FortiGate is the outside firewall that only does port forwarding from 172.16.116.151:55443 to 172.16.200.101:443. FortiGate AA is configured to allow full Agentless VPN access to the network in port2.



This Agentless VPN portal allows users from the user group *saml_grp* and SAML server *saml_test* to log in. The FortiAuthenticator acts as the SAML identity provider (IdP), while the FortiGate is the SAML SP. External users are directed to the FortiAuthenticator IdP login URL to authenticate.

The FortiAuthenticator in this example has the following configuration:

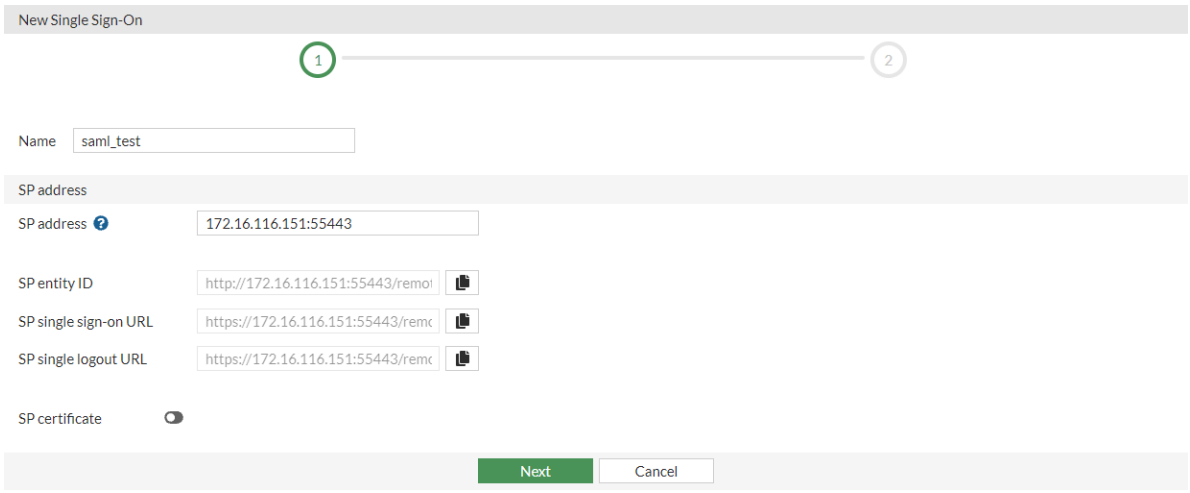


For a deep-dive into how to configure FortiAuthenticator as an IdP, including integration with Windows AD via LDAP for user authentication, see the FortiGate Agentless VPN with FortiAuthenticator as SAML IdP section in the FortiAuthenticator Examples Guide.

This example also demonstrates using FortiAuthenticator to act as a root CA to sign certificates for the SP, IdP, and Agentless VPN portal.

To configure FortiGate AA as an SP:

1. Create a new SAML server entry:
   a. Go to *User & Authentication* > *Single Sign-On* and click *Create New*. The single-sign on wizard opens.
   b. Enter a name (*saml_test*). The other fields will automatically populate based on the FortiGate's WAN IP and port.



Click the icon beside the *SP entity ID*, *SP single sign-on URL*, and *SP single logout URL* fields to copy the text.

c. Click *Next*.

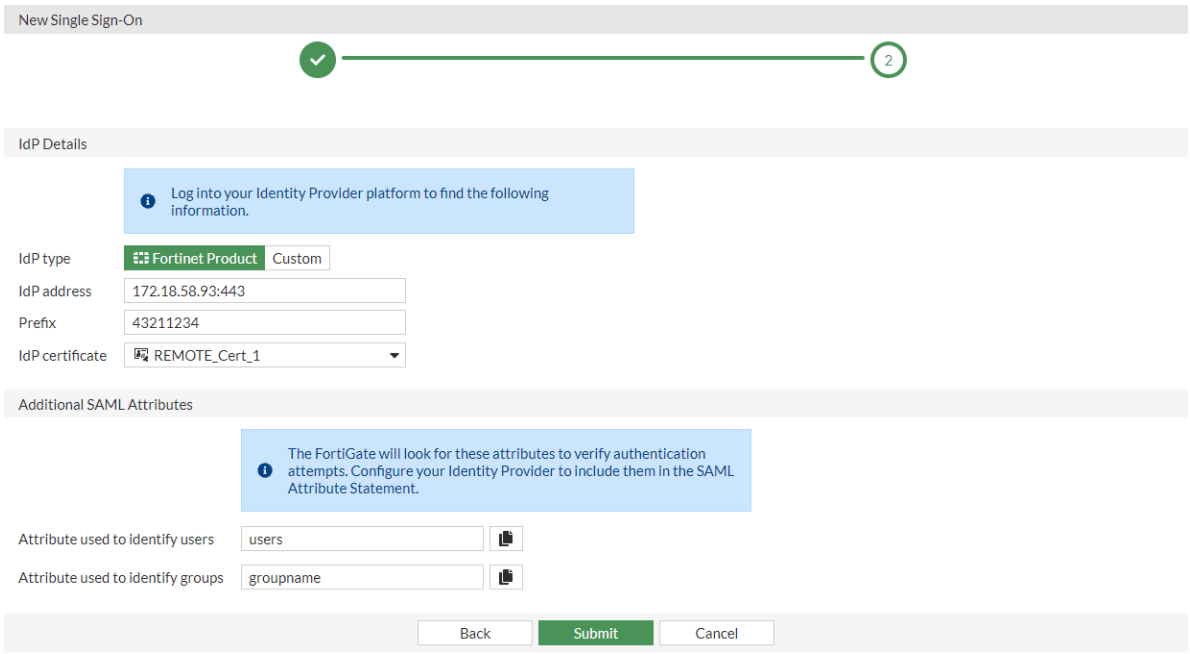
d. Enter the FortiAuthenticator IdP details:

IdP address	172.18.58.93:443
Prefix	43211234
IdP certificate	REMOTE_Cert_1

e. Enter the additional SAML attributes that will be used to verify authentication attempts:

Attribute used to identify users	users
Attribute used to identify groups	groupname

The IdP must be configured to include these attributes in the SAML attribute statement. In FortiAuthenticator, this is configured in the *Assertion Attributes* section.



f. Click *Submit*.

The following is created in the backend:

```
config user saml
    edit "saml_test"
        set cert "fgt_gui_automation"
        set entity-id "http://172.16.116.151:55443/remote/saml/metadata/"
        set single-sign-on-url "https://172.16.116.151:55443/remote/saml/login/"
        set single-logout-url "https://172.16.116.151:55443/remote/saml/logout/"
        set idp-entity-id "http://172.18.58.93:443/saml-idp/43211234/metadata/"
        set idp-single-sign-on-url "https://172.18.58.93:443/saml-idp/43211234/login/"
        set idp-single-logout-url "https://172.18.58.93:443/saml-idp/43211234/logout/"
        set idp-cert "REMOTE_Cert_1"
        set user-name "users"
        set group-name "groupname"
        set digest-method sha1
    next
end
```

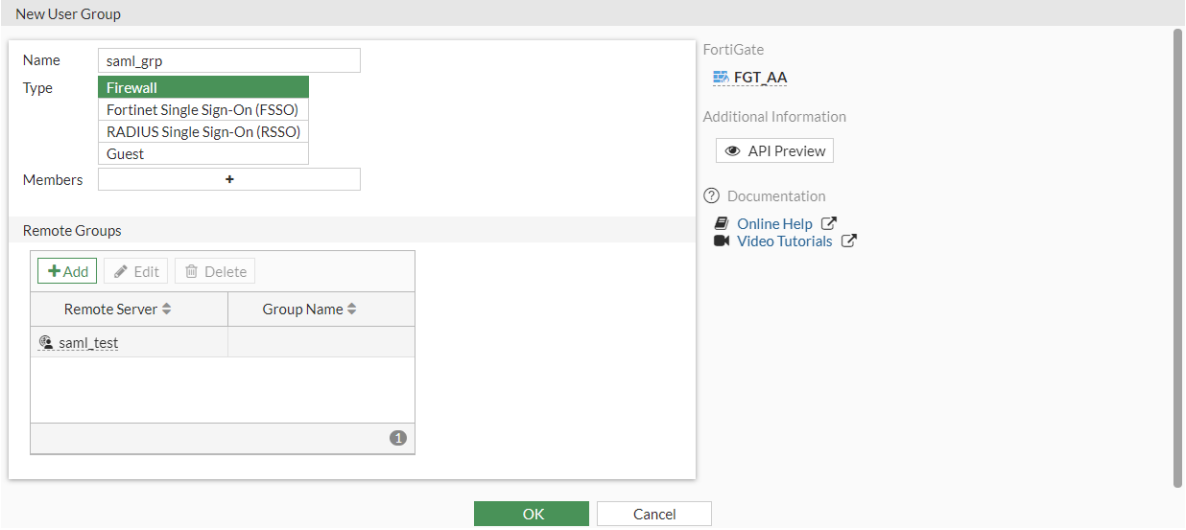
2. Create the SAML group:

a. Go to *User & Authentication* > *User Groups* and click *Create New*.

b. Enter a name, *saml_grp*.

c. In the *Remote Groups* table, click *Add*.

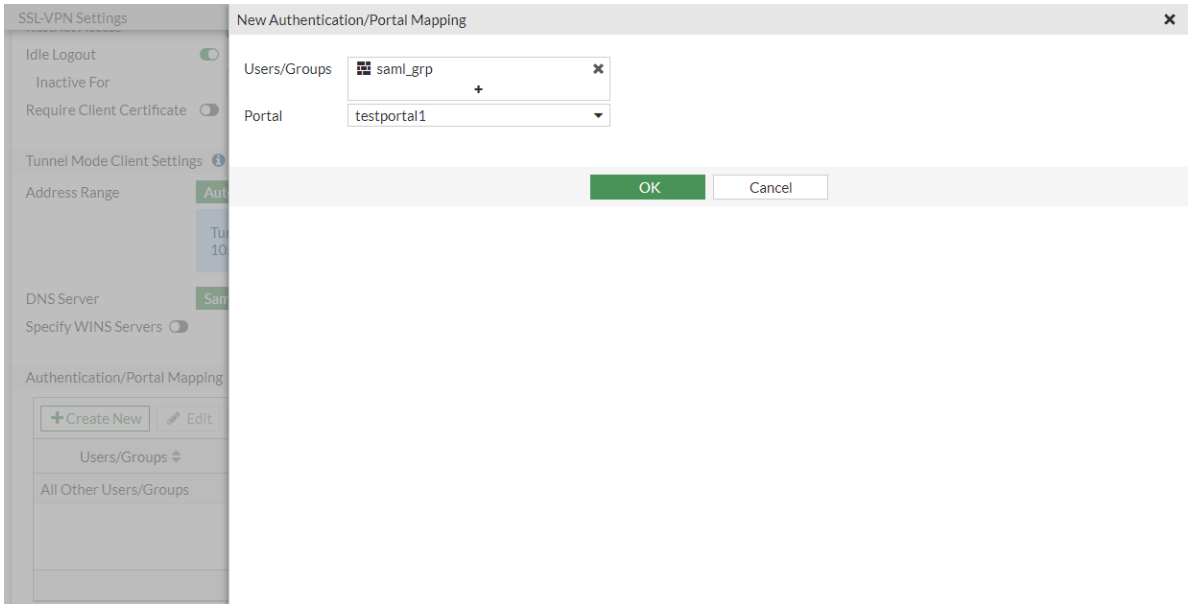
d. In the *Remote Server* dropdown, select *saml_test* and click *OK*.



e. Click *OK*.
The following is created in the backend:

```
config user group
    edit "saml_grp"
        set member "saml_test"
    next
end
```

3. Add the SAML group in the Agentless VPN settings:
   a. Go to *VPN > Agentless VPN Settings*.
   b. In the *Authentication/Portal Mapping* table, click *Create New*.
   c. For *Users/Groups*, click the + and select *saml_grp*.
   d. Select the *Portal* (*testportal1*).
   e. Click *OK*.



f. Click *Apply*.

4. Configure the firewall policy:
   a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
   b. Enter the following:

Incoming Interface	ssl.root
Outgoing Interface	port2
Source	all, saml_grp, saml_test

   c. Configure the other settings as needed.
   d. Click *OK*.
5. On the client, log in with SAML using the Agentless VPN web portal.
6. In FortiOS, go to *Dashboard* > *Agentless VPN Monitor* to expand to full view and verify the connection information.

   If the Agentless VPN Monitor is hidden, click + under the *Dashboard*, search for *Agentless VPN Monitor*, and add it to the display.

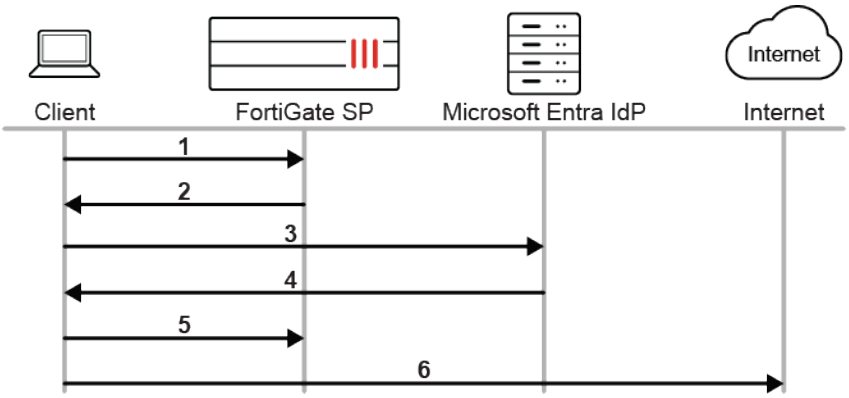
IPsec VPN with SAML IdP

For information about configuring IPsec VPN with SAML IdP, see SAML-based authentication for FortiClient remote access VPNs on page 3124.

Outbound firewall authentication with Microsoft Entra ID as a SAML IdP

In this example, users are managed through Microsoft Entra ID (formerly Azure Active Directory). The FortiGate is configured for SSO firewall authentication for outbound traffic, with authentication performed by the Microsoft Entra ID as a SAML identity provider (IdP).

The SAML interaction occurs as follows:

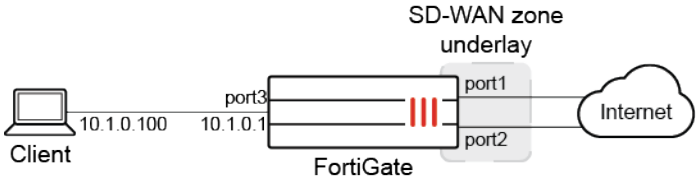


1. The user initiates web traffic to the internet.
2. The FortiGate redirects to the local captive portal port (default is 1003), then redirects the user to the SAML IdP.
3. The user connects to the Microsoft log in page for the SAML authentication request.
4. The SAML IdP sends the SAML assertion containing the user and group.
5. The browser forwards the SAML assertion to the SAML SP.
6. If the user and group are allowed by the FortiGate, the user is allowed to access the internet.

In this example environment, a user is added in the Microsoft Entra ID belonging to the security group called Firewall.

- Username: John Locus
- User login: jlocus@azure.kldocs.com
- Group: Firewall (ID 62b699ce-4f80-48c0-846e-c1dfde2dc667)

The goal is to allow users in the Firewall group to access the internet after passing firewall authentication.



Configuring the Microsoft Entra ID

The following Microsoft Entra ID configuration demonstrates how to add the FortiGate as an enterprise non-gallery application. This application provides SAML SSO connectivity to the Microsoft Entra IdP. Some steps are performed concurrently on the FortiGate.

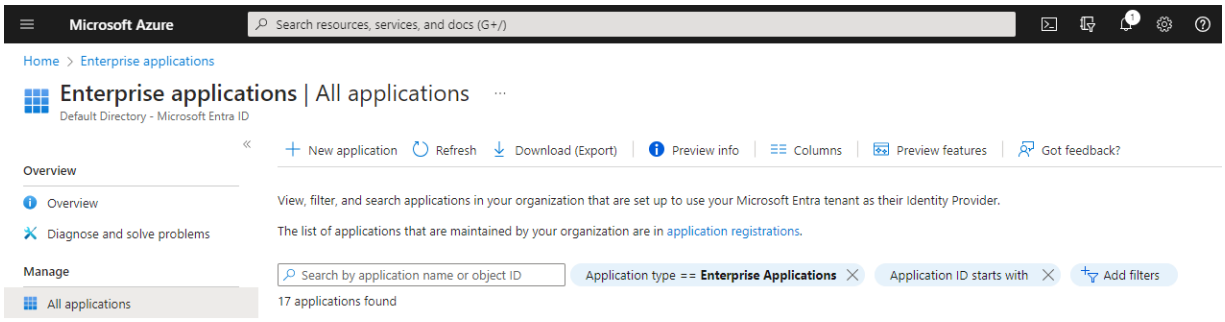
> This example is configured with an Microsoft Entra ID free-tier directory. There may be limitations to managing users in Azure in this tier that are not limited in other tiers. Consult the Microsoft Entra ID documentation for more information.

There are three steps to configure the Microsoft Entra ID:

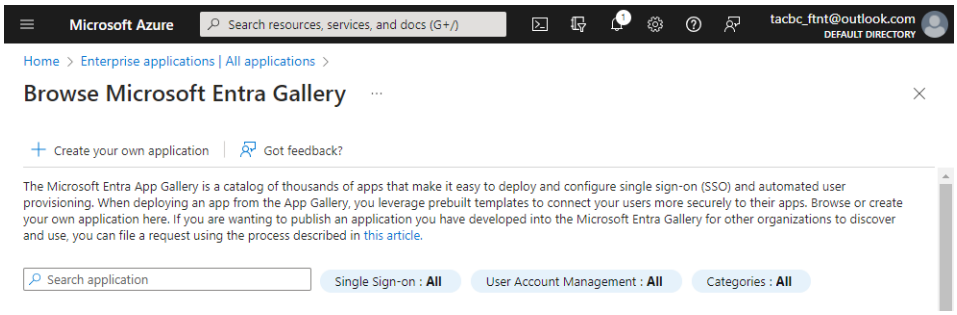
1. Create a new enterprise application.
2. Configure the SAML SSO settings on the application and FortiGate.
3. Assign Microsoft Entra ID users and groups to the application.

To create a new enterprise application:

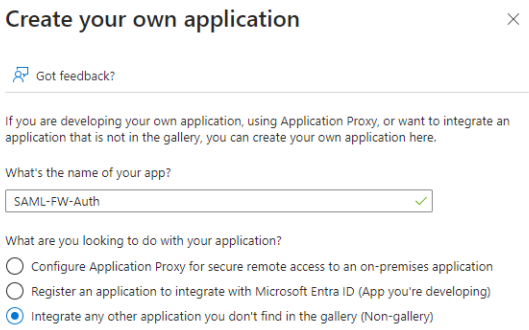
1. Log in to the Azure portal.
2. In the Azure portal menu, click *Microsoft Entra ID*.
3. In the left-side menu go *Manage* > *Enterprise applications*.
4. Click *New application*.



5. Click *Create your own application*.



6. Enter a name for the application (*SAML-FW-Auth*) and select *Integrate any other application you don't find in the gallery (Non-gallery)*.

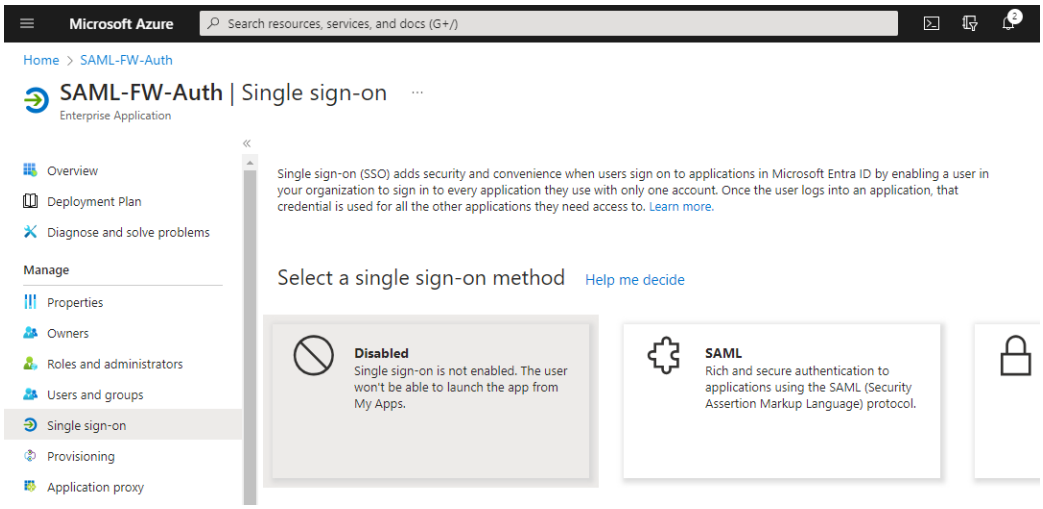


7. Click *Create*.

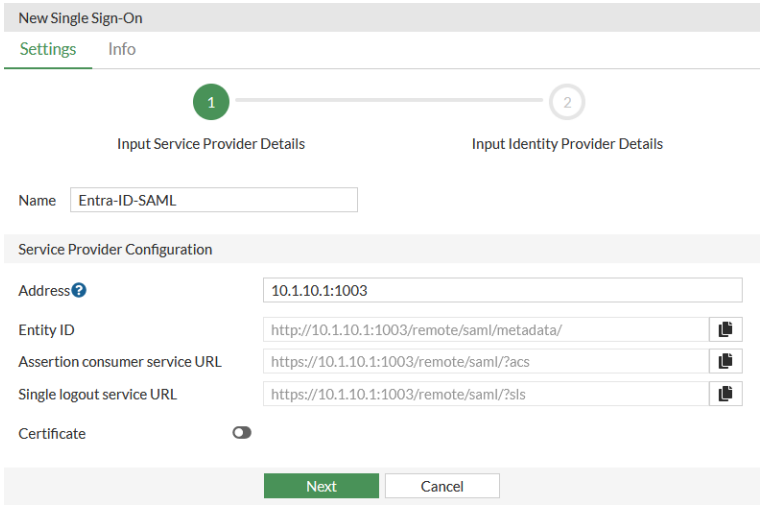
To configure the SAML SSO settings on the application and FortiGate:

This procedure requires going back and forth between Azure and the FortiGate GUI. Leave the FortiGate GUI open for the entire procedure.

1. On the *Overview* page for your new application, go to *Manage* > *Single sign-on* and select *SAML* as the single sign-on method.



2. The *Basic SAML Configuration* section in Azure describes the SAML SP entity and links that Azure will reference. Configure these settings on the FortiGate by creating a new SAML server object and defining the SP address. The SP (IP or FQDN) address should be accessible by the user who is authenticating against the firewall. The port used should match the port used by the FortiGate firewall authentication captive portal. By default, this is port 1003 for HTTPS. A captive portal does not need to be configured separately.
   a. Go to *User & Authentication > Single Sign-On* and click *Create New*.
   b. Enter a *Name* for the SAML object, *Entra-ID-SAML*.
   c. Enter the *SP address*, *10.1.0.1:1003*. The three SP URLs are automatically populated.

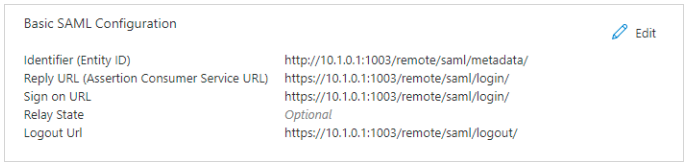


3. In Azure on the *Set up Single Sign-On with SAML* page, copy the following URLs from the FortiGate to the *Basic SAML Configuration* section:

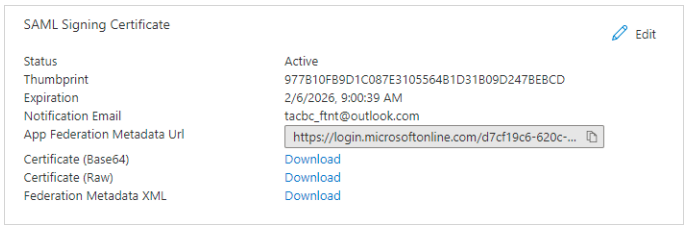
From FortiGate	To Azure field
Entity ID (http://10.1.0.1:1003/remote/saml/metadata/)	*Identifier (Entity ID)*, set to *Default*
Assertion consumer service URL (https://10.1.0.1:1003/remote/saml/login/)	*Reply URL* and *Sign on URL*

From FortiGate	To Azure field
Single logout service URL (https://10.1.0.1:1003/remote/saml/logout/)	*Logout URL*

4. Click *Save*.



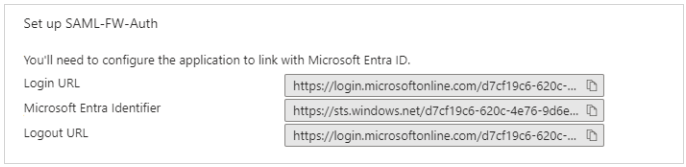
5. Under the *SAML Signing Certificate* section, download the Base64 certificate.



6. Import the certificate from Azure on the FortiGate as the IdP certificate:
   a. Go to *System* > *Certificates* and click *Create/Import* > *Remote Certificate*.
   b. Upload the certificate from Azure and click *OK*. The new certificate appears under the *Remote Certificate* section with the name *REMOTE_Cert_(N)*.
   c. Optionally, rename the certificate in the CLI to give it a more recognizable name:

```
config vpn certificate remote
    rename REMOTE_Cert_3 to ENTRA_ID_SAML_FW
end
```

7. In the *Set up <application name>* section, copy the URLs from Azure to the FortiGate in the *IdP Details* section:

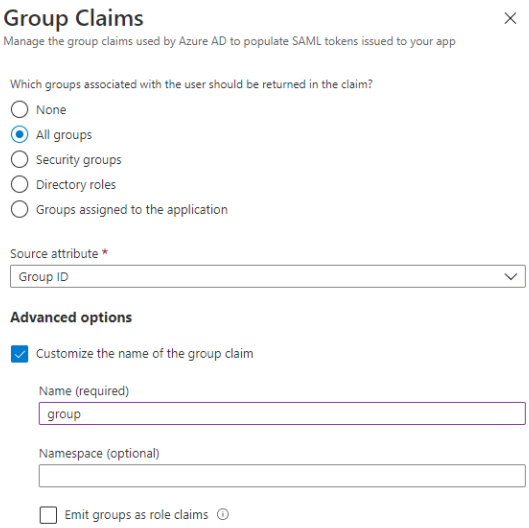


   a. On the FortiGate, click *Next*.
   b. For *IdP type*, select *Custom* and copy the following from Azure to the corresponding field:

From Azure	To FortiGate field
Microsoft Entra Identifier	*Entity ID*
Login URL	*Assertion consumer service URL*
Logout URL	*Single logout service URL*

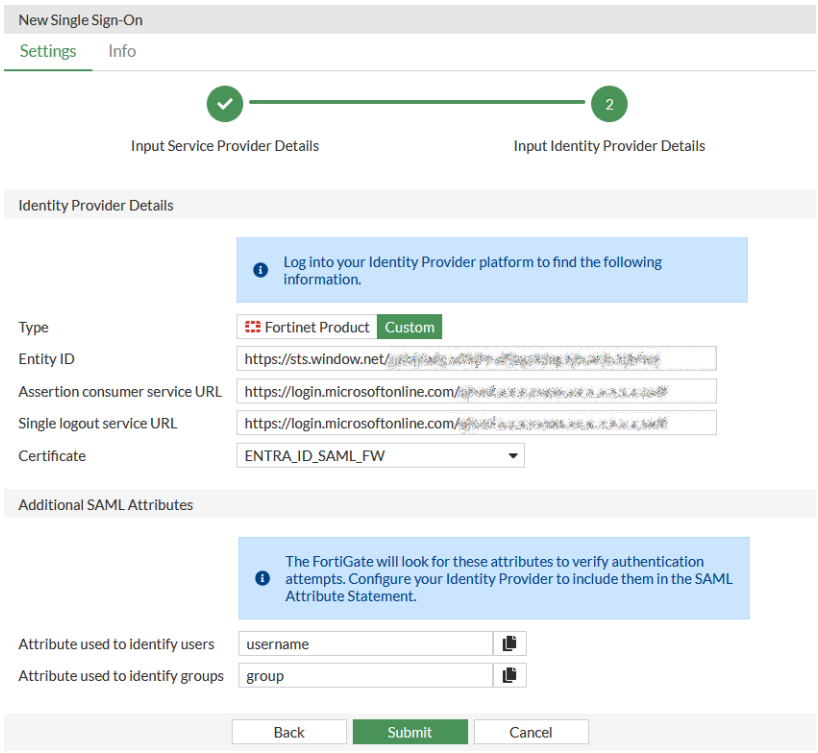
   c. For *Certificate*, select the remote certificate imported earlier.
8. In Azure, edit the *User Attributes & Claims* section. The attributes are returned in the SAML assertion, which the FortiGate uses to verify the user and group. Configuring group matching is optional.

a. Click *Add new claim*, name it *username*, and set the *Source attribute* to *user.displayname*. The source attribute can be any of the related username fields. The value of the username returned to the FortiGate will be used in logs and monitors to identify the user.
b. Click *Save*.
c. Click *Add a group claim* and in the *Group Claims* pane, select *All groups*.
d. In *Advanced Options*, select *Customize the name of the group claim*. Set the name to *group*.



e. Click *Save*. The *User Attributes & Claims* section displays the update settings.

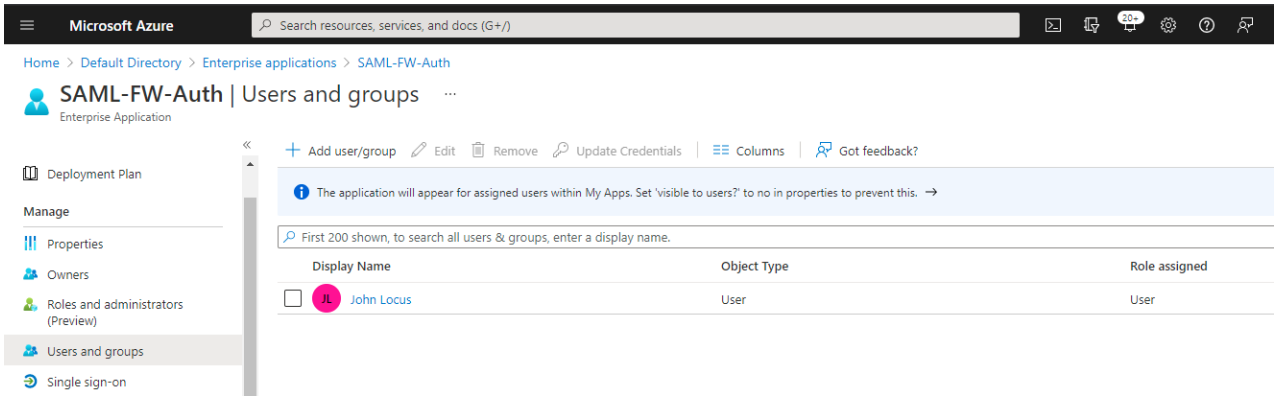
9. On the FortiGate, update the *Additional SAML Attributes* section with the username and group created in Azure:
   a. For *Attribute used to identify users*, enter *username*.
   b. For *Attribute used to identify groups*, enter *group*.



c. Click *Submit*.

To assign Microsoft Entra users and groups to the application:

1. In Azure, go to the application's *Overview* page.
2. Go to *Manage* > *Users and groups* and click *Add user/group*.
3. Click *Users* to select the users or groups (*John Locus* is selected in this example).
4. Click *Assign* to add the assignment.



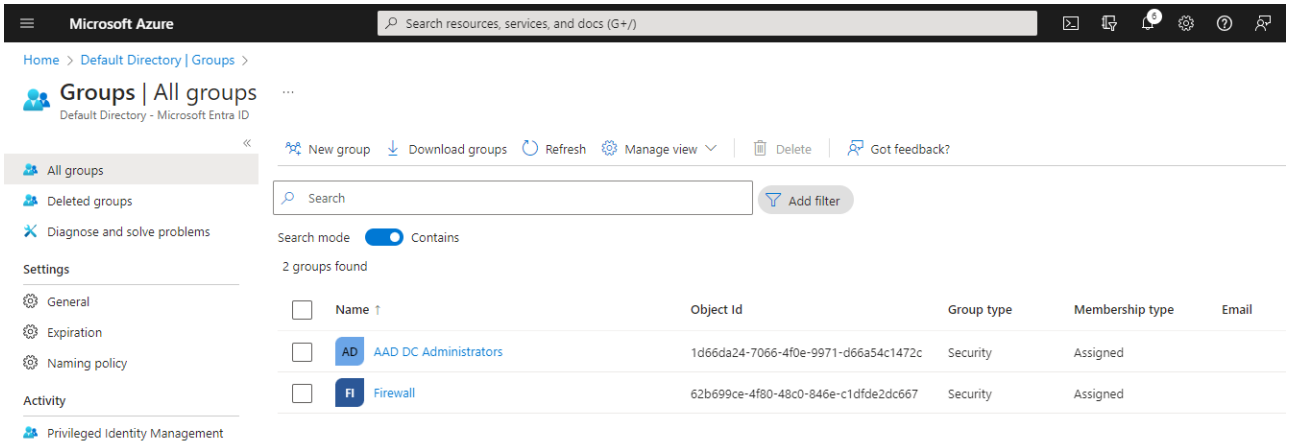
Configuring the FortiGate

The user group, user authentication settings, and firewall policies must be configured on the FortiGate.

Configuring the user group

A user group named *Azure-FW-Auth* is created with the member *Entra-ID-SAML*.

Configuring group matching is optional, and the *Object ID* from Azure is needed for the `config match` settings. In the Azure default directory, go to *Manage* > *Groups* and locate the *Object ID* for the *Firewall* group.



To configure the user group:

```
config user group
    edit "Azure-FW-Auth"
        set member "Entra-ID-SAML"
        config match
            edit 1
                set server-name "Entra-ID-SAML"
                set group-name "62b699ce-4f80-48c0-846e-c1dfde2dc667"
            next
        end
    next
end
```

Configuring the user authentication setting

When a user initiates traffic, the FortiGate will redirect the user to the firewall authentication captive portal before redirecting them to the SAML IdP portal. After the SAML IdP responds with the SAML assertion, the user is again redirected to the firewall authentication captive portal. If the firewall portal's certificate is not trusted by the user, they will receive a certificate warning. Use a custom certificate that the user trusts to avoid the certificate warning.

To configure a custom certificate:

1. Go to *User & Authentication* > *Authentication Settings*.
2. For *Certificate*, select the custom certificate. The custom certificate's SAN field should have the FQDN or IP from the SP URL.

Alternatively, assigning a CA certificate allows the FortiGate to automatically generate and sign a certificate for the portal page. This will override any assigned server certificate. In this example, the built-in Fortinet_CA_SSL is used.

To assign a CA certificate:

1. Edit the user setting:

```
config user setting
    set auth-ca-cert "Fortinet_CA_SSL"
end
```

2. Go to *System* > *Certificates* and download the certificate.
3. Install the certificate into the client's certificate store.

Configuring the firewall policies

Firewall policies must be configured to apply user authentication and still allow users behind the FortiGate to access the Microsoft log in portal without authentication.

To configure the firewall policies:

1. Configure a policy to allow traffic to the Microsoft Azure internet service:
   a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
   b. Enter the following:

Name	*LAN-to-AuthPortal*
Incoming Interface	*port3*
Outgoing Interface	*Underlay*
Source	*all*
Destination	*Microsoft-Azure* (under *Internet Service*)
Schedule	*always*
Service	*ALL*
Action	*ACCEPT*
NAT	Enable and select *NAT*.
Log Allowed Traffic	Enable and select *All Sessions*.

   c. Configure the other settings as needed.
   d. Click *OK*.
2. Configure a policy to apply user authentication:

a. Click *Create New* and enter the following:

Name	*LAN-auth-policy*
Incoming Interface	*port3*
Outgoing Interface	*Underlay*
Source	*all*, *Azure-FW-Auth*
Destination	*all*
Schedule	*always*
Service	*ALL*
Action	*ACCEPT*
NAT	Enable and select *NAT*.
Log Allowed Traffic	Enable and select *All Sessions*.

b. Configure the other settings as needed.

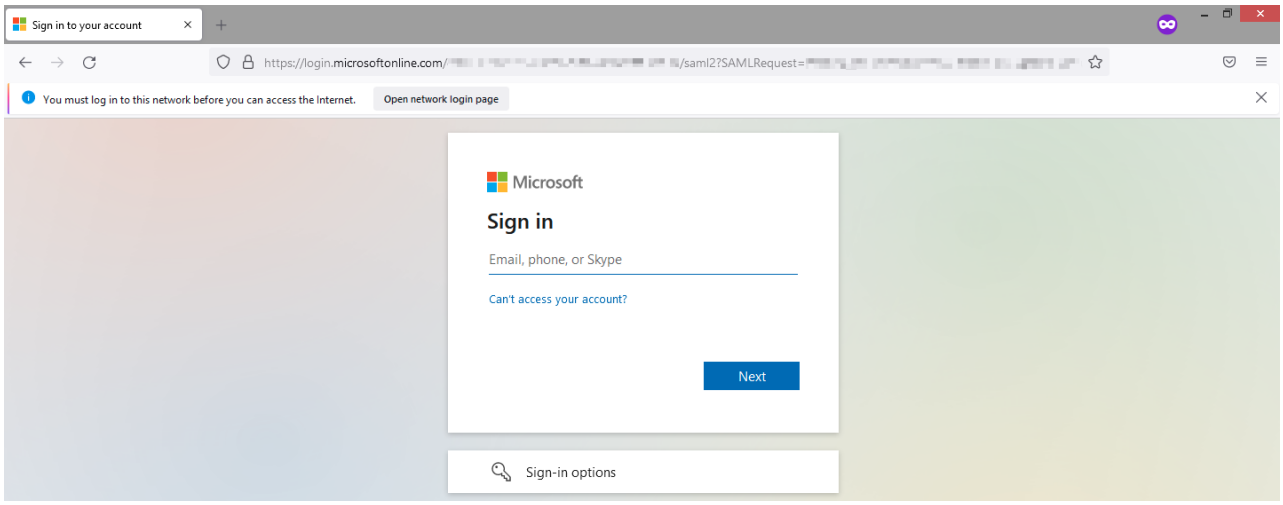
c. Click *OK*.

Connecting from the client

When the client connects to the internet from a browser, they will be redirected to the Microsoft log in page to authenticate against the Microsoft Entra ID. The FortiGate's authentication portal certificate should be installed on the client.

To connect from the client:

1. On the client, open a browser (such as Firefox) and go to a website. The user is redirected to the Microsoft log in page.
2. Enter the user credentials.



3. If the log in attempt is successful, the user is allowed to access the internet

Viewing logs and diagnostics

To verify user logins, go to the *Dashboard > Assets & Identities* and expand the *Firewall Users* widget, or enter the following in the CLI:

```
# diagnose firewall auth list
10.1.0.100, John Locus
        src_mac: 02:09:0f:00:03:03
        type: fw, id: 0, duration: 152, idled: 7
        expire: 292, allow-idle: 300
        server: Entra-ID-SAML
        packets: in 2097 out 932, bytes: in 2208241 out 143741
        group_id: 2
        group_name: Azure-FW-Auth
----- 1 listed, 0 filtered ------
```

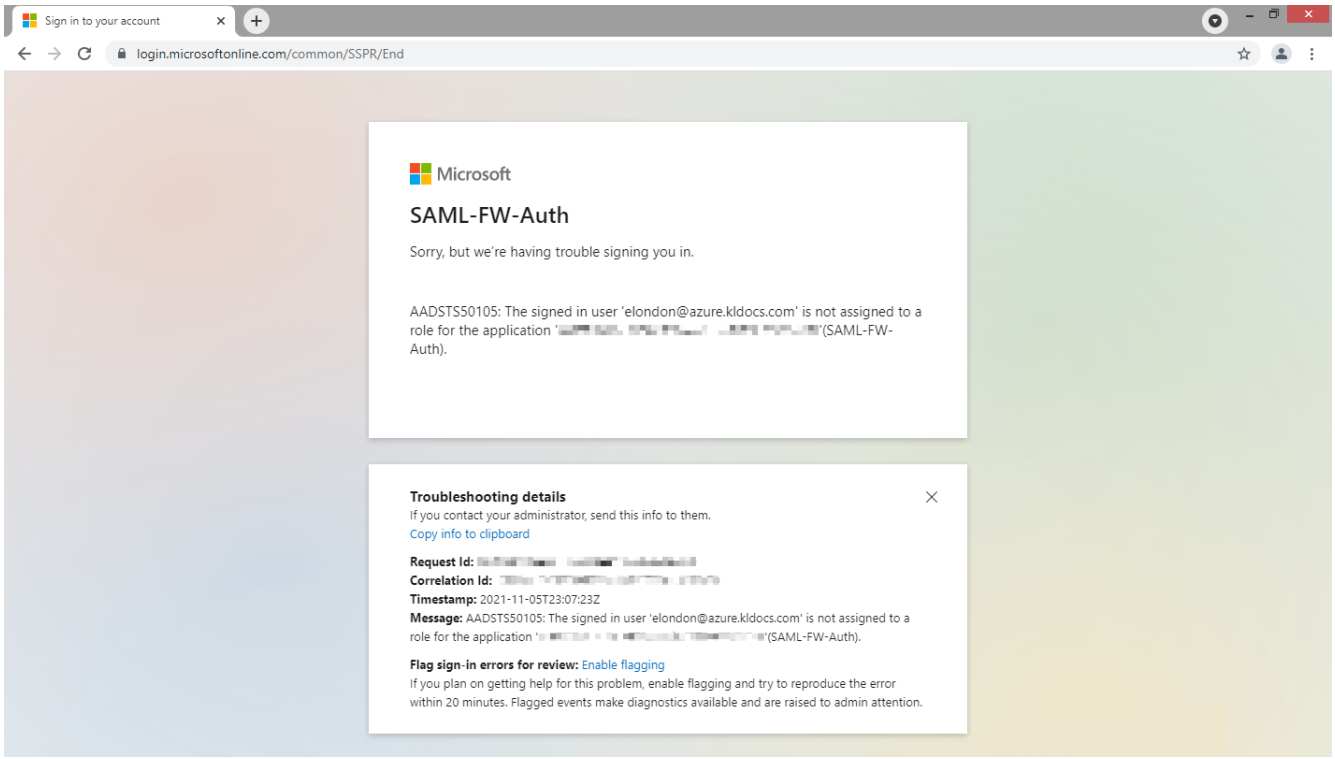
To verify user login logs, go to *Log & Report > System Events* and select the *User Events* card, or enter the following in the CLI:

```
# execute log filter category event
# execute log filter field subtype user
# execute log display
17 logs found.
10 logs returned.
7: date=2021-09-30 time=09:49:25 eventtime=1633020565577584390 tz="-0700" logid="0102043039"
type="event" subtype="user" level="notice" vd="root" logdesc="Authentication logon"
srcip=10.1.0.100 user="John Locus" authserver="Entra-ID-SAML" action="auth-logon" status="logon"
msg="User John Locus added to auth logon"

8: date=2021-09-30 time=09:49:25 eventtime=1633020565577075629 tz="-0700" logid="0102043008"
type="event" subtype="user" level="notice" vd="root" logdesc="Authentication success"
srcip=10.1.0.100 dstip=10.1.0.1 policyid=11 interface="port3" user="John Locus" group="Azure-FW-
Auth" authproto="HTTPS(10.1.0.100)" action="authentication" status="success" reason="N/A"
msg="User John Locus succeeded in authentication"
```

If user authentication is successful in Microsoft Entra ID, but their group does not match the one defined in the FortiGate user group, the user will receive a *Firewall Authentication Failed* message in the browser. A log is also recorded:

```
# execute log filter category event
# execute log filter field subtype user
# execute log display
1: date=2021-09-30 time=10:39:35 eventtime=1633023575381139214 tz="-0700" logid="0102043009"
type="event" subtype="user" level="notice" vd="root" logdesc="Authentication failed"
srcip=10.1.0.100 dstip=10.1.0.1 policyid=11 interface="port3" user="Adam Thompson" group="N/A"
authproto="HTTPS(10.1.0.100)" action="authentication" status="failure" reason="No matched SAML
user or group name in auth resp" msg="User Adam Thompson failed in authentication"
```

If a user receives the following error message, this means the user is not assigned to the enterprise application *SAML-FW-Auth* in Azure.



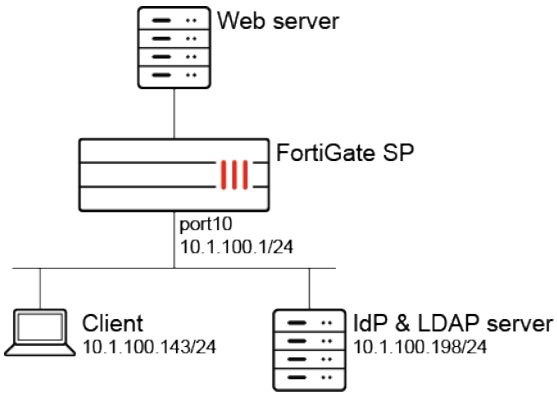
To troubleshoot SAML issues:

```
# diagnose debug application samld -1
# diagnose debug enable
```

SAML authentication in a proxy policy

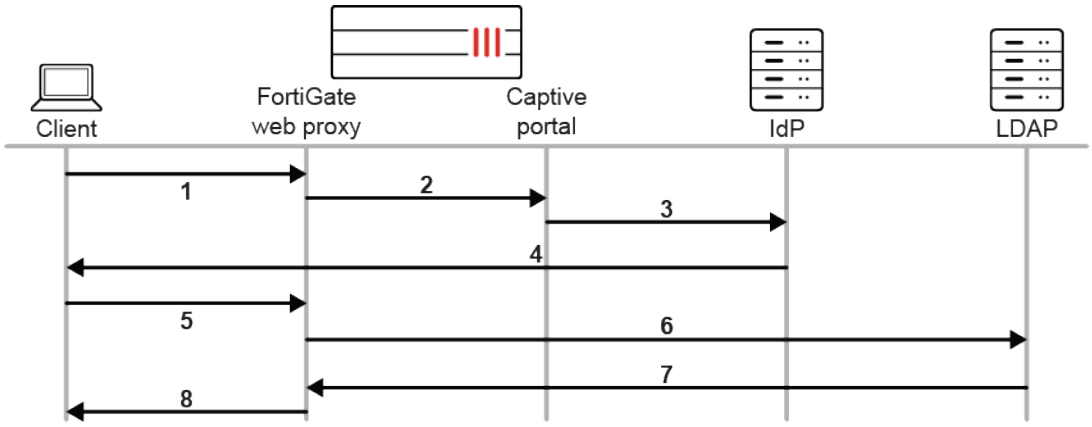
SAML user authentication can be used in explicit web proxies and transparent web proxies with the FortiGate acting as a SAML SP. SAML can be used as an authentication method for an authentication scheme that requires using a captive portal.

Topology



In this configuration, SAML authentication is used with an explicit web proxy. The IdP is a Windows 2016 server configured with ADFS. The LDAP and IdP servers are on the same server. The LDAP server is used as the user backend for the IdP to perform authentication; however, they are not required to be on the same server.

The authentication and authorization flow is as follows:



1. The client opens a browser and visits https://www.google.com.
2. The browser is redirected by the web proxy the captive portal.
3. The request is redirected to the IdP's sign-in page.
4. If the user signs in, the IdP authenticates the user and sends back a SAML assertion message to the user's browser with the user group information.
5. The browser forwards the SAML assertion response as a HTTP POST to the FortiGate SAML assertion consumer service URL (https://fgt9.myqalab.local:7831/XX/YY/ZZ/saml/login/).
6. If the FortiGate authentication scheme has a user database configured, the FortiGate will query the LDAP server for the user group information and ignore the user group information from the SAML message.
7. The user group information is returned. The FortiGate matches the user group information against the LDAP group in the proxy policy group settings. If there is a match, the request is authorized and the proxy policy is matched.
8. If all policy criteria match successfully, then the webpage is returned to the client.

To configure SAML authentication with an explicit web proxy:

1. Enable the web proxy:

```
config web-proxy explicit
    set status enable
    set http-incoming-port 8080
end
```

2. Enable the proxy captive portal:

```
config system interface
    edit "port10"
        set vdom "vdom1"
        set ip 10.1.100.1 255.255.255.0
        set allowaccess ping https ssh snmp http telnet
        set type physical
        set explicit-web-proxy enable
        set explicit-ftp-proxy enable
        set proxy-captive-portal enable
        set snmp-index 12
    next
end
```

3. Configure the LDAP server:

```
config user ldap
    edit "ldap-10.1.100.198"
        set server "10.1.100.198"
        set cnid "cn"
        set dn "dc=myqalab,dc=local"
        set type regular
        set username "cn=fosqa1,cn=users,dc=myqalab,dc=local"
        set password ************
        set group-search-base "dc=myqalab,dc=local"
    next
end
```

4. Configure the user group:

```
config user group
    edit "ldap-group-saml"
        set member "ldap-10.1.100.198"
    next
end
```

5. Configure SAML:

```
config user saml
    edit "saml_user"
        set cert "Fortinet_CA_SSL"
        set entity-id "https://fgt9.myqalab.local:7831/XX/YY/ZZ/saml/metadata/"
        set single-sign-on-url "https://fgt9.myqalab.local:7831/XX/YY/ZZ/saml/login/"
```

```
        set single-logout-url "https://fgt9.myqalab.local:7831/XX/YY/ZZ/saml/logout/"
        set idp-entity-id "http://MYQALAB.LOCAL/adfs/services/trust"
        set idp-single-sign-on-url "https://myqalab.local/adfs/ls"
        set idp-single-logout-url "https://myqalab.local/adfs/ls"
        set idp-cert "REMOTE_Cert_4"
        set digest-method sha256
        set adfs-claim enable
        set user-claim-type name
        set group-claim-type group
    next
end
```

6. Configure the authentication scheme, rule, and setting:

```
config authentication scheme
    edit "saml"
        set method saml
        set saml-server "saml_user"
        set user-database "ldap-10.1.100.198"
    next
end
```

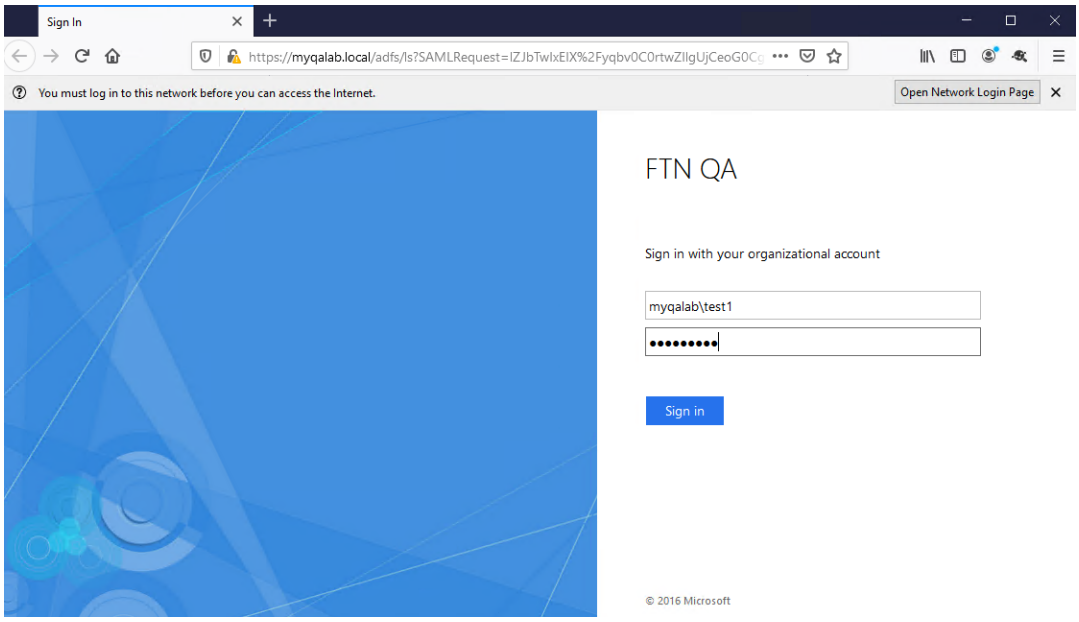
```
config authentication rule
    edit "saml"
        set srcaddr "all"
        set active-auth-method "saml"
    next
end
```

```
config authentication setting
    set captive-portal "fgt9.myqalab.local"
end
```

7. Configure the proxy policy:

```
config firewall proxy-policy
    edit 3
        set proxy explicit-web
        set dstintf "port9"
        set srcaddr "all"
        set dstaddr "all"
        set service "webproxy"
        set action accept
        set schedule "always"
        set logtraffic all
        set groups "ldap-group-saml"
        set utm-status enable
        set profile-protocol-options "protocol"
        set ssl-ssh-profile "deep-custom"
        set av-profile "av"
    next
end
```

When a user goes to www.google.com in a browser that is configured to use the FortiGate as a proxy, the IdP sign-in page appears.



Sample log

```
7: date=2021-03-16 time=21:11:19 eventtime=1615954279072391030 tz="-0700" logid="0000000010"
type="traffic" subtype="forward" level="notice" vd="vdom1" srcip=10.1.100.143 srcport=53544
srcintf="port10" srcintfrole="undefined" dstcountry="United States" srccountry="Reserved"
dstip=173.194.219.99 dstport=443 dstintf="port9" dstintfrole="undefined" sessionid=1751272387
service="HTTPS" wanoptapptype="web-proxy" proto=6 action="accept" policyid=3 policytype="proxy-
policy" poluuid="052ae158-7d40-51eb-c1d8-19235c4500c2" trandisp="snat" transip=172.16.200.1
transport=14844 duration=268 user="test1@MYQALAB.local" group="ldap-group-saml" authserver="ldap-
10.1.100.198" wanin=345633 rcvdbyte=345633 wanout=13013 lanin=5098 sentbyte=5098 lanout=340778
appcat="unscanned"
```

SAML authentication in a proxy policy using SCIM

The existing SCIM client support extends to authentication scheme using SAML, allowing `scim-client` to be used as `user-database`.

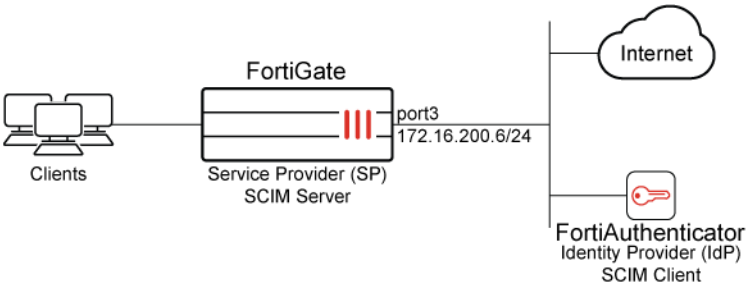
```
config authentication scheme
    edit <name>
        set method saml
        set saml-server <string>
        set user-database <Authentication server name>
    next
end
```

Example

In this example, FortiGate is configured as the SCIM server Service Provider (SP), and FortiAuthenticator is configured as the SCIM client Identity Provider (IdP). Two groups are configured on FortiAuthenticator: `scim_group` and `scim_group2`. The `scim_group` contains one user: `example@fortinet-us.com`.

Upon successful configuration, users and groups are provisioned on FortiGate. This setup can leverage SAML to provide access to authenticated users when using explicit web proxy policy.

Ensure the IdP root CA is imported into the FortiGate. Without this, the FortiGate will be unable to verify the IdP server certificate during the synchronization of SCIM client user information. In this example, we are using FortiAuthenticator as the IdP, and after importing, the FortiAuthenticator root certificate is referred to as `REMOTE_Cert_2` in the FortiGate.



To configure SAML authentication using SCIM with an explicit web proxy:

1. Configure FortiGate as an SCIM server:
   a. Allow SCIM access on an interface:

```
config system interface
    edit port3
        append allowaccess scim
    next
end
```

   b. Configure SCIM client entries:

```
config user scim
    edit "fac-scim"
        set status enable
        set base-url "https://172.16.200.6/fac-scim/scim/v2/"
        set auth-method base
        set secret **********
        set certificate REMOTE_Cert_2
        set client-identity-check disable
    next
end
```

2. Configure FortiAuthenticator as an SCIM client:
   a. On FortiAuthenticator, use your admin account to log in.
   b. Click *Authentication > SCIM > Service Provider*.

c. Click *Create New*.
d. Configure the following settings:

Field	Value
Name	Test-SCIM
SCIM endpoint	https://172.16.200.6:44559/fac-scim/scim/v2
Access token	********

e. Configure the remaining settings as required.
f. Click *Save*.
g. Once the setting is saved, double-click the newly created entry to open the settings pane, and click *Sync* to automatically add existing users to the SCIM server.

See Service providers for more information.

The SCIM endpoint and access token must match the `base-url` and `secret` respectively, as configured on the FortiGate.

Furthermore, an initial synchronization is necessary to commence provisioning for the first time. However, when alterations to identities occur in the IdP, including creation, updating, and deletion, these changes are automatically synchronized with SP in accordance with the SCIM protocol.

3. Verify whether users and groups are provisioned on the FortiGate:

```
# diagnose test scim list-users fac-scim
example@fortinet-us.com
total:1
# diagnose test scim list-groups fac-scim
displayName     externalId
scim_group      1
scim_group2     2
total:2

# diagnose test scim get-user fac-scim example@fortinet-us.com
schemas num: 1
        [0]:schema: 1
id: user_2
userName: example@fortinet-us.com
externalId: local3
name:
meta: resourceType:User, created:2025-05-27T22:54:15Z, lastModified:2025-05-29T18:11:20Z,
location:https://172.16.200.6/fac-scim/scim/v2//Users/user_2,version: 1.0
emails num: 0
groups num:1
        groups[0]: value:group_3, $ref:https://172.16.200.6/fac-scim/scim/v2//Groups/group_3,
display:scim_group
Active: 1
index: 0
```

For brevity, only the commands relevant to this enhancement are included. See SAML on page 3573 for more information about configuring SP and IdP.

4. Configure an authentication scheme and authentication rule to use `scim-client` as a `user-database`:

```
config authentication scheme
    edit "saml"
        set method saml
        set saml-server "test_example_saml"
        set user-database "fac-scim"
    next
end
config authentication rule
    edit "saml"
        set srcaddr "all"
        set active-auth-method "saml"
    next
end
```

5. Configure the user group:

```
config user group
    edit "ldap-group-saml"
        set member "ldap-10.1.100.198"
    next
end
```

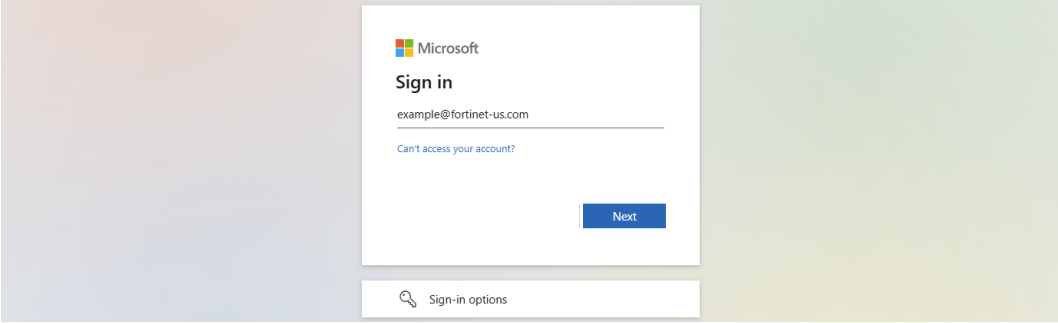
6. Enable the web proxy:

```
config web-proxy explicit
    set status enable
    set http-incoming-port 8080
end
```

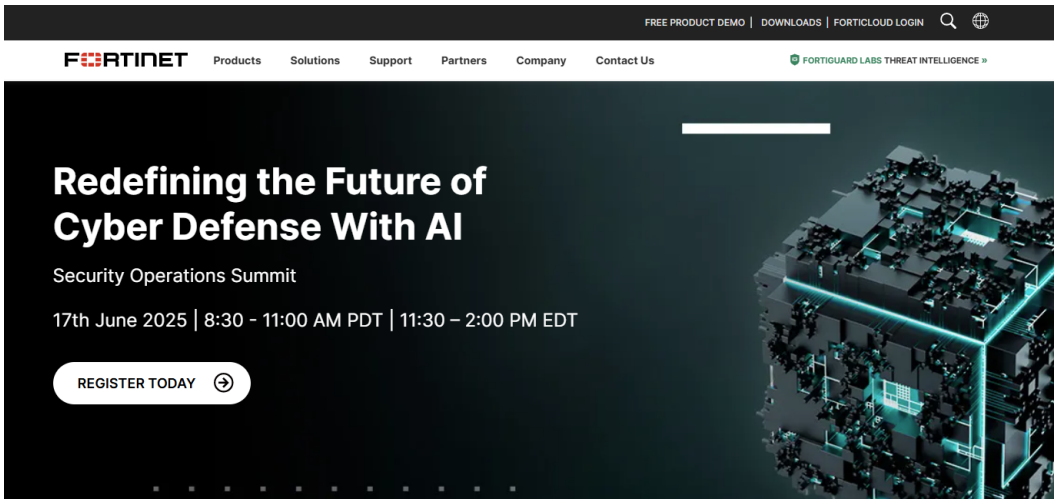
7. Apply SAML authentication using SCIM with an explicit web proxy:

```
config firewall proxy-policy
   edit 1
        set name "saml"
        set proxy explicit-web
        set dstintf "port3"
        set srcaddr "all"
        set dstaddr "all"
        set service "webproxy"
        set action accept
        set schedule "always"
        set logtraffic all
        set groups "saml"
    next
end
```

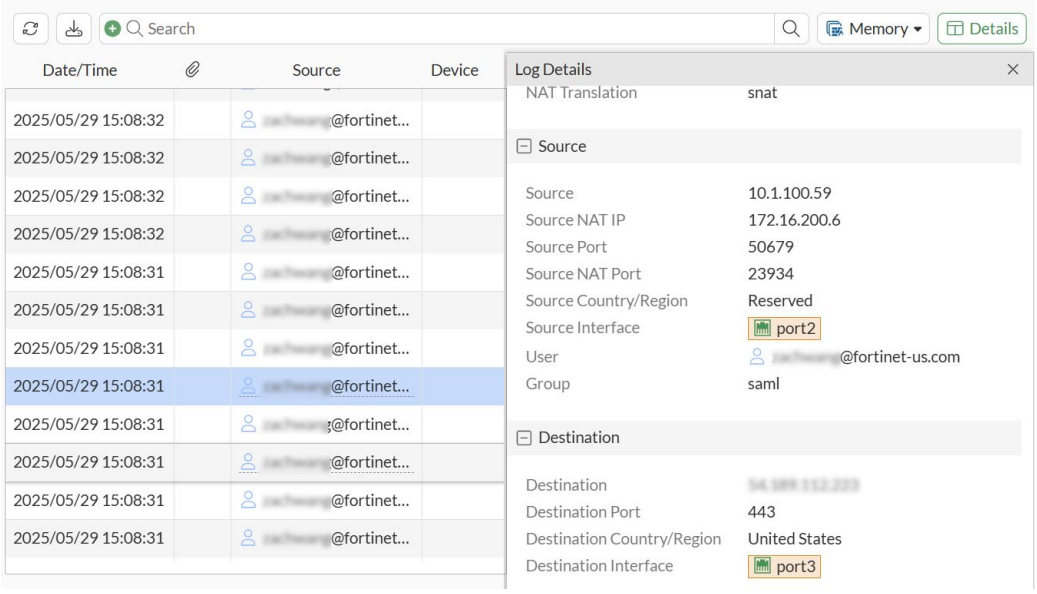
When a user goes to www.fortinet.com in a browser that is configured to use the FortiGate as a proxy, the authentication window will pop-up in the browser.



After entering user credentials and successfully passing authentication, the Fortinet page opens successfully.



Log are displayed in *Log & Report*.



Local entries for SAML users

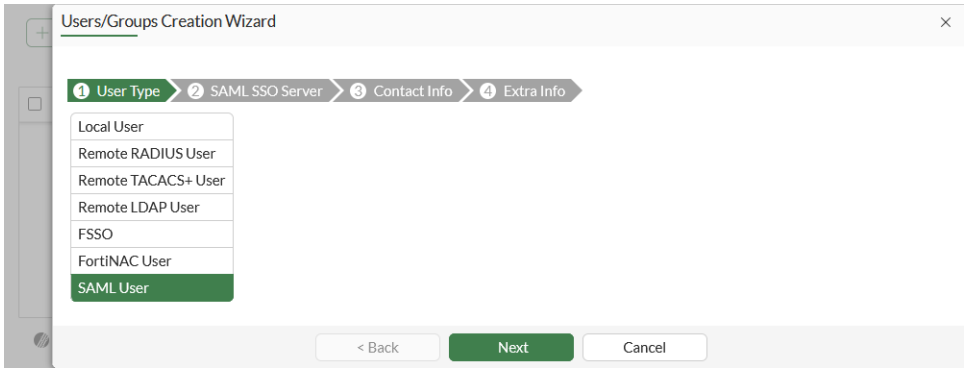
FortiOS supports SAML users when configuring local users, expanding its ability to define individual remote users for policy enforcement.

Administrators can create local entries for SAML users, enabling precise, user-specific control in both firewall policies and agentless VPNs. This streamlines policy management and enhances flexibility for environments using SAML authentication.

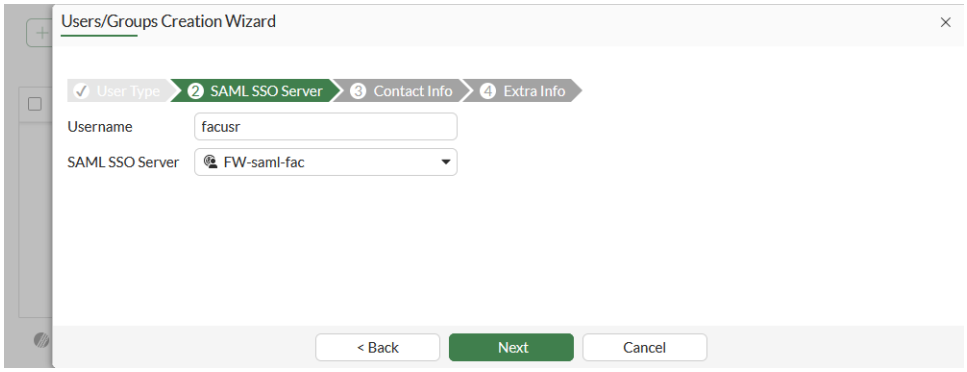
A working SAML IdP connection must already be configured.

To configure a local SAML user in the GUI:

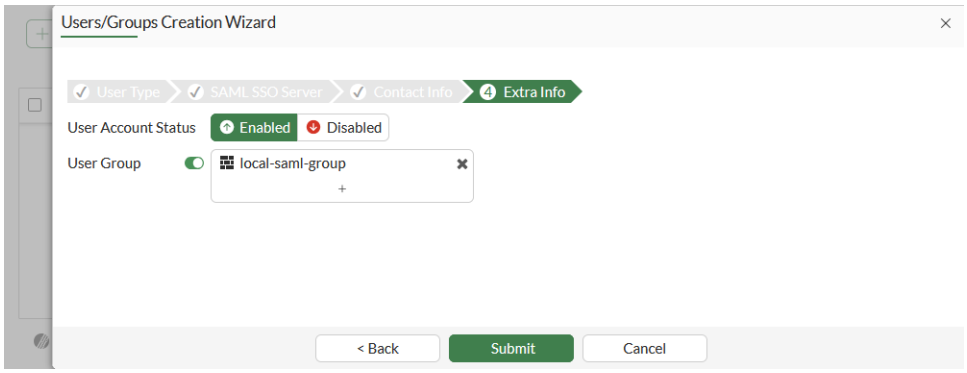
1. Go to *User & Authentication* > *Single Sign-On* and confirm that a SAML IdP connection is configured. See SAML for more information.
2. Go to *User & Authentication* > *User Definition* and click *Create New*.



3. Select *SAML User*, then click *Next*.
4. Enter a *Username* and select a *SAML SSO Server*, then click *Next*.



5. Optionally, enable and configure *Two-factor Authentication*, then click *Next*.
6. Add the SAML user to a local user group, then click *Submit*.



To configure a local SAML user in the CLI:

1. Make sure that a SAML server is configure:

```
show user saml
   edit "FW-saml-fac"
        set entity-id "http://10.1.100.7:1000/metadata/"
        set single-sign-on-url "https://10.1.100.7:1003/saml/acs"
        set single-logout-url "https://10.1.100.7:1003/saml/sls"
        set idp-entity-id "http://10.1.100.195/saml-idp/77778888/metadata/"
        set idp-single-sign-on-url "https://10.1.100.195/saml-idp/77778888/login/"
        set idp-single-logout-url "https://10.1.100.195/saml-idp/77778888/logout/"
        set idp-cert "REMOTE_Cert_2"
        set user-name "username"
        set group-name "group"
        set digest-method sha1
    next
end
```

2. Configure the local user as a SAML user, and map them to the SAML user:

```
config user local
    edit "facusr"
        set type saml
        set saml-server "FW-saml-fac"
    next
end
```

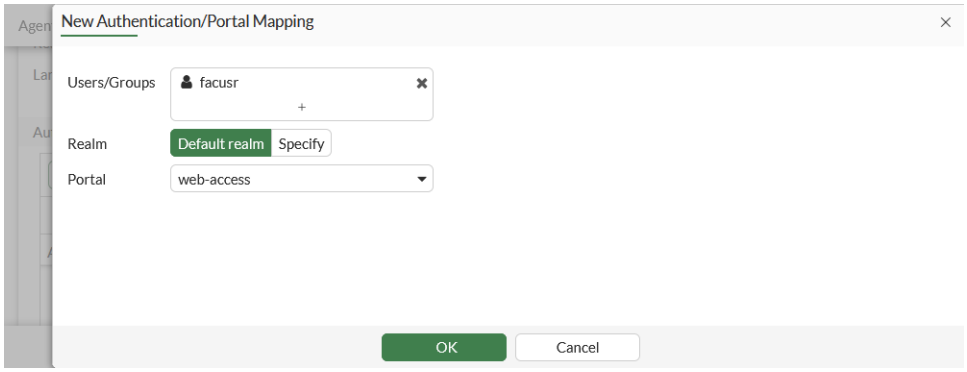
3. Optionally, add the user to a group:

```
config user group
    edit "local-saml-group"
        set member "facusr"
    next
end
```

Verification

To test the local SAML user on an agentless VPN portal in the GUI:

1. Add the user to an agentless VPN portal mapping:
   a. Go to *VPN > Agentless VPN Settings*.
   b. In the *Authentication/Portal Mapping* table click *Create New*.
   c. Set *Users/Groups* to the SAML user. The user group that the SAML user was added to could also be used.
   d. Set *Portal* to *web-access*.



   e. Click *OK*.
   f. Click *Apply*.
2. On a client PC, connect to the agentless VPN portal and enter valid SAML user credentials for authentication to access the portal page.
3. On the FortiGate, check the connection in the FortiView VPN widget.

To test the local SAML user on an agentless VPN portal in the CLI:

1. Add the user to an agentless VPN portal mapping:

```
config vpn ssl settings
    config authentication-rule
        edit 1
            set users "facusr"
            set portal "web-access"
        next
    end
end
```

The user group that the SAML user was added to could also be used:

```
config vpn ssl settings
    config authentication-rule
        edit 1
            set groups "local-saml-group"
        next
    end
end
```

2. On a client PC, connect to the agentless VPN portal and enter valid SAML user credentials for authentication to access the portal page.
3. On the FortiGate, check the connection:

```
# get vpn ssl monitor
Agentless VPN Login Users:
|Index|User  |Group|Auth Type|Idle-Timeout|Auth-Timeout|From        |HTTP in/out|HTTPS
in/out|Two-factor Auth|
|0|   facuser|N/A  |256(1)|273|28772                    |10.1.100.188|0/0|0/0|0|
```

To test the local SAML user used in a policy in the CLI:

1. Add the user to a firewall policy:

```
config firewall policy
    edit 2
        set users "facusr"
    next
end
```

   The user group that the SAML user was added to could also be used:

```
config firewall policy
    edit 2
        set groups "local-saml-group"
    next
end
```

2. On a client PC, use a browser to connect to an external website. The browser will redirect to the IdP login screen for authentication.
3. Enter valid SAML user credentials for authentication. After successful authentication, the browser reroutes to the external website.
4. On the FortiGate, check the list of authenticated users:

```
# diagnose firewall auth list

10.1.100.188, facuser
        src_mac: 00:0c:29:44:be:b9
        type: fw, id: 0, duration: 63, idled: 12
        expire: 237
        flag(804): hard no_idle
        server: FW-saml-fac
        packets: in 29 out 39, bytes: in 12594 out 4409
        user_id: 16777229
        group_id:
        group_name:

----- 1 listed, 0 filtered ------
```

Per-Session SAML authentication logging and logout support for ZTNA and explicit proxy users

FortiGate supports per-session SAML authentication logging for identical usernames from the same IP address for proxy policies, which are used with ZTNA and explicit proxy configurations. When session-based authentication with cookies is enabled and SAML is configured, with the added session logout option enabled, ZTNA and explicit proxy users can log out individually from each browser session.

The FortiGate configured as SAML Service Provider (SP) generates a unique logout URL for the individual user or session that can be used to sign the user out from both the SP and the SAML Identity Provider (IdP). The SAML single signout flow between client, SP (FortiGate), and IdP is as follows:

1. The client uses the SP single-logout-url configured on the FortiGate to send a logout request to the FortiGate as SP.
2. The FortiGate logs out of the local user corresponding to the SAML assertion and returns a unique IdP logout URL to the client.
3. The client uses this unique IdP logout URL to send a logout request to the IdP.
4. When the session has been logged out on the IdP, the FortiGate receives a SAML logout response.

The `session-logout` option in the proxy policy authentication rule can be enabled to turn on additional session tracking to support SAML single sign out:

```
config authentication rule
    edit < name >
        …
        set ip-based disable
        set active-auth-method < auth_method >
        set web-auth-cookie enable
        …
        set session-logout enable
    next
end
```

By default, `session-logout` is disabled. The option is only available when the following conditions are satisfied:

1. Session-based authentication is enabled (`set ip-based disable`).
2. Device certificate as authentication cookie is enabled (`set web-auth-cookie enable`).
3. SAML authentication is used:
   a. `set active-auth-method` refers to an authentication scheme using SAML.
   b. SAML configuration for SP and IdP are validly configured on the FortiGate. See SAML on page 3573.

Example

This example demonstrates an explicit proxy configuration using SAML SSO authentication with `session-logout` enabled to enable session tracking, required for SAML single sign out.

To configure a SAML server and SAML user group:

```
config user saml
    edit "saml"
        set cert "Fortinet_CA_SSL"
        set entity-id "http://192.0.2.1/remote/saml/metadata/"
        set single-sign-on-url "https://192.0.2.1:7831/remote/saml/login"
        set single-logout-url "https://192.0.2.1:7831/remote/saml/logout"
        set idp-entity-id "http://198.51.100.1/saml-idp/saml/metadata/"
        set idp-single-sign-on-url "https://198.51.100.1/saml-idp/saml/login/"
        set idp-single-logout-url "https://198.51.100.1/saml-idp/saml/logout/"
        set idp-cert "SAML"
        set user-name "Username"
        set group-name "Group"
        set digest-method sha1
    next
end
```

```
config user group
    edit "saml_group"
        set member "saml"
    next
end
```

The `single-logout-URL` of `https://192.0.2.1:7831/remote/saml/logout` will be used by the client to initiate SAML single sign out.

To configure the authentication rule and scheme to match the new SAML server:

```
config authentication rule
    edit "saml_rule"
        set srcaddr "all"
        set ip-based disable
        set active-auth-method "saml_scheme"
        set web-auth-cookie enable
        set cors-stateful enable
        set cors-depth 8
        set session-logout enable
    next
end
```

```
config authentication scheme
    edit "saml_scheme"
        set method saml
        set saml-server "saml"
    next
end
```

To enable user group authentication in a firewall proxy-policy:

```
config firewall proxy-policy
    edit 1
        set proxy explicit-web
        set dstintf "port3"
        set srcaddr "all"
        set dstaddr "all"
        set service "webproxy"
        set action accept
        set schedule "always"
        set logtraffic all
        set groups "saml_group"
        set ssl-ssh-profile "deep-custom"
    next
end
```

To test SAML session tracking and single sign out :

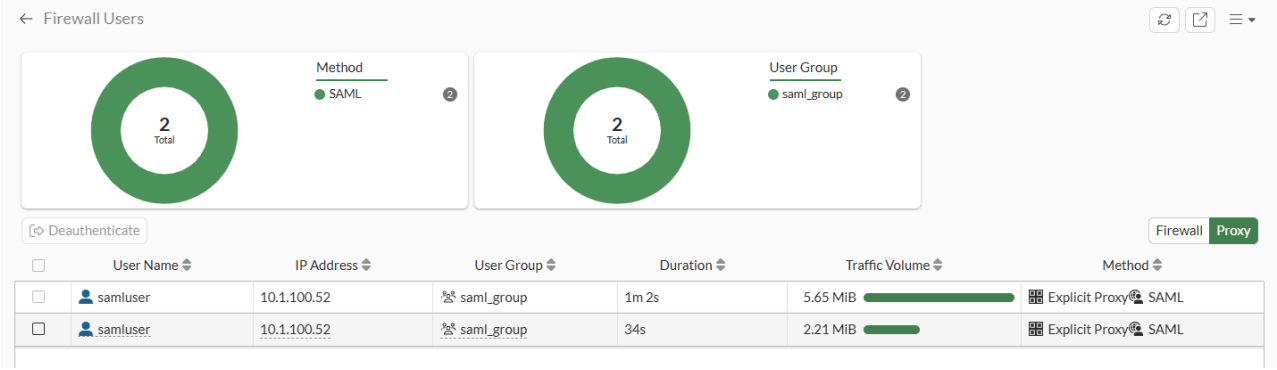
1. On a client PC, access one website (website 1) from one web browser (browser 1) and access another web site (website 2) from another web browser (browser 2). The FortiGate will record user information from different browsers with the same IP address.
2. The SAML IdP will present an authentication page when accessing website 1 from browser 1. Enter your username and password in the authentication page. Observe website 1 load as expected.
3. The SAML IdP will present an authentication page when accessing website 2 from browser 2. Enter your username and password in the authentication page. Observe website 2 load as expected.
4. In the CLI, check the WAD user list

```
# diagnose wad user list

ID: 2, VDOM: vdom1, IPv4: 10.1.100.52
  user name   : samluser
  worker      : 22
  duration    : 40
  auth_type   : Session
  auth_method : SAML
  pol_id      : 1
  g_id        : 10
  user_based  : 0
  expire      : no
  LAN:
    bytes_in=461676 bytes_out=5421390
  WAN:
    bytes_in=5331093 bytes_out=597696

ID: 3, VDOM: vdom1, IPv4: 10.1.100.52
  user name   : samluser
  worker      : 22
  duration    : 12
  auth_type   : Session
  auth_method : SAML
```

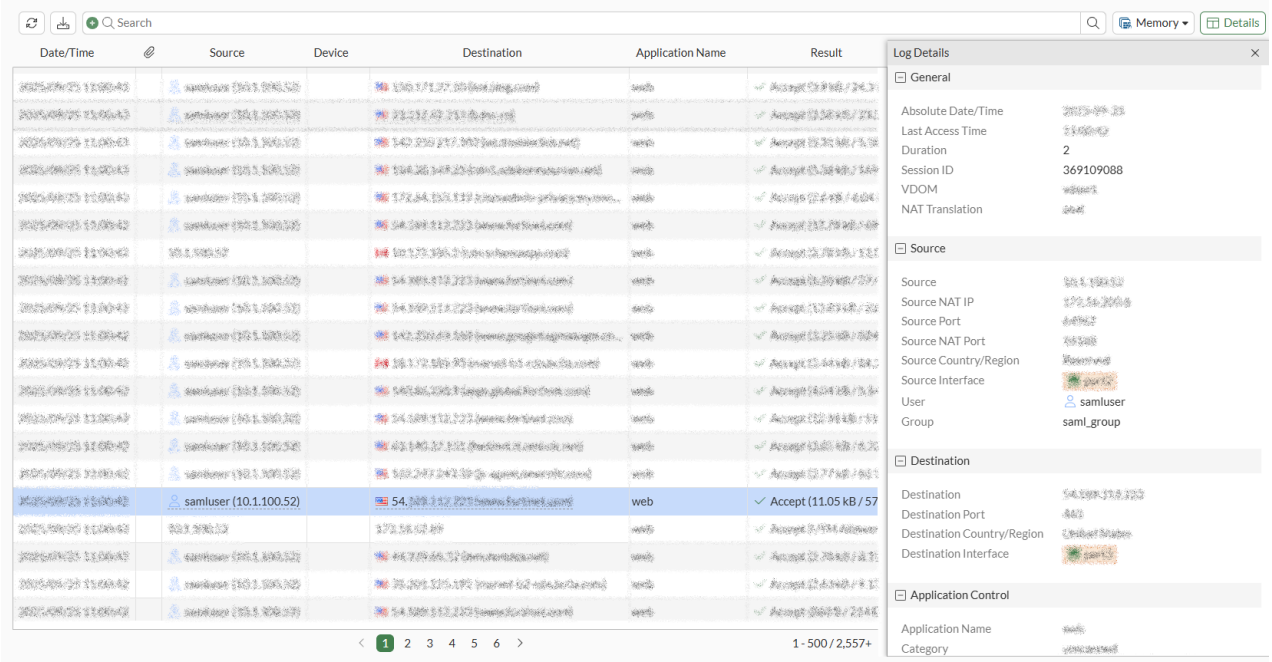
```
pol_id      : 1
g_id        : 10
user_based  : 0
expire      : no
LAN:
  bytes_in=33402 bytes_out=2253163
WAN:
  bytes_in=2238161 bytes_out=81347
```

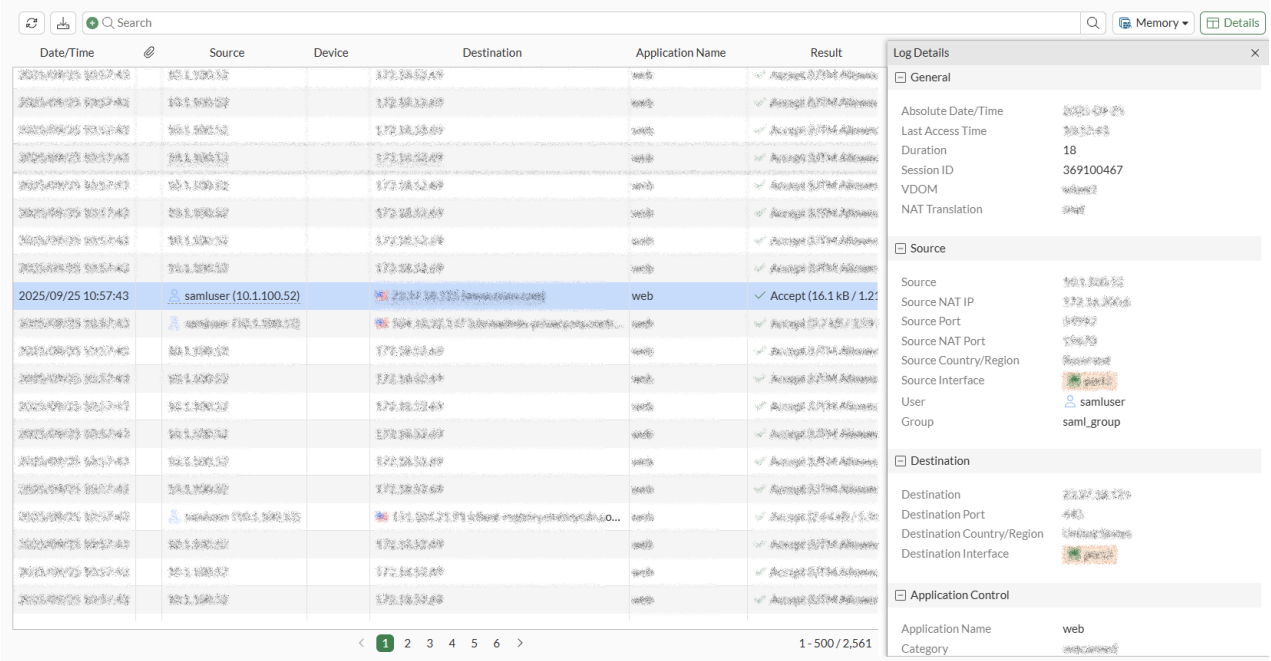
Observe two sets of user information entries corresponding to the two SAML authenticated sessions.

5. Go to *Dashboard* > *Status* and expand the *Firewall Users* widget. Observe the two user sessions:

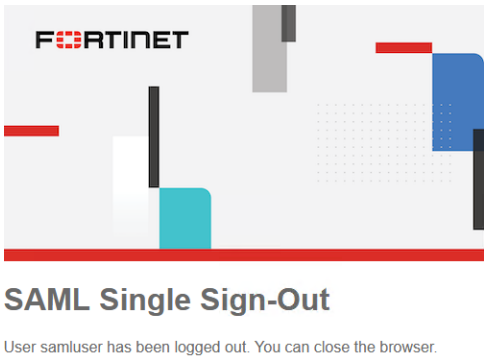


6. View the traffic logs and note the unique session IDs for the websites visited:





7. Recall from the SAML server configuration that the `single-logout-URL` of `https://192.0.2.1:7831/remote/saml/logout` was configured. On the client in web browser 2, initiate a SAML single sign out by accessing `https://192.0.2.1:7831/remote/saml/logout`.

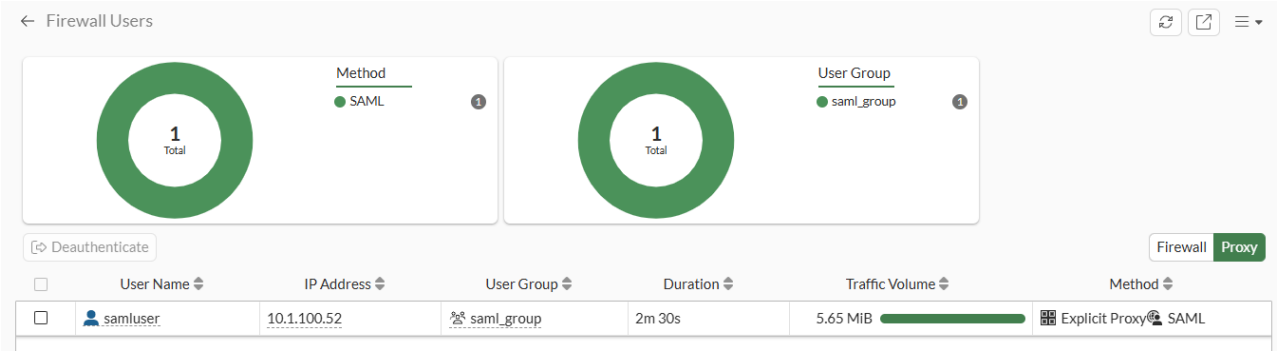


8. Check that the user information for the first SAML authenticated session for web browser 1 is still present:

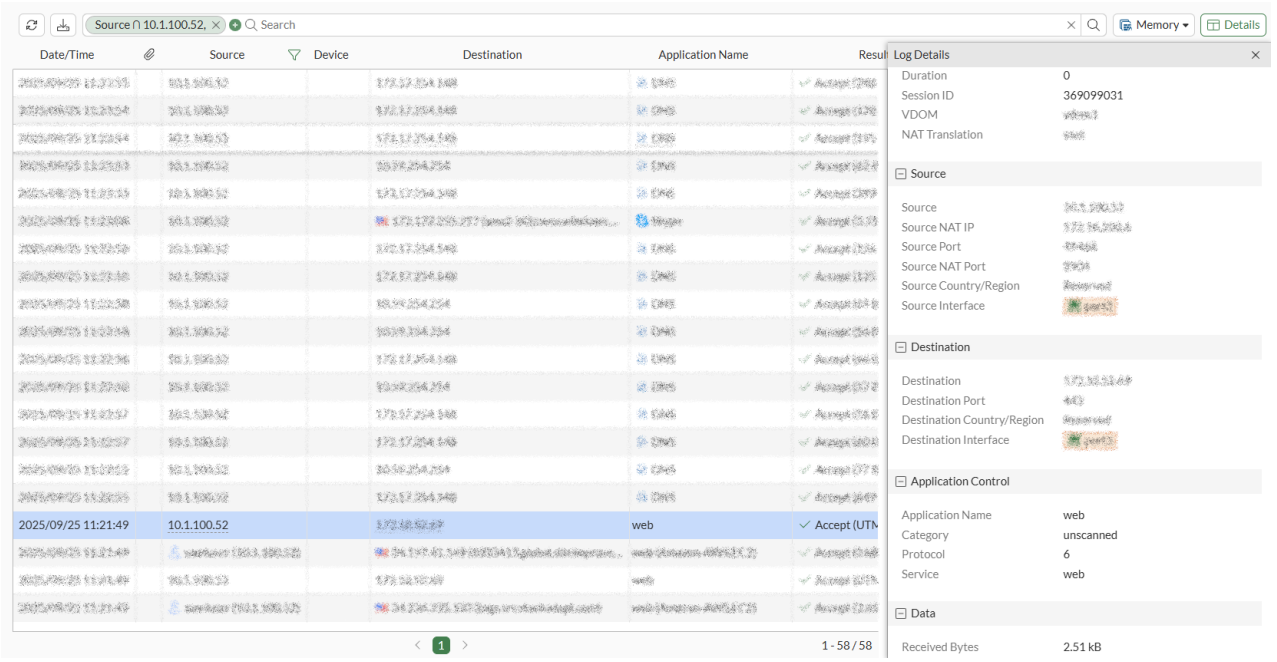
```
# diagnose wad user list

ID: 2, VDOM: vdom1, IPv4: 10.1.100.52
  user name    : samluser
  worker       : 22
  duration     : 133
  auth_type    : Session
  auth_method  : SAML
  pol_id       : 1
  g_id         : 10
  user_based   : 0
  expire       : no
  LAN:
    bytes_in=461676 bytes_out=5421755
```

```
WAN:
  bytes_in=5331434 bytes_out=597899
```

9. Go to *Dashboard* > *Status*, expand the *Firewall Users* widget, and see the first user session corresponding to web browser 1:



10. Check the traffic logs and note the unique session ID for the SAML log out request:



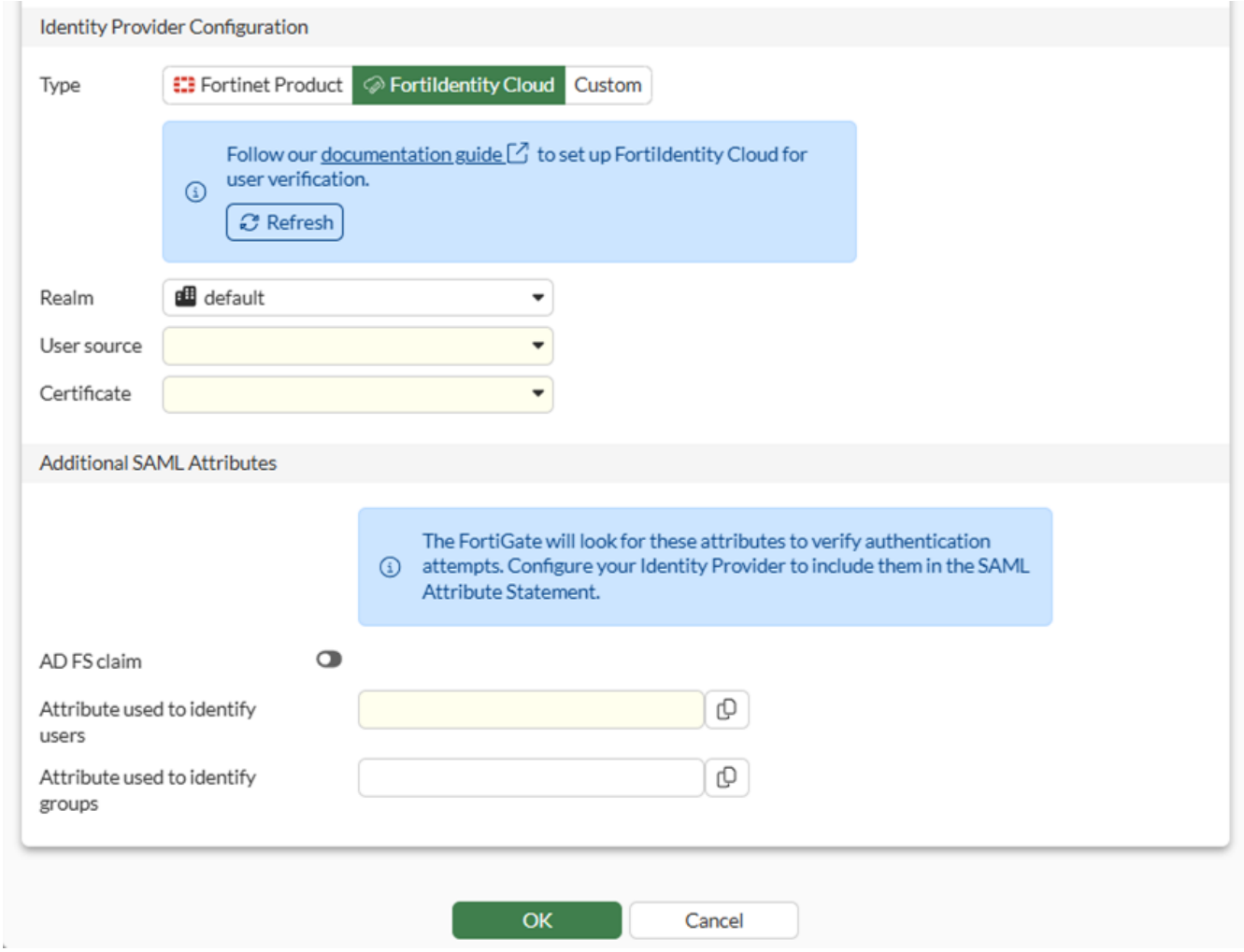
SSO integration with FortiIdentity Cloud

FortiIdentity Cloud can act as the local Identity Provider (IdP) for SAML and OIDC applications, or it can operate as an IdP proxy to broker authentication requests to other upstream IdPs. In this architecture, the FortiGate acts as the Service Provider (SP) for services such as VPN or ZTNA access and completes authentication via SAML against FortiIdentity Cloud.

In this enhancement, SSO is more tightly integrated between FortiGate and FortiIdentity Cloud. In the FortiGate's Single Sign-On configuration page, a new type, FortiIdentity Cloud, is added. When selected, the FortiGate can retrieve existing realms, user sources and certificates configured on FortiIdentity Cloud, as well as

retrieve the IdP URLs automatically. Once the fields are configured and saved, FortiIdentity Cloud automatically configures an SSO application corresponding to the FortiGate configuration and synchronizes the SP URLs.

In FortiOS, go to *User & Authentication* > *Single Sign-On* to configure an SSO server with type FortiIdentity Cloud:



When FortiIdentity Cloud is selected, the following fields are pre-populated based on your FortiIdentity Cloud configuration:

Field	Description
Realm	A realm is a container that has a set of users that can be controlled by the same realm settings, including MFA method and adaptive auth profile.
User source	The source of user identities. This could be provided by upstream Identity providers, active directory or sometimes the identity store can be local users.
Certificate	Certificate to identify the IdP.

FortiGates with FortiCare Premium or FortiCare Elite have 3 free seats to FortiIdentity Cloud. See FortiCloud Subscriptions for details.

To configure FortiIdentity Cloud, see Getting Started – FGT-FIC users.

Example

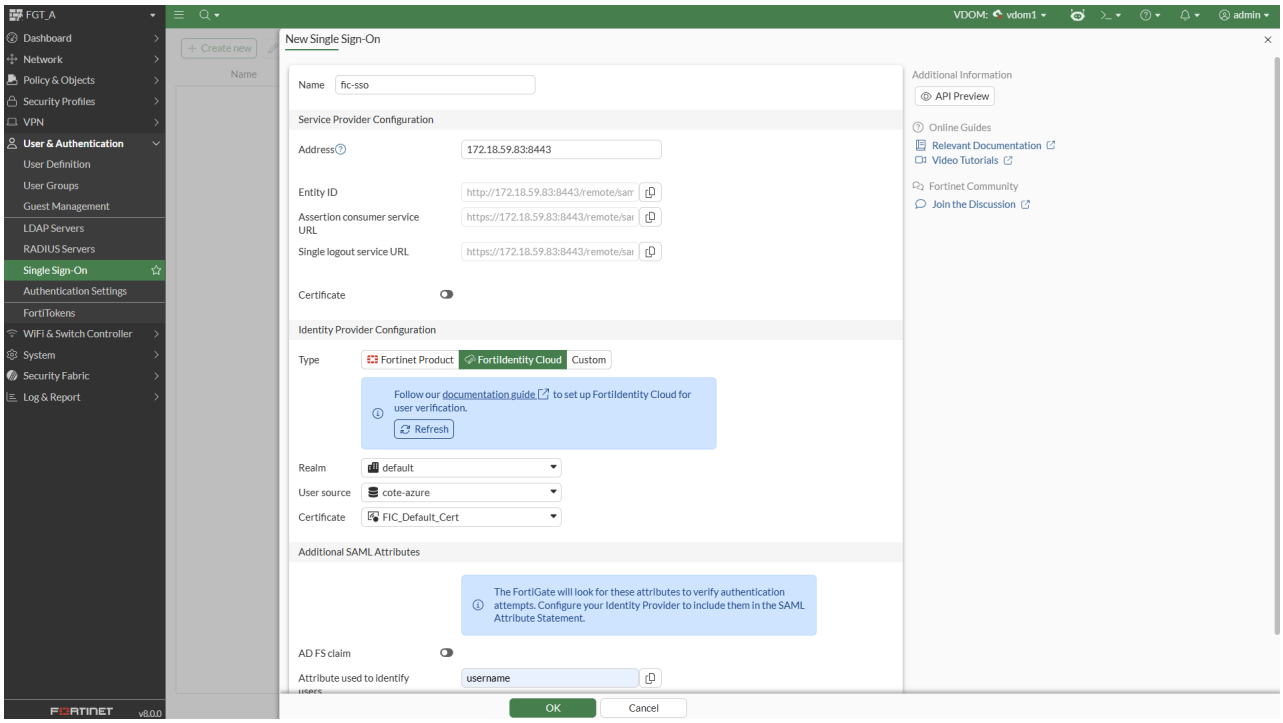
This example demonstrates the simple SSO configuration on the FortiGate and the subsequent synchronization of configurations between the FortiGate and the FortiIdentity Cloud. Once configured, the SSO server can be used for remote access VPNs, ZTNA, and other services that require authentication.

To configure and check the SSO configuration in the GUI:

1. On the FortiGate, go to *User & Authentication* > *Single Sign-On* and click *Create New*.
2. Set the *Name* of the SSO, *fic-sso* in this example.
3. Set *Address* to the address that the IdP will send SAML authentication requests to.
4. Under *Identity Provider Configuration*, set *Type* to *FortiIdentity Cloud* and configure the following:

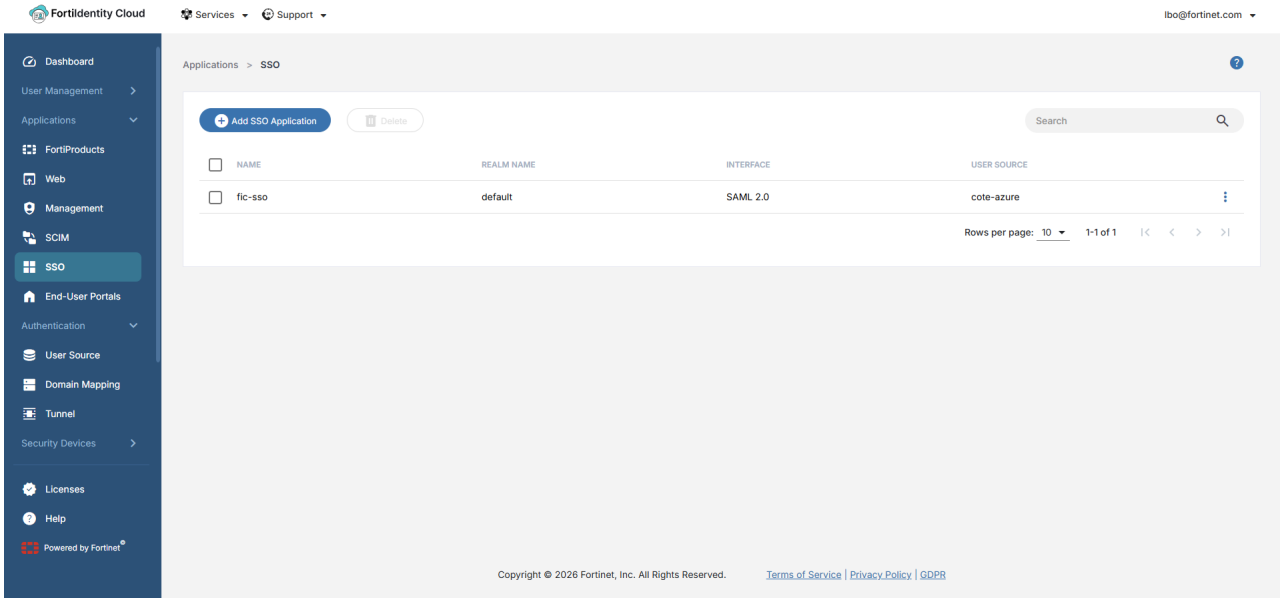
Field	Value
Realm	default
User source	cote-azure
Certificate	FIC_Default_Cert

5. Set *Attribute used to identify users* to *username*.
6. Set *Attribute used to identify groups* to *group*.

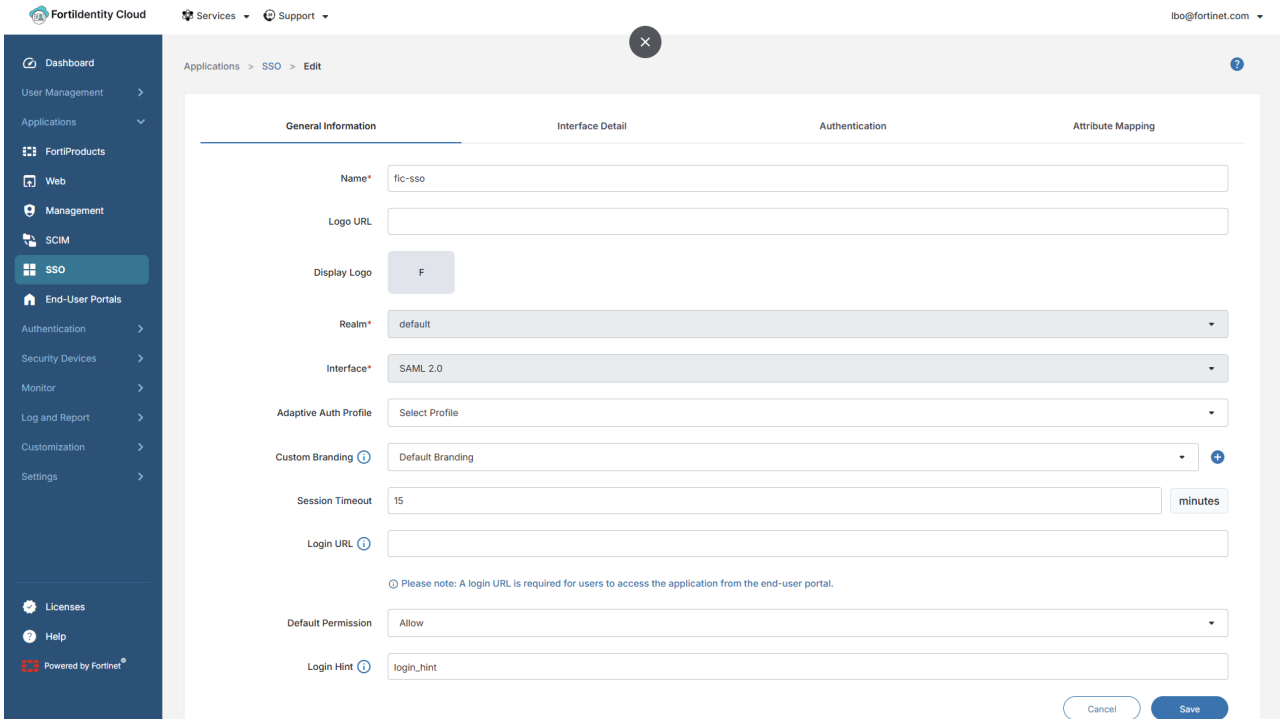


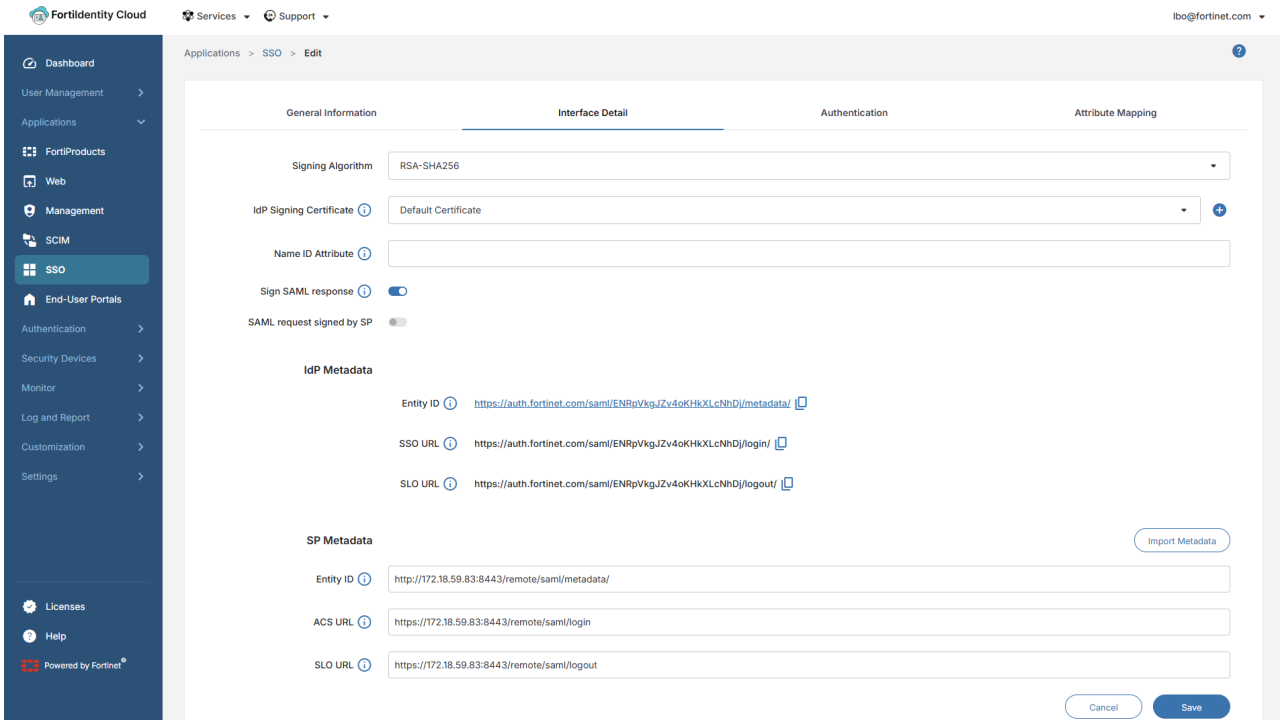
7. Click *OK*.

   A new SSO application with the same name is automatically created in FortiIdentity Cloud:

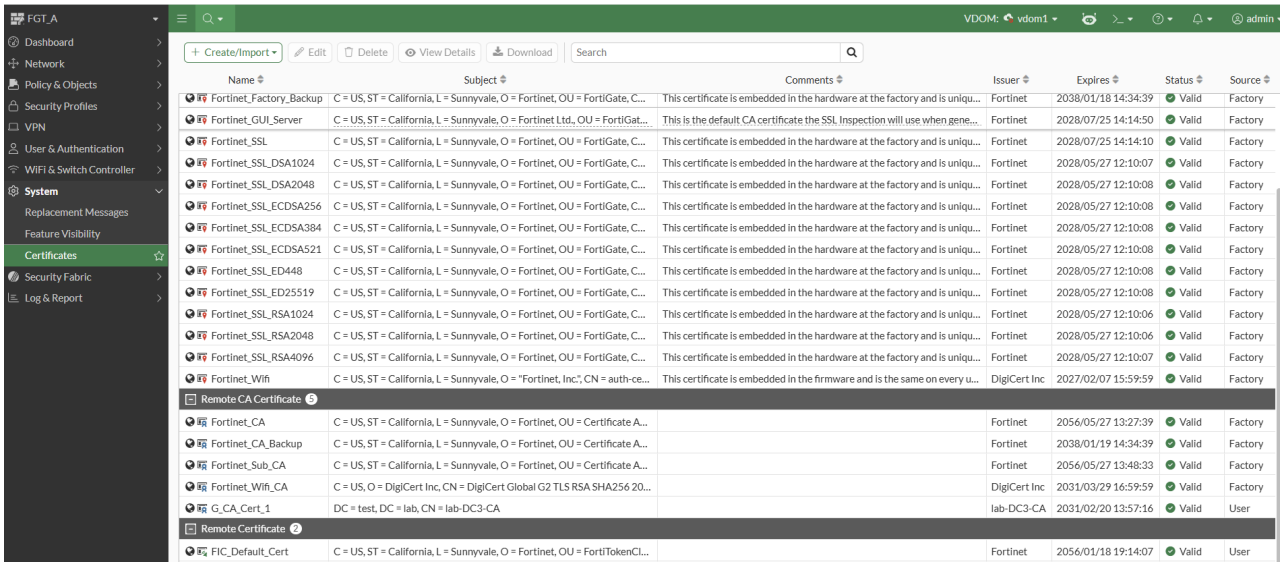


The metadata between FortiGate (SP) and FortiIdentity Cloud (IdP) is exchanged in the background:





The certificate from FortiIdentity Cloud is automatically imported into FortiGate as a remote certificate, completing the SSO configuration:



To configure the SSO in the CLI:

1. Configure an SSO with type FortiIdentity Cloud:

```
config user saml
    edit "fic-sso"
        set type fortiidentity-cloud
        set idp-cert "FIC_Default_Cert"
```

```
        set user-name "username"
        set group-name "group"
        set service-provider-address "172.18.59.83:8443"
        set user-source "cote-azure"
     next
end
```

2. Check that the SSO configuration is done:

```
# show user saml
config user saml
    edit "fic-sso"
        set uuid 52c8b5ee-3f68-51f1-8b53-b3a4fb3493d0
        set type fortiidentity-cloud
        set idp-cert "FIC_Default_Cert"
        set user-name "username"
        set group-name "group"
        set digest-method sha256
        set service-provider-address "172.18.59.83:8443"
        set entity-id "http://172.18.59.83:8443/remote/saml/metadata/"
        set single-sign-on-url "https://172.18.59.83:8443/remote/saml/login"
        set single-logout-url "https://172.18.59.83:8443/remote/saml/logout"
        set idp-entity-id "https://auth.fortinet.com/saml/ENRpVkgJZv4oKHkXLcNhDj/metadata/"
        set idp-single-sign-on-url
"https://auth.fortinet.com/saml/ENRpVkgJZv4oKHkXLcNhDj/login/"
        set idp-single-logout-url
"https://auth.fortinet.com/saml/ENRpVkgJZv4oKHkXLcNhDj/logout/"
        set user-source "cote-azure"
    next
end
```

TACACS+ servers

TACACS+ is a remote authentication protocol that provides access control for routers, network access servers, and other network devices through one or more centralized servers.

FortiOS sends the following proprietary TACACS+ attributes to the TACACS+ server during authorization requests:

Attribute	Description
`service=<name>`	User must be authorized to access the specified service.
`memberof`	Group that the user belongs to.
`admin_prof`	Administrator profile (admin access only).

Only `memberof` and `admin_prof` attributes are parsed in authentication replies.

You can configure up to ten remote TACACS+ servers in FortiOS. You must configure at least one server before you can configure remote users.

A TACACS+ server must first be added in the CLI to make the option visible in the GUI.

To configure TACACS+ authentication in the CLI:

1. Configure the TACACS+ server entry:

```
config user tacacs+
    edit "TACACS-SERVER"
        set server <IP address>
        set key <string>
        set authen-type ascii
        set source-ip <IP address>
    next
end
```

2. Configure the remote user group:

```
config user group
    edit "TACACS-GROUP"
        set group-type firewall
        set member "TACACS-SERVER"
    next
end
```

3. Configure the remote user:

```
config system admin
    edit TACACS-USER
        set remote-auth enable
        set accprofile "super_admin"
        set vdom "root"
        set wildcard enable
        set remote-group "TACACS-GROUP"
    next
end
```

To configure a TACACS+ server in the GUI:

1. Go to *User & Authentication* > *TACACS+ Servers*.
2. Click *Create New*.
3. Configure the following settings:

Name	Enter the TACACS+ server name.
Authentication Type	Select the authentication type used for the TACACS+ server. Selecting *Auto* tries PAP, MSCHAP, and CHAP, in that order.

Server IP/Name	Enter the domain name or IP address for the primary server.
Server Secret	Enter the key to access the primary server.

4. Click *OK*.

SCIM servers

System for Cross-domain Identity Management (SCIM) is an open-standard protocol that facilitates the exchange of identity data between platforms. See the following RFCs for more information:

- See RFC 7642 for information about definitions, overview, concepts, and requirements.
- See RFC 7643 for information about the core schema.
- See RFC 7644 for information about the protocol.

SCIM is based on a client-server model where the client is usually an identity provider (IdP) that maintains a directory of user identities, and the server is typically a service provider (SP). The client sends user and group information to the server, enabling automatic provisioning of users and groups between the SP and IdP.

With SCIM servers, FortiGate can communicate with an Identity Provider (IdP) using the SCIM 2.0 protocol, which facilitates the automatic provisioning of users and groups on FortiGate. Once users and groups are provisioned on FortiGate, they can be used with SAML to provide user authentication.

The *config system global* command includes SCIM settings:

```
config system global
    set scim-http-port <integer>
    set scim-https-port <integer>
    set scim-server-cert <string>
end
```

Option	Description
`scim-http-port <integer>`	Specify the port on which the SCIM server will listen for HTTP requests (default = 44558).
`scim-https-port <integer>`	Specify the port on which the SCIM server will listen for HTTPS requests (default = 44559).
`scim-server-cert <string>`	Specify the certificate that will be used if the HTTPS protocol is being used to communicate with the SCIM client. The certificate used by FortiGate must be trusted by the SCIM client.

To allow SCIM access in the GUI:

1. Go to *Network* > *Interfaces*.
2. Create or edit an interface.
3. In the *Administrative Access* section, select *SCIM*.
4. Click *OK*.

To allow SCIM access in the CLI:

For IPv4:

```
config system interface
    edit <name>
        append allowaccess scim
    next
end
```

For IPv6:

```
config system interface
    edit <name>
        ...
        config ipv6
            append ip6-allowaccess scim
        end
    next
end
```

To specify the SCIM server certificate and HTTP(S) port in the CLI:

```
config system global
    set scim-http-port <port>
    set scim-https-port <port>
    set scim-server-cert <certificate>
end
```

To configure SCIM clients in the CLI:

```
config user scim
    edit <name>
        set status {enable| disable}
        set base-url <string>
        set authe-method {token | base}
        set token-certificate <name>
        set secret <passwd>
        set certificate <name>
        set client-identity-check {enable| disable}
    next
end
```

Option	Description
`status {enable| disable}`	Enable/disable System for Cross-domain Identity Management (SCIM).
`base-url <string>`	Server URL to receive SCIM create, read, update, delete (CRUD) requests.
`auth-method {token | base}`	TLS client authentication methods (default = bearer token).
`token-certificate <name>`	Certificate for token verification.

Option	Description
`secret <passwd>`	Secret for token verification or base authentication.
`certificate <name>`	Certificate for client verification during TLS handshake.
`client-identity-check {enable| disable}`	Enable/disable client identity check.

To add a SCIM client to SAML server configuration in the CLI:

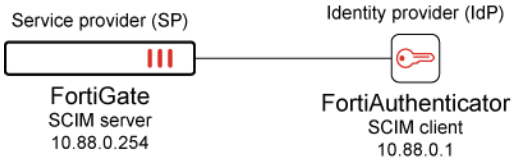
```
config user saml
    edit <name>
        set scim-client <name>
        set scim-group-attr-type {display-name | external-id}
    next
end
```

FortiGate as SCIM server and FAC as SCIM client

In this example, FortiGate is configured as the SCIM server (SP), and FortiAuthenticator is configured as the SCIM client (IdP). Two groups are configured on FortiAuthenticator: `IT` and `Pochiya clan`. The groups contain the following users:

- The `IT` group contains three users: `admin`, `sk`, and `sy`.
- The `Pochiya clan` group contains two users: `naynay` and `kiki`.

Upon successful configuration, users and groups are provisioned on FortiGate. This setup can leverage SAML to provide access to authenticated users.



To configure FortiGate as an SCIM server:

1. Allow SCIM access on an interface:
   In IPv4:

   ```
   config system interface
       edit port2
           append allowaccess scim
       next
   end
   ```

   In IPv6:

   ```
   config system interface
       edit port2
   ```

```
        ...
        config ipv6
            append ip6-allowaccess scim
        end
    next
end
```

2. Specify the SCIM server certificate:

```
config system global
    set scim-server-cert SCIM-Demo-CA-SSL
end
```

3. Configure SCIM client entries:

```
config user scim
    edit "SCIM-server-to-FAC"
        set status enable
        set base-url https://10.88.0.254/scim/v2
        set auth-method base
        set secret **********
        set certificate REMOTE_Cert_2
        set client-identity-check enable
    next
end
```

To configure FortiAuthenticator as an SCIM client:

1. On FortiAuthenticator, use your admin account to log in.
2. Click *Authentication > SCIM > Service Provider*.
3. Click *Create New*, and configure the following settings:

Option	Description
Name	Test-SCIM
SCIM endpoint	https://10.88.0.254:44559/scim/v2
Access token	**********

4. Configure the remaining settings as required, and click *Save*.
5. Once the setting is saved, double-click the newly created entry to open the settings pane, and click *Sync* to automatically add existing users to the SCIM server.

See Service providers for more information

The SCIM endpoint and access token must match the `base-url` and `secret` respectively, as configured on the FortiGate.

Furthermore, an initial synchronization is necessary to commence provisioning for the first time. However, when alterations to identities occur in the IdP, including creation, updating, and deletion, these changes are automatically synchronized with SP in accordance with the SCIM protocol.

To verify whether users and groups are provisioned on FortiGate:

```
# diagnose test scim list-clients
There are 1 clients in vdom.
id:1 name:SCIM-server-to-FAC
```

```
# diagnose test scim list-groups SCIM-server-to-FAC

IT
pochiya clan
total:2
```

```
# diagnose test scim list-users SCIM-server-to-FAC

admin
kiki
naynay
sk
sy
```

To add the SCIM group to a SAML server configuration:

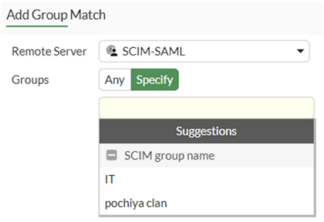
```
config user saml
    edit "SCIM-SAML"
        set scim-client "SCIM-server-to-FAC"
        set scim-group-attr-type {display-name | external-id}
    next
end
```

Modify the SAML user group used in the firewall policy

The configuration in this topic incorporates all the SCIM groups configured on FortiAuthenticator and enables all users to authenticate. However, if you want to limit authentication to users who belong to specific groups, such as the IT group, the following additional configuration is necessary:

To modify the SAML user group used in the firewall policy in the GUI:

1. Go to *User & Authentication* > *User Groups* and click *Create New*. *Firewall* is selected as the default *Type*.
2. Enter the group name, such as *saml-scim*.
3. In the *Remote Groups* section, click *Add*.
4. Set *Remote Server* to the SAML user, *SCIM-SAML*.
5. For the *Groups*, select *Specify*, then click the text box to see the groups pulled from SCIM client as *Suggestions*.



6. Select the required group then click *OK*.

To modify the SAML user group used in the firewall policy in the CLI:

```
config user group
    edit saml-scim
        set member "SCIM-SAML"
        config match
            edit 1
                set server-name "SCIM-SAML"
                set group-name "IT"
            next
        end
    next
end
```

For brevity, only the commands relevant to this enhancement are included. See SAML on page 3573 for more information about configuring SP and IdP.

Using SAML with a SCIM group in a firewall policy NEW

SCIM groups can now be used in firewall policies, eliminating the need for local group mapping and simplifying identity-based access control.

In this scenario, the user authenticates to the FortiGate using SAML, and SCIM is used to synchronize user and group information from the IdP through FortiAuthenticator (FAC) to the FortiGate. Because the SCIM group is referenced in the firewall policy, FortiGate knows which SCIM group the user belongs to after authentication. As a result, access decisions are enforced based on group membership received from SCIM, without needing any local group mapping or additional identity processing. This makes configuration simpler and more intuitive, as policies can directly reflect IdP-defined groups.

The `config user group` command includes SCIM settings:

```
config user group
    edit <id>
        set group-type scim
        set scim-group-attr-type {display-name | external-id}
        set scim-groups <string>
        set scim-user-attr-type {user-name | display-name | external-id | email}
        set scim-users <string>
    next
end
```

Option	Description
`group-type {scim}`	Set the `group-type` to `scim` to access additional SCIM options.
`scim-group-attr-type {display-name | external-id}`	Group attribute type used to match SCIM groups (default = display-name): • `display-name`: Display name. • `external-id`: External ID.
`scim_groups <string>`	Names of the SCIM groups.
`scim-user-attr-type {user-name | display-name | external-id | email}`	User attribute type used to match SCIM users (default = user-name): • `user-name`: User name. • `display-name`: Display name. • `external-id`: External ID. • `email`: Email.
`scim_users <string>`	Names of SCIM users.

To configure:

1. On FortiGate, enable and configure SCIM:

   The `base-url` defines the root path of the SAML service API. User information pushed by FortiAuthenticator is sent to this URL.

```
config user scim
    edit "scim_test"
        set id 2
        set status enable
        set base-url "https://10.1.100.64/scim/v2"
        set auth-method base
        set secret *
        set client-identity-check disable
        set cascade enable
    next
end
```

2. View user information pushed by SCIM:

```
# diagnose test scim list-users scim_test
userName        displayName     externalId      email   active
local1  NULL    local7  NULL    True
test1   NULL    local6  test1@fortinet.com      True
```

3. Configure SAML IdP:

```
config user saml
    edit "FAC_IdP"
        set entity-id "https://10.1.100.64:1003/remote/saml/metadata"
        set single-sign-on-url "https://10.1.100.64:1003/remote/saml/login/"
        set idp-entity-id "https://10.1.100.101/saml-idp/fgtvm/metadata/"
        set idp-single-sign-on-url "https://10.1.100.101/saml-idp/fgtvm/login/"
        set idp-single-logout-url "https://10.1.100.101/saml-idp/fgtvm/logout/"
```

```
        set idp-cert "REMOTE_Cert_1"
        set scim-client "scim_test"
        set user-name "username"
        set group-name "group"
        set digest-method sha256
    next
end
```

4. Configure a SCIM group:

```
config user group
    edit "scim_group"
        set group-type scim
        set member "FAC_IdP"
        set scim-user-attr-type user-name
        set scim-group-attr-type display-name
        set scim-users "local1"
    next
end
```

5. Add the SCIM group to the firewall policy:

```
config firewall policy
    edit 2
        set uuid 46a03322-4e50-51f1-059d-4e05169f3f21
        set srcintf "port2"
        set dstintf "port3"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set nat enable
        set groups "scim_group"
    next
end
```

6. After the user completes SAML authentication, check the firewall authentication list on FortiGate:

```
# diagnose firewall auth list

10.1.100.99, local1
        src_mac: 00:0c:29:3b:e1:12
        type: fw, id: 0, duration: 277, idled: 0
        expire: 300, allow-idle: 300
        server: FAC_IdP
        packets: in 11527 out 4983, bytes: in 12268358 out 1873022
        group_id: 3
        group_name: scim_group

----- 1 listed, 0 filtered ------
```

7. Check the log to verify that the traffic to the internet has used the correct firewall policy and user group:

```
1: date=2026-06-01 time=11:52:37 eventtime=1780339956990025702 tz="-0700" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="root" srcip=10.1.100.99 srcport=7624
srcintf="port2" srcintfrole="undefined" dstip=131.253.33.203 dstport=443 dstintf="port3"
dstintfrole="undefined" srccountry="Reserved" dstcountry="United States" sessionid=953 proto=6
action="server-rst" policyid=2 policytype="policy" poluuid="46a03322-4e50-51f1-059d-
4e05169f3f21" user="local1" group="scim_group" authserver="FAC_IdP" service="HTTPS"
trandisp="snat" transip=172.16.200.64 transport=7624 appcat="unscanned" duration=133
sentbyte=4399 rcvdbyte=10246 sentpkt=14 rcvdpkt=21 fctuid="2F30CF98569149CC9254544BE472ABA7"
unauthusersource="forticlient" srcdomain="fosqa.local" srcremote=0.0.0.0 sentdelta=0
rcvddelta=40 durationdelta=5 sentpktdelta=0 rcvdpktdelta=1
```

Bearer token authentication for SCIM servers

Bearer token authentication can be used for SCIM to improve security between the SCIM server and client. Bearer tokens generated by FortiOS can be temporary or long lived, and the SCIM client can verify the token.

Use the `execute gen-token` command to generate bearer tokens. The command is available for each VDOM.

```
execute gen-token <type> <string> <algorithm> <expire>
```

Option	Description
`<type>`	Type for token generation: • `cert`: certificate • `key`: pre-shared key
`<string>`	Name of the certificate or pre-shared key for token generation.
`<algorithm>`	Algorithm for token generation. Ensure that the signing algorithm supports the type. Available algorithms: • When the type is `cert`: `RS256`, `RS384`, `RS512`, `ES256`, `ES384`, or `ES512`. • When the type is `key`: `HS256`, `HS384`, or `HS512`.
`<expire>`	Expire interval, in hours (1 - 32767, 0 = long lived). When the token expires, administrators must manually generate another token on FortiGate and copy it to the SCIM client.

Use the `config user scim` command to configure token verification.

Example

This example explains how to generate a certificate type of bearer token and configure verification of the bearer token.

When generating a bearer token on FortiGate, remember:

- You can use any of the built-in or custom certificates available in the local certificate store.
- You must select a signing algorithm that supports the certificates.
- When the bearer token expires, the administrator must manually generate a new bearer token on FortiGate and copy the token to the SCIM client.

To generate a bearer token:

1. Enter `execute gen-token cert` to display the list of certificates available in the local certificate store:
   Custom and built-in certificates are displayed, and either can be used to generate tokens. In this example, `FGT401E-II-SAN-all` is a custom certificate, and `Fortinet_CA_SSL` is a built-in certificate.

```
# execute gen-token cert
<string>    Certificate or preshared-key for token generation.
Available certificates:
FGT401E-II-SAN-all      local
Fortinet_CA_SSL     local
Fortinet_CA_Untrusted   local           "
Fortinet_Factory        local           "
Fortinet_Factory_Backup local
Fortinet_GUI_Server     local           "
Fortinet_SSL    local
Fortinet_SSL_DSA1024    local
Fortinet_SSL_DSA2048    local
Fortinet_SSL_ECDSA256   local
Fortinet_SSL_ECDSA384   local
Fortinet_SSL_ECDSA521   local
Fortinet_SSL_ED448      local
Fortinet_SSL_ED25519    local
Fortinet_SSL_RSA1024    local
Fortinet_SSL_RSA2048    local
Fortinet_SSL_RSA4096    local
```

2. Enter `?` to display additional information:

```
# execute gen-token cert
Please specify <certificate or preshared-key> <algorithm> <expire>.
```

3. Enter the certificate name and `?` to display additional information:
   In this example, the `FGT401E-II-SAN-all` certificate is specified.

```
# execute gen-token cert FGT401E-II-SAN-all
<Algorithm>    Algorithm for token generation.
Please ensure that the signing algorithm matches the type of certificate and its corresponding
private key.
Available Algorithms:
RS256, RS384, RS512, ES256, ES384, ES512 for cert.
HS256, HS384, HS512 for key.
```

4. Enter the algorithm and `?` to display additional information:
   In this example, the `RS256` algorithm is specified.

```
# execute gen-token cert FGT401E-II-SAN-all RS256
<Expire>    Expire interval in hours, 0 for long live token.
```

5. Enter the expiration value:
   In this example, no expiration time (`0`) is specified.

   ```
   # execute gen-token cert FGT401E-II-SAN-all RS256 0
   ```

   The following values display after successful token generation:

   ```
   Added key:iat val:1732547333
   Generated token:eyAidHlwIjogIkpXVCIsICJhbGciOXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
   ```

6. Copy the `Generated token` to your SCIM client.

   ```
   Generated token:eyAidHlwIjogIkpXVCIsICJhbGciOXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
   ```

   - On FortiAuthenticator, the generated token is called an *access token*.
   - For Azure, the generated token is called a *secret token*.

To configure verification of the bearer token:

1. Configure the certificate name or pre-shared key to use for verification of the bearer token.
   In this example, a certificate (`cert`) type of bearer token is configured. You must specify the same certificate used to generate the bearer token:

   ```
   config user scim
       edit "SCIM-server-to-FAC"
           set id 1
           set status enable
           set base-url "https://10.1.100.7/SCIM-server-to-FAC/scim/v2/"
           set auth-method token
           set token-certificate "FGT401E-II-SAN-all"
           set certificate "REMOTE_Cert_1"
           set client-identity-check disable
       next
   end
   ```

   When using a pre-shared key (`key`), you must specify the same value used to generate the bearer token, for example:

   ```
   config user scim
       edit "SCIM-server-to-FAC"
           set id 1
           set status enable
           set base-url "https://10.1.100.8/SCIM-server-to-FAC/scim/v2/"
           set auth-method token
           set secret ENC mSoyZXvQ/tM1v1VOuS31DOrCZRNQ383JiXXXXXXXXXXXXXXXXXXXXX"
           set certificate "REMOTE_Cert_2"
           set client-identity-check disable
       next
   end
   ```

FortiTokens

FortiTokens are security tokens used as part of a multi-factor authentication (MFA) system on FortiGate and FortiAuthenticator. A security token is a 6-digit or 8-digit (configurable) one-time password (OTP) that is used to authenticate one's identity electronically as a prerequisite for accessing network resources. FortiToken is available as either a mobile or a physical (hard) token. Mobile tokens can be purchased as a license, or consumed with points as part of the FortiIdentity Cloud (formerly FortiToken Cloud) service.

FortiToken Mobile and physical FortiTokens store their encryption seeds on the cloud. FortiToken Mobile seeds are generated dynamically when the token is provisioned. They are always encrypted whether in motion or at rest.

You can only register FortiTokens to a single FortiGate or FortiAuthenticator for security purposes. This prevents malicious third parties from making fraudulent requests to hijack your FortiTokens by registering them on another FortiGate or FortiAuthenticator. If re-registering a FortiToken Mobile or Hard Token on another FortiGate is required, you must contact Fortinet Customer Support.

To migrate FortiToken Mobile tokens from FortiAuthenticator to FortiIdentity Cloud, see Migrating FTM tokens from FortiAuthenticator to FortiIdentity Cloud.

Common usage for FortiTokens includes:

- Applying MFA to a VPN dialup user connecting to the corporate network
- Applying MFA to FortiGate administrators
- Applying MFA to firewall authentication and captive portal authentication

The MFA process commonly involves:

- **Something you know**: User password
- **Something you have**: The FortiToken OTP

A third factor of authentication is added to the authentication process:

- **Something you are**: Your fingerprint or face

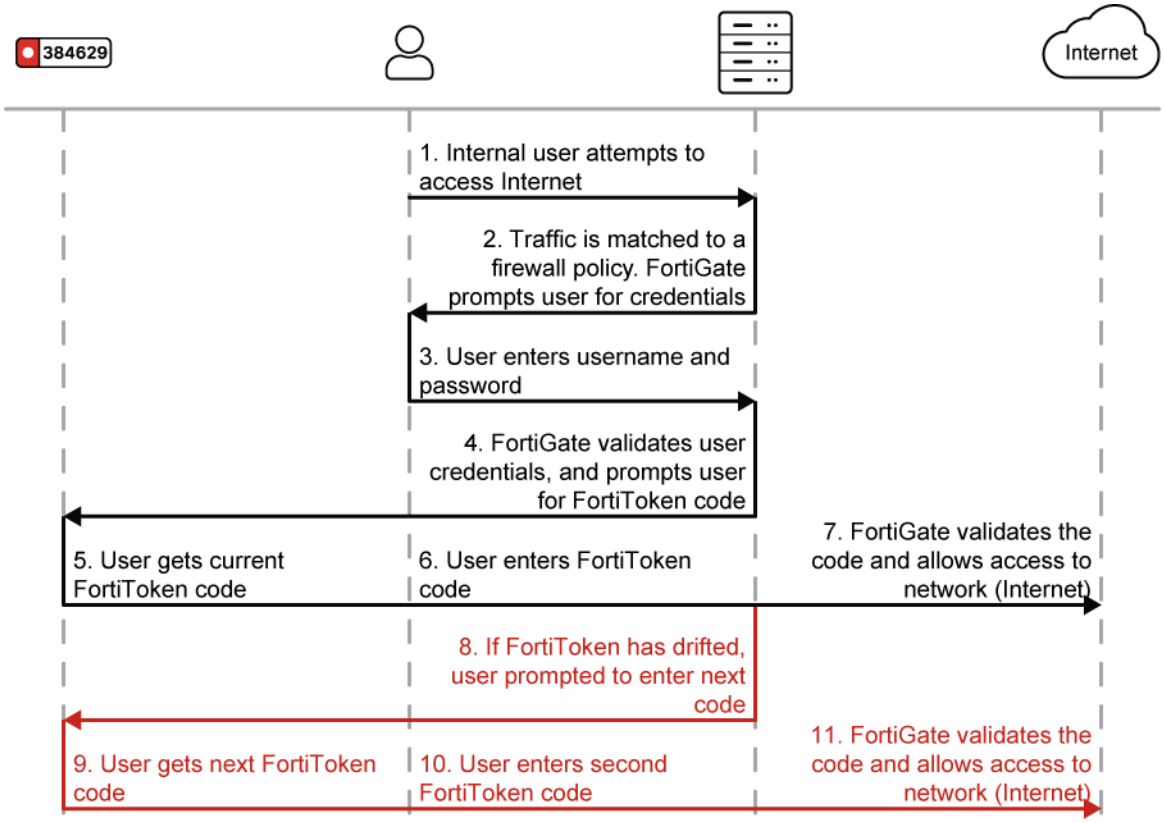
To enable the third factor, refer to the Activating FortiToken Mobile on a mobile phone on page 3637 section.

The following illustrates the FortiToken MFA process:

1. The user attempts to access a network resource.
2. FortiOS matches the traffic to an authentication security policy and prompts the user for their username and password.
3. The user enters their username and password.
4. FortiOS verifies their credentials. If valid, it prompts the user for the FortiToken code.
5. The user views the current code on their FortiToken. They enter the code at the prompt.
6. FortiOS verifies the FortiToken code. If valid, it allows the user access to network resources.

If the FortiToken has drifted, the following must take place for the FortiToken to resynchronize with FortiOS:

1. FortiOS prompts the user to enter a second code to confirm.
2. The user gets the next code from the FortiToken. They enter the code at the prompt.
3. FortiOS uses both codes to update its clock to match the FortiToken.



This section includes the following topics to quickly get started with FortiTokens:

- FortiToken Mobile quick start on page 3635
- FortiIdentity Cloud on page 3642
- Registering hard tokens on page 3642
- Managing FortiTokens on page 3645
- FortiToken Mobile Push on page 3647
- Synchronizing LDAP Active Directory users to FortiIdentity Cloud using the two-factor filter on page 3649
- Enable FortiIdentity Cloud seats directly from FortiGate on page 3653
- FortiGuard distribution of updated Apple certificates for push notifications on page 3655
- Configure FTM push with dynamic IP handling in the GUI on page 3656
- Troubleshooting and diagnosis on page 3658

FortiToken Mobile quick start

FortiToken Mobile is an OATH compliant, event- and time-based one-time password (OTP) generator for mobile devices. It provides an easy and flexible way to deploy and provision FortiTokens to your end users through mobile devices. FortiToken Mobile produces its OTP codes in an application that you can download onto your Android or iOS mobile device without the need for a physical token.

You can download the free FortiToken Mobile application for Android from the Google Play Store, and for iOS from the Apple App Store.

This section focuses on quickly getting started and setting up FortiToken Mobile for use on a FortiGate:

- Registering FortiToken Mobile on page 3635
- Provisioning FortiToken Mobile on page 3636
- Activating FortiToken Mobile on a mobile phone on page 3637
- Applying multi-factor authentication on page 3642

Registering FortiToken Mobile

To deploy FortiToken Mobile for your end users, you must first register the tokens on your FortiGate. After registering the tokens, you can assign them to your end users.

Each FortiGate comes with two free FortiToken Mobile tokens. These tokens should appear under *User & Authentication > FortiTokens*. If no tokens appear, you may import them. Ensure that your FortiGate is registered and has internet access to connect to the FortiToken servers to import the tokens.

To import FortiTokens from the FortiGate GUI:

1. Go to *User & Authentication > FortiTokens*.
2. Click the *Import Free Trial Tokens* icon at the top. The two free tokens are imported.

To import FortiTokens from the FortiGate CLI:

```
# execute fortitoken-mobile import 0000-0000-0000-0000-0000
# show user fortitoken
```

If only one free token appears, you can first delete that token and then follow the procedure to import the two free tokens from either the GUI or the CLI.

If you have the FortiToken Mobile redemption certificate, you can register FortiToken Mobile on a FortiGate.

To register FortiToken Mobile from the FortiGate GUI:

1. Go to *User & Authentication > FortiTokens* and click *Create New*. The *New FortiToken* dialog appears.
2. For the *Type* field, select *Mobile Token*.
3. Locate the 20-digit code on the redemption certificate and type it in the *Activation Code* field.
4. Click *OK*. The token is successfully registered.

If you attempt to add invalid FortiToken serial numbers, there is no error message. FortiOS does not add invalid serial numbers to the list.

To register FortiToken Mobile from the FortiGate CLI:

```
# execute fortitoken-mobile import <20-digit activation code>
# show user fortitoken
```

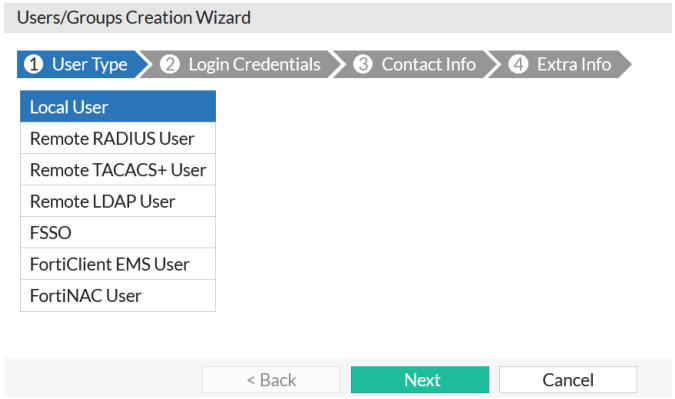
FortiToken Mobile stores its encryption seeds on the cloud. You can only register it to a single FortiGate or FortiAuthenticator.

Provisioning FortiToken Mobile

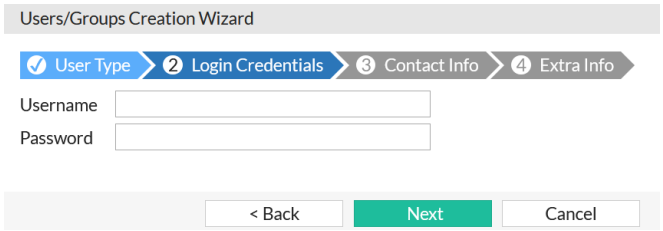
Once registered, FortiTokens need to be provisioned for users before they can be activated. In this example, you will provision a mobile token for a local user. Similar steps can be taken to assign FortiTokens to other types of users.

To create a local user and assign a FortiToken in the FortiGate GUI:

1. Go to *User & Authentication* > *User Definition*, and click *Create New*. The *Users/Groups Creation Wizard* appears.
2. In the *User Type* tab, select *Local User*, and click *Next*.

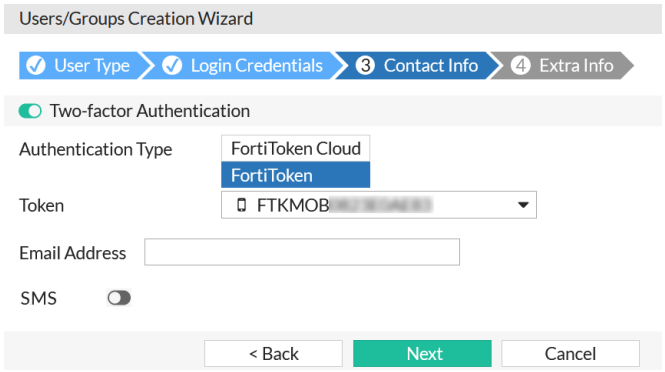


3. In the *Login Credentials* tab, enter a *Username* and *Password* for the user, and click *Next*.

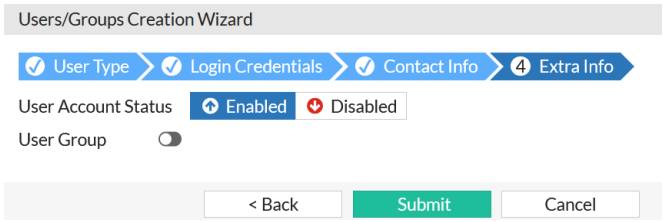


4. In the *Contact Info* tab:
   a. Enable the *Two-factor Authentication* toggle.
   b. Select *FortiToken* for *Authentication Type*.

c. Select a *Token* to assign to the user from the drop-down list.
d. Enter the user's email address in the *Email Address* field. This is the email where the user will receive the QR code for activation of the FortiToken.
e. Click *Next*.



5. In the *Extra Info* tab, make sure the *User Account Status* field is set to Enabled. You can also optionally assign the user to a user group by enabling the *User Group* toggle.



6. Click *Submit*. An activation code should be sent to the created user by email or SMS, depending upon the delivery method configured above.

FortiGate has the *Email Service* setting configured using the server *notifications.fortinet.net* by default. To see configuration, go to *System > Settings > General*.

The activation code expires if not activated within the 3-day time period by default. However, the expiry time period is configurable.

To configure the time period (in hours) for FortiToken Mobile, using the CLI:

```
config system global
    set two-factor-ftm-expiry <1-168>
end
```

To resend the email or SMS with the activation code, refer to the Managing FortiTokens on page 3645 section.

Activating FortiToken Mobile on a mobile phone

After your system administrator provisions your token, you receive a notification with an activation code and expiry date via SMS or email. If you do not activate your token by the expiry date, you must contact your system

administrator so that they can reassign your token for activation.

Platforms that support FortiToken Mobile:

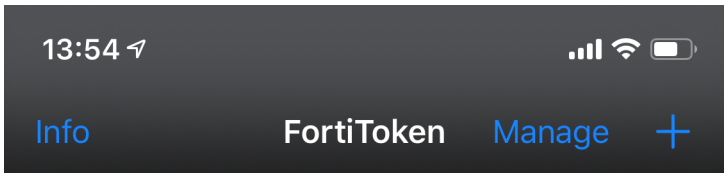
Platform	Device and firmware support
iOS	iPhone, iPad, and iPod Touch with iOS 6.0 and later.
Android	Phones and tablets with Android Jellybean 4.1 and later.
Windows	Windows 10 (desktop and mobile), Windows Phone 8.1, and Windows Phone 8. FortiToken is a Windows Universal Platform (UWP) application. To download FortiToken for Windows 10 desktop and mobile platforms, see FortiToken for Windows on the Microsoft Store.

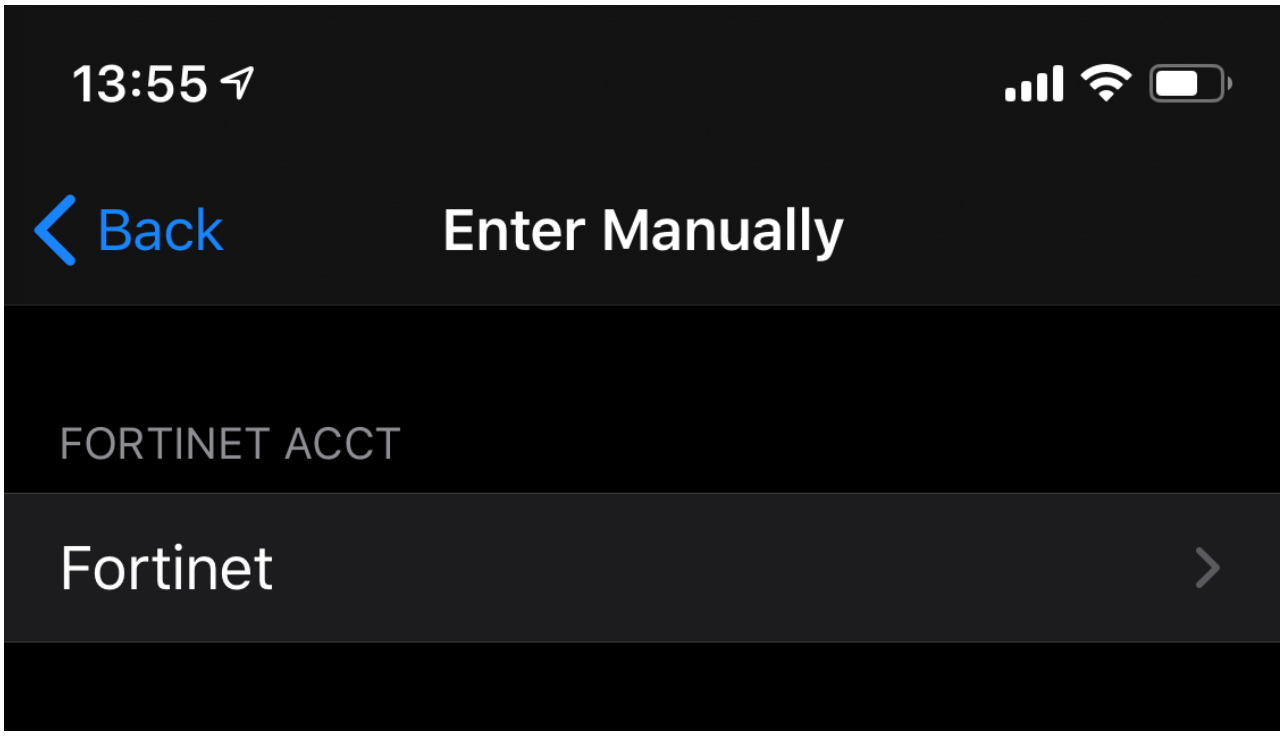
The following instructions describe procedures when using FortiToken Mobile for iOS on an iPhone. Procedures may vary depending on your device and firmware.

To activate FortiToken Mobile on iOS:

1. On your iOS device, tap on the FortiToken application icon to open the application. If this is your first time opening the application, it may prompt you to create a PIN for secure access to the application and tokens.

2. Tap on the + icon. The *Scan Barcode* screen appears.

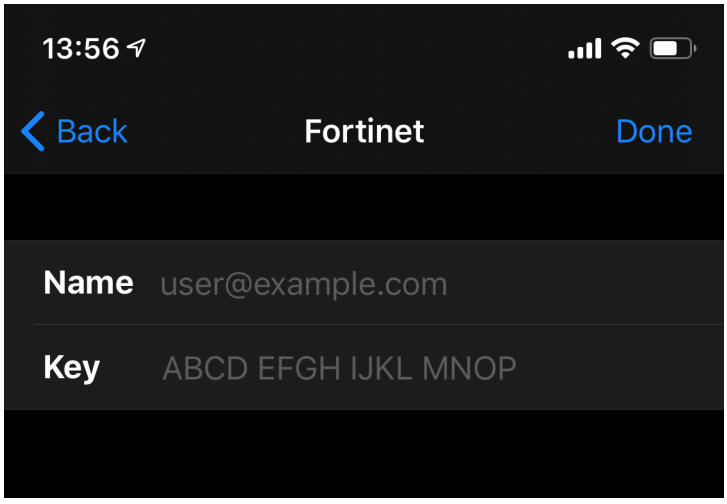


3. 

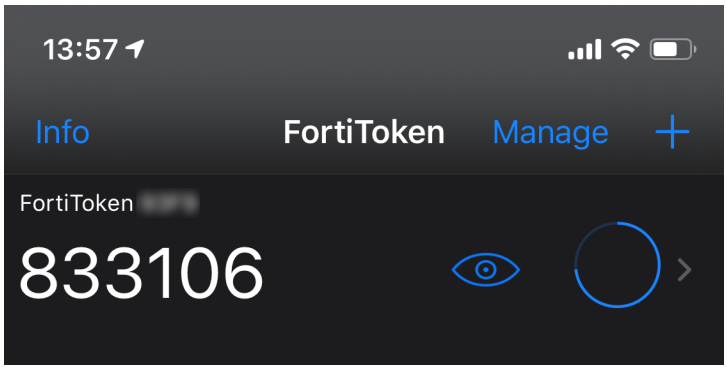
If you received the QR code via email, locate and scan the QR code in your email.

OR

If you received the activation key via SMS, tap on *Enter Manually* at the bottom of the screen, and tap on *Fortinet*. Enter your email address in the *Name* field, the activation key in the *Key* field, and tap *Done*.



4. FortiToken Mobile activates your token, and starts generating OTP digits immediately. To view or hide the OTP digits, tap the eye icon.

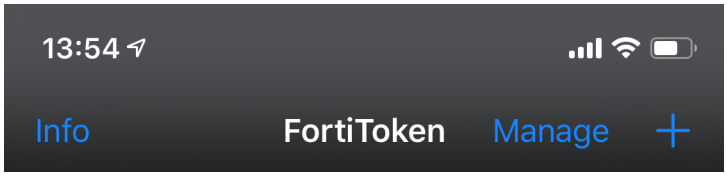


After you open the application, FortiToken Mobile generates a new 6-digit OTP every 30 seconds. All configured tokens display on the application homescreen.

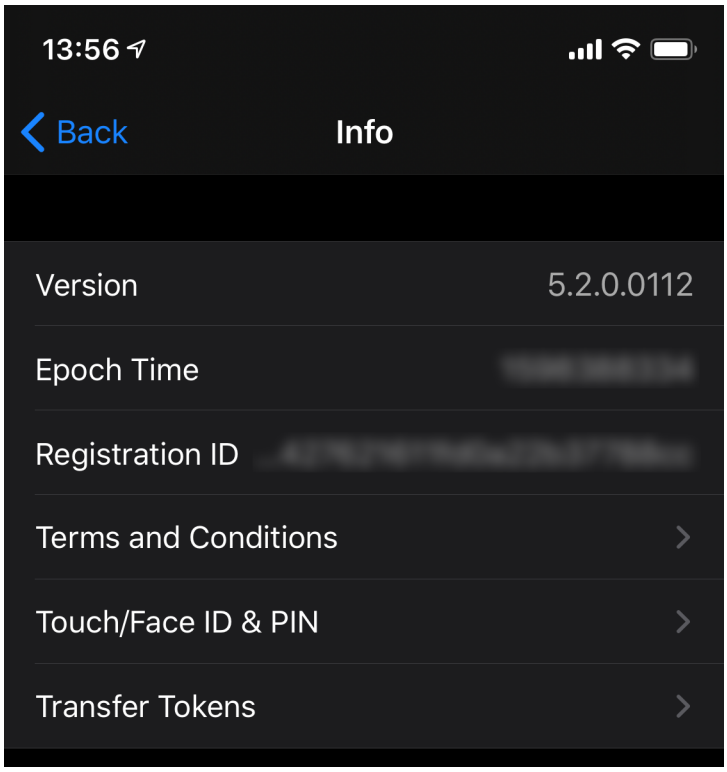
The FortiToken Mobile activation process described above caters to the MFA process that involves two factors (password and OTP) of the authentication process. A third factor (fingerprint or face) can be enabled as well.

To enable *Touch/Face ID* on iOS for FortiToken Mobile:

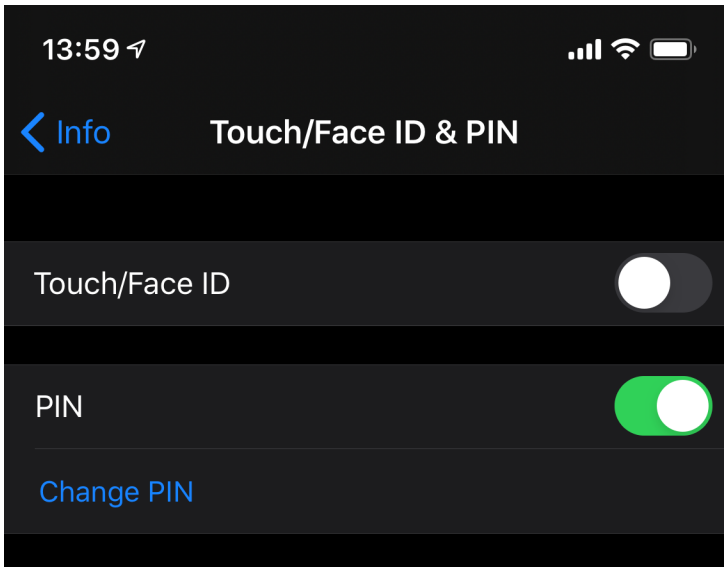
1. Open the FortiToken application and tap on *Info*.



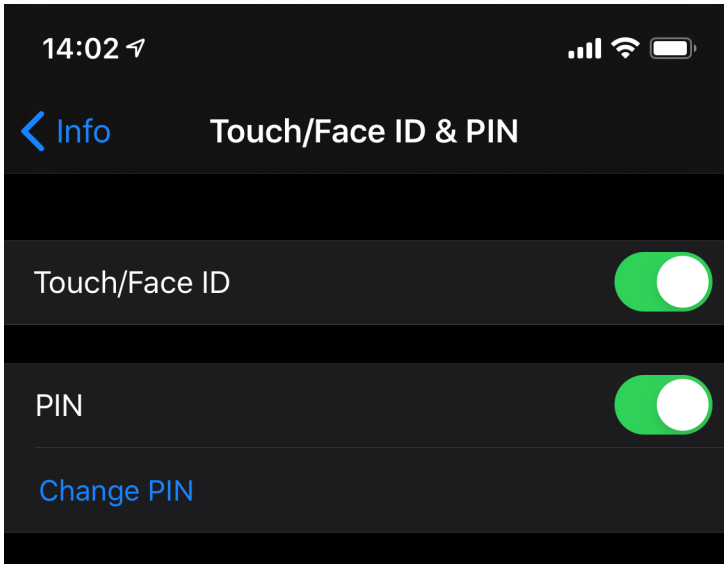
2. Tap on *Touch/Face ID & PIN*.



3. Enable and set up a 4-digit *PIN* for the application. The *PIN* is required to be enabled before you can enable *Touch/Face ID*.

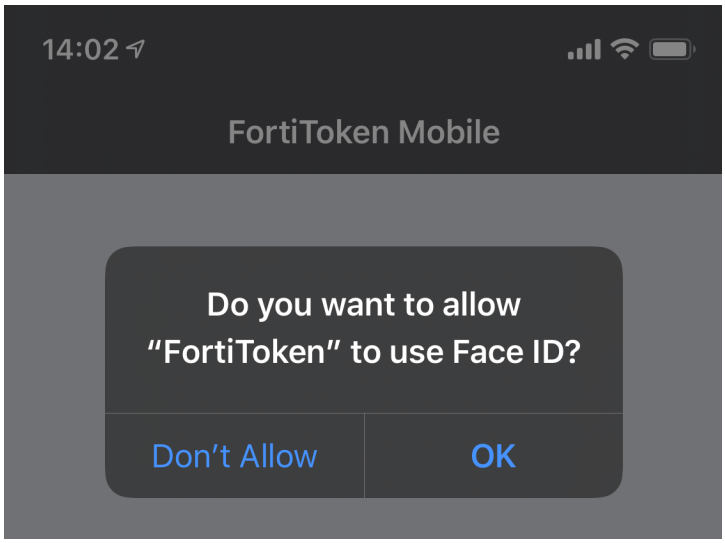


4. Enable *Touch/Face ID*.



> You cannot enable *Touch/Face ID* for FortiToken if *Touch/Face ID* is not set up and enabled for device unlock (*iPhone Unlock* in this case) on iOS. You must first set up and enable *Touch/Face ID* from *Settings* on your iOS device.

5. When prompted by iOS, allow the FortiToken application to use *Touch/Face ID* by tapping on *OK* in the prompt.



Applying multi-factor authentication

Multi-factor authentication (MFA) may also be set up for Agentless VPN users, administrators, firewall policy, wireless users, and so on. The following topics explain more about how you may use the newly created user in such scenarios:

- MFA for IPsec VPN: Add FortiToken multi-factor authentication on page 2985
- MFA for Administrators: Administrator account options on page 3706
- MFA with Captive Portal
- MFA for wireless users via Captive Portal
- Configuring FSSO firewall authentication on page 3680

FortiIdentity Cloud

FortiToken Cloud is now called FortiIdentity Cloud.

FortiIdentity Cloud is an Identity and Access Management as a Service (IDaaS) cloud service offering by Fortinet. It enables FortiGate and FortiAuthenticator customers to add MFA for their respective users, through the use of Mobile tokens or Hard tokens. It protects local and remote administrators as well as firewall and VPN users.

For information, see Getting started—FGT-FIC users in the FortiIdentity Cloud Administration Guide.

Registering hard tokens

Registering FortiTokens consists of the following steps:

1. Adding FortiTokens to FortiOS.
2. Activating FortiTokens.
3. Associating FortiTokens with user accounts.

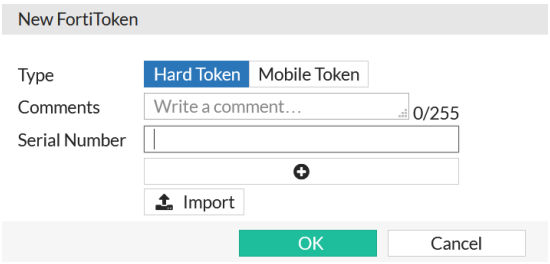
Adding FortiTokens to FortiOS

You can add FortiTokens to FortiOS in the following ways:

- Add FortiToken serial numbers using the GUI
- Add FortiToken serial numbers using the CLI
- Import FortiTokens using a serial number or seed file using the GUI

To manually add single hard token to FortiOS using the GUI:

1. Go to *User & Authentication* > *FortiTokens*.
2. Click *Create New*.
3. For *Type*, select *Hard Token*.
4. In the *Serial Number* field, enter one or more FortiToken serial numbers.
5. Click *OK*.

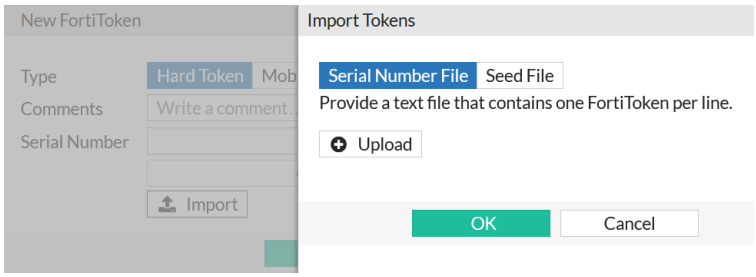


To add multiple FortiTokens to FortiOS using the CLI:

```
config user fortitoken
    edit <serial_number>
    next
    edit <serial_number2>
    next
end
```

To import multiple FortiTokens to FortiOS using the GUI:

1. Go to *User & Authentication* > *FortiTokens*.
2. Click *Create New*.
3. For *Type*, select *Hard Token*.
4. Click *Import*. The *Import Tokens* section slides in on the screen.



5. Select *Serial Number File*.

> Seed files are only used with FortiToken-200CD. These are special hardware tokens that come with FortiToken seeds on a CD. See the FortiToken Comprehensive Guide for details.

6. Click *Upload*.
7. Browse to the file's location on your local machine, select the file, then click *OK*.
8. Click *OK*.

Activating FortiTokens

You must activate the FortiTokens before starting to use them. FortiOS requires connection to FortiGuard servers for FortiToken activation. During activation, FortiOS queries FortiGuard servers about each token's validity. Each token can only be used on a single FortiGate or FortiAuthenticator. If tokens are already registered, they are deemed invalid for re-activation on another device. FortiOS encrypts the serial number and information before sending for added security.

To activate a FortiToken using the GUI:

1. Go to *User & Authentication* > *FortiTokens*.
2. Select the desired FortiTokens that have an *Available* status.
3. Click *Activate* from the menu above.
4. Click *Refresh*. The selected FortiTokens are activated.

To activate a FortiToken using the CLI:

```
config user fortitoken
    edit <token_serial_num>
        set status activate
    next
end
```

Associating FortiTokens with user accounts

You can associate FortiTokens with local user or administrator accounts.

To associate a FortiToken to a local user account using the GUI:

1. Ensure that you have successfully added your FortiToken serial number to FortiOS and that its status is *Available*.

2. Go to *User & Authentication* > *User Definition*. Edit the desired user account.
3. Enable *Two-factor Authentication*.
4. From the *Token* dropdown list, select the desired FortiToken serial number.
5. In the *Email Address* field, enter the user's email address.
6. Click *OK*.

To associate a FortiToken to a local user account using the CLI:

```
config user local
    edit <username>
        set type password
        set passwd "myPassword"
        set two-factor fortitoken
        set fortitoken <serial_number>
        set email-to "username@example.com"
        set status enable
    next
end
```

Before you can use a new FortiToken, you may need to synchronize it due to clock drift.

To associate a FortiToken to an administrator account, refer to the Administrator account options on page 3706 section.

Managing FortiTokens

This section focuses on the following:

- Resending an activation email on page 3645
- Locking/unlocking FortiTokens on page 3646
- Managing FortiTokens drift on page 3646
- Deactivating FortiTokens on page 3647
- Moving FortiTokens to another device on page 3647
- High availability clustering with FortiToken on page 3647

Resending an activation email

To resend an activation email/SMS for a mobile token on a FortiGate:

1. Go to *User & Authentication* > *User Definition* and edit the user.
2. Click *Send Activation Code Email* from the *Two-factor Authentication* section.

Locking/unlocking FortiTokens

To change FortiToken status to active or to lock:

```
config user fortitoken
    edit <token_serial_num>
        set status <active | lock>
    next
end
```

A user attempting to log in using a locked FortiToken cannot successfully authenticate.

Managing FortiTokens drift

If the FortiToken has drifted, the following must take place for the FortiToken to resynchronize with FortiOS:

1. FortiOS prompts the user to enter a second code to confirm.
2. The user gets the next code from the FortiToken. They enter the code at the prompt.
3. FortiOS uses both codes to update its clock to match the FortiToken.

If you still experience clock drift, it may be the result of incorrect time settings on your mobile device. If so, make sure that the mobile device clock is accurate by confirming the network time and the correct timezone.

If the device clock is set correctly, the issue could be the result of the FortiGate and FortiTokens being initialized prior to setting an NTP server. This will result in a time difference that is too large to correct with the synchronize function. To avoid this, selected Tokens can be manually drift adjusted.

To show current drift and status for each FortiToken:

```
# diagnose fortitoken info
FORTITOKEN DRIFT STATUS
FTK200XXXXXXXXC 0 token already activated, and seed won't be returned
FTK200XXXXXXXXE 0 token already activated, and seed won't be returned
FTKMOBXXXXXXXXA 0 provisioned
FTKMOBXXXXXXXX4 0 new
Total activated token: 0
Total global activated token: 0
Token server status: reachable
```

This command lists the serial number and drift for each configured FortiToken. You can check if it is necessary to synchronize the FortiGate and any particular FortiTokens.

To adjust Mobile FortiToken for drift:

```
# execute fortitoken sync <FortiToken_ID> <token_code1> <next_token_code2>
```

Deactivating FortiTokens

To deactivate FortiToken on a FortiGate:

1. Go to *User & Authentication* > *User Definition*.
2. Select and edit the user for which you want to deactivate the token.
3. Disable the *Two-factor Authentication* toggle.
4. Click *OK*. The token will be removed from the user's *Two-factor Authentication* column. The user will also be removed from the token's *User* column under *User & Authentication* > *FortiTokens*.

Moving FortiTokens to another device

FortiTokens can only be activated on a single FortiGate or FortiAuthenticator. To move FortiTokens to another device, you would first have to reset the registered FortiTokens on a device and then reactivate them on another device.

To reset Hard tokens registered to a FortiGate appliance (non-VM model), you can reset all hardware FTK200 tokens from the Support Portal, or during RMA transfer. See the Migrating users and FortiTokens to another FortiGate KB article, for more information.

The above process will reset all Hard tokens and you cannot select individual tokens to reset.

To reset FortiToken Mobile, a single Hard token, a Hard token registered to a VM, and so on, an administrator must contact Customer Support and/or open a ticket on the Support Portal.

Once reset, the FortiTokens can be activated on another FortiGate or FortiAuthenticator.

High availability clustering with FortiToken

When setting up an HA cluster with multiple FortiGate or FortiAuthenticator units, you must register and apply any FortiToken Mobile licenses to the primary unit. This can be done before or after configuring the unit for HA operation. After HA is established, all tokens are automatically synchronized across all cluster members. As a result, only a single FortiToken Mobile license is required per HA cluster. FortiTokens cannot be provisioned from an unregistered FortiGate or a secondary units in an HA setup.

FortiToken Mobile Push

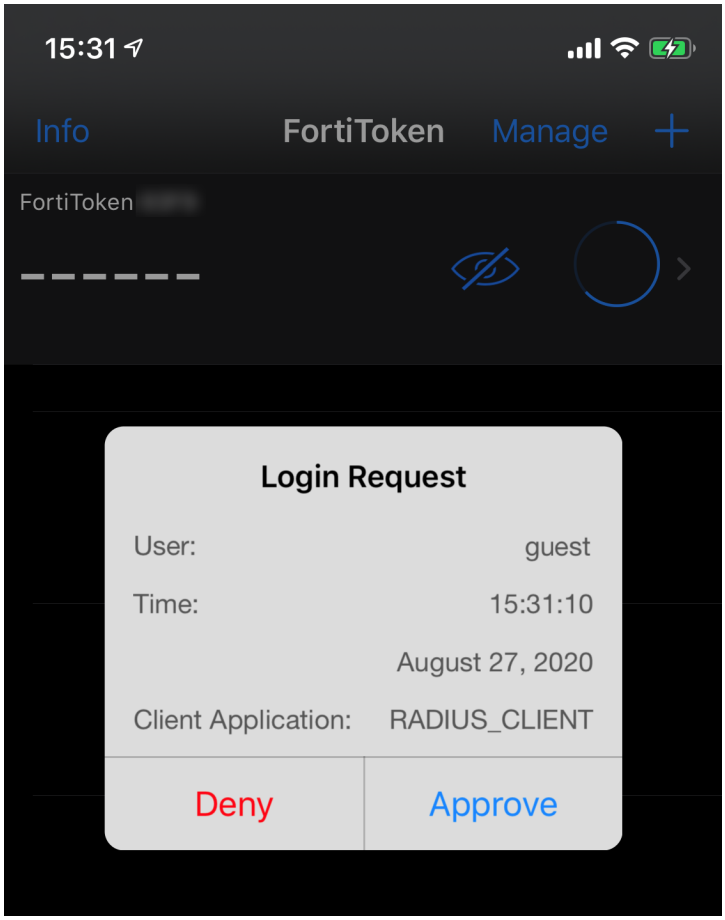
FortiToken Mobile Push allows authentication requests to be sent as push notifications to the end user's FortiToken Mobile application.

The FortiToken Mobile push service operates as follows:

1. FortiGate sends a DNS query to the FortiToken Mobile Push proxy server (*push.fortinet.com*).
2. FortiGate connects to the proxy server via an encrypted connection over TCP/443.
3. The proxy server handles the notification request by making a TLS connection with either Apple (for iOS) or Google (for Android) notification servers. Notification data may include the recipient, session, FortiGate

callback IP and port, and so on.

4. The notification service from either Apple or Google notifies the user's mobile device of the push request.
5. The FortiToken Mobile application on the user's mobile displays a prompt for the user to either *Approve* or *Deny* the request.



To configure FortiToken Mobile push services using the CLI:

```
config system ftm-push
    set proxy {enable | disable}
    set interface <string>
    set server-port [1-65535]
    set server-cert <certificate-name>
    set server <ip-address>
    set status enable
end
```

The default server port is 4433.

The server IP address is the public IP address of the FortiOS interface that FortiToken Mobile calls back to. FortiOS uses this IP address for incoming FortiToken Mobile calls.

If an SSL VPN user authenticates with their token, then logs out and attempts to reauthenticate within a minute, a *Please wait x seconds to login again* message displays. This replaces a previous error/permission denied message. The x value depends on the calculation of how much time is left in the current time step.

```
config system interface
    edit "guest"
        set allowaccess ftm
    next
end
```

FortiOS supports FortiAuthenticator-initiated FortiToken Mobile Push notifications for users attempting to authenticate through an SSL VPN and/or RADIUS server (with FortiAuthenticator as the RADIUS server).

Synchronizing LDAP Active Directory users to FortiIdentity Cloud using the two-factor filter

To synchronize Active Directory users and apply two-factor authentication using FortiIdentity Cloud (formerly FortiToken Cloud), two-factor authentication can be enabled in the `user ldap` object definition in FortiOS. By default, FortiOS retrieves all Active Directory users in the LDAP server with a valid email or mobile number (`mail` and `mobile` attributes), and synchronizes the users to FortiIdentity Cloud. Users are then created on FortiIdentity Cloud and activation is sent out using email or SMS.

Two-factor filters can be used to reduce the number of the Active Directory users returned, and only synchronize the users who meet the filter criteria.

```
config user ldap
    edit <name>
        set dn <string>
        set two-factor {disable | fortitoken-cloud}
        set two-factor-filter <string>
    next
end
```

`dn <string>`	Set the distinguished name used to look up entries on the LDAP server. The search for users and groups starts here based on what is defined.
`two-factor {disable | fortitoken-cloud}`	Enable/disable two-factor authentication: • `disable`: disable two-factor authentication • `fortitoken-cloud`: use the FortiIdentity Cloud service
`two-factor-filter <string>`	Set the filter used to synchronize users to FortiIdentity Cloud.

When configuring an LDAP connection to an Active Directory server, an administrator must provide Active Directory user credentials.

- To secure this connection, use LDAPS on both the Active Directory server and FortiGate. See Configuring an LDAP server on page 3509 and Configuring client certificate authentication on the LDAP server on page 3524.
- Apply the principle of least privilege. For the LDAP regular bind operation, do not use credentials that provide full administrative access to the Windows server when using credentials. See Configuring least privileges for LDAP admin account authentication in Active Directory on page 3517.

Two-factor filter examples

In the following examples, a `user ldap` object is defined to connect to an Active Directory on a Windows server. The search will begin in the root of the fortinet-fsso.com directory.

To configure a default LDAP server configuration without a two-factor filter:

```
config user ldap
    edit "ad-ldap-auth"
        set server <ip_address>
        set cnid "cn"
        set dn "dc=fortinet-fsso,dc=com"
        set type regular
        set two-factor fortitoken-cloud
        set username "cn=Administrator,cn=users,dc=fortinet-fsso,dc=com"
        set password **********
    next
end
```

When a two-factor filter is not used, all users in the Active Directory with a valid email or mobile number will be retrieved.

Example 1: specific users and email address with wildcard

With this `two-factor-filter`, users under fortinet-fsso.com that have oliver* in their username and *fortinet* in their email will be matched.

```
config user ldap
    edit "ad-ldap-auth"
        set two-factor-filter "(&(SAMAccountName=oliver*)(mail=*fortinet*))"
    next
end
```

Example 2: all users with matching email

With this `two-factor-filter`, all users under fortinet-fsso.com with *fortinet* in their email will be matched.

```
config user ldap
    edit "ad-ldap-auth"
        set two-factor-filter "(&(SAMAccountName=*)(mail=*fortinet*))"
    next
end
```

Example 3: all users in a group

With this `two-factor-filter`, all users within the group fortinet-fsso.com > Testing > ftc-users will be matched.

```
config user ldap
    edit "ad-ldap-auth"
        set two-factor-filter "(&(objectCategory=Person)(sAMAccountName=*)( memberOf=cn=ftc-
users,ou=Testing,dc=fortinet-fsso,DC=com))"
    next
end
```

Example configuration

In this example, Active Directory users are configured to be synchronized to FortiIdentity Cloud. The same two-factor filter is used from example 1 and searches the Active Directory for users named oliver* with email *fortinet*.

Before configuring the FortiGate:

1. Gather the information to connect to the Active Directory server through LDAP. Include all necessary fields, such as the server IP, port, CN name identifier, DN for the start of the search, bind type, and username associated with a regular bind.
2. Consider the users or groups that require two-factor authentication and should be synchronized. If necessary, group the users under the same group in the Active Directory.
3. If using a two-factor filter, formulate the `two-factor-filter` string to limit the match. For this example, `(&(SAMAccountName=oliver*)(mail=*fortinet*))`.
4. Test the filter by using the FortiOS CLI to perform a quick LDAP search:

   ```
   # diagnose test authserver ldap-search <server_ip> 389 "ou=Testing,dc=fortinet-fsso,DC=com" cn
   Administrator@fortinet-fsso.com PASSWORD 0 '(&(SAMAccountName=oliver*)(mail=*fortinet*))' 2

   searching 'ou=Testing,dc=fortinet-fsso,DC=com, cn=cn' on 10.1.100.131:389 for
   (Administrator@fortinet-fsso.com, PASSWORD), secure(0), filter((&(SAMAccountName=oliver*)
   (mail=*fortinet*))), flag(0x2), page_no(0)...
   CN=oliver2022,OU=Testing,DC=Fortinet-FSSO,DC=COM (oliver2022, 0 entries)
   ```

   The user, `oliver2022`, was found.
5. Estimate how many users will be retrieved, and ensure that the FortiIdentity Cloud account has enough user licenses to support the number of users.

To configure Active Directory users to be synchronized to FortiIdentity Cloud:

1. Configure the user LDAP settings:

```
config user ldap
    edit "ad-ldap-auth"
        set server "10.1.100.131"
        set cnid "cn"
        set dn "dc=fortinet-fsso,dc=com"
        set type regular
        set two-factor fortitoken-cloud
        set username "cn=Administrator,cn=users,dc=fortinet-fsso,dc=com"
        set password **********
        set two-factor-filter "(&(SAMAccountName=oliver*)(mail=*fortinet*))"
    next
end
```

2. In the background, the FortiGate FAS daemon scans the LDAP server for users to be synchronized based on the two-factor filter pattern, but will not send them to the FortiIdentity Cloud server yet. Optionally, verify the users that are retrieved from the Active Directory based on the filter:

```
# diagnose fortitoken-cloud debug enable
# diagnose debug enable
# diagnose fortitoken-cloud sync
...
fas_sync_ftc[2788]: Sending packet to FTC server: "IP-of-FTC-server" Port: 8686(length:444)
fas_sync_ftc[2792]: FTC User Sync Packet(length:444):
POST /api/v1/user_sync HTTP/1.1
Host: ftc.fortinet.com
Connection: keep-alive
User-Agent: FortiGate-401E v7.0.6,build****
Content-Type: application/json
Content-Length: 246
{"users":[{"username":"oliver2022","vdom":"vdom1","email":"o****@fortinet.com","mobile_
number":"XXXXXXXXXXX","user_data":1,"action":"create"}],"sn":"FG4H1E5819900000","cluster_
members":[ "FG4H1E5819900000" ],"group_name":"FGT400D","group_id":"0"}
Reminder: User sync packet not actually sent out because of diagnose purpose!
```

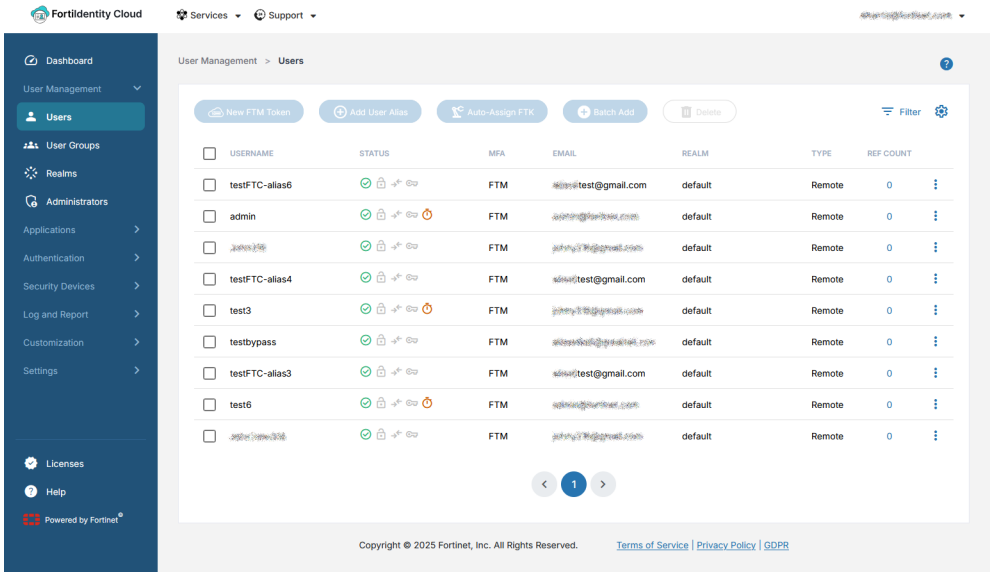
   As expected, only the user that matches the current filter is returned.

3. Manually trigger the synchronization process with FortiIdentity Cloud:

```
# execute fortitoken-cloud sync
```

   The user is added to FortiIdentity Cloud, and an activation email or SMS message is sent to the user.

4. In FortiIdentity Cloud, go to *User Management > Users* to verify that the user was added.



If the activation email was sent, but user has not downloaded and activated the mobile token yet, a pending symbol appears in the *Status* column (such as for the *admin*, *test6*, and *test3* users).

5. In FortiOS, add the `ad-ldap-auth` object in a user group. The user group can be used for VPN, firewall authentication, and so on.

> The `ldap user` object should not be used in remote LDAP user groups that require group matching because it is not supported.

Enable FortiIdentity Cloud seats directly from FortiGate

Three seats of FortiIdentity Cloud (formerly FortiToken Cloud) are available for FortiGates with FortiCare Premium or FortiCare Elite. A valid support contract must be registered to a FortiCare/FortiCloud account with Fortinet Technical Support. See Product registration with FortiCare on page 115 for more information.

You can access the seats by enabling two-factor authentication within a user or administrator configuration.

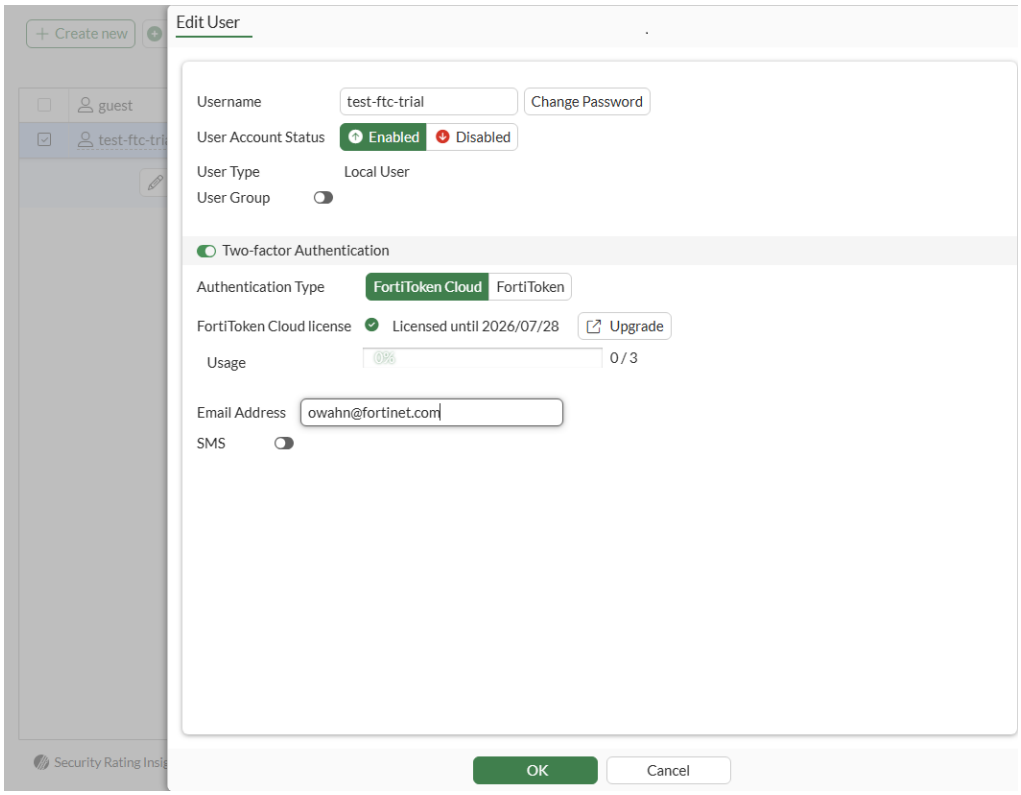
The FortiIdentity Cloud seats remains active as long as the support contract is valid.

To enable the FortiIdentity Cloud seats for a local user:

1. Go to *User & Authentication* > *User Definition*.
2. Click *Create new*.
3. Configure settings as needed.
4. Enable *Two-factor Authentication*.
5. Set *Authentication Type* to *FortiToken Cloud*.

   The *FortiToken Cloud license* field displays *Licensed* and the expiration date of the license.

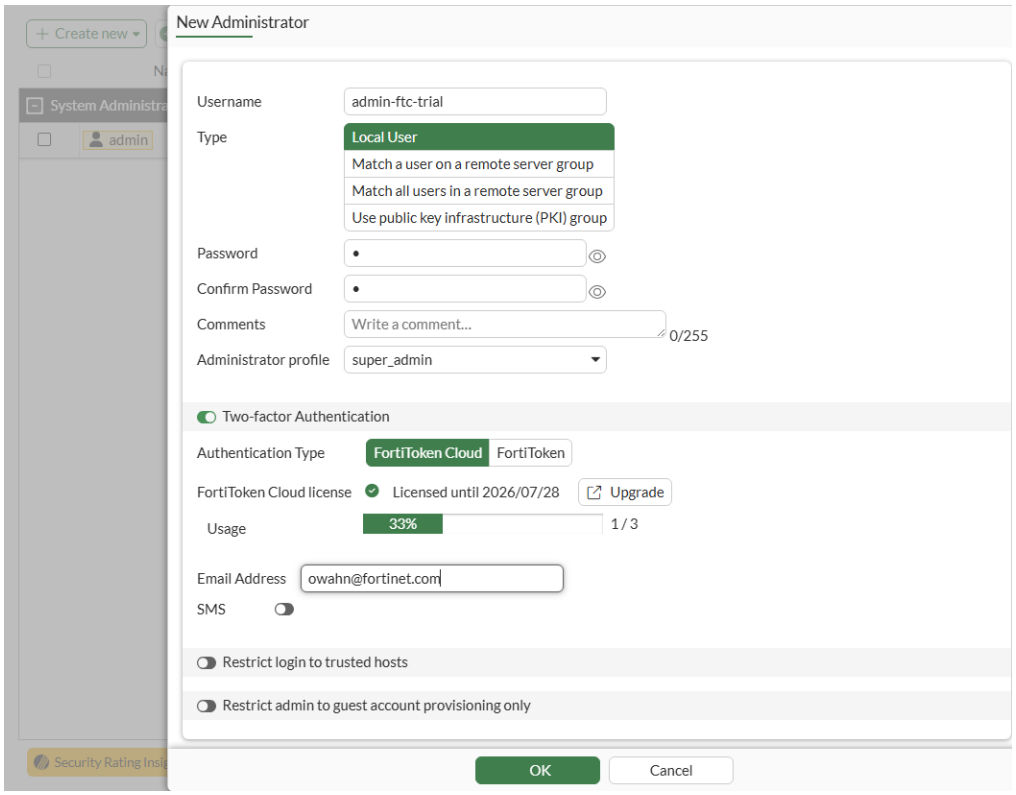
   The *Usage* field displays the number of available and used seats. In this example, 0 of 3 (*0/3*) seats are used because the user configuration hasn't been saved yet.



6. Complete the remaining fields, and click *OK*.
   A FortiIdentity Cloud enrollment email is sent to the provided email.

To enable the FortiIdentity Cloud for an administrator:

1. Go to *System* > *Administrators*.
2. Click *Create new* > *Administrator*.
3. Enable *Two-factor Authentication*.
4. Set *Authentication Type* to *FortiToken Cloud*.
   The *FortiToken Cloud license* field displays *Licensed* and the expiration date of the license.
   The *Usage* field displays the number of available and used seats. In this example, 1 of 3 (*1/3*) seats are used. After this administrator configuration is saved, a second seat will be used.



5. Complete the remaining fields, and click *OK*.
   A FortiIdentity Cloud enrollment email is sent to the provided email.

To view the FortiIdentity Cloud status in the CLI:

```
# execute fortitoken-cloud show
FortiToken Cloud service status: licensed. Region code: 01. Region name: eu.
Service balance: 1.00 users. Expiration date: 2026-07-29. Customer ID: 1336761.
FortiToken Cloud account number of users: 2, max number of users: 3.
```

FortiGuard distribution of updated Apple certificates for push notifications

Push notifications for iPhone (for the purpose of two-factor authentication) require a TLS server certificate to authenticate to Apple. As this certificate is only valid for one year, a service extension allows FortiGuard to distribute updated TLS server certificates to FortiGate when needed.

FortiGuard update service updates local Apple push notification TLS server certificates when the local certificate is expired. FortiGuard update service also reinstalls certificates when the certificates are lost.

You can verify that the feature is working on the FortiGate by using the CLI shell.

To verify certificate updates:

1. Using FortiOS CLI shell, verify that all certificates are installed:

```
/data/etc/apns # ls -al
drwxr-xr-x    2 0        0       Tue Jan 15 08:42:39 2019          1024 .
drwxr-xr-x   12 0        0       Tue Jan 15 08:45:00 2019          2048 ..
-rw-r--r--    1 0        0       Sat Jan 12 00:06:30 2019          2377 apn-dev-cert.pem
-rw-r--r--    1 0        0       Sat Jan 12 00:06:30 2019          1859 apn-dev-key.pem
-rw-r--r--    1 0        0       Sat Jan 12 00:06:30 2019          8964 apn-dis-cert.pem
-rw-r--r--    1 0        0       Sat Jan 12 00:06:30 2019          4482 apn-dis-key.pem
```

2. Rename all current Apple certificates.
   Apple push notification no longer works after you rename the certificates.

```
/data/etc/apns # mv apn-dis-cert.pem apn-dis-cert.pem.save
/data/etc/apns # mv apn-dev-key.pem apn-dev-key.pem.save
/data/etc/apns # mv apn-dev-cert.pem apn-dev-cert.pem.save
/data/etc/apns # mv apn-dis-key.pem apn-dis-key.pem.save
/data/etc/apns # ls -al
drwxr-xr-x   2 0  0   Tue Jan 15 08:51:15 2019   1024 .
drwxr-xr-x  12 0  0   Tue Jan 15 08:45:00 2019   2048 ..
-rw-r--r--   1 0  0   Sat Jan 12 00:06:30 2019   2377 apn-dev-cert.pem.save
-rw-r--r--   1 0  0   Sat Jan 12 00:06:30 2019   1859 apn-dev-key.pem.save
-rw-r--r--   1 0  0   Sat Jan 12 00:06:30 2019   8964 apn-dis-cert.pem.save
-rw-r--r--   1 0  0   Sat Jan 12 00:06:30 2019   4482 apn-dis-key.pem.save
```

3. Run a FortiGuard update, and verify that all certificates are installed again:

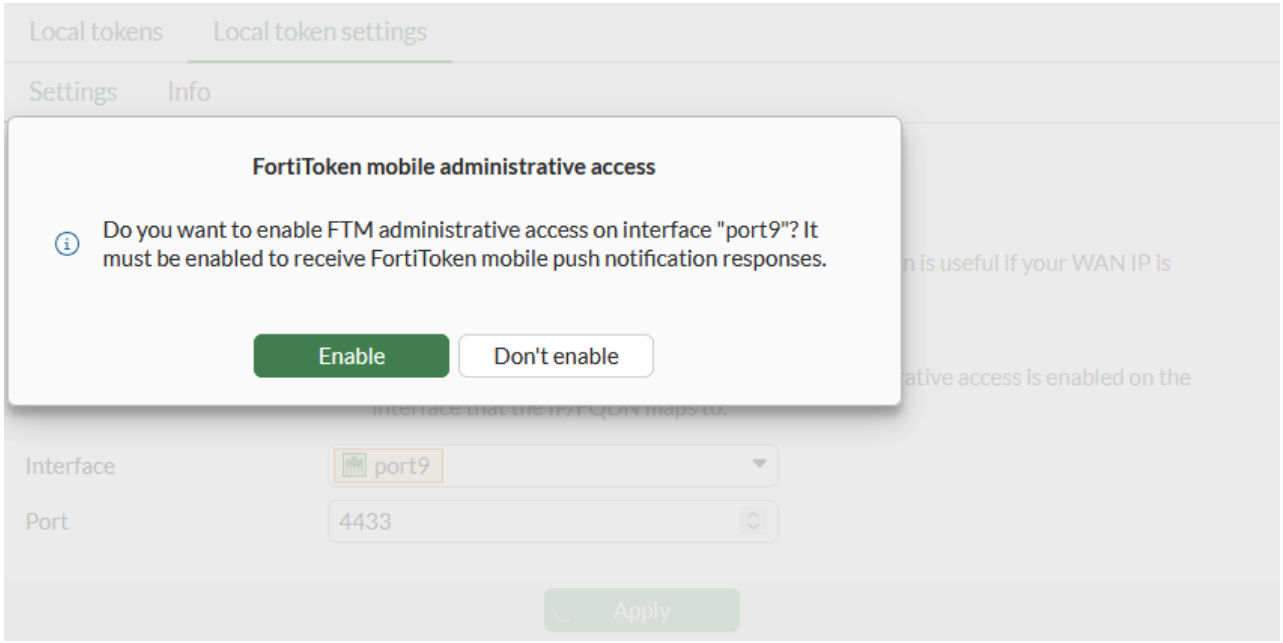
```
/data/etc/apns # ls -al
drwxr-xr-x   2 0  0   Tue Jan 15 08:56:20 2019   1024 .
drwxr-xr-x  12 0  0   Tue Jan 15 08:56:15 2019   2048 ..
-rw-r--r--   1 0  0   Sat Jan 12 00:06:30 2019   2377 apn-dev-cert.pem.save
-rw-r--r--   1 0  0   Sat Jan 12 00:06:30 2019   1859 apn-dev-key.pem.save
-rw-r--r--   1 0  0   Tue Jan 15 08:56:20 2019   2167 apn-dis-cert.pem  <-- downloaded from
FortiGuard
-rw-r--r--   1 0  0   Sat Jan 12 00:06:30 2019   8964 apn-dis-cert.pem.save
-rw-r--r--   1 0  0   Tue Jan 15 08:56:20 2019   1704 apn-dis-key.pem   <-- downloaded from
FortiGuard
-rw-r--r--   1 0  0   Sat Jan 12 00:06:30 2019   4482 apn-dis-key.pem.save
-rw-r--r--   1 0  0   Tue Jan 15 08:56:20 2019     41 apn-version.dat   <-- downloaded from
FortiGuard
/data/etc/apns #
```

Configure FTM push with dynamic IP handling in the GUI

A new GUI-based configuration page for FTM push has been added to complement the existing CLI setup. Previously, users had to manually enter the IPv4 address or domain name of the FortiToken Mobile push services server, which required updates when the IP address changed. The new option allows users to select an interface instead. The system will automatically use the current IP address of the selected interface, making it ideal for environments where the WAN IP is dynamically assigned.

To configure FTM push dynamic IP handling in the GUI:

1. Go to *User & Authentication* > *FortiTokens* and select the *Local token settings* tab.
2. Set *Push notifications* to *Enabled*.
3. By default, *Direct push responses to* in set to *Interface*. The IP address of the selected interface is used. This is useful if your WAN IP is dynamically assigned.
4. Set *Interface* to a physical interface.
5. Optionally, change the default listening *Port*.
6. Click *Apply*.
7. If FortiToken mobile administrative access is not enabled on the selected interface, a pop-up message will ask to enable it. Click *Enable*.



FortiToken mobile administrative access can also be configured in the interface settings; see Configure administrative access to interfaces on page 229.

When an interface is configured in the GUI, the `server` option is empty in the CLI:

```
config system ftm-push
    set proxy enable
    set interface "port9"
    set server ''
    set server-port 4433
    set server-cert "Fortinet_GUI_Server"
    set status enable
end
```

If the IP/FQDN is configured in the GUI, after upgrading from previous FortiOS version, the `interface` option (`server-ip` in previous versions) will be empty and the `server` options will contain the static IP address or FQDN:

```
config system ftm-push
    set proxy enable
```

```
    set interface ''
    set server "172.18.60.184"
    set server-port 4433
    set server-cert "Fortinet_GUI_Server"
    set status enable
end
```

Troubleshooting and diagnosis

This section contains some common scenarios for FortiTokens troubleshooting and diagnosis:

- FortiToken Statuses on page 3658
- Recovering trial FortiTokens on page 3659
- Recovering lost Administrator FortiTokens on page 3660
- Agentless VPN with multi-factor authentication expiry timers on page 3661

FortiToken Statuses

When troubleshooting FortiToken issues, it is important to understand different FortiToken statuses. FortiToken status may be retrieved either from the CLI or the GUI, with a slightly different naming convention.

Before you begin, verify that the FortiGate has Internet connectivity and is also connected to both the FortiGuard and registration servers:

```
# execute ping fds1.fortinet.com
```

```
# execute ping directregistration.fortinet.com
```

```
# execute ping globalftm.fortinet.net
```

The `globalftm.fortinet.net` server is the Fortinet Anycast server added in FortiOS 6.4.2.

If there are connectivity issues, retrieving FortiToken statuses or performing FortiToken activation could fail. Therefore, troubleshoot connectivity issues before continuing.

To retrieve FortiToken statuses:

- In the CLI:

  ```
  # diagnose fortitoken info
  ```

- In the GUI:
  Go to *User & Authentication* > *FortiTokens*.

Various FortiToken statuses in either the CLI or the GUI may be described as follows:

CLI	GUI	Description
`new`	*Available*	Newly added, not pending, not activated, not yet assigned.
`active`	*Assigned*	Assigned to a user, hardware token.
`provisioning`	*Pending*	Assigned to a user and waiting for activation on the FortiToken Mobile app.
`provisioned`	*Assigned*	Assigned to user and activated on the FortiToken Mobile app.
`provision timeout`		Token provided to user but not activated on the FortiToken Mobile app. To fix, the token needs to be re-provisioned and activated in time.
`token already activated, and seed won't be returned`	*Error*	Token is locked by FortiGuard FDS. The hardware token was already activated on another device and locked by FDS.
`locked`		Either manually locked by an Administrator (`set status lock`), or locked automatically, for example, when the token is unassigned and the FortiCare FTM provisioning server was unreachable to process that change.

Recovering trial FortiTokens

You can recover trial FortiTokens if deleted from a FortiGate, or if stuck in a state where it is not possible to provision to a user.

When a token is stuck in an unusual state or with errors, delete the FortiTokens from the unit and proceed to recover trial FortiTokens.

To recover trial tokens via the GUI:

1. Go to *User & Authentication* > *FortiTokens*.
2. Click the *Import Free Trial Tokens* button at the top. The two free trial tokens are recovered.

To recover trial tokens via the CLI:

```
# execute fortitoken-mobile import 0000-0000-0000-0000-0000
```

- Before attempting to recover the trial tokens, both the tokens should be deleted from the unit first.
- If VDOMs are enabled, trial tokens are in the management VDOM (`root` by default).

Following error codes might come up in the CLI:

- If the device is not registered:

```
# execute fortitoken-mobile import 0000-0000-0000-0000-0000
import fortitoken license error: -7571
```

- If the serial number format is incorrect:

```
# execute fortitoken-mobile import 0000-0000-0000-0000-00
import fortitoken license error: -7566
```

Recovering lost Administrator FortiTokens

If an Administrator loses their FortiToken or the FortiToken is not working, they will not be able to log into the admin console through the GUI or the CLI. If there is another Administrator that can log into the device, they may be able to reset the two-factor settings configured for the first Administrator, or create a new Admin user for them. Note that a *super_admin* user will be able to edit other admin user settings, but a *prof_admin* user will not be able to edit *super_admin* settings.

In the case where there are no other administrators configured, the only option is to flash format the device and reload a backup config file. You must have console access to the device in order to format and flash the device. It is recommended to be physically on site to perform this operation.

Before formatting the device, verify that you have a backup config file. You may or may not have the latest config file backed up, though you should consider using a backed up config file, and reconfigure the rest of the recent changes manually. Otherwise, you may need to configure your device starting from the default factory settings.

To recover lost Administrator FortiTokens:

1. If you have a backed up config file:
   a. Open the config file and search for the specific admin user. For representational purposes we will use `Test` in our example.

```
# edit "Test"
    set accprofile "super_admin"
    set vdom "root"
    set two-factor fortitoken
    set fortitoken "FTKXXXXXXXXXX"
    set email-to "admin@email.com"
    set password *********
    next
end
```

   b. Once you find the settings for the `Test` user, delete the `fortitoken`-related settings:

```
# edit "Test"
    set accprofile "super_admin"
    set vdom "root"
    set password *********
    next
end
```

2. Format the boot device during a maintenance window and reload the firmware image using instructions in the Formatting and loading FortiGate firmware image using TFTP KB article.
3. Once the reload is complete, log into the admin console from the GUI using the default admin user credentials, and go to *Configuration > Restore* from the top right corner to reload your config file created in Step 1 above.

4. Once the FortiGate reboots and your configuration is restored, you can log in with your admin user credentials.

Agentless VPN with multi-factor authentication expiry timers

When Agentless VPN is configured with multi-factor authentication (MFA), sometimes you may require a longer token expiry time than the default 60 seconds.

To configure token expiry timers using the CLI:

```
config system global
    set two-factor-ftk-expiry <number of seconds>
    set two-factor-ftm-expiry <number of seconds>
    set two-factor-sms-expiry <number of seconds>
    set two-factor-fac-expiry <number of seconds>
    set two-factor-email-expiry <number of seconds>
end
```

These timers apply to the tokens themselves and remain valid for as long as configured above. However, Agentless VPN does not necessarily accept tokens for the entire duration they are valid. To ensure SSLVPN accepts the token for longer durations, you need to configure the remote authentication timeout setting accordingly.

To configure the remote authentication timeout:

```
config system global
    set remoteauthtimeout <1-300 seconds>
end
```

Agentless VPN waits for a maximum of five minutes for a valid token code to be provided before closing down the connection, even if the token code is valid for longer.

> The `remoteauthtimout` setting shows how long Agentless VPN waits not only for a valid token to be provided before closing down the connection, but also for other remote authentication like LDAP, RADIUS, and so on.

PKI

The following topics include information about public key infrastructure (PKI):

- Configuring a PKI user on page 3662
- Using the SAN field for LDAP-integrated certificate authentication on page 3666
- Agentless VPN with certificate authentication on page 3392
- Agentless VPN with LDAP-integrated certificate authentication on page 3398

Configuring a PKI user

PKI users are users who are identified by a digital certificate they hold. Defining a PKI user in FortiOS specifies:

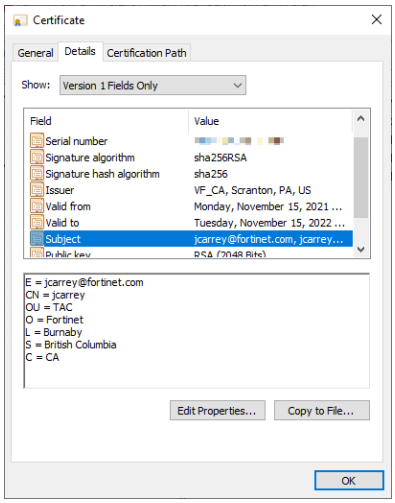
- Which CA certificate to use to validate the user's certificate
- The field and value of the user's certificate that FortiOS will check to verify a user

These peer users can then be used in a FortiGate user group, or as a peer certificate group used for IPsec VPN configurations that accept RSA certificate authentication.

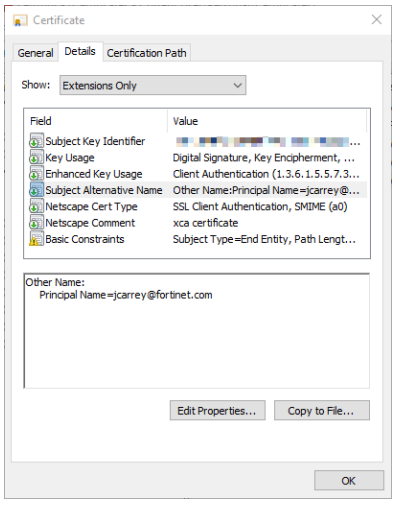
Example X.509 certificate

The following certificate demonstrates which FortiGate settings can be used to match on different fields.

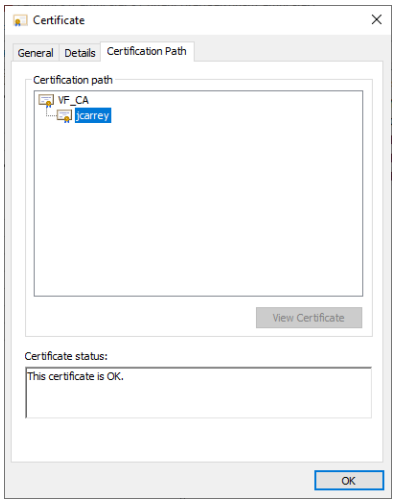
Subject:



Subject Alternative Name:



Certification path:



To configure a PKI user:

```
config user peer
    edit <name>
        set ca <string>
        set mandatory-ca-verify {enable | disable}
        set subject <string>
        set cn <string>
        set cn-type {string | email | FQDN | ipv4 | ipv6}
        set mfa-server <string>
        set mfa-username <string>
        set mfa-password <string>
        set mfa-mode {none | password | subject-identity}
    next
end
```

Command	Description
`ca <string>`	Specify which certificate on the FortiGate is used to validate the client's certificate. This can be any CA in the client's certificate chain. You may need to upload a CA certificate to the FortiGate specifically to identify PKI peer users (see CA certificate on page 3967).
`mandatory-ca-verify {enable | disable}`	Control the action if the CA certificate used to sign the client's certificate is not installed on the FortiGate (default = enable). Disabling this setting makes the FortiGate consider any certificate presented by the peer as valid. In the example certificate, the certification path shows that VF_CA signed jcarrey's certificate.
`subject <string>`	Enter the peer certificate name constraints.
`cn <string>`	Enter the peer certificate common name.
`cn-type {string | email | FQDN | ipv4 | ipv6}`	Set the peer certificate common name type: string, email, FQDN, IPv4 address, or IPv6 address. See CN on page 3665 for more details.

Command	Description
`mfa-server <string>`	Enter the name of a multi-factor authentication server defined under `config user ldap` for performing client access rights checks. See LDAP servers on page 3509 for more details.
`mfa-mode {none | password | subject-identity}`	Set the mode for remote peer authentication, either by password or subject identity extracted from certificate. See LDAP on page 3665 for more details.
`mfa-username <string>`	Enter the username for the remote multi-factor authentication server bind when the MFA mode is password.
`mfa-password <string>`	Enter the password for the multi-factor authentication server bind when the MFA mode is password.

Identifying users based on their client certificate

When the client's certificate is valid, or `mandatory-ca-verify` is disabled, the FortiGate can then inspect the certificate to check specific fields for matching values. There are three ways of specifying which certificate field to verify: by subject, CN, or LDAP. All string comparisons are case sensitive.

Subject

This basic method verifies that the subject string defined in the PKI user setting matches a value or substring in the subject field of the user certificate. Further matching is controlled in the following VPN certificate settings.

```
config vpn certificate setting
    set subject-match {substring | value}
    set subject-set {superset | subset}
    set cn-match {substring | value}
    set cn-allow-multi {enable | disable}
end
```

Command	Description
`subject-match {substring | value}`	Control how to do relative distinguished name (RDN) value matching with the certificate subject name: • substring: find a match if any string in the certificate subject name matches the name being searched for (such as `set subject jcarrey`). • value: find a match if any attribute value string in a certificate subject name is an exact match with the name being searched for (such as `set subject "OU=TAC"` or `set subject "C=CA, CN=jcarrey, OU=TAC"`).
`subject-set {superset | subset}`	Control how to do RDN value matching with the certificate subject name: • superset: a certificate only passes verification if it contains all the RDNs defined in the subject settings (such as `set subject "E = jcarrey@fortinet.com, CN = jcarrey, OU = TAC, O = Fortinet, L = Burnaby, S = British Columbia, C = CA"`). • subset: a certificate passes verification if the RDN is a subset of the certificate subject (such as `set subject "CN = jcarrey, OU = TAC"`).

Command	Description
`cn-match {substring | value}`	Control how to do CN value matching with the certificate subject name: • substring: find a match if any string in the certificate subject name matches the name being searched for. • value: find a match if any attribute value string in a certificate subject name is an exact match with the name being searched for.
`cn-allow-multi {enable | disable}`	Enable/disable allowing multiple CN entries with the certificate subject name (default = enable).

CN

Common name (CN) certificate verification compares the CN in the subject field with the configured string (such as `set cn "jcarrey"`. The following logic is used when configuring different CN types:

Type	Action
`string`	Based on the `cn-match` setting, perform a substring or exact match in the certificate subject.
`email`	Look for a match in the certificate subject.
`FQDN`	Look for a match in the certificate subject, then compare the mapped IP and client IP. The FQDN is only retrieved from the CN.
`ipv4`	Look for a match in the certificate subject, then compare the IP.
`ipv6`	Look for a match in the certificate subject, then compare the IP.

The CN type also controls the format checking of the CN string. In this example, if the CN type is set to email, the CN must be in email format (`set cn "jcarrey@fortinet.com"`).

LDAP

LDAP-integrated user authentication allows the FortiGate to check the connecting user against an LDAP server in two ways: through a username and password, or the certificate's principal name. The `password` method requires the username and password of each authenticating user to be entered, so it is not recommended when configuring PKI users. The `subject-identity` method is recommended.

The UPN in the user certificate's Subject Alternative Name (SAN) field is used to look up the user in the LDAP directory. The SAN in the certificate for UPN matching can the UPN on the AD LDAP server (default), RFC 822 Name (corporate email address), or DNS name. If a match is found, then authentication succeeds. This type of configuration scales well since only one PKI user needs to be created on the FortiGate. Connecting clients use their unique user certificate to match within the configured LDAP server. See Using the SAN field for LDAP-integrated certificate authentication on page 3666 for an example.

Using the SAN field for LDAP-integrated certificate authentication

Certificate-based authentication against Active Directory LDAP (AD LDAP) supports the UserPrincipalName (UPN), RFC822 Name (corporate email address) defined in the Subject Alternative Name (SAN) extension of the certificate, the DNS defined in the user certificate as the unique identifier in the SAN field for peer user certificates, and the Subject Common Name (CN) defined in the certificate.

```
config user ldap
    edit <name>
        set account-key-cert-field {othername | rfc822name | dnsname | cn}
    next
end
```

Command	Description
`account-key-cert-field {othername | rfc822name | dnsname | cn}`	Define subject identity field in certificate for user access right checking: • `othername`: match to UPN in SAN (default). • `rfc822name`: match to RFC822 email address in SAN. • `dnsname`: match to DNS name in SAN. • `cn`: match to CN in subject.

The LDAP server configurations are applied to the user peer configuration when the PKI user is configured.

```
config user peer
    edit <name>
        set ca <string>
        set cn <string>
        set mfa-server <string>
        set mfa-mode subject-identity
    next
end
```

When a user authenticates to the FortiGate for an administrative log in, Agentless VPN, IPsec dialup, or firewall authentication using a user certificate, it presents a signed certificate issued by a trusted CA to the FortiGate. The following sequence of events occurs as the FortiGate processes the certificate for authentication:

1. The FortiGate verifies if the certificate is issued by a trusted CA. If the CA is not a public CA, ensure that the CA certificate is uploaded and trusted by the FortiGate, and is applied to the user peer configurations (`set ca <string>`).
2. The FortiGate verifies that the CN field of the certificate matches the CN specified in the user peer configurations (`set cn <string>`).
3. If the `user peer` configuration has `mfa-server` configured and the `mfa-mode` is set to `subject-identity`, the FortiGate uses the unique identifier in the certificate to authenticate against the LDAP server.
   - If `set account-key-cert-field othername` is configured (the default setting), the FortiGate uses the UPN in the certificate's SAN field to authenticate against LDAP.
   - If `set account-key-cert-field rfc822name` is configured, the FortiGate uses the RFC822 Name in the certificate's SAN field to authenticate against LDAP.

- If `set account-key-cert-field dnsname` is configured, the FortiGate uses the DNS name in the certificate's SAN field to authenticate against LDAP.
- If `account-key-cert-field` is set to `cn`, then the FortiGate uses the CN in the certificate's subject to authenticate against LDAP.

4. By default, the FortiGate tries to match the UPN attribute on the AD LDAP. If this needs to be changed to another field, configure the `account-key-filter` setting on the LDAP configuration:

```
config user ldap
    edit <name>
        set account-key-filter <string>
    next
end
```

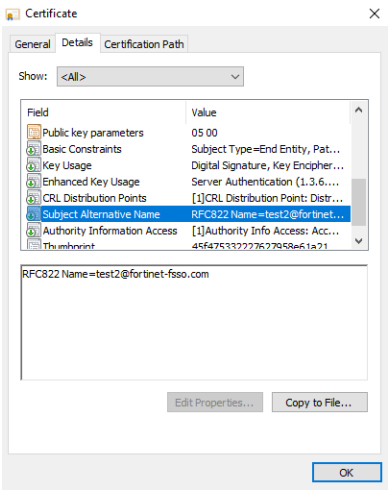
When configuring an LDAP connection to an Active Directory server, an administrator must provide Active Directory user credentials.

- To secure this connection, use LDAPS on both the Active Directory server and FortiGate. See Configuring an LDAP server on page 3509 and Configuring client certificate authentication on the LDAP server on page 3524.
- Apply the principle of least privilege. For the LDAP regular bind operation, do not use credentials that provide full administrative access to the Windows server when using credentials. See Configuring least privileges for LDAP admin account authentication in Active Directory on page 3517.

Example

In this example, a user certificate is issued by a customer's CA to a user. The user uses this certificate to authenticate to the Agentless VPN web portal. The administrator decides to use the RFC822 Name in the SAN field to authenticate against their corporate AD LDAP. The Active Directory attribute to check against the RFC822 Name field is the mail attribute.

User certificate information:



The configuration used in this example assumes the following:

- The CA certificate has already been uploaded to the FortiGate.
- The Agentless VPN configurations have already been configured, pending the assignment of the PKI user group.

To configure the authentication settings:

1. Configure the LDAP server:

```
config user ldap
    edit "ad-ldap-peer-user"
        set server "10.1.100.131"
        set cnid "cn"
        set dn "dc=fortinet-fsso,dc=com"
        set type regular
        set username "cn=Administrator,cn=users,dc=fortinet-fsso,dc=com"
        set password ENC XXXXXXXXXXXXXXXX
        set password-renewal enable
        set account-key-cert-field rfc822name
        set account-key-filter "(&(mail=%s)(!(UserAccountControl:1.2.840.113556.1.4.803:=2)))"
    next
end
```

   By default, the `account-key-filter` filters on the UPN attribute uses the following string: (&(**userPrincipalName=%s**)(!(UserAccountControl:1.2.840.113556.1.4.803:=2))).
   - (userPrincipalName=%s) matches the UPN attribute on the AD LDAP.
   - (!(UserAccountControl:1.2.840.113556.1.4.803:=2)) filters out inactive and locked AD accounts.
2. Configure the local peer user:

```
config user peer
    edit "peer-RFC822-name"
        set ca "CA_Cert_2"
        set cn "test2"
        set mfa-server "ad-ldap-peer-user"
        set mfa-mode subject-identity
    next
end
```

3. Configure the firewall user group for Agentless VPN authentication:

```
config user group
    edit "vpn-group"
        set member "peer-RFC822-name"
    next
end
```

4. Apply the user group to the Agentless VPN configuration and firewall policy.

Verification

When the Agentless VPN user authenticates in a browser, the FortiOS fnbamd daemon first validates the certificate supplied by the user. If the certificate check is successful, the information in the SAN field of the user certificate is used to find a matching user record on the AD LDAP.

To verify the configuration:

```
# diagnose debug app fnbamd -1
# diagnose debug enable
```

The output includes the following information.

- Validate the certificate:

```
...
    __check_crl-***CERTIFICATE IS GOOD***
[567] fnbamd_cert_verify-Issuer found: CA_Cert_2 (SSL_DPI opt 1)
[500] fnbamd_cert_verify-Following cert chain depth 1
[675] fnbamd_cert_check_group_list-checking group with name 'vpn-group'
[490] __check_add_peer-check 'peer-RFC822-name'
[366] peer_subject_cn_check-Cert subject 'C = CA, ST = BC, L = Burnaby, CN = test2'
[294] __RDN_match-Checking 'CN' val 'test2' -- match.
[404] peer_subject_cn_check-CN is good.
```

- Bind to LDAP and try to match the content of the SAN in the user certificate with the user record in the AD LDAP:

```
...
_cert_ldap_query-LDAP query, idx 0
[448] __cert_ldap_query-UPN = 'test2@fortinet-fsso.com'
[1717] fnbamd_ldap_init-search filter is: (&(mail=test2@fortinet-fsso.com)(!
(UserAccountControl:1.2.840.113556.1.4.803:=2)))
```

- Confirm the successful match:

```
...
     __cert_ldap_query_cb-LDAP ret=0, server='ad-ldap-peer-user', req_id=269178889
[388] __cert_ldap_query_cb-Matched peer 'peer-RFC822-name'
...
[1066] fnbamd_cert_auth_copy_cert_status-req_id=269178889
[1074] fnbamd_cert_auth_copy_cert_status-Matched peer user 'peer-RFC822-name'
[833] fnbamd_cert_check_matched_groups-checking group with name 'vpn-group'
[895] fnbamd_cert_check_matched_groups-matched
[1193] fnbamd_cert_auth_copy_cert_status-Cert st 290, req_id=269178889
[209] fnbamd_comm_send_result-Sending result 0 (nid 672) for req 269178889, len=2155
```

FSSO

FortiOS can provide single sign-on capabilities to Windows AD, Citrix, VMware Horizon, Novell eDirectory, Microsoft Exchange, and third-party access point users with the help of FortiAuthenticator or agent software installed on these networks. These entities send information about user logons to the FortiGate unit. With user information such as IP address and user group memberships from the network, FortiGate security policies can allow authenticated network access to users who belong to the appropriate user groups without requesting their credentials again.

Fortinet Single Sign-On (FSSO), through FortiAuthenticator or agents installed on the network, monitors user logons and passes that information to the FortiGate unit. When a user logs on at a workstation in a monitored domain, FSSO:

- Detects the logon event and records the workstation name, domain, and user,
- Resolves the workstation name to an IP address,
- Determines which user groups the user belongs to,
- Sends the user logon information, including IP address and groups list, to the FortiGate unit, and
- Creates one or more log entries on the FortiGate unit for this logon event as appropriate.

When the user tries to access network resources, the FortiGate unit selects the appropriate security policy for the destination. If the user belongs to one of the permitted user groups associated with that policy then the connection is allowed, otherwise the connection is denied.

FortiAuthenticator

One or more FortiAuthenticator devices can act as the central source for FSSO on the network. They can collect real-time user logon/logoff information from the widest variety of sources, distill that information into FSSO events, and distribute these FSSO events to one or a large number of FortiOS devices. On busy networks, it can relieve FortiOS devices from the burden of processing the heavy user activity from many network devices into a single authoritative source and supports redundancy options when high availability is required.

FortiAuthenticator can collect user events from:

- SSO Mobility Agent (SSOMA): Standalone agent installed on endpoints reporting real-time user logon/logoff events to the FortiAuthenticator. This is the only FSSO method that allows quick and reliable detection of endpoints going offline.
- FortiClient: Include an SSOMA sub-component that can be enabled for FSSO.
- Third-party network access devices: FortiAuthenticator is able to receive Syslog and/or RADIUS accounting messages from network devices to extract real-time user logon/logoff events.
- Domain Controllers (DC): In a Windows AD network, FortiAuthenticator can optionally obtain logon information by polling the AD domain controllers.
- Domain Controller and Terminal Server agents: Gather information from one or many agents on the network.
- Web-based portals: FortiAuthenticator offer a variety of friendly user portals that can be integrated within a network as a central point of authentication for the network users.

Agent-based FSSO

Several different FSSO agents can be used in an FSSO implementation:

- Domain Controller (DC) agent
- eDirectory agent
- Citrix/Terminal Server (TS) agent
- Collector Agent

Consult the latest FortiOS Release Notes for operating system compatibility information.

Domain Controller agent

The Domain Controller (DC) agent must be installed on every domain controller when you use DC Agent mode. The DC agents monitor user logon events and pass the information to the Collector agent, which stores the information and sends it to the FortiGate unit.

eDirectory agent

The eDirectory agent is installed on a Novell network to monitor user logons and send the required information to the FortiGate unit. It functions much like the Collector agent on a Windows AD domain controller. The agent can obtain information from the Novell eDirectory using either the Novell API or LDAP.

Terminal Server agent

The Terminal Server (TS) agent can be installed on a Citrix, VMware Horizon 7.4, or Windows Terminal Server to monitor user logons in real time. It functions much like the DC Agent on a Windows AD domain controller.

Collector agent

The Collector Agent (CA) is installed as a service on a server in the Windows AD network to monitor user logons and send the required information to the FortiGate unit. The Collector agent can collect information from a DC agent (Windows AD) and TS agent (Citrix or VMware Horizon Terminal Server).

In a Windows AD network, the Collector agent can optionally obtain logon information by polling the AD domain controllers. In this case, DC agents are not needed.

The CA is responsible for DNS lookups, group verification, workstation checks, and updating FortiGates on logon records. The FSSO CA sends Domain Local Security Group and Global Security Group information to FortiGate units. The CA communicates with the FortiGate over TCP port 8000 and it listens on UDP port 8002 for updates from the DC agents.

The FortiGate device can have up to five CAs configured for redundancy. If the first CA on the list is unreachable, the next is attempted, and so on down the list until one is contacted.

All DC agents must point to the correct CA port number and IP address on domains with multiple DCs.

A FortiAuthenticator device can act much like a CA, collecting Windows AD user logon information and sending it to the FortiGate device. It is particularly useful in large installations with several FortiGate units. For more information, see the FortiAuthenticator Administration Guide.

Agentless FSSO

For Windows AD networks, FortiGate devices can also provide SSO capability by directly polling Windows Security Event log entries on Windows DC for user log in information. This configuration does not require a CA or DC agent.

FortiGate configuration

To configure FSSO on a FortiGate, go to *Security Fabric > External Connectors*.

When creating a new connector, several options for connectors are available under Endpoint/Identity:

- Fortinet single sign-on agent on page 4679

  For most FSSO Agent-based deployments, this connector option will be used. Specify either Collector Agent or Local as User Group Source to collect user groups from the Collector Agent, or to match users to user groups from a LDAP server.
- Poll Active Directory server on page 4680

  This connection option directly polls Windows Security Event log entries on Windows DC for user log in information.
- RADIUS single sign-on agent on page 4688

  FortiGate can authenticate users who have authenticated on a remote RADIUS server by monitoring the RADIUS accounting records forwarded by the RADIUS server to the FortiGate.
- Exchange Server connector on page 4691

  FortiGate collects information about authenticated users from corporate Microsoft Exchange Servers.
- Symantec endpoint connector on page 4681

  This connector uses client IP information from Symantec Endpoint Protection Manager (SEPM) to assign dynamic IP addresses on FortiOS.

  Since FSSO is commonly associated with Agent-based FSSO and Agentless FSSO, this chapter will primarily focus on the first two Security Fabric External Connector options.

FSSO polling connector agent installation

This topic gives an example of configuring a local FSSO agent on the FortiGate. The agent actively polls Windows Security Event log entries on Windows Domain Controller (DC) for user login information. The FSSO user groups can then be used in a firewall policy.

This method does not require any additional software components, and all the configuration can be done on the FortiGate.

To configure a local FSSO agent on the FortiGate:

1. Configure an LDAP server on the FortiGate on page 3672
2. Configure a local FSSO polling connector on page 3673
3. Add the FSSO groups to a policy on page 3673

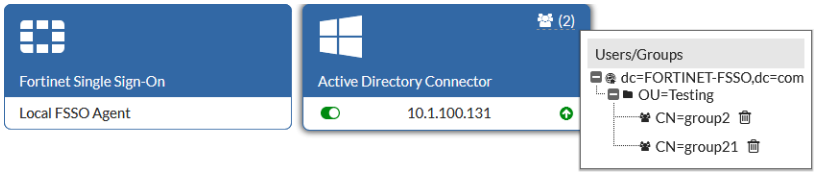
Configure an LDAP server on the FortiGate

Refer to Configuring an LDAP server on page 3509. The connection must be successful before configuring the FSSO polling connector.

Configure a local FSSO polling connector

To configure a local FSSO polling connector:

1. Go to *Security Fabric* > *External Connectors* and click *Create New*.
2. In the *Endpoint/Identity* section, select *Poll Active Directory Server*.
3. Fill in the required information.
4. For *LDAP Server*, select the server you just created.
5. Configure the group settings:
   a. For *Users/Groups*, click *Edit*. The structure of the LDAP tree is shown in the *Users/Groups* window.
   b. Click the *Groups* tab.
   c. Select the required groups, right-click on them, and select *Add Selected*. Multiple groups can be selected at one time by holding the CTRL or SHIFT keys. The groups list can be filtered or searched to limit the number of groups that are displayed.
   d. Click the *Selected* tab and verify that the required groups are listed. To remove a group, right-click and select *Remove Selected*.
   e. Click *OK* to save the group settings.
6. Click *OK* to save the connector settings.
7. Go back to *Security Fabric* > *External Connectors*.
8. There should be two new connectors:



   - The *Local FSSO Agent* is the backend process that is automatically created when the first FSSO polling connector is created.
   - The *Active Directory Connector* is the front end connector that can be configured by FortiGate administrators.

   To verify the configuration, hover the cursor over the top right corner of the connector; a popup window will show the currently selected groups. A successful connection is also shown by a green up arrow in the lower right corner of the connector.

   If you need to get log in information from multiple DCs, then you must configure other Active Directory connectors for each additional DC to be monitored.

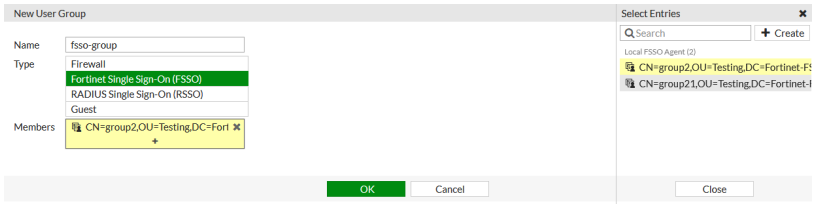
Add the FSSO groups to a policy

FSSO groups can be used in a policy by either adding them to the policy directly, or by adding them to a local user group and then adding the group to a policy.

To add the FSSO groups to a local user group:

1. Go to *User & Authentication* > *User Groups* and click *Create New*.
2. Enter a name for the group in the *Name* field.
3. Set the *Type* to *Fortinet Single Sign-On (FSSO)*.

4. In the *Members* field, click the *+* and add the FSSO groups.



5. Click *OK*.
6. Add the local FSSO group to a policy.

To add the FSSO groups directly to a firewall policy:

1. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
2. In the *Source* field, click the *+*. In the *Select Entries* pane, select the *User* tab.
3. Select the FSSO groups.
4. Configure the remaining settings as required.
5. Click *OK*.

Troubleshooting

If an authenticated AD user cannot access the internet or pass the firewall policy, verify the local FSSO user list:

```
# diagnose debug authd fsso list
----FSSO logons----
IP: 10.1.100.188 User: test2 Groups: CN=group2,OU=Testing,DC=Fortinet-FSSO,DC=COM Workstation:
MemberOf: CN=group2,OU=Testing,DC=Fortinet-FSSO,DC=COM
Total number of logons listed: 1, filtered: 0
----end of FSSO logons----
```

1. Check that the group in *MemberOf* is allowed by the policy.
2. If the expected AD user is not in list, but other users are, it means that either:
   - The FortiGate missed the log in event, which can happen if many users log in at the same time, or
   - The user's workstation is unable to connect to the DC, and is currently logged in with cached credentials, so there is no entry in the DC security event log.
3. If there are no users in the local FSSO user list:
   a. Ensure that the local FSSO agent is working correctly:

```
# diagnose debug enable
# diagnose debug authd fsso server-status

Server Name                        Connection Status    Version            Address
-----------                        -----------------    -------            -------
FGT_A (vdom1) # Local FSSO Agent   connected            FSAE server 1.1    127.0.0.1
```

   The connection status must be `connected`.

   b. Verify the Active Directory connection status:

```
# diagnose debug fsso-polling detail 1
AD Server Status (connected):
ID=1, name(10.1.100.131),ip=10.1.100.131,source(security),users(0)
port=auto username=Administrator
read log eof=1, latest logon timestamp: Fri Jul 26 10:36:20 2019

polling frequency: every 10 second(s) success(274), fail(0)
LDAP query: success(0), fail(0)
LDAP max group query period(seconds): 0
LDAP status: connected

Group Filter: CN=group2,OU=Testing,DC=Fortinet-
FSSO,DC=com+CN=group21,OU=Testing,DC=Fortinet-FSSO,DC=COM
```

If the polling frequency shows successes and failures, that indicates sporadic network problems or a very busy DC. If it indicates no successes or failures, then incorrect credentials could be the issue.

If the LDAP status is connected, then the FortiGate can access the configured LDAP server. This is required for AD group membership lookup of authenticated users because the Windows Security Event log does not include group membership information. The FortiGate sends an LDAP search for group membership of authenticated users to the configure LDAP server.

FortiGate adds authenticated users to the local FSSO user list only if the group membership is one of the groups in `Group Filter`.

4. If necessary, capture the output of the local FortiGate daemon that polls Windows Security Event logs:

```
# diagnose debug application fssod -1
```

This output contains a lot of detailed information which can be captured to a text file.

Limitations

- NTLM based authentication is not supported.
- If there are a large number of user log ins at the same time, the FSSO daemon may miss some. Consider using FSSO agent mode if this will be an issue. See Public and private SDN connectors on page 4597 for information.
- The FSSO daemon does not support all of the security log events that are supported by other FSSO scenarios. For example, only Kerberos log in events 4768 and 4769 are supported.

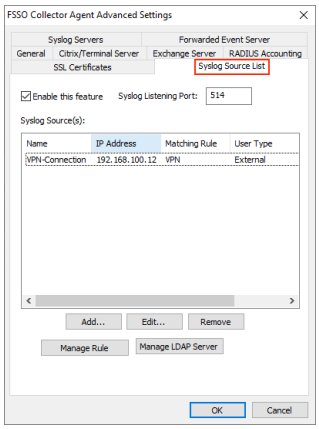
FSSO using Syslog as source

This example describes how to configure Fortinet Single Sign-On (FSSO) agent on Windows using syslog as the source and a custom syslog matching rule.

The FSSO collector agent must be build 0291 or later, and in advanced mode (see How to switch FSSO operation mode from Standard Mode to Advanced Mode).

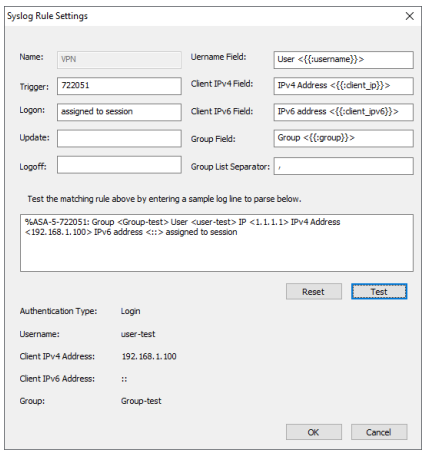
To configure the FSSO agent on Windows:

1. Open the FSSO agent on Windows.
2. Click *Advanced Settings*.
3. Go to the *Syslog Source List* tab.
4. Select *Enable this feature*.
5. Set *Syslog Listening Port*, or use the default port.



6. Click *Manage Rule*.
7. Create a new syslog rule:
   a. Click *Add*.
   b. Configure the rule:

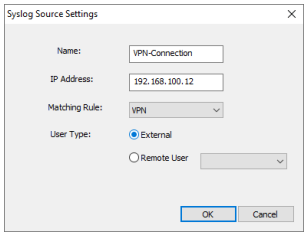
Trigger	*722051*
Logon	*assigned to session*
Username Field	*User <{{:username}}>*
Client IPv4 Field	*IPv4 Address <{{:client_ip}}>*
Client IPv6 Field	*IPv6 Address <{{:client_ipv6}}>*
Group Field	*Group <{{:group}}>*
Groups List Separator	,



c. To test the rule, enter a sample log line, then click *Test*.
d. Click *OK*.

8. Create a new syslog source:
   a. On the *Advanced Settings* window, click *Add*.
   b. Configure the source:

Name	*VPN-Connection*
IP Address	*192.168.100.12*
Matching Rule	*VPN*
User Type	*External*: Users are not defined on the CA and user groups come from the source. *Remote User*: Users are defined on a remote LDAP server and user groups are retrieved from the specified LDAP server. Any group from the syslog messages are ignored. See Connect to a remote LDAP server on page 3677.



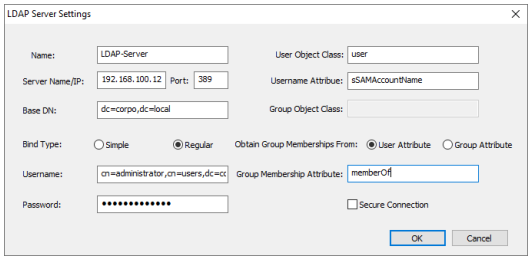
   c. Click *OK*.
9. Click *OK*.

Connect to a remote LDAP server

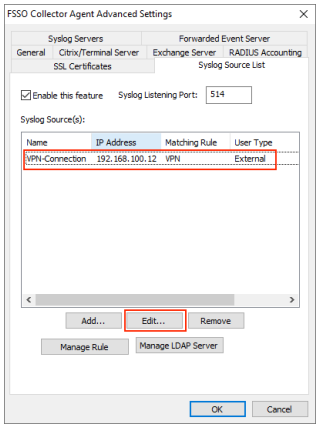
This section describes how to connect to a remote LDAP server to match the user identity from the syslog server with an LDAP server.

To connect to a remote LDAP server:

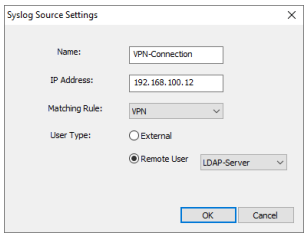
1. Open the FSSO agent on Windows.
2. Click *Advanced Settings*.
3. Go to the *Syslog Source List* tab.
4. Click *Manage LDAP Server*.
5. Click *Add* and configure the LDAP server settings:



6. Click *OK*.
7. Select the syslog source and click *Edit*.



8. Set *User Type* to *Remote User*, and select the LDAP server from the drop-down list.



9. Click *OK*.

Managing collector agent connection failures

Maintaining a stable connection to the collector agent is vital to the operations of FSSO. However, it is also important to plan for failures and mitigate against connection issues.

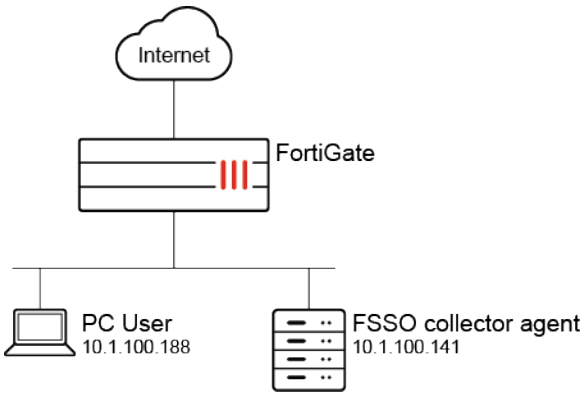
One option is to configure multiple collector agent servers. The primary server (first server in the configuration) will be preferred. If it is offline, then the second server will be polled and so on down the list of five servers. The FortiGate polls every 10 seconds to check connectivity. If the FortiGate fails over to the next server, there is no automatic fail back mechanism when a preferred server comes back online. You must disconnect the current server for the preferred server to become primary.

Another method to mitigate against connection issues is to configure the `logon-timeout` option. This option is used to manage how long authenticated FSSO users on the FortiGate will remain on the list of authenticated FSSO users when a network connection to the collector agent is lost.

```
config user fsso
    edit <name>
        set server <string>
        set password <string>
        set logon-timeout <integer>
    next
end
```

Option	Description
`logon-timeout <integer>`	Enter the interval to keep logons after the FSSO server is down, in minutes (1 - 2880, default = 5).

Example



In this example, the logon timeout is set for four minutes.

To configure the FSSO logon timeout:

1. Set the timeout value:

```
config user fsso
    edit "ad"
        set server "10.1.100.141"
        set password ********
        set logon-timeout 4
    next
end
```

2. Log on to a PC with a valid FSSO user account.
3. Enable real-time debugging and check for authd polling collector agent information. During this time, the connection to the collector agent is lost:

```
# diagnose debug enable
# diagnose debug application authd -1
# diagnose debug application fssod -1
021-06-10 16:20:41 authd_timer_run: 2 expired
2021-06-10 16:20:41 authd_epoll_work: timeout 39970
2021-06-10 16:20:46 fsae_io_ctx_process_msg[ad]: received heartbeat 100031
2021-06-10 16:20:46 authd_epoll_work: timeout 1690
2021-06-10 16:20:47 authd_timer_run: 1 expired
2021-06-10 16:20:47 authd_epoll_work: timeout 39990
2021-06-10 16:20:56 fsae_io_ctx_process_msg[ad]: received heartbeat 100032
2021-06-10 16:20:56 authd_epoll_work: timeout 31550
2021-06-10 16:21:00 _event_error[ad]: error occurred in epoll_in: Success
2021-06-10 16:21:00 disconnect_server_only[ad]: disconnecting
```

```
2021-06-10 16:21:00 authd_timer_run: 1 expired
2021-06-10 16:21:00 authd_epoll_work: timeout 9620
```

4. After about three minutes, check that the FSSO user is still in the list of authenticated users and can connect to the internet:

```
# diagnose firewall auth l
10.1.100.188, TEST1
        type: fsso, id: 0, duration: 229, idled: 229
        server: ad
        packets: in 0 out 0, bytes: in 0 out 0
        user_id: 16777219
        group_id: 3 33554433
        group_name: ad CN=GROUP1,OU=TESTING,DC=FORTINET-FSSO,DC=COM

----- 1 listed, 0 filtered ------
```

5. After four minutes, check the debugs again. Note that the FSSO users are cleared:

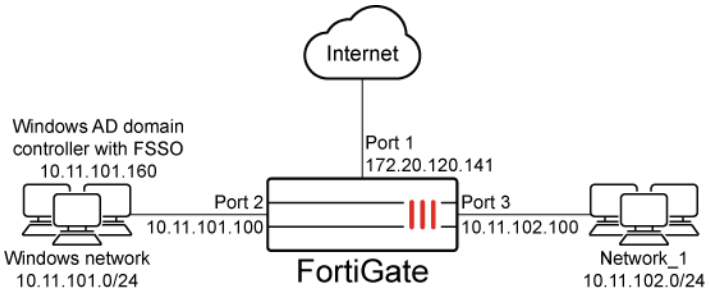
```
...
2021-06-10 16:24:57 authd_timer_run: 3 expired
2021-06-10 16:24:57 authd_epoll_work: timeout 60000
2021-06-10 16:24:59 [fsae_db_logoff:248]: vfid 0, ip 10.1.100.188, id(0), port_range_sz(0)
2021-06-10 16:24:59 [authd_fp_notify_logoff:444]: vfid 0, ip 10.1.100.188, id 0
2021-06-10 16:24:59 [authd_fp_on_user_logoff:412]: vfid 0, ip 10.1.100.188
2021-06-10 16:24:59 [authd_fp_on_user_logoff:412]: vfid 0, ip 10.1.100.188
2021-06-10 16:24:59 [authd_fp_on_user_logoff:412]: vfid 0, ip 10.1.100.188
2021-06-10 16:24:59 [authd_fpc_on_msg:545]: code 0, type 132, len 28 seq 0
2021-06-10 16:24:59 [authd_fp_on_user_logoff:412]: vfid 0, ip 10.1.100.188
2021-06-10 16:24:59 authd_epoll_work: timeout 21990
```

```
# diagnose firewall auth l

----- 0 listed, 0 filtered ------
```

After the connection to the collector agent is restored, all users remain in the list of authenticated users and are synchronized to the FortiGate. The users do not need to log in again for authentication.

Configuring FSSO firewall authentication

In this example, a Windows network is connected to the FortiGate on port 2, and another LAN, Network_1, is connected on port 3.



All Windows network users authenticate when they log on to their network. Engineering and Sales groups members can access the Internet without reentering their authentication credentials. The example assumes that you have already installed and configured FSSO on the domain controller.

LAN users who belong to the Internet_users group can access the Internet after entering their username and password. The example shows two users: User1, authenticated by a password stored in FortiOS; and User 2, authenticated on an external authentication server. Both users are local users since you create the user accounts in FortiOS.

1. Create a locally authenticated user account
2. Create a RADIUS-authenticated user account
3. Create an FSSO user group
4. Create a firewall user group
5. Define policy addresses
6. Create security policies

Creating a locally authenticated user account

User1 is authenticated by a password stored in FortiOS.

To create a locally authenticated user account in the GUI:

1. Go to *User & Authentication* > *User Definition* and click *Create New*.
2. Configure the following settings:

User Type	Local User
User Name	User1
Password	hardtoguess1@@1
User Account Status	Enabled

3. Click *Submit*.

To create a locally authenticated user account in the CLI:

```
config user local
    edit user1
        set type password
        set passwd hardtoguess1@@1
    next
end
```

Creating a RADIUS-authenticated user account

You must first configure FortiOS to access the external authentication server, then create the user account.

To create a RADIUS-authenticated user account in the GUI:

1. Go to *User & Authentication* > *RADIUS Servers* and click *Create New*.
2. Configure the following settings:

Name	OurRADIUSsrv
Authentication method	Default
Primary Server	
IP/Name	10.11.101.15
Secret	OurSecret

3. Click *OK*.
4. Go to *User & Authentication* > *User Definition* and click *Create New*.
5. Configure the following settings:

User Type	Remote RADIUS User
User Name	User2
RADIUS Server	OurRADIUSsrv
User Account Status	Enabled

6. Click *Submit*.

To create a RADIUS-authenticated user account in the CLI:

```
config user radius
    edit OurRADIUSsrv
        set server 10.11.102.15
        set secret OurSecret
        set auth-type auto
    next
end
config user local
    edit User2
        set name User2
        set type radius
        set radius-server OurRADIUSsrv
    next
end
```

Creating an FSSO user group

This example assumes that you have already set up FSSO on the Windows network and that it used advanced mode, meaning that it uses LDAP to access user group information. You must do the following:

- Configure LDAP access to the Windows AD global catalog
- Specify the collector agent that sends user log in information to FortiOS

- Select Windows user groups to monitor
- Select and add the Engineering and Sales groups to an FSSO user group

To create an FSSO user group in the GUI:

1. Configure LDAP for FSSO:
   a. Go to *User & Authentication > LDAP Servers* and click *Create New*.
   b. Configure the following settings:

Name	ADserver
Server Name / IP	10.11.101.160
Distinguished Name	dc=office,dc=example,dc=com
Bind Type	Regular
Username	cn=FSSO_Admin,cn=users,dc=office,dc=example,dc=com
Password	Enter a secure password.

   c. Leave other fields as-is. Click *OK*.
2. Specify the collector agent for FSSO;
   a. Go to *Security Fabric > External Connectors* and click *Create New*.
   b. Under *Endpoint/Identity*, select *FSSO Agent on Windows AD*.
   c. Configure the following settings:

Name	Enter the Windows AD server name. This name appears in the Windows AD server list when you create user groups. In this example, the name is WinGroups.
Server IP/Name	Enter the IP address or name of the server where the agent is installed. The maximum name length is 63 characters. In this example, the IP address is 10.11.101.160.
Password	Enter the password of the server where the agent is installed. You only need to enter a password for the collector agent if you configured the agent to require authenticated access. If the TCP port used for FSSO is not the default, 8000, you can run the `config user fsso` command to change the setting in the CLI.
Collector Agent AD access mode	Advanced
LDAP Server	Select the previously configured LDAP server. In this example, it is ADserver.
User/Groups/Organization Units	Select the users, groups, and OUs to monitor.

   d. Click *OK*.
3. Create the FSSO_Internet_users user group:
   a. Go to *User & Authentication > User Groups* and click *Create New*.
   b. Configure the following settings:

Name	FSSO_Internet_users
Type	Fortinet Single Sign-On (FSSO)
Members	Engineering, Sales

c. Click *OK*.

To create an FSSO user group in the CLI:

```
config user ldap
    edit "ADserver"
        set server "10.11.101.160"
        set dn "cn=users,dc=office,dc=example,dc=com"
        set type regular
        set username "cn=administrator,cn=users,dc=office,dc=example,dc=com"
        set password set_a_secure_password
    next
end
config user fsso
    edit "WinGroups"
        set ldap-server "ADserver"
        set password ********
        set server "10.11.101.160"
    next
end
config user group
    edit FSSO_Internet_users
        set group-type fsso-service
        set member CN=Engineering,cn=users,dc=office,dc=example,dc=com
CN=Sales,cn=users,dc=office,dc=example,dc=com
    next
end
```

Creating a firewall user group

This example shows a firewall user group with only two users. You can add additional members.

To create a firewall user group in the GUI:

1. Go to *User & Authentication* > *User Groups* and click *Create New*.
2. Configure the following settings:

Name	Internet_users
Type	Firewall
Members	User1, User2

3. Click *OK*.

To create a firewall user group in the CLI:

```
config user group
    edit Internet_users
        set group-type firewall
        set member User1 User2
    next
end
```

Defining policy addresses

To define policy addresses:

1. Go to *Policy & Objects* > *Addresses* and select *Address*.
2. Click *Create New*.
3. Configure the following settings:

Name	Internal_net
Type	Subnet
IP/Netmask	10.11.102.0/24
Interface	Port 3

4. Click *OK*.
5. Create another new address by repeating steps 2-4 using the following settings:

Name	Windows_net
Type	Subnet
IP/Netmask	10.11.101.0/24
Interface	Port 2

Creating security policies

You must create two security policies: one for the firewall group connecting through port 3, and one for the FSSO group connecting through port 2.

To create security policies using the GUI:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Click *Create New*.
3. Configure the following settings:

Incoming Interface	Port2
Source Address	Windows_net

Source User(s)	FSSO_Internet_users
Outgoing Interface	Port1
Destination Address	all
Schedule	always
Service	ALL
NAT	Enabled.
Security Profiles	You can enable security profiles as desired.

4. Click *OK*.
5. Create another new policy by repeating steps 2-4 using the following settings:

Incoming Interface	Port3
Source Address	Internal_net
Source User(s)	Internet_users
Outgoing Interface	Port1
Destination Address	all
Schedule	always
Service	ALL
NAT	Enabled.
Security Profiles	You can enable security profiles as desired.

6. Click *OK*.

To create security policies using the CLI:

```
config firewall policy
    edit 0
        set srcintf port2
        set dstintf port1
        set srcaddr Windows_net
        set dstaddr all
        set action accept
        set groups FSSO_Internet_users
        set schedule always
        set service ANY
        set nat enable
    next
    edit 1
        set srcintf port3
        set dstintf port1
        set srcaddr internal_net
        set dstaddr all
        set action accept
```

```
        set schedule always
        set groups Internet_users
        set service ANY
        set nat enable
    next
end
```

Include usernames in logs

Usernames can be included in logs, instead of just IP addresses. The benefits of doing this include:

- FortiOS monitors and FortiAnalyzer reports display usernames instead of IP addresses, allowing you to quickly determine who the information pertains to. Without the usernames, it is difficult to correlate the IP addresses with specific users.
- User activity can be correlated across multiple IP addresses.

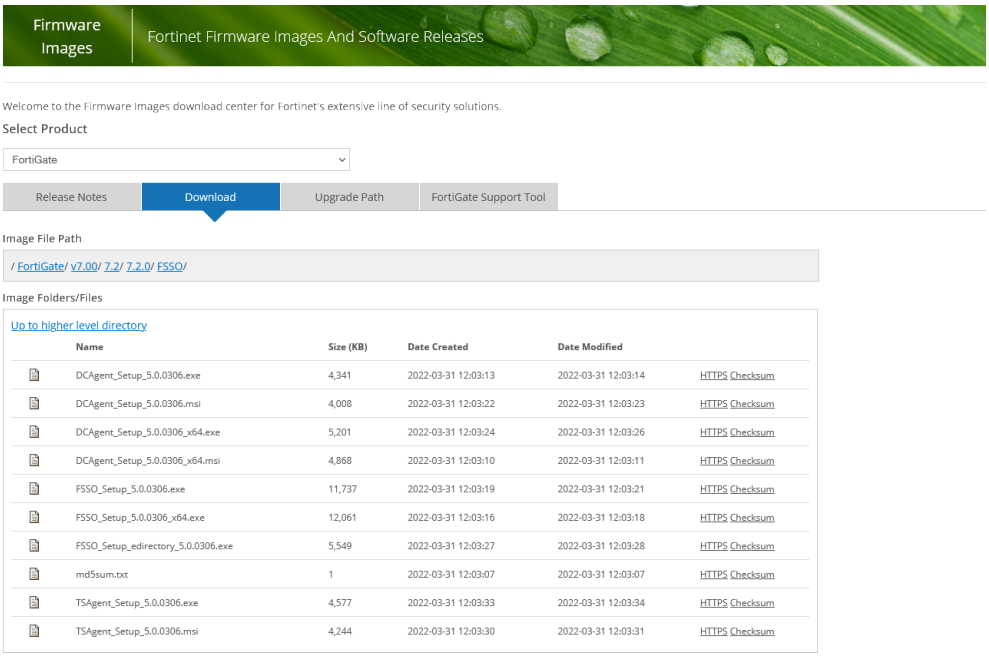
  For example, if DHCP is used a user might receive different IP addresses every day, making it difficult to track a specific user by specifying an IP address as the match criterion.

In this example, a collector agent (CA) is installed on a Windows machine to poll a domain controller (DC) agent (seeFSSO on page 3669 for more information). On the FortiGate, an external connector to the CA is configured to receives user groups from the DC agent. The received group or groups are used in a policy, and some examples of the usernames in logs, monitors, and reports are shown.

Install and configure FSSO Agent

To download the FSSO agent:

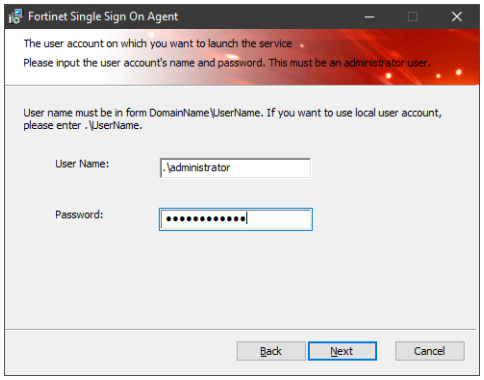
1. Sign in to your FortiCloud account.
2. Go to *Support* > *Firmware Download* and select the *Download* tab.
3. Browse to the appropriate directory for the version of the FSSO agent that you need to download.



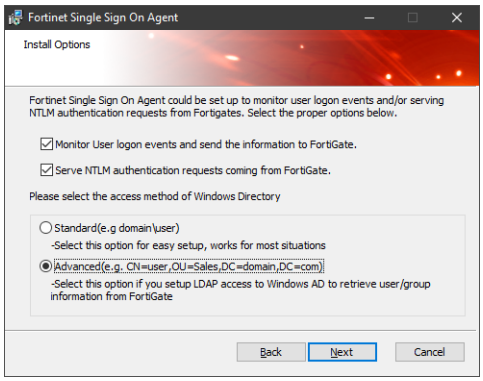
4. Click *HTTPS* to download the appropriate *FSSO_Setup* file.

To install the FSSO agent:

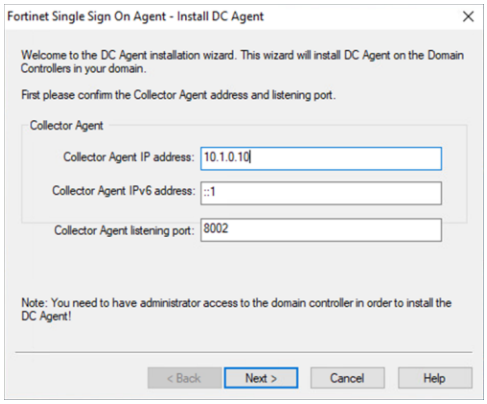
1. Run the *FSSO_Setup* file with administrator privileges.
2. Click *Next*, accept the terms of the license agreement, and click *Next* again.
3. Select the installation directory, or use the default location, then click *Next*.
4. Enter the User Name and Password, then click *Next*.



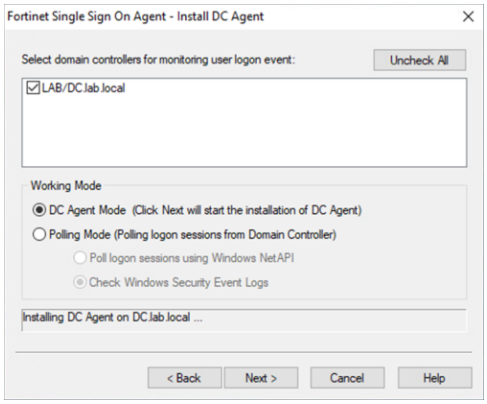
5. On the *Install Options*, select *Advanced*, then click *Next*.



6. Click *Install*.
7. After the FSSO Agent installs, run *Install DC Agent*.
8. Update the Collector Agent IP address and listening port as needed, then click *Next*.



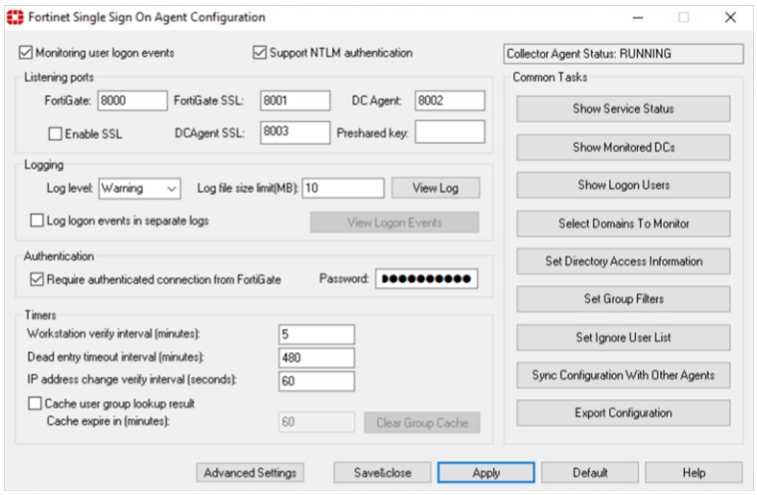
9. Select the domain, in this example *LAB:lab.local*, then click *Next*.
10. Set the *Working Mode* to *DC Agent Mode*, then click *Next* to install the agent.



11. After the DC agent mode installation finishes, Reboot the DC to complete the setup.

To configure the FSSO agent:

1. Open the FSSO agent.
2. Enable *Require authentication from FortiGate* and enter a password for FortiGate authentication.



3. Click *Set Group Filters*, and create a default group filter to limit the groups that are sent to the FortiGate.
4. Click *Save&close*.

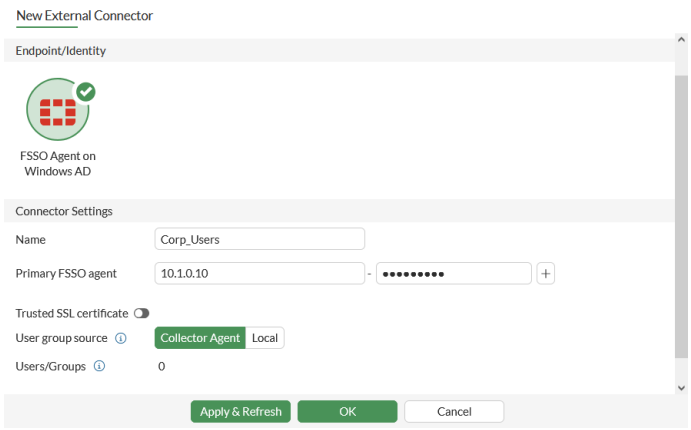
Configure the FortiGate

Create an external connector to the FSSO agent to receive the AD user groups. Add the user group or groups as the source in a firewall policy to include usernames in traffic logs. Enable security profiles, such as web filter or antivirus, in the policy to include the usernames in UTM logs.

Event logs include usernames when the log is created for a user action or interaction, such as logging in.

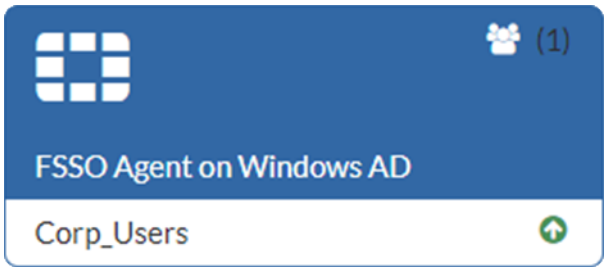
To create an external connector:

1. On the FortiGate, go to *Security Fabric > External Connectors*.
2. Click *Create New* and select *FSSO Agent on Windows AD*.
3. Set the *Primary FSSO agent* to the previously configured *Collector Agent IP address* and authentication password.



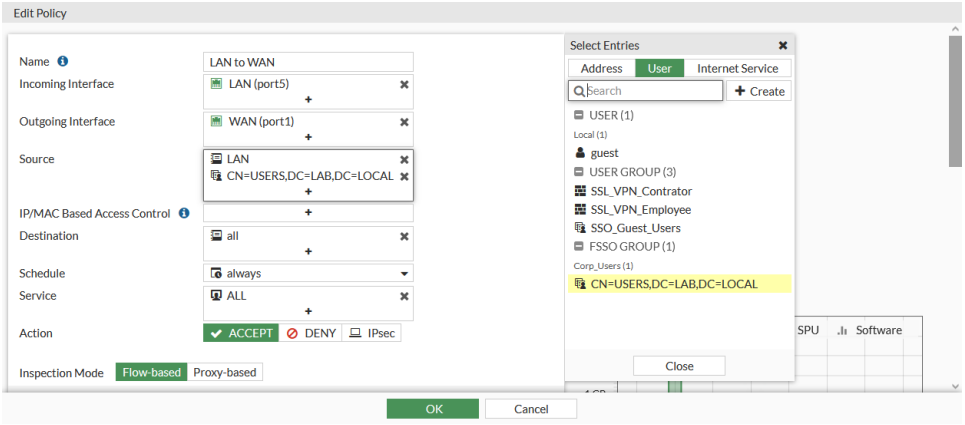
4. Click *OK*

The connector shows a green arrow when the connection is established, and a number in the top right indicating the number of AD groups received from the DC agent. Edit the connector to view the user groups.



To configure a policy with an imported user group and web filter in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Edit an existing policy, or create a new one. See Firewall policy on page 1842 for information.
3. Add the FSSO groups or groups as sources:
   a. Click in the *Source* field.
   b. Select the *User* tab.
   c. Select the group or groups.



   d. Click *Close*.
4. Under *Security Profiles*, enable *Web Filter* and select a profile that monitors or blocks traffic, such as the *monitor-all* profile. See Web filter on page 2311 for information.
5. Click *OK*.

To configure a policy with an imported user group and web filter in the CLI:

```
config firewall policy
    edit 0
        set name "LAN to WAN"
        set srcintf "port5"
        set dstintf "port1"
        set action accept
        set srcaddr "LAN"
        set dstaddr "all"
        set schedule "always"
```

```
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "certificate-inspection"
        set webfilter-profile "monitor-all"
        set logtraffic all
        set nat enable
        set fsso-groups "CN=USERS,DC=LAB,DC=LOCAL"
    next
end
```

Log, monitor, and report examples

For more information about logs, see the FortiOS Log Message Reference.

Traffic logs:

Without a web filter profile applied:

```
date=2022-05-24 time=13:50:47 eventtime=1653425447661722283 tz="-0700" logid="0000000015"
type="traffic" subtype="forward" level="notice" vd="root" srcip=10.1.0.11 identifier=0
srcintf="port5" srcintfrole="lan" dstip=192.168.2.200 dstintf="port1" dstintfrole="wan"
srccountry="Reserved" dstcountry="Reserved" sessionid=708558 proto=1 action="start" policyid=15
policytype="policy" poluuid="5bf426fe-794b-51ec-dedf-4318a843c5b5" policyname="LAN to WAN"
user="USER2" authserver="Corp_Users" service="PING" trandisp="snat" transip=192.168.2.99
transport=0 duration=0 sentbyte=0 rcvdbyte=0 sentpkt=0 rcvdpkt=0 appcat="unscanned"
```

With a web filter profile applied:

```
date=2022-05-25 time=12:16:54 id=7101754911016091650 itime=2022-05-25 12:16:07 euid=1039 epid=1037
dsteuid=3 dstepid=101 type=traffic subtype=forward level=notice action=close utmaction=allow
policyid=15 sessionid=683 srcip=10.1.0.11 dstip=104.26.1.188 transip=192.168.2.99 srcport=64494
dstport=443 transport=64494 trandisp=snat duration=7 proto=6 sentbyte=1855 rcvdbyte=18631
sentpkt=16 rcvdpkt=21 logid=0000000013 user=USER2 group=CN=USERS,DC=LAB,DC=LOCAL service=HTTPS
app=HTTPS appcat=unscanned srcintfrole=lan dstintfrole=wan srcserver=0 policytype=policy
eventtime=1653506215490475553 countweb=1 poluuid=5bf426fe-794b-51ec-dedf-4318a843c5b5
srcmac=00:0c:29:5e:f5:25 mastersrcmac=00:0c:29:5e:f5:25 srchwvendor=VMware
srchwversion=Workstation pro srcfamily=Virtual Machine srcswversion=10 devtype=Server
osname=Windows srccountry=Reserved dstcountry=United States srcintf=port5 dstintf=port1
authserver=Corp_Users policyname=LAN to WAN hostname=www.yellow.com catdesc=Reference tz=-0700
devid=FGVM01TM22000459 vd=root dtime=2022-05-25 12:16:54 itime_t=1653506167
```

UTM log:

```
date=2022-05-25 time=12:16:46 id=7101754876656353280 itime=2022-05-25 12:15:59 euid=1039 epid=1037
dsteuid=3 dstepid=101 type=utm subtype=webfilter level=notice action=passthrough sessionid=683
policyid=15 srcip=10.1.0.11 dstip=104.26.1.188 srcport=64494 dstport=443 proto=6 cat=39
logid=0317013312 service=HTTPS user=USER2 group=CN=USERS,DC=LAB,DC=LOCAL
eventtime=1653506207694977460 sentbyte=548 rcvdbyte=0 srcintfrole=lan dstintfrole=wan
direction=outgoing method=domain reqtype=direct url=https://www.yellow.com/
```

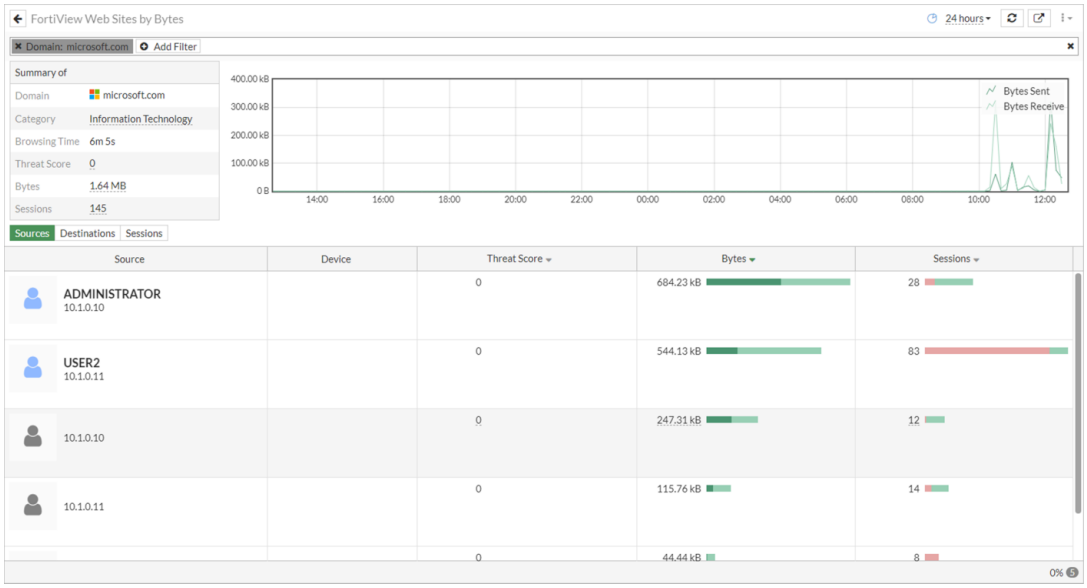
```
hostname=www.yellow.com profile=default catdesc=Reference eventtype=ftgd_allow srcintf=port5
dstintf=port1 authserver=Corp_Users msg=URL belongs to an allowed category in policy tz=-0700
srcuuid=41cad638-794b-51ec-a8c9-8128712cb495 dstuuid=e1067f08-8e38-51eb-4b07-64f219140388
policytype=policy srccountry=Reserved dstcountry=United States poluuid=5bf426fe-794b-51ec-dedf-
4318a843c5b5 devid=FGVM01TM22000459 vd=root dtime=2022-05-25 12:16:46 itime_t=1653506159
```

Event log:

```
date=2019-05-13 time=11:20:54 logid="0100032001" type="event" subtype="system" level="information"
vd="vdom1" eventtime=1557771654587081441 logdesc="Admin login successful" sn="1557771654"
user="admin" ui="ssh(172.16.1.1)" method="ssh" srcip=172.16.200.254 dstip=172.16.200.2
action="login" status="success" reason="none" profile="super_admin" msg="Administrator admin
logged in successfully from ssh(172.16.200.254)"
```

FortiOS monitors:

The *FortiView Web Sites by Bytes* monitor shows a list of visited websites. Double click a specific domain (or manually create a filter), such as *microsoft.com*, to see a breakdown of the usernames and IP addresses that visited that domain. See Monitors and widgets on page 184 for more information.



FortiAnalyzer reports:

The *User Detailed Browsing Log* report require a username or IP address to run. If a username is used, the report includes logs related to that user regardless of their IP address. For example, the following report show two source IP addresses:

FORTINET

User: USER2
Source IP: 10.1.0.11, 10.1.0.222
Hostname (MAC): 00:0c:29:5e:f5:25, LAB, PC1
Source Interface: port5
Devices: FGVM01TM22000459

Copy of Detailed Web Browsing Log

#	Timestamp	Category	Website	Action	Bytes
1	2022-05-25 13:05:50	Information Technology	edgedl.me.gvt1.com	allow	10.82 MB
2	2022-05-25 12:39:02	Sports	bdata-producedclips.mlb.com	allow	6.44 MB
3	2022-05-25 12:39:00	Streaming Media and Download	rr8---sn-uxa0n-t8gl.googlevideo.com	allow	3.22 MB
4	2022-05-25 13:02:07	Personal Vehicles	www.dodge.com	allow	3.10 MB
5	2022-05-25 12:16:54	Reference	www.yellow.com	allow	2.97 MB
6	2022-05-25 12:39:02	Sports	builds.mlbstatic.com	allow	2.92 MB
7	2022-05-25 13:01:47	Business	www.ford.com	allow	1.44 MB
8	2022-05-25 12:39:02	Sports	www.mlbstatic.com	allow	1.20 MB
9	2022-05-25 13:05:44	Information Technology	www.googletagmanager.com	allow	1.10 MB
10	2022-05-25 12:39:02	Search Engines and Portals	ampcid.google.com	allow	914.17 KB
11	2022-05-25 12:39:01	Sports	ca.global.nba.com	allow	802.94 KB
12	2022-05-25 12:39:02	Information	imasdk.googleapis.com	allow	715.34 KB

The *Web Usage* report includes all usernames and IP addresses that match the specified conditions, like most visited categories.

Web Usage Report

Web Usage Summary
- Requests Summary
- Browsing Time Summary
- Bandwidth Summary

Web Activity
- Top 20 Most Active Users
- Top 20 Most Visited Categories
- Top 50 Most Visited Sites

Web Browsing
- Top 10 Online Users
- Top 10 Categories
- Top 50 Sites By Browsing Time

Internet Bandwidth Usage
- Top 20 Bandwidth Users
- Top 20 Categories By Bandwidth
- Top 50 Sites (and Category) by Bandwidth

Most Blocked
- Top 20 Most Blocked Users
- Top 20 Most Blocked

Top 20 Most Active Users

#	User (or IP)	Hostname	Requests
1	10.1.0.11	PC1	513
2	USER2	PC1	506
3	USER2	10.1.0.222	181
4	ADMINISTRATOR	DC	38
5	10.1.0.10	DC	15

Top 20 Most Visited Categories

#	Category	Requests
1	Information Technology	470
2	Advertising	216
3	Business	140
4	Search Engines and Portals	77
5	Web Analytics	50
6	Sports	45
7	Content Servers	36
8	Personal Vehicles	24
9	Meaningless Content	19
10	Streaming Media and Download	17
11	Social Networking	12
12	Information and Computer Security	8
13	Shopping	7
14	Instant Messaging	6
15	Internet Radio and TV	5
16	Reference	3
17	Newsgroups and Message Boards	2
18	Web-based Applications	2
19	Games	1
20	File Sharing and Storage	1

Top 50 Most Visited Sites

#	Website	Category	Requests
1	edgedl.me.gvt1.com	Information Technology	58
2	v10.events.data.microsoft.com	Information Technology	24
3	settings-win.data.microsoft.com	Information Technology	20
4	cms.nhl.bamgrid.com	Streaming Media and Download	12
5	websocket.dg.toyota.com	Personal Vehicles	11

See Reports in the FortiAnalyzer Administration guide for more information.

Wireless configuration

See the FortiWiFi and FortiAP Configuration Guide.

Switch Controller

Use the Switch Controller function, also known as FortiLink, to remotely manage FortiSwitch units. In the commonly-used layer 2 scenario, the FortiGate that is acting as a switch controller is connected to distribution FortiSwitch units. The distribution FortiSwitch units are in the top tier of stacks of FortiSwitch units and connected downwards with Convergent or Access layer FortiSwitch units. To leverage CAPWAP and the Fortinet proprietary FortiLink protocol, set up data and control planes between the FortiGate and FortiSwitch units.

FortiLink allows administrators to create and manage different VLANs, and apply the full-fledged security functions of FortiOS to them, such as 802.1X authentication and firewall policies. Most of the security control capabilities on the FortiGate are extended to the edge of the entire network, combining FortiGate, FortiSwitch, and FortiAP devices, and providing secure, seamless, and unified access control to users.

FortiLink is not supported if there is a 3rd party switch in between the FortiGate and FortiSwitch in a layer 2 network. However, FortiLink is supported over a layer 3 network. See FortiLink mode over a layer-3 network in the FortiLink Guide.

See FortiSwitch devices managed by FortiOS.

System

This topic contains information about FortiGate administration and system configuration that you can do after installing the FortiGate in your network.

Basic system settings

Administrators

By default, FortiGate has an administrator account with the username *admin* and no password. See Administrators on page 3700 for more information.

Administrator profiles

An administrator profile defines what the administrator can see and do on the FortiGate. See Administrator profiles on page 3736 for more information.

Password policy

Starting in FortiOS 7.6.5, the password policy is enabled and enforced. See Password policy on page 3721 for more information.

Interfaces

Physical and virtual interface allow traffic to flow between internal networks, and between the internet and internal networks. See Interfaces on page 222 for more information.

Advanced system settings

SNMP

The simple network management protocol (SNMP) allows you to monitor hardware on your network. See SNMP on page 3877 for more information.

DHCP server

You can configure one or more DHCP servers on any FortiGate interface. See DHCP servers and relays on page 550 for more information.

VDOM

You can use virtual domains (VDOMs) to divide a FortiGate into multiple virtual devices that function independently. See Virtual Domains on page 3835 for more information.

High availability

You can configure multiple FortiGate devices, including private and public cloud VMs, in HA mode. See High Availability on page 4070 for more information.

Certificates

You can manage certificates on the FortiGate. See Certificates on page 3953 for more information.

Operating modes

A FortiGate or VDOM (in multi-vdom mode) can operate in either NAT/route mode or transparent mode.

NAT/route mode

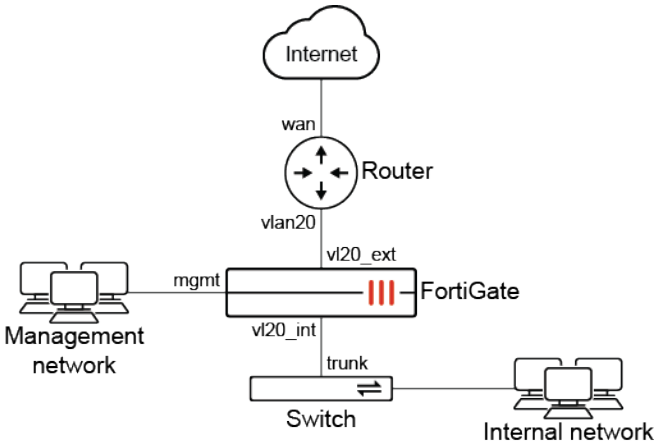
The FortiGate or VDOM is installed as a gateway or router between multiple networks, such as a private network and the internet. One function of NAT/route mode is to allow the FortiGate to hide the IP addresses on the private network using NAT. NAT/route mode can also be used to connect to multiple ISPs in an SD-WAN setup, and to route traffic between different networks. .

By default, new VDOMs are set to NAT/route operation mode.

Transparent mode

The FortiGate or VDOM operates in layer 2 to forward traffic between network devices such as routers, firewalls, and switches. For example. it can be installed inline between a router and a switch to perform security scanning without changing the network topology or modifying the IP addresses. When you add a FortiGate that is in transparent mode to a network, it only needs to be provided with a management IP address in order to access the device. It is recommended that a dedicated interface is used to connect to the management network in transparent mode.

The following topology is an example of a transparent mode FortiGate inserted inline between a router and a switch:



Using transparent mode VDOMs is recommended when multiple VLANs pass through the FortiGate. Otherwise, they must be separated into different forwarding domains within the same VDOM.

Changing modes

The following is a sample configuration for changing from NAT/route operation mode to transparent operation mode in the CLI:

```
config system settings
    set opmode transparent
    set manageip <IP_address>
    set gateway <gateway_address>
end
```

The gateway setting is optional. However, once the operation mode is changed from NAT/route to transparent, the gateway configuration is found under the static router settings:

```
config router static
    edit <seq-num>
        set gateway <IP_address>
    next
end
```

The following is a sample configuration for changing from transparent operation to NAT/route operation mode in the CLI:

```
config system settings
    set opmode nat
    set ip <IP_address>
    set device <interface>
    set gateway <gateway_address>
end
```

The IP and device settings are mandatory. Once the operation mode is changed from transparent to NAT/route, the IP address configuration is found under the corresponding interface settings:

```
config system interface
    edit <interface>
        set ip <IP_address>
    next
end
```

The gateway setting is optional. However, once the operation mode is changed, the gateway configuration is found under the static router settings:

```
config router static
    edit <seq-num>
        set gateway <IP_address>
        device <interface>
    next
end
```

Administrators

By default, FortiGate has an administrator account with the username *admin* and no password. To prevent unauthorized access to the FortiGate, this account must be protected with a password. Additional administrators can be added for various functions, each with a unique username, password, and set of access privileges.

The following topics provide information about administrators:

- Local authentication on page 3701
- Remote authentication for administrators on page 3703
- Administrator account options on page 3706
- REST API administrator on page 3712
- SSO administrators on page 3714
- FortiCloud SSO on page 3715
- Allowing the FortiGate to override FortiCloud SSO administrator user permissions on page 3716
- Password policy on page 3721
- Public key SSH access on page 3725
- Separating the SSHD host key from the administration server certificate on page 3728
- Restricting SSH and Telnet jump host capabilities on page 3729
- Restricting local administrator logins through the console on page 3730
- Remote administrators with TACACS+ VSA attributes on page 3732

Local authentication

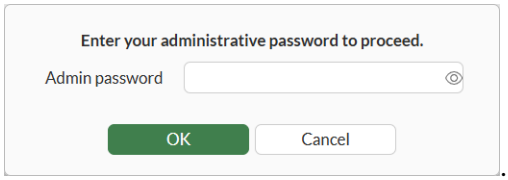
By default, FortiGate has one super admin named `admin`. You can create more administrator accounts with different privileges.

To create an administrator account in the GUI:

1. Go to *System* > *Administrators*.
2. Select *Create New* > *Administrator*.
3. Specify the *Username*.

   - Usernames can include lower and upper case letters (a-z, A-Z), numbers (0-9), underscores (_), and dashes (-)
   - Usernames cannot start with a dash (-)
   - Usernames can end with dollar symbol ($)
   - Usernames must not use the following characters: `< > ( ) # " '`. Using these characters in an administrator username might have a cross site scripting (XSS) vulnerability.

4. Set *Type* to *Local User*.
5. Set the password.
6. Set *Administrator profile* to a profile.
7. Set other fields as needed.
8. Click *OK*. The *Enter your administrative password to proceed* dialog box displays.



9. Enter your password and click *OK*.

To create an administrator account in the CLI:

```
config system admin
    edit <admin_name>
        set accprofile <profile_name>
        set vdom <vdom_name>
        set password <password for this admin>
        Please enter current administrator password: <your password>
    next
end
```

Administrative authentication and session monitoring

A `http_authd` daemon performs all administrative authentication, enhancing the efficiency and centralization of authentication processes.

The `diagnose http_authd` command is available to monitor session entries, providing improved oversight and management of authentication activities.

Command	Description
`http_authd session list`	List all HTTP authentication database entries.
`http_authd saml-dump clean`	Run a manual SAML dump clean up.

The following example demonstrates reviewing HTTP authentication session entries:

```
# diagnose  http_authd  session  list
http_authd session list
[
  {
    "login_context":{
      "admin_name":"Local_Process_Access",
      "login_name":"Local_Process_Access",
      "admin_profile":"LOCAL_PROCESS_PROFILE",
      "gui_custom_prof":{
        "name":"LOCAL_PROCESS_PROFILE",
        "global":true,
        "admintimeout_override":0,
        "cli_diagnose":true,
        "cli_get":false,
        "cli_show":false,
        "cli_exec":false,
        "exec_ssh":false,
        "exec_telnet":false,
        "accgrp":{
          "secfabgrp":"read",
          "ftviewgrp":"read",
          "netgrp":"read",
          "sysgrp":"read",
          "fwgrp":"read",
          "utmgrp":"read",
          "vpngrp":"read",
          "authgrp":"read",
          "wifi":"read",
          "loggrp":"read"
        }
      },
      "userfrom":"Node.js",
      "login_method":"https",
      "is_remote_auth":"false",
      "radius_vdom_override_sz":0
    },
    "request_source":"local",
    "remote_vdom":"root",
    "source_addr":"",
    "destination_addr":"",
    "auth_method":"Fabric trusted access",
    "login_type":"none",
```

```
    "timeout":1750981662,
    "login_time":1750981362
  },
....
```

Remote authentication for administrators

Administrators can use remote authentication, such as LDAP, RADIUS, and TACACS+ to connect to the FortiGate.

Configuring remote authentication with an LDAP server is shown. For more information about configuring LDAP, see Configuring an LDAP server on page 3509.

For information about configuring RADIUS or TACACS+ servers, see Configuring a RADIUS server on page 3528 and TACACS+ servers on page 3620. To use a RADIUS or TACACS+ server for remote authentication, configure the server, and then add it to the user group instead of the LDAP server.

Local logins can also be restricted when remote authentication servers are available, see Restricting logins from local administrator accounts when remote servers are available on page 3706.

Configuring remote authentication for administrators using LDAP includes the following steps:

1. Configuring the LDAP server on page 3703
2. Adding the LDAP server to a user group on page 3704
3. Configuring the administrator account on page 3704

Configuring the LDAP server

To configure the LDAP server in the GUI:

1. Go to *User & Authentication* > *LDAP Servers* and click *Create New*.
2. Enter the server *Name* and *Server IP/Name*.
3. Enter the *Common Name Identifier* and *Distinguished Name*.
4. Set the *Bind Type* to *Regular* and enter the *Username* and *Password*.
5. Click *OK*.

To configure the LDAP server in the CLI:

```
config user ldap
    edit <name>
        set server <server_ip>
        set cnid "cn"
        set dn "dc=XYZ,dc=fortinet,dc=COM"
        set type regular
        set username "cn=Administrator,dc=XYA, dc=COM"
        set password <password>
    next
end
```

Adding the LDAP server to a user group

After configuring the LDAP server, create a user group that includes that LDAP server.

To create a user group in the GUI:

1. Go to *User & Authentication* > *User Groups* and click *Create New*.
2. Enter a *Name* for the group.
3. In the *Remote groups* section, select *Create New*.
4. Select the *Remote Server* from the dropdown list.
5. Click *OK*.

To create a user group in the CLI:

```
config user group
    edit <name>
        set member <ldap_server_name>
    next
end
```

Configuring the administrator account

After configuring the LDAP server and adding it to a user group, create a new administrator. For this administrator, instead of entering a password, use the new user group for authentication.

A remote authentication server can allow authentication of either a single user or any user from a specified group.

> ⚠ When using remote authentication, FortiOS checks for an exact match admin account first and then falls back to wildcard accounts, with the global wildcard always taking precedence. For this reason, you should configure either one global wildcard administrator or one wildcard administrator per VDOM, but never both. Creating multiple wildcard accounts within the same VDOM or multiple global wildcard accounts can lead to unintended authentication behavior.

Public key infrastructure (PKI) administrator authentication requires a PKI user instead of a remote server. For information about creating a PKI user, see Configuring a PKI user on page 3662.

To create an administrator to match a single user in the GUI:

1. Go to *System* > *Administrators* and click *Create New* > *Administrator*.
2. Specify the *Username*.
   This username is used when the administrator logs in, and is what FortiOS sends to the remote authentication server for authorization.
3. Set *Type* to *Match a user on a remote server group*.
4. In *Remote User Group*, select the user group that you created.
5. Select an *Administrator Profile*.

6. Enter a *Backup Password*, to be used if the remote authentication server is unreachable.
7. Click *OK*.

To create an administrator match a single user in the CLI:

```
config system admin
    edit <name>
        set remote-auth enable
        set accprofile super_admin
        set remote-group <ldap_group_name>
        set password **********
    next
end
```

To create an administrator to match all users in a remote server group in the GUI:

1. Go to *System > Administrators* and click *Create New > Administrator*.
2. Specify the *Username*.
   This username is only used to identify this administrator group. Administrators can log in with any username in the remote user group.
3. Set *Type* to *Match all users in a remote server group*.
4. In *Remote User Group*, select the user group that you created.
5. Select an *Administrator Profile*.
6. Click *OK*.

To create an administrator to match all users in a remote server group in the CLI:

```
config system admin
    edit <name>
        set remote-auth enable
        set accprofile super_admin
        set wildcard enable
        set remote-group <ldap_group_name>
    next
end
```

To create an administrator that uses a PKI group in the GUI:

1. Go to *System > Administrators* and click *Create New > Administrator*.
2. Specify the *Username*.
3. Set *Type* to *Use public key infrastructure (PKI) group*.
4. In *Remote User Group*, select the user group that you created.
5. Select an *Administrator Profile*.
6. Click *OK*.

To create an administrator that uses a PKI group in the CLI:

```
config system admin
    edit <name>
        set remote-auth enable
        set accprofile super_admin
        set peer-group <pki_group_name>
    next
end
```

Restricting logins from local administrator accounts when remote servers are available

Logins from local administrator accounts can be restricted when remote servers are available. When enabled, FortiOS will check if all of the remote servers used by administrators are down before allowing a local administrator to log in. This option is applied globally, and is disabled by default.

To restrict local administrator authentication when a remote authentication server available:

```
config system global
    set admin-restrict-local enable
end
```

Administrator account options

Options to further define the access and abilities of an administrator account include:

- Multi-factor authentication on page 3706
- Restricting logins to trusted hosts on page 3708
- Restricting administrators to guest account provisioning on page 3709
- Global and VDOM administrators on page 3709
- Using FortiAuthenticator and FortiToken for 2FA on page 3709

Multi-factor authentication

Multi-factor authentication (MFA) requires authenticating administrators to supply more than one factor to identify themselves in addition to their password, such as a FortiToken.

> Before enabling MFA, it is recommended that you create second administrator account that is configured to guarantee administrator access to the FortiGate if you are unable to authenticate on the main account for any reason.

Multi-factor authentication options include:

- FortiToken
- FortiIdentity Cloud
- Email
- SMS

FortiToken

To associate a FortiToken to an administrator account using the GUI:

1. Ensure that you have successfully added your FortiToken serial number to FortiOS and that its status is *Available*.
2. Go to *System* > *Administrators*. Edit the admin account. This example assumes that the account is fully configured except forMFA.
3. Enable *Two-factor Authentication* and for *Authentication Type*, select *FortiToken*.
4. From the *Token* dropdown list, select the FortiToken serial number.
5. In the *Email Address* field, enter the administrator's email address.
6. Click *OK*.

For a mobile token, click *Send Activation Code* to send the activation code to the configured email address. The admin uses this code to activate their mobile token. You must have configured an email service in *System* > *Settings* > *General* to send the activation code.

To associate a FortiToken to an administrator account using the CLI:

```
config system admin
    edit <username>
        set password "myPassword"
        set two-factor fortitoken
        set fortitoken <serial_number>
        set email-to "username@example.com"
    next
end
```

The `fortitoken` keyword is not visible until you select `fortitoken` for the `two-factor` option.

Before you can use a new FortiToken, you may need to synchronize it due to clock drift.

FortiIdentity Cloud

FortiIdentity Cloud (formerly FortiToken Cloud) is an Identity and Access Management as a Service (IDaaS) cloud service provided by Fortinet. It enables FortiGate and FortiAuthenticator customers to add MFA for their users using Mobile or Hard tokens.

For more information, see Getting started—FGT-FTC users in the FortiIdentity Cloud Administration Guide.

Email

Enter an email address to send an MFA code to that address.

SMS

Enable *SMS* then select the *Country Dial Code* and enter the *Phone Number* (`sms-phone` in the CLI) to send an MFA code to.

SMS messages can also be sent to the FortiGuard SMS server or a custom server.

```
config system admin
    edit "admin"
        ...
        set sms-server {fortiguard | custom}
        set sms-server-custom <string>
        ...
    next
end
```

Restricting logins to trusted hosts

Administrator accounts can be configured to only be accessible to a user using a trusted host. You can set a specific IP address for the trusted host, or use a subnet. Up to ten trusted hosts can be specified for an administrator.

When trusted hosts are defined for all of the administrators on the FortiGate, the administrative access on each interface will be restricted to the trusted hosts that are defined for the administrator, except for ping. If ping is enabled on an interface, it works regardless of the trusted hosts.

Configuring a trusted host for SSO administrator accounts

Trusted Hosts can only be configured for local and REST API administrator accounts. It is not possible to directly configure trusted hosts on SAML-based SSO administrator accounts, but an SSO administrator account must be able to access the FortiGate GUI in order to login.

A scenario could arise where local administrators all have trusted hosts defined for a particular management subnet and an SSO administrator needs to log in from a different subnet but is unable to define a trusted host.

Instead, create another local administrator account that has trusted hosts configured but no actual administrative access to anything (that is, a dummy account). The general steps to do this are:

1. Create an administrator profile in *System > Admin Profile* called *no-access* that has all of its permissions set to *None*.
2. Create a dummy local administrator account in *System > Administrator* and assign it the *no-access* administrator profile. Give the administrator a long, random password to secure the account, even though this account does not have access to anything.
3. Configure trusted hosts (enable *Restrict login to trusted hosts*) and add the source subnets/addresses that the SSO administrator will access the FortiGate from.
4. Save the configuration.

Restricting administrators to guest account provisioning

To simplify guest account creation, an administrator account can be created exclusively for guest user management. This allows new accounts to be created without requiring full administrative access to FortiOS.

When enabling this option, a guest group must be specified for the administrator to provision new accounts to. See Configuring guest user groups on page 3493 for information about creating such a group.

Global and VDOM administrators

When a FortiGate is in multi-VDOM mode, it can be managed by either global or per-VDOM administrators. Each type of administrator will have a different view of the GUI that corresponds to their role. For more information, see Administrator roles and views on page 3837.

For information about configuring per-VDOM administrators, see Create per-VDOM administrators on page 3843.

Using FortiAuthenticator and FortiToken for 2FA

FortiAuthenticator can be used to centralize identity and access management and to enforce 2FA with FortiTokens.

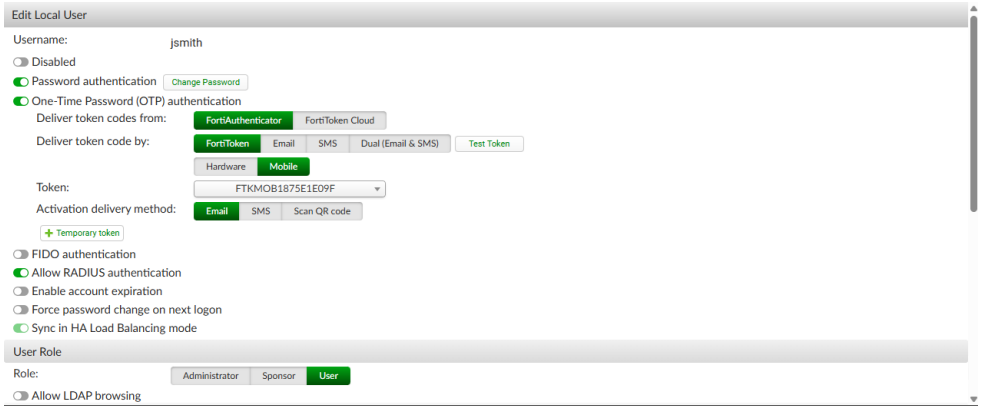
In this example, FortiAuthenticator has local users configured with a FortiToken applied to each user. A remote user group is configured on the FortiGate to connect to the FortiAuthenticator via MSCHAPv2. This user group is applied to the Administrator account.

If you encounter the error *No Message-Authenticator attribute* when you test your RADIUS connection, ensure that your FortiAuthenticator is running a version that resolves CVE-2024-3596.

See RADIUS Connection on page 3530 for details.

To configure the FortiAuthenticator:

1. Go to *Authentication* > *User Management* > *Local Users* to define a *Local User*.
2. Enable *One-Time Password (OTP) authentication*.

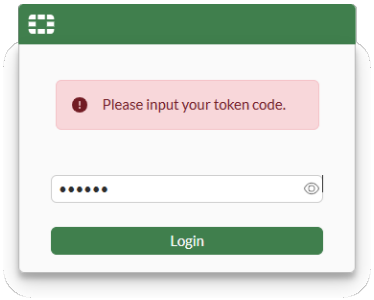


3. Save the user.

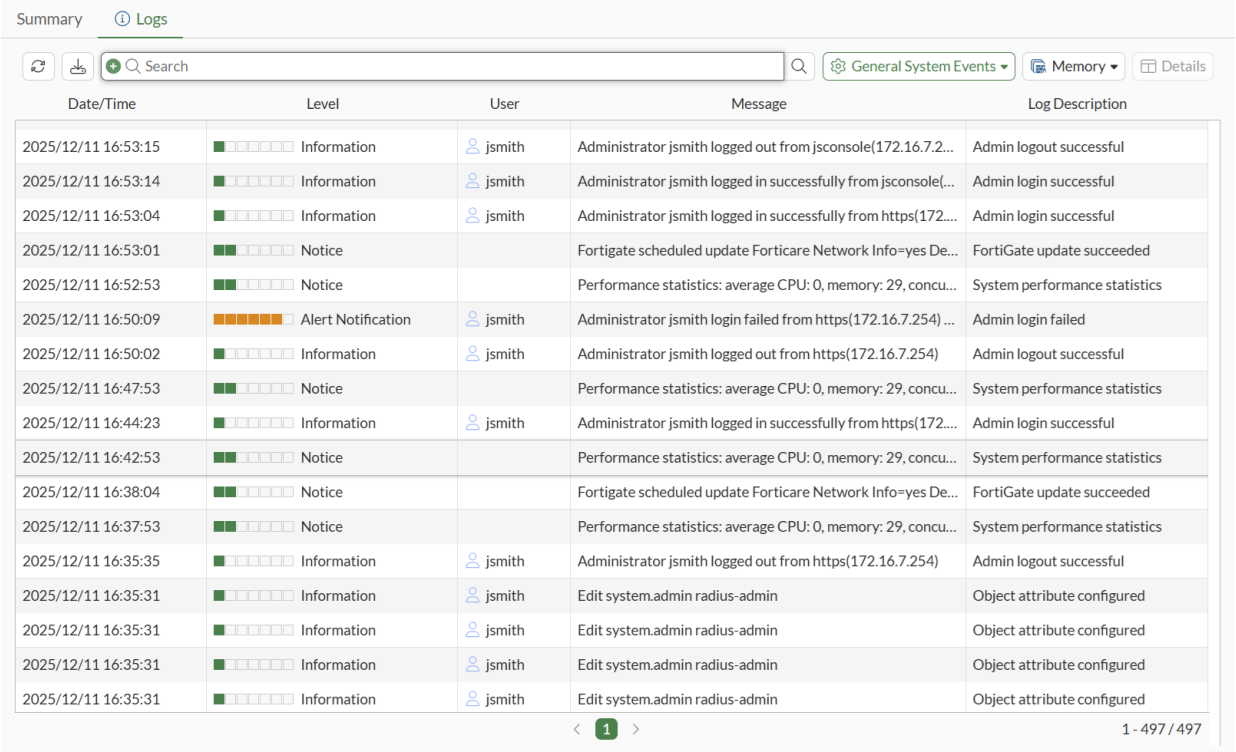
4. Go to *Authentication* > *RADIUS Service* > *Clients* to create a new client:
   a. Enter the IP of the connecting FortiGate.
   b. Create a Secret.
   c. Save the client.
5. Go to *Authentication* > *RADIUS Service* > *Policies* to create a new policy:
   a. Select the newly created *RADIUS Client*.
   b. Set *Identity Source* to *local* for *Local Users*.
   c. Save the policy.

To configure the FortiGate:

1. Add a RADIUS server:
   a. Go to *User & Authentication* > *RADIUS Servers* and click *Create New*.
   b. Optionally, specify your *Authentication method*. For example, specify *MS-CHAP-v2*.
   c. For the *Primary Server*, set *IP/Name* to the IP address of the FortiAuthenticator and enter the *Secret*.
   d. Click *OK*.
2. Create a user group:
   a. Go to *User & Authentication* > *User Groups* and click *Create New*.
   b. In the *Remote Groups* table, click *Add*.
   c. Add the *Remote Server* that was just created and click *OK*.
   d. Click *OK*.
3. Create a new admin user:
   a. Go to *System* > *Administrators* and click *Create New* > *Administrator*.
   b. Set *Type* to *Match all users in a remote server group*.
   c. Choose an *Administrator profile*.
   d. Set *Remote User Group* to the new group that was just created.
   e. Click *OK*.
4. Log out and then back into the FortiGate.
5. When prompted, enter the user credentials for the FAC users.
6. If the password is correct, you will be prompted for the FortiToken key.



7. Enter the key or approve the push notification.
   You are now logged into the FortiGate.
8. Go to *Log & Events* > *System Events* and find the user logs indicating the remote user has logged in successfully.




```
FGT-Corp $ execute log filter category event

 FGT-Corp $ execute log display
 1028 logs found.
 10 logs returned.

 1: date=2025-12-11 time=19:38:34 eventtime=1765510714165848638 tz="-0800" logid="0100032001"
type="event" subtype="system" level="information" vd="root" logdesc="Admin login successful"
sn="1765510714" user="jsmith" ui="jsconsole(172.16.7.254)" method="jsconsole"
srcip=172.16.7.254 dstip=172.16.7.1 action="login" status="success" reason="none"
profile="prof_admin" msg="Administrator jsmith logged in successfully from jsconsole
(172.16.7.254)"
```

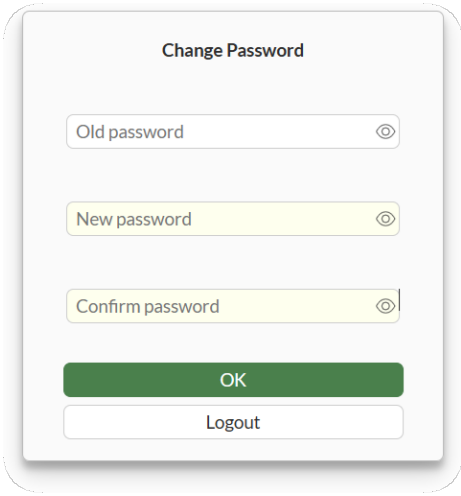
When using MSCHAP or MSCHAPv2, the FortiToken must be managed by the FortiAuthenticator. It will not function when using FortiTokens managed by FortiIdentity Cloud.

When the token code is entered by the user, this code is encoded into the MSCHAP message as it is sent to FortiAuthenticator. FortiAuthenticator cannot decode the original code, but it can perform a comparison by calculating the hash from the FortiToken seed that resides on the FortiAuthenticator. To work with FortiIdentity Cloud, PAP must be used, preferably with RadSec.

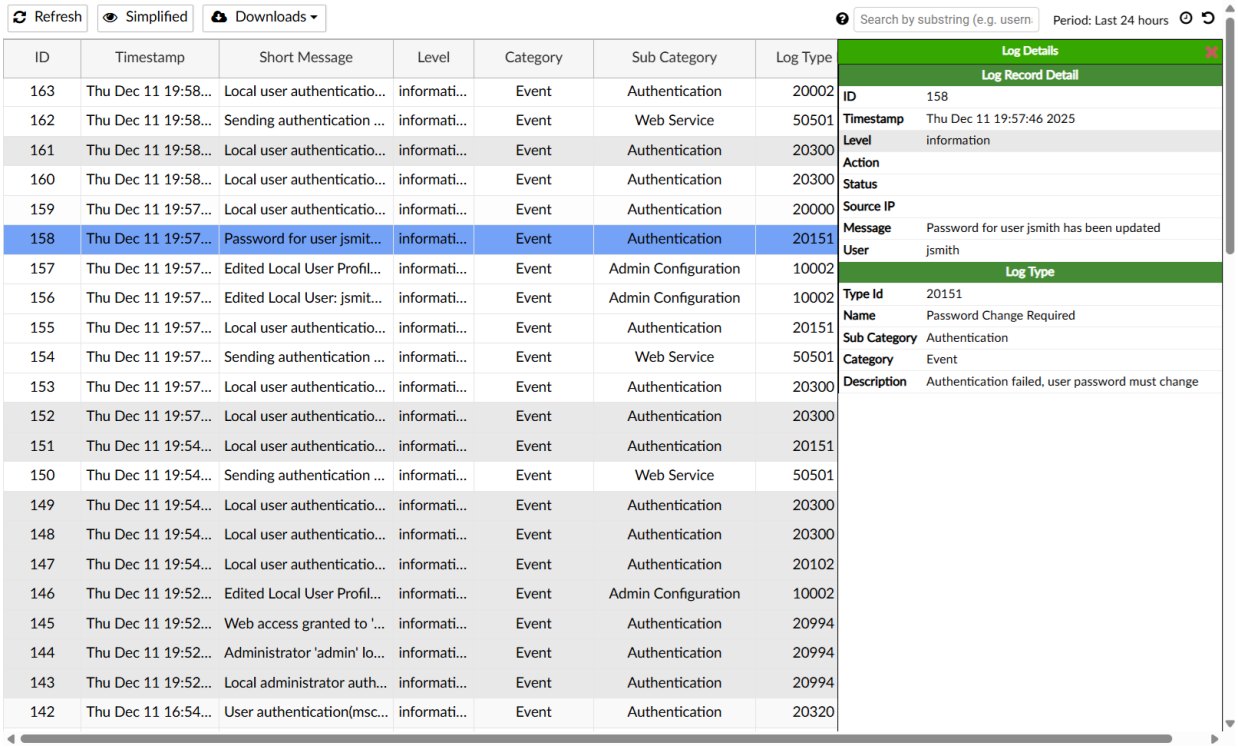
Changing your remote user password after expiry

When a user password expires on the FortiAuthenticator, the FortiGate can force the admin user to change their password. To support this feature, the FortiGate must use MSCHAP or MSCHAPv2 to communicate with the FortiAuthenticator.

When a password is expired and an administrator logs into the FortiGate, after passing authentication and 2FA, they will be prompted to change their password.



On the FortiAuthenticator, find the logs for the password change in *Logging* > *Log Access* > *Logs*.



REST API administrator

REST API administrator accounts are used for automated configuration, backup creation, and monitoring of the FortiGate.

For more information about the REST API, see the Fortinet Development Network (FNDN). Note that an account is required to access the FNDN.

Only an administrator with the *super_admin* profile can create a REST API administrator by using the GUI or CLI.

To create a REST API administrator in the GUI:

1. Go to *System* > *Administrators*.
2. Select *Create New* > *REST API Admin*.
3. Configure the administrator:

Username	The username of the administrator. Do not use the characters < > ( ) # " ' in the administrator username. Using these characters in an administrator username might have a cross site scripting (XSS) vulnerability.
Administrator Profile	Where permissions for the REST API administrator are defined. A REST API administrator should have the minimum permissions required to complete the request.
PKI Group	Certificate matching is supported as an extra layer of security. Both the client certificate and token must match to be granted access to the API.
CORS Allow Origin	Cross Origin Resource Sharing (CORS) allows third-party web apps to make API requests to the FortiGate using the token.
Trusted Hosts	The following can be used to restrict access to FortiGate API: • Multiple trusted hosts/subnets can be configured • IPv6 hosts are supported • Allow all (0.0.0.0/0) is not allowed You need your *Source Address* to create the trusted host.

4. Click *OK*.
   An API token is generated. Make note of the token, as it is only shown once.

To create a REST API administrator in the CLI:

1. Create the REST API administrator:

```
config system api-user
    edit "api-admin"
        set comments <string>
        set api-key ************
        set accprofile "API profile"
        set vdom "root"
        set peer-auth enable
        set peer-group <group>
        config trusthost
            edit 1
```

```
                set ipv4-trusthost <class_ip&net_netmask>
            next
            ...
        end
    next
end
```

2. Generate the API token:

```
# execute api-user generate-key <API username> [expiry in minutes]
```

Make note of the token, as it is only shown once.

It is highly recommended to apply the principle of least privilege to the REST API administrator account and avoid using the generic *super_admin* profile for permissions. The profiles *admin_no_access* or *super_admin_readonly* can be used for generic read-only access. However, consider defining a new administrator profile with the required permissions for the account. For example, you could use a specific API user to query the FortiGate for just their own status. In that case, the profile would be configured as read-only.

SSO administrators

SSO administrators are automatically created when the FortiGate acts as a SAML service provider (SP) with *SAML Single Sign-On* enabled in the Security Fabric settings.

On the system login page, an administrator can log in with their username and password against the root FortiGate acting as the identity provider (IdP) in the Security Fabric. After the first successful log in, this user is added to the administrators table (*System > Administrators* under *Single Sign-On Administrator*). The default profile selected is based on the SP settings (*Default admin profile*). See Configuring a downstream FortiGate as an SP on page 4458 for more information.

SSO administrators can be manually configured in FortiOS.

To manually configure an SSO administrator in the GUI:

1. Go to *System > Administrators* and click *Create New > SSO Admin*.
2. Enter the username.
3. Select an administrator profile.
4. Click *OK*.

To manually configure an SSO administrator in the CLI:

```
config system sso-admin
    edit <name>
        set accprofile <profile>
        set vdom <vdom>
    next
end
```

FortiCloud SSO

FortiGate can be configured to allow administrators to log in using FortiCloud single sign-on. Both IAM and non-IAM users on the FortiCloud support portal are supported. Non-IAM users must be the FortiCloud account that the FortiGate is registered to.

It is recommended to only allow administrative access through a non-public facing interface. In addition, use the principle of least privileges when determining user permissions. If a public interface must be used, apply local-in policy to allow only trusted hosts.

To configure an IAM user in FortiCloud:

1. Log in to your FortiCloud account at support.fortinet.com.
2. Select *Services* > *IAM*.
3. See the FortiCloud Identity & Access Management (IAM) guide for more information.

To manually enable FortiCloud single sign-on in the GUI:

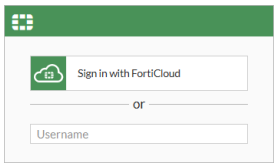
1. Log in to the FortiGate and go to *System* > *Settings* > *Access*.
2. In the *Administration Settings* section, enable *Allow administrative login using FortiCloud SSO*.
3. Click *Apply*.

To manually enable FortiCloud single sign-on in the CLI:

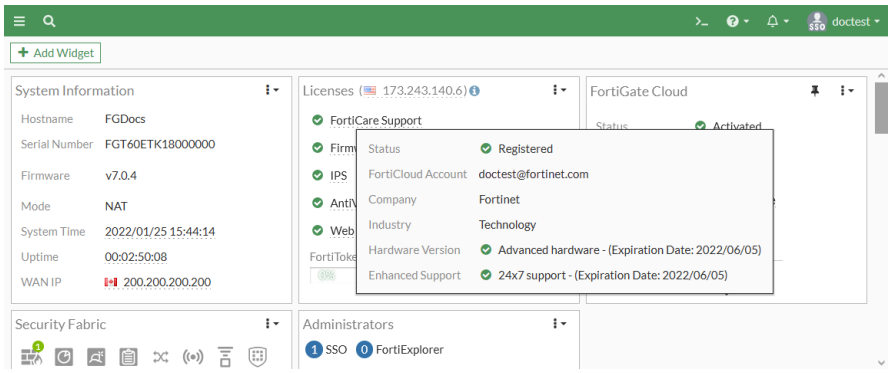
```
config system global
    set admin-forticloud-sso-login {enable | disable}
end
```

To log in to the FortiGate with the FortiCloud user:

1. Go to the FortiGate log in screen.



2. Click *Sign in with FortiCloud*. The FortiCloud log in page opens.
3. Enter the FortiCloud account credentials and click *Login*.
   You are logged in to the FortiOS GUI. The SSO username is shown in the top right corner of the GUI.



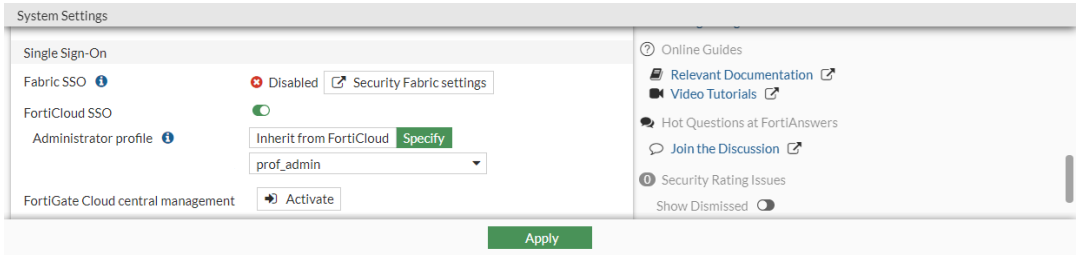
Allowing the FortiGate to override FortiCloud SSO administrator user permissions

The FortiGate can allow single sign-on (SSO) from FortiCloud and FortiCloud IAM users with administrator profiles inherited from FortiCloud or overridden locally by the FortiGate. Similarly, users accessing the FortiGate remotely from FortiGate Cloud can have their permissions inherited or overridden by the FortiGate.

It is recommended to only allow administrative access through a non-public facing interface. In addition, use the principle of least privileges when determining user permissions. If a public interface must be used, apply local-in policy to allow only trusted hosts.

To enable FortiCloud SSO in the GUI:

1. Go to *System* > *Settings* > *Access*.
2. In the *Single Sign-On* section, enable *FortiCloud SSO*.
3. Set the default *Administrator profile* to assign: *Inherit from FortiCloud*, or *Specify* and select a profile from the dropdown.



4. Click *Apply*.

To enable FortiCloud SSO in the CLI:

```
config system global
    set admin-forticloud-sso-login enable
    set admin-forticloud-sso-default-profile <profile>
end
```

The following administrator profiles are assigned based on the inherited or overwritten permissions:

User type	Inherited from FortiCloud/FortiGate Cloud	Specify
FortiCloud	Uses the super_admin profile.	Local user profile
FortiCloud IAM	Is based on the IAM permission profile's FortiOS SSO portal settings: • If *Access* is disabled = no access • If *Access* is enabled and the *Access Type* is set to *SuperAdmin* = super_admin profile • If *Access* is enabled and the *Access Type* is set to *Read Only*= super_admin_readonly profile	Local user profile
FortiGate Cloud subscription tier	Uses the super_admin profile.	Local user profile
FortiGate Cloud free tier	Has read-only access.	Cannot override

This topic includes four use case examples:

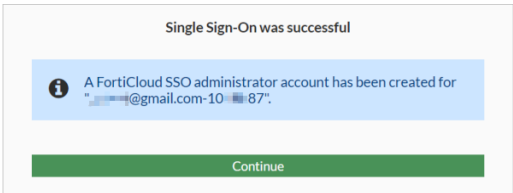
- Example 1: specifying permissions for a FortiCloud SSO user
- Example 2: inheriting FortiCloud permissions for a FortiCloud SSO user
- Example 3: specifying a local user profile for a FortiCloud IAM user
- Example 4: accessing a FortiGate remotely from FortiGate Cloud

Example 1: specifying permissions for a FortiCloud SSO user

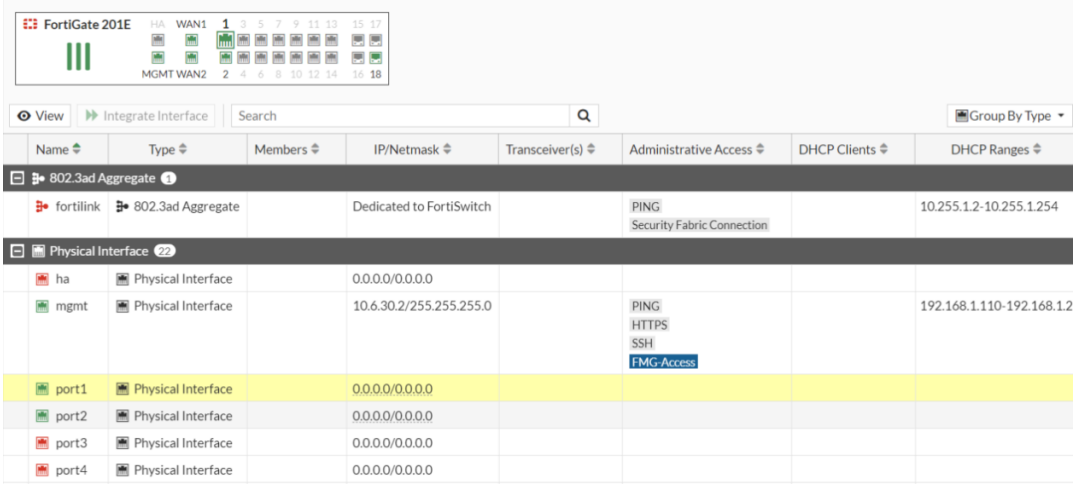
In this example, a FortiCloud SSO user is configured to override permissions and use the prof_admin profile, which is a local read-only profile.

To configure the FortiCloud SSO user:

1. Go to *System* > *Settings* > *Access*.
2. In the *Single Sign-On* section, enable *FortiCloud SSO*.
3. Set *Administrator profile* to *Specify*, and select *prof_admin*. The FortiCloud SSO user will be created upon the first login.
4. Get the user to log in to the FortiGate:
   a. On the FortiOS login screen, click *Sign in with FortiCloud*. The FortiCloud log in page opens.
   b. Click *Email Login*.
   c. Enter the FortiCloud account credentials and click *Log In*.
   The new SSO user is created.



Since the profile has read-only access, the SSO user can only view items (such as interfaces) and cannot edit them.

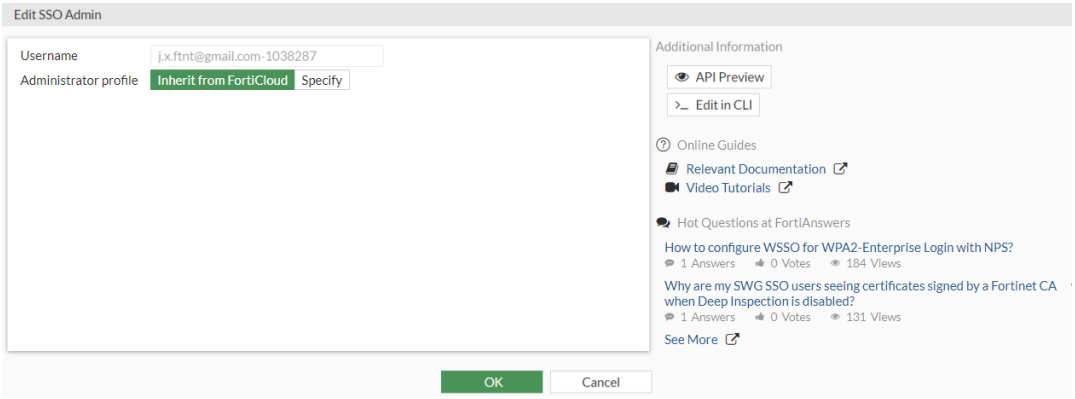


Example 2: inheriting FortiCloud permissions for a FortiCloud SSO user

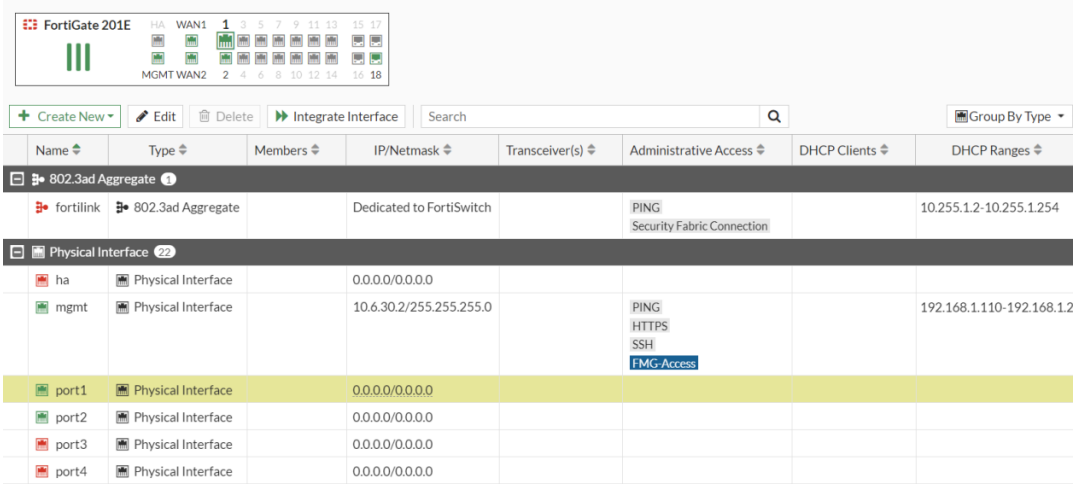
In this example, a local administrator changes the permissions of an existing FortiCloud SSO user (created in the previous example) to *Inherit from FortiCloud*, which means the super_admin profile will be used.

To configure the existing SSO user:

1. Go to *System* > *Administrators* and edit the user in the *FortiCloud SSO Administrator* section (*********@gmail.com*).
2. Set *Administrator profile* to *Inherit from FortiCloud*.



3. Click *OK*.
4. Get the user to log in to the FortiGate. Since the profile changed to super_admin, they can modify items (such as interfaces).



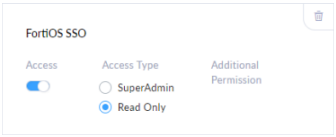
Example 3: specifying a local user profile for a FortiCloud IAM user

In this example, a FortiCloud IAM user is configured to have read-only SSO access based on the settings in the FortiOS SSO portal. Once the FortiCloud IAM user logs in, an administrator with super_admin access changes the permission of the IAM user to have super_admin access.

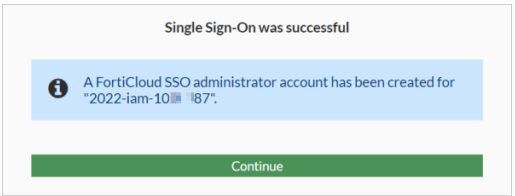
This example assumes the *FortiOS SSO* portal has already been added to the IAM permission profile. See Creating a permission profile and Managing permission profiles in the Identity & Access Management (IAM) Guide for more information about configuring permission profiles in FortiCloud.

To configure the FortiCloud IAM user:

1. In FortiCloud, configure the permission profile:
   a. Go to *Services* > *IAM*, then click *Permission Profiles*.
   b. Select a profile and click *Edit*.
   c. In the *FortiOS SSO* portal, enable *Access*. Set the *Access Type* to *Read Only*.



   d. Click *Update*.
2. Get the user to log in to the FortiGate:
   a. On the FortiOS login screen, click *Sign in with FortiCloud*. The FortiCloud log in page opens.
   b. Click *IAM Login*.
   c. Enter the IAM account credentials and click *Log In*.
   The new SSO user is created with a super_admin_readonly profile.



3. Update the IAM user permission to have super_admin access:
   a. Log in to the FortiGate with a super_admin administrator account.
   b. Go to *System* > *Administrators* and edit the IAM user (*2022*).
   c. Set *Administrator profile* to *Specify* and select *super_admin*.
   d. Click *OK*.
4. Get the user to log in to the FortiGate again. Since the profile changed to super_admin, they can modify items.

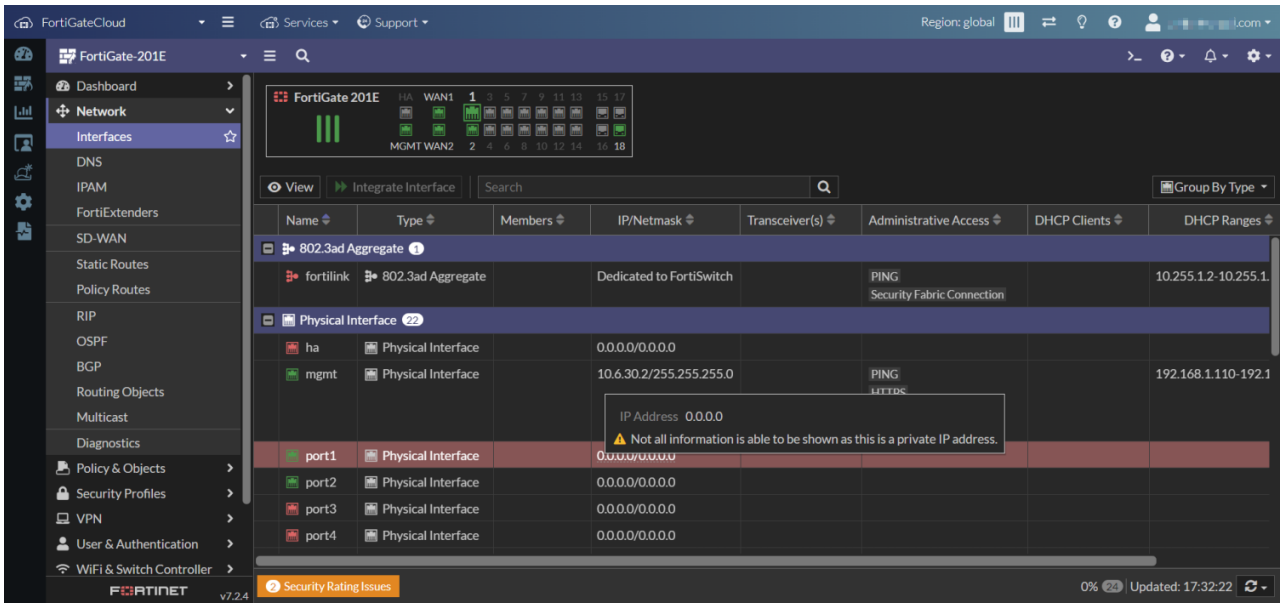
Example 4: accessing a FortiGate remotely from FortiGate Cloud

In this example, a FortiGate Cloud user with a paid subscription accesses the FortiGate remotely from FortiGate Cloud. When the user logs in with SSO, the profile has super_admin access. After the FortiGate Cloud user logs in, an administrator with super_admin access changes the permission of the FortiGate Cloud user to have prof_admin (read-only) access.

FortiGate Cloud must be accessed from a FortiGate Cloud 2.0 portal (also called FortiGate Cloud Premium) in order to have remote access using the FortiGate Cloud proxy. See Getting started with FortiGate Cloud 2.0 for more details.

To access a FortiGate remotely from FortiGate Cloud:

1. Log in to the FortiGate Cloud 2.0 portal.
2. Go to *Inventory* > *Asset List*. Select the desired FortiGate, then click *Remote Access*.
   FortiGate Cloud accesses the FortiGate using the FortiGate Cloud proxy and creates a super_admin user. The FortiOS interface is displayed in the current browser window.
3. Log out of FortiGate Cloud.
4. Update the FortiGate Cloud user permission to have prof_admin access:
   a. Log in to the FortiGate with a super_admin administrator account.
   b. Go to *System* > *Administrators* and edit the user in the *FortiGate Cloud SSO Administrator* section (*********@gmail.com*).
   c. Set *Administrator profile* to *Specify* and select *prof_admin*.
   d. Click *OK*.
5. Log in to the FortiGate Cloud 2.0 portal and access the FortiGate remotely again. Since the profile changed to prof_admin, they can only view items (such as interfaces).



Password policy

Brute force password software can launch more than just dictionary attacks. It can discover common passwords where a letter is replaced by a number. For example, if `p4ssw0rd` is used as a password, it can be cracked.

Using secure passwords is vital for preventing unauthorized access to your FortiGate. Starting in FortiOS 7.6.5, the password policy is enabled by default and enforced. Your password must meet the following requirements before you can log in to FortiOS:

- 1 uppercase letter
- 1 lowercase letter
- 1 special character
- 1 number (0-9)
- A minimum length of 12 characters

If your password meets the requirements, you can log in to FortiOS after upgrade.

During upgrade to FortiOS 7.6.5 or later, any pre-shared secrets (`psksecret`) for VPN remote gateways are retained, even when they fail to meet the password policy length requirements. This allows FortiOS to retain the IPsec configuration during upgrade.

If your password does not meet the requirements, you must change your password before you can log in to the GUI or CLI.

When changing the password, consider the following to ensure better security:

- Do not use passwords that are obvious, such as the company name, administrator names, or other obvious words or phrases.
- Use numbers in place of letters, for example: `passw0rd`.

- Administrator passwords can be up to 64 characters.
- Include a mixture of numbers, symbols, and upper and lower case letters.
- Use multiple words together, or possibly even a sentence, for example: `correcthorsebatterystaple`.
- Use a password generator.
- Change the password regularly and always make the new password unique and not a variation of the existing password. For example, do not change from `password` to `password1`.
- Limit the number of passwords that can be reused.
- Make note of the password and store it in a safe place away from the management computer, in case you forget it; or ensure at least two people know the password in the event one person becomes unavailable. Alternatively, have two different admin logins.

FortiGate allows you to create a password policy for administrators and IPsec pre-shared keys. With this policy, you can enforce regular changes and specific criteria for a password policy.

If you add a password policy or change the requirements on an existing policy, the next time that administrator logs in to the FortiGate, the administrator is prompted to update the password to meet the new requirements before proceeding to log in.

For information about setting passwords, see Default administrator password on page 3810.

Although not recommended, you can disable the password policy.

To disable the password policy in the GUI:

1. Go to *System* > *Settings* > *Access*.
2. In the *Security* section, set *Password scope* to *Off*.
3. Click *Apply*.

To disable a system password policy in the CLI:

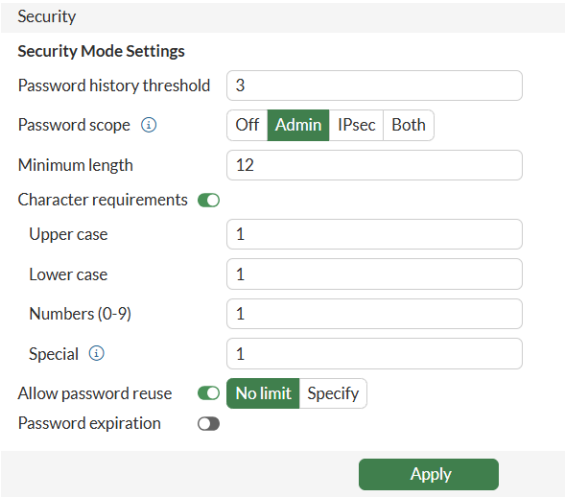
```
config system password-policy
    set status disable
end
```

To create or edit a system password policy the GUI:

1. Go to *System* > *Settings* > *Access*.
2. In the *Security* section, configure the options:

Password history threshold	Global maximum number of previous passwords saved for each local user and system administrator.
Password scope	Where the password policy applies: • *Off* - Password policy is disabled. • *Admin* - Password policy applies to administrators. • *IPsec* - Password policy applies to IPsec pre-shared keys. • *Both* - Password policy applies to both administrators and IPsec pre-shared keys.

Minimum length	The minimum length, between 12 and 128 characters.
Character requirements	Enable to specify character requirements for the password: • *Upper case* - If the password must contain uppercase (A, B, C) characters. • *Lower case* - If the password must contain lowercase (a, b, c) characters. • *Numbers (0 - 9)* - If the password must contain numbers (1, 2, 3). • *Special* - If the password must contain special or non-alphanumeric characters: !, @, #, $, %, ^, &, *, (, and)
Allow password reuse	Applies only to administrators. Available when *Password scope* set to *Admin* or *Both*. Disable for unlimited password reuse. Enable to specify the number of passwords that can be reused. Choose from: • *No limit* - Allow unlimited password reuse. • *Specify* - Specify how many times a password can be reused. Must be lower than the *Password history threshold*. Password history is visible in the backup configuration.
Password expiration	Enable to specify the duration of the password before a new one must be entered.



3. Click *Apply*.

To create or edit a system password policy the CLI:

The password policy applies to all administrator accounts when enabled, including the built-in admin account named *admin*. If an existing system administrator account fails to comply with the enabled password policy, the administrator is forced to change passwords on next login.

1. Configure a global history password limit.
   In this example, the global policy is to save three passwords for each local user and system administrator.

```
config system global
    set user-history-password-threshold 3
end
```

2. Configure a password policy for system administrators:
   a. Enable the password policy.

```
config system password-policy
    set status enable
end
```

   b. Enable password expiration:

```
config system password-policy
    set expire-status enable
end
```

   c. Set the number of days after which passwords expire, the password criteria, and password reuse limit.
   In this example, the `reuse-password-limit` is set to `1`, which means one of the globally-set three saved passwords can be reused.

```
config system password-policy
    set apply-to {admin-password | ipsec-preshared-key}
    set minimum-length <12-128>
    set min-lower-case-letter <0-128>
    set min-upper-case-letter <0-128>
    set min-non-alphanumeric <0-128>
    set min-number <0-128>
    set expire-day 90
    set reuse-password enable
    set reuse-password-limit 1
end
```

To create a password policy for a local user in the CLI:

Multiple password policies can be created and applied to different local user accounts.

1. Configure a global password history limit.
   In this example, the global policy is to save three passwords for each local user and system administrator.

```
config system global
    set user-history-password-threshold 3
end
```

2. Configure a password policy for local users:
   a. Before you can configure the password limit, enable `expire-status` and `reuse-password`.

```
config user password-policy
    edit 1
        set expire-status enable
        set reuse-password enable
```

```
    next
end
```

b. Specify the maximum number of times a user can reuse a password.

In this example, the `reuse-password-limit` is set to `1`, which means one of the globally-set three saved passwords can be reused.

```
config user password-policy
    edit 1
        set reuse-password-limit 1
    next
end
```

3. Assign the password policy to a local user.

In this example, password policy `1` is assigned to local user `local2`.

```
config user local
    edit "local2"
        set type password
        set passwd-policy "1"
        set passwd ********
    next
end
```

4. Add the local user to a firewall policy, an Agentless VPN policy, or to FortiGate user groups used in policies.

Before the password for the local user expires, the FortiOS GUI provides the option to change the password during login or skip the password change.

If the password for the local user has expired, the FortiOS GUI provides the option to change the password during login. When the local user enters a password that adheres to the policy, the login continues. If the new password has been used too many times before, a warning message is displayed.

Public key SSH access

Public-private key pairs can be used to authenticate administrators connecting to the CLI using an SSH client. These keys can be RSA, ECDSA, or EdDSA.

Weigh the pros and cons of using key-pair authentication, versus passwords, when considering their use:

	Key-pair	Password
Pros	More secure (higher complexity)	Easy to remember
	Restricts logon to hosts that have the private key	Easy to update
	Never sent to the FortiGate	Can log in from any system
	Can add a password in addition to the key	

	Key-pair	Password
Cons	More complex to implement	Might be guessable or brute forced
	The private key is only as secure as the system storing it	Could be reused and compromised on another system
	More complicated to train users and administrators to use keys	Might be stored in plain text on an authenticating device (This does not apply to FortiGates)
		Can be phished or observed if written down

Key-pair authentication is often implemented when connecting to the FortiGate without any human interaction, such as when using a script. The script can leverage existing mechanisms to secure private keys, instead of trying to develop a way to securely store a username and password.

Generating the key pair

Key pairs can be generated and added in multiple different ways. This example shows generating a key pair using *PuTTY Key Generator* and adding the private key to the endpoint using *PuTTY Pageant*.

To create the key pair using PuTTY:

1. Download and install PuTTY.
2. Run *PuTTYgen.exe*.
3. Set *Type of key to generate* to *RSA*, *ECDSA*, or *EdDSA*.
4. Click *Generate*, then move the mouse cursor around in the blank space to generate randomness while the keys are generated.
5. Save both the public and private keys. Optionally, a key passphrase can be entered to protect the private key. This passphrase will be requested during login.

To add the public key to the FortiGate:

1. Ensure that you have configured an appropriate admin account profile for this new user, that only has the permissions that this admin will use to satisfy the rule of least privileges.
2. Delete the *Key comment*, then copy the public key from the *PuTTY Key Generator*.
   Conversely, you can also open the saved public key in Notepad, remove the line breaks from the key, then remove extraneous lines:

   ```
   ---- BEGIN SSH2 PUBLIC KEY ----
   Comment: "rsa-key-2022XXXX"
   ```

   ```
   ---- END SSH2 PUBLIC KEY ----
   ```

3. Add the key to the FortiGate:

```
config system admin
    edit <admin>
        set vdom <vdom>
        set accprofile <profile>
        set ssh-public-key1 "<key_type> <key_value>"
        set password <password>
    next
end
```

Where <key_value> is the copied key, and <key_type> depends on the type of key that was generated:

RSA	ssh-rsa
ECDSA	ecdsa-sha2-nistp256 ecdsa-sha2-nistp384 ecdsa-sha2-nistp521
EdDSA	ssh-ed25519

To add the private key to the endpoint:

1. Open *PuTTY Pageant*.
2. Click *Add Key* or *Add Key (encrypted)* and select the previously saved private key.
3. Click *Close*.

When creating the user, several requirements must be met:

- When multi-vdom mode is enabled, a VDOM must be specified.
- An admin account profile must be selected. Use the rule of least privileges.
- A password must be set for backup. Since the objective is to login with a private key, you may want to assign a long random string as the password.
- The administrator must enter its own password to verify its credentials.

You can now log in to the FortiGate on an SSH connection without using a password.

If using PuTTY, the username can be entered under *Connection > Data* in the *Auto-login username* field. The key can be applied under *Connection > SSH > Auth > Credentials*.

If a key passphrase was specified during key creation, then the SSH session will require the user to enter the passphrase during login:

```
Using username "ssh-user".
Authenticating with public key "rsa-key-20251215"
Passphrase for key "rsa-key-20251215": <enter the passphrase here>
FGDocs $
```

The generated keys can also be used in a certificate to authenticate with the FortiGate.
See Administrative access using certificates on page 3982 for information about generating and using certificates for administrative authentication.

Separating the SSHD host key from the administration server certificate

Separating the SSHD host key from the administration server certificate addresses the issue where the administration server key tends to overwrite one of the key files, which can lead to complications. This resolves the problem where the SSH module regenerates the host key files after a factory reset. This action previously prompted a warning message when an older SSH client attempted to log in to the FortiGate using SSH.

```
config system global
    set ssh-hostkey-override {enable | disable}
    set ssh-hostkey-password <password>
    set ssh-hostkey <encrypted_private_key>
end
```

The `ssh-hostkey-algo` option under `config system global` supports ECDSA 384 and ECDSA 256, allowing the SSHD to accommodate the most commonly used host key algorithms.

To configure SSH host key override in SSHD:

1. Using the ssh-keygen tool, generate the host key (ecdsa-sha2-nistp384 is used in this example).
2. Configure the SSH host key override settings:

```
config system global
    set ssh-hostkey-override enable
    set ssh-hostkey-algo ecdsa-sha2-nistp384
    set ssh-hostkey-password **********
    set ssh-hostkey <encrypted_private_key>
end
```

3. On a PC, attempt to log in to the FortiGate with the defined ecdsa-sha2-nistp384 algorithm:

```
root@PC05:~# ssh admin@172.16.200.1
The authenticity of host '172.16.200.1 (172.16.200.1)' can't be established.
ECDSA key fingerprint is SHA256:mcrMXSjtN/YjY3zQgZpxk77ezxPVGGGOL/GUOG8Oijs.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.16.200.1' (ECDSA) to the list of known hosts.
```

4. Verify the server host key algorithms:

```
root@PC05:~# nmap -sV --script ssh2-enum-algos 172.16.200.1
Starting Nmap 7.01 ( https://nmap.org ) at 2023-11-07 15:47 PST
Nmap scan report for FGT_A (172.16.200.1)
Host is up (0.00013s latency).
```

```
Not shown: 995 filtered ports
PORT    STATE  SERVICE      VERSION
22/tcp  open   ssh          (protocol 2.0)
| ssh2-enum-algos:
|   kex_algorithms: (8)
|       diffie-hellman-group14-sha256
|       diffie-hellman-group16-sha512
|       diffie-hellman-group18-sha512
|       diffie-hellman-group-exchange-sha256
|       curve25519-sha256@libssh.org
|       ecdh-sha2-nistp256
|       ecdh-sha2-nistp384
|       ecdh-sha2-nistp521
|   server_host_key_algorithms: (1)
|       ecdsa-sha2-nistp384
|   encryption_algorithms: (3)
```

Restricting SSH and Telnet jump host capabilities

Jump hosts are used to access devices in separate security zones, such as the internet and an internal network. Administrator access profiles can be configured to prevent administrators from using the FortiGate as a jump host for SSH and Telnet connections.

To configure permission to execute SSH or Telnet commands in an access profile:

```
config system accprofile
    edit <name>
        set system-execute-ssh {enable | disable}
        set system-execute-telnet {enable | disable}
    next
end
```

To block SSH and Telnet connections for an administrator:

1. Disable permission to execute SSH or Telnet commands in an administrator access profile:

```
config system accprofile
    edit "test_accprofile"
        set system-execute-ssh disable
        set system-execute-telnet disable
    next
end
```

2. Configure an administrator in the profile:

```
config system admin
    edit "admin1"
        set accprofile "test_accprofile"
        set vdom "root"
```

```
        set password **********
    next
end
```

3. Log in as the new administrator, and attempt to connect to another host using SSH or Telnet:

```
# execute ssh root@172.16.200.55
You are not entitled to run the command.
Command fail. Return code -37
```

```
# execute ssh6 root@2000:172:16:200::55
You are not entitled to run the command.
Command fail. Return code -37
```

```
# execute telnet 172.16.200.55
You are not entitled to run the command.
Command fail. Return code -37
```

Restricting local administrator logins through the console

FortiOS can now restrict local administrator logins using the console when FortiGate can reach the remote authentication server. This enhancement provides more control over local administrator logins to improve system security.

```
config system global
    set admin-restrict-local {all | non-console-only | disable}
end
```

Command	Description
`admin-restrict-local {all | non-console-only | disable}`	Restrict local administrator logins when the remote authentication server is reachable. • `all`: Enable local administrator authentication restriction, including the console. • `non-console-only`: Enable local administrator authentication restriction, excluding the console. • `disable`: Disable local administrator authentication restriction.

Example 1

In this example, the local administrator restriction is set to `non-console-only`. As a result, local administrators cannot use non-console methods, such as SSH, to log in to FortiGate when the remote authentication server is reachable. However, local administrators can use the console to log in to FortiGate.

To exclude the console from local administrator login restrictions:

1. In FortiOS, set the administrator restriction to `non-console-only`:

```
config system global
    set admin-restrict-local non-console-only
end
```

2. Using SSH and the local administrator account, log in to FortiOS.
   Login is denied:

```
ssh admin@<ip address>
admin@<ip address>'s password:
Permission denied, please try again.
```

   The login failure is captured in the logs:

```
1: date=2024-04-01 time=15:42:08 eventtime=1712011328452918375 tz="-0700" logid="0100032002"
type="event" subtype="system" level="alert" vd="root" logdesc="Admin login failed" sn="0"
user="admin" ui="ssh(172.16.200.100)" method="ssh" srcip=172.16.200.100 dstip=172.16.200.1
action="login" status="failed" reason="none" msg="Administrator admin login failed from ssh
(172.16.200.100)"
```

3. Using the FortiGate console and the local administrator account, log in to FortiOS.
   Login is allowed:

```
FGT login: admin
Password:
Welcome!
```

   The successful login is captured in the logs:

```
1: date=2024-04-01 time=15:43:36 eventtime=1712011415358013250 tz="-0700" logid="0100032001"
type="event" subtype="system" level="information" vd="root" logdesc="Admin login successful"
sn="1712011415" user="admin" ui="console" method="console" srcip=0.0.0.0 dstip=0.0.0.0
action="login" status="success" reason="none" profile="super_admin" msg="Administrator admin
logged in successfully from console"
```

Example 2

In this example, the local administrator restriction is set to `all`. As a result, local administrators cannot use any method to log in to FortiGate when the remote authentication server is reachable.

To restrict all local administrator logins:

1. In FortiOS, set the local administrator restriction to `all`:

```
config system global
    set admin-restrict-local all
end
```

2. Using SSH and the local administrator account, log in to FortiOS.

Login is denied and the failure is captured in the logs.

3. Using the FortiGate console and the local administrator account, log in to FortiOS.
   Login is denied, and the failure is captured in the logs:

```
2: date=2024-04-01 time=16:26:56 eventtime=1712014017124846849 tz="-0700" logid="0100032002"
type="event" subtype="system" level="alert" vd="root" logdesc="Admin login failed" sn="0"
user="admin" ui="console" method="console" srcip=0.0.0.0 dstip=0.0.0.0 action="login"
status="failed" reason="none" msg="Administrator admin login failed from console"
```

Remote administrators with TACACS+ VSA attributes

Vendor-Specific Attributes (VSAs) can be used with TACACS+ authentication and authorization in wildcard system administrator access to FortiGates from browsers and SSH. The `memberof` VSA can be used in remote TACACS+ user group for group matching. The `vdom` VSA returned from TACACS+ can be used to overwrite the VDOM in the `system admin` settings. The `admin_prof` VSA returned from TACACS+ can be used to overwrite the `accprofile` in the `system admin` settings.

Example

In this example, a FortiGate is configured with multiple VDOMs, and the root acts as the management VDOM. Administrators attempt to log in with SSH or HTTPS through each VDOM.

Using the VSA values for the `vdom` and `admin_prof` attributes returned from the TACACS+ server, the FortiGate can allow access only to the VDOMs returned with the permissions from the corresponding administrator profile. If no VSA values are returned from TACACS+ , then the FortiGate uses the default values under the `config system admin` settings.

The TACACS+ server settings are configured as follows:

```
user = admin-all-vdom {
    default service = permit
    member = sys_admin_all_vdom
    …
}
user = admin-vdom1 {
    default service = permit
    member = sys_admin_vdom1
    …
}
group = sys_admin_all_vdom {
default service = permit
    service = fortigate {
        memberof = group3
        admin_prof = admin_all_vdom
    }
}
group = sys_admin_vdom1 {
default service = permit
    service = fortigate {
```

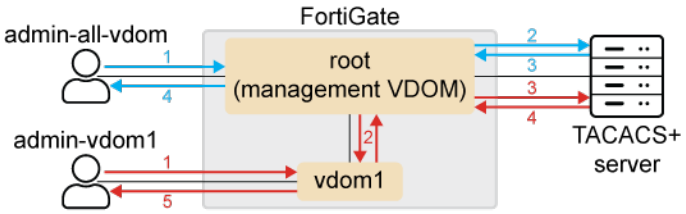
```
        memberof = group3
        admin_prof = admin_vdom1
        vdom = vdom1
    }
}
```

For multiple VDOMs, each VDOM must be specified in a separate field. For example, for access to vdom1 and vdom2:

```
vdom = vdom1
vdom = vdom2
```

Some TACACS+ servers, such as Linux TACACS+ servers, may only return the last VDOM specified.

The authentication process is as follows:



Authentication for admin-all-vdom:

1. The administrator attempts to log in to the FortiGate over the remote TACACS+ user group, `remote-tacacs`.
2. The FortiGate sends an authorization request to the TACACS+ server.
3. TACACS+ authenticates the admin-all-vdom user. The user matches the `sys_admin_all_vdom` TACACS+ group. TACACS+ returns following VSA values:
   - `memberof = group3`
   - `admin_prof = admin_all_vdom`
4. The FortiGate authenticates and authorizes the user based on the returned `memberof` group. The admin_prof value overwrites the `accprofile` setting configured under `system admin`. Since no other VDOM VSA is returned, the FortiGate matches the user to the default VDOM configured under `system admin`, which is `admin_no_access`.

Authentication for admin-vdom1:

1. The administrator attempts to log in to the FortiGate over the remote TACACS+ user group, `remote-tacacs`.
2. vdom1 forwards the request to the management VDOM, which is the root.
3. The FortiGate sends an authorization request to the TACACS+ server through the management VDOM.
4. TACACS+ authenticates the admin-vdom1 user. The user matches the `sys_admin_vdom1` TACACS+ group. TACACS+ returns following VSA values:
   - `memberof = group3`
   - `admin_prof = admin_vdom1`
   - `vdom = vdom1`

5. The FortiGate authenticates and authorizes the user based on the returned `memberof` group. The other VSA values overwrite the `accprofile` and VDOM settings configured under `system admin`. The user is only allowed to access vdom1 with the administrative permissions allowed for admin_vdom1.

To configure the FortiGate:

1. Create two system administrator profiles.
   a. Configure admin_vdom1 who has read-write access to vdom1 (except for firewall policies) and is redistricted from using diagnose commands in the CLI:

```
config system accprofile
    edit "admin_vdom1"
        set secfabgrp read-write
        set ftviewgrp read-write
        set authgrp read-write
        set fwgrp custom
        set cli-diagnose disable
        config fwgrp-permission
            set policy read
            set address read
            set service read
            set schedule read
            set others read
        end
    next
end
```

   b. Configure admin_all_vdom who has read-write access to all VDOMs, but not with super_admin permissions:

```
config system accprofile
    edit "admin_all_vdom"
        set secfabgrp read-write
        set ftviewgrp read-write
        set authgrp read-write
        set sysgrp read
        set netgrp read-write
        set loggrp read-write
        set fwgrp read-write
        set vpngrp read
        set utmgrp read
        set wanoptgrp read
        set wifi read
    next
end
```

2. Configure the TACACS+ server:

```
config user tacacs+
    edit "tac1"
        set server "10.1.100.34"
        set key XXXXXXXXXXXX
```

```
        set authorization enable
    next
end
```

3. Configure the remote TACACS+ group with group matching:

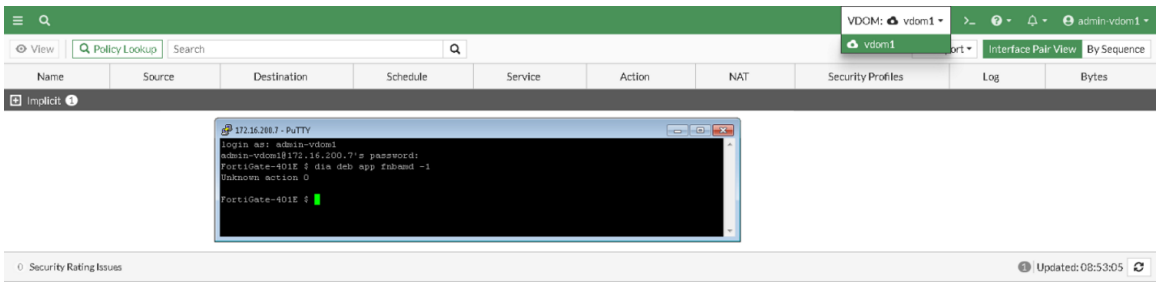
```
config user group
    edit "remote-tacacs"
        set member "tac1"
        config match
            edit 1
                set server-name "tac1"
                set group-name "group3"
            next
        end
    next
end
```

4. Configure the wildcard administrative user assigned to the remote TACACS+ group:

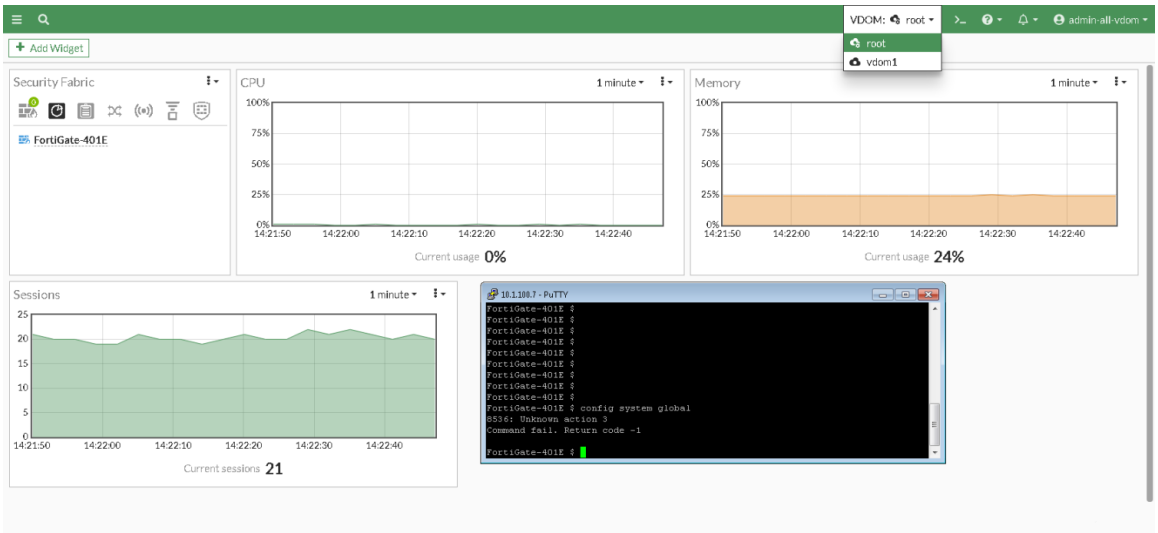
```
config system admin
    edit "remote-admin"
        set remote-auth enable
        set accprofile "admin_no_access"
        set vdom "root" "vdom1"
        set wildcard enable
        set remote-group "remote-tacacs"
        set accprofile-override enable
        set vdom-override enable
    next
end
```

To verify the configuration:

1. Log in as admin-vdom1 using a browser and SSH. The following behavior is expected:
   - The user can only access vdom1 (returned by TACACS+ in the `vdom` VSA).
   - The user can view firewall policies, but they cannot not create new policies.
   - The user cannot run `diagnose debug application` commands in the PuTTY SSH session.



2. Log in as admin_all_vdom using a browser and SSH. The following behavior is expected:
   - The user has no VSA VDOM configured on the TACACS+ server, so the default setting in the `system admin` configuration should apply. The user can access the root and vdom1 VDOMs.
   - The user has no access to `system global` in the CLI, and the prompt symbol is a `$` instead of a `#`.



Administrator profiles

Administrator profiles define what the administrator can do when logged into the FortiGate. When you set up an administrator account, you also assign an administrator profile which dictates what the administrator sees. Depending on the nature of the administrator's work, access level or seniority, you can allow them to view and configure as much or as little as is required. Access to CLI diagnose commands can also be disabled for global and VDOM level administrators.

By default, the FortiGate has an *admin* administrator account that uses the *super_admin* profile.

Access to FortiAI

Administrator profiles control administrator access to FortiAI. FortiAI calls the FortiOS REST APIs with the same permissions as the logged in administrator.

To configure FortiAI access in administrator profiles in the GUI:

1. Go to *System* > *Admin Profiles*, and double-click an administrator profile to open it for editing.
   Of the default administrator profiles provided with FortiOS, you can edit the setting in the *prof_admin* profile but not the *super_admin* profile.
2. Enable/disable *Allow using AI Assistant* and click *OK*.
   When disabled, the FortiAI icon remains visible in the FortiOS banner for administrators; however, administrators cannot submit queries and receive answers.

To configure FortiAI access in admin profiles in the CLI:

```
config system accprofile
    edit "admin-withAI"
        set comments ''
        set secfabgrp read-write
        ...
        set gui-ai-assistant enable
    next
end
```

super_admin profile

This profile has access to all components of FortiOS, including the ability to add and remove other system administrators. For certain administrative functions, such as backing up and restoring the configuration, super_admin access is required. To ensure that there is always a method to administer the FortiGate, the super_admin profile cannot be deleted or modified.

Lower level administrator profiles cannot back up or restore the FortiOS configuration.

The super_admin profile is used by the default admin account. It is recommended that you add a password and rename this account once you have set up your FortiGate. In order to rename the default account, a second admin account is required.

Creating customized profiles

To create a profile in the GUI:

1. Go to *System* > *Admin Profiles* and click *Create New*.
2. Configure the following settings:

Name	Enter a name for the profile.
Access Permissions	Set the access permissions for this profile.
Permit usage of CLI commands	Enable/disable usage of CLI commands for this profile. Select *Custom* to enable/disable specific commands.
Override theme	Configure a GUI theme for the profile. • *Inherit from system*: Use the GUI theme selected on *System* > *Settings* > *Preference* pane. • *Customize*: Select a GUI theme for this profile. You can also create a custom GUI theme. See Setting GUI color themes NEW on page 3827.
Allow using AI Assistant	Enable/disable use of the AI assistant for this profile.

3. Click *OK*.

To create a profile in the CLI:

```
config system accprofile
    edit <name>
        set secfabgrp {none | read | read-write}
        set ftviewgrp {none | read | read-write}
        set authgrp {none | read | read-write}
        set sysgrp {none | read | read-write | custom}
        set netgrp {none | read | read-write | custom}
        set loggrp {none | read | read-write | custom}
        set fwgrp {none | read | read-write | custom}
        set vpngrp {none | read | read-write}
        set utmgrp {none | read | read-write | custom}
        set wanoptgrp {none | read | read-write}
        set wifi {none | read | read-write}
        set admintimeout-override {enable | disable}
        set cli-diagnose {enable | disable}
        set cli-get {enable | disable}
        set cli-show {enable | disable}
        set cli-exec {enable | disable}
        set cli-config {enable | disable}
        set gui-theme-type {predefined | custom}
        set gui-theme <string>
    next
end
```

The CLI profile configuration includes additional options to allow usage of the `get`, `show`, `execute`, and `config` CLI commands.

Many diagnostic commands have privileged access. As a result, using them could unintentionally grant unexpected access or cause serious problems, so understanding the risks involved is crucial.

Controlling CLI system permissions

Administrator profiles can control administrator access to CLI commands based on role, access level, or seniority.

To configure CLI command access in administrative profiles:

```
config system accprofile
    edit <name>
        set cli-diagnose {enable | disable}
        set cli-get {enable | disable}
        set cli-show {enable | disable}
        set cli-exec {enable | disable}
        set cli-config {enable | disable}
    next
end
```

This command allows the administrator to configure the administrator profiles by enabling specific CLI commands as needed. The default setting for all the CLI command options is `disable`.

Displaying execute commands for custom system permissions

A custom access profile can have customized system permissions. In this example, a profile is created for maintenance read access, and the profile is applied to a new system administrator account. Once the administrator logs in, they can view the available execute commands by entering `execute ?` in the CLI.

To create the profile:

1. Configure the access profile:

```
config system accprofile
    edit "mnt test"
        set sysgrp custom
        config sysgrp-permission
            set mnt read
        end
    next
end
```

2. Configure the system administrator account:

```
config system admin
    edit "mnt"
        set accprofile "mnt test"
        set vdom "root"
        set password ********
    next
end
```

To display the list of the execute commands:

```
$ execute ?
backup                backup
fctems                fctems
ping                  PING command.
ping-options          ping-options
ping6                 PINGv6 command. [Take 0-100 arg(s)]
ping6-options         ping6-options
ssh-options           SSH options.
ssh6-options          IPv6 SSH options.
telnet-options        telnet-options
traceroute            Traceroute {IP|hostname}.
traceroute-options    traceroute-options
tracert6              Traceroute for IPv6. [Take 0-32 arg(s)]
usb-device            usb-device
```

```
usb-disk              usb-disk
vm-license-options    VM license options.
```

The output will vary based on the FortiGate model. A FortiGate VM is used in this example. For more information about using the CLI, see CLI basics on page 76.

Editing profiles

To edit a profile in the GUI:

1. Go to *System* > *Admin Profiles*.
2. Select the profile to be edited and click *Edit*.
3. Make the required changes.
4. Click *OK* to save any changes.

To edit a profile in the CLI:

```
config system accprofile
    edit "sample"
        set secfabgrp read
    next
end
```

Deleting profiles

To delete a profile in the GUI:

1. Go to *System* > *Admin Profiles*.
2. Select the profile to be deleted and click *Delete*.
3. Click *OK*.

To delete a profile in the CLI:

```
config system accprofile
    delete "sample"
end
```

Firmware & Registration

The *Firmware & Registration* page allows administrators to manage the firmware running on all supported devices, including FortiGate, FortiAP, FortiSwitch, and FortiExtender devices. You can upgrade individual devices or all devices. Supports Security Fabric and non-Security Fabric devices.

From the *Firmware & Registration* page, administrators can also authorize, deauthorize, and register devices.

The *Firmware & Registration* page displays a summary of devices that includes:

- Total number of devices and how many devices of each type
- Upgrade status of all devices
- Device name
- Device status
- Registration status
- Firmware version and maturity level
- Upgrade status

If the *Firmware & General Updates* (FMWR) license is invalid or the FortiGate appliance has reached End of Support (EOS), the FortiGate will enforce an automatic update to the latest patch of the current minor version. A valid FMWR license is required to update to the next minor or major version; without it, the FortiGate cannot be upgraded to these versions. See Required firmware upgrades for FortiGates with invalid support contracts or that have reached EOES on page 3762.

From the *Firmware & Registration* page, you can:

Upgrade all	Click *Upgrade all* and choose whether to upgrade all devices, extension devices, FortiAPs, FortiSwitches, or FortiExtenders. Available for Security Fabric and non-Security Fabric devices. See Upgrading all devices on page 3752.
Upgrade	Select one or more FortiGate, FortiAP, FortiSwitch, or FortiExtender devices to display and click the *Upgrade* button. Available for Security Fabric and non-Security Fabric devices. See Upgrading individual devices on page 3749.
Register	Select one or more devices to display and click the *Register* button. Use to register selected devices to FortiCare. See Registering devices on page 3755. The FortiGate, and then its service contract, must be registered to have full access to Fortinet Customer Service and Support, and FortiGuard services. The FortiGate can be registered in either the FortiGate GUI or the FortiCloud support portal. The service contract can be registered from the FortiCloud support portal. See also Registering FortiGate on page 39.
Authorization	Select one or more devices to display and click the *Authorization* or *Deauthorize* buttons. Use to authorize, deauthorize, or reject selected devices. See Authorizing devices on page 3773.

Reports	After a successful upgrade, view the firmware upgrade report. It compares statistics and configurations before and after a firewall upgrade. See Viewing firmware upgrade reports on page 3769.

Before upgrading FortiGate firmware, it is recommended to learn about firmware updates, firmware maturity levels, Special Technical Support (STS) firmware, and selected availability (SA) versions. See About firmware installations on page 3742 and Firmware labels on page 3743.

This section also includes the following topics:

- Enabling automatic firmware upgrades on page 3757
- Firmware upgrade notifications on page 3775
- Downloading a firmware image on page 3775
- Testing a firmware version on page 3776
- Installing firmware from system reboot on page 3777
- Restoring from a USB drive on page 3780
- Using controlled upgrades on page 3781
- Downgrading individual device firmware on page 3781
- Downloading the EOS support package for supported Fabric devices on page 3784
- How the FortiGate firmware license works on page 3787
- Enhanced firmware upgrade management for extension devices on page 3791

About firmware installations

Fortinet periodically releases new FortiGate firmware to include new features and resolve important issues. After successful registration of your FortiGate unit, firmware updates are available from FortiGuard and from the Fortinet Customer Service & Support website. See Registering FortiGate on page 39 for more information.

If the *Firmware & General Updates* (FMWR) license is invalid or the FortiGate appliance has reached End of Support (EOS), the FortiGate will enforce an automatic update to the latest patch of the current minor version. A valid FMWR license is required to update to the next minor or major version; without it, the FortiGate cannot be upgraded to these versions. See Required firmware upgrades for FortiGates with invalid support contracts or that have reached EOES on page 3762.

Installing a new firmware image replaces the current antivirus and attack definitions, along with the definitions included with the firmware release that is being installing. After you install new firmware, make sure that the antivirus and attack definitions are up to date.

It is recommended to back up your configuration before making any firmware changes. You will be prompted to back up your configuration as part of the upgrade process. See also Configuration backups and reset on page 4053.

Before you install any new firmware, follow the below steps:

1. Understand the maturity level of the current and target firmware releases to help you determine whether to upgrade. See Firmware maturity levels on page 3743. See also Selected availability (SA) versions on page

3748.

2. Review the Release Notes for a new firmware release.
3. Review the Supported Upgrade Paths.
4. Download a copy of the currently installed firmware, in case you need to revert to it. See Downloading a firmware image on page 3775 and Downgrading individual device firmware on page 3781 for details.
5. Have a plan in place in case there is a critical failure, such as the FortiGate not coming back online after the update.

   This could include having console access to the device (Connecting to the CLI on page 73), ensuring that your TFTP server is working (Installing firmware from system reboot on page 3777), and preparing a USB drive (Restoring from a USB drive on page 3780).
6. Back up the current configuration, including local certificates. The upgrade process prompts you to back up the current configuration. See also Configuration backups and reset on page 4053 for details.
7. Test the new firmware until you are satisfied that it applies to your configuration. See Testing a firmware version on page 3776 and Using controlled upgrades on page 3781 for details.

Installing new firmware without reviewing release notes or testing the firmware may result in changes to settings and unexpected issues.

Only FortiGate admin users and administrators whose access profiles contain system read and write privileges can change the FortiGate firmware.

Firmware labels

Fortinet uses the following labels to communicate important information about the version of FortiOS firmware running on FortiGate:

Maturity level	Identifies whether the firmware includes new, major features in addition to bug fixes and vulnerability patches. See Firmware maturity levels on page 3743.
Special Technical Support	Identifies a non-GA build of firmware issued by Fortinet support. See Special Technical Support firmware on page 3746.
Selected Availability	Identifies a special build of firmware issued by Fortinet to use for a long time. See Selected availability (SA) versions on page 3748.

Firmware maturity levels

Starting with FortiOS 7.2.0, released FortiOS firmware images use tags to indicate the following maturity levels:

- The *Feature* tag indicates that the firmware release includes new features. It can also include bug fixes and vulnerability patches where applicable.
- The *Mature* tag indicates that the firmware release includes no new, major features. Mature firmware will contain bug fixes and vulnerability patches where applicable.

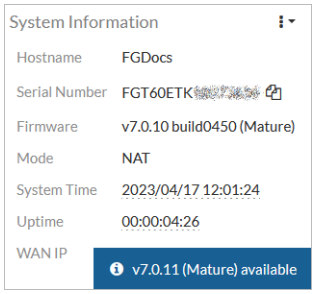
Administrators can use the tags to identify the maturity level of the current firmware in the GUI or CLI.

Administrators can view the maturity level of each firmware image that is available for upgrade on the *Firmware & Registration* page. When upgrading from mature firmware to feature firmware, a warning message is displayed.

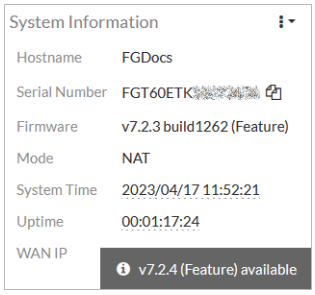
To view maturity levels for firmware in the GUI:

1. Go to *Dashboard > Status*. The *Firmware* field in the *System Information* widget displays the version with build number and either *(Mature)* or *(Feature)*.

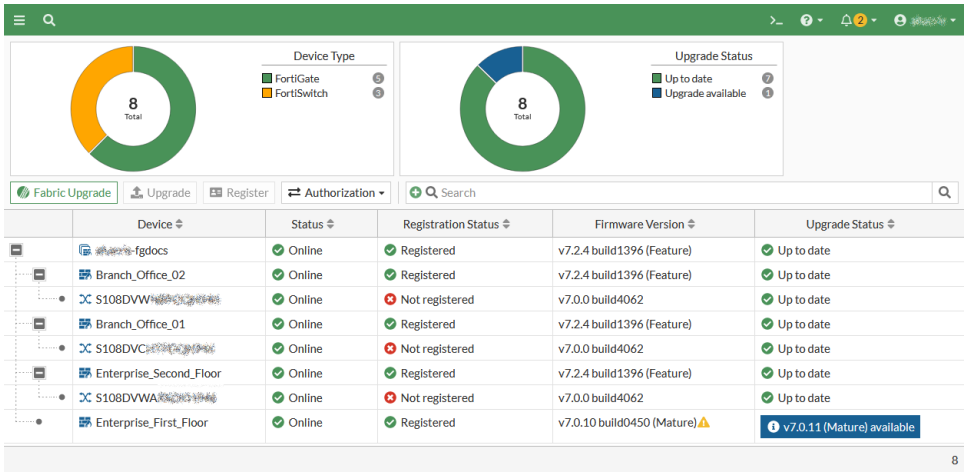
   The following is an example of firmware with the *(Mature)* tag:



   The following is an example of firmware with the *(Feature)* tag:

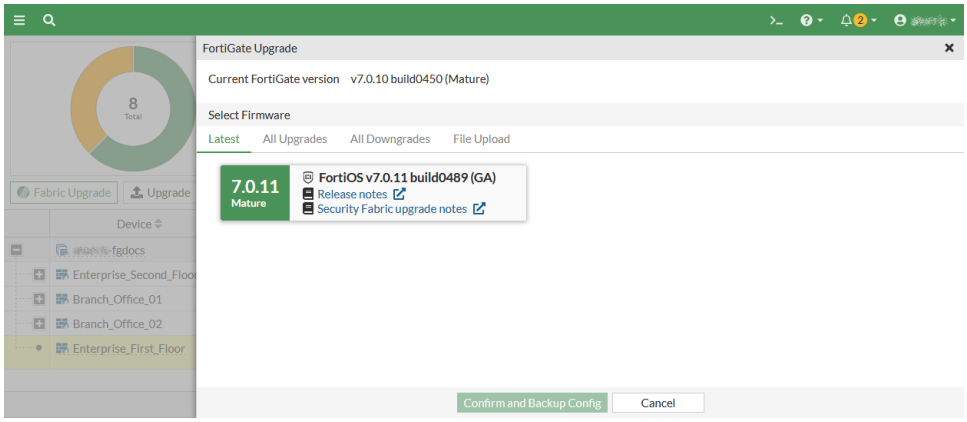


2. Go to *System > Firmware & Registration*. The *Firmware Version* column displays the version with build number and either *(Mature)* or *(Feature)*.



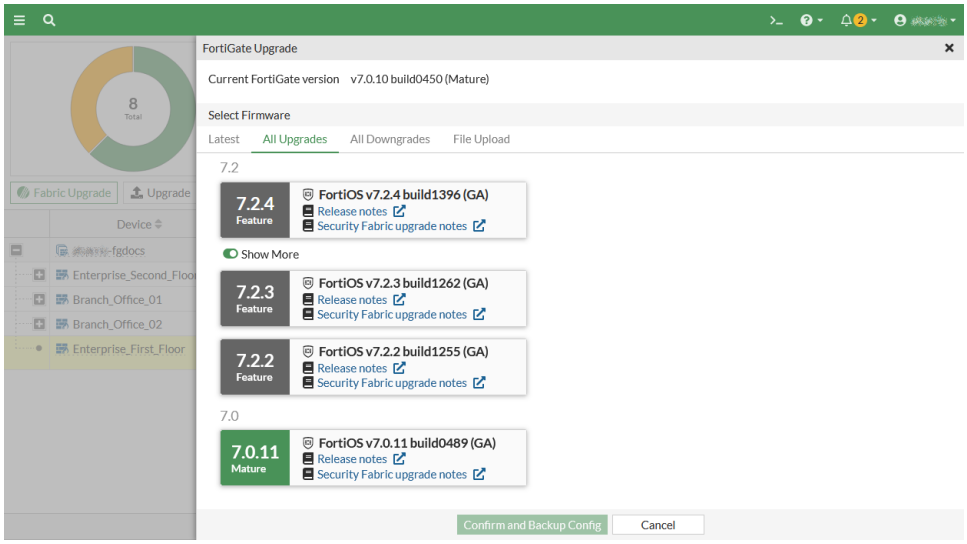
3. Select a device and click *Upgrade*. The *FortiGate Upgrade* pane is displayed.

- Click the *Latest* tab to view the latest available firmware version with its maturity level.



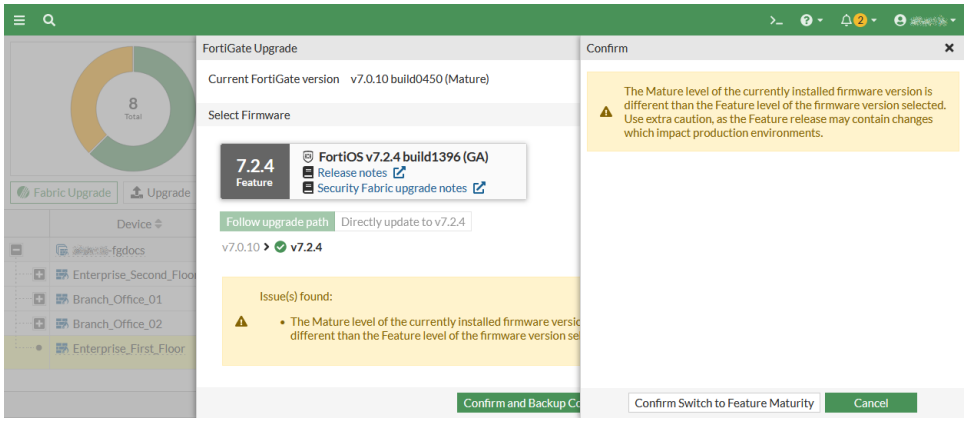
- Click the *All Upgrades* tab to view all available firmware versions and their maturity levels.

  A gray box around the version number and the label *Feature* identifies feature firmware version. A green box around the version with the label *Mature* identifies a mature firmware version.



4. Select a version and click *Confirm and Backup Config*.

   When the firmware version is a feature release, a warning is displayed; click *Confirm Switch to Feature Maturity*.



To view maturity levels for firmware in the CLI:

In this example, the `Version` field includes `.F` to indicate that the maturity level is feature:

```
# get system status
Version: FortiWiFi-81F-2R-POE v7.2.4,build1396,230131 (GA.F)
...
```

In this example, the `Version` field includes `.M` to indicate that the maturity level is mature:

```
# get system status
Version: FortiWiFi-81F-2R-POE v7.0.10,build0450,230221 (GA.M)
...
```

Special Technical Support firmware

Special Technical Support firmware was formerly known as Top3 builds. When Special Technical Support (STS) firmware is running on FortiGate instead of General Availability (GA) firmware, it is labeled as STS in the FortiOS GUI and CLI, and warning messages about the risks are displayed. STS builds are signed by Fortinet.

This example shows how to use the FortiOS GUI and CLI to identify when FortiGate is running an STS build of firmware.

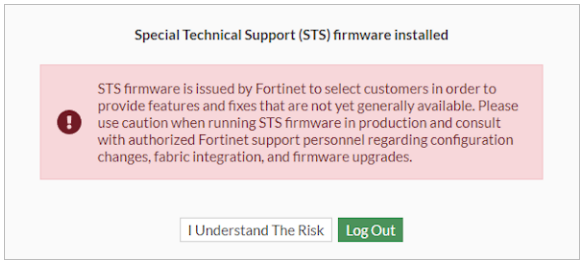
The STS firmware warning message does not apply to FIPS-CC certified builds.

FIPS-CC certified builds (non-GA) are categorized as STS builds, which means that the warning message might be displayed upon login even though, unlike STS builds, FIPS-CC certified builds are available to all customers and the warning does not apply.

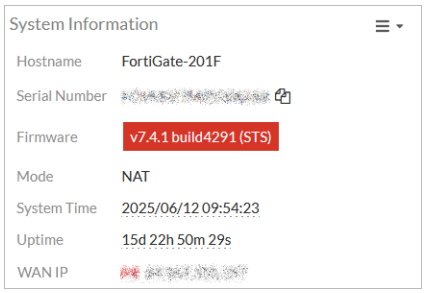
See also BIOS-level signature and file integrity checking on page 4005.

To view an STS build in the GUI:

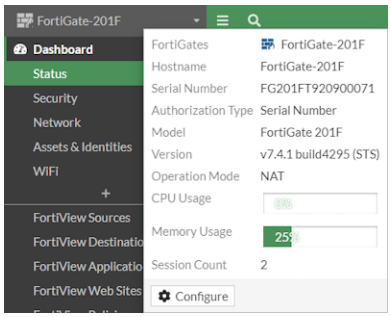
1. Log in to FortiOS. A warning message is displayed.



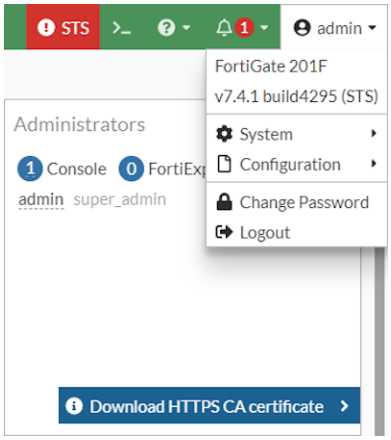
2. Click *I Understand The Risk* to acknowledge the warning and complete the login process.
3. View the STS build label and warnings:
   - Go to *Dashboard > Status > System Information* widget to view the red (STS) label in the *Firmware* field, for example, *7.4.1 build4291 (STS)*.



- Hover over the FortiGate name on the left corner of the banner to display a tooltip. The *Version* fields displays *(STS)*, for example, *7.4.1 build4295 (STS)*.



- Click *admin* on the top-right of the banner to display a menu. The *(STS)* label appears at the end of the version, for example, *7.4.1 build4295 (STS)*.



- Hover over the exclamation mark beside *STS* on the top-right of the banner. A *Special Technical Support (STS) firmware installed* warning is displayed.

To view an STS build in the CLI:

1. Log in to the CLI. The console prints the following warning message:

```
Login: admin
Password
******WARNING: This is a Special Technical Support (STS) firmware.******
STS firmware is issued by Fortinet to select customers in order to provide features and fixes
that are not yet generally available. Plesae use caution when running STS firmware in
```

```
production and consult with authorized Fortinet support personnel regarding configuration
changes, fabric integration, and firmware upgrades.

Welcome!
```

2. Get the system status.
   In this example, the STS build is identified by (STS) and a certified firmware signature.

```
# get system status
Version: FortiGate-201F v7.4.1,build4295,230817 (STS)
Security Level: 2
Firmware Signature: certified
...
```

Selected availability (SA) versions

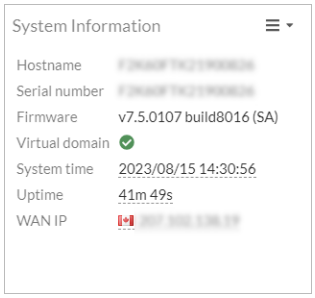
A selected availability (SA) version and label identifies special builds that are provided to customers to use for a long time. The SA version uses an odd number as the minor version and a four digit number for the patch version. The SA version and label are visible in the GUI and CLI.

SA builds are dual-signed by the Fortinet CA and a third-party CA.

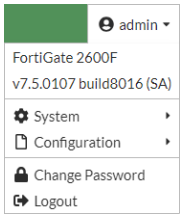
In the following example, special build 0107 is based on FortiOS 7.4.0 build 8016 and is labeled *v7.5.0107 build8016 (SA)*.

To view the SA version and label in the GUI:

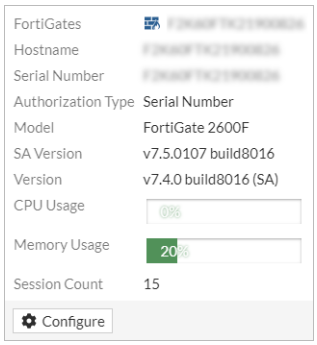
1. Go to *Dashboard* > *Status* > *System Information*. The *Firmware* option displays the SA version and label of *v7.5.0107 build8016 (SA)*.



2. On the top-right of the banner, click *<administrator name>*, such as *admin*. The SA version and label is displayed.



3. On the top-left corner of the banner, click the FortiGate name. A tooltip displays the SA version and label.



To view the SA version and label in the CLI:

```
# get system  status
Version: FortiGate-2600F v7.4.0,build8016,230711 (SA)
SA Version: v7.5.0107,build8016
Security Level: 0
Firmware Signature: certified
...
```

The `SA Version` is displayed as `v7.5.0107, build8016`.

Upgrading individual devices

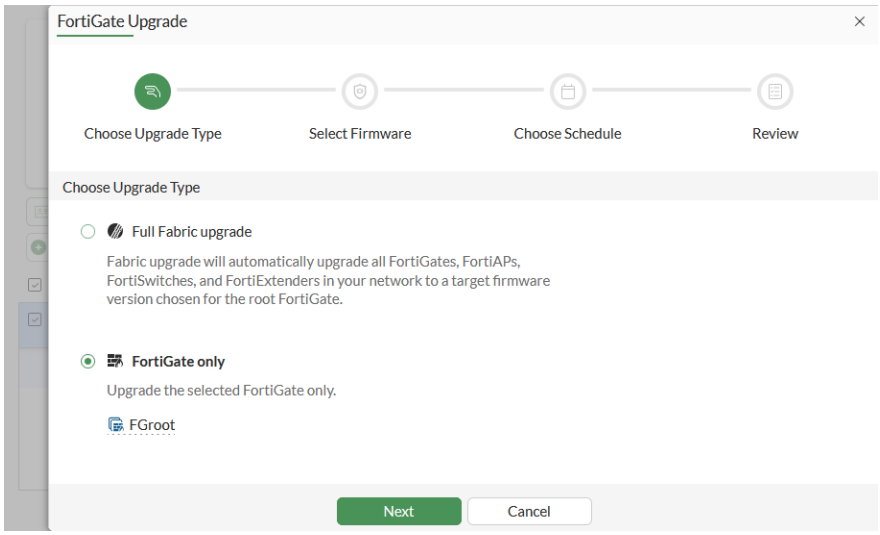
On the *System > Firmware & Registration* page, select one or more supported devices to display the *Upgrade* button. Supported devices include FortiGate, FortiAP, FortiExtender, or FortiSwitch. Click the *Upgrade* button to open the *FortiGate Upgrade* wizard and complete the following steps:

- Choose Upgrade Type
- Select Firmware
- Choose Schedule
- Review

To upgrade individual device firmware in the GUI:

1. Log in to the FortiGate GUI as an administrative user.
2. Go to *System > Firmware & Registration*.
3. Select one or more devices, and click *Upgrade* to open the *FortiGate Upgrade* wizard.
   The following devices and scenarios are supported:
   - Select one or more FortiGate devices.
   - Select one or more FortiAP devices.
   - Select one or more FortiExtender devices.
   - Select one or more FortiSwitch devices.
   - Select a mix of FortiAP, FortiExtender, and/or FortiSwitch devices.

   When upgrading non-FortiGate devices, the wizard opens at the *Select Firmware* step.
4. If the *Choose Upgrade Type* step is displayed, select *FortiGate only* then click *Next*.



5. For the *Select Firmware* step:
   a. Select firmware from one of the following tabs:

Recommended	Displays the recommended firmware from FortiGuard. When FortiGate is running the latest firmware from FortiGuard, the following message is displayed: *The firmware is up to date*.
All Upgrades	Displays all available firmware from FortiGuard. When FortiGate is running the latest firmware from FortiGuard, the following message is displayed: *No upgrades available*.
All Downgrades	Displays all available firmware downgrades from FortiGuard. Downgrading the firmware is not recommended. See Downgrading individual device firmware on page 3781.
File Upload	Upload a firmware file that you previously downloaded from the Fortinet Customer Service & Support website. See Downloading a firmware image on page 3775.

   b. If the selected firmware version spans multiple builds in the upgrade path, choose one of the following options:

Follow upgrade path	Automatically upgrade FortiGate to each firmware in the upgrade path before upgrading to the selected version. Recommended. FortiGate automatically completes the upgrades, including any required reboots, by downloading the chosen firmware directly from FortiGuard.
Directly update to v<version and build number>	Bypass the upgrade path to immediately upgrade FortiGate to the selected firmware. A warning message is displayed: *Upgrading to FortiOS v<version and build number> directly may result in the loss of configuration*.

   Information about the selected firmware is displayed for you to consider. For example, when upgrading from mature firmware to feature firmware, a warning message appears about the maturity level of the selected firmware for the upgrade. See Firmware maturity levels on page 3743 for more information.

c. Click *Next*.

6. For the *Choose Schedule* step:
   a. Choose a schedule:

Immediate	Select to start the upgrade immediately after completing the *FortiGate Upgrade* wizard.
Specify	Select to schedule a date and time to start the upgrade after completing the *FortiGate Upgrade* wizard.

> Scheduling an upgrade using a manually uploaded file is only available for FortiGate devices that have physical disk storage.
> - If the FortiGate has a disk, *File Upload* upgrades can be scheduled for it and its extension units both in the GUI and using the Security Fabric.
> - If the FortiGate does not have a disk, *File Upload* upgrades cannot be scheduled for it or its extension units.

   b. Click *Next* to proceed to the *Review* step.
7. For the *Review* step, review the upgrade plan, and click *Confirm and Backup Config* to proceed. The *Confirm* dialog box is displayed.
8. Click *Yes* to start the upgrade. The *Important* dialog box is displayed.
9. Read the information and click *Close*.
10. Use the *Firmware Upgrade* tray to monitor upgrade progress.
    The FortiGate unit backs up the current configuration to the management computer, uploads the firmware image file, upgrades to the new firmware version, and restarts. This process takes a few minutes.

To upgrade individual device firmware in the CLI:

1. Make sure that the TFTP server is running.
2. Copy the new firmware image file to the root directory of the TFTP server.
3. Log in to the CLI.
4. Ping the TFTP server to ensure that the FortiGate can connect to it:

```
# execute ping <tftp_ipv4>
```

5. Enter the following command to copy the firmware image from the TFTP server to the FortiGate unit:

```
# execute restore image tftp <filename> <tftp_ipv4>
```

   The FortiGate unit responds with the message:

```
This operation will replace the current firmware version!
Do you want to continue? (y/n)
```

6. Type y. The FortiGate unit uploads the firmware image file, verifies the signature of the firmware image, and determines the firmware maturity level.
   When you are upgrading to a feature firmware image, you are asked to confirm whether to continue with the upgrade.

When you proceed with the upgrade, the upgrade image is installed and FortiGate restarts. This process takes a few minutes.

```
Please wait...
Connect to tftp server 172.16.200.55 ...
#################################################################################
Get image from tftp server OK.
Verifying the signature of the firmware image.

Warning: Upgrading to an image with Feature maturity notation.
Image file uploaded is marked as a Feature image, are you sure you want to upgrade?
Do you want to continue? (y/n)y
Please confirm again. Are you sure you want to upgrade using uploaded file?
Do you want to continue? (y/n)y
Checking new firmware integrity ... pass
Please wait for system to restart.
Firmware upgrade in progress ...
Done.
The system is going down NOW !!
```

7. Reconnect to the CLI.
8. Update the antivirus and attack definitions:

```
# execute update-now
```

Upgrading all devices

On the *System > Firmware & Registration* page, use the *Upgrade all* button to upgrade firmware on all supported devices. Supported devices include FortiGate, FortiAP, FortiExtender, and FortiSwitch, and the option is available for Security Fabric and non-Security Fabric devices.

Click the *Upgrade all* button and choose what device types to upgrade. The *FortiGate Upgrade* wizard opens to help you complete the following steps:

- Choose Upgrade Type
- Select Firmware
- Choose Schedule
- Review

During a full Fabric upgrade, FortiGate upgrades devices by groups in the following order:

1. PoE PD (Power over Ethernet Powered Devices)
2. PoE PSE (Power Source Equipment) and non-PoE devices
3. FortiGate itself

Group 2 (PoE PSE and non-POE devices) waits until group 1 (PoE PD) finishes upgrading to new firmware before starting its upgrade.

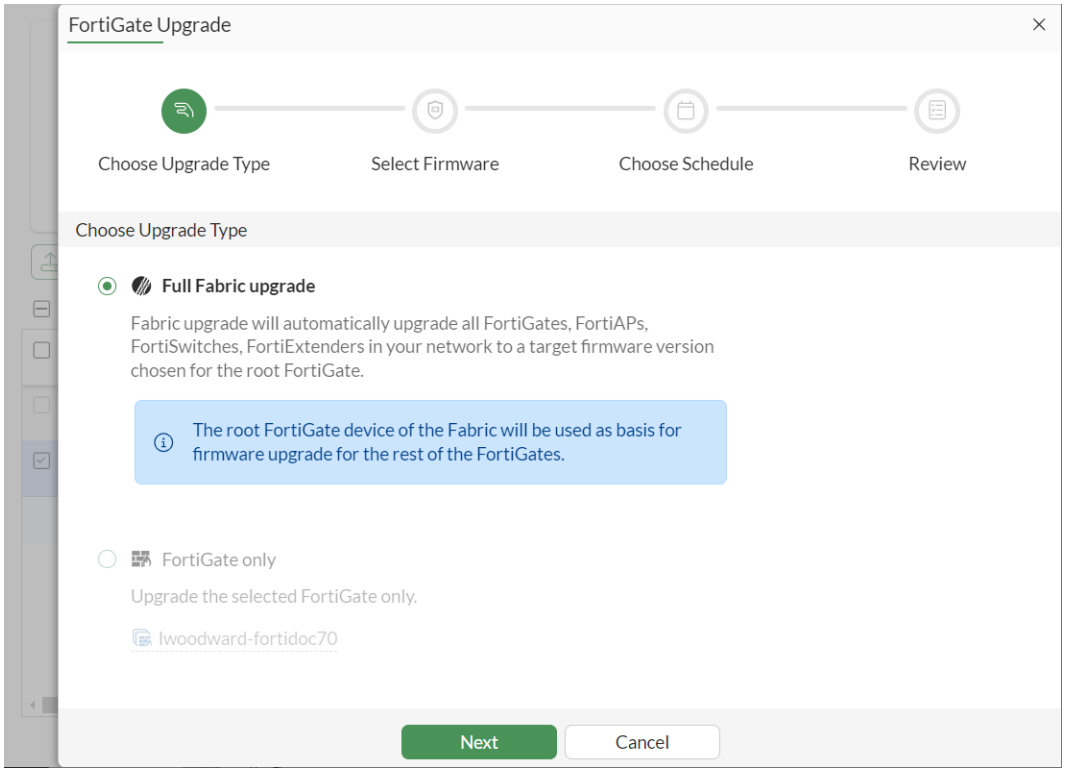
Once all upgrades are complete, and the FortiGate is back up, it verifies that all devices are using the new firmware version before labeling the upgrade as done.

To upgrade all device firmware in the GUI:

1. Log in to the FortiGate GUI as an administrative user.
   When you are upgrading the Security Fabric, you must log in to the root FortiGate.
2. Go to *System > Firmware & Registration*, click *Upgrade all*, and select one of the following options:

Upgrade all > Devices	Upgrades all devices, including FortiGate, FortiAP, FortiSwitch, and FortiExtender devices.
Upgrade all > Extension devices	Upgrades all FortiAP, FortiSwitch, and FortiExtender devices.
Upgrade all > FortiAPs	Upgrades all FortiAP devices.
Upgrade all > FortiExtenders	Upgrades all FortiExtender devices.
Upgrade all > FortiSwitches	Upgrades all FortiSwitch devices.

   The *FortiGate Upgrade* wizard opens to the *Choose Upgrade Type* step.



3. On the *Choose Upgrade Type* step, select *Full Fabric upgrade*, and click *Next* to proceed to the *Select Firmware* step.
4. For the *Select Firmware* step:

a. Select firmware from one of the following tabs:

Recommended	Displays the recommended firmware from FortiGuard. When the devices are running the latest firmware from FortiGuard, the following message is displayed: *The firmware is up to date*.
All Upgrades	Displays all available firmware from FortiGuard. When the devices are running the latest firmware from FortiGuard, the following message is displayed: *No upgrades available*.
All Downgrades	Displays all available firmware downgrades from FortiGuard. Downgrading the firmware is not recommended. See Downgrading individual device firmware on page 3781.

On managed FortiAP, FortiExtender, and FortiSwitch devices, the upgrade adheres to the respective compatibility matrix information maintained on the FortiGuard Distribution Network (FDN).

b. If the selected firmware version spans multiple builds in the upgrade path, choose one of the following options:

Follow upgrade path	Automatically upgrade to each firmware in the upgrade path before upgrading to the selected version. Recommended. FortiGate automatically completes the upgrades, including any required reboots, by downloading the chosen firmware directly from FortiGuard.
Directly update to v<version and build number>	Bypass the upgrade path to immediately upgrade FortiGate to the selected firmware. A warning message is displayed: *Upgrading to v<version and build number> directly may result in the loss of configuration*.

Information about the selected firmware is displayed for you to consider. For example, when upgrading from mature firmware to feature firmware, a warning message appears about the maturity level of the selected firmware for the upgrade. See Firmware maturity levels on page 3743 for more information.

c. Click *Next* to proceed to the *Choose Schedule* step.

5. For the *Choose Schedule* step:

a. Choose a schedule:

Immediate	Select to start the upgrade immediately after completing the *FortiGate Upgrade* wizard.
Specify	Select to schedule a date and time to start the upgrade after completing the *FortiGate Upgrade* wizard.

The configuration is backed up when the upgrade is scheduled, not when the upgrade occurs.

If configuration changes are made after the upgrade is scheduled but before the upgrade occurs, those changes will not be saved in the configuration backup.

Scheduling an upgrade using a manually uploaded file is only available for FortiGate devices that have physical disk storage.

- If the FortiGate has a disk, *File Upload* upgrades can be scheduled for it and its extension units both in the GUI and using the Security Fabric.
- If the FortiGate does not have a disk, *File Upload* upgrades cannot be scheduled for it or its extension units.

b. Click *Next* to proceed to the *Review* step.

6. For the *Review* step, review the upgrade plan, and click *Confirm and Backup Config* to proceed. The *Confirm* dialog box is displayed.
7. Click *Yes* to start the upgrade. The *Important* dialog box is displayed.
8. Read the information and click *Close*.
9. Use the *Firmware Upgrade* tray to monitor upgrade progress.
   The FortiGate unit backs up the current configuration to the management computer, uploads the firmware image file, upgrades to the new firmware version, and restarts. This process takes a few minutes.

To upgrade all device firmware in the CLI:

The following options are available in `execute federated-upgrade <option>`:

Option	Description
`cancel`	Cancel the currently configured upgrade.
`initialize`	Set up a federated upgrade.
`status`	Show the current status of a federated upgrade.
`restart`	Restart the currently configured federated upgrade.

The `config system federated-upgrade` command is read-only. Attempting to configure federated upgrade using the `config` command will show the following error message:

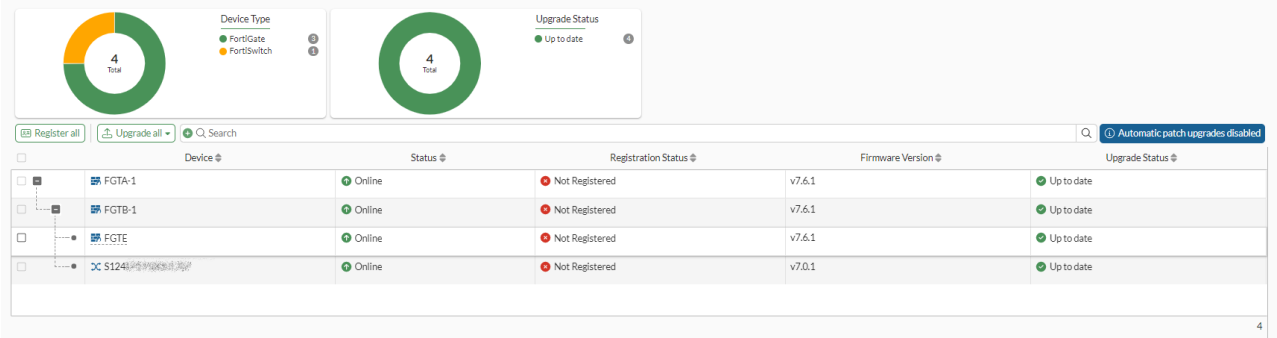
```
Federated upgrade cannot be configured directly.
Please use 'execute federated-upgrade ...' to configure.
```

Registering devices

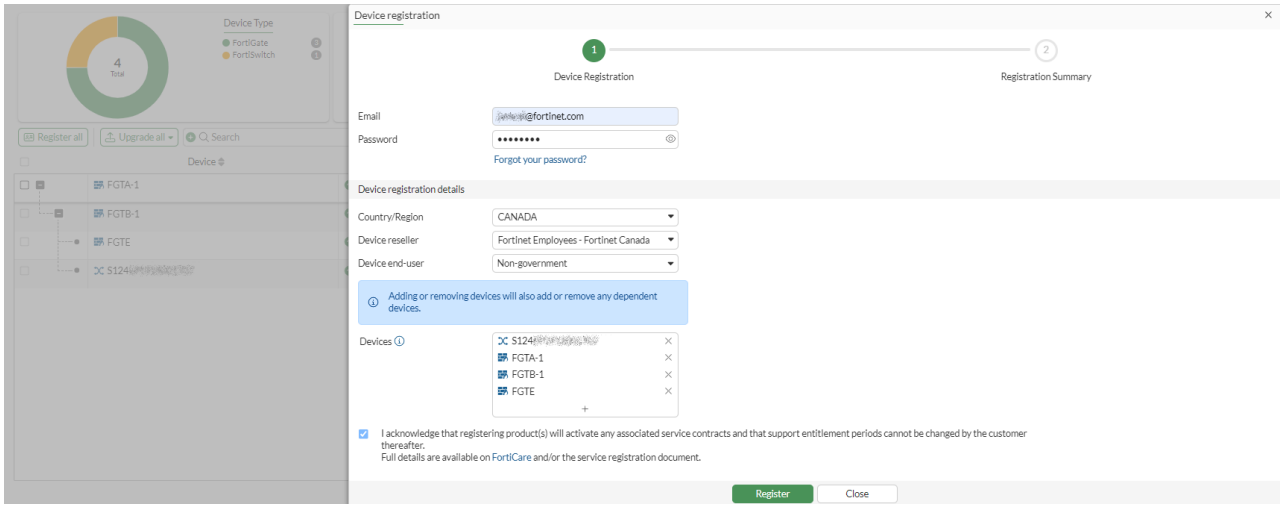
All Fortinet devices (FortiGates, FortiAPs, and FortiSwitches) that are in the same Security Fabric can be registered to the same FortiCare account at the same time, as opposed to having to register each device individually.

To register devices from the Firmware & Registration page:

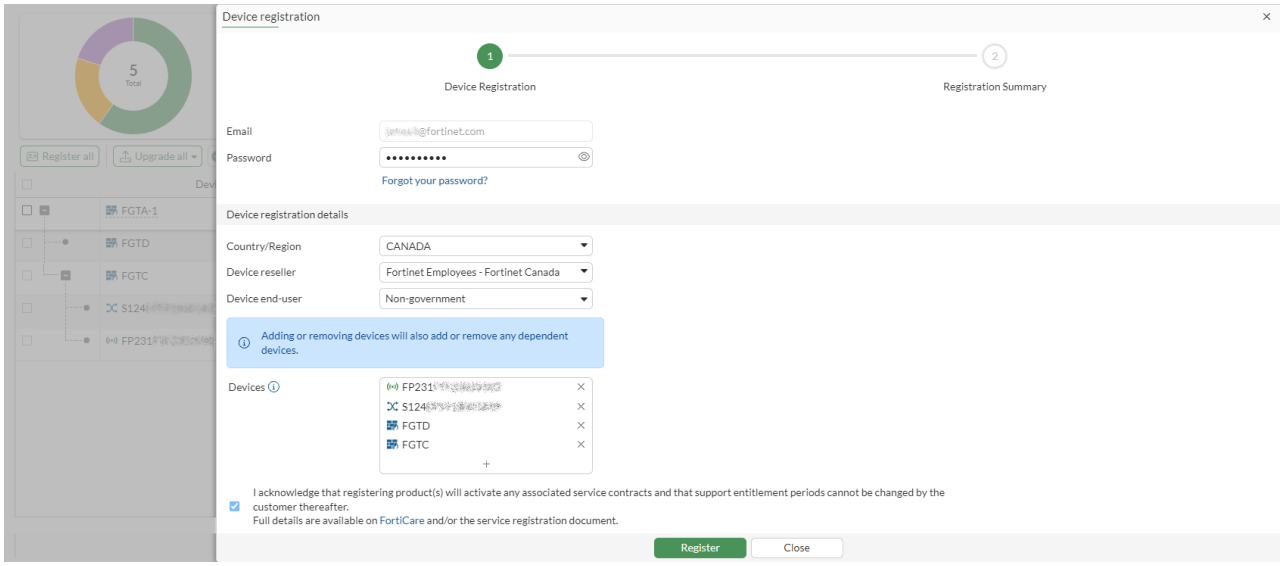
1. On the Fabric root FortiGate, go to *System* > *Firmware & Registration*. The *Registration Status* column shows whether or not a device has been registered to FortiCare.



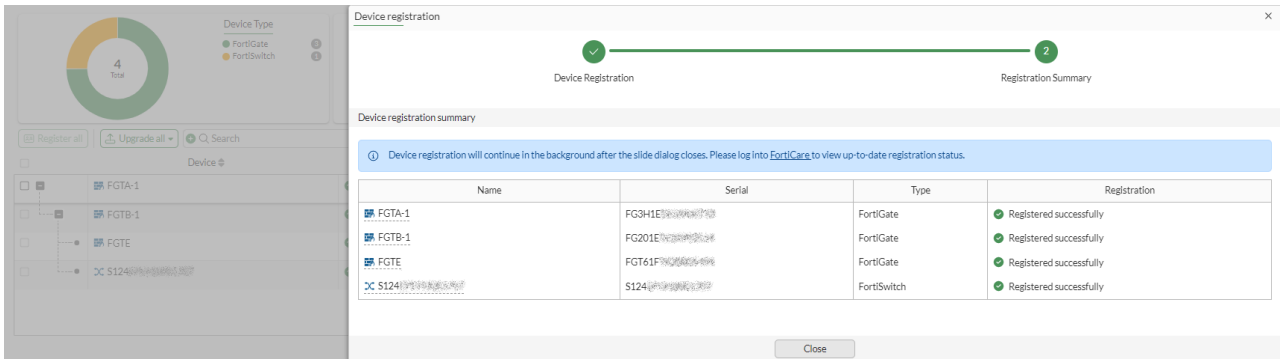
2. Click *Register All*.
3. Fill in the *Device registration* pane, adding all of the unregistered devices in the *Devices* field.



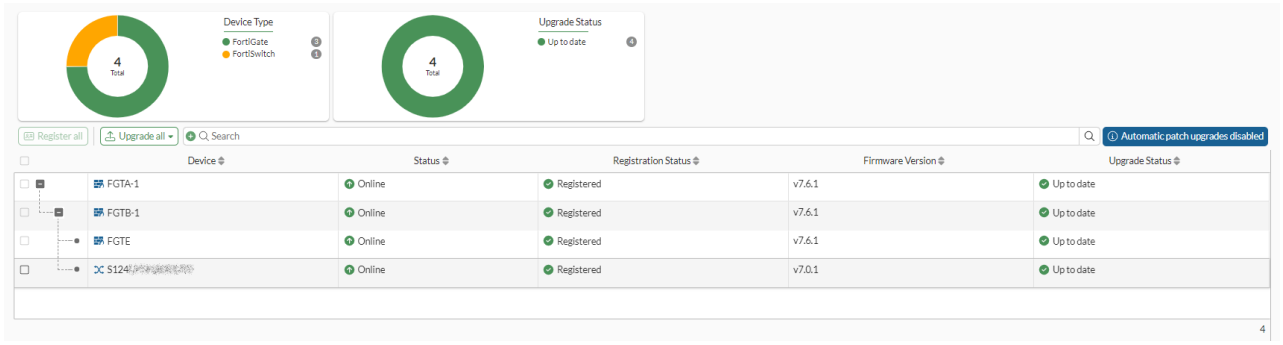
If the root FortiGate device is already registered, the *Email* address will already be set.



4. Click *Register*. A registration request for all of the selected devices is sent to FortiCare and the *Device registration summary* table is shown.



5. Click *Close*. The *Firmware & Registration* page shows the updated registration statuses.



Enabling automatic firmware upgrades

Automatic firmware upgrades are enabled by default on all FortiGate models, including FortiGate VMs, for increased device security. The FortiGate automatically upgrades when a new FortiOS patch release is available.

Automatic firmware upgrades can be configured from the *FortiGate Setup* wizard, the *System > Firmware & Registration* pane, or in the CLI with the following commands:

```
config system fortiguard
    set auto-firmware-upgrade {enable | disable}
    set auto-firmware-upgrade-day {sunday monday tuesday wednesday thursday friday saturday}
    set auto-firmware-upgrade-delay <integer>
    set auto-firmware-upgrade-start-hour <integer>
    set auto-firmware-upgrade-end-hour <integer>
end
```

`auto-firmware-upgrade {enable | disable}`	Enable/disable automatic patch-level firmware upgrades from FortiGuard.
`auto-firmware-upgrade-day {sunday monday tuesday wednesday thursday friday saturday}`	Enter the allowed day or days of the week to start the automatic patch-level firmware upgrade from FortiGuard.
`auto-firmware-upgrade-delay <integer>`	Enter the number of days to wait before automatically installing the automatic patch-level firmware upgrade from FortiGuard (default = 3).

`auto-firmware-upgrade-start-hour <integer>`	Set the start time of the designated time window for the automatic patch-level firmware upgrade from FortiGuard (in hours, 0 - 23, default = 2). The actual upgrade time is randomly selected in the time window. See Reviewing upgrade status on page 3759 for more information on confirming the scheduled upgrade time.
`auto-firmware-upgrade-end-hour <integer>`	Set the end time of the designated time window for the automatic patch-level firmware upgrade from FortiGuard (in hours, 0 - 23, default = 4). When this value it is smaller than the start time, it will be treated as the same time in the next day. The actual upgrade time is randomly selected in the time window. See Reviewing upgrade status on page 3759 for more information on confirming the scheduled upgrade time.

The `auto-firmware-upgrade-delay` and `auto-firmware-upgrade-day` commands are mutually exclusive. The `auto-firmware-upgrade-delay` command overrides the `auto-firmware-upgrade-day` command. Disable `auto-firmware-upgrade-delay` by setting it to zero if you would rather use the `auto-firmware-upgrade-day` command to select a day of the week for automatic installation, regardless of when the patch release is detected.

When enabled, FortiGates use the FortiGuard upgrade path to check FortiGuard for firmware updates within the same minor release. Checks are performed within a specified time period. When a new patch release is available, a firmware upgrade is scheduled.

After the patch release is successfully installed, an email is sent to the FortiCloud account that the FortiGate is registered to.

- Automatic firmware upgrades cannot be enabled for FortiGates that are part of a Security Fabric, managed by FortiManager, or serving as a secondary unit in a high availability (HA) pair. However, HA groups will still receive automatic firmware upgrades through the primary FortiGate. In such cases, the enforced upgrade proceeds as intended on the primary unit, while the secondary FortiGate receives the upgrade through a cluster upgrade initiated by the primary, since automatic upgrades are disabled on secondary units.
- Automatic upgrades will only upgrade to a newer patch within that minor version. For example, a FortiOS version 7.2.x image will only auto-upgrade to another 7.2.x image. It will not upgrade to a 7.4.x image.

For information on special considerations when upgrading with an invalid FMWR license or a FortiGate at End of Support, see Required firmware upgrades for FortiGates with invalid support contracts or that have reached EOES on page 3762.

For FortiGates managed by FortiGate Cloud, automatic firmware patch may be enabled depending on the FortiGate Cloud version and portal in use. See the Administration Guide for the applicable FortiGate Cloud version and portal:

- Standard Portal Administration Guide
- 25.1.a Portal (Beta) Administration Guide
- Premium Portal Administration Guide

Reviewing upgrade status

The following commands can be used to review the status of the automatic upgrade.

The `diagnose test application forticldd 13` command lists when the most recent firmware image upgrade check occurred as well as when the next check is scheduled.

If the FortiGate is part of a Fabric or managed by FortiManager, the `Automatic image upgrade` option will be set to `disabled`.

```
# diagnose test application forticldd 13
...
Automatic image upgrade: disabled.
```

If a newer, valid firmware patch is detected, the `show sys federated-upgrade` command will list when the firmware upgrade will occur. The firmware upgrade schedule will depend on the configured automatic upgrade settings. If the settings are changed before the upgrade occurs, the image installation will be rescheduled to respect the new requirements.

The `show sys federated-upgrade` command also lists previous firmware upgrades.

The following debug commands are available for troubleshooting:

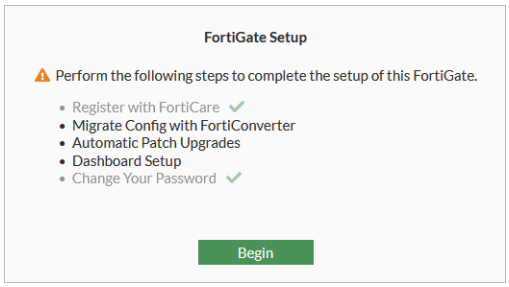
```
# diagnose debug en
# diagnose debug application forticldd -1
# diagnose debug application sfupgraded -1
```

Example

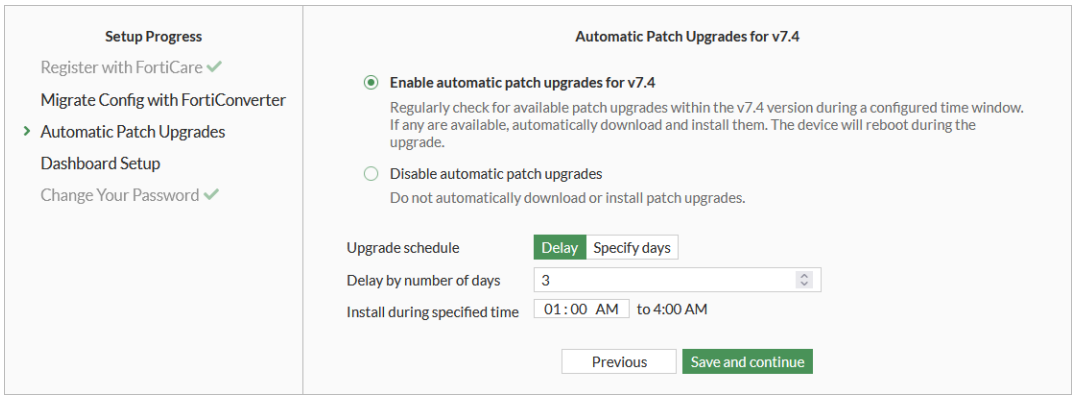
The following example demonstrates setting automatic firmware upgrades after a delay of three days.

To configure automatic firmware upgrades in the GUI:

1. Log in to the FortiGate GUI and click *Begin*.

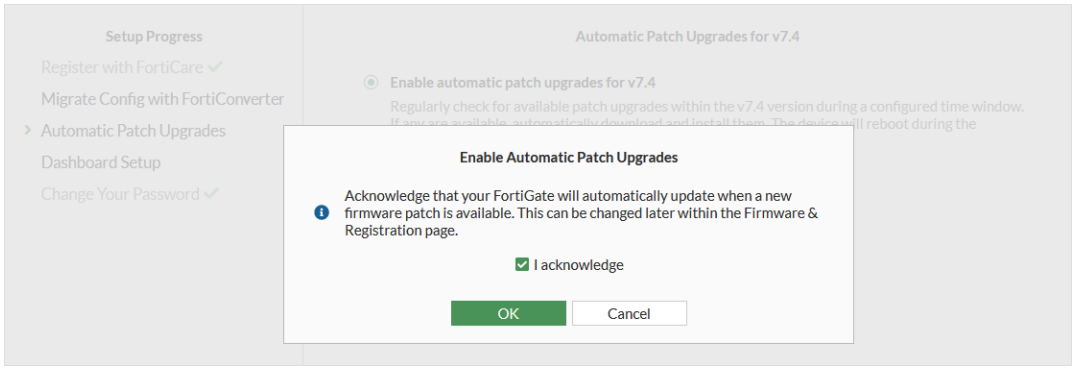


2. Select *Enable automatic patch upgrades for v7.4* (default setting).
3. Edit the upgrade and installation settings as needed (*Upgrade schedule*, *Delay by number of days*, *Install during specified time*), then click *Save and continue*.



If *Disable automatic patch upgrades* is selected, this can be changed later from the *System > Firmware & Registration* page by clicking the *Disable automatic patch upgrades* notification.

4. The *Enable Automatic Patch Upgrades* dialog opens. Select *I acknowledge* and click *OK* to proceed.



The FortiGate will be updated based on the configured schedule when a new patch is available.

5. An email is sent to alert the administrator that the firmware upgrade schedule has changed.

   Sample email after configuring automatic firmware upgrades:

```
From: DoNotReply@fortinet-notifications.com <DoNotReply@fortinet-notifications.com>
Sent: Tuesday, July 25, 2023 11:08 AM
To: ********** <*****@fortinet.com>
Subject: Automatic firmware upgrade schedule changed

date=2023-07-25 time=11:07:34 devid="FG81EPTK19000000" devname="FortiGate-81E-POE"
eventtime=1690308454221334719 tz="-0700" logid="0100032263" type="event" subtype="system"
level="notice" vd="root" logdesc="Automatic firmware upgrade schedule changed" user="system"
msg="System patch-level auto-upgrade regular check enabled."
```

6. Once a patch is detected, an email is sent to alert the administrator that a new image installation is scheduled.

   Sample email after a new image installation is scheduled:

```
From: DoNotReply@fortinet-notifications.com <DoNotReply@fortinet-notifications.com>
Sent: Friday, July 21, 2023 1:17 PM
To: ********** <*****@fortinet.com>
Subject: Automatic firmware upgrade schedule changed
```

```
date=2023-07-21 time=13:16:50 devid="FG81EPTK19000000" devname="FortiGate-81E-POE"
eventtime=1689970609076391174 tz="-0700" logid="0100032263" type="event" subtype="system"
level="notice" vd="root" logdesc="Automatic firmware upgrade schedule changed" user="system"
msg="System patch-level auto-upgrade new image installation scheduled between local time Sat
Jul 22 13:03:56 2023 and local time Sat Jul 22 14:00:00 2023."
```

7. After the image installation is completed, an email is sent to alert the administrator that the federated upgrade is complete.
   Sample email after the federated upgrade is complete:

```
From: DoNotReply@fortinet-notifications.com <DoNotReply@fortinet-notifications.com>
Sent: Wednesday, February 4, 2026 4:24 PM
To: ****** <******@fortinet.com>
Subject: Firmware Update Notification : FGT-device

An automated firmware update has completed successfully. The device FGT-device, serial number
FGVM<serial number>, was updated to firmware version 8.0.0 on 2026-02-04 16:23:56.

This is an informational notification and was generated by automation stitch configuration on
FGT-device.
```

   Sample event log after the federated upgrade is complete:

```
date=2026-02-04 time=16:23:56 id=7603170309633474560 itime="2026-02-04 16:23:57" euid=3
epid=104 dsteuid=3 dstepid=3 logflag=0 logver=800000123 logid=0100022094 type="event"
subtype="system" level="information" msg="Federated upgrade complete" logdesc="A federated
upgrade was completed by the root FortiGate" version="8.0.0" eventtime=1770251036796900232
tz="-0800" nodecount=1 fgtcount=1 devid="FGVM<serial number>" vd="root" csf="csf_97_vdom"
dtime="2026-02-04 16:23:56" itime_t=1770251037 devname="FGT-device"
```

To configure automatic firmware upgrades in the CLI:

1. Configure the automatic firmware upgrade schedule:

```
config system fortiguard
    set auto-firmware-upgrade enable
    set auto-firmware-upgrade-delay 3
    set auto-firmware-upgrade-start-hour 2
    set auto-firmware-upgrade-end-hour 4
end
```

   The FortiGate will perform a check between the start and end hours set for the firmware upgrade to review if there is an upgrade available.
2. Review the firmware upgrade check schedule:

```
# diagnose test application forticldd 13
Scheduled push image upgrade: no
Scheduled Config Restore: no
Scheduled Script Restore: no
Automatic image upgrade: Enabled.
        Next upgrade check scheduled at (local time) Thu Mar 29 03:10:56 2023
```

When an available patch upgrade is detected, the automatic firmware update will be scheduled based on the set upgrade delay.

Sample event log after a new patch upgrade is detected:

```
date=2023-03-29 time=03:10:56 eventtime=1679336380720695924 tz="-0700"
logid="0100032263" type="event" subtype="system" level="notice" vd="vdom1"
logdesc="Automatic firmware upgrade schedule changed" user="system"
msg="System patch-level auto-upgrade new image installation scheduled
     between local time Sat Apr 01 03:10:56 2023 and local time Sat Apr 01 04:00:00 2023."
```

3. Review the installation window of the new patch release:

```
# diagnose test application forticldd 13
Scheduled push image upgrade: no
Scheduled Config Restore: no
Scheduled Script Restore: no
Automatic image upgrade: Enabled.
        Next upgrade check scheduled at (local time) Mon Mar 30 03:10:56 2023
        New image 7.4.1b2305(07004000FIMG0021204001) installation is scheduled to
                start at Sat Apr 01 03:10:56:21 2023
                end by Sat Apr 01 04:00:00 2023
```

Once the firmware patch is successfully installed, an event log is created to track the change and an email is sent to the FortiCloud account under which the FortiGate is registered.

Sample event log after successfully updating firmware:

```
date=2023-04-01 time=03:13:04 devid="FG3H1E5819904039" devname="D"
eventtime=1679590383750408029 tz="-0700"
logid="0100022094" type="event" subtype="system" level="information" vd="vdom1"
logdesc="A federated upgrade was completed by the root FortiGate"
msg="Federated upgrade complete" version="7.4.1"
```

Required firmware upgrades for FortiGates with invalid support contracts or that have reached EOES

Since FortiOS 7.4.8 and 7.6.4, FortiGates are required to upgrade to the latest patch within their current minor version when:

- FortiGate has an invalid Firmware & General Updates (FMWR) license.
- FortiGate is running a minor FortiOS version that has reached end of engineering support (EOES).

This proactive measure ensures that all devices remain protected with the most up-to-date security features.

FortiGate checks FortiGuard daily for new patch upgrades. When a new patch is available, and the firmware license is invalid or the FortiGate firmware has reached EOES, FortiGate schedules a required upgrade to the new firmware image.

You cannot cancel required upgrades after they are scheduled; however, you can use the following method to adjust the installation schedule:

- Use the `execute auto-upgrade delay-installation` command to postpone the installation for a fixed 7-day period. This can be run an unlimited number of times. However, it cannot be run on the same day that the update is scheduled.

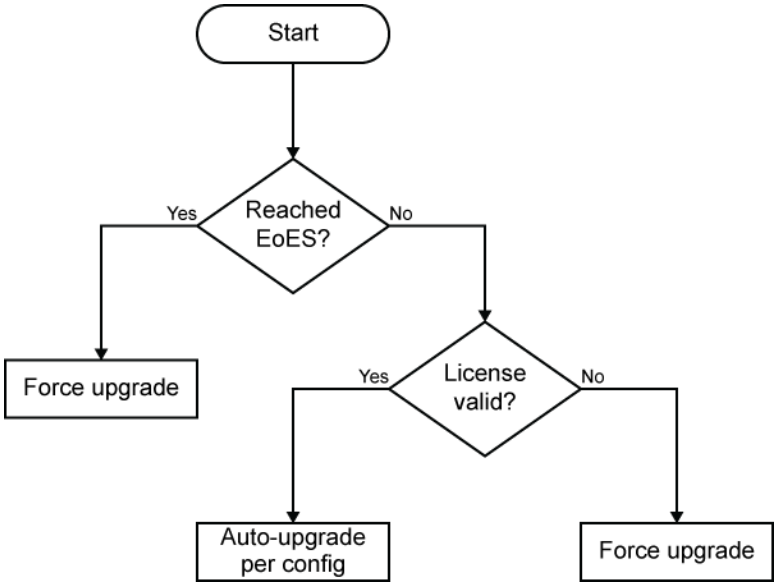
For required upgrades on FortiGates with an invalid license, the following information is communicated through the FortiGate FortiGuard communication protocol (FCPC) to FortiGuard:

- `ForcedUpdate` flag
- Current <major>.<minor>.<patch>-<build> version, for example `7.4.10-2878`

The `ForcedUpdate` flag instructs FortiGuard server to ignore the license check for the device and parse its firmware version. When the major and minor version on the upgrade-from and upgrade-to firmware are the same, the upgrade is allowed.

The firmware will upgrade to the latest patch in its current minor version. For example, if the current version is FortiOS 7.4.10, the firmware will automatically upgrade to the latest 7.4.x version. It will not upgrade to another minor version, such as 7.6.x. If there are multiple steps on the upgrade path, it will follow the upgrade path to reach the final patch.

The following decision tree summarizes when a required automatic upgrade occurs on a FortiGate:



Notification methods

Several methods are used to notify administrators about a required upgrade. Before a required upgrade starts, the FortiOS GUI banner shows a new firmware version is available, for example, *A new firmware version is available: v7.6.8*. After FortiOS confirms a valid upgrade path for the required upgrade:

- On the *System > Firmware & Registration* page:
  - The *Upgrade Status* column displays *Upgrade to v7.6.8 shortly*.
  - The *Cancel Fabric Upgrade* button is grayed out because forced upgrades cannot be canceled.
- The Console prints the following information when scheduled to install the new image:

```
Auto-upgrade new image check triggered.
```

```
Firmware Update from Management Service
Upgrade to image 7.6.8b3800-F (07006000FIMG0025906008) has been scheduled between Sat Apr  4
14:08: 36 2026 and Sat Apr  4 15:00:00 2026.
This upgrade is forced and cannot be cancelled, to delay this upgrade, please use "execute
auto-upgrade delay-installation".
```

- Default automation stitches trigger FortiGate to send email notifications:
  - FortiGate sends an email notification to the email address for the registered FortiCare account before and after upgrade using the `Firmware Upgrade Scheduled` and `Firmware Upgrade Complete` automation stitches. Following is an example of the email after upgrade:

    ```
    An automated firmware update has completed successfully. The device FortiGate-40F, serial
    number FGT40FTK<number>, was updated to firmware version 7.6.8 on 2026-4-4 15:00:06.

    This is an informational notification and was generated by automation stitch configuration
    on FortiGate-40F.
    ```

    See Default automation stitches on page 4494 for more information.

Example 1: FortiGate required upgrade when firmware license is invalid

In this hypothetical scenario, a FortiGate is running FortiOS 7.6.7, and its firmware license expired on Friday, January 2, 2026:

```
System contracts:
    FMWR,Fri Jan  2 2026
```

Checking its auto-upgrade status, the `Support Contract Status` is marked `Expired`:

```
# execute auto-upgrade status
Current Image Reached End of Life: no
Is current firmware a special build: No
Support Contract Status: Expired
Scheduled push image upgrade: no
Scheduled Config Restore: no
Scheduled Script Restore: no
Automatic image upgrade: Enabled.
        New image information may be fetched.
        New image installation may be cancelled by the user.
        Next new image info fetch scheduled at (local time) Wed Apr  1 21:07:59 2026
```

A new firmware is available on the upgrade path on FortiGuard. FortiGate schedules an upgrade.

```
# execute auto-upgrade status
Current Image Reached End of Life: no
Is current firmware a special build: No
Support Contract Status: Expired
Scheduled push image upgrade: no
Scheduled Config Restore: no
Scheduled Script Restore: no
Automatic image upgrade: Enabled (Forced).
        New image information may be fetched.
```

```
        New image installation will be forced.
        Next new image info fetch scheduled at (local time) Thu Apr  2 21:19:00 2026
        New image 7.6.8b3800(07006000FIMG0025906008) installation is scheduled to:
                start at Sat Apr  4 14:08:36 2026
                end by Sat Apr  4 15:00:00 2026
        Last new image info fetch executed at (local time) Wed Apr  1 21:07:59 2026
```

When the scheduled time is reached, FortiGate will request the download from the FortiGuard server. When FortiGuard server identifies this is a `Required-Upgrade`, it will bypass license check and allow the firmware image to be downloaded.

FortiGate will install the new image, reboot, and send an email notification to the administrator.

Example 2: FortiGate required upgrade when EOES reached

In this hypothetical scenario, a FortiGate has a valid firmware license but the current firmware has reached the EOES date.

The current firmware version is lower than the patch level image released on FortiGuard server.

Current firmware:

```
# get system status | grep Version
Version: FortiGate-40F v7.6.7,build3679,260330
```

Available firmware on FortiGuard:

```
# diagnose fdsm image-list
Last update: 895 secs ago. Total: 5
... ...
07006000FIMG0025906008  v7.6 MR6-GA-F P8 b3800 (upgrade)
07002000FIMG0025902012  v7.2 MR2-GA-F P12 b1800 (downgrade)
07002000FIMG0025902011  v7.2 MR2-NA- b1737 (downgrade)
```

Upgrade path:

```
# diagnose fdsm image-upgrade-matrix
Last update: 1013 secs ago. Total: 315
v8.0.0.(b150) -> v8.0.1.F(b1234)        (id:08000000FIMG0025900001)
v7.6.7.(b3679) -> v7.6.8.F(b3800)       (id:07006000FIMG0025906008)
```

Checking its auto-upgrade status, the end of life status is set to `yes`.

```
FortiGate-40F (Interim)# execute auto-upgrade status
Current Image Reached End of Life: yes
Is current firmware a special build: No
Support Contract Status: Active
Scheduled push image upgrade: no
Scheduled Config Restore: no
Scheduled Script Restore: no
Automatic image upgrade: Enabled.
        New image information may be fetched.
        New image installation may be cancelled by the user.
        Next new image info fetch scheduled at (local time) Thu Apr  2 14:33:55 2026
```

To force FortiGate to trigger a manual update-check, run this command:

```
# execute auto-upgrade check-for-new-image
Manually triggering auto-upgrade new image check...
Auto-upgrade new image check triggered.
Firmware Update from Management Service
Upgrade to image 7.6.8b3800-F (07006000FIMG0025906008) has been scheduled between Sat Apr  4
14:21:30 2026 and Sat Apr  4 15:00:00 2026.
This upgrade is forced and cannot be cancelled, to delay this upgrade, please use "execute auto-
upgrade delay-installation".
```

The FortiGate is now scheduled to perform the upgrade:

```
# execute auto-upgrade status
Current Image Reached End of Life: yes
Is current firmware a special build: No
Support Contract Status: Active
Scheduled push image upgrade: no
Scheduled Config Restore: no
Scheduled Script Restore: no
Automatic image upgrade: Enabled (Forced).
        New image information may be fetched.
        New image installation will be forced.
        Next new image info fetch scheduled at (local time) Thu Apr  2 14:51:45 2026
        New image 7.6.8b3800(07006000FIMG0025906008) installation is scheduled to:
                start at Sat Apr  4 14:21:30 2026
                end by Sat Apr  4 15:00:00 2026
        Last new image info fetch executed at (local time) Wed Apr  1 17:41:10 2026
```

Upon the completion of the upgrade, FortiGate will trigger an automation stitch to send the registered FortiCare account owner an email stating:

```
An automated firmware update has completed successfully. The device FortiGate-40F, serial number
      FGT40FTK19004135, was updated to firmware version 7.6.8 on 2026-4-4 15:00:06.
This is an informational notification and was generated by automation stitch configuration on
      FortiGate-40F.
```

Administrators can delay an upgrade for one (1) week using the command:

```
# execute auto-upgrade delay-installation
Postponing auto-upgrade image installation to a week later...
Auto-upgrade image installation rescheduled to: start at local time Sat Apr 11 10:12:08
2026
        end by local time Sat Apr 11 11:00:00 2026
```

Special considerations

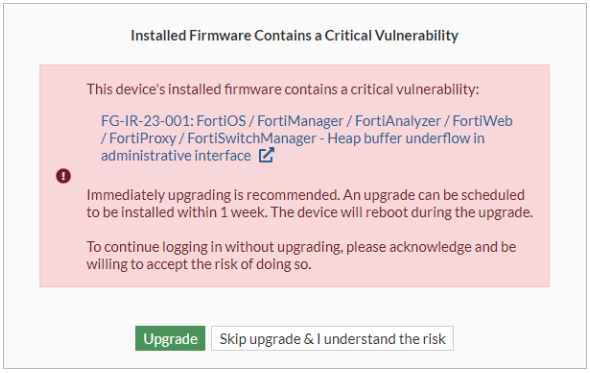
The status of the FortiGate may affect required upgrades as follows:

- When FortiGate is running a special FortiOS build, required upgrades do not apply.
- When FortiGate is a part of the Security Fabric, it will not automatically upgrade the firmware. Alternatively, when a required upgrade is scheduled, FortiGate cannot join a Security Fabric.

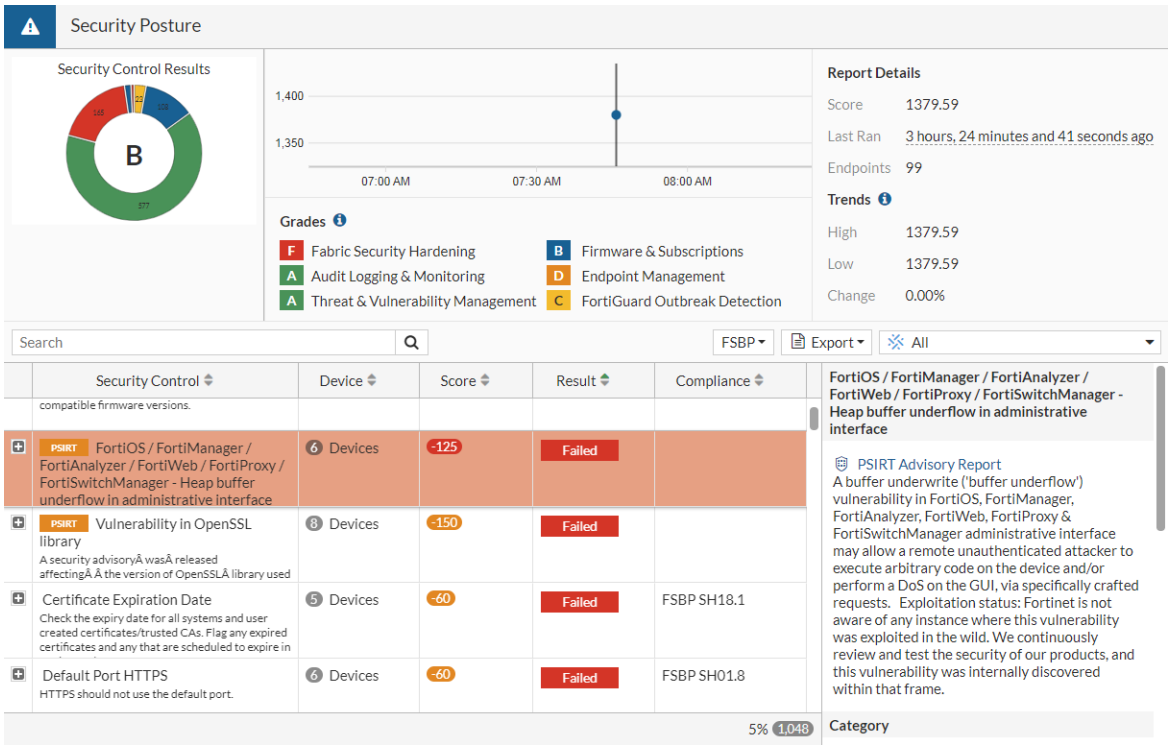
- When FortiGate is connected to FortiManager, it will not automatically upgrade the firmware. Likewise, when a required upgrade is scheduled, FortiGate can still connect to FortiManager, and the required firmware upgrade is canceled.
- When FortiGate is part of an HA pair, the required firmware upgrade occurs for the primary FortiGate. The secondary FortiGate will not perform the required firmware upgrade on its own because automatic upgrade is disabled on secondary; however, the secondary FortiGate will receive the upgrade through a cluster upgrade initiated by the primary FortiGate.
- When a required firmware upgrade is scheduled, it will block any new federated upgrades from occurring.

Upgrade prompt when a critical vulnerability is detected upon login

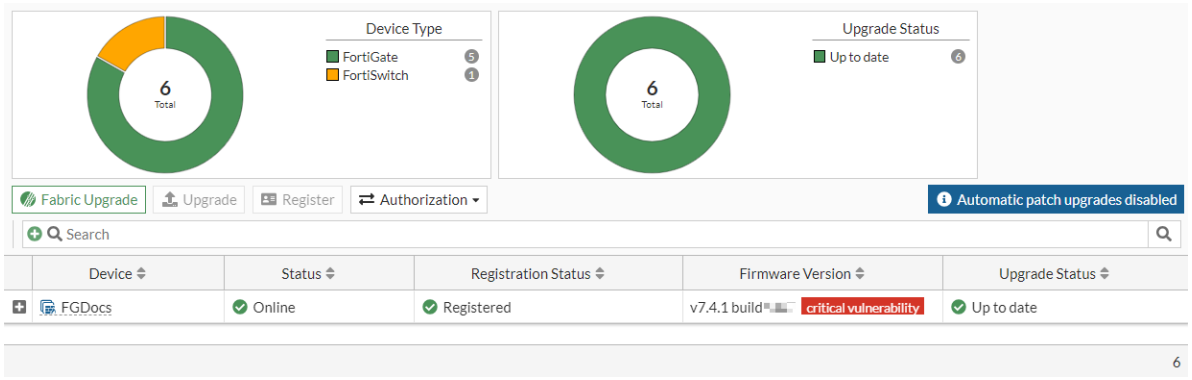
When FortiOS detects a critical vulnerability, an upgrade prompt is shown after logging into the FortiGate. A warning message is displayed in the GUI about the critical vulnerability and allows the administrator to either upgrade or skip it. This ensures that the administrator is aware of any potential security risks and can take immediate action to address them.



Clicking the hyperlinked vulnerability name opens the *Security Fabric > Security Rating* page, which displays more information about the vulnerability. See PSIRT-related notifications on page 4475 for more information.



Clicking the *Upgrade* button opens the *System > Firmware & Registration* page where the administrator can upgrade the device. See Firmware & Registration on page 3741 for more information.



Clicking the *Skip upgrade & I understand the risk* button continues the log in process as usual.

Diagnostics

To view vulnerability results after performing security rating scan:

```
# diagnose report-runner vuln-read
Index: 0
Name: FG-IR-23-001: FortiOS / FortiManager / FortiAnalyzer / FortiWeb / FortiProxy /
FortiSwitchManager - Heap buffer underflow in administrative interface
FortiGate Serial: FGVM02TM23000000
```

To clear the vulnerability result:

```
# diagnose report-runner vuln-clean
Deleted temporary critical vulnerability file
```

Viewing firmware upgrade reports

FortiOS now includes a firmware upgrade report that compares statistics and configurations before and after a firewall upgrade. Automatic generation of the report is enabled by default, and the report enhances the upgrade process by providing detailed assurance of successful upgrades.

The `config system global` command includes a new option:

```
config system global
    set upgrade-report {enable | disable}
end
```

Command	Description
`upgrade-report {enable | disable}`	Enable/disable automatic generation of an upgrade report when upgrading firmware (default = enable).

The `diagnose` command includes new options:

```
# diagnose sys upgrade-report
current     Display the stats currently.
on-disk     Display the stats saved on disk.
generate    Generate upgrade report and replace the one on disk.
```

Command	Description
`diagnose sys upgrade-report current`	Display the current statistics.
`diagnose sys upgrade-report on-disk`	Display the upgrade report saved on disk.
`diagnose sys upgrade-report generate`	Generate an upgrade report and replace the report saved on disk.

For FortiGates without disks, the report is stored on flash memory. All FortiGate models are equipped with flash memory.

Example

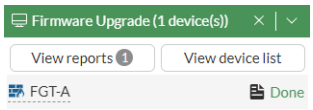
The firmware upgrade report is available after a successful FortiOS upgrade.

You can disable automatic generation of the firmware upgrade report in the CLI.

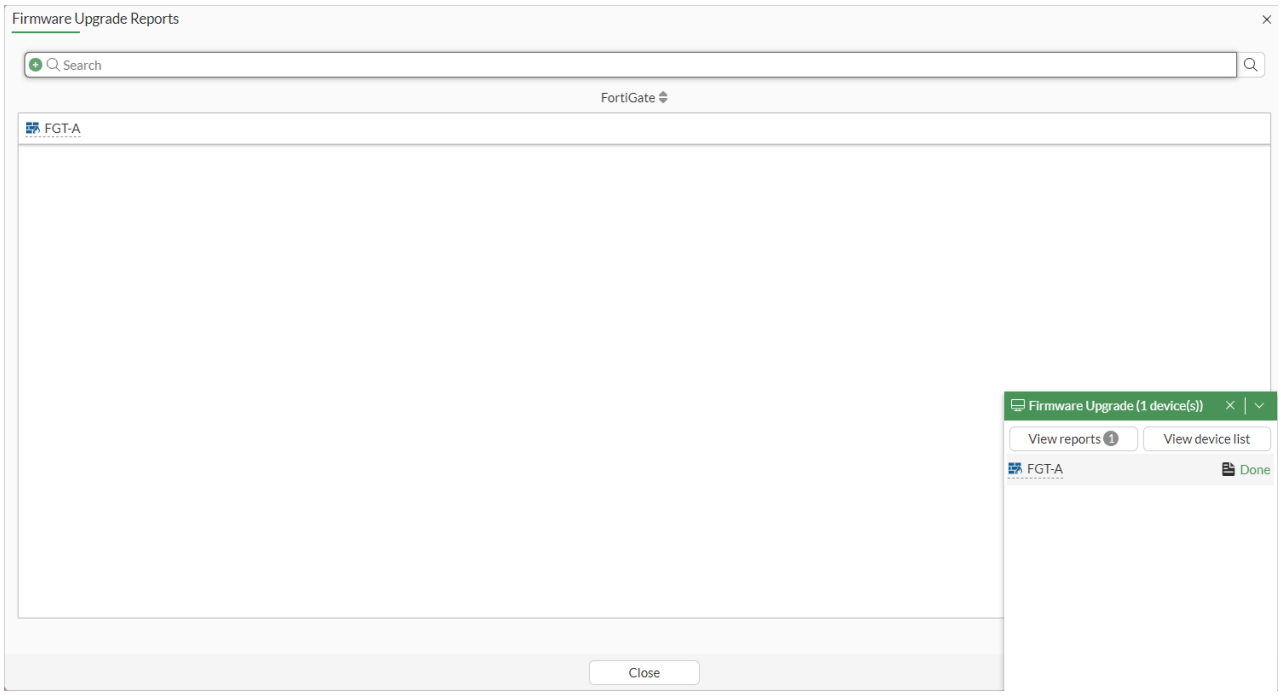
To view the firmware upgrade report in the GUI:

1. After a successful upgrade, log in to the GUI, and view the *Firmware Upgrade* monitor in the bottom-right corner. The following buttons are available:
   - *View reports <number>*

     The <number> indicates how many FortiGates have an available firmware upgrade report.
   - *View device list*



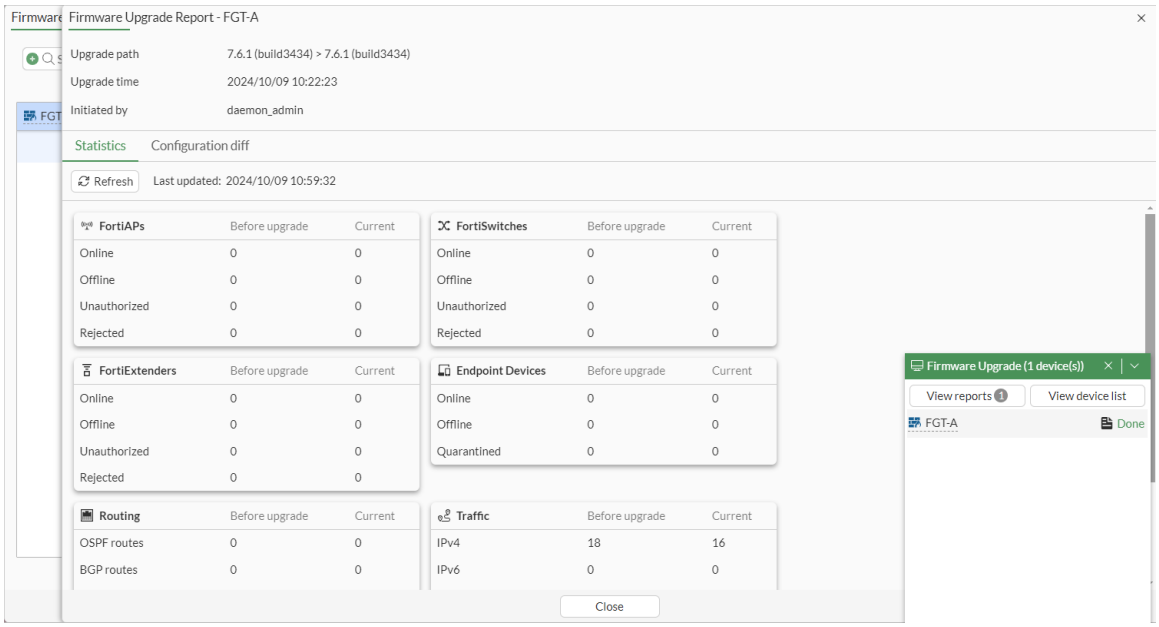
2. Click the *View reports <number>* button or the document icon beside *Done*. A *Firmware Upgrade Reports* pane opens to show the list of FortiGates with an available firmware upgrade report:



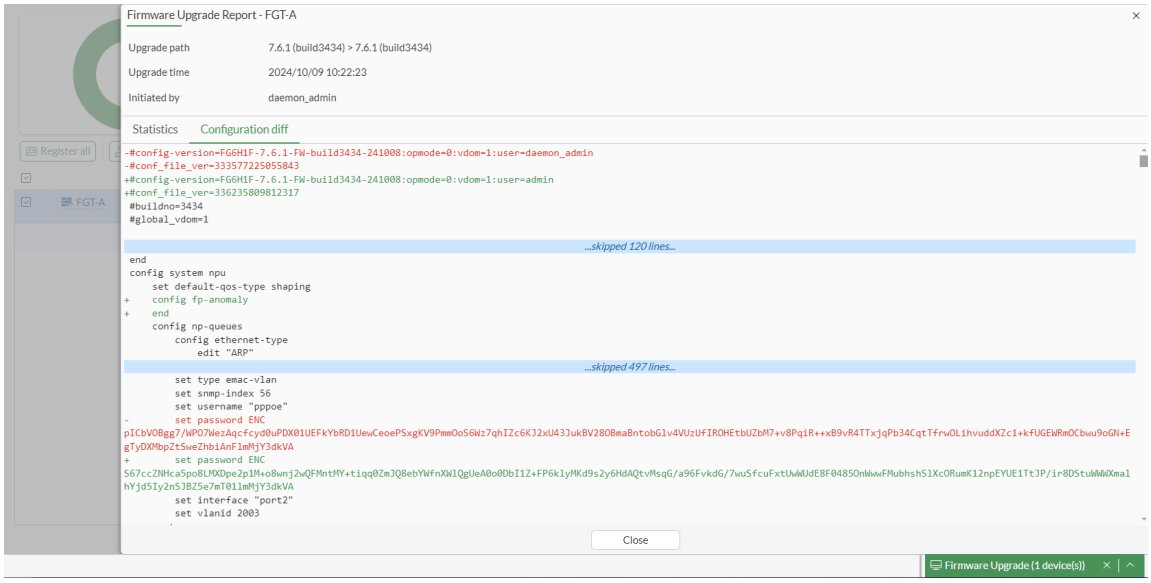
3. Double-click a FortiGate to display its firmware upgrade report. The firmware upgrade report contains the following tabs: *Statistics* and *Configuration diff*.
   - The *Statistics* tab displays general information at the top about *Upgrade path*, *Upgrade time*, and *Initiated by*.

     Below the general section are categories of information from before and after the upgrade. The categories of information depend on whether the FortiGate is part of a Security Fabric.

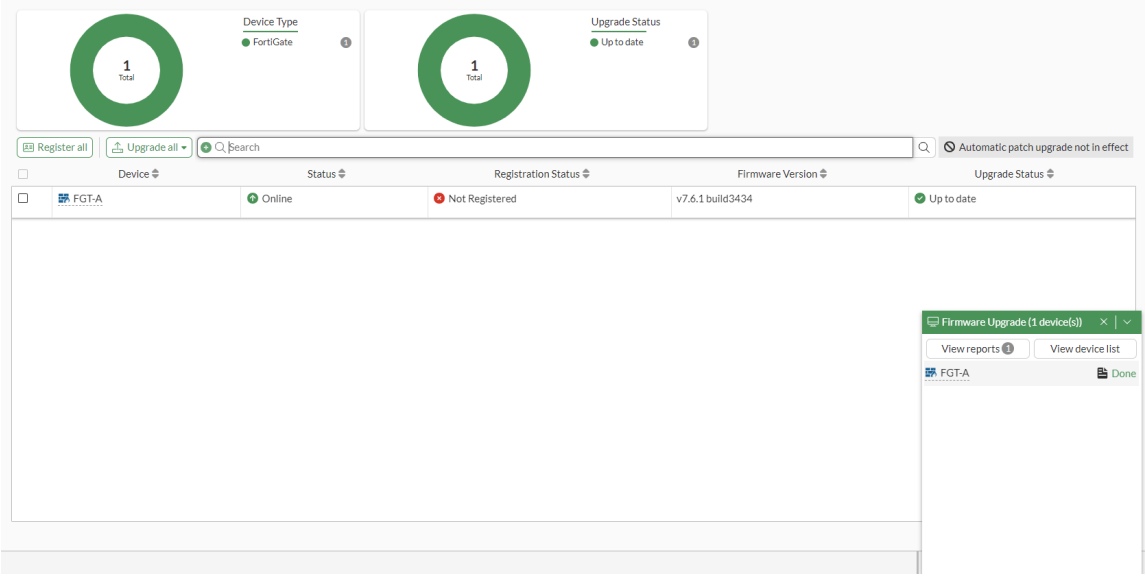
     In this example, the FortiGate is not part of a Security Fabric, and the following categories of information are displayed: *FortiAPs*, *FortiSwitches*, *FortiExtenders*, *Endpoint Devices*, *Routing*, *Traffic*, *Connectivity*, and *Resources*.



- Click *Refresh* to retrieve the latest current statistics.
- Click the *Configuration diff* tab to display a comparison of the current (green) and previous (red) configurations.



4. In the *Firmware Upgrade* monitor, click *View device list* to display the *Firmware & Registration* pane.



To display the firmware upgrade report in the CLI:

1. After a successful firmware upgrade, display the firmware upgrade report:

```
# diagnose sys upgrade-report on-disk
```

To regenerate the firmware upgrade report in the CLI:

1. Regenerate the firmware upgrade report:

```
# diagnose sys upgrade-report generate
This act will delete the upgrade report on disk and create a new one.
Do you want to continue? (y/n)
```

2. Display the firmware upgrade report:

```
# diagnose sys upgrade-report on-disk
```

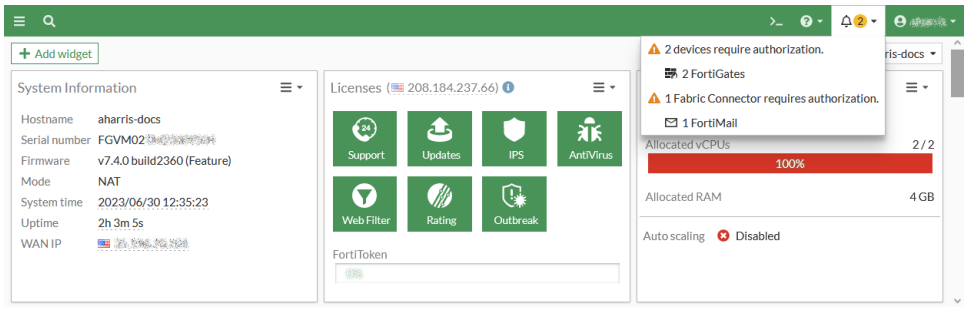
To view current statistics in the CLI:

```
# diagnose sys upgrade-report current
IPv4 session count = 24
IPv6 session count = 0
SSLVPN tunnel count = 0
IPSEC tunnel count = 6
CPU usage = 0%
Memory usage = 36%
Total routes = 18
Total OSPF routes = 0
Total BGP routes = 5
Total FortiGates = 1
  Online = 1
```

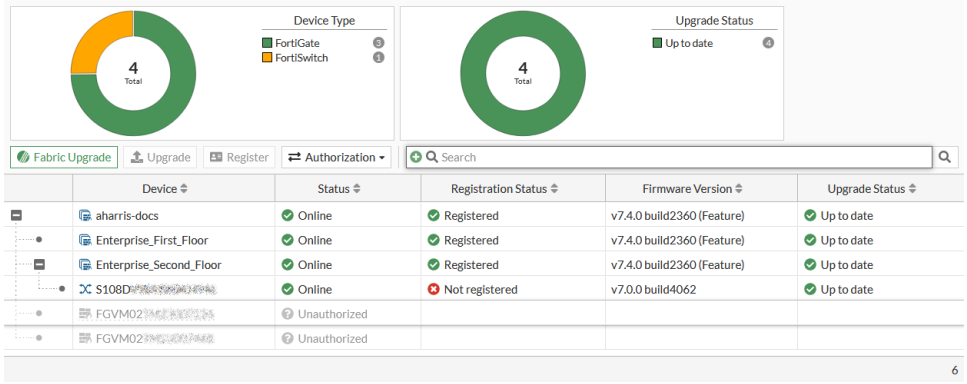
```
  Unauthorized = 0
Total FortiAPs = 0
  Online = 0
  Offline = 0
  Unauthorized = 0
  Rejected = 0
Total FortiSwitches = 0
  Online = 0
  Offline = 0
  Unauthorized = 0
  Rejected = 0
Total FortiExtenders = 0
  Online = 0
  Offline = 0
  Unauthorized = 0
  Rejected = 0
Total Endpoints = 0
  Online = 0
  Offline = 0
  Quarantined = 0
```

Authorizing devices

If there are any notifications in the top banner dropdown for unauthorized devices or devices that require authorization, clicking the notification redirects the user to the *System > Firmware & Registration* page. In this example, two devices require authorization.

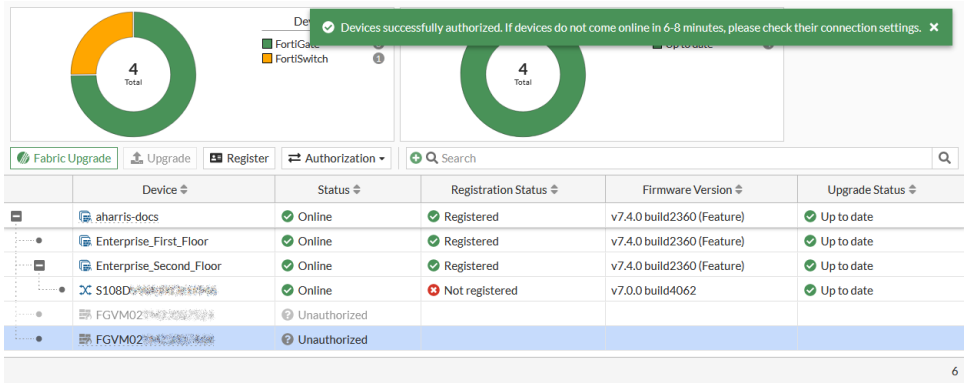


On the *Firmware & Registration* page, the unauthorized devices are grayed out, and their status is *Unauthorized*.

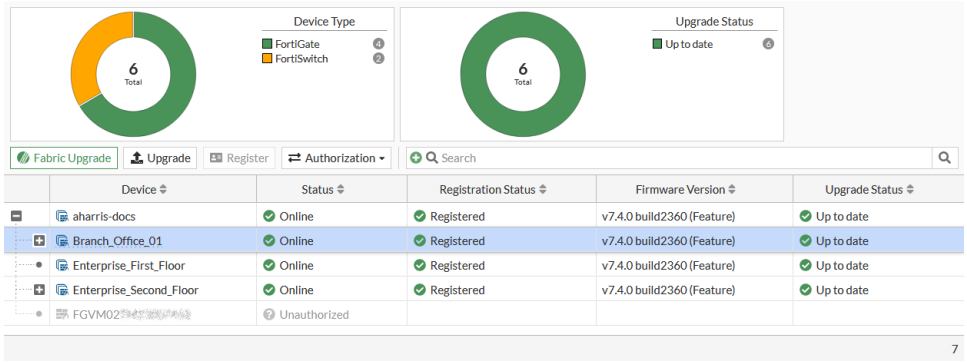


To authorize a Security Fabric device from the Firmware & Registration page:

1. Go to *System* > *Firmware & Registration*, and select an unauthorized device.
2. Click *Authorization* > *Authorize* (below the donut charts), or right-click and select *Authorization* > *Authorize*. A notification appears when the device is authorized.

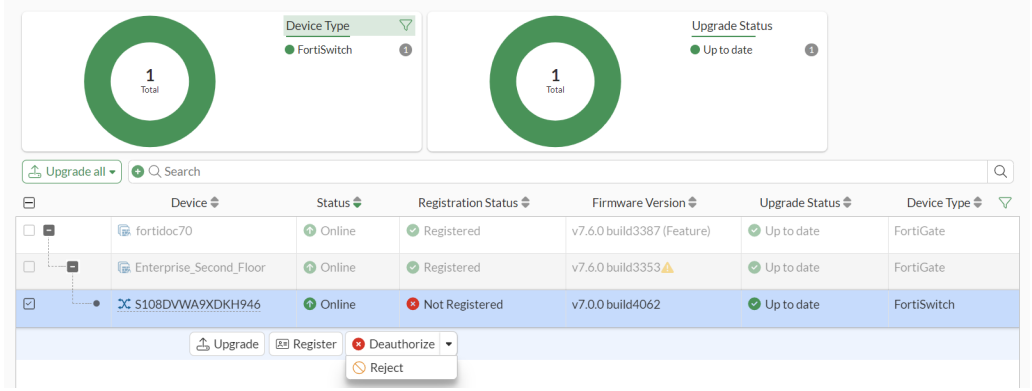


3. Refresh the page. The device's status should now be *Online*.



To deauthorize a Security Fabric device from the Firmware & Registration page:

1. Go to the *System* > *Firmware & Registration* page, and select a device.
2. Click *Deauthorize*.



3. Refresh the page to see that the device has been deauthorized.

Firmware upgrade notifications

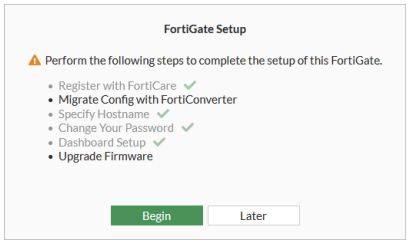
FortiGates with a firmware upgrade license that are connected to FortiGuard display upgrade notifications in the setup window, banner, and FortiGuard menu. The firmware notifications are enabled by default.

To configure firmware notifications in the CLI:

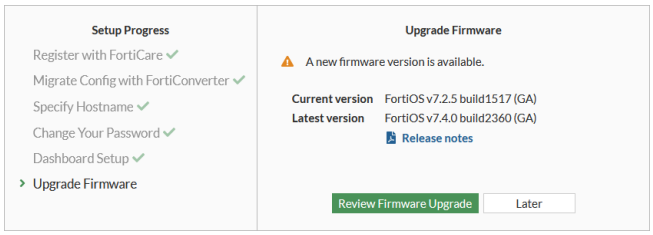
```
config system global
    set gui-firmware-upgrade-warning {enable | disable}
end
```

To use the firmware upgrade notifications in the GUI:

1. When you log in to FortiGate, the *FortiGate Setup* window includes an *Upgrade firmware* step. Click *Begin*.



2. Follow the steps in the *Setup Progress*, then click *Review Firmware Upgrade*.



   The *System* > *Firmware & Registration* page opens.
3. Notifications appear below the *Notification* icon in the banner, and beside *Firmware & Registration* in the tree menu.

Downloading a firmware image

Firmware images for all FortiGate units are available on the Fortinet Customer Service & Support website.

To download firmware:

1. Log into the support site with your user name and password.
2. Go to *Support* > *Firmware Download*.

   A list of Release Notes is shown. If you have not already done so, download and review the Release Notes for the firmware version that you are upgrading your FortiGate unit to.
3. Select the *Download* tab.
4. Navigate to the folder for the firmware version that you are upgrading to.

5. Find your device model on the list. FortiWiFi devices have file names that start with *FWF*.
6. Click *HTTPS* in the far right column to download the firmware image to your computer.

Firmware can also be downloaded using FTP, but as FTP is not an encrypted file transferring protocol, HTTPS downloading is recommended.

Security levels are pre-configured on the BIOS. See BIOS-level signature and file integrity checking on page 4005 and Real-time file system integrity checking on page 4009 for more information.

Testing a firmware version

The integrity of firmware images downloaded from Fortinet's support portal can be verified using a file checksum. A file checksum that does not match the expected value indicates a corrupt file. The corruption could be caused by errors in transfer or by file modification. A list of expected checksum values for each build of released code is available on Fortinet's support portal.

Image integrity is also verified when the FortiGate is booting up. This integrity check is done through a cyclic redundancy check (CRC). If the CRC fails, the FortiGate unit will encounter an error during the boot process.

Firmware images are signed and the signature is attached to the code as it is built. When upgrading an image, the running OS will generate a signature and compare it with the signature attached to the image. If the signatures do not match, the new OS will not load.

Testing before installation

FortiOS lets you test a new firmware image by installing the firmware image from a system reboot and saving it to system memory. After completing this procedure, the FortiGate unit operates using the new firmware image with the current configuration. The new firmware image is not permanently installed. The next time the FortiGate unit restarts, it operates with the originally installed firmware image using the current configuration. If the new firmware image operates successfully, you can install it permanently using the procedure explained in Upgrading individual devices.

For this procedure, you must install a TFTP server that you can connect to from the FortiGate internal interface. The TFTP server should be on the same subnet as the internal interface.

To test the new firmware version:

1. Connect to the CLI using an RJ-45 to USB (or DB-9) or null modem cable.
2. Ensure that the TFTP server is running.
3. Copy the new firmware image file to the root directory on the TFTP server.
4. Ensure that the FortiGate unit can connect to the TFTP server using the `execute ping` command.
5. Restart the FortiGate unit: `execute reboot`. The following message is shown:

```
This operation will reboot the system!
Do you want to continue? (y/n)
```

6. Type `y`. As the FortiGate unit starts, a series of system startup messages appears.

7. When the following messages appears:

```
Press any key to display configuration menu.........
```

   Immediately press any key to interrupt the system startup.

   You have only three seconds to press any key. If you do not press a key during this time, the FortiGate will reboot, and you will have to log in and repeat the `execute reboot` command.

   If you successfully interrupt the startup process, the following messages appears:

```
[G]: Get firmware image from TFTP server.
[F]: Format boot device.
[B]: Boot with backup firmware and set as default
[C]: Configuration and information
[Q]: Quit menu and continue to boot with default firmware.
[H]: Display this list of options.
Enter G, F, Q, or H:
```

8. Type *G* to get the new firmware image from the TFTP server. The following message appears: `Enter TFTP server address [192.168.1.168]:`
9. Type the address of the TFTP server, then press *Enter*. The following message appears: `Enter Local Address [192.168.1.188]:`
10. Type the IP address of the FortiGate unit to connect to the TFTP server.

    The IP address must be on the same network as the TFTP server.

    Make sure that you do not enter the IP address of another device on this network.

    The following message appears:

```
Enter File Name [image.out]:
```

11. Enter the firmware image file name then press *Enter*. The TFTP server uploads the firmware image file to the FortiGate unit and the following message appears:

```
Save as Default firmware/Backup firmware/Run image without saving: [D/B/R]
```

12. Type *R*. The FortiGate image is installed to system memory and the FortiGate unit starts running the new firmware image, but with its current configuration.

Test the new firmware image as required. When done testing, reboot the FortiGate unit, and the it will resume using the firmware that was running before you installed the test firmware.

Installing firmware from system reboot

In the event that the firmware upgrade does not load properly and the FortiGate unit will not boot, or continuously reboots, it is best to perform a fresh install of the firmware from a reboot using the CLI. If configured, the firmware can also be automatically installed from a USB drive; see Restoring from a USB drive on page 3780 for details.

This procedure installs a firmware image and resets the FortiGate unit to factory default settings. You can use this procedure to upgrade to a new firmware version, revert to an older firmware version, or re-install the current firmware.

To use this procedure, you must connect to the CLI using the FortiGate console port and a RJ-45 to USB (or DB-9), or null modem cable. You must also install a TFTP server that you can connect to from the FortiGate internal interface. The TFTP server should be on the same subnet as the internal interface.

Before beginning this procedure, ensure that you backup the FortiGate unit configuration. See Configuration backups and reset on page 4053 for details. If you are reverting to a previous FortiOS version, you might not be able to restore the previous configuration from the backup configuration file.

Installing firmware replaces your current antivirus and attack definitions, along with the definitions included with the firmware release you are installing. After you install new firmware, make sure that antivirus and attack definitions are up to date.

To install firmware from a system reboot:

1. Connect to the CLI using the RJ-45 to USB (or DB-9) or null modem cable.
2. Ensure that the TFTP server is running.
3. Copy the new firmware image file to the root directory of the TFTP server.
4. Ensure that the FortiGate unit can connect to the TFTP server using the `execute ping` command.
5. Restart the FortiGate unit: `execute reboot`. The following message is shown:

```
This operation will reboot the system!
Do you want to continue? (y/n)
```

6. Type `y`. As the FortiGate unit starts, a series of system startup messages appears.
7. When the following messages appears:

```
Press any key to display configuration menu.....
```

   Immediately press any key to interrupt the system startup.

   You have only three seconds to press any key. If you do not press a key during this time, the FortiGate will reboot, and you will have to log in and repeat the `execute reboot` command.

   If you successfully interrupt the startup process, the following messages appears:

```
[C]: Configure TFTP parameters.
[R]: Review TFTP parameters.
[T]: Initiate TFTP firmware transfer.
[F]: Format boot device.
[I]: System information.
[B]: Boot with backup firmware and set as default.
[Q]: Quit menu and continue to boot.
[H]: Display this list of options.

Enter C,R,T,F,I,B,Q,or H:
```

8. If necessary, type `C` to configure the TFTP parameters, then type `Q` to return to the previous menu:

```
[P]: Set firmware download port.
[D]: Set DHCP mode.
```

```
[I]: Set local IP address.
[S]: Set local subnet mask.
[G]: Set local gateway.
[V]: Set local VLAN ID.
[T]: Set remote TFTP server IP address.
[F]: Set firmware file name.
[E]: Reset TFTP parameters to factory defaults.
[R]: Review TFTP parameters.
[N]: Diagnose networking(ping).
[Q]: Quit this menu.
[H]: Display this list of options.

Enter P,D,I,S,G,V,T,F,E,R,N,Q,or H:
```

The IP address must be on the same network as the TFTP server.
Make sure that you do not enter the IP address of another device on this network.

9. Type `T` get the new firmware image from the TFTP server.
   The FortiGate unit loads the firmware.
10. Save the firmware as the default (`D`) or backup (`B`) firmware image, or run the image without saving it (`R`).
    The FortiGate unit installs the new firmware image and restarts. The installation might take a few minutes to complete.

Factory resetting the FortiGate when the password is lost

For security reasons, users who lose their password must have physical access to the FortiGate and perform a TFTP restore of the firmware in order to regain access to the FortiGate. They will not have access to the current running configurations through the FortiGate. Configurations will be reset to the factory default once the firmware is reloaded. This process requires a connection to the TFTP server where the firmware image is stored.

To restore the FortiGate:

This procedure may vary depending on whether the FortiGate is a physical appliance or a VM.

1. Connect to the console port.
2. Ensure you can see the FortiGate prompt from the console terminal.
3. Physically power off the device, then power on the device.
4. Boot into the boot menu by pressing a key when prompted.
5. Follow the steps in the previous procedure to reload the firmware. Configurations will be reset to the factory default once the firmware is installed.
6. Once the firmware reload is complete, log in to the FortiGate to reconfigure the settings.

It is recommended to preform regular configuration backups and to store the backup on a secure server (see Configuration changes in the FortiOS Best Practices for more details). In the event that a password is lost, the configuration backup can be used to restore a configuration after the user completes the firmware installation process. This assumes the user knows the password from the previous backed up configuration. If the user does not know the password, they can still reload the configuration if it is not encrypted.

The following procedure describes how to edit an unencrypted backup configuration file so that the administrator password can be replaced before restoring the file.

To edit the configuration file when a password is lost:

1. Locate the line in the configuration file where `config system admin` is defined.
2. Edit an administrator account with an `accprofile` set to `super_admin`. This will ensure you can log in and perform any operations afterward.
3. Locate the line with `set password ENC xxxxxx`, and edit it to set a temporary new password in clear text (such as `set password cleartextpassword`).
4. Reload the configuration file.
5. Log in to the console using the temporary password, and then change the password.

The configuration backup allows the administrator to confirm the firmware that the FortiGate is running, so the same firmware can be restored. This information is listed in the first line of the configuration: `config-version=FGT61F-7.2.4-FW-build1396-230131:opmode=0:vdom=0:user=admin`.

Restoring from a USB drive

The FortiGate firmware can be manually restored from a USB drive, or installed automatically from a USB drive after a reboot.

To restore the firmware from a USB drive:

1. Copy the firmware file to the root directory on the USB drive.
2. Connect the USB drive to the USB port of the FortiGate device.
3. Connect to the FortiGate CLI using the RJ-45 to USB (or DB-9) or null modem cable.
4. Enter the following command:

```
execute restore image usb <filename>
```

The FortiGate unit responds with the following message:

```
This operation will replace the current firmware version! Do you want to continue? (y/n)
```

5. Type y. The FortiGate unit restores the firmware and restarts. This process takes a few minutes.
6. Update the antivirus and attack definitions:

```
execute update-now
```

To install firmware automatically from a USB drive:

1. Go to *System* > *Settings* > *Device*.
2. In the *Start up settings* section, enable *Detect firmware* and enter the name of the firmware file.
3. Copy the firmware file to the root directory on the USB drive.

4. Connect the USB drive to the USB port of the FortiGate device.
5. Reboot the FortiGate device.

Using controlled upgrades

Using a controlled upgrade, you can upload a new version of the FortiOS firmware to a separate partition in the FortiGate memory for later upgrade. The FortiGate unit can be configured so that when it is rebooted, it will automatically load the new firmware. Using this option, you can stage multiple FortiGate units to upgrade simultaneously using FortiManager or a script.

To load the firmware for later installation:

```
execute restore secondary-image {ftp | tftp | usb} <filename_str>
```

To set the FortiGate unit so that when it reboots, the new firmware is loaded:

```
execute set-next-reboot {primary | secondary}
```

where `{primary | secondary}` is the partition with the preloaded firmware.

Downgrading individual device firmware

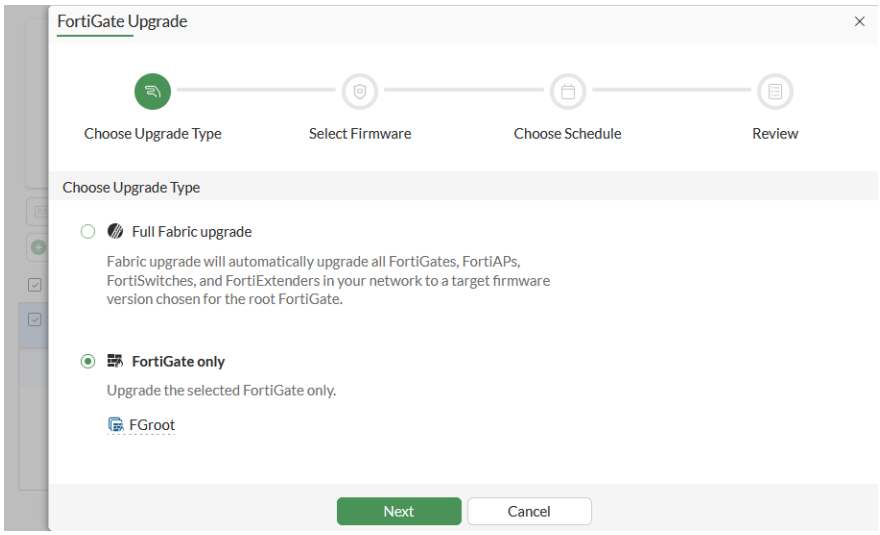
Downgrading the firmware is not recommended.

Downgrading FortiGates in an HA cluster causes all cluster members to be downgraded simultaneously. This process, also known as an interrupted downgrade, leads to a temporary interruption in the cluster's communication.

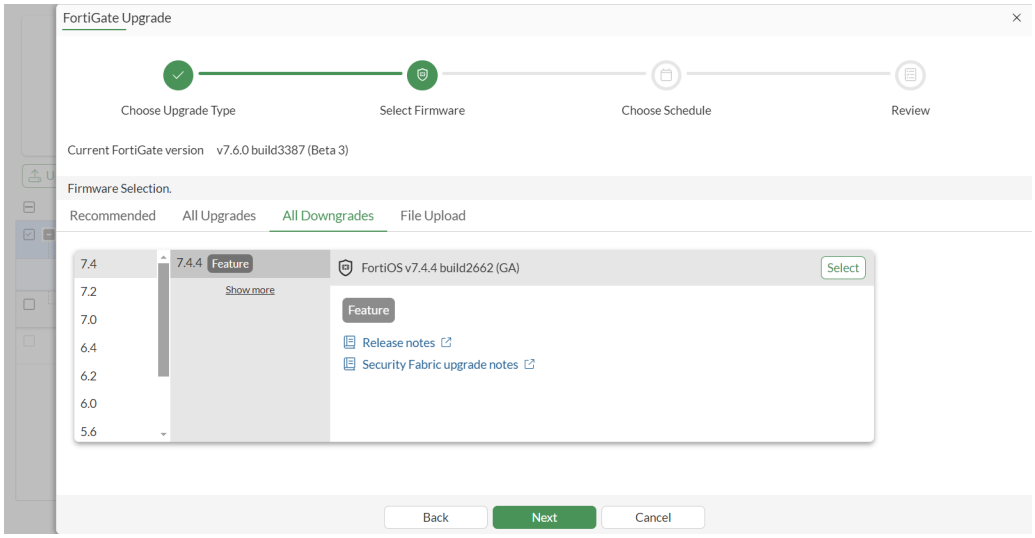
This procedure downgrades the FortiGate to a previous firmware version. After downgrading, you may be unable to restore the backup configuration.

To downgrade to a previous firmware version in the GUI:

1. Log into the FortiGate GUI as the admin administrative user.
2. Go to *System* > *Firmware & Registration*. The *Firmware Version* column displays the version and either *(Feature)* or *(Mature)*.
3. Select the FortiGate, and click *Upgrade*. The *FortiGate Upgrade* pane opens.



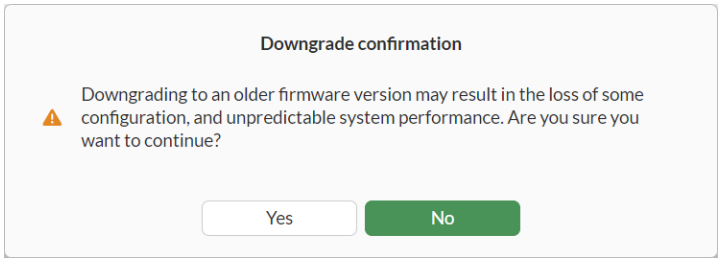
4. Select *FortiGate* only, and click *Next* to proceed to the *Select Firmware* step.
5. For the *Select Firmware* step:



   a. On the *All Downgrades* tab, select a firmware version, and click *Select*.

   > You can also click the *File Upload* tab to upload a firmware file that you previously downloaded from the Fortinet Customer Service & Support website.
   >
   > See Downloading a firmware image on page 3775.

   b. Click *Next*. A *Downgrade confirmation* message is displayed



c. Click *Yes* to proceed to the *Choose Schedule* step.

6. For the *Choose Schedule* step:
   a. Choose a schedule:

Immediate	Select to start the upgrade immediately after completing the *FortiGate Upgrade* wizard.
Specify	Select to schedule a date and time to start the upgrade after completing the *FortiGate Upgrade* wizard.

   b. Click *Next* to proceed to the *Review* step.
7. For the *Review* step, review the downgrade plan, and click *Confirm and Backup Config* to proceed. The *Confirm* dialog box is displayed.
8. Click *Yes* to continue with the downgrade.
   The FortiGate unit backs up the current configuration to the management computer, uploads the firmware image file, downgrades to the firmware version, and restarts. This process takes a few minutes.

To downgrade to a previous firmware version in the CLI:

1. Make sure that the TFTP server is running.
2. Copy the new firmware image file to the root directory of the TFTP server.
3. Log into the CLI.
4. Ping the TFTP server to ensure that the FortiGate can connect to it:

```
execute ping <tftp_ipv4>
```

5. Enter the following command to copy the firmware image from the TFTP server to the FortiGate unit:

```
execute restore image tftp <filename> <tftp_ipv4>
```

   The FortiGate unit responds with the message:

```
This operation will replace the current firmware version!
Do you want to continue? (y/n)
```

6. Type y. The FortiGate unit uploads the firmware image file, then a message similar to the following is shown:

```
Get image from tftp server OK.
Check image OK.
This operation will downgrade the current firmware version!
Do you want to continue? (y/n)
```

7. Type y. The FortiGate unit downgrades to the old firmware version and restarts. This process takes a few minutes.
8. Reconnect to the CLI.
9. Update the antivirus and attack definitions:

```
execute update-now
```

Downloading the EOS support package for supported Fabric devices

FortiGates, FortiSwitches, FortiAPs, and FortiExtenders can download an EOS (end of support) package automatically from FortiGuard during the bootup process or by using manual commands. Based on the downloaded EOS package files, when a device passes the EOS date, a warning message is displayed in the device's tooltip. The device is also highlighted in the following GUI locations:

- *System > Firmware & Registration* page
- *Security Fabric > Physical Topology* and *Logical Topology* pages
- *Dashboard > Status > System Information* widget

FortiGuard updates

The EOS packages can be downloaded automatically from FortiGuard, but they can also be downloaded manually.

To manually download the EOS package from the FortiGuard server:

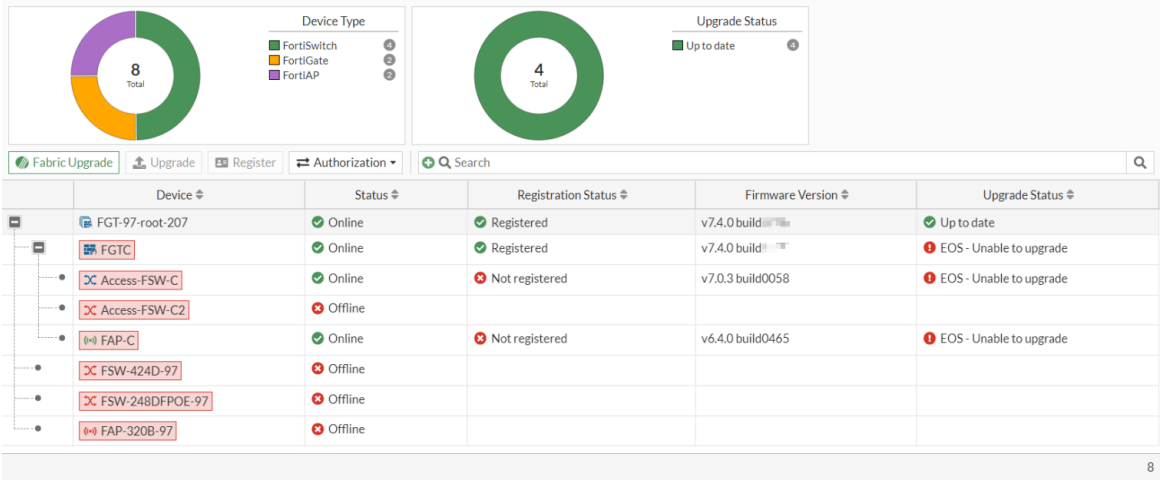
```
# diagnose fortiguard-resources update <product>-end-of-support
```

Product	Description
`fortigate-end-of-support`	FortiGate product life cycle information.
`fortiswitch-end-of-support`	FortiSwitch product life cycle information.
`fortiap-end-of-support`	FortiAP product life cycle information.
`fortiextender-end-of-support`	FortiExtender product life cycle information.

In the event the EOS package files are not downloaded due to a connection issue, use `diagnose fortiguard-resources update <product>-end-of-support` to download the package files.

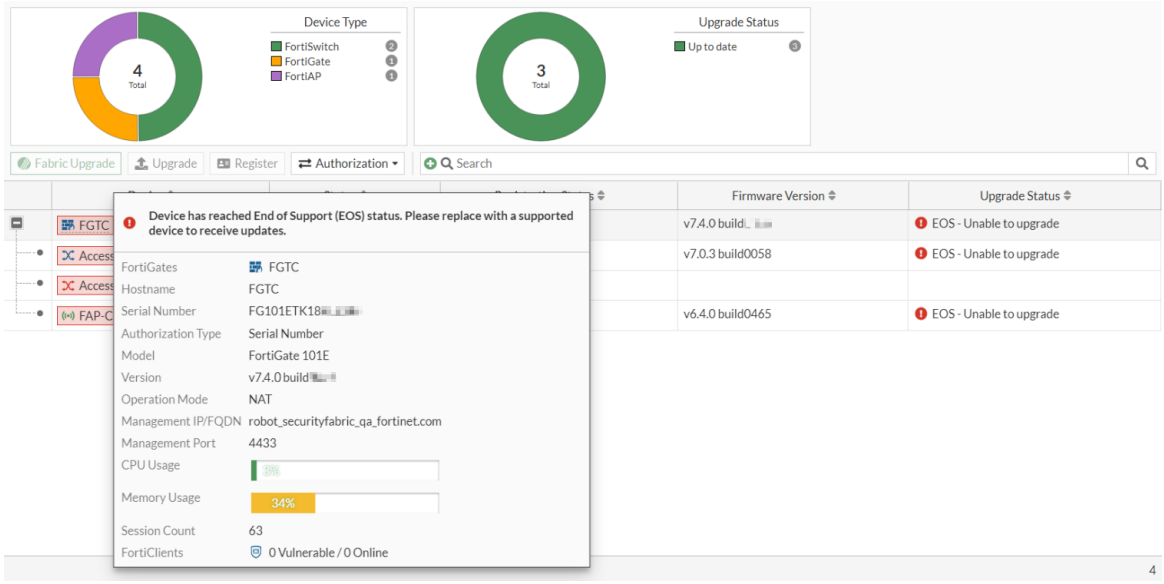
GUI warnings

On the *System > Firmware & Registration* page, devices that have reached EOS are highlighted in red, and their *Status* is *EOS - Unable to upgrade*.

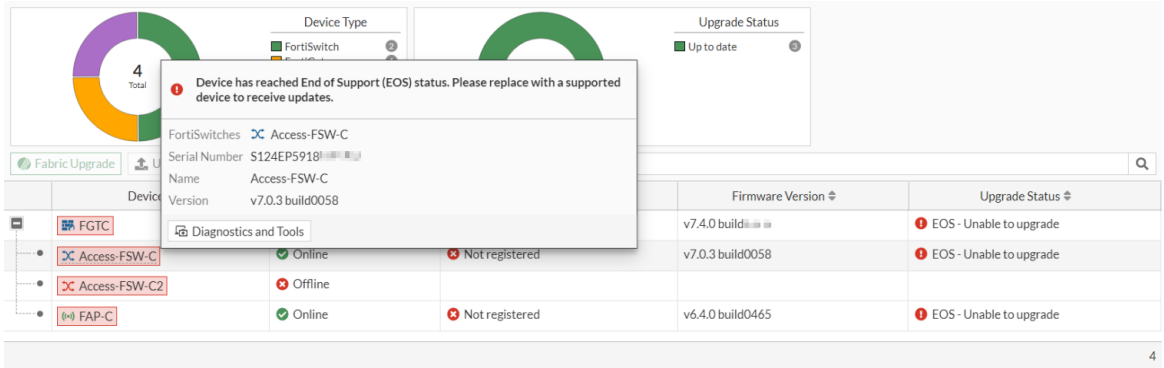


Hover over a device name to view the tooltip, which includes an EOS warning.

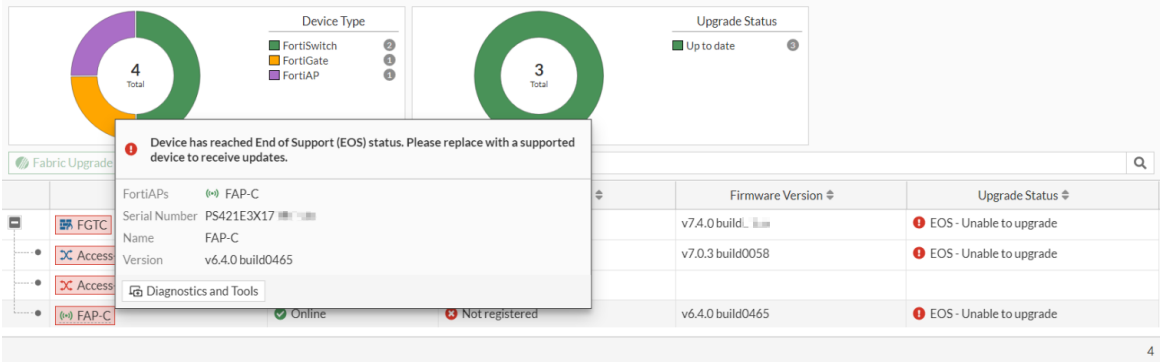
- Sample FortiGate tooltip:



- Sample FortiSwitch tooltip:

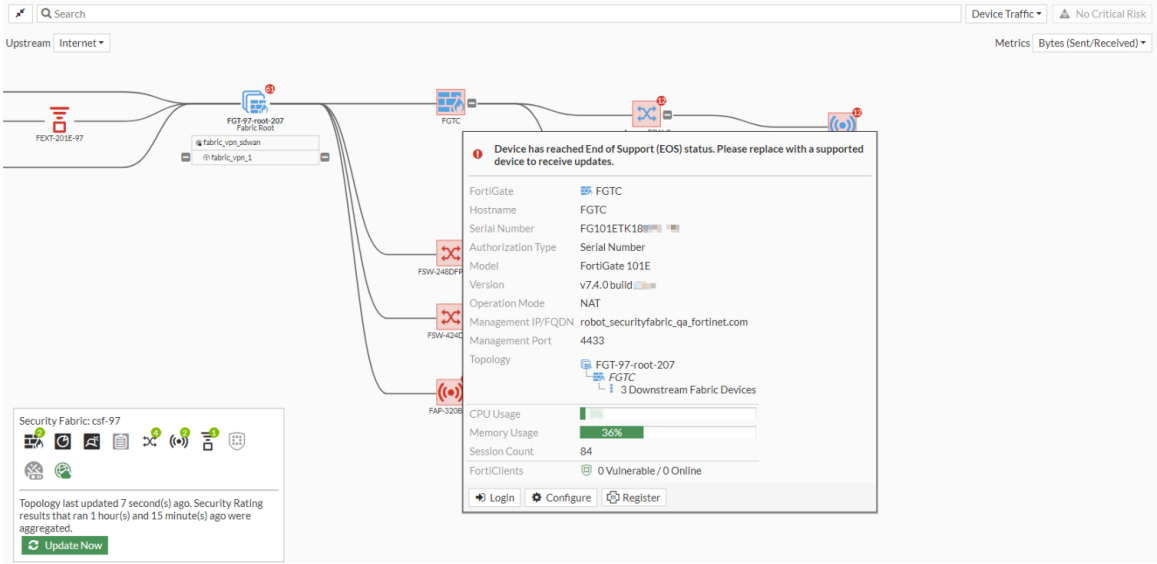


- Sample FortiAP tooltip:

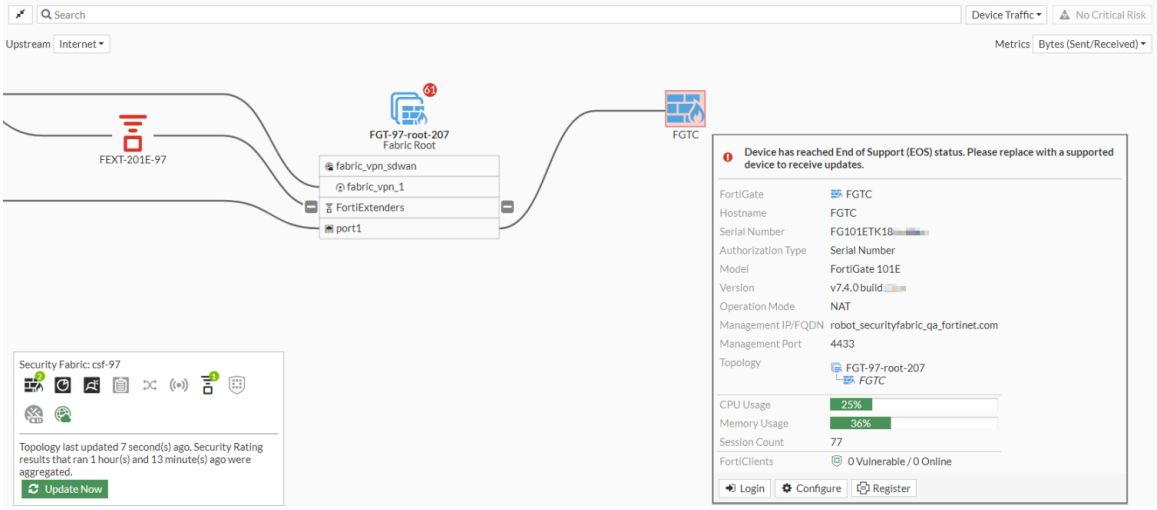


On the *Security Fabric > Physical Topology* and *Logical Topology* pages, devices that have reached EOS are highlighted in red. The device tooltips also include an EOS warning.

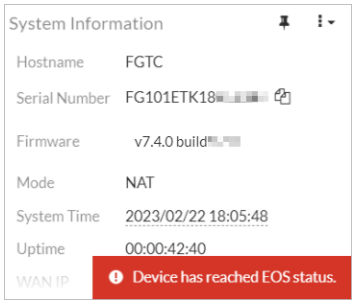
- Sample *Security Fabric > Physical Topology* page with tooltip:



- Sample *Security Fabric > Logical Topology* page with tooltip:



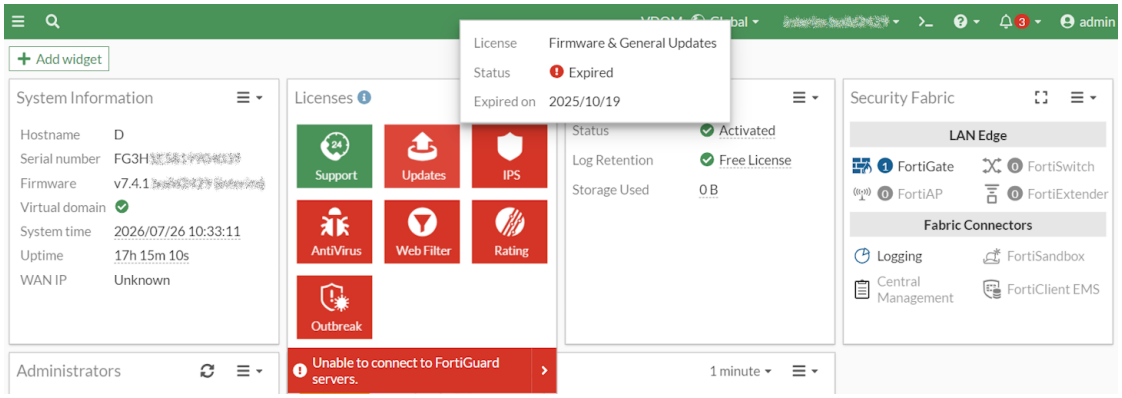
The *Dashboard* > *Status* > *System Information* widget includes a warning at the bottom of the widget that the *Device has reached EOS status*.



How the FortiGate firmware license works

> Firmware upgrades are performed in the *System* > *Firmware & Registration* page or in the CLI. To demonstrate the functionality of this feature, the following explanations and examples use FortiGates that are running and upgrading to fictitious build numbers with fictitious release dates. For more information on performing an upgrade, see Upgrading individual devices on page 3749.

You can confirm the *Firmware & General Updates* (FMWR) contract expiry date in the *System* > *FortiGuard* page, by using the `diagnose test update info contract | grep FMWR` command, or by hovering your mouse over the *Updates* tile in the *Licenses* widget in *Dashboard* > *Status*.



Maintaining an active support contract for your FortiGate allows you to access the latest firmware upgrades and downgrades including:

- Updates between major versions, such as upgrading from FortiOS 7.0 to 8.0
- Updates between minor versions, such as upgrading from FortiOS 7.4 to 7.6
- Updates between patch versions, such as upgrading from FortiOS 7.6.1 to 7.6.2

If you have an invalid support contract or the FortiGate version is End of Support, enforced, automatic firmware upgrades will upgrade the FortiGate to the latest patch of the current, minor version. For example, if the current version is FortiOS 7.4.3, the firmware will automatically upgrade to the latest 7.4.x version. It will not upgrade to another minor version, such as 7.6.x. See Required firmware upgrades for FortiGates with invalid support contracts or that have reached EOES on page 3762.

In FortiOS 7.4.2 and above, enforcement is based on the expiry date of the current firmware license compared to the release date of the first GA release of a major or minor version. For example, for FortiOS 7.4.x firmware upgrades, enforcement is based on the expiry date of the current support contract compared to the release date of FortiOS 7.4.0 GA.

Therefore, upgrades and downgrades between major and minor versions are only allowed if the firmware license is valid relative to the release date of the first GA release of a major or minor version. If the firmware license expiry date is earlier than the firmware first GA major or minor release date, then the firmware update to that version will not be allowed.

For example, the release dates of minor versions are as follows:

- 7.4.0 GA release on May 8, 2023
- 7.6.0 GA release on March 31, 2024
- 7.8.0 GA release on March 31, 2025

This example is using fictitious GA release dates of future versions for illustrative purposes only. These dates do not indicate the official FortiOS release schedule.

Firmware license expiry date	Is a FortiGate firmware upgrade allowed to the target version?		
	7.4.x	7.6.x	7.8.x
March 31, 2025 or later	Yes	Yes	Yes
March 25, 2025	Yes	Yes	No
March 25, 2024	Yes	No	No
May 2, 2023	No	No	No

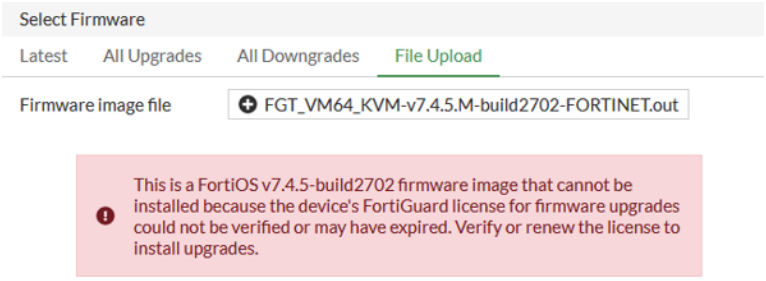
Downgrades from one major or minor version to another are not blocked because the FortiGate should have had a firmware expiry date that is later than the release date of the older firmware major or minor version.

For example, if the firmware license expiry date was March 25, 2024, the FortiGate is currently running 7.4.2 and you wanted to downgrade to 7.2.7, since the release date of 7.2.0 GA was March 31, 2022 then this firmware downgrade would be allowed. The firmware license expiry date is later than the release date of the older firmware minor version, 7.2.0 GA.

Upgrading firmware in the GUI through file upload

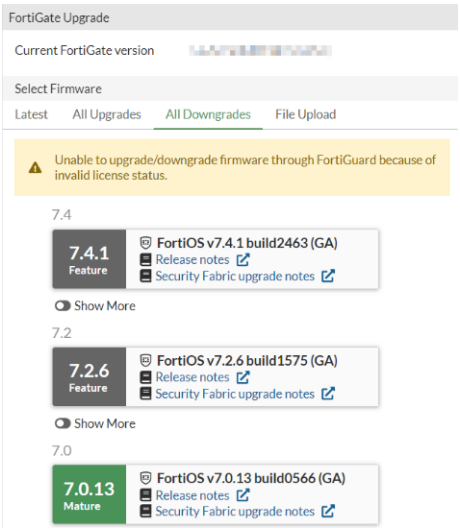
If the contract is expired relative to the major firmware GA .0 version release date, the following upgrade attempt will be blocked in the GUI *System > Firmware & Registration* page.

If the contract expiry date is earlier than the release date for the GA .0 version of a higher major, minor, or patch version of firmware that is being uploaded, then the upgrade is denied and an error is displayed.



Upgrading firmware in the GUI through FortiGuard

Until the support contract is renewed, FortiGuard upgrades will be unavailable, namely, the *Confirm and Backup Config* button will be grayed out. However, you will be able to view the FortiGate firmware images available on FortiGuard using *Latest*, *All Upgrades*, and *All Downgrades* tabs and this functionality will be restored upon support contract renewal.



Upgrading firmware in the CLI

The following example demonstrates what occurs when upgrading the firmware to a patch build and to a higher version with a license expiry date that is earlier than the major GA .0 version release date in the CLI. The major upgrade attempts and fails to upgrade the firmware from FortiOS 7.4.0 to 7.6.3.

This behavior is also observed for minor and patch upgrades with a license expiry date that is earlier than the major GA .0 version release date of the minor or patch firmware targeted for the upgrade.

To upgrade the firmware to a higher major version:

1. Confirm the current firmware version:

```
# get system status
Version: FortiGate-301E v7.4.0,build2303,230307 (interim)
```

2. Upgrade the firmware:

```
# execute restore image tftp  v763-B1505-GA-F_B234847_FGT_301E.out 172.16.200.55
......
Firmware update licence is expired! Please update to a valid licence.
Command fail. Return code -180
```

If your firmware support contract has expired, please contact your Fortinet Sales/Partner for details on renewing it.

Troubleshooting

To verify the presence of the FMWR license:

```
# diagnose test update info contract
…
System contracts:
    ENHN,Thu May  1 2025
    COMP,Thu May  1 2025
    FMWR,Thu May  1 2025
…
Support contract: pending_registration=255 got_contract_info=1
    account_id=[xxxxxxxx@fortinet.com] company=[Fortinet] industry=[Technology]

SerialNumber=FGVMxxxxxxxxxx|Contract=AVDB-1-06-20250503:0:1:1:0*AVEN-1-06-20250503:0:1:1:0*NIDS-1-
06-20250503:0:1:1:0*SPRT-1-20-202
50503:0:1:1:0*ZHVO-1-06-20250503:0:1:1:0*SWNO-1-06-20250503:0:1:1:0*SWNM-1-06-
20250503:0:1:1:0*SWNC-1-06-20250503:0:1:1:0*FGSA-1-06-2
0250503:0:1:1:0*SPAM-1-06-20250503:0:1:1:0*ISSS-1-06-20250503:0:1:1:0*IPMC-1-06-
20250503:0:1:1:0*IOTH-1-06-20250503:0:1:1:0*FURL-1-06
-20250503:0:1:1:0*FRVS-1-06-20250503:0:1:1:0*FMWR-1-06-20250503:0:1:1:0*FMGC-1-06-
20250503:0:1:1:0*FCSS-1-10-20250503:0:1:1:0*FAZC-1-
06-20250503:0:1:1:0*DLDB-1-06-20250516:0:1:1:0*ENHN-1-20-20250503:0:1:1:0*COMP-1-20-
20250503:0:1:1:0|
…
```

In the special scenario where the FMWR license information is missing, the FortiGate will not be allowed to upgrade. This scenario may occur when running on an EVAL license, where FMWR license is not supported.

Enhanced firmware upgrade management for extension devices

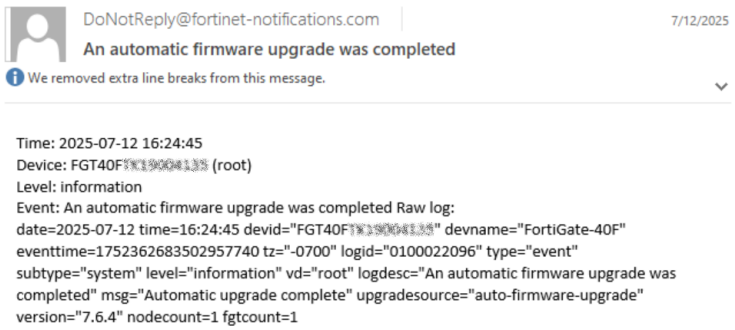
FortiGate allows users to dismiss specific firmware upgrade prompts for extension devices, helping reduce unnecessary notifications. When a recommended firmware release is available, an *Upgrade Available* notification appears on both the *System > Firmware & Registration* page and the device management pages. Users can choose to hide notifications for builds they do not intend to install.

Upgrade logs IDs have been added to distinguish between automatic and manual upgrades, and email alerts include detailed status updates. Logs are triggered when auto-upgrade is enabled or disabled, a new firmware is detected, and when the upgrade completes or fails. After disabling auto-upgrade and performing an update, the login GUI prompts users to manually confirm their auto-upgrade preference.

New log IDs:

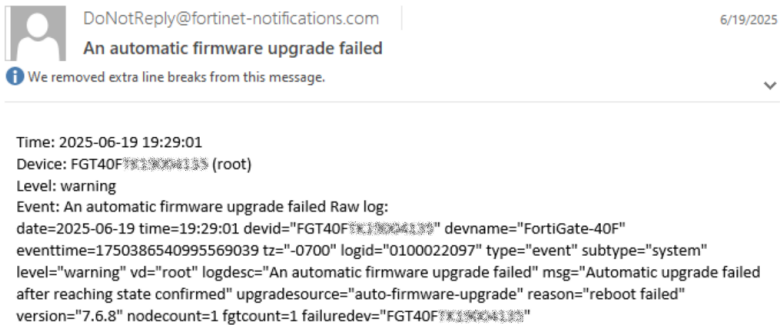
- Log ID **22096**: Auto-upgrade completed successfully

```
date=2026-02-04 time=16:26:00 id=7603170842209419270 itime="2026-02-04 16:26:01" euid=3 epid=3
dsteuid=3 dstepid=3 logflag=0 logver=800000123 logid=0100022096 type="event" subtype="system"
level="information" msg="Automatic upgrade complete" logdesc="An automatic firmware upgrade
was completed" version="8.0.0" eventtime=1770251160711843738 tz="-0800" nodecount=1 fgtcount=1
upgradesource="auto-firmware-upgrade" devid="FGVM<serial number>" vd="root" csf="csf_97_vdom"
dtime="2026-02-04 16:26:00" itime_t=1770251161 devname="FGT-97-vdom-209"
```



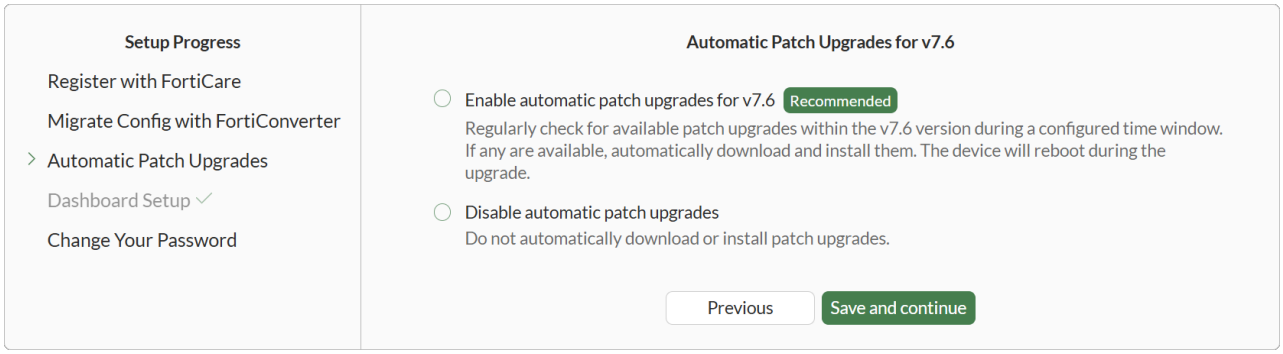
- Log ID **22097**: Auto-upgrade failed

```
date=2025-07-12 time=18:12:58 eventtime=1752369177089407840 tz="-0700" logid="0100022097"
type="event" subtype="system" level="warning" vd="root" logdesc="An automatic firmware upgrade
failed" msg="Automatic upgrade failed after reaching state confirmed" upgradesource="auto-
firmware-upgrade" reason="firmware changed" version="7.6.7" nodecount=1 fgtcount=1
failuredev="FGT40F0000000000"
```



Login prompt:

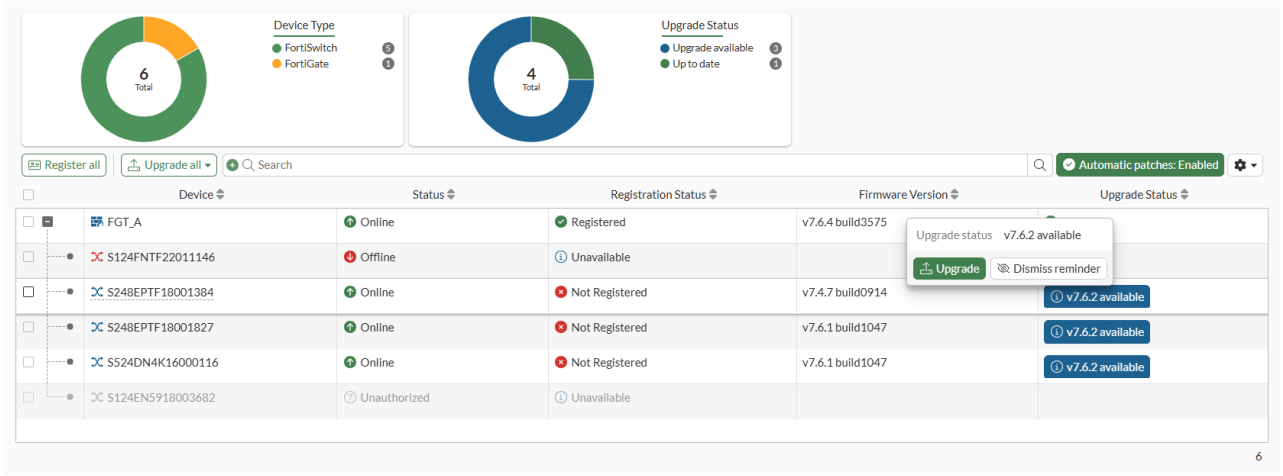
- When auto-upgrade is disabled, no option is preselected in the setup wizard and Enable is marked as recommended. Enable or Disable must be selected to proceed.



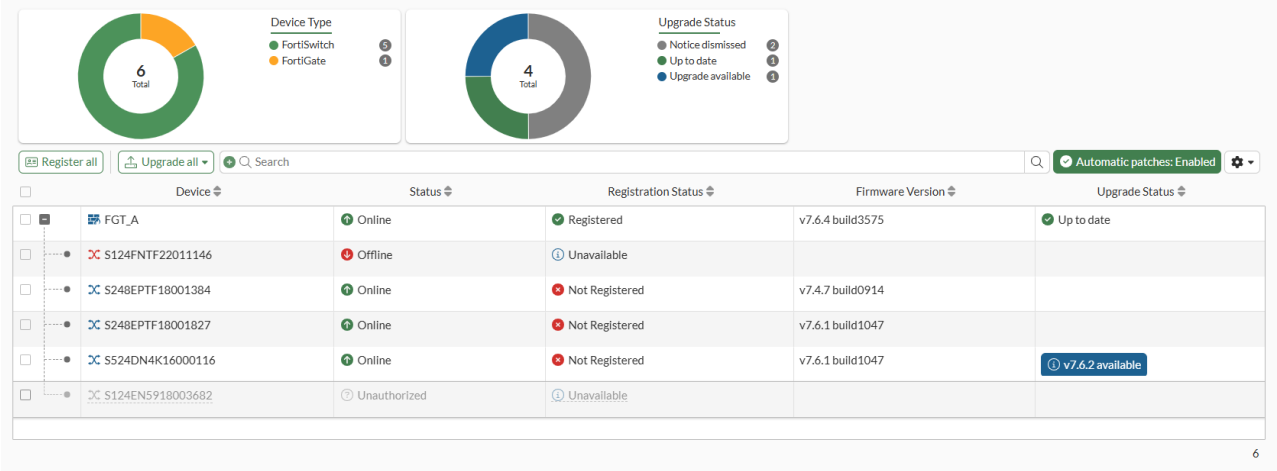
- When auto-upgrade is enabled, *Enable* is pre-selected.

To dismiss extension device upgrade notifications in the GUI:

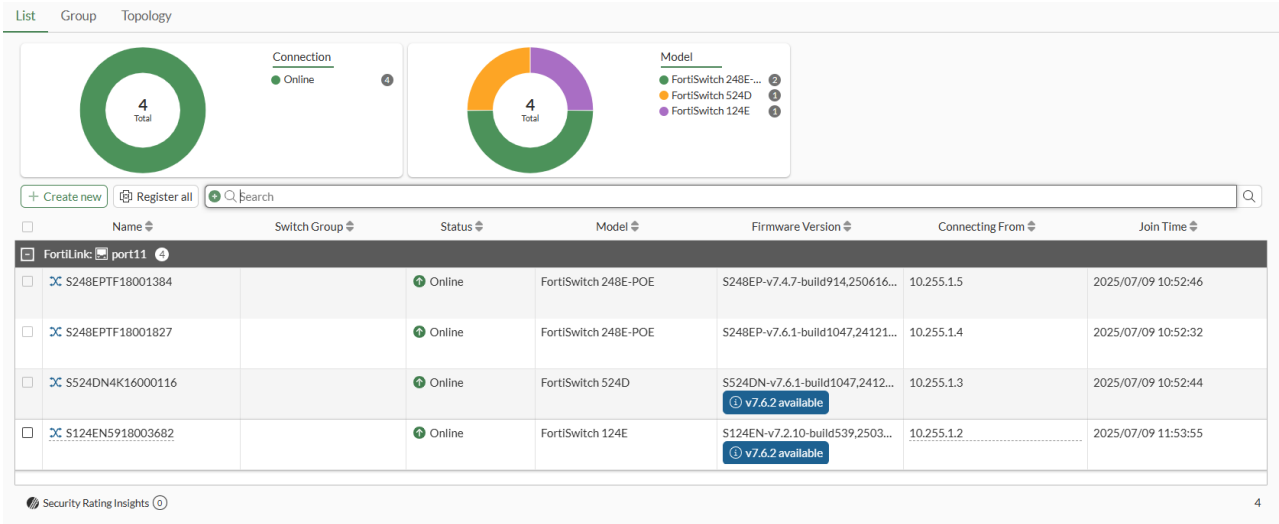
1. Go to *System* > *Firmware & Registration*.
   The recommended upgrade notifications are shown in the *Upgrade Status* column.
2. Hover over a notification and click *Dismiss reminder* to remove the notification.



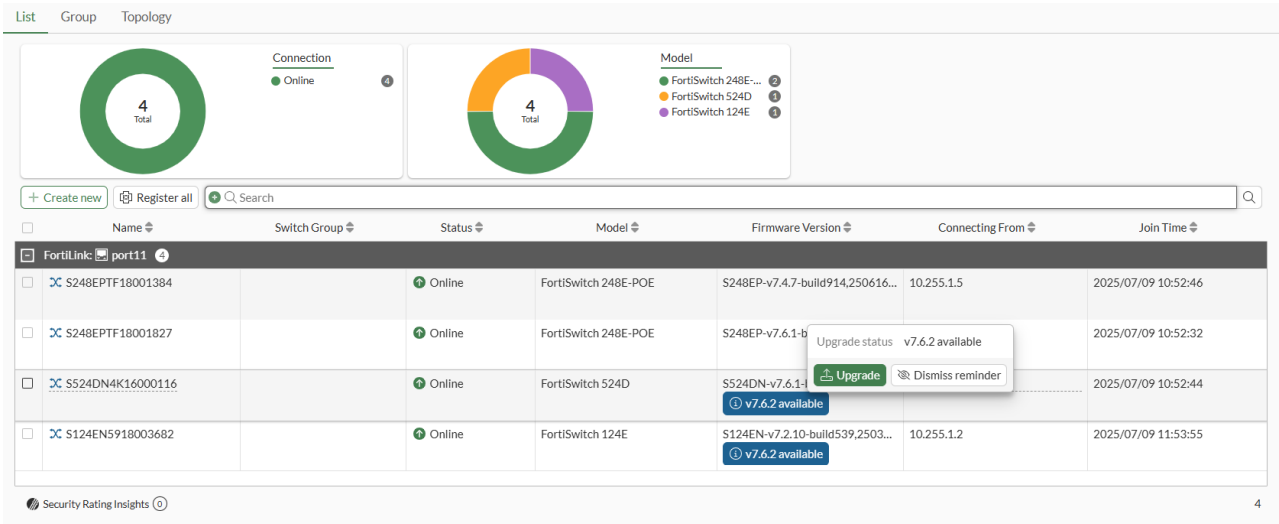
The page refreshes and the notification is no longer displayed.



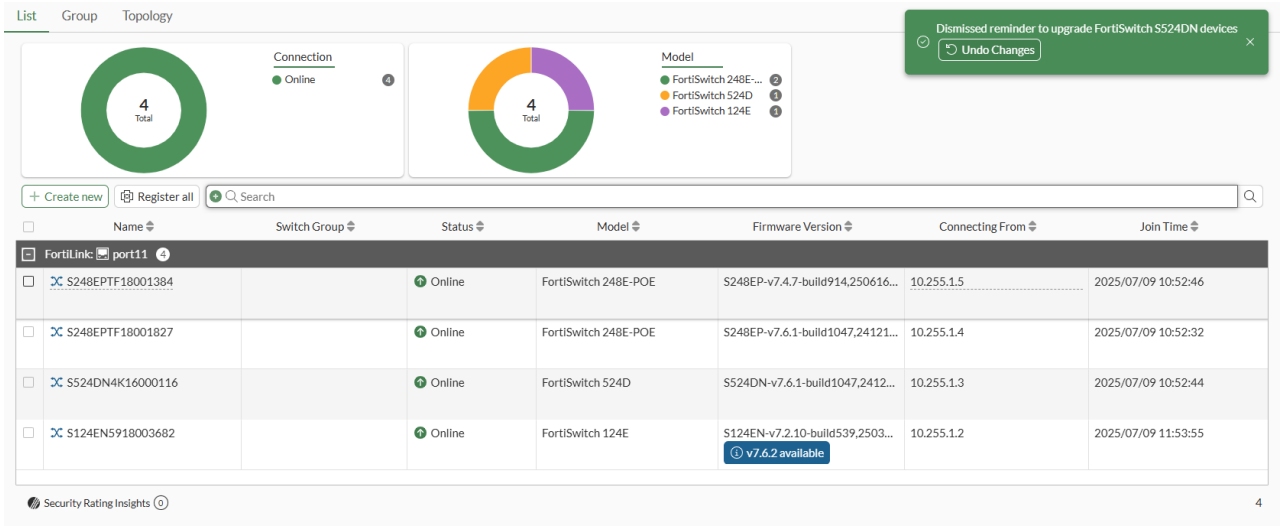
3. Go to *WiFi & Switch Controller* > *Managed FortiSwitches*.
   The notification is also no longer displayed on the page.



4. Hover over a notification on this page and click *Dismiss reminder* to remove the notification.



The page refreshes and the notification is no longer displayed.



To dismiss extension device upgrade notifications in the CLI:

```
config system device-upgrade-exemptions
    edit 1
        set fortinet-device <device>
        set version <version>
    next
end
```

Option	Description
`fortinet-device <device>`	Fortinet extension device (FortiAP, FortiSwitch, FortiExtender).
`version <version>`	Highest version of Fortinet firmware to exempt (format in Major.minor.patch, such as 7.6.4).

For example, to dismiss the version 7.6.2 notification for the FortiSwitch 248E-POE:

```
config system device-upgrade-exemptions
    edit 1
        set fortinet-device "S248EP"
        set version 7.6.2
    next
end
```

Settings

The default administrator password should be configured immediately after the FortiGate is installed, see Default administrator password on page 3810.

On the *System* > *Settings* page, options are organized across the following tabs:

General	Use the *General* tab to: • Change the host name. See Changing the host name on page 3796. • Set the system time. See Setting the system time on page 3797. • Configure an email service for alerts. See Email alerts on page 3805. • Set the Next Generation Firewall (NGFW) mode. • Enable VLAN switch mode. • Download a FortiCare debug report.
Access	Use the *Access* tab to: • Configure ports. See Configuring ports on page 3812 and Custom default service port range on page 3812. • Configure an administrator password policy. See Setting the password policy on page 3813 and Password policy on page 3721. • Configure administrator password retries and lockout time. See Setting the administrator password retries and lockout time on page 3813. • Configure the idle timeout. See Setting the idle timeout time on page 3813. • Configure HTTPS server certificate for admin access. See Using the default certificate for HTTPS administrative access on page 3819.
Device	Use the *Device* tab to: • Allow FortiConverter access to the FortiGate configuration file. • Configure automatic installation of a FortiGate configuration file or firmware from a USB stick. • Configure disk settings for WAN optimization or logging.
Preference	Use the *Preference* tab to: • Configure the save mode. See Using configuration save mode on page 3824. • Set the language. See Setting the FortiOS language on page 3825. • Set the GUI color theme.

General tab

Use the *System* > *Settings* > *General* tab to change the system time, configure email service, specify the system operation settings, and so on.

To configure the General tab:

1. Go to *System* > *Settings*. The *General* tab is displayed.
2. Under *System time*, set the system time. See Setting the system time on page 3797.
3. Under *Email service*, configure alerts. See Email alerts on page 3805.
4. Under *System operation settings*, configure the following options:

VLAN switch mode	Enable/disable VLAN switch mode. See Virtual VLAN switch on page 258.
NGFW mode	Set the Next Generation Firewall mode: • *Profile-based*: Application and web-filtering are configured using profiles applied to policy entries. • *Policy-based*: Application and web-filtering are configured as policy match conditions. See NGFW policy on page 1868.
Central SNAT	Defines and controls the address translation performed by FortiGate. See Central SNAT on page 1924 and Central DNAT on page 1962. Always enabled when *NFGW mode* is set to *Policy-based*. Can be enabled or disabled when *NFGW mode* is set to *Profile-based*.
Virtual domains	Enable/disable virtual domains. See Virtual Domains on page 3835.

5. Under *Configuration*, configure the following options:

Create a configuration revision on logout	Enable/disable automatic creation of a revision when administrators log out of FortiGate and changes were made during the session. See Configuration revision on page 4060 for more information.

6. Under *FortiCare debug* report, click *Download* to download the debug log file for the FortiGate.
7. Click *Apply* to save the changes.

Changing the host name

The FortiGate host name is shown in the *Hostname* field in the *System Information* widget on a dashboard, as the command prompt in the CLI, as the SNMP system name, as the device name on FortiGate Cloud, and other places. If the FortiGate is in an HA cluster, use a unique host name to distinguish it from the other devices in the cluster.

An administrator requires *System* > *Configuration* read/write access to edit the host name. See Administrator profiles on page 3736 for details.

To change the host name in the GUI:

1. Go to *System* > *Settings*.
2. In the *Host name* field, enter a new name.
3. Click *Apply*.

To change the host name in the CLI:

```
config system global
    set hostname <hostname>
end
```

Setting the system time

You can either manually set the FortiOS system time, or configure the device to automatically keep its system time correct by synchronizing with a Network Time Protocol (NTP) or Precision Time Protocol (PTP) server.

For many features to work, including scheduling, logging, and SSL-dependent features, the FortiOS system time must be accurate.

To configure the date and time in the GUI:

1. Go to *System* > *Settings* > *General*.
2. In the *System Time* section, configure the following settings to either manually set the time or use an NTP server:

Current system time	Displays the current system time.
Time Zone	Select a time zone from the list. This should be the time zone that the FortiGate is in.
Set Time	Select either *NTP*, *PTP*, or *Manual settings*.
NTP	To use an NTP server other than FortiGuard, the CLI must be used. In the *Sync interval* field, enter how often, in minutes, that the device synchronizes its time with the NTP server.
PTP	• Set the *Mode* to *Multicast* or *Hybrid*. • Select the *Delay mechanism*: *E2E* or *P2P*. • Set the *Request interval*, in seconds. • Select the *Interface*. For more information, see PTPv2 on page 3800.
Manual settings	Manually enter the *Date*, and *Time*.
Setup device as local NTP server	Enable to configure the FortiGate as a local NTP server. This option is not available if *Set Time* is *PTP*. In the *Listen on Interfaces* field, set the interface or interfaces that the FortiGate will listen for NTP requests on.

3. Click *Apply*.

To configure the date and time in the CLI:

1. Configure the timezone:

```
config system global
    set timezone <integer>
end
```

2. Either manually configure the date and time, or configure an NTP or PTP server:
   - Manual:

```
execute date <yyyy-mm-dd>
execute time <hh:mm:ss>
```

   - NTP server:

```
config system ntp
    set ntpsync enable
    set type {fortiguard | custom}
    set syncinterval <integer>
    set source-ip <ip_address>
    set source-ip6 <ip6_address>
    set server-mode {enable | disable}
    set interface <interface>
    set authentication {enable | disable}
    set key-type {MD5 | SHA1}
    set key <password>
    set key-id <integer>
    config ntpserver
        edit <server_id>
            set server <ip_address or hostname>
            set ntpv3 {enable | disable}
            set authentication {enable | disable}
            set interface-select-method {auto | sdwan | specify}
            set key <password>
            set key-id <integer>
        next
    end
end
```

   - PTP server:

```
config system ptp
    set status enable
    set mode {multicast | hybrid}
    set delay-mechanism {E2E | P2P}
    set request-interval <integer>
    set interface <string>
end
```

Cryptographic hash function authentication support

Cryptographic hash function authentication support allows the NTP client to verify that severs are known and trusted and not intruders masquerading (accidentally or intentionally) as legitimate servers.

The supported cryptographic hash functions include:

- MD5: Supported by NTPv3 for authentication.
- MD5, SHA-1, and SHA-256: Supported by NTPv4, providing stronger security and greater flexibility.

> ⚠ Cryptographic hash function authentication support is only available for NTP clients, not NTP servers.

To configure authentication on a FortiGate NTP client:

```
config system ntp
    set ntpsync enable
    set type custom
    set syncinterval 1
    config ntpserver
        edit "883502"
            set server "10.1.100.11"
            set authentication enable
            set key-type MD5
            set key **********
            set key-id 1
        next
    end
end
```

Command	Description
`authentication <enable | disable>`	Enable/disable MD5/SHA1/SHA256 authentication (default = disable).
`key-type`	Select NTP authentication type (default = MD5): • `MD5`: Enable MD5(NTPv3) authentication. • `SHA1`: Enable SHA1(NTPv4) authentication. • `SHA256`: Enable SHA256(NTPv4) authentication.
`key <passwd>`	Key for MD5(NTPv3)/SHA1(NTPv4)/SHA256(NTPv4) authentication. Enter a password value.
`key-id <integer>`	Key ID for authentication. Enter an integer value from 0 to 4294967295.

To confirm that NTP authentication is set up correctly:

```
# diagnose sys ntp status
synchronized: yes, ntpsync: enabled, server-mode: disabled
ipv4 server(10.1.100.11) 10.1.100.11 -- reachable(0xff) S:4 T:6 selected
server-version=4, stratum=3
```

If NTP authentication is set up correctly, the server version is equal to 4.

PTPv2

Precision Time Protocol (PTP) enables sub-microsecond clock synchronization across packet networks, far exceeding the accuracy achievable with NTP. FortiGate implements PTPv2 (IEEE 1588-2008), allowing it to synchronize its system clock to an upstream timeTransmitter or to act as a timing relay for downstream, timeReceiver devices. This is essential in deployments where high-precision time is required, such as financial trading networks, industrial automation, and telecommunications infrastructure.

Background

Prior to PTP support, FortiGate relied on NTP for clock synchronization, which provides accuracy in the millisecond range. PTPv2 support was introduced in FortiOS 6.2, adding hardware-timestamped, sub-microsecond synchronization as an alternative for time-sensitive environments. PTP and NTP are mutually exclusive on FortiGate and cannot run simultaneously.

How it works

FortiGate implements PTPv2 as defined in IEEE 1588-2008, using IP/UDP multicast transport (UDP ports 319 and 320). The protocol exchanges timestamped `Announce`, `Sync`, `Follow_Up`, `Delay_Req`, and `Delay_Resp` messages between the timeTransmitter and the timeReceiver to compute and correct the time offset. FortiGate supports two path delay mechanisms:

- End-to-End (E2E): measures total path delay between the timeTransmitter and the timeReceiver using `Delay_Req` and `Delay_Resp` messages.
- Peer-to-Peer (P2P): measures delay per link segment using `PDelay_Req` and `PDelay_Resp` messages exchanged between adjacent nodes; requires P2P Transparent Clock (TC) support on intermediate devices.

PTP is configured exclusively in the management VDOM. FortiGate synchronizes a single shared system clock; per-VDOM PTP configuration is not supported, as multiple sources competing to control the same clock would cause instability.

CLI syntax

Disable NTP (required before enabling PTP)

```
config system ntp
    set ntpsync disable
end
```

Command	Description
`ntpsync {enable | disable}`	Enable/disable NTP synchronization (default = disable).

Basic Syntax

```
config system ptp
    set status {enable | disable}
    set mode {multicast | hybrid}
    set delay-mechanism {E2E | P2P}
    set request-interval <integer>
    set interface <interface>
end
```

Command	Description
`status {enable | disable}`	Enable/disable the FortiGate system time by synchronizing with a PTP server (default = disable).
`mode {multicast | hybrid}`	Use multicast or hybrid transmission (default = multicast).
`delay-mechanism {E2E | P2P}`	Use end-to-end (E2E) or peer-to-peer (P2P) delay detection (default = E2E).
`request-interval <integer>`	The logarithmic mean interval between the delay request messages sent by the client to the server in seconds (default = 1).
`interface <interface>`	The interface that the PTP client will reply through.

Configure PTP — Ordinary Clock (OC)

```
config system ptp
    set status enable
    set interface "port1"
end
```

Command	Description
`status {enable | disable}`	Enable/disable the PTP daemon (default = disable).
`interface <port>`	Interface receiving PTP messages from the timeTransmitter; must have a valid IP address assigned.

Configure PTP — Boundary Clock (BC)

```
config system ptp
    set status enable
    set interface "port1"
    set server-mode enable
    config server-interface
        edit 1
            set server-interface-name "port2" "port3"
        next
    end
end
```

Command	Description
`status {enable | disable}`	Enable/disable the PTP daemon (default = disable).
`interface <port>`	Interface receiving PTP messages from the timeTransmitter; must have a valid IP address assigned.
`server-mode {enable | disable}`	Enable BC mode; FortiGate regenerates PTP messages on downstream server interfaces (default = disable).
`server-interface-name <port> [<port> ...]`	One or more downstream interfaces on which PTP Announce/Sync/Follow_Up are generated.

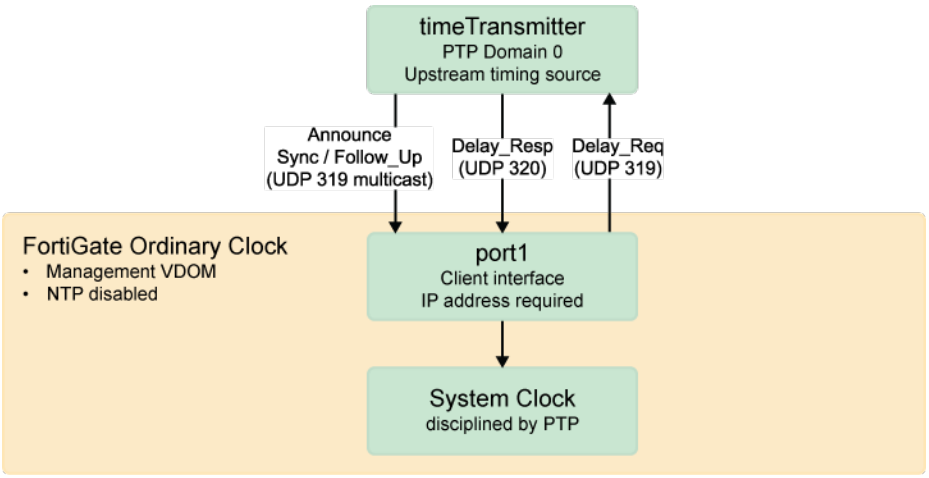
Scope and limitations

- PTP must be configured in the management VDOM only. Per-VDOM PTP is not supported.
- Transport: IP/UDP multicast only (UDP 319/320). L2 Ethernet multicast transport is not supported.
- PTP domain: Fixed to 0. No other domain value is configurable.
- NTP conflict: PTP and NTP cannot run simultaneously. NTP must be disabled before enabling PTP.
- The client interface defined under `set interface` must have a valid IP address assigned; the PTP daemon will not start without one.
- In P2P delay mode, all intermediate switching devices must support and have enabled P2P TC. Non-TC-aware devices will not update the Correction Field, resulting in incorrect offset calculation.

Example 1: FortiGate as an Ordinary Clock

FortiGate acts as a timeReceiver. It receives timing from an upstream timeTransmitter on port1, measures path delay via E2E (Delay_Req / Delay_Resp), and disciplines its system clock. No PTP messages are generated toward downstream devices.

Topology



Pre-requisites

- port1 must have a valid IP address assigned.
- NTP synchronization must be disabled.
- Upstream timeTransmitter configured for PTP domain 0, IP/UDP multicast transport.

Configuration

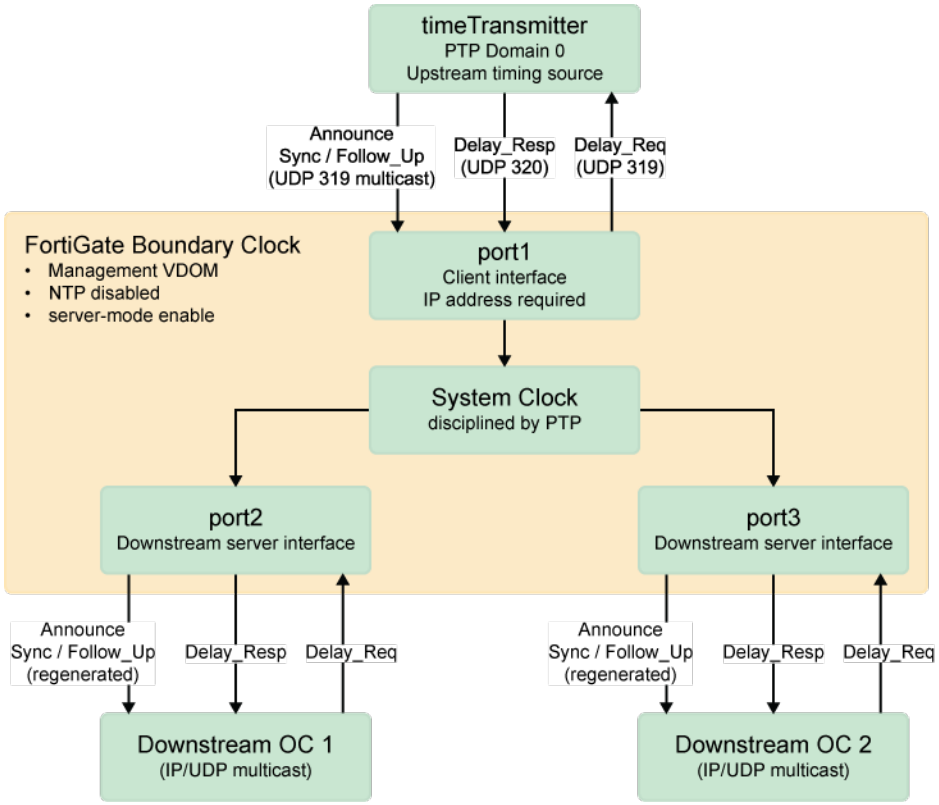
```
config system ntp
    set ntpsync disable
end
```

```
config system ptp
    set status enable
    set interface "port1"
end
```

Example 2: FortiGate as a Boundary Clock

FortiGate Synchronizes to the upstream timeTransmitter on port1 and regenerates PTP Announce and Sync/Follow_Up messages on port2 and port3 toward downstream Ordinary Clock devices. This role maintains PTP continuity across network segments.

Topology



Pre-requisites

- port1 (upstream), port2, port3 (downstream) must be correctly cabled.
- port1 must have a valid IP address assigned.
- NTP synchronization must be disabled.
- Firewall policies must permit PTP multicast (UDP 319/320) on all relevant interfaces.
- Downstream OCs must be configured for IP/UDP multicast transport.

Configuration

```
config system ntp
    set ntpsync disable
end
```

```
config system ptp
    set status enable
    set interface "port1"
    set server-mode enable
    config server-interface
        edit 1
            set server-interface-name "port2" "port3"
        next
    end
end
```

Troubleshooting and debugging

Diagnostic commands

Check current PTP state, selected timeTransmitter identity, and measured offset from the timeTransmitter:

```
diagnose sys ptp status
```

Enable real-time debug output from the PTP daemon:

```
diagnose debug application ptpd -1
diagnose debug enable
```

Disable debug output once the issue is identified:

```
diagnose debug disable
```

Capture PTP multicast traffic on a specific interface:

```
diagnose sniffer packet port1 "udp port 319 or udp port 320" 4
```

FortiGate does not synchronize to the timeTransmitter

NTP is still enabled

PTP and NTP cannot control the system clock simultaneously. Verify that NTP is off:

```
config system ntp
    set ntpsync disable
end
```

No IP address on the client interface

The PTP daemon requires an IP address on the interface defined under `set interface`.

Verify the interface has a valid IP assigned.

PTP multicast traffic is blocked

Verify that UDP 319 and 320 are not blocked by a firewall policy on the upstream path.

Wrong PTP domain on the timeTransmitter

FortiGate only supports domain 0. Confirm the timeTransmitteris configured for PTP domain 0.

PTP is configured but the system clock remains incorrect

Delay mechanism mismatch

The delay mechanism on FortiGate (E2E or P2P) must match the mechanism used by the timeTransmitter and all intermediate network devices. A mismatch causes failed delay calculations and incorrect offset correction.

P2P: non-TC-aware intermediate device

In P2P mode, all intermediate switching devices must support and have enabled P2P Transparent Clock. A device that does not update the Correction Field will cause the OC to compute an incorrect offset.

Upstream interface unreachable or not receiving PTP traffic

Confirm the interface defined under `set interface` is reachable and receiving PTP multicast. Use a packet capture to verify:

```
diagnose sniffer packet port1 "udp port 319 or udp port 320" 4
```

BC mode: downstream OCs do not synchronize

- Verify that `server-mode enable` is set.
- Confirm that the correct downstream interfaces are listed under `server-interface`.
- Confirm that firewall policies permit PTP multicast (UDP 319/320) on the downstream server interfaces.
- Confirm that downstream OCs are configured for IP/UDP multicast transport. FortiGate does not emit L2 Ethernet multicast PTP frames.

Email alerts

Alert emails are used to notify administrators about events on the FortiGate device, allowing a quick response to any issues.

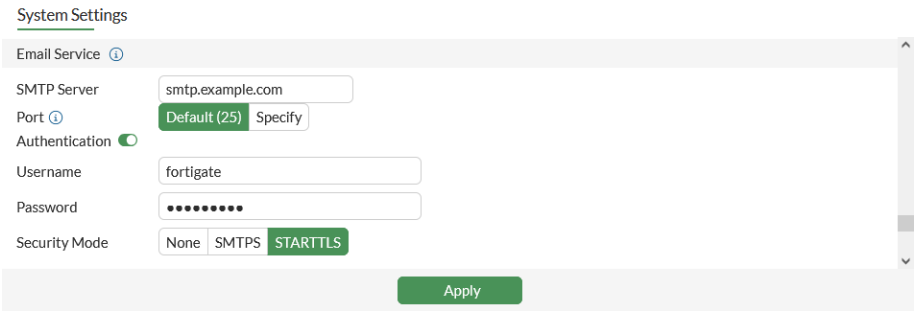
There are two methods that can be used to configure email alerts:

- Automation stitches on page 3807
- Alert emails on page 3809

The FortiGate has a default SMTP server, *fortinet-notifications.com*, that provides secure mail service with SMTPS. It is used for all emails that are sent by the FortiGate, including alert emails, automation stitch emails, and FortiToken Mobile activations. You can also configure a custom email service.

To configure a custom email service in the GUI:

1. Go to *System* > *Settings* > *General*.
2. In the *Email Service* section, enable *Use custom settings*.
3. Configure the following settings:



SMTP Server	If required, change the address or name of the SMTP server, such as *smtp.example.com*.
Port	If required, select *Specify* and enter a specific port number. The default is port depends on the selected security mode: *None* and *STARTTLS* default to port 25 and *SMTPS* defaults to 465.
Authentication	If required by the email server, enable authentication. If enabled, enter the *Username* and *Password*.
Security Mode	Set the security mode: *None*, *SMTPS*, or *STARTTLS*.

4. Click *Apply*.

To configure a custom email service in the CLI:

```
config system email-server
    set server "smtp.example.com"
    set port 465
    set authenticate enable
    set username "fortigate"
    set password *********
    set security smtps
end
```

The reply-to address for the source email is automatically set to *DoNotReply@fortinet-notifications.com* for all servers, including custom servers. For custom servers, if a username is configured, then MAIL FROM is set to the username, but if no username is configured, then MAIL FROM is the same as MAIL TO. You cannot customize the address when configuring a custom email server in the GUI or CLI.

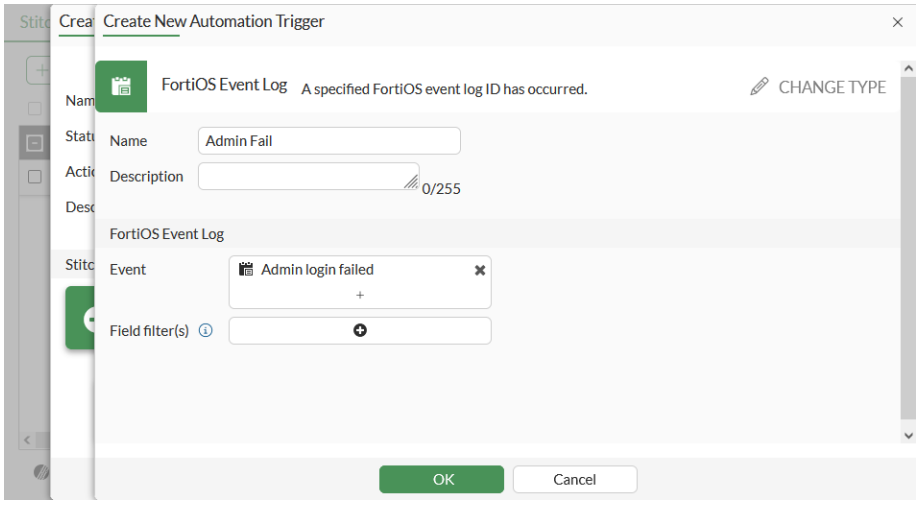
Automation stitches

Automation stitches can be configured to send emails based on a variety of triggers, giving you control over the events that cause an alert, and who gets alerted. For more information, see Automation stitches on page 4487.

In this example, the default mail service sends an email to two recipients when an Admin login failed event occurs or there is a configuration change.

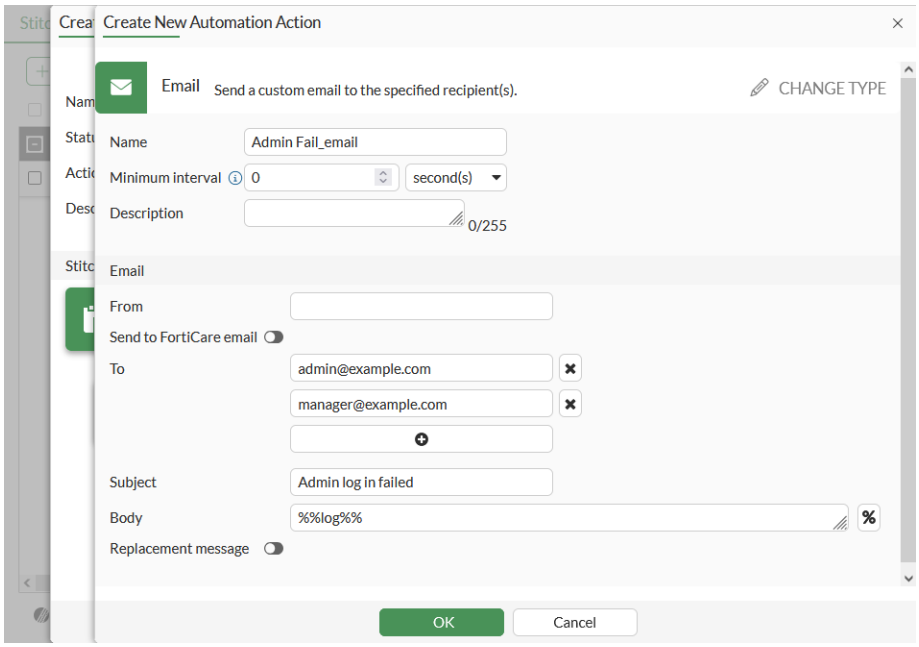
To configure the automation stitch in the GUI:

1. On the root FortiGate, go to *Security Fabric* > *Automation* and, on the *Stitch* tab, click *Create New*.
2. Enter a name for the stitch, such as *Admin Fail*.
3. Configure the trigger:
   a. Click *Add Trigger*.
   b. Click *Create* and select *FortiOS Event Log*.
   c. Enter a name for the trigger, such as *Admin Fail*.
   d. Click in the *Event* field, and in the slide out pane, search for and select *Admin login failed*.



   e. Click *OK*.
   f. Select the trigger in the list and click *Apply*.
4. Configure the action:
   a. Click *Add Action*.
   b. Click *Create* and select *Email*.
   c. Configure the following settings:

Name	Enter a name for the action, such as *Admin Fail_email*.
To	Enter the two email recipients' addresses, such as *admin@example.com* and *manager@example.com*.
Subject	Enter an subject, such as *Admin log in failed*.
Body	Edit as required. By default, the email body will include all the fields from the log event that triggered the stitch.



d. Click *OK*.

e. Select the action in the list and click *Apply*.

5. Click *OK*.
6. Create a second stitch with *Configuration Change* as the trigger, and an email action with a different subject line (such as *Configuration Change Detected*).

To configure the automation stitch in the CLI:

1. Create the automation triggers:

```
config system automation-trigger
    edit "Admin Fail"
        set event-type event-log
        set logid 32002
    next
    edit "Config Change"
        set event-type config-change
    next
end
```

2. Create automation actions to send the email messages:

```
config system automation-action
    edit "Admin Fail_email"
        set action-type email
        set email-to "admin@example.com" "manager@example.com"
        set email-subject "Admin log in failed"
    next
    edit "Config Change_email"
        set action-type email
        set email-to "admin@example.com" "manager@example.com"
```

```
        set email-subject "Configuration Change Detected"
    next
end
```

3. Create the automation stitches:

```
config system automation-stitch
    edit "Admin Fail"
        set trigger "Admin Fail"
        config actions
            edit 1
                set action "Admin Fail_email"
                set required enable
            next
        end
    next
    edit "Config Change"
        set trigger "Config Change"
        config actions
            edit 1
                set action "Config Change_email"
                set required enable
            next
        end
    next
end
```

Alert emails

When configuring an alert email, you can define the threshold when an issue becomes critical and requires attention. When the threshold is reached, an email is sent to up to three recipients on the configured schedule to notify them of the issue.

Alert email messages can be configured in the CLI. For more information on the available CLI commands, see Configure alert email settings.

> Alert email messages (under `config alertemail setting`) cannot monitor and notify users of the current logging status or the status of the miglogd daemon. In the event that the miglogd daemon is unresponsive, alert email messages cannot be triggered.

IPS, SSH, violation traffic, antivirus, and web filter logs are supported as triggers in automation stitches. For more information, see Event log category triggers on page 4521.

In this example, the FortiGate is configured to send email messages to two addresses, admin@example.com and manager@example.com, every two minutes when multiple intrusions, administrator log in or out events, or configuration changes occur.

To configure an alert email:

```
config alertemail setting
    set username fortigate@example.com
```

```
    set mailto1 admin@example.com
    set mailto2 manager@example.com
    set filter-mode category
    set email-interval 2
    set IPS-logs enable
    set configuration-changes-logs enable
    set admin-login-logs enable
end
```

Access tab

Use the *System* > *Settings* > *Access* tab to configure administration ports, single sign-on, and password security.

To configure the Access tab:

1. Go to *System* > *Settings* > *Access*. The *Access* tab is displayed.
2. Under *Administration settings*, configure the ports. See Configuring ports on page 3812.
3. Under *Single sign-in*, configure the following options:

Fabric SSO		Displays whether Fabric SSO is enabled. See Configuring single-sign-on in the Security Fabric on page 4456.
	Mode	Displays the mode of the FortiGate in the Fabric SSO.
FortiCloud SSO		Enable/disable FortiCloud SSO. See FortiCloud SSO on page 3715 and Allowing the FortiGate to override FortiCloud SSO administrator user permissions on page 3716.
FortiGate Cloud central management		Activate FortiGate Cloud. See also FortiCare and FortiGate Cloud login on page 115.

4. Under *Security*, configure the password policy. See Password policy on page 3721.
5. Click *Apply* to save the changes.

Default administrator password

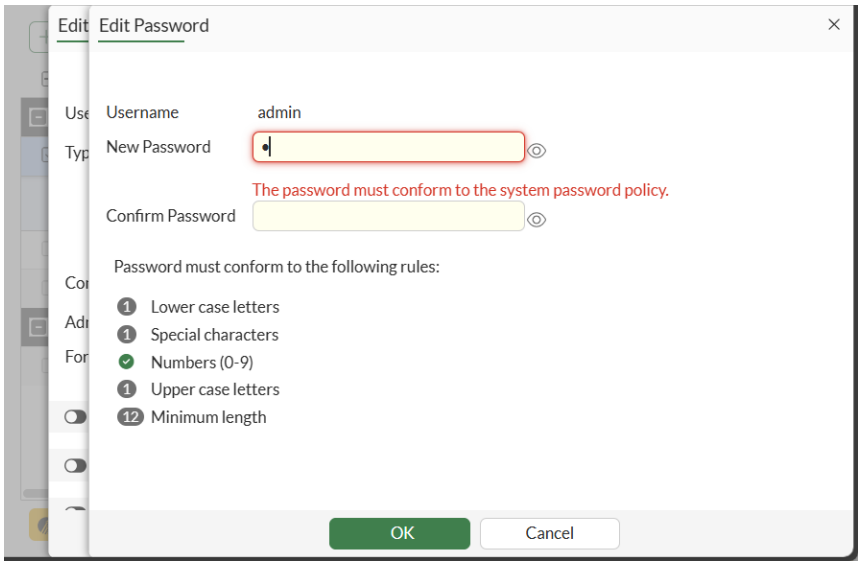
By default, your FortiGate has an administrator account set up with the username `admin` and no password.

Starting in FortiOS 7.6.5, the password policy is enabled by default and enforced, and your password must meet the policy requirements. If not, you are prompted to change your password when you log in to FortiOS. See Password policy on page 3721 for more information.

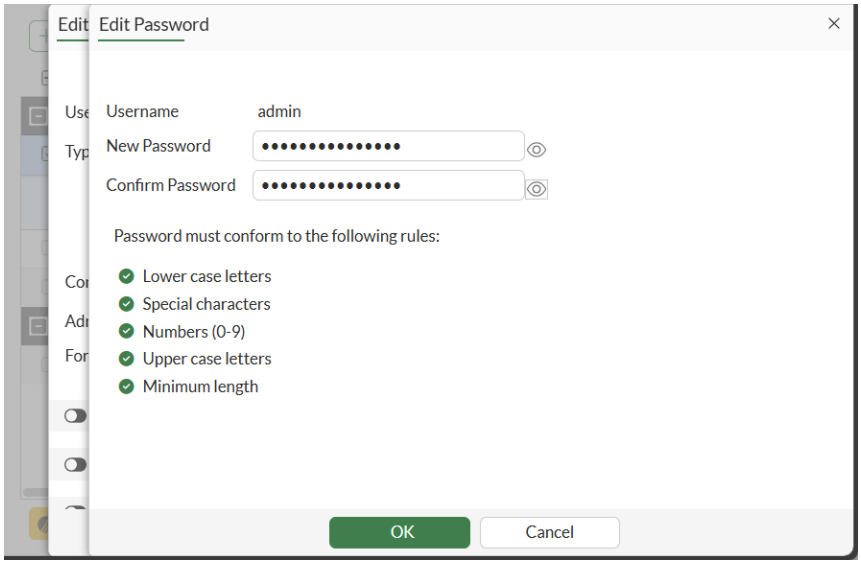
Adding a password to the *admin* administrator is mandatory. You will be prompted to configure it the first time you log in to the FortiGate using that account, after a factory reset, and after a new image installation.

To change the default password in the GUI:

1. Go to *System* > *Administrators*.
2. Edit the *admin* account.
3. Click *Change Password*.
4. Enter a password in the *New Password* field, then enter it again in the *Confirm Password* field.
   If the password does not conform to the password policy, an error is shown:



   If the password conforms to the password policy, no error message is shown:



5. Click *OK*.

To change the default password in the CLI:

```
config system admin
    edit admin
        set password <new password>
```

```
        Please enter current administrator password: <your password>
    next
end
```

It is also recommended that you change the user name of this account; however, since you cannot change the user name of an account that is currently in use, a second administrator account must be created in order to do this.

Configuring ports

To improve security, the default ports for administrative connections to the FortiGate can be changed. Port numbers must be unique. If a conflict exists with a particular port, a warning message is shown.

When connecting to the FortiGate after a port has been changed, the port number be included, for example: `https://192.168.1.99:100`.

Be aware of port conflicts with the IKE/IPsec TCP port. In 7.6.1 and later, the default IKE/IPsec TCP port is 443. If IPsec and GUI Management access are configured on the same interface, then IPsec access will take precedence.

See GUI warnings for IKE-TCP port conflicts on page 2811 for more details.

To configure the ports in the GUI:

1. Go to *System* > *Settings* > *Access*.
2. In the *Administration Settings* section, set the HTTP, HTTPS, SSH, and Telnet ports.
3. Enable *Redirect to HTTPS* to prevent HTTP from being used by administrators.
4. Click *Apply*.

To configure the ports in the CLI:

```
config system global
    set admin-port <port>
    set admin-sport <port>
    set admin-https-redirect {enable | disable}
    set admin-ssh-port <port>
    set admin-telnet-port <port>
end
```

Custom default service port range

The default service port range can be customized using the following CLI command:

```
config system global
    set default-service-source-port <port range>
end
```

Where <port range> is the new default service port range, that can have a minimum value of 0 and a maximum value up to 65535. The default value is 1 to 65535.

This change effects the TCP/UDP protocol.

Setting the idle timeout time

The idle timeout period is the amount of time that an administrator will stay logged in to the GUI without any activity. This is to prevent someone from accessing the FortiGate if the management PC is left unattended. By default, it is set to five minutes.

A setting of higher than 15 minutes will have a negative effect on a security rating score. See Security rating on page 4470 for more information.

To change the idle timeout in the GUI:

1. Go to *System* > *Settings* > *Access*.
2. In the *Administration Settings* section, set the *Idle timeout* to up to 480 minutes.
3. Click *Apply*.

To change the idle timeout in the CLI:

```
config system global
    set admintimeout <1-480>
end
```

Setting the password policy

Starting in FortiOS 7.6.5, the password policy is enabled by default and enforced for administrators, and your password must meet the policy requirements. See Password policy on page 3721 for information. You can edit the password policy and also enable a policy for IPsec pre-shared keys.

Setting the administrator password retries and lockout time

By default, the number password retry attempts is set to three, allowing the administrator a maximum of three attempts at logging in to their account before they are locked out for a set amount of time (by default, 60 seconds).

The number of attempts and the default wait time before the administrator can try to enter a password again can be configured using the CLI.

A maximum of ten retry attempts can be configured, and the lockout period can be 1 to 2147483647 seconds (over 68 years). The higher the retry attempts, the higher the risk that someone might be able to guess the password.

To configure the lockout options:

```
config system global
    set admin-lockout-threshold <failed_attempts>
    set admin-lockout-duration <seconds>
end
```

For example, to set the number of retry attempts to 1, and the lockout time to 5 minutes:

```
config system global
    set admin-lockout-threshold 1
    set admin-lockout-duration 300
end
```

If the time span between the first failed log in attempt and the lockout threshold failed attempt is less than lockout time, the lockout will be triggered.

TLS configuration

The minimum TLS version that is used for local out connections from the FortiGate can be configured in the CLI:

```
config system global
    set ssl-min-proto-version {SSLv3 | TLSv1 | TLSv1-1 | TLSv1-2 | TLSv1-3}
end
```

By default, the minimum version is TLSv1.2. The FortiGate will try to negotiate a connection using the configured version or higher. If the server that FortiGate is connecting to does not support the version, then the connection will not be made. Some FortiCloud and FortiGuard services do not support TLSv1.3.

Minimum SSL/TLS versions can also be configured individually for the following settings, not all of which support TLSv1.3:

Setting	CLI
Email server	`config system email-server`
Certificate	`config vpn certificate setting`
FortiSandbox	`config system fortisandbox`
FortiCloud	`config log fortiguard setting`
FortiAnalyzer	`config log fortianalyzer setting`
Syslog	`config log syslogd setting`
User Authentication	`config user setting`
LDAP server	`config user ldap`
POP3 server	`config user pop3`
Exchange server	`config user exchange`

A minimum (`ssl-min-proto-ver`) and a maximum (`ssl-max-proto-ver`) version can be configured for Agentless VPN. See TLS 1.3 support on page 3371

Controlling return path with auxiliary session

When multiple incoming or outgoing interfaces are used in ECMP or for load balancing, changes to routing, incoming, or return traffic interfaces impacts how an existing sessions handles the traffic. Auxiliary sessions can be used to handle these changes to traffic patterns.

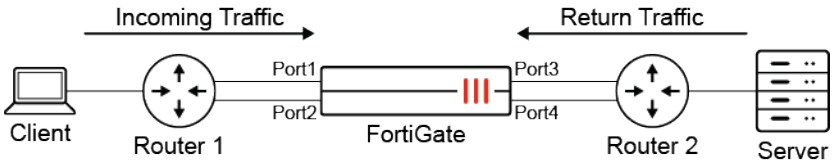
- In FortiOS 6.0 and earlier, the auxiliary session feature is not supported.
- In FortiOS 6.2.0 to 6.2.2, the auxiliary session feature is permanently enabled.
- In FortiOS 6.2.3 and later, the auxiliary session feature is disabled by default, and can be enabled if required.

To enable or disable the auxiliary session feature:

```
config system settings
    set auxiliary-session {enable | disable*}
end
```

When enabling auxiliary sessions, consider the impact of routing in both traffic directions. In topologies such as SD-WAN hub and spoke or ADVPN deployments, the symmetry of the return traffic is important for maintaining the stability of the session. It is expected that the spoke selects the outbound interface and path, and the other nodes obey and reply symmetrically. It is recommended to disable auxiliary in these scenarios, and others where incoming and return traffic symmetry is expected.

Scenarios



Incoming traffic is from the client to the server. Return traffic is from the server to the client.

Scenario 1 - Return traffic returns on the original outgoing interface

In this scenario, a session is established between port1 and port3. When the return traffic hits port3:

Auxiliary sessions disabled:

The reply to the client egresses on the original incoming interface, port1. If policy routes or SD-WAN rules are configured, the next hop gateway is applied if the output device is the same as the original incoming interface.

Auxiliary sessions enabled:

The reply to the client egresses on the best route in the routing table:

- If the best route is port1, then it will egress on port1.
- If the best route is port2, then it will egress on port2.

If policy routes or SD-WAN rules are configured, they must be matched to determine the egress interface. If both are configured, policy routes have higher priority.

Scenario 2 - Return traffic returns on an interfaces other than the original outgoing interfaces

In this scenario, a session is established between port1 and port3. When the return traffic hits port4:

Auxiliary sessions disabled:

- The session is dirtied and then gets refreshed, and interfaces on the session are updated. This continuous state change to dirty prevents the session from being offloaded.
- If there is a high traffic volume or flapping between the interfaces, the CPU usage increases.

Auxiliary sessions enabled:

An auxiliary session is created for the existing session, and traffic returns to the client as normal on the auxiliary session.

Scenario 3 - Incoming traffic enters on an interfaces other than the original incoming interfaces

In this scenario, a session is established between port1 and port3. When the incoming traffic hits port2:

Auxiliary sessions disabled:

The session is dirtied and then gets refreshed, and interfaces on the session are updated. This continuous state change to dirty prevents the session from being offloaded.

Auxiliary sessions enabled:

An auxiliary session is created for the existing session, and traffic is forwarded to the server as normal on the auxiliary session.

Scenario 4 - the routing table is changed

In this scenario, a session has been established between port1 and port3, when a new route on port4 is updated as the route to the server.

Auxiliary sessions disabled:

As long as there is a route to the destination, the session will not be dirtied or refreshed. Even though there is a better route, traffic continues on the original path between port1 and port3.

Auxiliary sessions enabled:

The session is dirtied and then gets refreshed, and interfaces on the session are updated. This continuous state change to dirty prevents the session from being offloaded.

Effect on NPU offloading sessions

When the auxiliary session feature is disabled, there is always one session. If the incoming or return interface changes, the FortiGate marks the session as dirty and updates the session's interfaces. This cannot be done by the NPU, so the session is not offloaded to the NPU, and is processed by the CPU instead. If Equal-Cost Multi-Path (ECMP) causes the interface to keep changing, then it will use significant CPU resources.

When the auxiliary session feature is enabled and the incoming or return interface changes, it creates an auxiliary session, and all traffic can continue to be processed by the NPU.

Verification

When an auxiliary, or reflect, session is created, it will appear as a reflect session below the existing session:

```
# diagnose sys session list
session info: proto=17 proto_state=00 duration=111 expire=175 timeout=0 flags=00000000 socktype=0
sockport=0 av_idx=0 use=4
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty npu
statistic(bytes/packets/allow_err): org=131/4/1 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=36->38/38->36 gwy=10.1.2.3/0.0.0.0
hook=pre dir=org act=noop 10.1.100.22:51926->172.16.204.44:5001(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.204.44:5001->10.1.100.22:51926(0.0.0.0:0)
src_mac=90:6c:ac:19:19:58
misc=0 policy_id=1 auth_info=0 chk_client_info=0 vd=2
serial=00002b11 tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id=00000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x000400
npu info: flag=0x91/0x00, offload=8/0, ips_offload=0/0, epid=129/0, ipid=142/0, vlan=0x0016/0x0000
vlifid=142/0, vtag_in=0x0016/0x0000 in_npu=1/0, out_npu=1/0, fwd_en=0/0, qid=4/0
no_ofld_reason:
reflect info 0:
dev=37->38/38->37
npu_state=0x000400
npu info: flag=0x91/0x00, offload=8/0, ips_offload=0/0, epid=129/0, ipid=142/0, vlan=0x0017/0x0000
vlifid=142/0, vtag_in=0x0017/0x0000 in_npu=1/0, out_npu=1/0, fwd_en=0/0, qid=4/0
total reflect session num: 1
total session 1
```

When a session is dirtied, a dirty flag is added to it:

```
# diagnose sys session list
session info: proto=17 proto_state=00 duration=28 expire=152 timeout=0 flags=00000000 socktype=0
sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=dirty may_dirty npu
statistic(bytes/packets/allow_err): org=68/2/1 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 2/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=0->0/0->0 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 10.1.100.22:51926->172.16.204.44:5001(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.204.44:5001->10.1.100.22:51926(0.0.0.0:0)
src_mac=90:6c:ac:19:19:58  dst_mac=02:6c:ac:5c:c6:f9
misc=0 policy_id=1 auth_info=0 chk_client_info=0 vd=2
serial=00002b2c tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id=00000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x000400
npu info: flag=0x00/0x00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0, vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
no_ofld_reason:
total session 1
```

When an auxiliary session is created, NPU offloading will continue in the reflect session:

```
# diagnose sys session list
session info: proto=17 proto_state=01 duration=169 expire=129 timeout=0 flags=00000000 socktype=0
sockport=0 av_idx=0 use=4
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty npu
statistic(bytes/packets/allow_err): org=131/4/1 reply=66/2/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=36->38/38->36 gwy=10.1.2.3/172.17.2.1
hook=pre dir=org act=noop 10.1.100.22:51926->172.16.204.44:5001(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.204.44:5001->10.1.100.22:51926(0.0.0.0:0)
src_mac=90:6c:ac:19:19:58
misc=0 policy_id=1 auth_info=0 chk_client_info=0 vd=2
serial=00002b11 tos=ff/ff app_list=0 app=0 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id=00000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x000c00
npu info: flag=0x91/0x81, offload=8/8, ips_offload=0/0, epid=129/142, ipid=142/128,
vlan=0x0016/0x0016
vlifid=142/128, vtag_in=0x0016/0x0016 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=4/4
reflect info 0:
dev=37->38/38->37
npu_state=0x000400
npu info: flag=0x91/0x00, offload=8/0, ips_offload=0/0, epid=129/0, ipid=142/0, vlan=0x0017/0x0000
vlifid=142/0, vtag_in=0x0017/0x0000 in_npu=1/0, out_npu=1/0, fwd_en=0/0, qid=4/0
```

```
total reflect session num: 1
total session 1
```

Using the default certificate for HTTPS administrative access

By default, the FortiGate uses the certificate named Fortinet_GUI_Server for HTTPS administrative access. This certificate is generated and signed by the built-in Fortinet_CA_SSL certificate, which dynamically updates the SAN field of the Fortinet_GUI_Server certificate with the IP addresses of all interfaces enabled for HTTPS. After installing the Fortinet_CA_SSL CA certificate on a PC, administrators can access the FortiGate GUI through a browser without any warnings.

How the certificate works

The Fortinet_GUI_Server certificate is generated by the built-in certificate authority (CA) with the Fortinet_CA_SSL certificate, which is unique to each FortiGate. This CA certificate is also used in SSL deep inspection. When the Fortinet_GUI_Server certificate is generated, the SAN (Subject Alternative Name) extension field is populated with the IP addresses of all physical and logical (VLAN, loopback, and so on) interfaces enabled for HTTPS. It is also populated with the management IP address whenever this field is an IP address and not an FQDN. If there are any changes to the IP addresses on the interface or management IP, the Fortinet_GUI_Server certificate is updated and regenerated with the new IP. If the Fortinet_CA_SSL certificate itself is updated, the Fortinet_GUI_Server certificate is regenerated.

Because the root CA is not a public CA, the Fortinet_CA_SSL CA certificate must be installed in the trusted certificate store on the client PC in order for the trusted certificate chain to be recognized by a browser. This certificate can be downloaded from the FortiGate in several ways.

The Fortinet_GUI_Server certificate can only be used for HTTPS administrative access. It cannot be used anywhere else.

Example

The HTTPS server certificate can be configured in the GUI or CLI.

To configure the HTTPS server certificate in the GUI:

1. On an administrative PC, log in to the FortiGate GUI and go to *System > Settings > Access*.
2. In the *Administration Settings* section, set the *HTTPS server certificate* to *Fortinet_GUI_Server*.
3. Download the Fortinet_CA_SSL certificate using one of the following methods:
   - On the *System > Settings > Access* page, click *Download HTTPS CA certificate* (below the *HTTPS server certificate* option).
   - Go to *System > Certificates*. In the *Local CA Certificate* section, select *Fortinet_CA_SSL*, and click *Download*.
   - Go to *Dashboard > Status*. In the *Administrator* widget, click *Download HTTPS CA certificate*.
4. Install the certificate in the PC's trusted certificate store. Refer to your OS documentation if needed.
5. Reload the FortiGate GUI. The browser now trusts the certificate and does not display a certificate warning.

6. If you are connecting to the FortiGate over DNAT or port forwarding, the certificate needs to add the NATed management IP into the SAN field so that the browser does not display a warning about an invalid CN. Configure the management IP in the global settings:

```
config system global
    set management-ip <IP_address>
end
```

To configure the HTTPS server certificate in the CLI:

1. Configure the HTTPS server certificate:

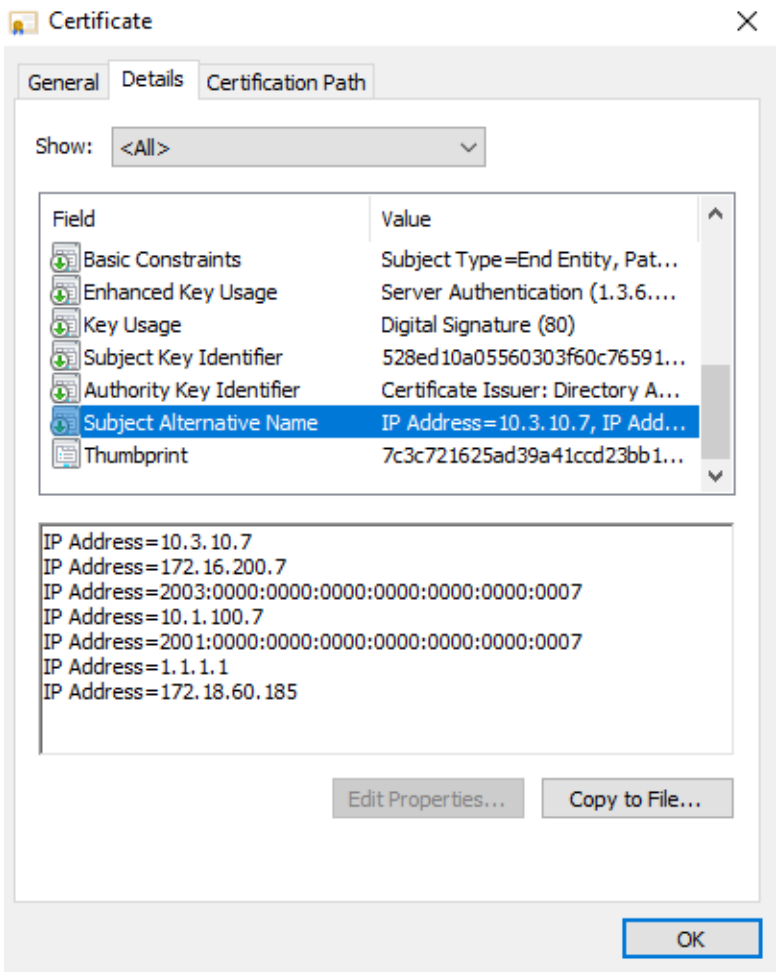
```
config system global
    set admin-server-cert Fortinet_GUI_Server
end
```

2. Download the Fortinet_CA_SSL certificate on the administrative PC through TFTP:

```
# execute vpn certificate local export tftp Fortinet_CA_SSL cer <file_name> <server_IP>

Done.
```

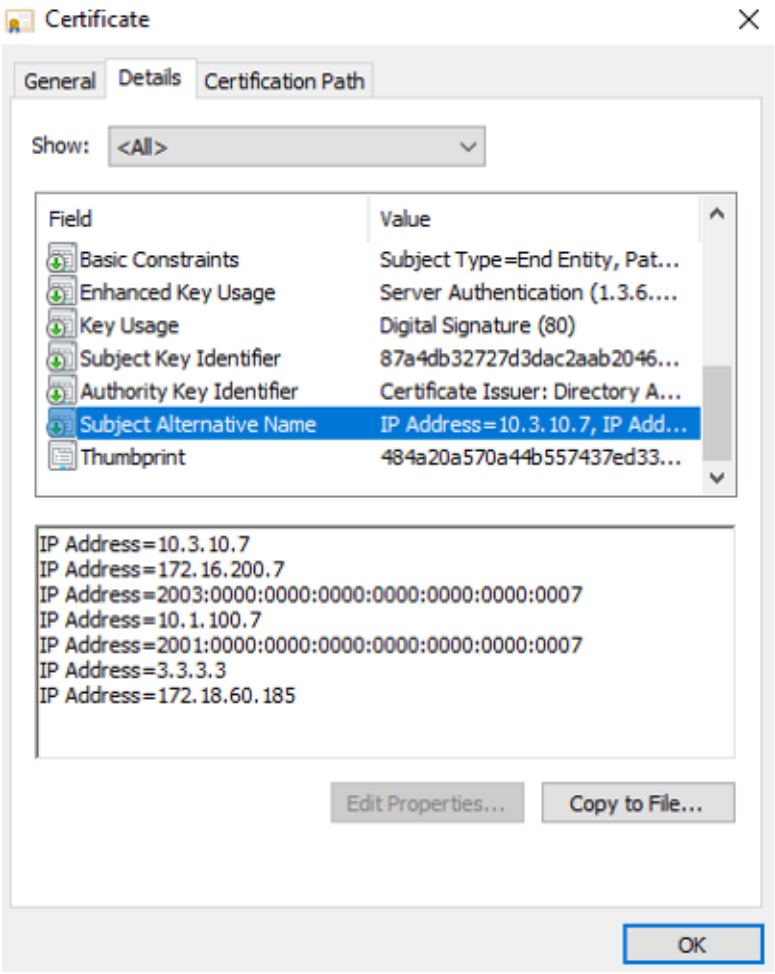
To verify the connection:

1. Access the FortiGate from a browser and verify the certificate information. For example, in Chrome:
   a. In the left side of the address bar, click the icon to view the site information.
   b. Click *Certificate*.
   c. Click the *Details* tab.
   d. Locate the *Subject Alternative Name* (SAN) field, and note the IP addresses that are listed (*1.1.1.1*).



e. Click *OK*.

2. In FortiOS, change one of the interface addresses. In this example, the port11 address is changed from 1.1.1.1 to 3.3.3.3.
3. Reload the browser and review the certificate information again. The IP 1.1.1.1 in the SAN field is updated to *3.3.3.3*.



4. In FortiOS, go to *System > Certificates* and double-click *Fortinet_GUI_Server* to view the *Certificate Details*.
5. At the bottom of the pane, the *X509v3 Subject Alternative Name* field displays the IP addresses from the certificate.
6. Verify the logs when the certificate is regenerated:

```
# execute log filter category 1
# execute log display
12 logs found.
10 logs returned.
```

```
1: date=2022-06-23 time=09:11:44 eventtime=1656000704674434910 tz="-0700" logid="0100022205"
type="event" subtype="system" level="information" vd="root" logdesc="Certificate succeed to
auto-generate" user="system" action="certificate-generate" status="successful" name="Fortinet_
GUI_Server" msg="Successfully generated GUI management cert"
```

```
2: date=2022-06-23 time=09:11:44 eventtime=1656000704674432668 tz="-0700" logid="0101041986"
type="event" subtype="vpn" level="information" vd="root" logdesc="Certificate regenerated"
action="info" user="N/A" ui="forticron" name="Fortinet_GUI_Server" msg="A certificate is
regenerated" cert-type="Local" status="success"
```

```
3: date=2022-06-23 time=09:11:31 eventtime=1656000691825397831 tz="-0700" logid="0100044547"
type="event" subtype="system" level="information" vd="root" logdesc="Object attribute
configured" user="admin" ui="ssh(172.16.200.254)" action="Edit" cfgtid=35782662
cfgpath="system.interface" cfgobj="port11" cfgattr="ip[1.1.1.1 255.255.255.0->3.3.3.3
255.255.255.0]" msg="Edit system.interface port11"
```

Device tab

Use the *System > Settings > Device* tab to configure startup settings, such as FortiConverter access and whether firmware can be installed from a USB stick.

To configure the Device tab:

1. Go to *System > Settings > Device*. The *Device* tab is displayed.
2. Under *Start up settings*, configure the following options:

Allow FortiConverter to obtain config file once	Enable/disable FortiConverter to obtain a configuration file from FortiGate once. See also Migrating a configuration with FortiConverter on page 101.
USB auto-install	On restart, automatically update the named firmware and/or configuration if found on the connected USB key. See Restoring from a USB drive on page 3780.
Detect configuration	Enter the name of the configuration file.
Detect firmware	Enter the name of the firmware file.

3. Under *Disk settings*, view the disk usage setting for logging and WAN optimization.
   See Disk usage on page 4868 for information.
4. Click *Apply* to save the changes.

Preference tab

Use the *System > Settings > Preference* tab to configure workflow management, language, and themes.

To configure the Preference tab:

1. Go to *System > Settings > Preference*. The *Preference* tab is displayed.
2. Under *Workflow management*, configure the following options:

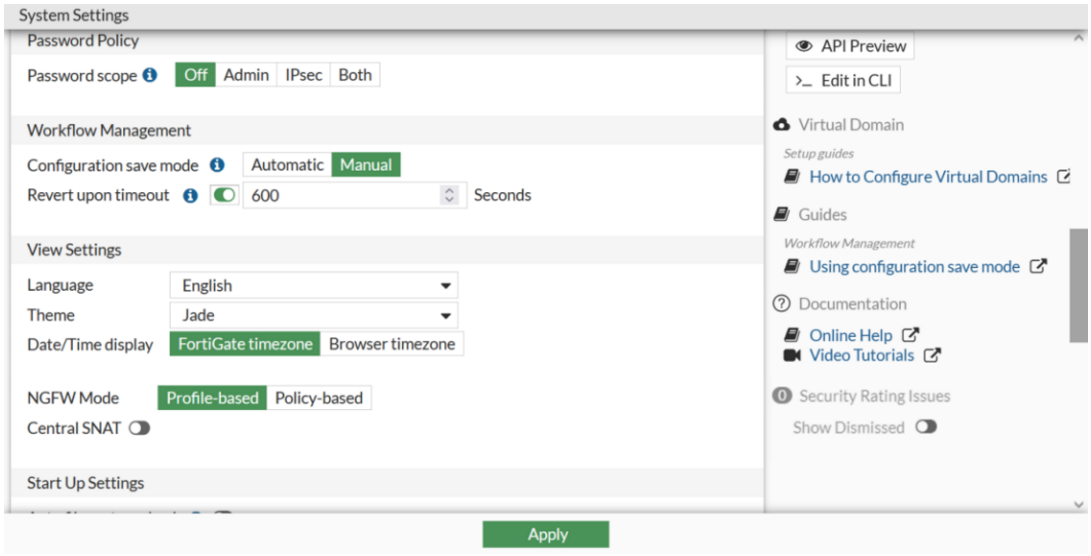
Configuration save mode	• *Automatic*: Automatically saves configuration changes. • *Manual*: Stage configuration changes and save them once you are ready. • *Revert*: Pending configuration changes are reverted after the configured number of seconds.

	See Using configuration save mode on page 3824.
Policy change summary	Enable/disable policy change summary. Enable and choose: • *Require*: A change summary is required. • *Optional*: A change summary is optional. When enabled, enforcement is only applied in the GUI. See Add Policy change summary and Policy expiration to Workflow Management on page 1850 for more information.
Policies expire by default	Enable/disable policy expiration.
Expire after	When *Policies expire by default* is enabled, specify how many days to retain the policy before it expires.

3. Under *Language*, select the language for the FortiOS GUI and CLI. See Setting the FortiOS language on page 3825.
4. Under *Theme*, set *System theme* to a color theme. You can also create custom themes. See Setting GUI color themes NEW on page 3827.
5. Click *Apply* to save the changes.

Using configuration save mode

Administrators can use configuration save mode set to *Manual* to implement strict change control by requiring changes to be manually committed to the flash. To configure the setting in the GUI, go to *System* > *Settings* > *Preference*.

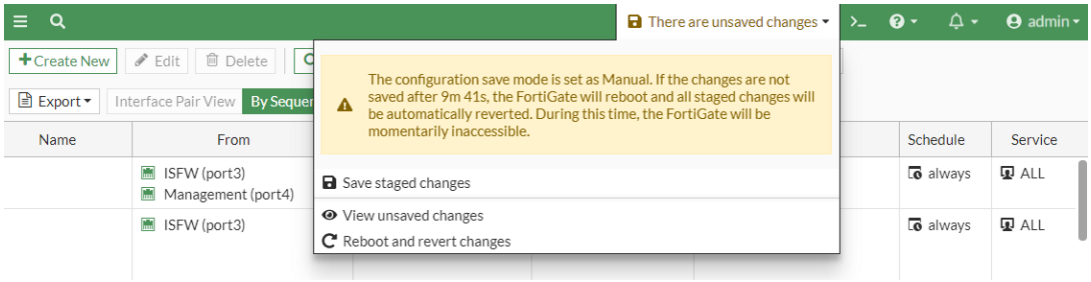


When *Configuration save mode* is set to *Automatic* (default), configuration changes are automatically saved to both memory and flash.

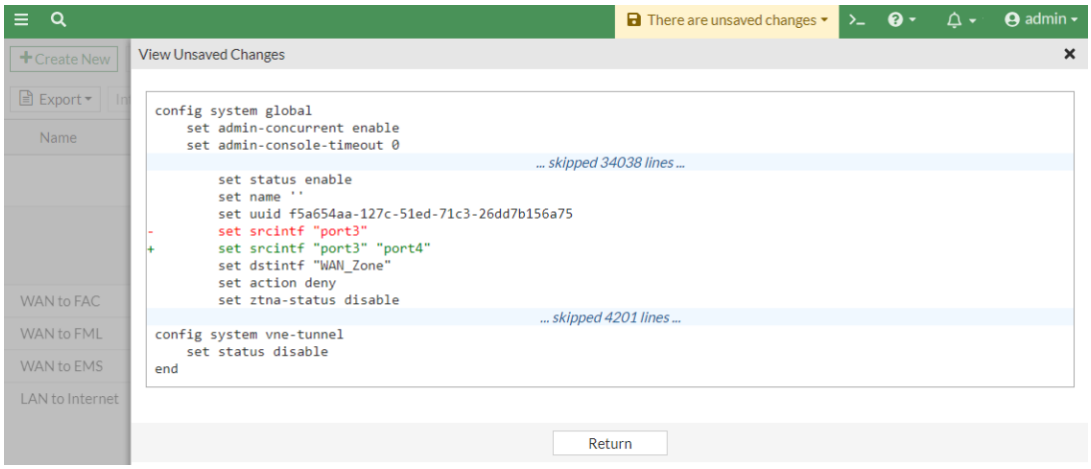
When *Configuration save mode* is set to *Manual*, configuration changes are saved to memory, but not to flash. The changes take effect immediately, but must be manually saved to flash. Unsaved changes are reverted when the device is rebooted. If *Revert upon timeout* is enabled, the system might be unresponsive for a short time after the configured timeout while it reverts the changes back to the previous save point. Prior to the timeout

expiring, a pop-up warning gives you the option to postpone reverting the configuration by one minute, revert the configuration immediately, or save the configuration changes.

In *Manual* mode, a warning is shown in the banner when there are unsaved changes. Click the warning to save, view, or revert the changes. When *Reboot and revert changes* is clicked, the system might be unresponsive for a short time while it reverts the changes back to the previous save point.



Clicking *View Unsaved Changes* opens a pane highlighting the changes that have not been committed.



This feature is also available in the CLI:

```
config system global
    set cfg-save {automatic | manual | revert}
    set cfg-revert-timeout <integer>
end
```

```
# execute cfg {reload | save}
```

Setting the FortiOS language

FortiOS GUI and CLI support several languages.

Custom languages can also be added for Agentless VPN portals. See Custom languages on page 3826.

To set the language in the GUI:

1. Go to *System* > *Settings* > *Preference*.
2. Set *Language* to one of the following options:

English, *French*, *Spanish*, *Portuguese*, *Japanese*, *Traditional Chinese*, *Simplified Chinese*, *Korean*.

3. Click *Apply*.

To set the language in the CLI:

```
config system global
    set language {english | french | spanish | portuguese | japanese | trach | simch | korean}
end
```

Custom languages

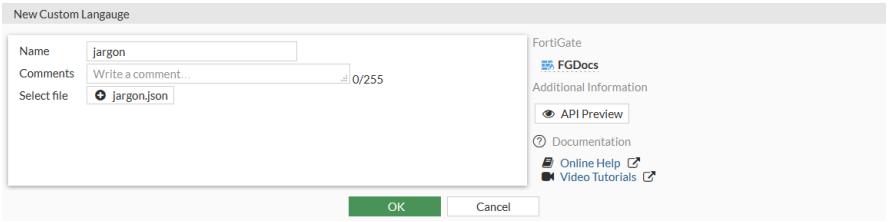
Custom languages can be uploaded and used for Agentless VPN portals. Custom languages must be enabled before they can be added in the GUI.

To enable custom languages:

```
config system global
    set gui-custom-language enable
end
```

To configure a custom language in the GUI:

1. Go to *System* > *Custom Languages* and click *Create New*.
2. Enter the name of the language.
3. Optionally, enter a comment.
4. Click *Upload* and upload the language JSON file from your management computer.



5. Click *OK*.

To configure a language for Agentless VPN portals in the GUI:

1. Go to *VPN* > *Agentless VPN Settings*.
2. Set *Language* to *Browser preference* or *System*.
3. Click *OK*.

To configure a custom language in the CLI:

```
config system custom-language
    edit <language>
        set filename <file>
```

```
    next
end
```

To configure a language in an Agentless VPN portal in the CLI:

```
config vpn ssl web portal
    edit <portal>
        set web-mode enable
        set custom-lang <language>
    next
end
```

Setting GUI color themes NEW

FortiOS include several predefined color themes for the GUI. You can also create custom color themes.

Predefined and custom themes can be associated with the FortiOS system, admin profiles, or individual administrators.

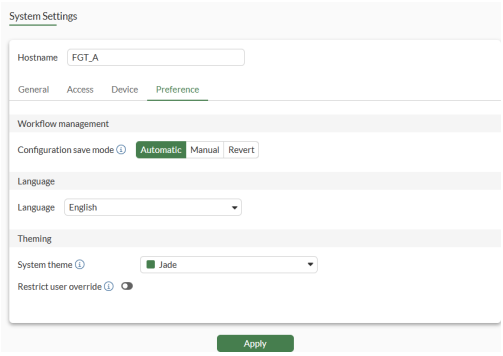
Banner messages associated with a severity can be created for custom themes using the CLI.

Creating custom color themes

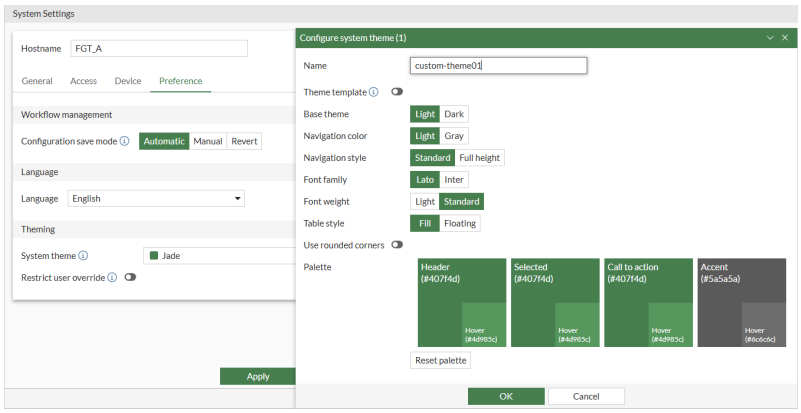
You can create one or more custom GUI themes for FortiOS. Then you can assign different predefined or custom themes to the system, admin profiles, and administrators.

To create a custom theme in the GUI:

1. Go to *System* > *Settings* > *Preference*.



2. Beside *System theme*, click the list, and select Create (*+*). The *Configure system theme* pane is displayed.



3. (Optional) Enable *Theme template* and select an existing template to customize.
4. Set the options, and click *OK*. The custom theme is selected.
5. (Optional) *Enable Restrict user override* to restrict theme customization to administrators assigned the super_admin profile.
6. Click *Apply* to apply the changes to the GUI.

To create a custom theme in the CLI:

```
config system theme
    edit "custom theme 1"
        set nav-color gray
        set nav-style full-height
        set font inter
        set font-weight light
        set table-style float
        set border-radius enable
        set header-color "#e64747"
        set selected-color "#2cbaa2"
        set call-to-action-color "#ba2cb5"
        set accent-color "#df1616"
        set banner-msg ''
    next
end
```

Assigning color themes

You can assign predefined and custom themes to the system, admin profiles, and administrators.

Administrators can also choose a theme.

To assign a theme to an administrator profile in the GUI:

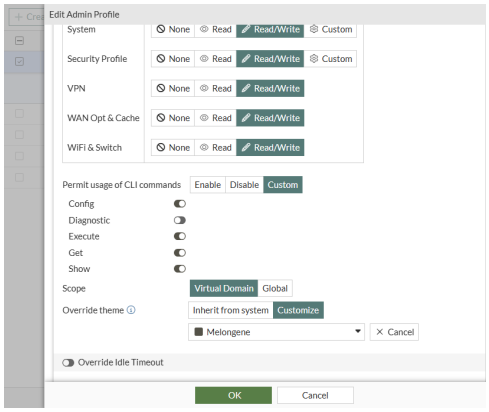
1. Go to *System* > *Admin Profiles*.
2. Open an existing administrator profile for editing, or click *Create new* to create a new profile.
3. Set *Override Theme* to *Customize* and select a theme, or click Create (+) to create a custom theme.
4. Click *OK* to save the settings.
   The selected theme is used for administrators assigned this profile.

To assign a theme to an administrator profile in the CLI:

```
config system accprofile
    edit "prof_admin"
        set gui-theme custom
        set gui-custom-theme custom theme 1
    next
end
```

To assign a theme to individual administrators in the GUI:

1. Go to *System* > *Administrators* and double-click an administrator to open it for editing.
2. Set *Override Theme* to *Customize* and select a theme, or click Create (*+*) to create a custom theme.



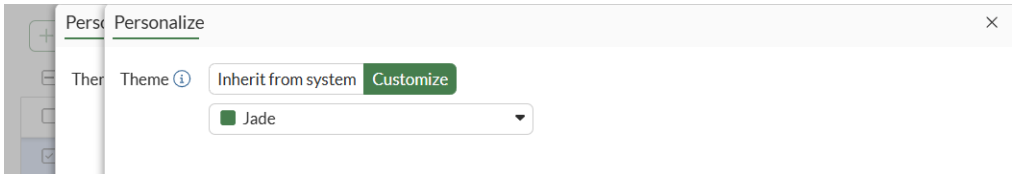
3. Click *OK* to save the settings.
   The selected theme is used for the administrator.

To assign a custom theme to individual administrators in the CLI:

```
config system admin
    edit "admin"
        set gui-theme custom
        set gui-custom-theme custom theme 1
    next
end
```

To personalize the GUI:

1. From the top-right corner, click your administrator name and select *Personalize*. The *Personalize* dialog box is displayed.



2. Set *Theme* to *Customize* and select a theme, or click Create (*+*) to create a custom theme.
3. Click *OK* to save the settings.

The selected theme is used when you log in to FortiOS.

Configuring a banner

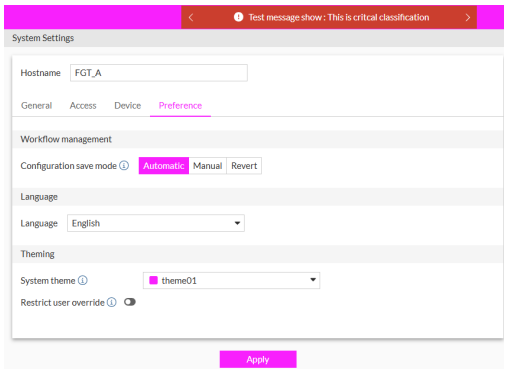
A banner can be configured for a custom color theme using the CLI.

To configure a banner for a custom theme in the CLI:

1. Configure the banner message and severity:

```
config system theme
    edit theme01
        set banner-msg "Test message show: This is critical classification"
        set banner-msg-severity critical
    next
end
```

The message displays in the banner for the custom theme:



Trusted platform module support

On supported FortiGate hardware devices, the Trusted Platform Module (TPM) can be used to protect your password and key against malicious software and phishing attacks. The dedicated module hardens the FortiGate by generating, storing, and authenticating cryptographic keys. To help prevent tampering, the chip is soldered on the motherboard to reduce the risk of data transaction interceptions from attackers.

By default, the TPM is disabled. When enabling private data encryption, the FortiGate will generate a random master-encryption-password that encrypts sensitive data on the FortiGate using AES128-CBC. With the password, TPM generates a 2048-bit primary key to secure the master-encryption-password through RSA-2048 encryption. The master-encryption-password protects the data. The primary key protects the master-encryption-password.

The TPM module does not encrypt the disk drive of eligible FortiGates.

The primary key binds the encrypted configuration file to a specific FortiGate unit and never leaves the TPM. When backing up the configuration, the TPM uses the primary key to encrypt the master-encryption-password

in the configuration file. When restoring a configuration that includes a TPM protected master-encryption-password:

- If TPM is disabled, then the configuration cannot be restored.
- If TPM is enabled but has a different master-encryption-password than the configuration file, then the configuration cannot be restored.
- If TPM is enabled and the master-encryption-password is the same in the configuration file, then the configuration can be restored.

For information on backing up and restoring the configuration, see Configuration backups and reset on page 4053.

Passwords and keys that can be encrypted by the master-encryption-key include:

- Alert email user's password
- BGP and other routing related configurations
- External resource
- FortiGuard proxy password
- FortiToken/FortiToken Mobile’s seed
- HA password
- IPsec pre-shared key
- Link Monitor, server side password
- Local certificate's private key
- Local, LDAP. RADIUS, FSSO, and other user category related passwords
- Modem/PPPoE
- NST password
- NTP Password
- SDN connector, server side password
- SNMP
- Wireless Security related password

To check if your FortiGate device has a TPM:

Verify all the following commands exist. Otherwise, the platform does not support it.

```
# diagnose hardware test info
List of test cases:
    bios: sysid
    bios: checksum
    bios: license
    bios: detect

# diagnose hardware deviceinfo tpm
TPM capability information of fixed properties:
========================================================
TPM_PT_FAMILY_INDICATOR: 2.0
TPM_PT_LEVEL: 0
TPM_PT_REVISION: 138
TPM_PT_DAY_OF_YEAR: 8
TPM_PT_YEAR: 2018
```

```
TPM_PT_MANUFACTURER: NTC
# diagnose hardware test tpm
========== Fortinet Hardware Test Report ==================
TPM
TPM Device Detection..................................... PASS
================ Fortinet Hardware Test PASSED ==============
# diagnose tpm
get-property Get TPM properties. [Take 0-1 arg(s)]
get-var-property Get TPM var properties.
read-clock Read TPM internal clock.
shutdown-prepare Prepare for TPM power cycle.
selftest Perform self tests.
generate-random-number Generate a 4-byte random number
SHA-1 HASH a sequence of num with SHA-1 algo
SHA-256 HASH a sequence of num with SHA-256 algo
```

To enable TPM and input the master-encryption-password:

```
config system global
    set private-data-encryption enable
end
This operation will generate a random private data encryption key!
Previous config files encrypted with the system default key cannot be restored after this
operation!
Do you want to continue? (y/n)y
Private data encryption key generation succeeded!
```

When enabling private-data-encryption the FortiGate will generate a random password itself. This increases security as the master-encryption-password is not known and cannot be stolen or leaked.

As such, a configuration that is backed up while private-data-encryption is enabled cannot be restored when private-data-encryption is disabled or when private-data-encryption is re-enabled with a different random key.

For FortiGates managed by FortiManager, refer to Verifying devices with private data encryption enabled in the FortiManager Administration Guide for steps on managing the FortiGate.

HA behavior

HA will form when both units have the basic matching HA required settings, including the HA group and HA password, regardless of whether private-data-encryption is enabled on both units before or after forming HA.

If private-data-encryption is enabled separately before forming HA, the FortiGates will first form HA and then synchronize the private-data-encryption key after. Once both devices have the same private-data-encryption key, configurations are synchronized from the primary to the secondary.

If private-data-encryption is enabled after the HA cluster is formed, then the primary unit will generate the random master-encryption-password. The password is synchronized from the primary to the secondary immediately after it is created.

Configure TCP NPU session delay globally

The TCP NPU session delay can be applied globally, eliminating the need to set this command for each firewall policy.

```
config system global
    set delay-tcp-npu-session {enable | disable}
end
```

This global setting is disabled by default. When it is disabled, if the host interface is busy, it is possible that the third TCP session establishment ACK received from the client is transmitted to the server after the data packets. When it is enabled, the packet order of the three-way handshake is guaranteed.

A sniffer trace will display the following when the setting is disabled:

```
# diagnose sniffer packet port1 'tcp' 6 0 a
interfaces=[port1]
filters=[tcp]
2024-04-17 20:42:48.920621 port1 -- 172.16.200.55.45028 -> 10.1.100.11.80: syn 1844864123
0x0000   8439 8ff2 9c30 000c 2960 1955 0800 4500        .9...0..)`.U..E.
0x0010   003c 868a 4000 4006 d1dd ac10 c837 0a01        .<..@.@......7..
0x0020   640b afe4 0050 6df6 647b 0000 0000 a002        d....Pm.d{......
0x0030   70bc 3f42 0000 0204 05a3 0402 080a 5026        p.?B..........P&
0x0040   e2f1 0000 0000 0103 0307                       ..........

2024-04-17 20:42:48.921391 port1 -- 10.1.100.11.80 -> 172.16.200.55.45028: syn 2427492278 ack
1844864124
0x0000   000c 2960 1955 8439 8ff2 9c30 0800 4500        ..)`.U.9...0..E.
0x0010   003c 0000 4000 3e06 5a68 0a01 640b ac10        .<..@.>.Zh..d...
0x0020   c837 0050 afe4 90b0 97b6 6df6 647c a012        .7.P......m.d|..
0x0030   7120 d861 0000 0204 0576 0402 080a 5029        q..a.....v....P)
0x0040   ee07 5026 e2f1 0103 0307                       ..P&......

2024-04-17 20:42:48.921586 port1 -- 172.16.200.55.45028 -> 10.1.100.11.80: ack 2427492279
0x0000   8439 8ff2 9c30 000c 2960 1955 0800 4500        .9...0..)`.U..E.
0x0010   0034 868b 4000 4006 d1e4 ac10 c837 0a01        .4..@.@......7..
0x0020   640b afe4 0050 6df6 647c 90b0 97b7 8010        d....Pm.d|......
0x0030   00e2 772e 0000 0101 080a 5026 e2f1 5029        ..w.......P&..P)
0x0040   ee07                                           ..

2024-04-17 20:42:48.922499 port1 -- 10.1.100.11.80 -> 172.16.200.55.45028: ack 1844864277
0x0000   000c 2960 1955 8439 8ff2 9c30 0800 4500        ..)`.U.9...0..E.
0x0010   0034 79b0 4000 3e06 e0bf 0a01 640b ac10        .4y.@.>.....d...
0x0020   c837 0050 afe4 90b0 97b7 6df6 6515 8010        .7.P......m.e...
0x0030   00eb 768c 0000 0101 080a 5029 ee07 5026        ..v.......P)..P&
0x0040   e2f1
```

A sniffer trace will display the following when the setting is enabled:

```
# diagnose sniffer packet port1 'tcp' 6 0 a
interfaces=[port1]
filters=[tcp]
```

```
2024-04-17 20:37:11.440240 port1 -- 172.16.200.55.43672 -> 10.1.100.11.80: syn 780932462
0x0000   8439 8ff2 9c30 000c 2960 1955 0800 4500        .9...0..)`.U..E.
0x0010   003c 8c31 4000 4006 cc36 ac10 c837 0a01        .<.1@.@..6...7..
0x0020   640b aa98 0050 2e8c 156e 0000 0000 a002        d....P...n......
0x0030   70bc 1c99 0000 0204 05a3 0402 080a 5025        p.............P%
0x0040   995f 0000 0000 0103 0307                       ._........

2024-04-17 20:37:11.440925 port1 -- 10.1.100.11.80 -> 172.16.200.55.43672: syn 3325091396 ack
780932463
0x0000   000c 2960 1955 8439 8ff2 9c30 0800 4500        ..)`.U.9...0..E.
0x0010   003c 0000 4000 3e06 5a68 0a01 640b ac10        .<..@.>.Zh..d...
0x0020   c837 0050 aa98 c630 de44 2e8c 156f a012        .7.P...0.D...o..
0x0030   7120 833c 0000 0204 0576 0402 080a 5028        q..<.....v....P(
0x0040   a476 5025 995f 0103 0307                       .vP%._....

2024-04-17 20:37:11.441126 port1 -- 172.16.200.55.43672 -> 10.1.100.11.80: ack 3325091397
0x0000   8439 8ff2 9c30 000c 2960 1955 0800 4500        .9...0..)`.U..E.
0x0010   0034 8c32 4000 4006 cc3d ac10 c837 0a01        .4.2@.@..=...7..
0x0020   640b aa98 0050 2e8c 156f c630 de45 8010        d....P...o.0.E..
0x0030   00e2 2209 0000 0101 080a 5025 995f 5028        .".......P%._P(
0x0040   a476                                           .v

2024-04-17 20:37:11.441518 port1 -- 172.16.200.55.43672 -> 10.1.100.11.80: psh 780932463 ack
3325091397
0x0000   8439 8ff2 9c30 000c 2960 1955 0800 4500        .9...0..)`.U..E.
0x0010   00cd 8c33 4000 4006 cba3 ac10 c837 0a01        ...3@.@......7..
0x0020   640b aa98 0050 2e8c 156f c630 de45 8018        d....P...o.0.E..
0x0030   00e2 feba 0000 0101 080a 5025 995f 5028        ..........P%._P(
0x0040   a476 4745 5420 2f76 6972 7573 2f69 6d61        .vGET./virus/ima
0x0050   6765 2e6f 7574 2048 5454 502f 312e 310d        ge.out.HTTP/1.1.
0x0060   0a55 7365 722d 4167 656e 743a 2057 6765        .User-Agent:.Wge
0x0070   742f 312e 3137 2e31 2028 6c69 6e75 782d        t/1.17.1.(linux-
0x0080   676e 7529 0d0a 4163 6365 7074 3a20 2a2f        gnu)..Accept:.*/
0x0090   2a0d 0a41 6363 6570 742d 456e 636f 6469        *..Accept-Encodi
0x00a0   6e67 3a20 6964 656e 7469 7479 0d0a 486f        ng:.identity..Ho
0x00b0   7374 3a20 3130 2e31 2e31 3030 2e31 310d        st:.10.1.100.11.
0x00c0   0a43 6f6e 6e65 6374 696f 6e3a 204b 6565        .Connection:.Kee
0x00d0   702d 416c 6976 650d 0a0d 0a                    p-Alive....

2024-04-17 20:37:11.441883 port1 -- 10.1.100.11.80 -> 172.16.200.55.43672: ack 780932616
0x0000   000c 2960 1955 8439 8ff2 9c30 0800 4500        ..)`.U.9...0..E.
0x0010   0034 7a33 4000 3e06 e03c 0a01 640b ac10        .4z3@.>..<..d...
0x0020   c837 0050 aa98 c630 de45 2e8c 1608 8010        .7.P...0.E......
0x0030   00eb 2167 0000 0101 080a 5028 a476 5025        ..!g......P(.vP%
0x0040   995f
```

Virtual Domains

Virtual Domains (VDOMs) are used to divide a FortiGate into two or more virtual units that function independently. VDOMs can provide separate security policies and, in NAT mode, completely separate configurations for routing and VPN services for each connected network.

Multiple VDOMs can be created and managed as independent units in multi-VDOM mode.

The number of FortiGate units is dependent on the FortiGate series and many FortiGate models support purchasing a license key to increase the maximum number. For example, 200 to 400 series FortiGates support 25 VDOMs while 500 to 900 series FortiGates support 50 VDOMs. Some exceptions may apply.

The following topics provide an overview of VDOM concepts, topologies, best practices, and the general configurations involved when working with multi-VDOM mode:

- VDOM overview on page 3835
- General configurations on page 3840
- Configuring global profiles on page 3848
- Backing up and restoring configurations in multi-VDOM mode on page 3849

The following topics provide examples of configuring VDOMs:

- Inter-VDOM routing configuration example: Internet access on page 3853
- Inter-VDOM routing configuration example: Partial-mesh VDOMs on page 3863

VDOM overview

The following sections provide conceptual information on VDOMs:

- Multi-VDOM mode on page 3835
- Global settings on page 3836
- Global and per-VDOM resources on page 3836
- Management VDOM on page 3836
- VDOM types on page 3836
- Administrator roles and views on page 3837
- Inter-VDOM routing on page 3837

The following sections provide information on methods of VDOM configuration:

- Topologies on page 3838
- Best practices on page 3839

Multi-VDOM mode

In multi-VDOM mode, the FortiGate can have multiple VDOMs that function as independent units. When multi-VDOM mode is first enabled, all VDOM configurations will move to the root VDOM by default. The root VDOM cannot be deleted, and remains in the configuration even if it is not processing any traffic. New VDOMs can be created, up to the VDOM limit allowable on your device.

Global settings

Global settings are configured outside of a VDOM. They effect the entire FortiGate, and include settings such as interfaces, firmware, DNS, some logging and sandboxing options, and so on. Global settings should only be changed by top level administrators.

Global and per-VDOM resources

Global and per-VDOM resources can be configured when the FortiGate is in multi-VDOM mode. Global resources apply to resources that are shared by the whole FortiGate, while per-VDOM resources are specific to each VDOM.

By default, all per-VDOM resource settings are set to have no limits. This means that any single VDOM can use all of the FortiGate device's resources. This could deprive other VDOMs of the resources that they require, to the point that could be unable to function. We recommend setting maximum values on the resources that are vital to you.

Management VDOM

The management VDOM refers to the specific role that must be designated to one of the VDOMs. By default, the root VDOM is the management VDOM, and management-related services such as FortiGuard updates and other local out (self-originating) traffic such as logs to remote servers originate from the management VDOM. The management VDOM cannot be deleted. See Management VDOM on page 3841 for configuration details.

VDOM types

When a FortiGate is in multi-VDOM mode, a VDOM can be configured as an *Admin*, *Traffic*, or LAN extension type VDOM.

When the VDOM type is set to *Admin*, the VDOM is used to administer and manage the FortiGate. Usually, the *Admin* VDOM resides in a management network which is only accessible by administrators. Global and VDOM administrators can log in to the FortiGate using SSH, HTTPS, and so on but traffic cannot pass through this *Admin* VDOM. A FortiGate does not need to have an *Admin* VDOM and, at most, there can only be one *Admin* VDOM per FortiGate.

When VDOM type is set to *Traffic*, the VDOM can pass traffic like a regular firewall. Most VDOMs will be *Traffic* type VDOMs. Network interfaces on a *Traffic* VDOM can also enable SSH, HTTPS, and so on for administrative and management purposes.

In general, an *Admin* VDOM has a subset of a *Traffic* VDOM's capabilities. See Configure an administrative VDOM type on page 3844 for configuration details.

A LAN extension mode VDOM allows a remote FortiGate to provide remote connectivity to a local FortiGate over a backhaul connection. It can only be configured in the CLI. See FortiGate LAN extension on page 995 for details.

FortiGate-VM supports having at least two VDOMs; one that supports an administrative VDOM and another that supports a traffic VDOM.

Administrator roles and views

When a FortiGate has been configured in multi-VDOM mode, the device can be managed by global administrators and per-VDOM administrators. Each type of administrator will have a different view of the GUI in multi-VDOM mode which corresponds to their role.

Global administrators

Global administrators have complete visibility and access because the scope of their role is to manage the entire physical FortiGate device. An example of a global administrator is an administrator working for a managed security services provider (MSSP) providing the FortiGate as a multi-tenant environment to its clients.

When global administrators log into the GUI, from the *VDOM: Global* view they will see all pages for global settings shared between VDOMs, and VDOM-specific settings.

To create a global administrator that has access to all VDOMs and access to global settings, it must be created at the global level and must use the *super_admin* administrator profile.

See Administrator profiles on page 3736 and Local authentication on page 3701 for configuration details.

VDOM administrators

VDOM administrators will be unable to view global settings or VDOMs not assigned to them because the scope of their role is restricted to managing specific VDOMs only. An example of a VDOM administrator is the administrator working for a company which is a client, or tenant, of an MSSP's multi-tenant FortiGate.

When VDOM administrators log into the GUI, from the *VDOM:<VDOM>* view they will see pages for settings specific to the VDOM they have been configured to administer such as interfaces, routes, firewall policies, and security profiles.

See Create per-VDOM administrators on page 3843 for configuration details.

Inter-VDOM routing

VDOM links are virtual interfaces that allow VDOMs to communicate internally without using additional physical interfaces. A VDOM link contains a pair of interfaces, each one connected to a VDOM to form each end of the inter-VDOM connection. Inter-VDOM routing can be configured in order to communicate between one VDOM to another.

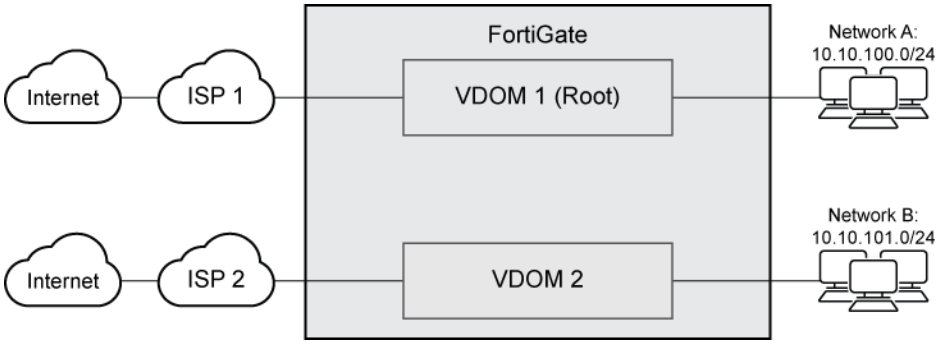
When VDOMs are configured on your FortiGate unit, configuring inter-VDOM routing and VDOM links is similar to creating a VLAN interface. VDOM links can be managed in either the CLI or in the network interface list in the GUI.

See Inter-VDOM routing configuration example: Internet access on page 3853 for more information.

Topologies

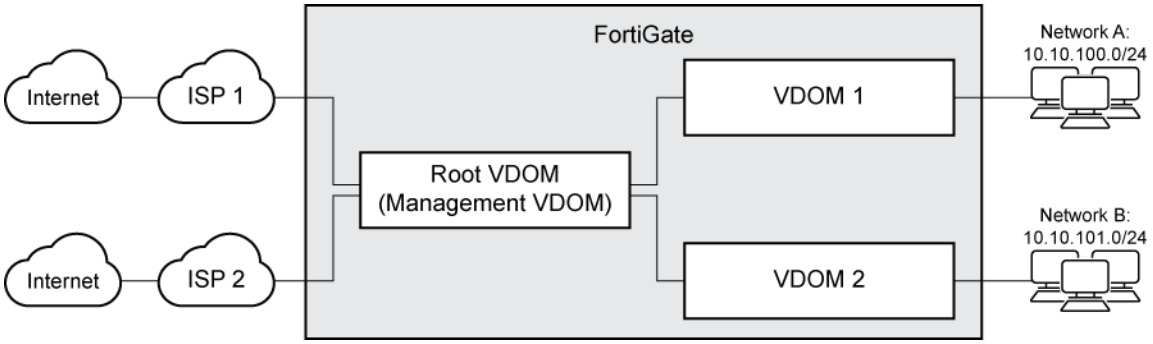
These are the main configuration types in multi-VDOM mode:

Independent VDOMs



Multiple, completely separate VDOMs are created. Any VDOM can be the management VDOM, as long as it has Internet access to connect to FortiGuard services and other management resources. There are no inter-VDOM links, and each VDOM is independently managed.

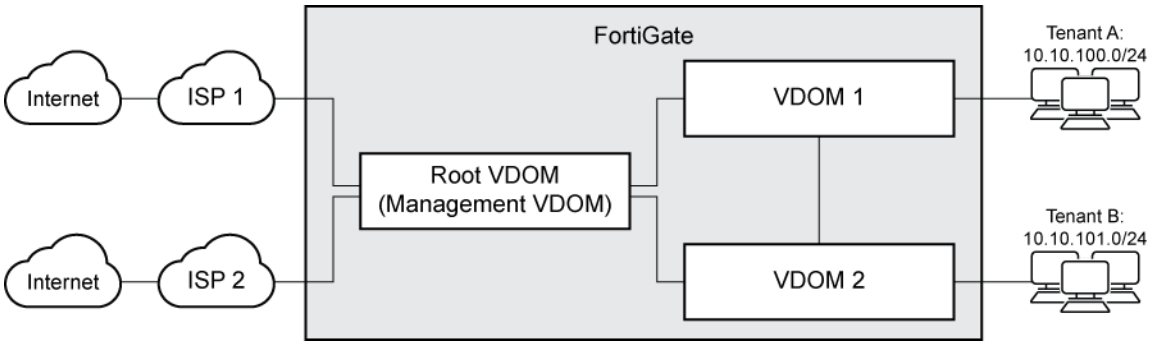
Internet access VDOM



In the Internet access VDOM configuration, Internet access is provided primarily by a single VDOM; for example, the management VDOM (depicted as root VDOM in the preceding diagram). Each tenant connects to the management VDOM via an inter-VDOM link. The management VDOM has complete control over Internet access, including the types of traffic that are allowed in both directions. This can improve security, as there is only one point of ingress and egress.

There is no communication between the other VDOMs.

Meshed VDOMs

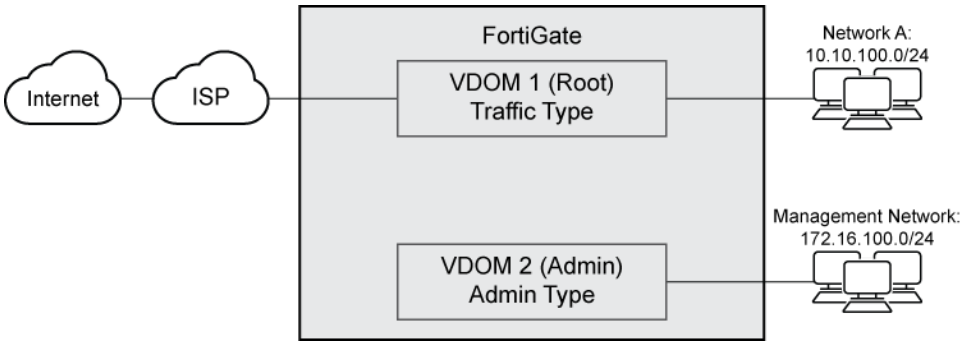


VDOMs can communicate with inter-VDOM links. In full-mesh configurations, all the VDOMs are interconnected. In partial-mesh configurations, only some of the VDOMs are interconnected.

In this configuration, inter-vdom links between tenants are created by the global administrator, but each tenant controls the firewall policies to allow access to other tenants.

See Inter-VDOM routing on page 3846 and Inter-VDOM routing configuration example: Internet access on page 3853 for configuration details.

Administrative VDOM on a management network



The administrative VDOM type can be used to limit administrative access to the FortiGate using SSH, HTTPS and so on to administrators working from a management network. Administrators may be limited to management settings or may have global privileges to access other VDOMs. The user or tenant network (depicted as Network A in the diagram) uses a traffic type VDOM, which allows traffic to pass through it like a regular firewall and allows configuration of firewall-related settings. This configuration can improve security if the management network is a closed network and administrative access is not enabled on any interfaces on the traffic VDOM.

Best practices

VDOMs can provide separate firewall policies and, in NAT mode, completely separate configurations for routing and VPN services for each connected network or organization. This section provides a list of best practices for configuring VDOMs.

Per-VDOM resource setting

All per-VDOM resource settings are set to no limit by default. To ensure proper functionality of all VDOMs, it is recommended that you set some maximum values for the most vital resources. See Global and per-VDOM resources on page 3842 for configuration details.

Virtual domains in NAT mode

Once the virtual domains have been enabled and one or more VDOMs have been created, they must be configured. The following steps provide a general overview of the configuration process.

To configure VDOMs:

1. Change the management virtual domain.
2. Configure FortiGate interfaces for your VDOMs in NAT mode.
3. Configure VDOM routing.
4. Configure security policies for VDOMs in NAT mode.
5. Configure UTM profiles for VDOMs in NAT mode.
6. Test the configuration.

While you may not require all of the steps for your network topology, it is recommended that you perform them in the order given.

See General configurations on page 3840 for configuration details.

Virtual clustering

Virtual clustering is an extension of FGCP HA that provides failover protection between two instances of one or more VDOMs operating on two FortiGates that are in a virtual cluster. A standard virtual cluster consists of FortiGates that are operating in active-passive HA mode with multiple VDOMs enabled. See HA virtual cluster setup on page 4087 for more details.

Typically, virtual clustering is configured with override enabled and uses device priorities to distribute traffic between the primary and secondary FortiGates.

If you decide to disable override for clustering, as a result of persistent renegotiating, you should disable it for both cluster units.

General configurations

VDOMs can be configured in the GUI and the CLI. To ensure that no VDOMs are accidentally configured in the CLI, prompts can be enabled. These prompts will display to ask for confirmation that the VDOM is meant to be configured in the CLI.

To configure confirmation prompts:

```
config system global
    set edit-vdom-prompt enable
end
```

The following topics provide information on general VDOM configurations:

- Enable multi-VDOM mode on page 3841
- Management VDOM on page 3841
- Global and per-VDOM resources on page 3842
- Create per-VDOM administrators on page 3843
- Configure an administrative VDOM type on page 3844
- Assign interfaces to a VDOM on page 3845
- Inter-VDOM routing on page 3846
- Allow FortiGuard services and updates to initiate from a traffic VDOM on page 3847

Enable multi-VDOM mode

Enable multi-VDOM mode and create the VDOMs in the GUI and CLI.

On FortiGate 90 series models and lower, VDOMs can only be enabled using the CLI.

To enable VDOMs in the GUI:

1. Go to *System* > *Settings* > *General*.
2. In the *System operation settings* sections, enable *Virtual domains*.
3. Click *OK*.

To enable VDOMs in the CLI:

```
config system global
    set vdom-mode multi-vdom
end
```

You will be logged out of the device when the VDOM mode is enabled.

Management VDOM

By default, the management VDOM is *root*. The management VDOM can be manually assigned from the GUI or the CLI.

To assign the management VDOM in the GUI:

1. In the *Global* VDOM, go to *System* > *VDOM*.
2. Select the VDOM you want to assign as the management VDOM.

3. Click *Switch Management*.
4. Click *OK*.

To assign the management VDOM in the CLI:

```
config global
    config system global
        set management-vdom <vdom>
    end
end
```

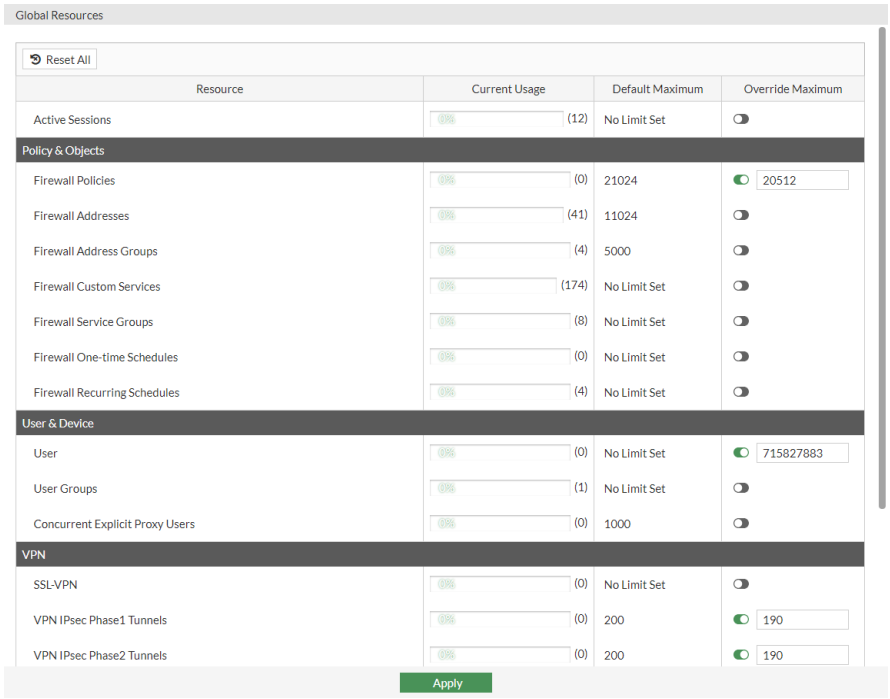
Only one management VDOM can exist at a time. It is strongly recommended that the management VDOM have Internet access otherwise management-related services, such as FortiGuard updates and queries, will not work.

Global and per-VDOM resources

Global resources apply to resources that are shared by the whole FortiGate, while per-VDOM resources are specific to each VDOM.

To configure global resources:

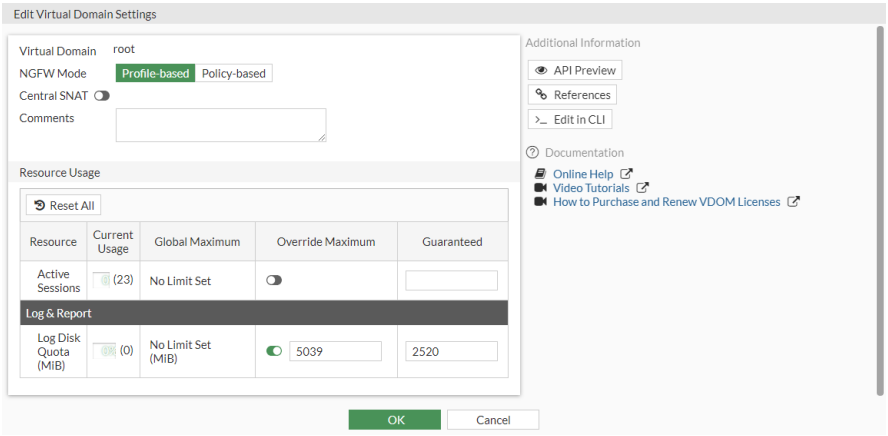
1. In the *Global* VDOM, go to *System > Global Resources*.
2. Enable the resource's override in the *Override Maximum* column, then enter the override value.



3. Click *Apply*.
   To reset all of the override values, click *Reset All*.

To configure per-VDOM resources:

1. In the *Global* VDOM, go to *System* > *VDOM*.
2. Select the VDOM whose resources need to be configured and click *Edit*.
3. Enable the resource's override in the *Override Maximum* column, then enter the override value.
4. Optionally, enter a value in the *Guaranteed* column.



5. Click *OK*.
   To reset all of the override values, click *Reset All*.

Create per-VDOM administrators

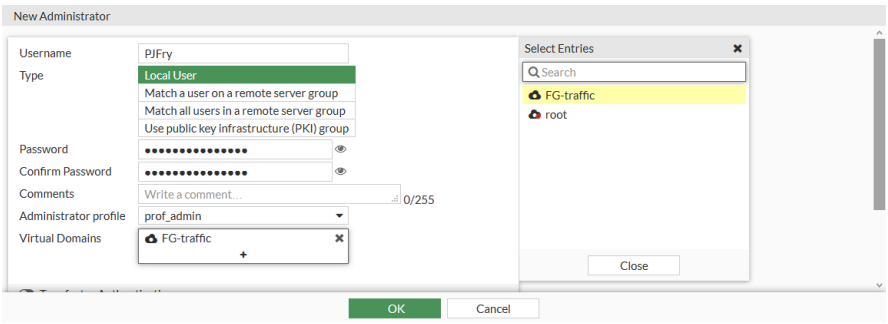
Per-VDOM administrators can be created that can access only the administrative or traffic VDOM. These administrators must use either the *prof_admin* administrator profile, or a custom profile.

A per-VDOM administrator can only access the FortiGate through a network interface that is assigned to the VDOM that they are assigned to. The interface must also be configured to allow management access. They can also connect to the FortiGate using the console port.

To assign an administrator to multiple VDOMs, they must be created at the global level. When creating an administrator at the VDOM level, the *super_admin* administrator profile cannot be used.

To create a per-VDOM administrator in the GUI:

1. On the FortiGate, connect to the *Global* VDOM.
2. Go to *System* > *Administrators* and click *Create New* > *Administrator*.
3. Fill in the required information, setting the *Type* as *Local User*.
4. In the *Virtual Domains* field, add the VDOM that the administrator will be assigned to, and if necessary, remove the other VDOM from the list.



5. Click *OK*.

To create a per-VDOM administrator using the CLI:

```
config global
    config system admin
        edit <name>
            set vdom <VDOM_name>
            set password <password>
            set accprofile <admin_profile>
            ...
        next
    end
end
```

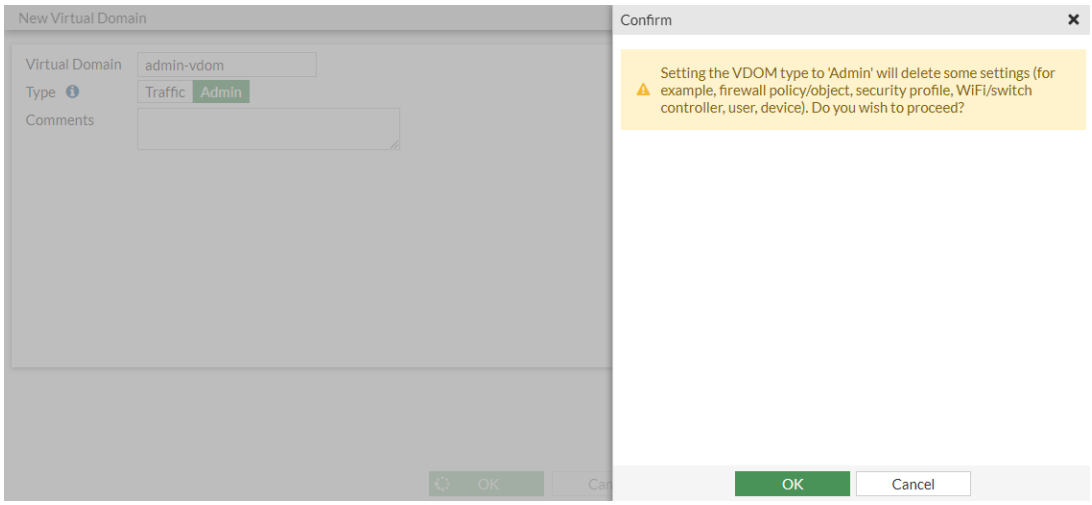
Configure an administrative VDOM type

Individual VDOMs can be configured as an administrative type in multi-VDOM mode.

> Only one administrative VDOM can exist at a time and cannot be set on a FortiWifi. A VDOM cannot be an administrative type and in transparent mode at the same time.

To configure an administrative VDOM in the GUI:

1. Go to *System* > *VDOM*.
2. Click *Create New*.
3. Enter a *Virtual Domain* name and set the *Type* to *Admin*.
4. Click *OK*.



5. Click *OK* in the confirmation pane. The administrative VDOM is created.

To configure the VDOM type in the CLI:

```
config system settings
    set vdom-type {traffic | admin}
end
```

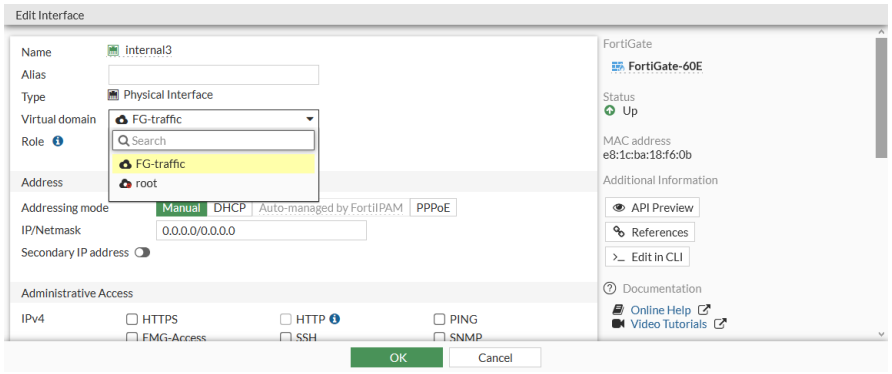
Assign interfaces to a VDOM

An interface can only be assigned to one of the VDOMs. An interface cannot be moved if it is referenced in an existing configuration.

> In the GUI, the interface list *Ref.* column shows if the interface is referenced in an existing configuration, and allows you to quickly access and edit those references.

To assign an interface to a VDOM in the GUI:

1. In the *Global* VDOM, go to *Network* > *Interfaces*.
2. Select the interface that will be assigned to a VDOM and click *Edit*.
3. Select the VDOM that the interface will be assigned to from the *Virtual Domain* list.



4. Click *OK*.

To assign an interface to a VDOM using the CLI:

```
config global
    config system interface
        edit <interface>
            set vdom <VDOM_name>
        next
    end
end
```

Inter-VDOM routing

VDOM links allow VDOMs to communicate internally without using additional physical interfaces.

A VDOM link cannot share the same name as a VDOM.

VDOM link does not support traffic offload. If you want to use traffic offload, use NPU-VDOM-LINK. See Configuring inter-VDOM link acceleration with NP6 processors in the Hardware Acceleration guide for details.

To configure a VDOM link in the GUI:

1. In the *Global* VDOM, go to *Network* > *Interfaces*.
2. Click *Create New* > *VDOM Link*.
3. Configure the fields, including the *Name*, *Virtual Domain*, IP information, *Administrative Access*, and so on, then click *OK*.

By default, VDOM links are created as point-to-point (ppp) links. If required, the link type can be changed in the CLI.

For example, when running OSPF in IPv6, a link-local address is required in order to communicate with OSPF neighbors. For a VDOM link to obtain a link-local address, its type must be set to `ethernet`.

To configure a VDOM link in the CLI:

```
config global
    config system vdom-link
        edit "<vdom-link-name>"
            set type {ppp | ethernet}
        next
    end
    config system interface
        edit "<vdom-link-name0>"
```

```
            set vdom "<VDOM Name>"
            set type vdom-link
        next
        edit "<vdom-link-name1>"
            set vdom "<VDOM Name>"
            set type vdom-link
        next
    end
end
```

To delete a VDOM link in the GUI:

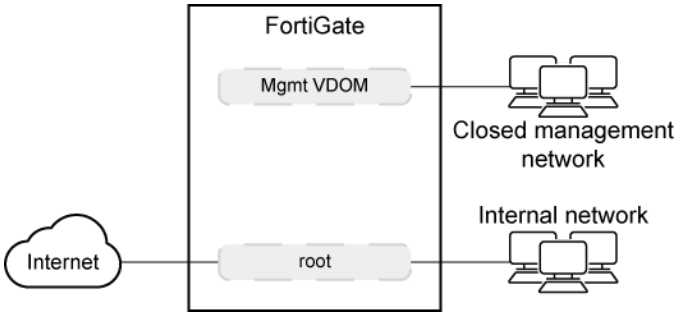
1. In the *Global* VDOM, go to *Network* > *Interfaces*.
2. Select a *VDOM Link* and click *Delete*.

To delete a VDOM link in the CLI:

```
config global
    config system vdom-link
        delete <VDOM-LINK-Name>
    end
end
```

Allow FortiGuard services and updates to initiate from a traffic VDOM

In multi-VDOM mode, users can choose from which VDOM FortiGuard services and updates are initiated from, instead of being locked to the management VDOM. This allows deployment scenarios where the management VDOM resides in a closed management network.



When the management VDOM resides in a closed network, it does not have internet access. FortiGuard services (FortiGuard updates, web filters, DNS proxy, DDNS, and so on) must be configured in a VDOM with Internet access in order to work. Therefore, in the example above, change the FortiGuard settings to initiate from the root VDOM.

To configure FortiGuard services on a traffic VDOM:

1. Set up a traffic VDOM for FortiGuard services:

```
config global
    config system fortiguard
        set vdom "root"
    end
end
```

2. Ensure the traffic VDOM has the correct gateway to reach the internet:

```
config vdom
    edit root
        config router static
            edit 1
                set gateway 172.16.200.254
                set device "wan1"
            next
        end
    next
end
```

3. Configure the DNS servers to ensure the FortiGuard services can resolve the server name through the traffic VDOM:

```
config vdom
    edit root
        config system vdom-dns
            set vdom-dns enable
            set primary 208.91.112.53
            set secondary 208.91.112.52
        end
    next
end
```

Configuring global profiles

Global profiles can be configured globally across multiple VDOMs, even when multi-VDOM mode is disabled. When used in multi-VDOM mode, some or all profiles may be commonly-shared across VDOMs. Global profiles are available as read-only for VDOM-level administrators and can only be edited or deleted from within the global settings. The name for any global profile must begin with *g-* for identification. Each security feature has at least one default global profile, available for all VDOMs to use.

Some security profile features, such as URL filters under web-filter, are not available for use in a global profile.

External feeds also support global profile setting by specifying the *g-* prefix. When the FortiGate is in multi-VDOM mode, a global external feed will connect to the feed through one of the management VDOM's interfaces. A per-VDOM external feed will connect to the feed through the specific VDOM's interfaces.

In a non-VDOM setup, a profile with a *g-* prefix is accepted and, when created, is applied in the root VDOM. If the FortiGate is changed to multi-VDOM mode, the *g-* profile will automatically be moved to the global VDOM.

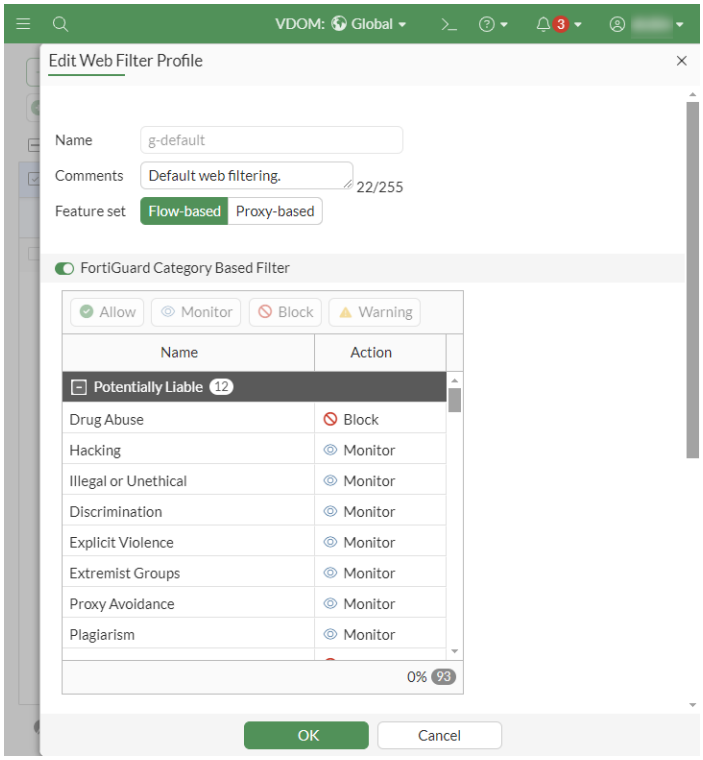
The following examples demonstrate configuring and editing Web Filter global security profiles. Similarly, you can view, edit, and configure other global security profiles for Antivirus, Application Control, Intrusion Prevention, and File Filter.

To configure a global security profile:

1. Go to *Security Profile* > *Web Filter*.
2. Click *Create new*.
3. Enter any suitable name that begins with *g-* and configure the web-filter settings required.
4. Click *OK*. This global web-filter is now available to be used in different VDOMs as required.

To view and edit a global security profile:

1. In the Global VDOM, go to *Security Profiles* > *Web Filter*. The names of the global web-filter security profile begins with *g-* for identification.
2. Select the default global web-filter profile named *g-default* and click *Edit*.
3. Under *FortiGuard Category Based Filter* select *Drug Abuse* and set the *Action* to *Block*.



4. Click *OK*.

Backing up and restoring configurations in multi-VDOM mode

When a FortiGate is in multi-VDOM mode, the configuration can be backed up or restored using the GUI or the CLI. Back up and restoration permissions depend on the VDOM administrator when in multi-VDOM mode:

- A global *super_admin* can back up and restore the global configuration or the configuration of a specific VDOM.
- A VDOM administrator of one VDOM can only back up and restore the configuration of the current VDOM.
- A VDOM administrator of multiple VDOMs can back up and restore the configuration of multiple VDOMs.

To back up the configuration using the GUI:

1. Click on the user name in the upper right-hand corner of the screen and select *Configuration* > *Backup*.
2. Select *VDOM* for the *Scope*. The *VDOM* dropdown menu is displayed.
3. Select the VDOM you want to back up.
4. Direct the backup to your *Local PC* or to a *USB Disk*.
5. Enable *Encryption*.

   This is recommended to secure your backup configurations and prevent unauthorized parties from reloading your configuration.
6. Enter a password, and enter it again to confirm it. This password will be required to restore the configuration.
7. Click *OK*.
8. When prompted, select a location on the PC or USB disk to save the configuration file. The configuration file will have a .conf extension.

To restore the FortiGate configuration using the GUI:

1. Click on the user name in the upper right-hand corner of the screen and select *Configuration* > *Restore*.
2. Select *VDOM* for the *Scope*. The *VDOM* dropdown menu is displayed.
3. Select the VDOM that you want to restore the configuration for.
4. Identify the source of the configuration file to be restored: your *Local PC* or a *USB Disk*.
   The *USB Disk* option will not be available if no USB drive is inserted in the USB port. You can restore from the FortiManager using the CLI.
5. Click *Upload*, locate the configuration file, and click *Open*.

   Confirm that the configuration file you are uploading is for the same VDOM selected from the dropdown menu.
6. Enter the password if required.
7. Click *OK*.

Backing up configurations in the CLI

Configuration backups can be performed in the CLI using the `execute backup` commands. If you are backing up a VDOM configuration instead of the global configuration, first enter the commands:

```
config vdom
    edit <vdom_name>
```

Configurations can be backed up in FortiOS and YAML format.

Configuration files can be backed up to various locations depending on the command:

- `flash`: Backup the configuration file to the flash drive.
- `ftp`: Backup the configuration file to an FTP server.
- `sftp`: Backup the configuration file to a SFTP server.
- `tftp`: Backup the configuration file to a TFTP server.
- `usb`: Backup the configuration file to an external USB drive.

Command	Description
`# execute backup config`	Back up the configuration in FortiOS format. Backup your configuration file to: • `flash` • `ftp` • `sftp` • `tftp` • `usb`
`# execute backup full-config`	Backup the configuration, including backups of default configuration settings. Backup your configuration file to: • `ftp` • `sftp` • `tftp` • `usb`
`# execute backup yaml-config`	Backup the configuration in YAML format. Backup your configuration file to: • `ftp` • `tftp`

To back up the configuration in FortiOS format using the CLI:

For FTP, note that port number and username are optional depending on the FTP site:

```
config vdom
    edit <vdom_name>
        execute backup config ftp <backup_filename> <ftp_server>[<:ftp_port>] [<user_name>]
[<password>] [<backup_password>]
```

or for TFTP:

```
config vdom
    edit <vdom_name>
        execute backup config tftp <backup_filename> <tftp_servers> [<backup_password>]
```

or for SFTP:

```
config vdom
    edit <vdom_name>
```

```
        execute backup config sftp <backup_filename> <sftp_server>[<:sftp_port>] <user> <password>
[<backup_password>]
```

or for an external USB:

```
config vdom
    edit <vdom_name>
        execute backup config usb <backup_filename> [<backup_password>]
```

To back up the configuration in YAML format using the CLI:

For FTP:

```
config vdom
    edit <vdom_name>
        execute backup yaml-config ftp <file_path> <ftp_server>[<:port>] [<user_name>] [<FTP
password>]
```

or for TFTP:

```
config vdom
    edit <vdom_name>
        execute backup yaml-config tftp <file_path> <tftp_server>
```

Restoring configurations in the CLI

Restoring configurations can be performed in the CLI using the `execute restore` command. If you are restoring a VDOM configuration instead of the global configuration, first enter the commands:

```
config vdom
    edit <vdom_name>
```

When restoring a VDOM configuration, ensure that the configuration file is for the correct VDOM specified.

Command	Description
`# execute restore config`	Restore a configuration that is in FortiOS or YAML format. Configurations can be loaded from: • `dhcp`: Load the configuration though DHCP. • `flash`: Load the configuration file from flash to firewall. • `ftp`: Load the configuration file from an FTP server. • `tftp`: Load the configuration from a TFTP server. • `usb`: Load the configuration file from an external USB disk to firewall.

To restore the FortiGate configuration in FortiOS or YAML format using the CLI:

For FTP, note that port number and username are optional depending on the FTP site:

```
config vdom
    edit <vdom_name>
        execute restore config ftp <file_path> <ftp_server>[<:port>] [<user_name>] [<FTP
password>] [<password>]
```

or for TFTP:

```
config vdom
    edit <vdom_name>
        execute restore config tftp <file_name> <tftp_server> [<password>]
```

or for DHCP:

```
config vdom
    edit <vdom_name>
        execute restore config dhcp <port> [<VLAN_ID>]
```

or for flash:

```
config vdom
    edit <vdom_name>
        execute restore config flash <revision_ID>
```

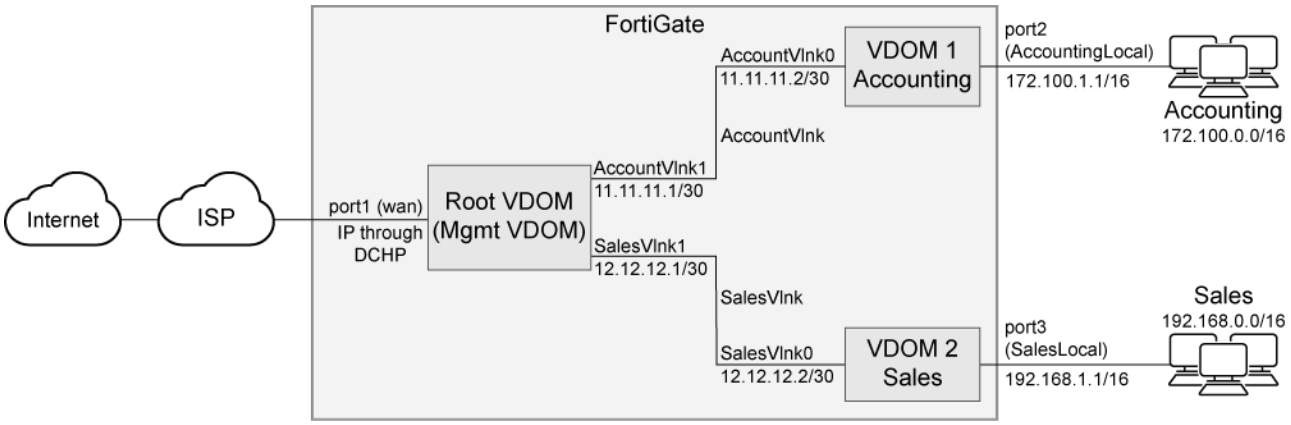
or for an external USB:

```
config vdom
    edit <vdom_name>
        execute restore config usb <file_name> [<password>]
```

Inter-VDOM routing configuration example: Internet access

This example shows how to configure a FortiGate unit to use inter-VDOM routing to route outgoing traffic from individual VDOMs to a root VDOM with Internet access. See Inter-VDOM routing on page 3846 for more information.

Two departments of a company, Accounting and Sales, are connected to one FortiGate. The company uses a single ISP to connect to the Internet. This is an example of the Internet access configuration. See Topologies on page 3838 for details.



This example assumes that the interfaces of the FortiGate have already been configured with the IP addresses depicted in the preceding diagram.

General steps for this example

This example includes the following general steps. We recommend following the steps in the order below:

1. Enable multi-VDOM mode and create the VDOMs on page 3854
2. Assign interfaces to VDOMs on page 3855
3. Configure the VDOM links on page 3855
4. Configure inter-VDOM routing on page 3857
5. Configure the firewall policies on page 3858
6. Test the configuration on page 3859

This example demonstrates how to configure these steps first using the GUI and then, at the end of the section, using the CLI. See Configuration with the CLI on page 3860 for details.

Enable multi-VDOM mode and create the VDOMs

Create the Accounting and Sales VDOMs.

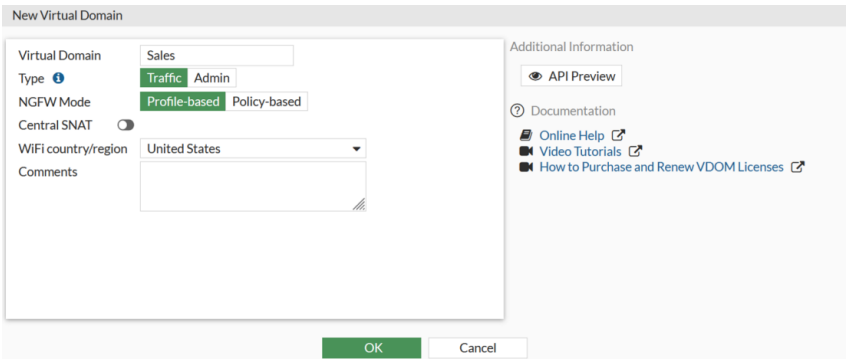
To enable VDOMs in the GUI:

1. Go to *System* > *Settings* > *General*.
2. In the *System operation settings* section, enable *Virtual domains*.
3. Click *OK*.

On FortiGate 90 series models and lower, VDOMs can only be enabled using the CLI.

To create the Sales and Accounting VDOMs in the GUI:

1. In the *Global* VDOM, go to *System* > *VDOM*.
2. Click *Create New*.
3. In the *Virtual Domain* field, enter *Sales*.



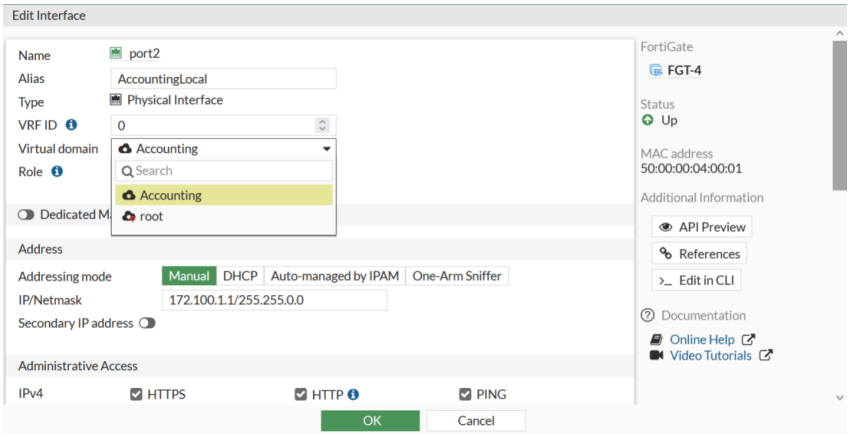
4. If required, set the *NGFW Mode*. If the *NGFW Mode* is *Profile-based*, *Central SNAT* can be enabled.
5. Click *OK* to create the VDOM.
6. Repeat the above steps for *Accounting*.

Assign interfaces to VDOMs

This example uses three interfaces on the FortiGate unit: port2 (AccountingLocal), port3 (SalesLocal), and port1 (WAN). Port2 and port3 interfaces each have a department's network connected. Port1 is for all traffic to and from the Internet and uses DHCP to configure its IP address, which is common with many ISPs.

To assign interfaces to VDOMs in the GUI:

1. In the *Global* VDOM, go to *Network* > *Interfaces*.
2. Select *port2* and click *Edit*.
3. From the *Virtual domain* list, select *Accounting*.



4. Click *OK*.
5. Repeat the preceding steps to assign *port3* to the *Sales* VDOM.
6. Repeat the preceding steps to assign *port1* to the *root* VDOM.

Configure the VDOM links

To complete the connection between each VDOM and the management VDOM, add the two VDOM links. One pair is the Accounting – management link and the other is the Sales – management link. Each side of these links will be assigned IP addresses since they will be handy in configuring inter-VDOM routing in the next step.

To configure the Accounting and management VDOM link in the GUI:

1. In the *Global* VDOM, go to *Network* > *Interfaces*.
2. Select *Create New* > *VDOM Link*.
3. Enter the following information:

Name		AccountVlnk
Interface 0		
	Virtual Domain	Accounting
	IP/Netmask	11.11.11.2/255.255.255.252
	Administrative Access	HTTPS, PING, SSH
	Comment	Accounting side of the VDOM link
Interface 1		
	Virtual Domain	root
	IP/Netmask	11.11.11.1/255.255.255.252
	Administrative Access	HTTPS, PING, SSH
	Comment	Management side of the VDOM link

4. Click *OK*.

To configure the Sales and management VDOM link in the GUI:

1. In the *Global* VDOM, go to *Network* > *Interfaces*.
2. Select *Create New* > *VDOM link*.
3. Enter the following information:

Name		SalesVlnk
Interface 0		
	Virtual Domain	Sales
	IP/Netmask	12.12.12.2/255.255.255.252
	Administrative Access	HTTPS, PING, SSH
	Comment	Accounting side of the VDOM link
Interface 1		
	Virtual Domain	root
	IP/Netmask	12.12.12.1/255.255.255.252

Administrative Access	HTTPS, PING, SSH
Comment	Management side of the VDOM link

4. Click *OK*.

Configure inter-VDOM routing

A default static route can be configured on each VDOM to provide Internet access. In other words, this static route would provide inter-VDOM routing between each department VDOM and the root VDOM.

For this static route, these settings are used:

- Default Gateway: IP address of the management side of the VDOM link
  - Accounting VDOM: 11.11.11.1
  - Sales VDOM: 12.12.12.1
- Interface: Interface on the department VDOM side of the VDOM link
  - Accounting VDOM: AccountVlnk0
  - Sales VDOM: SalesVlnk0
- IP address: 0.0.0.0/0.0.0.0 (default)

To configure the default static route to the Internet in the Accounting VDOM:

1. In the *Accounting* VDOM, go to *Network* > *Static Routes*.
2. Click on *Create New* and select the version you need.
3. Enter the following information:

Destination	Subnet
IP address	0.0.0.0/0.0.0.0
Gateway	11.11.11.1
Interface	AccountVlink0
Administrative Distance	10

4. Click *OK*.

To configure the default static route to the Internet in the Sales VDOM:

1. In the *Sales* VDOM, go to *Network* > *Static Routes*.
2. Click on *Create New* and select the version you need.
3. Enter the following information:

Destination	Subnet
IP address	0.0.0.0/0.0.0.0
Gateway	12.12.12.1

Interface	SalesVlink0
Administrative Distance	10

4. Click *OK*.

Configure the firewall policies

With the VDOMs, physical interfaces, VDOM links, and static routes configured, the firewall must now be configured to allow the proper traffic. Firewalls are configured per-VDOM, and firewall objects and routes must be created for each VDOM separately.

To configure the firewall policies from AccountingLocal to Internet in the GUI:

1. In the *Accounting* VDOM, go to *Policy & Objects > Firewall Policy*.
2. Click *Create New*.
3. Enter the following information:

Name	Account-Local-to-Management
Incoming Interface	port2
Outgoing Interface	AccountVlnk0
Source	All
Destination	All
Schedule	always
Service	ALL
Action	ACCEPT
NAT	enabled

4. Click *OK*.
5. In the *root* VDOM, go to *Policy & Objects > Firewall Policy*.
6. Click *Create New*.
7. Enter the following information:

Name	Account-VDOM-to-Internet
Incoming Interface	AccountVlnk1
Outgoing Interface	port1
Source	All
Destination	All
Schedule	always
Service	ALL
Action	ACCEPT

NAT	enabled

8. Click *OK*.

To configure the firewall policies from SalesLocal to Internet in the GUI:

1. In the *Sales* VDOM, go to *Policy & Objects* > *Firewall Policy*.
2. Click *Create New*.
3. Enter the following information:

Name	Sales-Local-to-Management
Incoming Interface	port3
Outgoing Interface	SalesVlnk0
Source	All
Destination	All
Schedule	always
Service	ALL
Action	ACCEPT
NAT	enabled

4. Click *OK*.
5. In the *root* VDOM, go to *Policy & Objects* > *Firewall Policy*.
6. Click *Create New*.
7. Enter the following information:

Name	Sales-VDOM-to-Internet
Incoming Interface	SalesVlnk1
Outgoing Interface	port1
Source	All
Destination	All
Schedule	always
Service	ALL
Action	ACCEPT
NAT	enabled

8. Click *OK*.

Test the configuration

When the inter-VDOM routing has been configured, test the configuration to confirm proper operation. Testing connectivity ensures that physical networking connections, FortiGate unit interface configurations, and firewall

policies are properly configured.

The easiest way to test connectivity is to use the `ping` and `traceroute` commands on hosts in the Accounting and Sales networks, respectively, to confirm the connectivity of different routes on the network. Test connectivity with hosts connected to port2 (AccountingLocal) in the Accounting VDOM to the internet and hosts connected to port3 (SalesLocal) in the Sales VDOM to the internet.

Configuration with the CLI

The example can also be configured in the CLI.

To configure inter-VDOM routing in the CLI:

1. Enable multi-VDOM mode:

```
config system global
    set vdom-mode multi-vdom
end
```

   You will be logged out of the device when VDOM mode is enabled.
2. Create the Sales and Accounting VDOMs:

```
config vdom
    edit Accounting
    next
    edit Sales
    next
end
```

3. Assign interfaces to the VDOMs:

```
config global
    config system interface
        edit port2
            set vdom Accounting
        next
        edit port3
            set vdom Sales
        next
        edit port1
            set vdom root
        next
    end
end
```

4. Configure the Accounting and management VDOM link:

```
config global
    config system vdom-link
        edit AccountVlnk
        next
```

```
    end
    config system interface
        edit AccountVlnk0
            set vdom Accounting
            set ip 11.11.11.2 255.255.255.252
            set allowaccess https ping ssh
            set description "Accounting side of the VDOM link"
        next
        edit AccountVlnk1
            set vdom root
            set ip 11.11.11.1 255.255.255.252
            set allowaccess https ping ssh
            set description "Management side of the VDOM link"
        next
    end
end
```

5. Configure the Sales and management VDOM link:

```
config global
    config system vdom-link
        edit SalesVlnk
        next
    end
    config system interface
        edit SalesVlnk0
            set vdom Sales
            set ip 12.12.12.2 255.255.255.252
            set allowaccess https ping ssh
            set description "Sales side of the VDOM link"
        next
        edit SalesVlnk1
            set vdom root
            set ip 12.12.12.1 255.255.255.252
            set allowaccess https ping ssh
            set description "Management side of the VDOM link"
        next
    end
end
```

6. Configure the default static route to the Internet in the Accounting VDOM:

```
config vdom
    edit Accounting
    config router static
        edit 1
            set gateway 11.11.11.1
            set device "AccountVlnk0"
        next
    end
end
```

7. Configure the default statis route to the Internet in the Sales VDOM:

```
config vdom
    edit Sales
    config router static
        edit 1
            set gateway 12.12.12.1
            set device "SalesVlnk0"
        next
    end
end
```

8. Configure the firewall policies from AccountingLocal to the Internet:

```
config vdom
    edit Accounting
        config firewall policy
            edit 1
                set name "Accounting-Local-to-Management"
                set srcintf port2
                set dstintf AccountVlnk0
                set srcaddr all
                set dstaddr all
                set action accept
                set schedule always
                set service ALL
                set nat enable
            next
        end
    next
    edit root
        config firewall policy
            edit 2
                set name "Accounting-VDOM-to-Internet"
                set srcintf AccountVlnk1
                set dstintf port1
                set srcaddr all
                set dstaddr all
                set action accept
                set schedule always
                set service ALL
                set nat enable
            next
        end
    next
end
```

9. Configure the firewall policies from SalesLocal to the Internet:

```
config vdom
    edit Sales
        config firewall policy
            edit 3
                set name "Sales-local-to-Management"
```

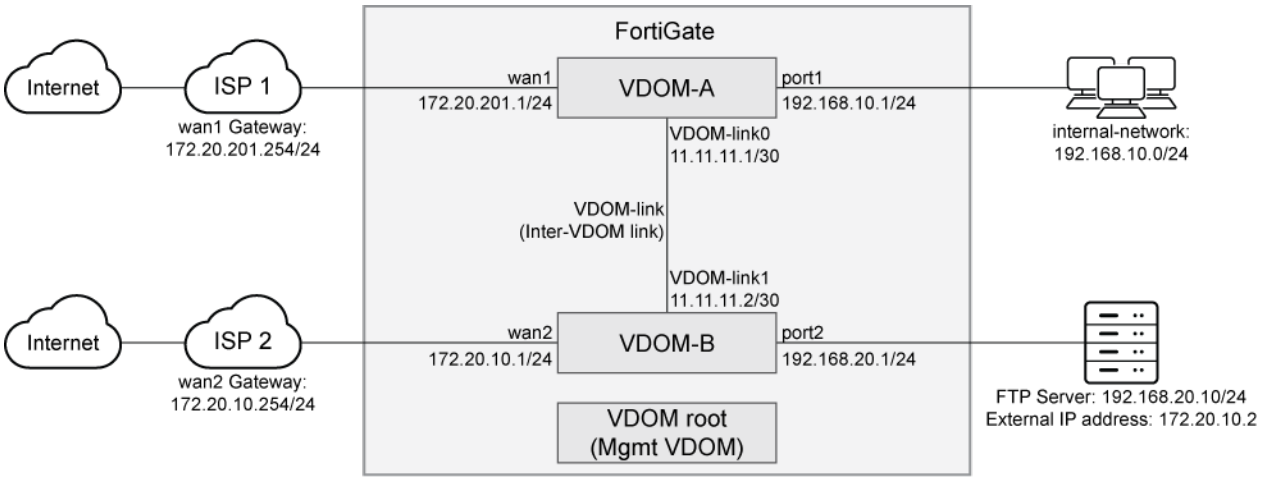
```
                set srcintf port3
                set dstintf SalesVlnk0
                set srcaddr all
                set dstaddr all
                set action accept
                set schedule always
                set service ALL
                set nat enable
            next
        end
    next
    edit root
        config firewall policy
            edit 4
                set name "Sales-VDOM-to-Internet"
                set srcintf SalesVlnk1
                set dstintf port1
                set srcaddr all
                set dstaddr all
                set action accept
                set schedule always
                set service ALL
                set nat enable
            next
        end
    next
end
```

Inter-VDOM routing configuration example: Partial-mesh VDOMs

This example shows how to configure a FortiGate unit to use inter-VDOM routing to route traffic between an internal network and FTP server that are each behind separate VDOMs. See Inter-VDOM routing on page 3846 for more information.

The following example shows how to configure per-VDOM settings, such as operation mode, routing, and firewall policies, in a network that includes the following VDOMs:

- VDOM-A: allows the internal network to access the Internet.
- VDOM-B: allows external connections to an FTP server.
- root: the management VDOM.



You can use VDOMs in either NAT or transparent mode on the same FortiGate. By default, VDOMs operate in NAT mode. In this example, both VDOM-A and VDOM-B use NAT mode. An inter-VDOM link is created and inter-VDOM routes configured to allow users on the internal network to access the FTP server.

This is an example of the partial-mesh VDOMs configuration since only VDOM-A is connected to VDOM-B but neither of those VDOMs are connected to the root VDOM. See Topologies on page 3838 for details.

This example assumes that the interfaces of the FortiGate have already been configured with the IP addresses depicted in the preceding diagram.

General steps for this example

This configuration requires the following general steps:

1. Enable Multi-VDOM mode and create the VDOMs on page 3864
2. Assign interfaces to VDOMs on page 3865
3. Configure VDOM-A on page 3866
4. Configure VDOM-B on page 3867
5. Configure the VDOM link on page 3869
6. Configure inter-VDOM routing on page 3869
7. Configure firewall policies using the VDOM link on page 3871

This example demonstrates how to configure these steps first using the GUI and then, at the end of the section, using the CLI. See Configuration with the CLI on page 3872 for details.

Enable Multi-VDOM mode and create the VDOMs

Multi-VDOM mode can be enabled in the GUI or CLI. Enabling it does not require a reboot, but does log you out of the device. The current configuration is assigned to the root VDOM.

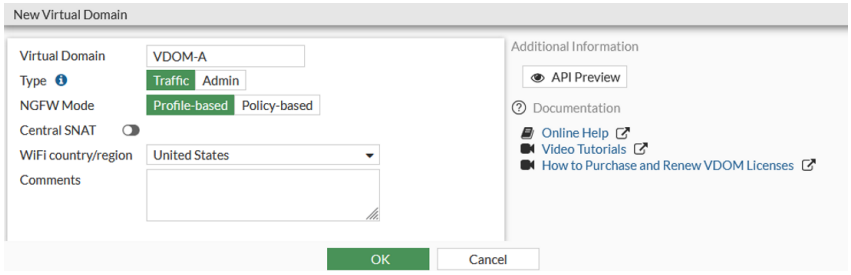
On FortiGate 90 series models and lower, VDOMs can only be enabled using the CLI.

To enable multi-VDOM mode in the GUI:

1. On the FortiGate, go to *System* > *Settings* > *General*.
2. In the *System operation settings* section, enable *Virtual domains*.
3. Click *OK*.

To create the VDOMs in the GUI:

1. In the *Global* VDOM, go to *System* > *VDOM*.
   Click *Create New*.
2. In the *Virtual Domain* field, enter *VDOM-A*.



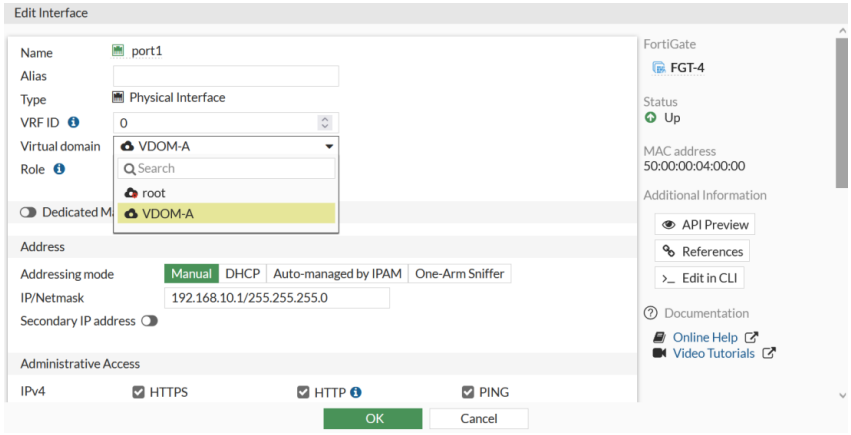
3. If required, set the *NGFW Mode*. If the *NGFW Mode* is *Profile-based*, *Central SNAT* can be enabled.
4. Click *OK* to create the VDOM.
5. Repeat the above steps for *VDOM-B*.

Assign interfaces to VDOMs

This example uses three interfaces on the FortiGate unit: port1 (internal network), port2 (FTP server), wan1 (WAN link for VDOM-A), and wan2 (WAN link for VDOM-B). The port1 and port2 interfaces are connected to the internal network and FTP server, respectively. The wan1 and wan2 interfaces are static assigned with IP addresses and default gateways provided by the ISPs for those WAN links.

To assign interfaces to VDOMs in the GUI:

1. In the *Global* VDOM, go to *Network* > *Interfaces*.
2. Select *port1* and click *Edit*.
3. From the *Virtual domain* list, select *VDOM-A*.



4. Click *OK*.
5. Repeat the preceding steps to assign *port2* to *VDOM-B*.
6. Repeat the preceding steps to assign *wan1* to *VDOM-A*.
7. Repeat the preceding steps to assign *wan2* to *VDOM-B*.

Configure VDOM-A

VDOM-A allows connections from devices on the internal network to the Internet. WAN1 and port1 are assigned to this VDOM.

The per-VDOM configuration for VDOM-A includes the following:

- A firewall address for the internal network
- A static route to the ISP gateway
- A firewall policy allowing the internal network to access the Internet

All procedures in this section require you to connect to VDOM-A, either using a global or per-VDOM administrator account.

To add the firewall addresses in the GUI:

1. Go to *Policy & Objects* > *Addresses* and select *Address*.
2. Click *Create new*.
3. Enter the following information:

Name	internal-network
Type	Subnet
IP/Netmask	192.168.10.0/255.255.255.0
Interface	port1

4. Click *OK*.

To add a default route in the GUI:

1. Go to *Network* > *Static Routes* and create a new route.
2. Enter the following information:

Destination	Subnet
IP address	0.0.0.0/0.0.0.0
Gateway	172.20.201.254
Interface	wan1
Administrative Distance	10

3. Click *OK*.

To add the firewall policy in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Click *Create New*.
3. Enter the following information:

Name	VDOM-A-Internet
Incoming Interface	port1
Outgoing Interface	wan1
Source	internal-network
Destination	all
Schedule	always
Service	ALL
Action	ACCEPT
NAT	enabled

4. Click *OK*.

Configure VDOM-B

VDOM-B allows external connections to reach an internal FTP server. WAN2 and port2 are assigned to this VDOM.

The per-VDOM configuration for VDOM-B includes the following:

- A firewall address for the FTP server
- A virtual IP address for the FTP server
- A static route to the ISP gateway
- A firewall policy allowing external traffic to reach the FTP server

The procedures described above require you to connect to VDOM-B, either using a global or per-VDOM administrator account.

To add the firewall addresses in the GUI:

1. Go to *Policy & Objects* > *Addresses* and select *Address*.
2. Click *Create new*.
3. Enter the following information:

Name	FTP-server
Type	Subnet
IP/Netmask	192.168.20.10/255.255.255.255
Interface	port2

4. Click *OK*.

To add the virtual IP address in the GUI:

1. Go to *Policy & Objects* > *Virtual IPs* and navigate to the *Virtual IP* tab.
2. Click *Create new*.
3. Enter the following information:

Name	FTP-server-VIP
Interface	wan2
External IP address/range	172.20.10.2
Map To	192.168.20.10

4. Click *OK*.

To add a default route in the GUI:

1. Go to *Network* > *Static Routes* and create a new route.
2. Enter the following information:

Destination	Subnet
IP address	0.0.0.0/0.0.0.0
Gateway	172.20.201.254
Interface	wan2
Administrative Distance	10

3. Click *OK*.

To add the firewall policy in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Click *Create New*.
3. Enter the following information:

Name	Access-server
Incoming Interface	wan2
Outgoing Interface	port2
Source	all
Destination	FTP-server-VIP
Schedule	always
Service	FTP
Action	ACCEPT
NAT	enabled

4. Click *OK*.

Configure the VDOM link

The VDOM link allows connections from VDOM-A to VDOM-B. The VDOM link interface configured in this step will be used for inter-VDOM routing.

This step requires you to connect to the global VDOM using a global administrator account.

To add the VDOM link in the GUI:

1. In the *Global* VDOM, go to *Network* > *Interfaces*.
2. Create *New* > *VDOM link*.
3. Enter the following information:

Name	VDOM-link
Interface 0	
Virtual Domain	VDOM-A
IP/Netmask	11.11.11.1/255.255.255.252
Interface 1	
Virtual Domain	VDOM-B
IP/Netmask	11.11.11.2/255.255.255.252

4. Click *OK*.

Configure inter-VDOM routing

Inter-VDOM routing allows users on the internal network to route traffic to the FTP server through the FortiGate.

The configuration of inter-VDOM routing includes the following:

- Firewall addresses for the FTP server on VDOM-A and for the internal network on VDOM-B
- Inter-VDOM routing using static routes for the FTP server on VDOM-A and for the internal network on VDOM-B
- Policies allowing traffic using the VDOM link

The procedures described above require you to connect to both VDOM-A and VDOM-B, either using a global or per-VDOM administrator account.

To add the firewall address on VDOM-A in the GUI:

1. In the *VDOM-A* VDOM, go to *Policy & Objects* > *Addresses* and select Address.
2. Click *Create new*.
3. Enter the following information:

Name	FTP-server
Type	Subnet
IP/Netmask	192.168.20.10/32

Interface	VDOM-link2
Static route configuration	enabled

4. Click *OK*.

To add the static route on VDOM-A in the GUI:

1. Connect to *VDOM-A*.
2. Go to *Network* > *Static Routes* and create a new route.
3. Enter the following information:

Destination	Named Address
Named Address	FTP-server
Gateway	11.11.11.2
Interface	VDOM-link0

4. Click *OK*.

To add the firewall address on VDOM-B in the GUI:

1. In the *VDOM-B* VDOM, go to *Policy & Objects* > *Addresses* and select *Address*.
2. Click *Create new*.
3. Enter the following information:

Name	internal-network
Type	Subnet
IP/Netmask	192.168.10.0/24
Interface	VDOM-link1
Static route configuration	enabled

4. Click *OK*.

To add the static route on VDOM-B in the GUI:

1. In the *VDOM-B* VDOM, go to *Network* > *Static Routes* and create a new route.
2. Enter the following information:

Destination	Named Address
Named Address	internal-network
Gateway	11.11.11.1
Interface	VDOM-link1

3. Click *OK*.

Configure firewall policies using the VDOM link

Firewall policies using the VDOM link allows users on the internal network to access the FTP server through the FortiGate.

Configuring policies allowing traffic using the VDOM link require you to connect to both VDOM-A and VDOM-B, respectively, either using a global or per-VDOM administrator account.

To add the firewall policy on VDOM-A in the GUI:

1. In the *VDOM-A* VDOM, go to *Policy & Objects* > *Firewall Policy*.
2. Click *Create New*.
3. Enter the following information:

Name	Access-FTP-server
Incoming Interface	port1
Outgoing Interface	VDOM-link0
Source	internal-network
Destination	FTP-server
Schedule	always
Service	FTP
Action	ACCEPT
NAT	disabled

4. Click *OK*.

To add the firewall policy on VDOM-B in the GUI:

1. In the *VDOM-B* VDOM, go to *Policy & Objects* > *Firewall Policy*.
2. Click *Create New*.
3. Enter the following information:

Name	Internal-server-access
Incoming Interface	VDOM-link1
Outgoing Interface	port2
Source	internal-network
Destination	FTP-server
Schedule	always
Service	FTP
Action	ACCEPT
NAT	disabled

4. Click *OK*.

Configuration with the CLI

The example can also be configured in the CLI.

To configure the two VDOMs:

1. Enable multi-VDOM mode:

```
config system global
    set vdom-mode multi-vdom
end
```

   You will be logged out of the device when VDOM mode is enabled.

2. Create the VDOMs:

```
config vdom
    edit VDOM-A
    next
    edit VDOM-B
    next
end
```

3. Assign interfaces to the VDOMs:

```
config global
    config system interface
        edit port1
            set vdom VDOM-A
        next
        edit port2
            set vdom VDOM-B
        next
        edit wan1
            set vdom VDOM-A
        next
        edit wan2
            set vdom VDOM-B
        next
    end
end
```

4. Add the firewall addresses to VDOM-A:

```
config vdom
    edit VDOM-A
        config firewall address
            edit internal-network
                set associated-interface port1
                set subnet 192.168.10.0 255.255.255.0
```

```
            next
        end
    next
end
```

5. Add a default route to VDOM-A:

```
config vdom
    edit VDOM-A
        config router static
            edit 0
                set gateway 172.20.201.254
                set device wan1
            next
        end
    next
end
```

6. Add the firewall policy to VDOM-A:

```
config vdom
    edit VDOM-A
        config firewall policy
            edit 1
                set name "VDOM-A-Internet"
                set srcintf "port1"
                set dstintf "wan1"
                set srcaddr "internal-network"
                set dstaddr "all"
                set action accept
                set schedule "always"
                set service "ALL"
                set nat enable
            next
        end
    next
end
```

7. Add the firewall addresses to VDOM-B:

```
config vdom
    edit VDOM-B
        config firewall address
            edit FTP-server
                set associated-interface port2
                set subnet 192.168.20.10 255.255.255.255
            next
        end
    next
end
```

8. Add the virtual IP address to VDOM-B:

```
config vdom
    edit VDOM-B
        config firewall vip
            edit FTP-server-VIP
                set extip 172.20.10.2
                set extintf wan2
                set mappedip 192.168.20.10
            next
        end
    next
end
```

9. Add a default route to VDOM-B:

```
config vdom
    edit VDOM-B
        config router static
            edit 0
                set gateway 172.20.10.254
                set device wan2
            next
        end
    next
end
```

10. Add the firewall policy to VDOM-B:

```
config vdom
    edit VDOM-B
        config firewall policy
            edit 1
                set name "Access-server"
                set srcintf "wan2"
                set dstintf "port2"
                set srcaddr "all"
                set dstaddr "FTP-server-VIP"
                set action accept
                set schedule "always"
                set service "FTP"
                set nat enable
            next
        end
    next
end
```

To configure the VDOM link:

1. Configure the VDOM link:

```
config global
    config system vdom-link
        edit "VDOM-link"
```

```
        next
    end
    config system interface
        edit VDOM-link0
            set vdom VDOM-A
            set ip 11.11.11.1 255.255.255.252
            set allowaccess https ping ssh
            set description "VDOM-A side of the VDOM link"
        next
        edit VDOM-link1
            set vdom VDOM-B
            set ip 11.11.11.2 255.255.255.252
            set allowaccess https ping ssh
            set description "VDOM-A side of the VDOM link"
        next
    end
end
```

2. Configure the firewall addresses on VDOM-A:

```
config vdom
    edit VDOM-A
        config firewall address
            edit "FTP-server"
                set associated-interface "VDOM-link0"
                set allow-routing enable
                set subnet 192.168.20.10 255.255.255.255
            next
        end
    next
end
```

3. Add the firewall policy to VDOM-B:

```
config vdom
    edit VDOM-B
        config firewall policy
            edit 1
                set name "Access-server"
                set srcintf "wan2"
                set dstintf "port2"
                set srcaddr "all"
                set dstaddr "FTP-server-VIP"
                set action accept
                set schedule "always"
                set service "FTP"
                set nat enable
            next
        end
    next
end
```

4. Add the static route on VDOM-A:

```
config vdom
    edit VDOM-A
        config router static
            edit 0
                set device VDOM-link0
                set dstaddr FTP-server
                set gateway 11.11.11.2
            next
        end
    next
end
```

5. Configure the firewall addresses on VDOM-B:

```
config vdom
    edit VDOM-B
        config firewall address
            edit internal-network
                set associated-interface VDOM-link1
                set allow-routing enable
                set subnet 192.168.10.0 255.255.255.0
            next
        end
    next
end
```

6. Add the static route on VDOM-B:

```
config vdom
    edit VDOM-B
        config router static
            edit 0
                set device VDOM-link1
                set dstaddr internal-network
                set gateway 11.11.11.1
            next
        end
    next
end
```

7. Add the security policy on VDOM-A:

```
config vdom
    edit VDOM-A
        config firewall policy
            edit 0
                set name Access-FTP-server
                set srcintf port1
                set dstintf VDOM-link0
                set srcaddr internal-network
                set dstaddr FTP-server
                set action accept
```

```
                set schedule always
                set service FTP
            next
        end
    next
end
```

8. Add the firewall policy on VDOM-B:

```
config vdom
    edit VDOM-B
        config firewall policy
            edit 0
                set name Internal-server-access
                set srcintf VDOM-link1
                set dstintf port2
                set srcaddr internal-network
                set dstaddr FTP-server
                set action accept
                set schedule always
                set service FTP
            next
        end
    next
end
```

SNMP

SNMP enables you to monitor hardware on your network. You can configure the hardware, such as the FortiGate SNMP agent, to report system information and send traps (alarms or event messages) to SNMP managers. SNMP traps alert you to events that happen, such as when a log disk is full or a virus is detected.

The FortiGate SNMP implementation is read-only. SNMP v1/v2c and v3 compliant SNMP managers have read-only access to FortiGate system information through queries, and can receive trap messages from the FortiGate unit. See SNMP Overview for more information.

- Basic configuration on page 3877
- MIB files on page 3881
- Access control for SNMP on page 3882
- Important SNMP traps on page 3884
- SNMP traps and automation-stitch notifications for DIO module on page 3887
- SNMP examples on page 3890

Basic configuration

SNMP configuration has four steps that should be configured in order:

1. Configure interface access

   Before a remote SNMP manager can connect to the FortiGate SNMP agent, you must configure one or more FortiGate interfaces to accept SNMP connections.
2. Configure the SNMP agent

   The SNMP agent sends SNMP traps originating on the FortiGate to an external monitoring SNMP manager defined in an SNMP community. The SNMP manager can monitor the FortiGate system to determine if it is operating properly or if any critical events are occurring.

   The description, location, and contact information for this FortiGate system will be part of the information that the SNMP manager receives. This information is useful if the SNMP manager is monitoring many devices, and enables faster responses when the FortiGate system requires attention.
3. Configure SNMP v1/v2c communities

   An SNMP community is a grouping of equipment for network administration purposes. A single device can belong to multiple communities. It is not mandatory if SNMP v3 is configured.

   You must add an SNMP community to the FortiGate so that the SNMP manager can receive traps and system information. Up to three communities can be added.
4. Configure SNMP v3 users

   Authentication is used to ensure the identity of users. Privacy allows for the encryption of SNMP v3 messages to ensure the confidentiality of data. These protocols provide a higher level of security than is available in SNMP v1/v2c, which use community strings for security. Both authentication and privacy are optional.

To configure SNMP in the GUI:

1. Configure interface access:
   a. Go to *Network > Interfaces* and edit an interface.
   b. In the *Administrative Access* options, enable *SNMP*.
   c. Click *OK*.
2. Configure the SNMP agent:
   a. Go to *System > SNMP*.
   b. Enable *SNMP Agent* and configure the following:

Description	A description of the agent.
Location	The location of the FortiGate.
Contact Info	A contact or administrator for the SNMP agent or FortiGate.

   c. Click *Apply*.
3. Configure an SNMP v1/v2c community:
   a. Go to *System > SNMP*.
   b. In the *SNMP v1/v2c* table, click *Create New*.
   c. Configure the following:

Community Name	The name of the community.
Hosts	Enter the *IP Address* and select the *Host Type* for each SNMP manager.

Queries	Enable or disable v1 and v2c queries, then enter the port numbers that the SNMP managers in this community use for them.
Traps	Enable or disable v1 and v2c traps, then enter the local and remote port numbers that the SNMP managers in this community use for them.
SNMP Events	Enable or disable the events that activate traps in this community.

d. Click *OK*.

4. Configure an SNMP v3 user:
   a. Go to *System* > *SNMP*.
   b. In the *SNMP v3* table, click *Create New*.
   c. Configure the following:

User Name	The name of the user.
Security Level	Configure the security level: • *No Authentication*: No authentication or encryption. • *Authentication*: Select the authentication algorithm and password. • *Authentication and Private*: Select both the authentication and encryption algorithms and password.
Hosts	The *IP Address* for each SNMP manager.
Queries	Enable or disable queries, then enter the port number that the SNMP managers use for them.
Traps	Enable or disable traps, then enter the local and remote port numbers that the SNMP managers use for them
SNMP Events	Enable or disable the events that activate traps.

   d. Click *OK*.

To configure SNMP in the CLI:

1. Configure the Interface access:

```
config system interface
    edit <interface>
        append allowaccess snmp
        config ipv6
            append ip6-allowaccess snmp
        end
    next
end
```

2. Configure the SNMP agent:

```
config system snmp sysinfo
    set status enable
```

```
    set description <string>
    set contact-info <string>
    set location <string>
end
```

3. Configure an SNMP v1/v2c community:

```
config system snmp community
    edit <id>
        set name <string>
        set status {enable | disable}
        config hosts
            edit <host_id>
                set interface-select-method {auto | sdwan | specify}
                set ip <ip/mask>
                set source-ip <class_ip>
                set ha-direct {enable | disable}
                set host-type {any | query | trap}
            next
        end
        set query-v1-port <port_number>
        set query-v1-status {enable | disable}
        set query-v2c-port <port_number>
        set query-v2c-status {enable | disable}
        set trap-v1-lport <port_number>
        set trap-v1-rport <port_number>
        set trap-v1-status {enable | disable}
        set trap-v2c-lport <port_number>
        set trap-v2c-rport <port_number>
        set trap-v2c-status {enable | disable}
        set events <events>
    next
end
```

4. Configure an SNMP v3 user:

```
config system snmp user
    edit <user>
        set status {enable | disable}
        set trap-status {enable | disable}
        set trap-lport <port_number>
        set trap-rport <port_number>
        set queries {enable | disable}
        set query-port <port_number>
        set notify-hosts <class_ip> ... <class_ip>
        set source-ip <class_ip>
        set ha-direct {enable | disable}
        set interface-select-method {auto | sdwan | specify}
        set events <events>
        set security-level {no-auth-no-priv | auth-no-priv | auth-priv}
        set auth-proto {md5 | sha | sha224 | sha256 | sha384 | sha512}
        set auth-pwd <password>
```

```
        set priv-proto {aes | des | aes256 | aes256cisco}
        set priv-pwd <password>
    next
end
```

See SNMP examples on page 3890 for sample configurations.

MIB files

The FortiGate SNMP agent supports Fortinet proprietary MIBs, as well as the parts of RFC 2665 and RFC 1213 that apply to FortiGate unit configuration.

Your SNMP manager may already include standard and private MIBs in a compiled database that is ready to use. You must add the Fortinet proprietary MIBs to this database to have access to Fortinet specific information.

MIB file or RFC	Description
FORTINET-CORE-MIB.mib	The Fortinet core MIB includes all system configuration and trap information that is common to all Fortinet products. Your SNMP manager requires this information to monitor Fortinet device settings and receive traps from the FortiGate SNMP agent.
FORTINET-FORTIGATE-MIB.mib	The FortiGate MIB includes all system configuration information and trap information that is specific to FortiGate units. Your SNMP manager requires this information to monitor FortiGate settings and receive traps from the FortiGate SNMP agent.
RFC-1213 (MIB II)	The FortiGate SNMP agent supports MIB II groups with the following exceptions: • No support for the EGP group from MIB II (RFC 1213, section 3.11 and 6.10). • Protocol statistics returned for MIB II groups (IP/ICMP/TCP/UDP/etc.) do not accurately capture all Fortinet traffic activity. More accurate information can be obtained from the information reported by the Fortinet MIB.
RFC-2665 (Ethernet-like MIB)	The FortiGate SNMP agent supports Ethernet-like MIB information. FortiGate SNMP does not support for the dot3Tests and dot3Errors groups.

To download the MIB files:

1. Go to *System* > *SNMP*.
2. Click *Download FortiGate MIB File* and save the file to the management computer.
3. Click *Download Fortinet Core MIB File* and save the file to the management computer.

Access control for SNMP

Administrators can provide access control to SNMP users and communities based on restricting a MIB view to specific OID subtrees. They can also define access based on the VDOM. This allows multi-tenant FortiGate deployments to provide restricted access per VDOM.

- MIB view access control allows the SNMP clients to query specific OIDs that are filtered by the MIB view settings.
- VDOM access control allows the SNMP clients to query data from specific VDOMs that are filtered by the VDOM settings.

When access control is enabled, the users can only access the information that is allowed by the access control, and all other information is inaccessible. Administrators have granular control, and can easily restrict specific information based on access control.

To configure MIB views:

```
config system snmp mib-view
    edit <name>
        set include <OIDs>
        set exclude <OIDs>
    next
end
```

Command	Description
`include <OIDs>`	Enter the OID subtrees to be included in the view. A maximum of 16 subtrees can be added.
`exclude <OIDs>`	Enter the OID subtrees to be excluded in the view. A maximum of 64 subtrees can be added.

To configure access control based on MIB views and VDOMs for SNMP users and communities:

```
config system snmp user
    edit <user>
        set mib-view <view>
        set vdoms <vdoms>
    next
end
```

```
config system snmp community
    edit <community>
        set mib-view <view>
        set vdoms <vdoms>
    next
end
```

Command	Description
`mib-view <view>`	Set the SNMP access control MIB view.
`vdoms <vdoms>`	Set the SNMP access control VDOMs.

Example

In this example, two MIB views are created and, with VDOMs, used to control access for SNMP users and communities.

To configure access control for SNMP users and communities:

1. Configure two MIB views:

```
config system snmp mib-view
    edit "view1"
        set include "1.3.6.1.2"
    next
    edit "view2"
        set include "1.3.6.1.2.1"
        set exclude "1.3.6.1.2.1.2.1" "1.3.6.1.2.1.4.31" "1.3.6.1.2.1.1.9.1"
    next
end
```

2. Add the MIB view and VDOM restrictions to SNMP users:

```
config system snmp user
    edit "v3user"
        set mib-view "view1"
    next
    edit "v3user1"
        set vdom "vdom1"
    next
    edit "v3user2"
        set mib-view "view1"
        set vdoms "root" "vdom1"
    next
end
```

3. Add the MIB view and VDOM restrictions to SNMP communities:

```
config system snmp community
    edit 1
        set name "REGR-SYS"
        set vdoms "vdom1"
    next
    edit 2
        set name "REGR-SYS1"
        set mib-view "view2"
    next
```

```
    edit 3
       set name "REGR-SYS2"
       set mib-view "view1"
       set vdoms "root" "vdom1"
    next
end
```

Important SNMP traps

Link Down and Link Up traps

This trap is sent when a FortiGate port either goes down or is brought up.

For example, the following traps are generated when the state of port34 is set to down using `set status down`, and then brought up using `set status up`:

```
NET-SNMP version 5.7.3 2019-01-31 14:11:48 10.1.100.1(via UDP: [10.1.100.1]:162->
[10.1.100.11]:162) TRAP, SNMP v1, community REGR-SYS SNMPv2-MIB::snmpTraps Link Down Trap (0)
Uptime: 0:14:44.95 IF-MIB::ifIndex.42 = INTEGER: 42 IF-MIB::ifAdminStatus.42 = INTEGER: down(2)
IF-MIB::ifOperStatus.42 = INTEGER: down(2) FORTINET-CORE-MIB::fnSysSerial.0 = STRING:
FG140P3G15800330 SNMPv2-MIB::sysName.0 = STRING: FortiGate-140D-POE
```

```
2019-01-31 14:11:48 <UNKNOWN> [UDP: [10.1.100.1]:162->[10.1.100.11]:162]: DISMAN-EVENT-
MIB::sysUpTimeInstance = Timeticks: (88495) 0:14:44.95 SNMPv2-MIB::snmpTrapOID.0 = OID: IF-
MIB::linkDown IF-MIB::ifIndex.42 = INTEGER: 42 IF-MIB::ifAdminStatus.42 = INTEGER: down(2) IF-
MIB::ifOperStatus.42 = INTEGER: down(2) FORTINET-CORE-MIB::fnSysSerial.0 = STRING:
FG140P3G15800330 SNMPv2-MIB::sysName.0 = STRING: FortiGate-140D-POE 2019-01-31 14:12:01 10.1.100.1
(via UDP: [10.1.100.1]:162->[10.1.100.11]:162) TRAP, SNMP v1, community REGR-SYS SNMPv2-
MIB::snmpTraps Link Up Trap (0) Uptime: 0:14:57.98 IF-MIB::ifIndex.42 = INTEGER: 42 IF-
MIB::ifAdminStatus.42 = INTEGER: up(1) IF-MIB::ifOperStatus.42 = INTEGER: up(1) FORTINET-CORE-
MIB::fnSysSerial.0 = STRING: FG140P3G15800330 SNMPv2-MIB::sysName.0 = STRING: FortiGate-140D-POE
```

```
2019-01-31 14:12:01 <UNKNOWN> [UDP: [10.1.100.1]:162->[10.1.100.11]:162]: DISMAN-EVENT-
MIB::sysUpTimeInstance = Timeticks: (89798) 0:14:57.98 SNMPv2-MIB::snmpTrapOID.0 = OID: IF-
MIB::linkUp IF-MIB::ifIndex.42 = INTEGER: 42 IF-MIB::ifAdminStatus.42 = INTEGER: up(1) IF-
MIB::ifOperStatus.42 = INTEGER: up(1) FORTINET-CORE-MIB::fnSysSerial.0 = STRING: FG140P3G15800330
SNMPv2-MIB::sysName.0 = STRING: FortiGate-140D-POE
```

fgFmTrapIfChange trap

This trap is sent when any changes are detected on the interface. The change can be very simple, such as giving an IPV4 address.

For example, the user has given the IP address of 1.2.3.4/24 to port 1 and the EMS Manager has detected the following trap:

```
DISMAN-EXPRESSION-MIB::sysUpTimeInstance = Timeticks: (7975058) 22:09:10.58 SNMPv2-
MIB::snmpTrapOID.0 = OID: FORTINET-FORTIGATE-MIB::fgFmTrapIfChange FORTINET-CORE-
```

```
MIB::fnSysSerial.0 = STRING: FG140P3G15800330 IF-MIB::ifName.45 = STRING: port1 FORTINET-
FORTIGATE-MIB::fgManIfIp.0 = IpAddress: 1.2.3.4 FORTINET-FORTIGATE-MIB::fgManIfMask.0 = IpAddress:
255.255.255.0 FORTINET-FORTIGATE-MIB::fgManIfIp6.0 = STRING: 0:0:0:0:0:0:0:0
```

entConfigChange trap

The change to the interface in the previous example has also triggered the *ConfChange Trap* which is sent along with the *fgFmTrapIfChange* trap:

```
2018-11-15 09:30:23 FGT_A [UDP: [172.16.200.1]:162->[172.16.200.55]:162]: DISMAN-EXPRESSION-
MIB::sysUpTimeInstance = Timeticks: (8035097) 22:19:10.97 SNMPv2-MIB::snmpTrapOID.0 = OID: ENTITY-
MIB::entConfigChange
```

fgTrapDeviceNew trap

This trap is triggered when a new device, like a FortiSwitch, is connected to the FortiGate.

For example, the following scenario has given the device a new trap for adding FortiAP on a PoE interface a FortiGate 140D-POE. The trap has important information about the device name, device MAC address, and when it was last seen.

```
2018-11-15 11:17:43 UDP/IPv6: [2000:172:16:200::1]:162 [UDP/IPv6: [2000:172:16:200::1]:162]:
DISMAN-EXPRESSION-MIB::sysUpTimeInstance = Timeticks: (520817) 1:26:48.17 SNMPv2-
MIB::snmpTrapOID.0 = OID: FORTINET-FORTIGATE-MIB::fgTrapDeviceNew FORTINET-CORE-MIB::fnSysSerial.0
= STRING: FG140P3G15800330 SNMPv2-MIB::sysName.0 = STRING: FGT_A IF-MIB::ifIndex.0 = INTEGER: 0
FORTINET-FORTIGATE-MIB::fgVdEntIndex.0 = INTEGER: 0 FORTINET-FORTIGATE-MIB::fgDeviceCreated.0 =
Gauge32: 5 FORTINET-FORTIGATE-MIB::fgDeviceLastSeen.0 = Gauge32: 5 FORTINET-FORTIGATE-
MIB::fgDeviceMacAddress.0 = STRING: 90:6c:ac:f9:97:a0
```

```
2018-11-15 11:17:43 FGT_A [UDP: [172.16.200.1]:162->[172.16.200.55]:162]: DISMAN-EXPRESSION-
MIB::sysUpTimeInstance = Timeticks: (520817) 1:26:48.17 SNMPv2-MIB::snmpTrapOID.0 = OID: FORTINET-
FORTIGATE-MIB::fgTrapDeviceNew FORTINET-CORE-MIB::fnSysSerial.0 = STRING: FG140P3G15800330 SNMPv2-
MIB::sysName.0 = STRING: FGT_A IF-MIB::ifIndex.0 = INTEGER: 0 FORTINET-FORTIGATE-
MIB::fgVdEntIndex.0 = INTEGER: 0 FORTINET-FORTIGATE-MIB::fgDeviceCreated.0 = Gauge32: 5 FORTINET-
FORTIGATE-MIB::fgDeviceLastSeen.0 = Gauge32: 5 FORTINET-FORTIGATE-MIB::fgDeviceMacAddress.0 =
STRING: 90:6c:ac:f9:97:a0
```

fgTrapAvOversize trap

The `fgTrapAvOversize` trap is generated when the antivirus scanner detects an oversized file:

```
019-01-31 13:22:04 10.1.100.1(via UDP: [10.1.100.1]:162->[10.1.100.11]:162) TRAP, SNMP v1,
community REGR-SYS FORTINET-FORTIGATE-MIB::fgt140P Enterprise Specific Trap (602) Uptime: 1 day,
3:41:10.31 FORTINET-CORE-MIB::fnSysSerial.0 = STRING: FG140P3G15800330 SNMPv2-MIB::sysName.0 =
STRING: FortiGate-140D-POE 2019-01-31 13:22:29 <UNKNOWN> [UDP: [10.1.100.1]:162->
[10.1.100.11]:162]: DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (9967031) 1 day, 3:41:10.31
SNMPv2-MIB::snmpTrapOID.0 = OID: FORTINET-FORTIGATE-MIB::fgTrapAvOversize FORTINET-CORE-
MIB::fnSysSerial.0 = STRING: FG140P3G15800330 SNMPv2-MIB::sysName.0 = STRING: FortiGate-140D-POE
```

BIOS security level trap

SNMP clients can query the BIOS security level of a FortiGate using the OID 1.3.6.1.4.1.12356.101.4.1.38.

Memory usage traps

Both free memory usage and freeable memory of FortiGate devices can be monitored through the Simple Network Management Protocol (SNMP). SNMP object identifier (OID) entries are available in Fortinet MIB files to show the percentage of free memory usage and freeable memory in an SNMP manager:

- `1.3.6.1.4.1.12356.101.4.1.36 .fgSysFreeMemUsage`
- `1.3.6.1.4.1.12356.101.4.1.37 .fgSysFreeableMemUsage`

The following commands are available to configure memory thresholds to trigger SNMP traps:

```
config system snmp sysinfo
    set trap-free-memory-threshold <integer>
    set trap-freeable-memory-threshold <integer>
end
```

Command	Description
`trap-free-memory-threshold <integer>`	Use an integer from 1 to 100 (default 5) to identify what percentage of free memory usage will trigger an SNMP trap. SNMP traps are sent when the free memory is *lower* than the specified threshold. For example, the free memory threshold is set to 5, and SNMP traps are sent when free memory is lower than 5%.
`trap-freeable-memory-threshold <integer>`	Use an integer from 1 to 100 (default 60) to identify what percentage of freeable memory will trigger an SNMP trap. SNMP traps are sent when the freeable memory is *higher* than the specified threshold. For example, the freeable memory threshold is set to 60, and SNMP traps are sent when freeable memory is higher than 60%.

Example

In this example, the SNMP agent is configured to monitor FortiGate memory and send traps. The `trap-free-memory-threshold` is set to `10`, and the `trap-freeable-memory-threshold` is set to `50`. SNMP traps are triggered for both thresholds because:

- The free memory on the FortiGate is 9%, which is lower than the threshold of 10.
- The freeable memory on the FortiGate is 56%, which is higher than the threshold of 50.

To configure SNMP for monitoring memory usage on FortiGates:

1. Configure the SNMP agent to monitor FortiGate memory usage and freeable memory.
   In this example, the `trap-free-memory-threshold` is set to `10`, and the `trap-freeable-memory-threshold` is set to `50`.

```
config system snmp sysinfo
    set status enable
    set engine-id <string for local SNMP engine ID>
    set description <string>
    set contact-info <string>
    set location <string>
    set trap-high-cpu-threshold 60
    set trap-free-memory-threshold 10
    set trap-freeable-memory-threshold 50
end
```

2. Verify that the SNMP manager can successfully query and receive a response on the current memory status of the FortiGate.

   In the following example, the free memory on the FortiGate is reported as 9%, and the freeable memory on the FortiGate is reported as 56%.

```
# snmpwalk -v2c -c REGR-SYS 172.16.200.1 1.3.6.1.4.1.12356.101.4.1.36
FORTINET-FORTIGATE-MIB::fgSystemInfo.36.0 = Gauge32: 9
fosqa@pc05:~$ snmpwalk -v2c -c REGR-SYS 172.16.200.1 1.3.6.1.4.1.12356.101.4.1.37
FORTINET-FORTIGATE-MIB::fgSystemInfo.37.0 = Gauge32: 56
```

3. Use the SNMP manager to monitor memory usage on the FortiGate.

   Following is an example of the SNMP trap messages sent when thresholds are surpassed for freeable memory and free memory usage on FortiGates:

```
2023-12-08 19:53:14 172.16.200.1(via UDP: [172.16.200.1]:162->[172.16.200.55]:162) TRAP, SNMP
v1, community REGR-SYS
        FORTINET-FORTIGATE-MIB::fgModel.1001 Enterprise Specific Trap (102) Uptime: 1 day,
9:49:42.35
        FORTINET-CORE-MIB::fnSysSerial.0 = STRING: FG101FTK20006858     SNMPv2-MIB::sysName.0
= STRING: FGT_A   FORTINET-CORE-MIB::fnGenTrapMsg = STRING: freeable memory percentage is too
high
2023-12-08 19:56:33 <UNKNOWN> [UDP: [172.16.200.1]:162->[172.16.200.55]:162]:
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (12198187) 1 day, 9:53:01.87   SNMPv2-
MIB::snmpTrapOID.0 = OID: FORTINET-CORE-MIB::fnTrapMemThreshold       FORTINET-CORE-
MIB::fnSysSerial.0 = STRING: FG101FTK20006858     SNMPv2-MIB::sysName.0 = STRING: FGT_A
 FORTINET-CORE-MIB::fnGenTrapMsg = STRING: free memory percentage is too low
```

SNMP traps and automation-stitch notifications for DIO module

FortiGate Rugged 70F and FortiGate Rugged 70F-3G4G include a general purpose input output (GPIO) module, also known as a digital I/O (DIO) module. The module supports SNMP traps and automation-stitch notifications when DIO module alarm functionality is activated. The DIO module triggers an alarm when it detects a change in any digital input, and the digital output is activated. Notification support depends on previously configured `config system digital-io` and `execute digital-io set-output` settings prior to event notification. See FGR-70G/70F/70F-3G4G GPIO/DIO module on page 129 for more information.

CLI for configuring SNMP traps and automation-stitch notifications is available only on FortiGate Rugged 70F and FortiGate Rugged 70F-3G4G devices.

The `config system automation-condition` command includes the following relevant options:

```
config system automation-condition
    edit <name>
        set condition-type input
        set input-state {open|close}
    next
end
```

Command	Description
`condition-type input`	Configure the type of condition to `input` for the DIO module on the FortiGate 70F series.
`input-state {open|close}`	Configure the input state: • `open`: Input switch is open. • `close`: Input switch is closed.

The `config system snmp community` command includes the following relevant option:

```
config system snmp community
    edit <id>
        set events dio
    next
end
```

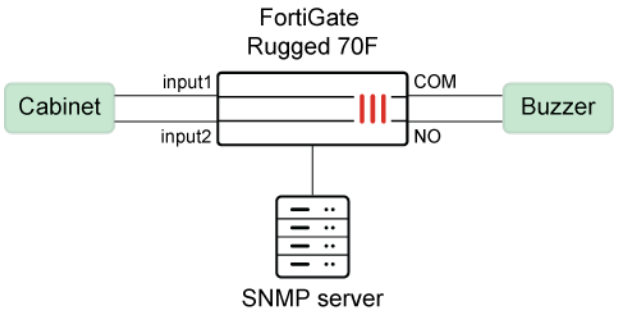
Command	Description
`events dio`	Configure the SNMP trap event for the DIO module of the FortiGate Rugged 70F series. When enabled, system events are also logged.

For more information about the DIO module, see the FortiGate Rugged 70F Series QuickStart Guide and the Technical Tip: Overview of the Digital Input/Output (DIO) Module in FortiGate Rugged 70F Series community article.

Example

In this example, a FortiGate Rugged 70F is configured to monitor the open/close status of a cabinet door, and the output is connected to a buzzer. An automation stitch and SNMP trap are also configured.

When the status of the cabinet door changes from open to closed or closed to open, FortiGate triggers the buzzer, automation stitch, and SNMP trap. A system event is also logged when SNMP traps are sent.



Before you configure automation stitches and SNMP traps for DIO module alarms, you must configure the alarms using the `config system digital-io` and `execute digital-io set-output` settings. See FGR-70G/70F/70F-3G4G GPIO/DIO module on page 129 for more information.

To configure an automation stitch for DIO module alarms:

1. Configure an automation-stitch condition for when the DIO module detects an input state of open:
   In this example, the condition type is set to `input`, and the input state is set to `open`.

```
config system automation-condition
    edit "Cabinet-Open"
        set description "Cabinet open"
        set condition-type input
        set input-state open
    next
end
```

2. Configure an automation-stitch trigger:

```
config system automation-trigger
    edit "DIO-trigger"
        set description "DIO-trigger"
    next
end
```

3. Configure a stitch to use the condition to trigger an action, such as an email notification:
   In this example, the automation stitch uses the previously configured trigger (DIO-trigger) and condition (Cabinet-Open) to trigger an email notification.

```
config system automation-stitch
    edit "dio"
        set description "DIO-stitch"
        set trigger "DIO-trigger"
        set condition "Cabinet-Open"
        config actions
            edit 1
                set action "Email Notification"
                set required enable
            next
        end
```

```
    next
end
```

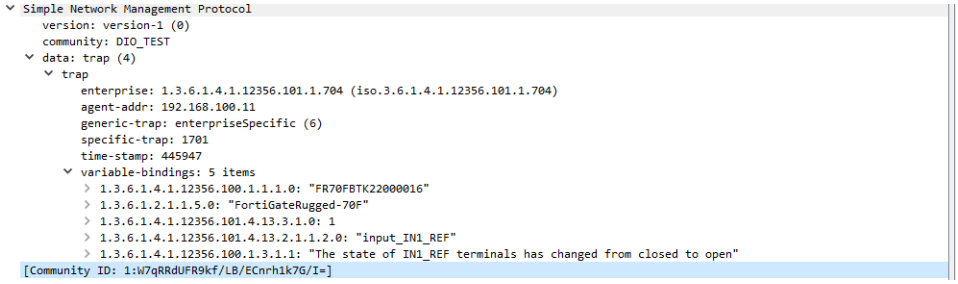
To configure an SNMP trap for DIO module alarms:

1. Configure a DIO module event in an SNMP community:
   With `set events dio` configured, SNMP traps are triggered for DIO module alarms.

```
config system snmp community
    edit 1
        set name "DIO_TEST"
        config hosts
            edit 1
                set ip 172.16.200.55 255.255.255.255
            next
        end
        set events dio
    next
end
```

Results:

When the cabinet door being monitored by the DIO module opens unexpectedly, it triggers an SNMP event:



It also triggers a system log:

```
9: date=2024-11-18 time=17:26:13 eventtime=1731979573581176380 tz="-0800" logid="0100022907"
type="event" subtype="system" level="notice" vd="root" logdesc="Digital-IO input state change"
connector="input_IN1_REF" mode="default" state="open" msg="The state of IN1_REF terminals has
changed from closed to open"
```

When the cabinet door is opened, it also triggers an email notification as configured by the automation stitch.

SNMP examples

This topic includes examples that incorporate several SNMP settings:

- Example 1: SNMP traps for monitoring interface status using SNMP v3 user on page 3891
- Example 2: SNMP traps and query for monitoring DHCP pool using SNMP v3 user on page 3893
- Example 3: Enabling the INDEX extension on page 3895

- Example 4: Ethernet Statistics Group on page 3897
- Example 5: Enabling non-management VDOMs to send queries using SNMP v3 on page 3898

Example 1: SNMP traps for monitoring interface status using SNMP v3 user

This configuration enables the SNMP manager (172.16.200.55) to receive notifications when a FortiGate port either goes down or is brought up. The SNMP manager can also query the current status of the FortiGate port.

To configure SNMP for monitoring interface status in the GUI:

1. Configure interface access:
   a. Go to *Network* > *Interfaces* and edit *port1*.
   b. In the *Administrative Access* options, enable *SNMP*.
   c. Click *OK*.
2. Configure the SNMP agent:
   a. Go to *System* > *SNMP*.
   b. Enable *SNMP Agent* and enter the following:

Description	Branch
Location	Burnaby
Contact Info	Jane Doe

   c. Click *Apply*.
3. Configure an SNMP v3 user:
   a. Go to *System* > *SNMP*.
   b. In the *SNMP v3* table, click *Create New*.
   c. Configure the following:

User Name	Interface_Status
Security Level	Authentication
Authentication Algorithm	SHA1
Password	*******
Hosts IP Address	172.16.200.55

   d. Click *OK*.

To configure SNMP for monitoring interface status in the CLI:

1. Configure interface access:

```
config system interface
    edit port1
        append allowaccess snmp
```

```
    next
end
```

2. Configure the SNMP agent:

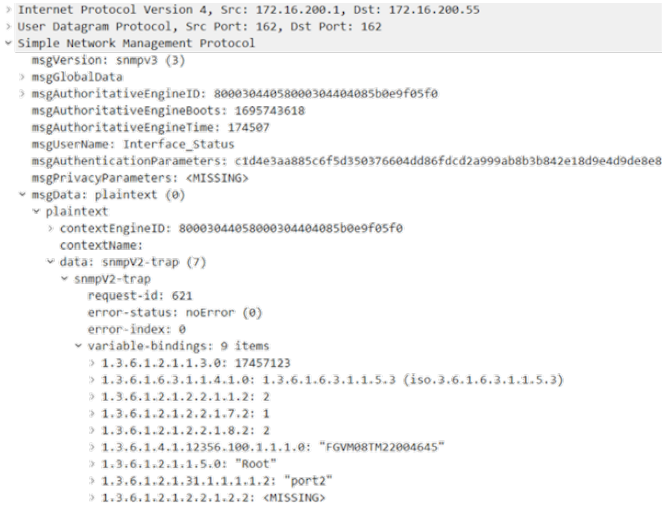
```
config system snmp sysinfo
    set status enable
    set description Branch
    set contact-info Jane Doe
    set location Burnaby
end
```

3. Configure an SNMP v3 user:

```
config system snmp user
    edit "Interface_Status"
        set notify-hosts 172.16.200.55
        set security-level auth-no-priv
        set auth-proto sha
        set auth-pwd ********************
    next
end
```

Verification

1. Start the packet capture on interface port1 with the filter set to port 162. See Using the packet capture tool on page 1025 for more information.
2. Turn off one of the FortiGate interface statuses to down, in this case, port2.
3. Save the packet capture.



The SNMP v3 trap is transmitted from port1 to the SNMP manager. Note that `msgAuthenticationParameters` is configured, indicating that authentication is active. The absence of `msgPrivacyParameters` suggests that encryption is not configured. This is further confirmed by `plaintext` in `msgData`.

4. Verify that the SNMP manager has received the trap. See Important SNMP traps on page 3884 for an example of a trap.

5. Verify that the SNMP manager can successfully query and receive receive a response on the current status of the FortiGate ports:

```
# snmpwalk -v3 -u Interface_Status  -l authNoPriv -a SHA -A xxxxxxxx 172.16.200.1
1.3.6.1.2.1.2.2.1.8
iso.3.6.1.2.1.2.2.1.8.1 = INTEGER: 1
iso.3.6.1.2.1.2.2.1.8.2 = INTEGER: 2
iso.3.6.1.2.1.2.2.1.8.3 = INTEGER: 1
iso.3.6.1.2.1.2.2.1.8.4 = INTEGER: 1
iso.3.6.1.2.1.2.2.1.8.5 = INTEGER: 1
iso.3.6.1.2.1.2.2.1.8.6 = INTEGER: 1
iso.3.6.1.2.1.2.2.1.8.7 = INTEGER: 1
iso.3.6.1.2.1.2.2.1.8.8 = INTEGER: 1
iso.3.6.1.2.1.2.2.1.8.9 = INTEGER: 1
iso.3.6.1.2.1.2.2.1.8.10 = INTEGER: 1
```

Example 2: SNMP traps and query for monitoring DHCP pool using SNMP v3 user

This configuration enables the SNMP manager (172.16.200.55) to receive DHCP-related notifications from FortiGate.

The SNMP DHCP event contains three traps and one query.

Traps are sent when:

- DHCP server IP pool usage reaches 90%
- DHCP server detect an IP address that is already in use
- DHCP client receives DHCP NAK

SNMP queries are accepted for DHCP lease usage information (OID = 1.3.6.1.4.1.12356.101.23). The query result is based on the leased out percentage.

To configure SNMP for monitoring DHCP pool in the GUI:

1. Configure interface access:
   a. Go to *Network* > *Interfaces* and edit *port1*.
   b. In the *Administrative Access* options, enable *SNMP*.
   c. Click *OK*.
2. Configure the SNMP agent:
   a. Go to *System* > *SNMP*.
   b. Enable *SNMP Agent* and enter the following:

Description	Branch
Location	Burnaby
Contact Info	Jane Doe

   c. Click *Apply*.
3. Configure an SNMP v3 user:

a. Go to *System* > *SNMP*.
b. In the *SNMP v3* table, click *Create New*.
c. Configure the following:

User Name	DHCP_Status
Security Level	Authentication
Authentication Algorithm	SHA384
Password	*******
Private	Enabled
Encryption Algorithm	AES256
Password	*******
Hosts IP Address	172.16.200.55

d. Click *OK*.

To configure SNMP for monitoring the DHCP pool in the CLI:

1. Configure interface access:

```
config system interface
    edit port1
        append allowaccess snmp
    next
end
```

2. Configure the SNMP agent:

```
config system snmp sysinfo
    set status enable
    set description Branch
    set contact-info Jane Doe
    set location Burnaby
end
```

3. Configure an SNMP v3 user:

```
config system snmp user
    edit "DHCP_Status"
        set notify-hosts 172.16.200.55
        set security-level auth-priv
        set auth-proto sha384
        set auth-pwd ********************
        set priv-proto aes256
        set priv-pwd **********************
    next
end
```

Verification

1. Start the packet capture on interface port1 with the filter set to port 162. See Using the packet capture tool on page 1025 for more information.
2. Overload the DHCP server IP pool.
3. Save the packet capture.

   The SNMP v3 trap is transmitted from port1 to the SNMP manager. Note that both `msgAuthenticationParameters` and `msgPrivacyParameters` are set up, indicating that authentication and encryption are active. This is further confirmed by `encryptedPDU` in `msgData`.
4. Verify that the SNMP manager has received the trap. See Important SNMP traps on page 3884 for an example of a trap.
5. Verify that the SNMP manager can successfully query and receive DHCP lease usage information for FortiGate:

```
# snmpwalk -v3 -u DHCP_Status  -l authPriv -a SHA384 -A xxxxxxxx -x AES256 -X xxxxxxxx
172.16.200.1 1.3.6.1.4.1.12356.101.23
iso.3.6.1.4.1.12356.101.23.1.1.0 = INTEGER: 6
iso.3.6.1.4.1.12356.101.23.2.1.1.2.1.1 = INTEGER: 0
iso.3.6.1.4.1.12356.101.23.2.1.1.2.1.2 = INTEGER: 0
iso.3.6.1.4.1.12356.101.23.2.1.1.2.1.3 = INTEGER: 0
iso.3.6.1.4.1.12356.101.23.2.1.1.2.1.4 = INTEGER: 0
iso.3.6.1.4.1.12356.101.23.2.1.1.2.1.5 = INTEGER: 0
iso.3.6.1.4.1.12356.101.23.2.1.1.2.1.6 = INTEGER: 100
```

Example 3: Enabling the INDEX extension

In the following example, the same IP address will be set on different ports in two VDOMs. The ipAddrTable SNMP Tree output will then be reviewed before and after enabling the *append-index* command.

When the `append-index` command is enabled:

- VDOM and interface indexes are appended as the INDEX extension in RFC tables.
- In multi-VDOM mode, duplicated IP addresses in different VDOMs will result in multiple entries in the RFC table.

When the `append-index` command is disabled:

- VDOM and interface indexes are not appended in RFC tables.
- In multi-VDOM mode, duplicated IP addresses in different VDOMs will only be presented once in the RFC table.

To enable the INDEX extension:

1. In two different VDOMs, set the same address on two different ports.

```
config system interface
    edit "port3"
        set vdom "vdom1"
        set ip 10.1.1.1 255.255.255.0
        set type physical
        set snmp-index 5
    next
end
config system interface
    edit "port4"
        set vdom "root"
        set ip 10.1.1.1 255.255.255.0
        set type physical
        set snmp-index 6
    next
end
```

2. Configure the SNMP information but do not enable the INDEX extension.

```
config system snmp sysinfo
    set status enable
    set description "REGR-SYS"
end
```

3. On your PC, review the ipAddrTable SNMP Tree (OID 1.3.6.1.2.1.4.20). The IP address 10.1.1.1 is only displayed once.

```
snmpwalk -v2c -c REGR-SYS 172.16.200.1 1.3.6.1.2.1.4.20

IP-MIB::ipAdEntAddr.10.1.1.1 = IpAddress: 10.1.1.1
IP-MIB::ipAdEntAddr.10.255.1.1 = IpAddress: 10.255.1.1
IP-MIB::ipAdEntAddr.172.16.200.1 = IpAddress: 172.16.200.1
IP-MIB::ipAdEntAddr.192.168.1.99 = IpAddress: 192.168.1.99
IP-MIB::ipAdEntIfIndex.10.1.1.1 = INTEGER: 5
IP-MIB::ipAdEntIfIndex.10.255.1.1 = INTEGER: 39
IP-MIB::ipAdEntIfIndex.172.16.200.1 = INTEGER: 3
IP-MIB::ipAdEntIfIndex.192.168.1.99 = INTEGER: 2
IP-MIB::ipAdEntNetMask.10.1.1.1 = IpAddress: 255.255.255.0
IP-MIB::ipAdEntNetMask.10.255.1.1 = IpAddress: 255.255.255.0
IP-MIB::ipAdEntNetMask.172.16.200.1 = IpAddress: 255.255.255.0
IP-MIB::ipAdEntNetMask.192.168.1.99 = IpAddress: 255.255.255.0
IP-MIB::ipAdEntBcastAddr.10.1.1.1 = INTEGER: 1
IP-MIB::ipAdEntBcastAddr.10.255.1.1 = INTEGER: 1
IP-MIB::ipAdEntBcastAddr.172.16.200.1 = INTEGER: 1
IP-MIB::ipAdEntBcastAddr.192.168.1.99 = INTEGER: 1
IP-MIB::ipAdEntReasmMaxSize.10.1.1.1 = INTEGER: 65535
IP-MIB::ipAdEntReasmMaxSize.10.255.1.1 = INTEGER: 65535
IP-MIB::ipAdEntReasmMaxSize.172.16.200.1 = INTEGER: 65535
IP-MIB::ipAdEntReasmMaxSize.192.168.1.99 = INTEGER: 65535
```

4. Enable the INDEX extension.

```
config system snmp sysinfo
    set status enable
    set description "REGR-SYS"
    set append-index enable
end
```

5. Review the ipAddrTable SNMP Tree (OID 1.3.6.1.2.1.4.20) again. The IP address 10.1.1.1 is now displayed twice.

```
snmpwalk -v2c -c REGR-SYS 172.16.200.1 1.3.6.1.2.1.4.20

IP-MIB::ipAdEntAddr.10.1.1.1.1 = IpAddress: 10.1.1.1
IP-MIB::ipAdEntAddr.10.1.1.1.2 = IpAddress: 10.1.1.1
IP-MIB::ipAdEntAddr.10.255.1.1.1 = IpAddress: 10.255.1.1
IP-MIB::ipAdEntAddr.172.16.200.1.2 = IpAddress: 172.16.200.1
IP-MIB::ipAdEntAddr.192.168.1.99.1 = IpAddress: 192.168.1.99
IP-MIB::ipAdEntIfIndex.10.1.1.1.1 = INTEGER: 6
IP-MIB::ipAdEntIfIndex.10.1.1.1.2 = INTEGER: 5
IP-MIB::ipAdEntIfIndex.10.255.1.1.1 = INTEGER: 39
IP-MIB::ipAdEntIfIndex.172.16.200.1.2 = INTEGER: 3
IP-MIB::ipAdEntIfIndex.192.168.1.99.1 = INTEGER: 2
IP-MIB::ipAdEntNetMask.10.1.1.1.1 = IpAddress: 255.255.255.0
IP-MIB::ipAdEntNetMask.10.1.1.1.2 = IpAddress: 255.255.255.0
IP-MIB::ipAdEntNetMask.10.255.1.1.1 = IpAddress: 255.255.255.0
IP-MIB::ipAdEntNetMask.172.16.200.1.2 = IpAddress: 255.255.255.0
IP-MIB::ipAdEntNetMask.192.168.1.99.1 = IpAddress: 255.255.255.0
IP-MIB::ipAdEntBcastAddr.10.1.1.1.1 = INTEGER: 1
IP-MIB::ipAdEntBcastAddr.10.1.1.1.2 = INTEGER: 1
IP-MIB::ipAdEntBcastAddr.10.255.1.1.1 = INTEGER: 1
IP-MIB::ipAdEntBcastAddr.172.16.200.1.2 = INTEGER: 1
IP-MIB::ipAdEntBcastAddr.192.168.1.99.1 = INTEGER: 1
IP-MIB::ipAdEntReasmMaxSize.10.1.1.1.1 = INTEGER: 65535
IP-MIB::ipAdEntReasmMaxSize.10.1.1.1.2 = INTEGER: 65535
IP-MIB::ipAdEntReasmMaxSize.10.255.1.1.1 = INTEGER: 65535
IP-MIB::ipAdEntReasmMaxSize.172.16.200.1.2 = INTEGER: 65535
IP-MIB::ipAdEntReasmMaxSize.192.168.1.99.1 = INTEGER: 65535
```

Example 4: Ethernet Statistics Group

FortiOS supports the Ethernet Statistics Group for Remote Network Monitoring (RMON), which provides detailed statistics about the traffic that passes through the Ethernet interface, such as drop events and collisions. This enhancement enables comprehensive monitoring and management of network performance, helping to ensure optimal operation and quickly identify any potential issues.

The Ethernet Statistics Group consists of the etherStatsTable .1.3.6.1.2.1.16.1.1 and can be leveraged using the following:

```
config system snmp rmon-stat
    edit 1
        set source <data source of the Ethernet statistics entry>
```

```
        set owner <owner of the Ethernet statistics entry>
    next
end
```

To enable entries to the Ethernet Statistics Group:

1. Configure the RMON group on the FortiGate:

```
config system snmp rmon-stat
    edit 1
        set source "port2"
        set owner "test"
    next
end
```

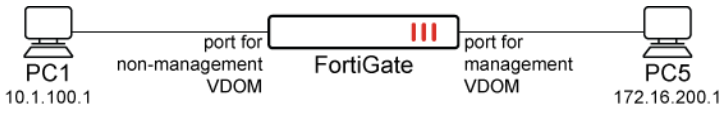
   An etherStatsEntry is created in the rmon-statistics table.

2. Query the etherStatsTable .1.3.6.1.2.1.16.1.1 on the SNMP server:

```
# snmpwalk -v2c -c REGR-SYS 172.16.200.1 etherStatsTable
RMON-MIB::etherStatsIndex.1 = INTEGER: 1
RMON-MIB::etherStatsDataSource.1 = OID: IF-MIB::ifIndex.8
RMON-MIB::etherStatsDropEvents.1 = Counter32: 0
RMON-MIB::etherStatsOctets.1 = Counter32: 39388877 Octets
RMON-MIB::etherStatsPkts.1 = Counter32: 125770 Packets
RMON-MIB::etherStatsBroadcastPkts.1 = Counter32: 0 Packets
RMON-MIB::etherStatsMulticastPkts.1 = Counter32: 84212 Packets
RMON-MIB::etherStatsCRCAlignErrors.1 = Counter32: 0 Packets
RMON-MIB::etherStatsUndersizePkts.1 = Counter32: 0 Packets
RMON-MIB::etherStatsOversizePkts.1 = Counter32: 0 Packets
RMON-MIB::etherStatsFragments.1 = Counter32: 0 Packets
RMON-MIB::etherStatsJabbers.1 = Counter32: 0 Packets
RMON-MIB::etherStatsCollisions.1 = Counter32: 0 Collisions
RMON-MIB::etherStatsPkts64Octets.1 = Counter32: 0 Packets
RMON-MIB::etherStatsPkts65to127Octets.1 = Counter32: 0 Packets
RMON-MIB::etherStatsPkts128to255Octets.1 = Counter32: 0 Packets
RMON-MIB::etherStatsPkts256to511Octets.1 = Counter32: 0 Packets
RMON-MIB::etherStatsPkts512to1023Octets.1 = Counter32: 0 Packets
RMON-MIB::etherStatsPkts1024to1518Octets.1 = Counter32: 0 Packets
RMON-MIB::etherStatsOwner.1 = STRING: "test"
RMON-MIB::etherStatsStatus.1 = INTEGER: valid(1)
```

Example 5: Enabling non-management VDOMs to send queries using SNMP v3

Although non-management and management VDOMs can perform queries using SNMP v3, this example shows how to enable non-management VDOMs to send queries.



PC1 connects to the port on FortiGate for the non-management VDOM, and SNMP v3 queries from non-management VDOMs are enabled. PC5 connects to the port on FortiGate for the management VDOM. With this configuration, SNMP queries are performed by both the non-management and the management VDOMs

To enable non-management VDOM SNMPv3 queries:

1. On FortiGate, enable non-management VDOM queries:

```
config system snmp sysinfo
    set status enable
    set engine-id-type text
    set engine-id ''
    set description ''
    set contact-info ''
    set location ''
    set trap-high-cpu-threshold 80
    set trap-low-memory-threshold 80
    set trap-log-full-threshold 90
    set trap-free-memory-threshold 5
    set trap-freeable-memory-threshold 60
    set append-index disable
    set non-mgmt-vdom-query enable
end
```

2. Check the SNMP information:

   This example uses the SNMP walk application to confirm that both management and non-management VDOMs are performing SNMP queries to PC1 and PC5.

```
pc01:~$ snmpwalk -v3 -u v3user 10.1.100.1 1.3.6.1.4.1.12356.101.5.1.2.1.1.1
FORTINET-FORTIGATE-MIB::fgFwPolID.1.0 = INTEGER: 0
FORTINET-FORTIGATE-MIB::fgFwPolID.1.1 = INTEGER: 1
FORTINET-FORTIGATE-MIB::fgFwPolID.2.0 = INTEGER: 0

pc05~$ snmpwalk -v3 -u v3user 172.16.200.1 1.3.6.1.4.1.12356.101.5.1.2.1.1.1
FORTINET-FORTIGATE-MIB::fgFwPolID.1.0 = INTEGER: 0
FORTINET-FORTIGATE-MIB::fgFwPolID.1.1 = INTEGER: 1
FORTINET-FORTIGATE-MIB::fgFwPolID.2.0 = INTEGER: 0
```

Replacement messages

FortiOS has replacement messages that are HTML and text files. These messages can be customized to meet user requirements. The content can be modified, and images can be added.

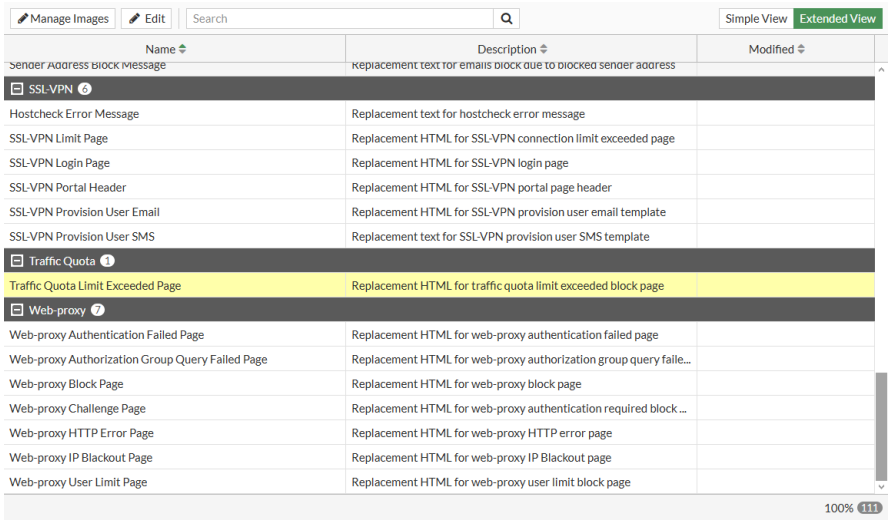
Modifying replacement messages

The *Replacement Messages* page has two views. *Simple View* (the default view) shows the most commonly used replacement messages. *Extended View* shows the entire list and all replacement message categories.

To modify a replacement message in the GUI:

1. Go to *System* > *Replacement Messages*.
2. Select a replacement message and click *Edit*.

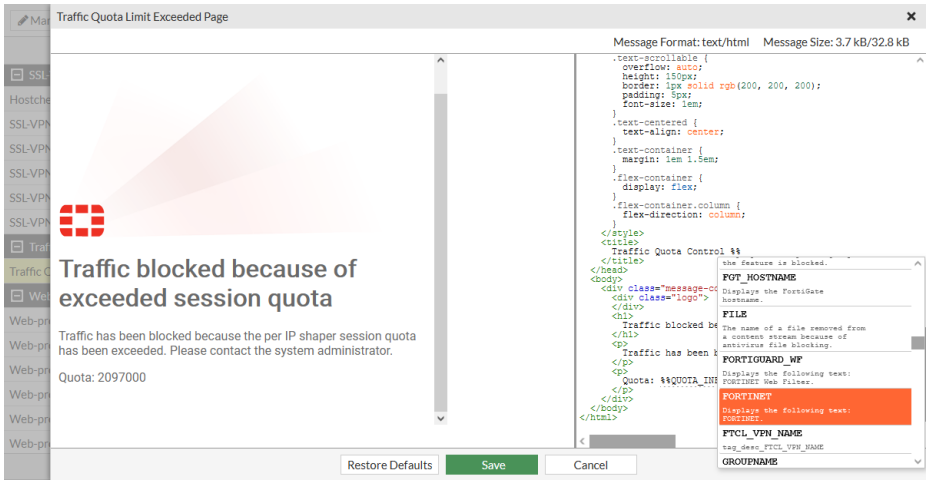
   If the message you want to edit is not visible, click *Extended View* in the upper right-hand corner of the top menu.



3. Edit the HTML code.

   The message is visible on the left alongside the HTML code on the right. The message view updates in real-time as you edit the content.

   When adding a variable to the code, right-click and select *Insert Tag* or type *%%* to view a list of the available variables, or start typing the variable name then press *Enter* or *TAB* to auto-complete the variable name.



4. Click *Save*.

Click *Restore Defaults* to return to the original message and code base.

To modify a replacement message in the CLI:

For example, to modify the *Traffic Quota Limit Exceeded Page* message:

```
config system replacemsg traffic-quota "per-ip-shaper-block"
    set buffer "<html>
  <head>
    <title>
      Traffic Quota Control
    </title>
  </head>
  <body>
    <font size=2>
      <table width=\"100%\">
        <tr>
          <td bgcolor=#3300cc align=\"center\" colspan=2>
            <font color=#ffffff>
              <b>
                Traffic blocked because exceeded session quota
              </b>
            </font>
          </td>
        </tr>
      </table>
      <br>
      <br>
      Traffic blocked because it exceeded the per IP shaper session quota. Please contact the
system administrator.
      <br>
      %%QUOTA_INFO%%
      <br>
      <br>
      <hr>
    </font>
  </body>
</html>"
    set header http
    set format html
end
```

Replacement message images

Images can be added to replacement messages on:

- Disclaimer pages
- Login pages
- Declined disclaimer pages

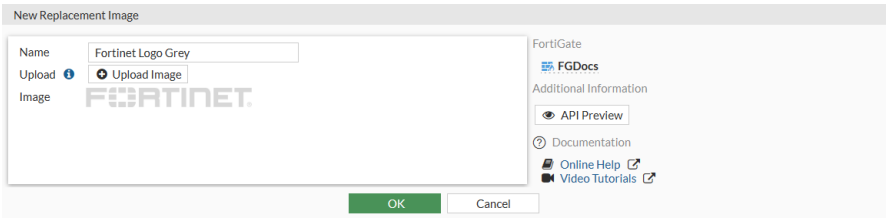
- Login failed pages
- Login challenge pages
- Keepalive pages

> The supported image formats are GIF, JPEG, TIFF, and PNG. The maximum file size supported is 24 KB.

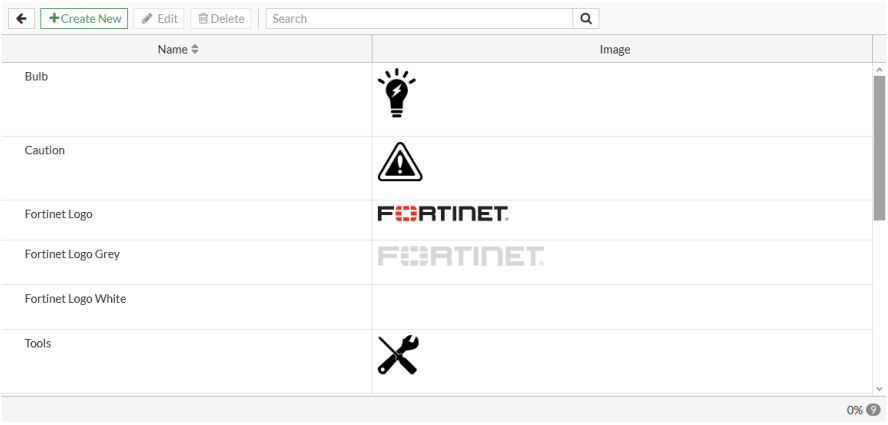
Adding images to replacement messages

To add images to replacement messages in the GUI:

1. Go to *System* > *Replacement Messages*.
2. In the top menu, click *Manage Images*.
3. Click *Create New*.
4. Enter a name for the image.
5. Click *Upload Image* and locate the file.



6. Click *OK*.
   The file is now visible in the list.



7. Return to the replacement message list and edit a message.
8. Right-click in the message code where you want to add the image, and select *Insert Image*.
9. Select the image from the list then press *Enter*, or double-click on the image to add it to the message.
10. Click *Save*.

To add images to replacement messages in the CLI:

1. Add the image to the FortiGate:

```
config system replacemsg-image
    edit <image_name>
        set image-type {gif | jpg | tiff | png}
        set image-base64 <string>
    next
end
```

2. Edit the replacement message, and include `%%IMAGE:<image name>%%` in the code to add the image.

Replacement message groups

Replacement message groups allow users to customize replacement messages for individual policies and profiles.

There are two types of replacement message groups:

Type	Usage	Customizable categories
utm	Used with UTM settings in firewall policies.	• admin • alertmail • custom-message • fortiguard-wf • ftp • http • icap • mail • nac-quar • spam • sslvpn • traffic-quota • utm • webproxy
auth	Used with authentication pages in firewall policies.	• auth • webproxy

The messages added to a group do not need to be customized. The message body content, header type, and format will use the default values if not customized.

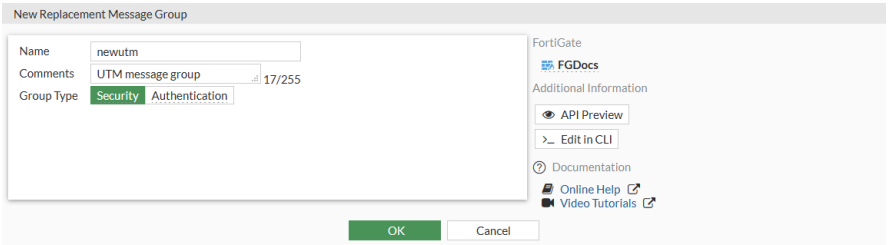
To make replacement message groups visible in the GUI:

```
config system global
    set gui-replacement-message-groups enable
end
```

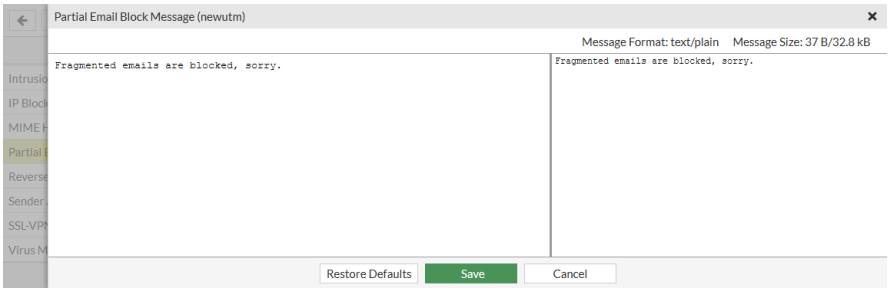
In the following example, two replacement message groups are created. The UTM message group includes custom mail-related messages and is assigned to an email filter profile. The authentication message group has a custom authentication success message that is applied to a proxy-based firewall policy that has an assigned email filter profile.

To create replacement message groups in the GUI:

1. Create the *Security* replacement message group:
   a. Go to *System* > *Replacement Message Groups*.
   b. Click *Create New*.
   c. For *Name*, enter *newutm*.
   d. In the *Comments* field, enter *UTM message group*.
   e. For *Group Type*, select *Security*.
   f. Click *OK*.

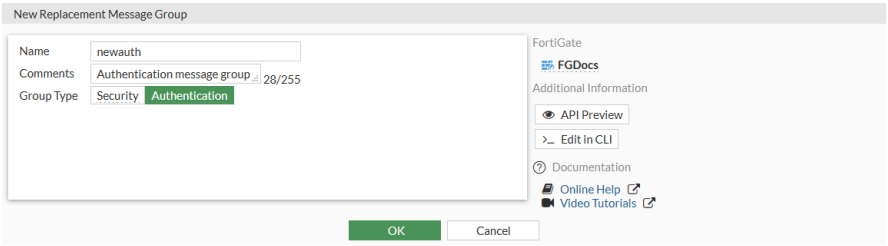


2. Customize the replacement messages in the *newutm* group:
   a. Go to *System* > *Replacement Message Groups*.
   b. Edit the *newutm* group.
   c. Select the *Partial Email Block Message*.



   d. Edit the message and click *Save*.
   e. Select the *ASE Block Message*.
   f. Edit the message and click *Save*.
3. Create the *Authentication* replacement message group:
   a. Go to *System* > *Replacement Message Groups*.
   b. Click *Create New*.
   c. For *Name*, enter *newauth*.
   d. In the *Comments* field, enter *Authentication message group*.
   e. For *Group Type*, select *Authentication*.

f. Click *OK*.



4. Apply the *newutm* replacement message group to an email filter profile in the CLI:

```
config emailfilter profile
    edit "newmsgs"
        set replacemsg-group "newutm"
    next
end
```

5. Apply the *newauth* replacement message group and the email filter profile to a firewall policy in the CLI:

```
config firewall policy
    edit 1
        ...
        set replacemsg-override-group "newauth"
        set inspection-mode proxy
        set emailfilter-profile "newmsgs"
        ...
    next
end
```

To create replacement message groups in the CLI:

1. Create the replacement message groups:

```
config system replacemsg-group
    edit "newutm"
        set comment "UTM message group"
        set group-type utm
        config mail
            edit "partial"
                set buffer "Fragmented emails are blocked, sorry."
            next
        end
        config spam
            edit "smtp-spam-ase"
                set buffer "This message has been blocked because ASE reports it as spam.
You\'re welcome."
            next
        end
    next
    edit "newauth"
        set comment 'Authentication message group'
        set group-type auth
```

```
        config auth
            edit "auth-success-msg"
                set buffer "Welcome to the firewall. Your authentication has been accepted,
please reconnect."
            next
        end
    next
end
```

2. Apply the message group to the email filter:

```
config emailfilter profile
    edit "newmsgs"
        set replacemsg-group "newutm"
    next
end
```

3. Apply the email filter and message group to the policy:

```
config firewall policy
    edit 1
        ...
        set replacemsg-override-group "newauth"
        set inspection-mode proxy
        set emailfilter-profile "newmsgs"
        ...
    next
end
```

FortiGuard

FortiGuard services are comprised of signature packages and querying services that provide content, web security, and device security. The services are delivered through various types of FortiGuard servers that are part of the FortiGuard Distribution Network (FDN).

FortiGuard service subscriptions can be purchased and registered to your FortiGate unit. The FortiGate must be connected to the Internet in order to automatically connect to the FDN to validate the license and download FDN updates or perform real-time queries.

FortiGuard options are distributed across the following tabs:

Tab	Description
Subscriptions	View FDN support contract information and expiration dates, expired licenses, and licenses available for purchase. Upload a license file. See Licensing in air-gap environments on page 3916.
FortiGuard settings	Configure update schedules, filtering, and FortiGuard server preferences.

FortiGuard servers use Anycast. See Anycast on page 3949 and Anycast and unicast services on page 3951. See also Online security tools on page 3952.

Subscriptions

Go to *System > FortiGuard > Subscriptions* to view the list of subscriptions, service entitlements, and license statuses. Information is organized across the following tabs:

- Licensed on page 3907
- Expired on page 3910
- Available for purchase on page 3910

Licensed

The *System > FortiGuard > Subscriptions > Licensed* tab displays all the subscriptions, service entitlements, and licenses. Expand each subscription or entitlement to view its details. Subscriptions and licenses are organized into the following sections:

- Basic subscriptions on page 3907
- FortiGuard Security Services on page 3908
- SD-WAN and SASE Services on page 3909
- NOC & SOC on page 3910

On the *Dashboard > Status* page, the *Licenses* widget also lists the status of major entitlements. Licensed entitlement icons are green, and unlicensed entitlement icons are orange. See Status dashboard on page 143.

Basic subscriptions

Subscription or service	FortiGuard service description
Support FortiCare Essential FortiCare Premium FortiCare Elite	
Firmware, VM, and base subscriptions Application Control Signatures Inline-CASB application definitions Device & OS identification definitions GeoIP definitions Trusted CA certificate definitions[1]	The FortiCare support entitlement includes firmware and general updates that come with various default signatures and definitions, including: • Application control signatures used in application control profiles • Device & OS identification used for device detection and asset management • Virtual patch signatures used in local-in policies • Inline CASB application definitions used in inline CASB profiles • ISDB destinations that can be applied in various policies and rules • PSIRT vulnerability definitions used in security ratings

Subscription or service	FortiGuard service description
Internet service and botnet IP definitions Local protection signatures PSIRT check definitions Time zone definitions	**Note**: FortiGates with FortiCare Premium or FortiCare Elite have 3 free seats to FortiClient EMS Cloud and 3 free seats to FortiIdentity Cloud. See FortiCloud Subscriptions for details.
FortiConverter Hosted Service	FortiConverter service

1 The Trusted CA certificate bundle follows Mozilla's published trusted CA list for updating its certificate bundle. To view the built-in and bundled CA certificates supported by your FortiGate, execute the command `get vpn certificate ca | grep ==`.

FortiGuard Security Services

Entitlement	FortiGuard service description
Intrusion Prevention System (IPS) IPS signatures IPS Engine Malicious URL definitions	The IPS service includes engines, databases, and definitions used in the IPS and application control profiles. In order to download updated IPS definitions, at least 1 policy with a security profile that has IPS scanning must be enabled. See Intrusion prevention on page 2478 and Application control on page 2428 for details.
Advanced Malware Protection AI-based heuristic antivirus engine Antivirus definitions Antivirus engine Mobile malware signatures Botnet domain definitions Virus outbreak prevention with file checksum query	The Advanced Malware Protection service includes various engines, databases, and definitions used in the AV profile. In order to download updated AV definitions, at least 1 policy with a security profile that has Antivirus scanning must be enabled. See Antivirus on page 2240 for details.
File analysis with FortiGate Cloud Sandbox query AI-based Inline Malware Prevention Service	Provides Inline protection against unknown/0-day threats - holding a file for up to 50 seconds for the verdict to be returned and based on it, files can either be blocked or released. • AV profile • Send Files to FortiSandbox for Inspection • Scan strategy to Inline
AntiSpam query	Consults FortiGuard servers to help identify spammer IP address or emails, known phishing URLs, known spam URLs, known spam email checksums, and others.

Entitlement	FortiGuard service description
	• Email filter profile • FortiGuard Spam Filtering
URL, DNS & Video Filtering Malicious certificate definitions DNS Filtering Video Filtering	The Web Security service includes: • FortiGuard categories used in web filter profiles • Malicious certificates used in SSL/SSH inspection profiles • FortiGuard categories used in DNS filter profiles • FortiGuard categories used in video filter profiles
Attack Surface Security IoT device identification definitions Outbreak check definitions Security rating and CIS compliance definitions	The Attack Surface Security service includes: • Running all the built-in free and paid security rating rules • Displaying CIS compliance information within security ratings • IoT Detection and IoT Query
Operational Technology (OT) Security Service OT threat definitions OT device identification definitions OT vulnerability correlation definitions & virtual patching signatures	The OT Security service includes OT-related threat definitions used in IPS and application control profiles. It also includes OT Detection Definitions and Virtual Patching Signatures used in the virtual patching profile. See Virtual patching on page 2675 for details.
Data Loss Prevention (DLP) Signatures DLP signatures	The Data Loss Prevention service offers a database of predefined DLP patterns such as data types, dictionaries, and sensors that are used in the DLP profile.

SD-WAN and SASE Services

Entitlement	FortiGuard service description
SD-WAN Underlay Bandwidth and Quality Monitoring	SD-WAN Underlay Bandwidth and Quality Monitoring service
SD-WAN Overlay-as-a-Service	Delivers efficient setup and management of new SD-WAN regions via the easy-to-use FortiCloud portal.
SD-WAN Connector for FortiSASE Secure Private Access	Organizations with new or existing FortiGate SD-WAN deployments can provide their FortiSASE remote users with access to private resources.
SASE connector for FortiSASE Secure Edge Management	Extend security capabilities to all endpoints, users, and devices across the network.

NOC & SOC

Entitlement	FortiGuard service description
FortiGate Cloud	FortiGate Cloud management, analysis, and log retention services
FortiGate Cloud Log Retention	Store and access different logs, including traffic, system, web, applications, and security events securely in the cloud.
FortiAnalyzer Cloud	FortiAnalyzer Cloud service
FortiManager Cloud	FortiManager Cloud service
FortiClient EMS Cloud	FortiClient EMS Cloud service
FortiSandbox Cloud	FortiSandbox Cloud service
FortiToken Cloud	FortiIdentity Cloud service

Expired

The *System > FortiGuard > Subscriptions > Expired* tab lists expired licenses and the date each license expired.

Contact your local reseller for subscription options.

See also License expiration on page 3917.

Available for purchase

The *System > FortiGuard > Subscriptions > Available for purchase* tab lists subscriptions and services available for purchase.

Contact your local reseller for subscription options.

IoT detection service

Internet of Things (IoT) detection is part of the Attack Surface Security Rating service that allows FortiGate to:

- Download an IoT Detection signature package (IOTD), which is used to detect and extract metadata of IoT devices
- Query FortiGuard IoT Query service for devices that are not detected by the local Device Database (CIDB) or by the IoT Detection signatures
- Query the FortiGuard vulnerability lookup server to look up vulnerabilities for a device

The service allows the FortiGate to accurately detect and identify connected IoT devices and to identify vulnerabilities that apply to these devices.

Applications

A subscription to the IoT detection service provides many practical applications.

Device information	Enable the FortiGate to obtain updated device information, store the information in the asset inventory, and display the information in various places, such as the *Dashboard > Asset and Identities* widget.
Vulnerability information	Enable the FortiGate to query and display vulnerabilities associated with a device on the *Asset and Identities* widget.
Allow managed FortiSwitch and FortiAP devices to query device info	Device detection is effective when devices are connected on an interface. As such, when devices are connected to managed FortiSwitch and FortiAP devices, they can utilize IoT detection to gather information about the connected devices. See FortiAP query to FortiGuard IoT service to determine device details on page 3915 for more information. See also FortiSwitch Devices Managed by FortiOS guide.
Perform NAC on IoT devices with vulnerabilities	NAC policies can be configured to detect devices with different levels of vulnerabilities and assign these devices dynamically to a quarantine VLAN. See OT and IoT virtual patching on NAC policies on page 2687.
Perform IoT Vulnerability rating check in Security Rating	Using the *Security Fabric > Security Rating* feature, perform a FortiGuard IoT Vulnerability check to identify devices with detected vulnerabilities.
Detect and log IoT devices in Application Control	IoT signatures are used in application control to detect and log IoT devices. In *Security Profiles > Application Signatures*, filter on the IoT category to see the list of IoT application signatures. Alternatively, view the signatures using the following command: `# get application name status | grep IoT -B2 -A10`

Device detection

When device detection is enabled on an interface, FortiGate will perform passive scanning on incoming traffic to collect information about devices such as their MAC address, IP address, Operating System, Hostname, username, endpoint tags and interface in which it entered . The firewall is able to scan different protocols to obtain basic device information. FortiGates by default comes with a built-in local Device Database (CIDB) containing information about known devices which can be used to obtain more detailed information.

When the CIDB cannot identify the device, FortiGate can utilize the IoT Query service by sending some device information to the FortiGuard collection server. If a new device is detected, FortiGate obtains the results from the FortiGuard query for more information about the device.

Device detection is intended for devices directly connected to your LAN and DMZ ports. Since the IoT detection service relies on device detection, the device must be on the same Layer 2 network as the FortiGate.

Configurations

This feature requires an Attack Surface Security Rating service license.

FortiGate device requirements:

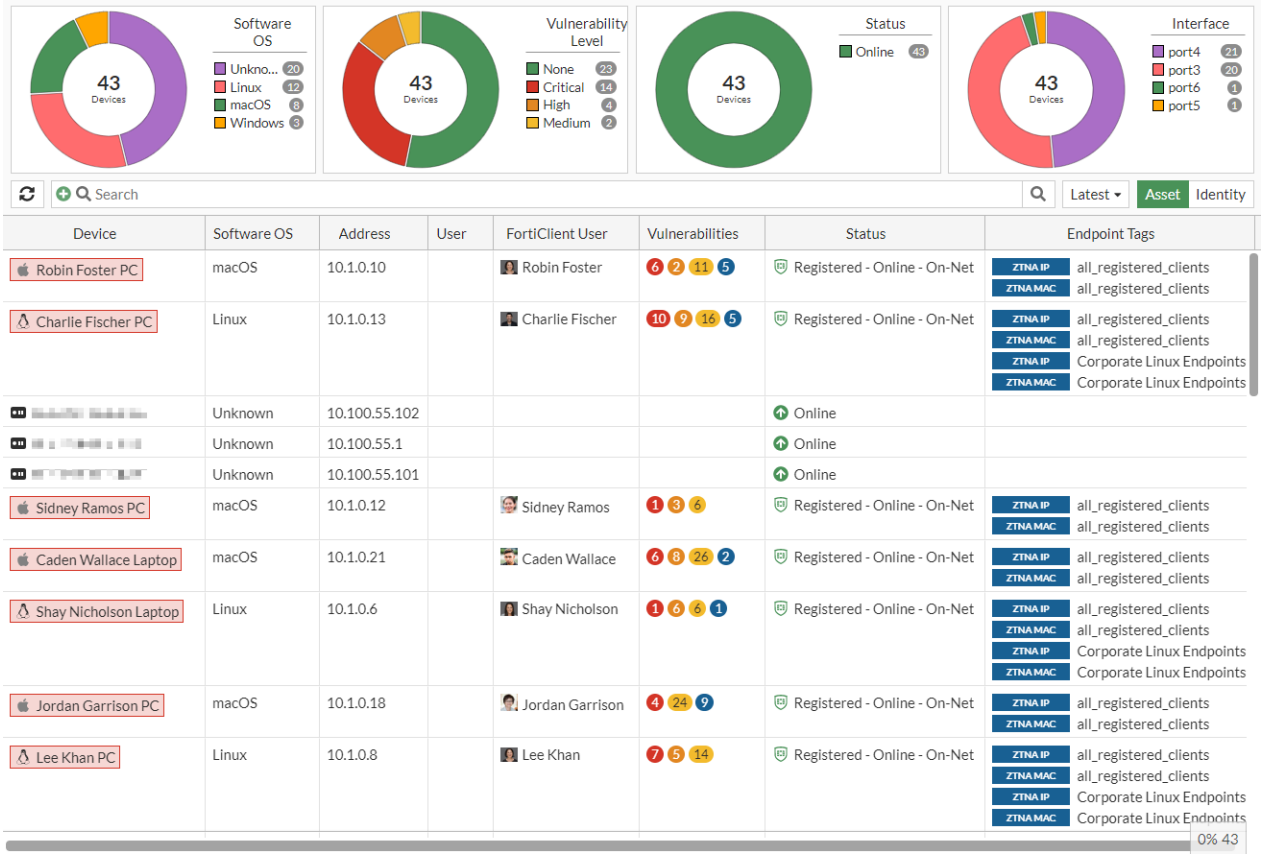
The FortiGate device must be:

- Registered with FortiCare
- Connected to an anycast FortiGuard server

How the service works:

1. Enable Device Detection on an interface.
2. FortiGate uses the interface to detect device traffic flow.
3. Upon detecting traffic from an unknown device, FortiGate sends the device data to the FortiGuard collection server.
4. The collection server returns data about the new device to the FortiGuard query server.
5. If the device signature does not appear in the local Device Database (CIDB) or some fields are not complete, FortiGate queries FortiGuard for more information about the device.

To view the latest device information in the GUI, go to *Dashboard* > *Assets & Identities* and expand the *Assets* widget. For more information, see Asset Identity Center page on page 4400.



To debug the IoT daemon in the CLI:

1. Enable iotd real-time debugs to collect information about the communication between FortiGate and the FortiGuard server:

```
# diagnose debug application iotd -1
# diagnose debug enable
```

2. Optionally, disable the local device database to force all queries to go to FortiGuard.

```
# diagnose cid sigs disable
```

3. View the debug output.
   FortiGate sends the device data to the FortiGuard collection server:

```
# [iotd] recv request from caller size:61
[iotd] service:collect hostname: ip: fd:-1 request tlv_len:41
[iotd] txt(.....y...w.....Jasons-iPhone6....579=23..)
[iotd] hex(020100070179030 60f77fc0203000e4a61736f6e732d6950686f6e653602040008353739 3d32330cff)
[iotd] service:collect hostname:globaldevcollect.fortinet.net ip: fd:-1 got server hostname
[iotd] service:collect hostname:globaldevcollect.fortinet.net ip:173.243.138.29 fd:-1 got
server ip
[iotd] service:collect hostname:globaldevcollect.fortinet.net ip:173.243.138.29 fd:13 socket
created
[iotd] service:collect hostname:globaldevcollect.fortinet.net ip:173.243.138.29 fd:13
connecting
[iotd] fd:13 monitor event:pollout
[iotd] service:collect hostname:globaldevcollect.fortinet.net ip:173.243.138.29 fd:13 build
req packet
[iotd] service:collect hostname:globaldevcollect.fortinet.net ip:173.243.138.29 fd:13 collect
resp:1(pending)
```

   The FortiGuard collection server returns new device data to the FortiGuard query server:

```
[iotd] service:query hostname:globaldevquery.fortinet.net ip:173.243.140.16 fd:17 got query
resp
[iotd] service:query hostname:globaldevquery.fortinet.net ip:173.243.140.16 fd:17 id:0 total_
len:48 header_len:16 tlv_len:32 confidence:100 mac:f8:87:f1:1f:ab:95
[iotd] service:query hostname:globaldevquery.fortinet.net ip:173.243.140.16 fd:17 remaining_
len:32 type:1 len:6
[iotd] service:query hostname:globaldevquery.fortinet.net ip:173.243.140.16 fd:17 got tlv
category:'Mobile'
[iotd] service:query hostname:globaldevquery.fortinet.net ip:173.243.140.16 fd:17 remaining_
len:24 type:2 len:6
[iotd] service:query hostname:globaldevquery.fortinet.net ip:173.243.140.16 fd:17 got tlv sub_
category:'Mobile'
[iotd] service:query hostname:globaldevquery.fortinet.net ip:173.243.140.16 fd:17 remaining_
len:16 type:3 len:5
[iotd] service:query hostname:globaldevquery.fortinet.net ip:173.243.140.16 fd:17 got tlv
vendor:'Apple'
[iotd] service:query hostname:globaldevquery.fortinet.net ip:173.243.140.16 fd:17 remaining_
len:9 type:4 len:0
[iotd] service:query hostname:globaldevquery.fortinet.net ip:173.243.140.16 fd:17 remaining_
len:7 type:5 len:3
[iotd] service:query hostname:globaldevquery.fortinet.net ip:173.243.140.16 fd:17 got tlv
os:'iOS'
[iotd] service:query hostname:globaldevquery.fortinet.net ip:173.243.140.16 fd:17 remaining_
len:2 type:6 len:0
[iotd] service:query hostname:globaldevquery.fortinet.net ip:173.243.140.16 fd:17 send query
response to caller size:48
[iotd] txt(............d0 ...Mobile..Mobile..Apple....iOS..)
[iotd] hex
(f887f11fab950000000000006430200001064d6f62696c6502064d6f62696c6503054170706c6504000503694f530
600)
```

```
[iotd] service:query hostname:globaldevquery.fortinet.net ip:173.243.140.16 fd:17 read resp:0
(good)
```

4. Upon completion of the FortiGuard query, the query server returns the device information including the information source (`src fortiguard`).

```
# diagnose user device list
    vd root/0  f8:87:f1:1f:ab:95  gen 26  req OUA/34
        created 503s  gen 23  seen 102s  lan  gen 7
        ip 192.168.1.110  src arp
        hardware vendor 'Apple'  src fortiguard  id 0  weight 100
        type 'Mobile'  src fortiguard  id 0  weight 100
        family 'Mobile'  src fortiguard  id 0  weight 100
        os 'iOS'  src fortiguard  id 0  weight 100
        host 'Jasons-iPhone6'  src dhcp
```

Using FortiManager as an override server for IoT query services

FortiGate can use FortiManager as an override server for IoT query services. The FortiManager must be running 7.2.1 or later.

All IoT daemon query and collected data can be sent to a FortiManager, instead of directly to FortiGuard. This is useful when there are strict policies controlling the kind of traffic that can go to the internet.

To send all IoT daemon query and collected data to a FortiManager in the GUI:

1. Go to *System* > *FortiGuard* > *FortiGuard settings*.
2. Under *FortiGuard servers*, set *Server preference* to one of the following:
   - Prioritize public FortiGuard servers. Fall back to custom servers if needed.
   - Prioritize custom servers. Fall back to public FortiGuard servers if needed.
3. Add a custom server:
   a. Under *Custom servers*, click *Create new*.
   b. Set *Address Type* to *IPv4*.
   c. In *Address*, enter the IPv4 address for FortiManager.
   d. Click the *Type* box, select the following server types, and click *Close*.
      - *IoT detection*
      - *Virtual patching query*
   e. Click *OK*. The custom server is displayed.
4. Click *Apply*.

To send all IoT daemon query and collected data to a FortiManager in the CLI:

```
config system central-management
    config server-list
        edit 1
            set server-type vpatch-query iot-collect
            set server-address <x.x.x.x>
        next
```

```
    end
end
```

Option	Description
`server-type vpatch-query iot-collect`	Set the FortiGuard service types: • `vpatch-query`: Virtual patch query server. • `iot-collect`: IoT device collection server.
`server-address <x.x.x.x>`	Enter the IPv4 address of the FortiManager.

When vpatch-query is not configured, the following debug command will show productapi.fortinet.com as the destination server:

```
# diagnose test application ipsmonitor 45
2025-06-29 14:37:25 cloud-communication: enabled
server: productapi.fortinet.com:443
```

When vpatch-query is configured, the override server address is displayed:

```
# diagnose test application ipsmonitor 45
2025-06-29 14:39:01 cloud-communication: enabled
server: 192.168.77.121:443
```

FortiAP query to FortiGuard IoT service to determine device details

A FortiAP collects packets from devices and queries FortiGuard with the help of the FortiGate. Device detection results are reported back to the FortiGate where this information is displayed. Querying the FortiGuard service requires an Attack Surface Security Rating service license.

The following attributes can be configured in `wireless-controller setting`:

Attribute	Description
device-weight <integer>	Set the device upper limit of confidence (0 - 255, default = 1, 0 = disable).
device-holdoff <integer>	Set the device lower limit of creation time, in minutes (0 - 60, default = 5).
device-idle <integer>	Set the device upper limit of idle time, in minutes (0 - 14400, default = 1440).

To query the FortiGuard IoT service:

```
config wireless-controller setting
    ...
    set device-weight 1
    set device-holdoff 5
    set device-idle 1440
    ...
end
```

```
# diagnose user device list
  vd root/0  54:27:1e:e6:26:3d  gen 89  req OUA/34
    created 70s  gen 86  seen 2s  port29  gen 28
    ip 10.29.1.214  src mac
    hardware vendor 'Asustek compute'  src fortiguard  id 0  weight 21
    type 'Home & Office'  src fortiguard  id 0  weight 21
    family 'Computer'  src fortiguard  id 0  weight 21
    os 'Linux'  src dhcp  id 822  weight 128
    host 'test-wifi'  src dhcp
```

Licensing in air-gap environments

In the Operational Technology industry, industrial equipment is critical and must not be connected to the internet. However, the equipment is still required to be protected by a firewall in this air-gap environment. Without a gateway to FortiGuard in air-gap environments, FortiGuard packages, such as AntiVirus and IPS, must be manually uploaded to the FortiGate. FortiGate licenses can be downloaded from FortiCloud and uploaded manually to the FortiGate.

Manual licensing for air-gap environments is supported on FortiGate hardware appliances and FortiGate virtual machine (VM) appliances running FortiOS 7.2.0 or later. When running on a Virtual appliance, the VM licensing still needs to connect to a licensing FortiManager or FortiGuard server. See VM license on page 4904 for details

To manually upload FortiGate licenses in the GUI:

1. Register the FortiGuard license on FortiCloud. See Registration for more information.
2. Download the product entitlement file in FortiCloud:
   a. Go to *Products* > *Product List*.
   b. Select the serial number of the FortiGate. The product page opens.
   c. In the *License & Key* section, click *Get The License File*. The file downloads to your device in the format `FG201E*********ProductEntitlement.lic`.
3. In FortiOS, go to the *System* > *FortiGuard* > *Subscriptions* tab.
4. From the *Update licenses & definitions now* dropdown, select *Upload License File*. The file explorer opens.
5. Navigate to the product entitlement file and click *Open*.
   The license file uploads to the FortiGate. This operation does not require reboot. Once the upload is complete, the FortiGate shows that it is registered and licensed.

To manually upgrade the AntiVirus Database in the GUI:

1. Download the static upgrade file from FortiCloud:
   a. Go to support.fortinet.com.
   b. Go to *Download* > *Download FortiGuard Service Updates* > *FortiGate*.
   c. Select the FortiOS version from the *OS Version* dropdown.
   d. Select the file from the appropriate FortiGate product model section. The file downloads to your device.

2. In FortiOS, go to the *System* > *FortiGuard* > *Subscriptions* tab and expand the *Advanced Malware Protection* widget to view the current licenses.
3. Click the upgrade button. The *Anti-Virus Database Upgrade* pane opens.
4. Click *Upload File*. The file explorer opens.
5. Navigate to the static upgrade file and click *Open*.
6. Click *OK*.

To manually upload FortiGate licenses in the CLI:

```
# execute restore manual-license {ftp | tftp} <license file name> <server> [args]
```

License expiration

The FortiGate will still function as a firewall if any or all of the FortiGuard licenses are expired. Valid FortiGuard licenses are required to receive database and signature updates, and to perform real-time or near-real-time security lookups to detect and quickly adjust your security posture for newly discovered attacks.

FortiGuard services are designed to be continuous. Any lapses in the service will require coverage back to the contract expiration date. For more information, see FortiCare/FortiGuard Renewal Continuous Service Policy.

License type	Expiration impact
Firmware & General Updates	Application Control Signatures, Device & OS Identification, FortiGate Virtual Patch Signatures, Inline-CASB Application Definitions, Internet Service Database Definitions, and PSIRT Package Definitions continue to work, but the databases are not updated and no new signatures are added. For example, if application control is used in a firewall policy that has an internet service applied to the source or destination addresses, then the policy will continue to inspect matching traffic using the FortiGate's existing application control signatures and ISDB definitions. Application Control Signatures, Device & OS Identification, FortiGate Virtual Patch Signatures, Inline-CASB Application Definitions, Internet Service Database Definitions, and PSIRT Package Definitions are included in the base services that are included with all FortiCare support contracts See FortiGuard Security Services for details. The FortiGate will automatically schedule an upgrade to the latest patch of the current minor version. The upgrade can be postponed by one week but cannot be cancelled. See Required firmware upgrades for FortiGates with invalid support contracts or that have reached EOES on page 3762.
Intrusion Prevention	IPS scanning continues to work, but the IPS databases are not updated and no new signatures are added. For example, if an IPS sensor with *Block malicious URLs* enabled is used in a firewall policy, then the policy will continue to inspect matching traffic using the FortiGate's existing IPS signatures and malicious URLs database.

License type	Expiration impact
	An active IPS license is critical for stopping sophisticated and zero-day attacks, as FortiGuard IPS provides near-real-time intelligence with thousands of intrusion prevention rules to detect and block known and zero-day threats. For more information, see Intrusion prevention on page 2478.
Botnet IPs/Domains	IPS sensors and DNS Filter profiles with Botnet C&C configured continue to work, but the Botnet IPs and Botnet Domain databases are not updated and no new signatures are added. While Botnet IPs and Domain are listed in the Intrusion Prevention category, they are actually part of the Firmware & General Updates contract. For more information, see Botnet C&C domain blocking on page 2396 and IPS with botnet C&C IP blocking on page 2502.
AntiVirus	Antivirus scanning continues to work, but the antivirus database is not updated and no new signatures are added. For more information, see Antivirus on page 2240.
Web and DNS Filtering	Category-based Web and DNS filtering stops working, as URLs and domains are sent to FortiGuard in real-time to determine the category. By default, all web and DNS traffic is dropped. If allowing website or DNS requests when a rating error occurs is enabled, then all web and DNS traffic passes through without filtering. If static URL or domain filtering is applied in a filter profile, those filters continue to work. Configurations where only specific URLs and domains are allowed and all others are blocked continue to work, but this is not a scalable solution blocking websites or performing category filtering. For more information, see FortiGuard filter on page 2315 and FortiGuard category-based DNS domain filtering on page 2393.
Email Filtering	Spam filtering stops working, as it queries the FortiGuard spam filtering server in real-time to check spammer IP addresses and emails (except those that are locally configured), phishing URLs, spam URLs, spam email checksums, and spam submissions. Anti-spam signatures are not updated. Profile options based on local spam filtering continue to work. For more information, see Email filter on page 2531.
Outbreak Prevention	Outbreak prevention stops working, as it uses real-time lookups to the FortiGuard Global Threat Intelligence database. For more information, see FortiGuard outbreak prevention on page 2297.
Security Rating & CIS Compliance	Paid security rating checks stop working. CIS security control mappings are also disabled.

License type	Expiration impact
	The Security Rating & CIS Compliance component of the Attack Surface Security Rating entitlement is required to run paid security rating checks across all of the devices in the Security Fabric. They allow rating scores to be submitted to and received from FortiGuard for network ranking. Without the Security Rating entitlement, only built-in security rating rules can be run. PSIRT-related vulnerability rules depend on the Firmware license support. For more information, see Security rating on page 4470.
Operational Technology (OT) Threat Definitions	OT Security Services signatures continue to work, but the database attack definitions are not updated and no new signatures are added. OT Security Services include application control and IPS signatures for OT applications and protocols. For example, if an IPS sensor enabled with OT Security Service signatures is used in a firewall policy, then the policy will continue to inspect matching traffic using the FortiGate's existing OT threat definition IPS signatures. For more information, see OT threat definitions on page 2490.

FortiGate GovRamp support

A GovRamp FortiGate SKU entitles the FortiGate to use dedicated FortiGuard servers located in the United States. It also entitles customers to access their support tickets through a dedicated FortiCare service located in the United States.

When you purchase a GovRamp FortiGate, you will receive a FortiGate that automatically boots up in GovRamp mode. It will contact the dedicated FortiGuard server to learn the rest of its entitlement.

All FortiGuard services that are supported by the GovRamp device are United States-based and use a specific FQDN. The FortiGuard servers only support connections through Anycast. Any unused cloud services are disabled on the FortiGate.

Supported FortiGuard services

The following table lists supported FortiGuard services:

Feature or service	FQDN	IP address
FortiGate firmware upgrade Contract / License Update	update.fortinetgov.com	23.249.62.6
FortiGuard Query	guardservice.fortinetgov.com	23.249.62.16
Video Query	videoquery.fortinetgov.com	23.249.62.18
SDNS	sdns.fortinetgov.com	23.249.62.53
Geo IP address Database	gip.fortinetgov.com	23.249.62.16
Device Query	devquery.fortinetgov.com	23.249.62.16

Feature or service	FQDN	IP address
Default DNS server		23.249.63.52 / 23.249.63.53
Default NTP server	ntp1.fortinetgov.com ntp2.fortinetgov.com	23.249.63.60/23.249.63.61 23.249.63.62 23.249.63.63

Unsupported FortiGuard services

The following lists the unsupported FortiGuard services:

- FortiCare server connection
- Central management to FortiManager or FortiGuard
- Logging to FortiAnalyzer
- FortiSandbox (FSA) and FSA Cloud configuration
- FortiGuard DDNS service
- FortiSwitch authorization
- FortiExtender pre-authorization
- Local FortiClient EMS
- FortiClient EMS cloud
- Product API: Device vulnerability on GUI device assets
- Security fabric CSF: Configured as root
- Security fabric CSF: Configured as leaf
- Alert email - User must configure their own email server
- FortiNDR
- Email Filter query to RBL_SERVER (dnsbl.sorbs.net)
- FortiToken server connection
- Logging to FortiGate Cloud server
- SD-WAN overlay
- Activating FortiGate Cloud account
- Regular FortiGuard DNS setting
- FortiAP pre-authorization
- Security rating under Security Fabric
- Attack Surface Security Rating

The following lists FortiGuard services that are subject to limitations:

- Security Rating, FortiSwitch, FortiAP, FortiClient, FortiExplorer, and FortiNAC related automation stitch, trigger, or action

Blocking unsupported features on GovRamp devices

When trying to enable services that are not supported on GovRamp devices, an error will be returned in the GUI and CLI. Likewise, some features are hidden in the GUI if they are disabled for GovRamp devices.

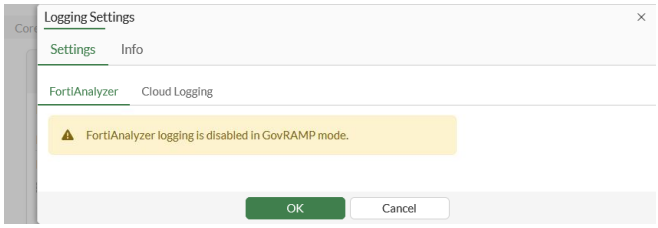
In the following example, the user attempts to enable FortiAnalyzer on a GovRamp FortiGate which is an unsupported service on GovRamp devices.

To view GovRamp device unsupported feature errors:

1. In the CLI, verify that the device has a GovRamp license:

```
# get system status
Version: FortiGate-1101E v7.6.4,build3596,250820 (GA.F)
First GA patch build date: 240724
Current Security Level: High
Firmware Signature: certified
...
License Status: GovRAMP
...
```

2. Test configuring the unsupported feature in the GUI:
   a. In the GUI, go to *Security Fabric* > *Fabric Connectors*.
   b. Edit *Logging & Analytics*.
   c. Attempt to enable FortiAnalyzer.



   An error is displayed and the *Switch Controller* feature is hidden.

3. Test configuring the unsupported feature in the CLI:
   a. Attempt to enable FortiAnalyzer.

```
config log fortianalyzer setting
    set status enable
        Cannot enable FortiAnalyzer logging when GovRAMP license is used.
        node_check_object fail! for status enable

        value parse error before 'enable'
        Command fail. Return code -39
```

   An error is displayed.

AMQP-powered subscription notifications for FortiGuard

Fortinet Inc. leverages AMQP (Advanced Message Queuing Protocol) to deliver real-time update notifications to FortiGate devices. When enabled, this allows FortiGate to receive notifications directly from FortiGuard, eliminating the need for polling or persistent HTTP connections. By leveraging Fortinet Inc.'s cloud infrastructure, AMQP enables event-driven updates, reducing resource consumption and minimizing overhead. Notifications are pushed instantly to devices, ensuring proactive management and swift response to critical updates.

The AMQP client daemon, `fortimq`, connects with the cloud server, `fortimq-cloud`. It works as a proxy for other FortiOS daemons to receive real-time updates for Fortinet Inc.'s cloud infrastructure. Once FortiGuard or an

account or device-level contract is updated, `fortimq-cloud` publishes notifications to the FortiGate and triggers the update procedure.

By default, `fortimq` stays idle until a feature explicitly subscribes to a topic, such as license alerts, database updates, and so on. When a subscription is created, `fortimq`:

1. Connects to the cloud.
2. Delivers updates automatically.
3. Disconnects once all subscriptions are removed.

The `fortimq` daemon requires the following pre-existing settings:

```
config system fortiguard
    set fortiguard-anycast enable
end
config system global
    set cloud-communication enable
end
```

CLI syntax

AMQP-powered subscription notifications for FortiGuard can be enabled and disabled in the CLI using the following command:

```
config system fortiguard
    set subscribe-update-notification {enable | disable}
end
```

This command automatically creates the following subscriptions:

- FortiGuard License Alerts
- Database Update Notifications

However, enabling `subscribe-update-notification` does not automatically disable persistent HTTP connections. If `persistent-connection` was manually enabled earlier, administrators should disable it after activating AMQP notifications to eliminate redundancy and reduce resource usage. Persistent HTTP connections can be disabled with the following command:

```
config system fortiguard
   set persistent-connection disable
end
```

Persistent HTTP connection is disabled by default on supported devices.

`fortimq` status can be tested using the following command:

```
diagnose test application fortimq <integer>
```

Example

The following example demonstrates enabling AMQP-powered subscription notifications and reviewing the logs.

To enable AMQP-powered subscription notifications for FortiGuard:

1. Enable AMQP-powered subscription notifications:

```
config system fortiguard
    set subscribe-update-notification enable
end
```

2. Monitor `fortimq` activities:

```
# diagnose debug application fortimq -1
```

After `fortimq` starts, it will remain idle:

```
<672> 02 fortimq_event()-211: handle event: restart (start)
<672> 08 fortimq_cleanup_unbound_payload_cb()-896
<672> 04 fortimq_client_try_start_cb()-1556
<672> 02 fortimq_event()-211: handle event: start-done (start)
<672> 02 fsm_update_state()-200: update state: idle (start)
```

It will leave the idle state when a feature explicitly subscribes to a topic:

- Once a new contract is set in the FortiGate, `fortimq` will receive the following message from FortiGuard:

```
<227> 08 fortimq_handle_basic_deliver()-1044: receive msg:
  delivery tag 1, channel 1 key FGD-LIC-UPDATE.TOKYO-APAC
  {"version":"1.0","type":"device_contract","geoloca
...
handle_fortimq_lic_notify_packet[328]-version=1.0, type=device_contract
handle_fortimq_lic_notify_packet[375]-contracts[0]=[{ "serial_number": "FG201E4QXXXXXXXX",
"contract": [ "AVDB-1-06-20260711:0:1:1:0", "COMP-1-20-20260711:0:1:1:0", "DLDB-1-06-
20260711:0:1:1:0", "ENHN-1-20-20260711:0:1:1:0", "FAIS-1-06-20260711:0:1:1:0", "FCSS-1-10-
20260711:0:1:1:0", "FGSA-1-06-20260711:0:1:1:0", "FMWR-1-06-20260711:0:1:1:0", "FRVS-1-06-
20260711:0:1:1:0", "FURL-1-06-20260711:0:1:1:0", "HDWR-1-05-20260711:0:1:1:0", "IOTH-1-06-
20260711:0:1:1:0", "ISSS-1-06-20260406:0:1:1:0", "NIDS-1-06-20260711:0:1:1:0", "SBCL-1-06-
20180716:0:1:1:0", "SPAM-1-06-20260711:0:1:1:0", "SPRT-1-20-20260711:0:1:1:0", "ZHVO-1-06-
20260711:0:1:1:0" ] }]

handle_fortimq_lic_notify_packet[404]-contract[0,12]=[ISSS-1-06-20260406:0:1:1:0]
```

- Once a new FortiGuard database is deployed, `fortimq` will receive the following message from FortiGuard:

```
3087> 08 fortimq_handle_basic_deliver()-1044: receive msg:
  delivery tag 1, channel 2 key
  {"version":"1.0","type":"package","geolocation":"T
```

```
handle_fortimq_obj_notify_packet[222]-version=1.0, type=package
handle_fortimq_obj_notify_packet[252]-version_string[0]=[07006000DBDB00100-00003.01214]
```

3. Review the `fortimq` status and bindings:

```
# diagnose test application fortimq 1
connection status: connected
    acct: 741008
    fqdn: qafortimq.fortinet.net
    port: 5671
    next-channel: 3
    msg count: 31
    attempts: 0

# diagnose test application fortimq 3
dump fortimq bindings:
    topic, queue, routing key, proc, pid, fd, cnt
    FGD-LIC_UPD 741008-FG201E4Q17901047 FGD-LIC-UPDATE.TOKYO-APAC updated 621 10 1
    FGD-DB-UPD FGD-DB-UPDATE-STREAM none updated 621 10 30
```

4. Review the results of the contract subscription:
   a. Review the push update for when the contract is triggered by the `fortimq` notification:

```
# diagnose debug application update -1
has_push_notification[690]-found notifcation for object=contract, version=00003.01214
cached_action_add[1284]-Cached action, act=00000002, add=1743727203, update=1743727203,
now=1743727207, adjust=-4.
upd_daemon[1636]-Received update notification from ForitGuard.

do_update[760]-Starting push UPDATE (not final retry)

update_status_obj[787]-ISDB contract expiry=Sun Apr  5 17:00:00 2026

upd_install_pkg[1410]-ALCI000(alci) installed successfully
```

   b. Review the FortiGuard contract update result:

```
# diagnose test update info
System contracts:
    ISDB,Sat Apr  4 2026
...
SerialNumber=FG201E4QXXXXXXXX|Contract=...*ISSS-1-06-20260406:0:1:1:0*
```

   c. Review the event log:

```
date=2025-04-03 time=17:40:27 eventtime=1743727227516567159 tz="-0700" logid="0100041000"
type="event" subtype="system" level="notice" vd="root" logdesc="FortiGate update
succeeded" status="update" msg="Fortigate notify update fcni=yes fsci=yes alci(0.00000)
from 192.168.100.76:443"
```

5. Review the results of the database subscription:

a. Review the push update for when the database is triggered by the `fortimq` notification:

```
# diagnose debug application update -1
has_push_notification[690]-found notifcation for object=dnsbot, version=00003.01214

upd_daemon[1636]-Received update notification from ForitGuard.

do_update[760]-Starting push UPDATE (not final retry)

upd_install_pkg[1410]-DBDB001(dnsbot) installed successfully
```

b. Review the FortiGuard database update result:

```
# diagnose autoupdate versions
Botnet Domain Database
---------
Version: 3.01214 signed
Contract Expiry Date: Thu Jul  9 2026
Last Updated using notify update on Thu Apr  3 16:09:22 2025
Last Update Attempt: Thu Apr  3 16:09:22 2025
Result: Updates Installed
```

c. Review the event log:

```
date=2025-04-03 time=16:09:22 eventtime=1743721762115544351 tz="-0700" logid="0100041000"
type="event" subtype="system" level="notice" vd="root" logdesc="FortiGate update
succeeded" status="update" msg="Fortigate notify update fcni=yes fsci=yes dnsbot(3.01214)
alci(0.00000) from 192.168.100.76:443"
```

d. Review the record of `fortimq` notification regarding the FortiGuard database:

```
# diagnose test update info
Update Notification: total 4, last received at Thu Apr  3 17:41:05 2025
        last notification:
                AVDB00201-00093.02170
                AVDB00701-00093.02170
                DBDB00100-00003.01214
                FSCI00100-00000.00000

Support contract: pending_registration=255 got_contract_info=1
    account_id=[xxxxx@fortinet.com] company=[Fortinet] industry=[Technology]

User ID: XXXXX
GeoLocation: TOKYO-APAC
```

FortiGuard settings

On the *System > FortiGuard > FortiGuard settings* tab, configure FortiGuard updates, filtering, and FortiGuard server preference. You can also view statistics about Fortinet service communications and connection status for FortiGuard filter rating servers.

FortiGuard updates	See for information: • Configuring FortiGuard updates on page 3926 • Using a proxy server to connect to the FortiGuard Distribution Network on page 3927 • Manual updates on page 3928 • Automatic updates on page 3929 • Scheduled updates on page 3930 • Sending malware statistics to FortiGuard on page 3931 • Update server location on page 3932
Filtering	See Filtering on page 3933.
FortiGuard servers	See for information: • FortiGuard server preference NEW on page 3934 • Using FortiManager as a local FortiGuard server on page 3939
Fortinet service communications	See for information: • Cloud service communication statistics on page 3942 • Disable all cloud communication on page 3946
FortiGuard Filter Rating Servers	See FortiGuard filter rating servers NEW on page 3947.

See also:

- FortiGate Cloud / FDN communication through an explicit proxy on page 3943
- FDS-only ISDB package in firmware images on page 3945

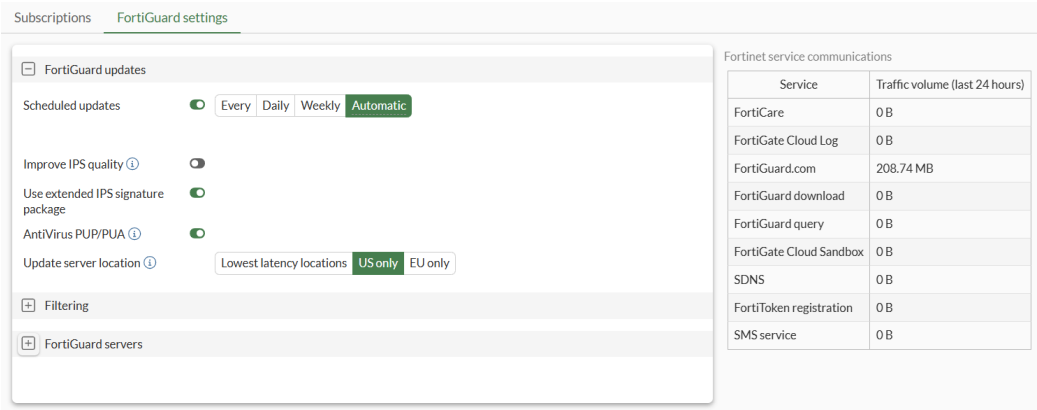
Configuring FortiGuard updates

To configure FortiGuard updates:

1. Go to the *System* > *FortiGuard* > *FortiGuard settings* tab.
2. In the *FortiGuard updates* section, configure the options for connecting and downloading definition files:

Immediately download updates	The option can be enabled on 2U and larger hardware models when the FortiGuard are servers are connected in anycast mode. The FortiGate forms a secure, persistent connection with FortiGuard to get notifications of new updates through an HTTPS connection. The FortiGate uses the fds_notify daemon to wait for the notification, then makes another connection to the FortiGuard server to download the updates.
Scheduled Updates	Enable to schedule updates to be sent to the FortiGate at the specified time or automatically. See Scheduled updates on page 3930 and Automatic updates on page 3929.
Improve IPS quality	Enable to send information to the FortiGuard servers when an attack occurs. This can help keep the FortiGuard database current as attacks evolve, and improve IPS signatures.

Use extended IPS signature package	Enable to use the extended IPS database, that includes protection from legacy attacks, along with the regular IPS database that protects against the latest common and in-the-wild attacks.
AntiVirus PUP/PUA	Enable antivirus grayware checks for potentially unwanted applications.
Update server location	The FortiGuard update server location. See Update server location on page 3932 for details.



3. Click *Apply*.

Using a proxy server to connect to the FortiGuard Distribution Network

You can configure FortiOS to use a proxy server to connect to the FortiGuard Distribution Network (FDN).

Proxy tunneling is supported only for registration, AV, and IPS updates. For FortiGate virtual machines, proxy tunneling can also be used for license validation. For web filtering or spam filtering, UDP protocol is used on ports 53 or 8888. UDP protocol traffic cannot be directed over a proxy server, even if you are using versions of FortiOS that support web filtering over port 443.

Consider the following before configuring FortiOS to use a proxy server to connect to FDN:

- FortiOS connects to the proxy server using the HTTP CONNECT method.
- The proxy server must not inspect the HTTPS traffic used for FortiOS communication.
- FortiOS sends to the proxy server an HTTP CONNECT request that specifies the IP address and port required for the FDN connection. Authentication information is optional for the request.
- FortiOS or the proxy server must be configured to use DNS servers that resolve the addresses of FDN servers to support AV and IPS updates.
- The proxy server establishes the connection to FDN and passes information between FortiOS and FDN.

Use the following syntax to configure a proxy server in the CLI:

```
config system fortiguard
    set proxy-server-ip <proxy_address>
    set proxy-server-port <proxy_port>
    set proxy-username <username>
```

```
    set proxy-password <password>
end
```

Alternatively, in a closed network without direct internet connection for web filtering or spam filtering, you can use FortiManager as a local FortiGuard server. FortiManager supports allowing FortiOS to retrieve its updates and ratings through FortiManager. See Using FortiManager as a local FortiGuard server on page 3939.

Example

In the following example, a proxy server with IP address 10.1.1.1 is configured to listen on port TCP/3128 without authentication.

To configure a proxy server:

```
config system fortiguard
    set proxy-server-ip 10.1.1.1
    set proxy-server-port 3128
end
```

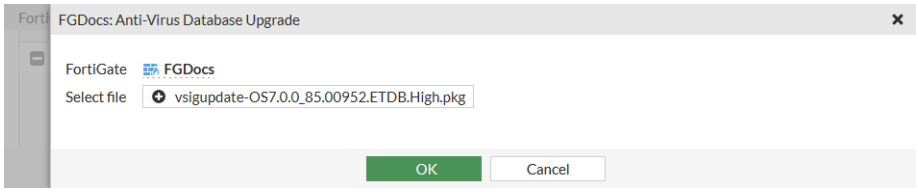
Manual updates

In order to download updated AV definitions, at least 1 policy with a security profile that has Antivirus scanning must be enabled. To download updated IPS definitions, at least 1 policy with a security profile that has IPS scanning must be enabled.

When needed, FortiGuard Distribution Network (FDN) updates can be manually uploaded.

To manually update the signature definitions files:

1. Log in to the Fortinet Support website.
2. Go to *Support* > *Service Updates*.
3. Select your *OS Version* from the dropdown list.
4. Locate your device in the table, and download the signature definitions files.
5. On the FortiGate, go to *System* > *FortiGuard*.
6. In the *License Information* table, locate and expand the definitions that you are updating.
7. From the *Actions* menu in the rightmost column, select *Upgrade Database*.
8. In the pane that opens, click *Upload*, locate the downloaded definitions file on your computer, then click *Open*.
   The download may take a few minutes to complete.



9. Click *OK*.

AV and IPS manual updates

To execute the update:

```
# execute restore ips tftp nids-720-19.261.pkg 172.16.200.55
```

To verify the manual AV and IPS package updates:

```
# diagnose debug app updated -1
# diagnose debug enable
```

Security levels are pre-configured on the BIOS. See BIOS-level signature and file integrity checking on page 4005 and Real-time file system integrity checking on page 4009 for more information.

Automatic updates

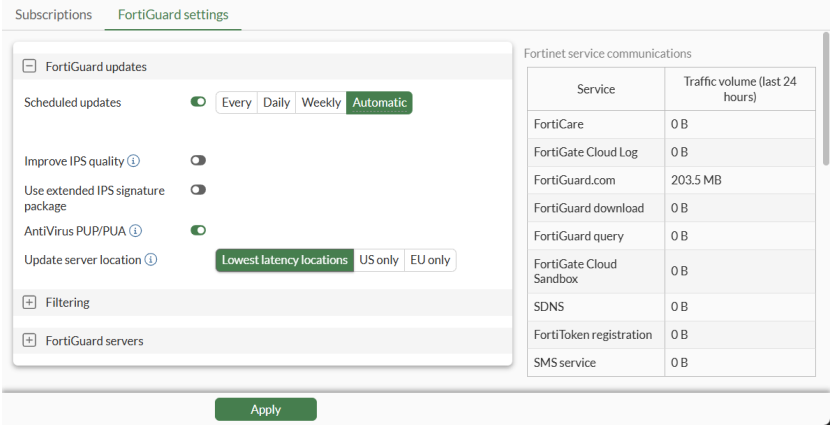
In order to download updated AV definitions, at least one policy with a security profile that has Antivirus scanning must be enabled. To download updated IPS definitions, at least one policy with a security profile that has IPS scanning must be enabled.

The default auto-update schedule for FortiGuard packages is daily within four hours of 1 AM.

When the schedule is set to automatic, the update interval is calculated based on the model and percentage of valid subscriptions, within one hour. For example, if a FortiGate 501E has 78% valid contracts, then based on this device model, the update schedule is calculated to be every 10 minutes. If you verify the system event logs (ID 0100041000), they are generated approximately every 10 minutes.

To configure automatic updates in the GUI:

1. Go to *System* > *FortiGuard* > *FortiGuard settings*
2. In the *FortiGuard Updates* section, enable *Scheduled Updates* and select *Automatic*.



3. Click *Apply*.

To configure automatic updates in the CLI:

```
config system autoupdate schedule
    set status enable
    set frequency automatic
end
```

Scheduled updates

In order to download updated AV definitions, at least 1 policy with a security profile that has Antivirus scanning must be enabled. To download updated IPS definitions, at least 1 policy with a security profile that has IPS scanning must be enabled.

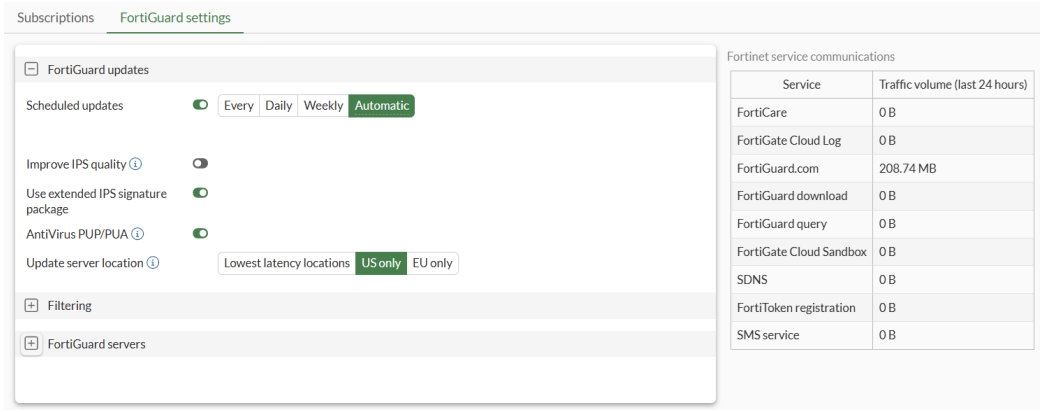
Scheduling updates ensures that the virus and IPS definitions are downloaded to your FortiGate on a regular basis.

Updating definitions can cause a brief disruption in traffic that is currently being scanned while the FortiGate unit applies the new signature database. Updates should be scheduled during off-peak hours when network usage is at a minimum to ensure that network activity will not be affected by downloading the definitions files.

A schedule of once a week means any urgent updates will not be pushed until the scheduled time. If an urgent update is required, click the *Update Licenses & Definitions Now* button to manually update the definitions.

To configure scheduled updates in the GUI:

1. Go to *System* > *FortiGuard* > *FortiGuard settings*
2. In the *FortiGuard Updates* section, enable *Scheduled Updates*.
3. Configure the update schedule:



4. Click *Apply*.

To configure scheduled updates in the CLI:

```
config system autoupdate schedule
    set status enable
    set frequency {every | daily | weekly}
    set time <hh:mm>
    set day <day_of_week>
end
```

Sending malware statistics to FortiGuard

FortiGate devices periodically send encrypted antivirus, IPS, botnet IP list, and application control statistics to FortiGuard. Included with these data is the IP address and serial number of the FortiGate, and the country that it is in. This information is never shared with external parties, Fortinet Privacy Policy.

The malware statistics are used to improve various aspects of FortiGate malware protection. For example, antivirus data allow FortiGuard to determine what viruses are currently active. Signatures for those viruses are kept in the Active AV Signature Database that is used by multiple Fortinet products.Inactive virus signatures are moved to the Extended AV Signature Database (see Configuring FortiGuard updates on page 3926). When events for inactive viruses start appearing in the malware data, the signatures are moved back into the AV Signature Database.

The FortiGate and FortiGuard servers go through a 2-way SSL/TLS 1.2 authentication before any data is transmitted. The certificates used in this process must be trusted by each other and signed by the Fortinet CA server.

The FortiGate only accepts data from authorized FortiGuard severs. Fortinet products use DNS to find FortiGuard servers and periodically update their FortiGate server list. All other servers are provided by a list that is updated through the encrypted channel.

Malware statistics are accumulated and sent every 60 minutes by default.

To configure sharing this information, use the following CLI command:

```
config system global
    set fds-statistics {enable | disable}
    set fds-statistics-period <minutes>
end
```

The submission of malware data is in accordance with the Fortinet Privacy Policy.

There is no sensitive or personal information included in these submissions. Only malware statistics are sent.

Fortinet uses the malware statistics collected in this manner to improve the performance of the FortiGate services and to display statistics on the Fortinet Support website for customers registered FortiGate devices.

Fortinet may also publish or share statistics or results derived from this malware data with various audiences. The malware statistics shared in this way do not include any customer data.

Update server location

Administrators can specify the location of the FortiGuard update server used by FortiGate. You can can set the location to only servers in the USA, only servers in the European Union (EU), or to servers with the lowest latency.

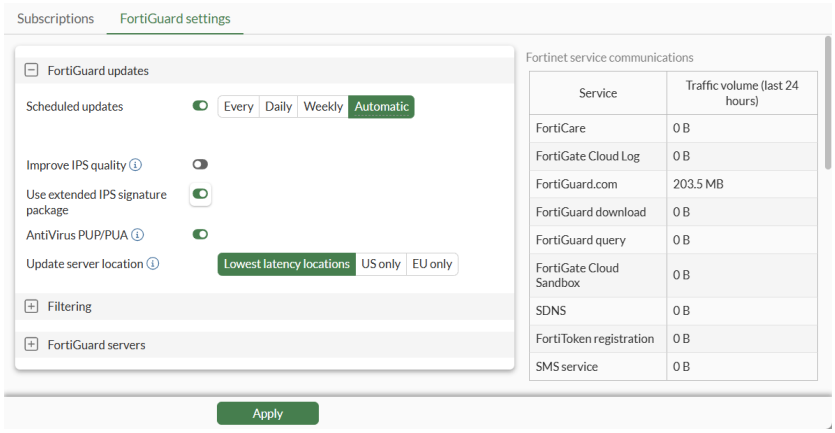
In EU locations, it can be required that certain traffic is only handled by servers located in the EU. By setting the update server location to EU only, the FortiGate will use EU domains to resolve to EU servers for FortiGuard traffic to update, URL rating, and IoT servers.

Server location	Anycast domain name	Non-Anycast FQDN addresses
EU only	euupdate.fortinet.net euguardservice.fortinet.net	
US only	usupdate.fortinet.net usguardservice.fortinet.net	usupdate.fortiguard.net UDP: usservice.fortiguard.net HTTPS: ussecurewf.fortiguard.net
Lowest latency (automatic)	globalupdate.fortinet.net globalguardservice.fortinet.net	update.fortiguard.net UDP: service.fortiguard.net HTTPS: securewf.fortiguard.net

On hardware FortiGate devices, the default is *Lowest latency locations*. On VM devices, the default is *US only*.

To configure the update server location in the GUI:

1. Go to the *System* > *FortiGuard* > *FortiGuard settings* tab.
2. In the *FortiGuard updates* section, set *Update server location* to *Lowest latency locations*, *US only*, or *EU only*.



3. Click *Apply*.

To configure the update server location in the CLI:

```
config system fortiguard
    set update-server-location {automatic | usa | eu}
end
```

Filtering

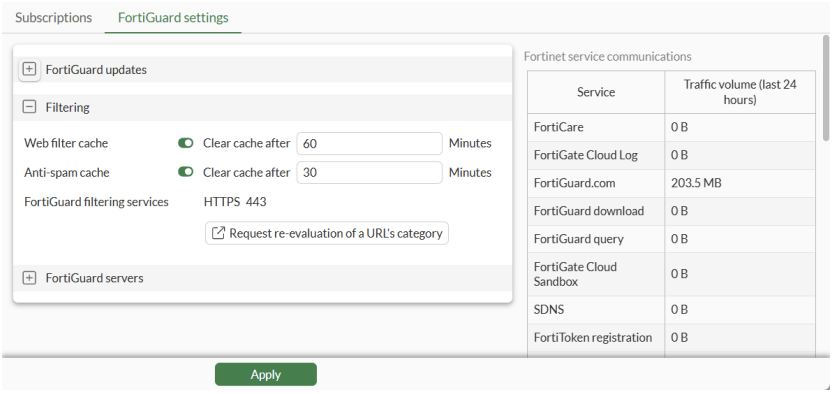
Web filtering is used to block access to harmful, inappropriate, and dangerous web sites. See FortiGuard filter on page 2315.

Email filtering is used to detect and block spam messages. Ssee FortiGuard-based filters on page 2540.

To configure filtering in the GUI:

1. Go to *System* > *FortiGuard* > *FortiGuard settings*
2. In the *Filtering* section, configure the settings as needed:

Web filter cache	Enable/disable web filter cache, and set the amount of time that the FortiGate will store a blocked IP address or URL locally. After the time expires, the FortiGate contacts the FDN to verify the address.
Anti-spam cache	Enable/disable anti-spam cache, and set the amount of time that the FortiGate will store an email address locally.
FortiGuard filtering services	The protocol and port used to contact the FortiGuard servers. These options can be changed in the CLI.
Filtering service availability	The status of the filtering service. Click *Refresh* if the filtering service is not available.
Request re-evaluation of a URL's category	Click to re-evaluate a URL category rating on the FortiGuard web filter service.



3. Click *Apply*.

To configure filtering in the CLI:

```
config system fortiguard
    set protocol {https | udp}
    set port {443 | 53 | 8888}
    set antispam-force-off {enable | disable}
    set antispam-cache {enable | disable}
    set antispam-cache-ttl <integer>
    set antispam-cache-mpercent <percent>
    set antispam-timeout <integer>
```

```
    set webfilter-force-off {enable | disable}
    set webfilter-cache {enable | disable}
    set webfilter-cache-ttl <integer>
    set webfilter-timeout <integer>
end
```

When anycast is enabled (by default), the protocol is HTTPS, and the port is 443.

FortiGuard server preference NEW

On the *System* > *FortiGuard* > *FortiGuard settings* page, use the *FortiGuard servers* section to configure FortiGuard server preference.

You can also configure the number of FortiGuard servers to use for load balancing:

```
config system fortiguard
    set load-balance-servers <integer>
end
```

To configure the FortiGuard server preference in the GUI:

1. Go to *System* > *FortiGuard* > *FortiGuard settings*.
2. Under *FortiGuard servers*, set *Server preference* to one of the following:

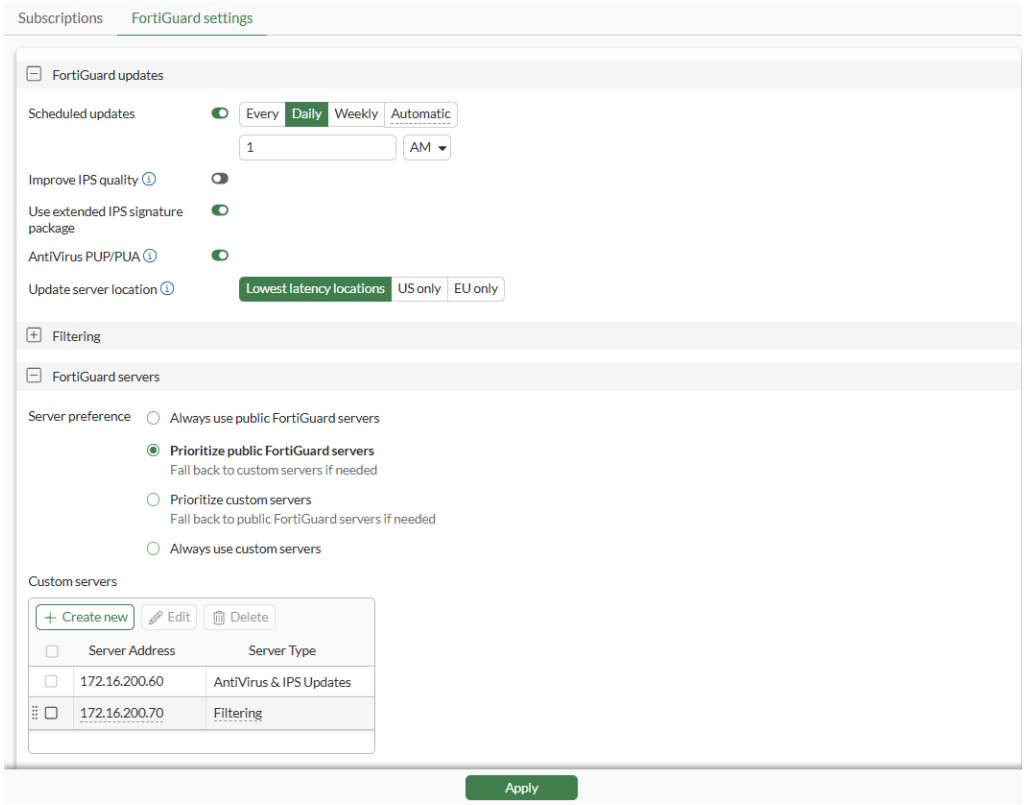
Always use public FortiGuard servers	Select for FortiGate to use only default FortiGuard servers. No custom servers are used. Hidden when central management is enabled and set to *FortiGate Cloud* in the GUI (`fortiguard` in the CLI).
Prioritize public FortiGuard servers Fall back to custom servers if needed	Select for FortiGate to prioritize public FortiGuard servers and fall back to custom servers if needed.
Prioritize custom servers Fall back to public FortiGuard servers if needed	Select for FortiGate to prioritize custom servers and fall back to public FortiGuard servers if needed. Hidden when *Always use public FortiGuard servers* is selected.

3. Add custom servers, if using:
   a. Under *Custom servers*, click *Create new*.
   b. Set *Address Type* to *FQDN*, *IPv4*, or *IPv6*.
   c. In *Address*, enter the server address.
   d. Click the *Type* box, select one or more server types, and click *Close*.
      Choose from:
      - *AntiVirus & IPS Updates*
      - *Filtering*
      - *IoT detection*

- *Virtual patching query*

e. Click *OK*. The custom server is displayed.

f. Repeat this step to add additional custom servers.



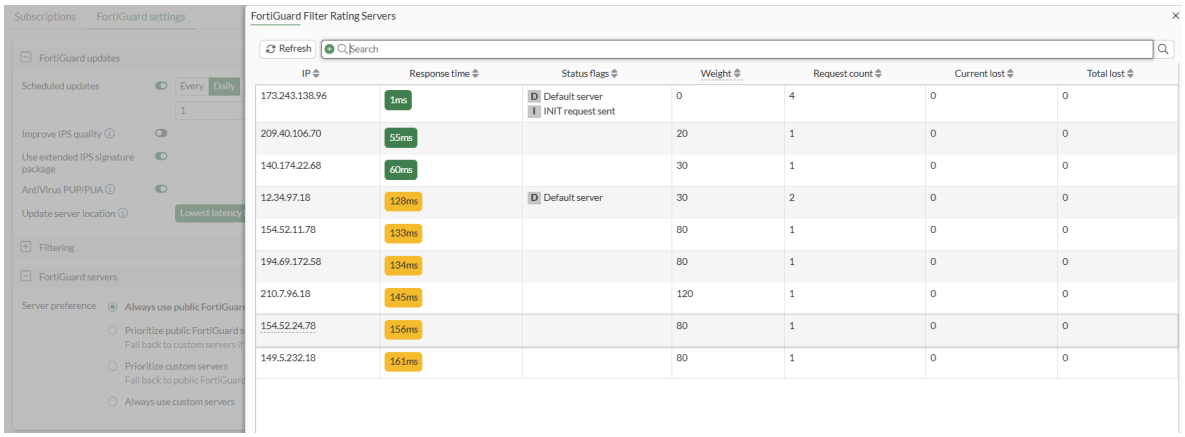
4. Click *Apply*.

Using only public FortiGuard servers

This example describes how to configure FortiGate to use only default FortiGuard servers. No custom servers are used.

To configure FortiGuard servers in the GUI:

1. Go to *System* > *FortiGuard* > *FortiGuard settings*.
2. Under *FortiGuard servers*, set *Server preference* to *Always use public FortiGuard servers*, and click *Apply*.
3. In the right-side gutter, under *FortiGuard Filter Rating Servers*, click *View Details*.
   Only default FortiGuard servers are used. No custom servers are used.



To configure FortiGuard servers in the CLI:

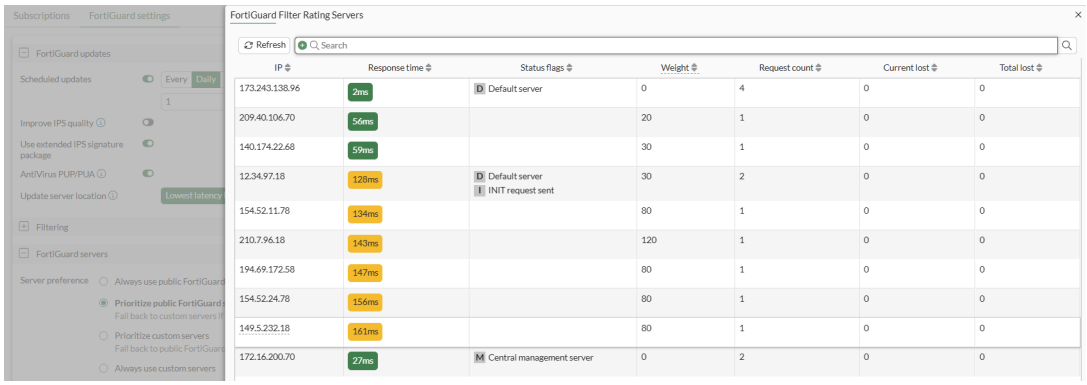
```
config system central-management
    set include-default-servers enable
    set use-default-servers-as-main enable
end
```

Prioritizing public FortiGuard servers

This example describes how to configure FortiGate to prioritize public FortiGuard servers and fall back to custom servers if needed.

To configure FortiGuard servers in the GUI:

1. Go to *System* > *FortiGuard* > *FortiGuard settings*.
2. Under *FortiGuard servers*, set *Server preference* to *Prioritize FortiGuard servers*.
3. Add one or more custom servers.
4. Click *Apply*.
5. In the right-side gutter, under *FortiGuard Filter Rating Servers*, click *View Details*.
   The custom server (172.16.200.70) is below the default FortiGuard servers.



To configure FortiGuard servers in the CLI:

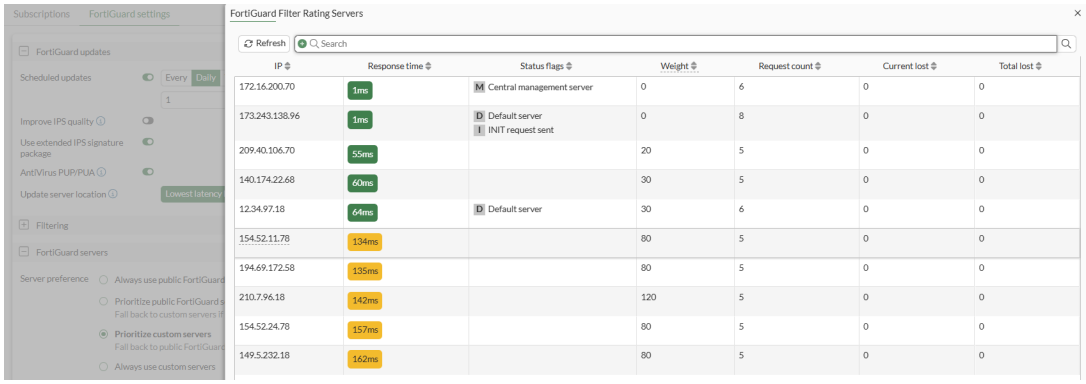
```
config system central-management
    config server-list
        edit 1
            set server-type update
            set addr-type ipv4
            set server-address 172.16.200.60
        next
        edit 2
            set server-type rating
            set addr-type ipv4
            set server-address 172.16.200.70
        next
    end
    set include-default-servers enable
    set use-default-servers-as-main enable
end
```

Prioritizing custom servers

This example describes how to configure FortiGate to prioritize custom servers and fall back to public FortiGuard servers if needed.

To configure FortiGuard servers in the GUI:

1. Go to *System* > *FortiGuard* > *FortiGuard settings*.
2. Under *FortiGuard servers*, set *Server preference* to *Prioritize custom servers*.
3. Add one or more custom servers.
4. In the right-side gutter, under *FortiGuard Filter Rating Servers*, click *View Details*.
   The custom server (172.16.200.70) is above the default FortiGuard servers.



To configure FortiGuard servers in the CLI:

```
config system central-management
    config server-list
        edit 1
            set server-type update
```

```
            set addr-type ipv4
            set server-address 172.16.200.60
        next
        edit 2
            set server-type rating
            set addr-type ipv4
            set server-address 172.16.200.70
        next
    end
    set include-default-servers enable
    set use-default-servers-as-main disable
end
```

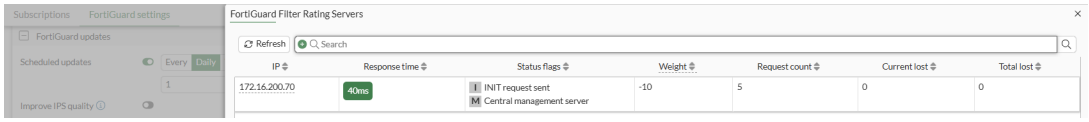
Using only custom servers

This example describes how to configure FortiGate to always use custom servers.

When central management is enabled and set to *FortiGate Cloud* in the GUI (`fortiguard` in the CLI), the *Always use custom servers* option is hidden.

To configure FortiGuard servers in the GUI:

1. Go to *System* > *FortiGuard* > *FortiGuard settings*.
2. Under *FortiGuard servers*, set *Server preference* to *Always use custom servers*.
3. Add one or more custom servers.
4. In the right-side gutter, under *FortiGuard Filter Rating Servers*, click *View Details*.
   Only custom servers are used. No default FortiGuard servers are used.



To configure FortiGuard servers in the CLI:

```
config system central-management
    config server-list
        edit 1
            set server-type update
            set addr-type ipv4
            set server-address 172.16.200.60
        next
        edit 2
            set server-type rating
            set addr-type ipv4
            set server-address 172.16.200.70
        next
    end
    set include-default-servers disable
end
```

Using FortiManager as a local FortiGuard server

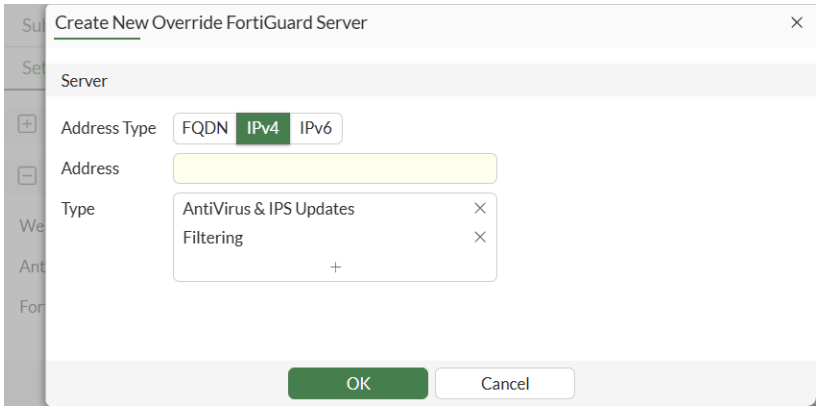
FortiManager can provide a local FortiGuard server with port 443 access.

Anycast FortiGuard settings force the rating process to use port 443, even with an override server. Using a unique address in the same subnet as the FortiManager access IP address, the FortiManager can provide local FortiGuard updates and rating access with a dedicated IP address and port 443.

> ⚠ On FortiManager, use the *Bind to IP* addresses for the update and rating services over TCP/443.
>
> The *Bind to IP* address does not need to be configured for update services if the default port was not changed to TCP/443. See Configuring network interfaces in the FortiManager Administration Guide for more information.

To use a FortiManager as a local FortiGuard server in the GUI:

1. Go to the *System* > *FortiGuard* > *FortiGuard settings* tab.
2. Under *FortiGuard servers*, set *Server preference* to *Prioritize public FortiGuard servers* or *Prioritize custom servers*.
3. In the *Custom servers* table, click *Create New*. The *Create New Override FortiGuard Server* pane opens.



4. Select the server address type: *IPv4*, *IPv6*, or *FQDN*.
5. Enter the FortiManager address in the *Address* field.
6. Select the type of server: *AntiVirus & IPS Updates*, *Filtering*, *IoT device collection*, or *Virtual patching query*.
7. Click *OK*.
8. Click *Create New* again to add a second override FortiManager for *Filtering*.
9. Click *OK*, then click *Apply*.

To use a FortiManager as a local FortiGuard server in the CLI:

```
config system central-management
    set type fortimanager
    set fmg "172.18.37.148"
    config server-list
        edit 1
            set server-type update
            set server-address 172.18.37.150
```

```
        next
        edit 2
            set server-type rating
            set server-address 172.18.37.149
        next
    end
    set fmg-update-port 443
    set include-default-servers enable
end
```

When `fmg-update-port` is set to `443`, the update process will use port 443 to connect to the override update server, which is the local FortiGuard server in the FortiManager. If this is not set, the update process will use port 8890, and the server address setting has to be the FortiManager access IP address. Override FortiGuard services come from the server list that is the local FortiGuard server in the FortiManager, and use the traditional, non-OCSP TLS handshake. If override servers in the FortiManager are not available, the default FortiGuard servers are connected, and the anycast OCSP TLS handshake is used.

The FortiManager IP address used in `set server-address` (for example, `set server-address 172.18.37.149`) corresponds to the *Bind to IP* setting configured on the FortiManager interface.

HA considerations

When FortiGate and FortiManager units are both in high availability (HA) clusters, and FortiGate is using FortiManager as an update server, you must configure the following secondary IP addresses:

- Configure secondary IP addresses for the service on both FortiManager units in the HA cluster
- Configure FortiGate to use the secondary FortiManager IP addresses.

In this example, the primary FortiManager unit is configured to use the secondary IP address of `10.4.1.204` for the rating service, and the secondary FortiManager unit is configured to use the secondary IP address of `10.4.1.205` for the rating service. FortiGate is configured to use the secondary IP addresses on FortiManager.

To configure FortiManager units in an HA cluster as web filter rating servers:

1. On the primary FortiManager in the HA cluster, configure a secondary IP address for the rating service:
   The `set rating-service-ip` is set to the secondary IP address for the rating query.

```
config system interface
    edit "port1"
        set ip 10.4.1.104 255.255.0.0
        set allowaccess ping https ssh snmp http webservice
        set serviceaccess fgtupdates fclupdates webfilter-antispam
        set rating-service-ip 10.4.1.204 255.255.0.0
        set type physical
    next
end
```

2. On the secondary FortiManager in the HA cluster, configure a secondary IP address for the rating service:
   The `set rating-service-ip` is set to the secondary IP address for the rating query.

```
config system interface
    edit "port1"
```

```
        set ip 10.4.1.105 255.255.0.0
        set allowaccess ping https ssh snmp http webservice
        set serviceaccess fgtupdates fclupdates webfilter-antispam
        set rating-service-ip 10.4.1.205 255.255.0.0
        set type physical
    next
end
```

3. On both the primary and secondary FortiManager units in the HA cluster, enable web filter query:

```
config fmupdate service
    set query-webfilter enable
end
```

4. On both the primary and secondary FortiManager units in the HA cluster, enable all web filter logs:

```
config fmupdate web-spam fgd-setting
    set wf-log all
end
```

5. As needed, run the `diagnose debug application fgdsvr 255` command.
   For additional details, use shell and run `bash$ tail -f /var/log/fgdsvr.log` to display the incoming URL queries.

To configure FortiGate:

1. On FortiGate, configure central management to use the secondary FortiManager IP addresses for the rating service:
   The `set rating-service-ip` is set to the secondary IP address on the primary and secondary FortiManager units in the HA cluster.

```
config system central-management
    set type fortimanager
    config server-list
        edit 1
            set server-type update
            set server-address 10.4.1.104
        next
        edit 3
            set server-type update
            set server-address 10.4.1.105
        next
        edit 2
            set server-type rating
            set server-address 10.4.1.204
        next
        edit 4
            set server-type rating
            set server-address 10.4.1.205
        next
    end
```

```
    set include-default-servers disable
end
```

2. Disable Anycast:

```
config system fortiguard
    set fortiguard-anycast disable
end
```

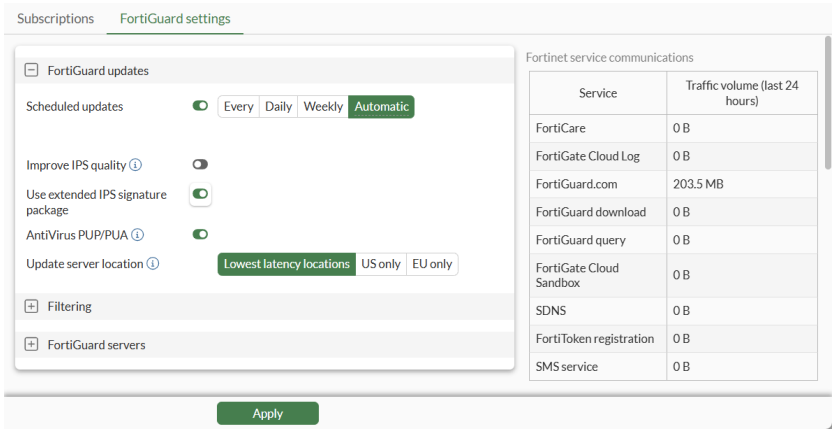
3. As needed, run the following diagnose commands:
   - Run `diagnose debug app urlfilter -1` to show the rating query response.
   - Run `diagnose debug rating` to show the rating service information.

Cloud service communication statistics

Fortinet service communications statistics are displayed on the *FortiGuard* page. The statistics correspond with the output from `diagnose sys service-communication`. The traffic volume values in the GUI are the sums of data from the last 24 hours.

To view Fortinet service communications statistics:

1. Go to the *System* > *FortiGuard* > *FortiGuard settings* tab.
   The *Fortinet service communications* statistics are displayed on the right side of the screen:



2. Enter the following CLI command:

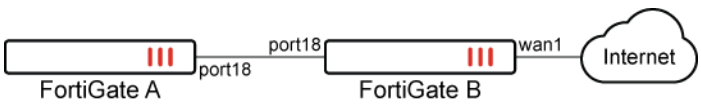
```
# diagnose sys service-communication
FortiCare:
The last 1 hour(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0
The last 24 hours(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
The last 7 days(in bytes):  0 0 0 0 0 0 0
FortiGuard Download:
The last 1 hour(in bytes):  0 648 3024 0 0 3056744 0 0 3024 0 0 3336
The last 24 hours(in bytes):  3066776 2960576 47616 71768 102384 68864 48224 82368 95376 70984
68048 75360 75112 83496
69848 51640 128360 20437472 99456 104384 48376 143024 25238104 73992
The last 7 days(in bytes):  6758344 120704952 0 0 0 0 0
```

```
FortiGuard Query:
The last 1 hour(in bytes):  0 10731 1622 0 0 1622 0 0 1622 0 0 1980
The last 24 hours(in bytes):  17577 21819 18521 18961 17309 16925 16937 18321 18614 17763
16266 17207 17775 17360 18076
 18496 18704 18848 18600 17923 17686 17935 29384 31393
The last 7 days(in bytes):  216220 267569 0 0 0 0 0
FortiCloud Log:
The last 1 hour(in bytes):  0 0 0 0 0 0 0 0 0 0 0
The last 24 hours(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
The last 7 days(in bytes):  0 0 0 0 0 0 0
FortiSandbox Cloud:
The last 1 hour(in bytes):  0 0 0 0 0 0 0 0 0 0 0
The last 24 hours(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
The last 7 days(in bytes):  0 0 0 0 0 0 0
FortiGuard.com:
The last 1 hour(in bytes):  0 0 1647 0 0 0 0 0 0 0 0
The last 24 hours(in bytes):  1647 428908 1647 318 1647 102859 318 318 1647 102859 318 318 318
104188 318 318 1647 1028
59 318 318 318 104188 318 1647
The last 7 days(in bytes):  642804 1100822 0 0 0 0 0
SDNS Service:
The last 1 hour(in bytes):  0 0 0 0 0 0 0 0 0 0 0
The last 24 hours(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
The last 7 days(in bytes):  0 0 0 0 0 0 0
FortiToken Registration:
The last 1 hour(in bytes):  0 0 0 0 0 0 0 0 0 0 0
The last 24 hours(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
The last 7 days(in bytes):  0 0 0 0 0 0 0
SMS Service:
The last 1 hour(in bytes):  0 0 0 0 0 0 0 0 0 0 0
The last 24 hours(in bytes):  0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
The last 7 days(in bytes):  0 0 0 0 0 0 0
```

FortiGate Cloud / FDN communication through an explicit proxy

Explicit proxy communication to FortiGate Cloud and FortiGuard servers from FortiGate is enabled. A proxy server can be configured in the FortiGuard settings so that all FortiGuard connections under the `forticldd` process can be established through the proxy server.

> Not all FortiGuard services are supported by these proxy settings. For example, web filter service traffic to FortiGuard will not be directed to the configured proxy.



To configure a proxy server and communicate with FortiGate Cloud though it:

1. Configure FortiGate B as a proxy server:

```
config firewall proxy-policy
    edit 1
        set proxy explicit-web
        set dstintf "wan1"
        set srcaddr "all"
        set dstaddr "all"
        set service "webproxy"
        set action accept
        set schedule "always"
        set logtraffic all
        set users "guest1"
    next
end
config user local
    edit "guest1"
        set type password
        set passwd 123456
    next
end
config authentication scheme
    edit "local-basic"
        set method basic
        set user-database "local-user-db"
    next
end
config authentication rule
    edit "local-basic-rule"
        set srcaddr "all"
        set ip-based disable
        set active-auth-method "local-basic"
    next
end
```

2. Configure a firewall policy on FortiGate B to allow FortiGate A to get DNS resolution:

```
config firewall policy
    edit 1
        set name "dns"
        set srcintf "port18"
        set dstintf "wan1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "DNS"
        set fsso disable
        set nat enable
    next
end
```

3. Configure the FortiGuard proxy settings on FortiGate A:

```
config system fortiguard
    set proxy-server-ip 10.2.2.2
    set proxy-server-port 8080
    set proxy-username "guest1"
    set proxy-password 123456
end
```

4. On FortiGate A, log in to FortiGate Cloud to activate the logging service:

```
execute fortiguard-log login <username> <password>
```

5. On FortiGate A, view the `forticldd` debug message to see the connection to the log controller through the proxy server:

```
# diagnose test application forticldd 1
```

FDS-only ISDB package in firmware images

FortiOS firmware images include Fortinet objects in the built-in Internet Service Database (ISDB).

```
# diagnose firewall internet-service list
List internet service in kernel(global):
Internet Service Database Kernel Table: size 14974 bytes, Entry size 5844 bytes, number of index
table entries 165 number of IP range table entries 0

Group(0): Weight(15), number of entries(162)
......
```

This lightweight ISDB package allows firewall rules and policy routes that use ISDB to access FortiGuard servers to continue working after upgrading FortiOS. For example, the following policy will work after an upgrade:

```
config firewall policy
    edit 440
        set name "Fortinet Updates"
        set srcintf "port25"
        set dstintf "port1"
        set srcaddr "FortiAnalyzer" "FortiAuthenticator" "Tesla Management Interface"
"BackupFortinet" "SipFW" "ConnectVPNMgmt"
        set internet-service enable
        set internet-service-id 1245187 1245326 1245324 1245325 1245193 1245192 1245190 1245185
        set action accept
        set schedule "always"
        set logtraffic all
        set fsso disable
    next
end
```

After the FortiGate reboots after a firmware update, an automatic update will run in five minutes so that the FortiGate can get the ISDB, whether or not scheduled update is enabled.

```
# diagnose autoupdate versions | grep Internet -A 6

Internet-service Full Database
---------
Version: 7.02217 signed
Contract Expiry Date: n/a
Last Updated using manual update on Thu Mar 10 12:06:58 2022
Last Update Attempt: Thu Mar 10 12:07:27 2022
```

Disable all cloud communication

The FortiGate communicates with various services, such as FortiGuard download and query services and FortiCloud and other cloud related services to download service packages. It also makes queries for various real-time filtering capabilities, and performs logging and synchronization tasks.

The communication statistics can be viewed in the GUI at *System > FortiGuard* on the *FortiGuard settings* tab. The statistics can also be retrieved from the CLI:

```
# diagnose sys service-communication
```

To disable these communications, use the following CLI command::

```
config system global
    set cloud-communication disable
end
```

When cloud-communication is disabled, the forticldd and updated daemons are shutdown and multiple settings are disabled.

The following settings are automatically changed:

```
config system global
    set fds-statistics disable
end
config system central-management
    set type none
    set include-default-servers disable
end
config system fortiguard
    set antispam-force-off enable
    set outbreak-prevention-force-off enable
    set webfilter-force-off enable
end
config system email-server
    set server ''
end
config system ntp
    set ntpsync disable
end
config system autoupdate schedule
    set status disable
end
```

```
config system autoupdate tunneling
    set status disable
end
config log fortiguard setting
    set status disable
end
```

To reenable cloud communications, each individual setting must be changed after running the following CLI command:

```
config system global
    set cloud-communication enable
end
```

For example, to reenable automatically connecting and logging in to FortiCloud:

```
config system fortiguard
    set auto-join-forticloud enable
end
```

To reenable the email server:

```
config system email-server
    set server "fortinet-notifications.com"
    set port 465
    set security smtps
end
```

To reenable NTP synchronization:

```
config system ntp
    set ntpsync enable
end
```

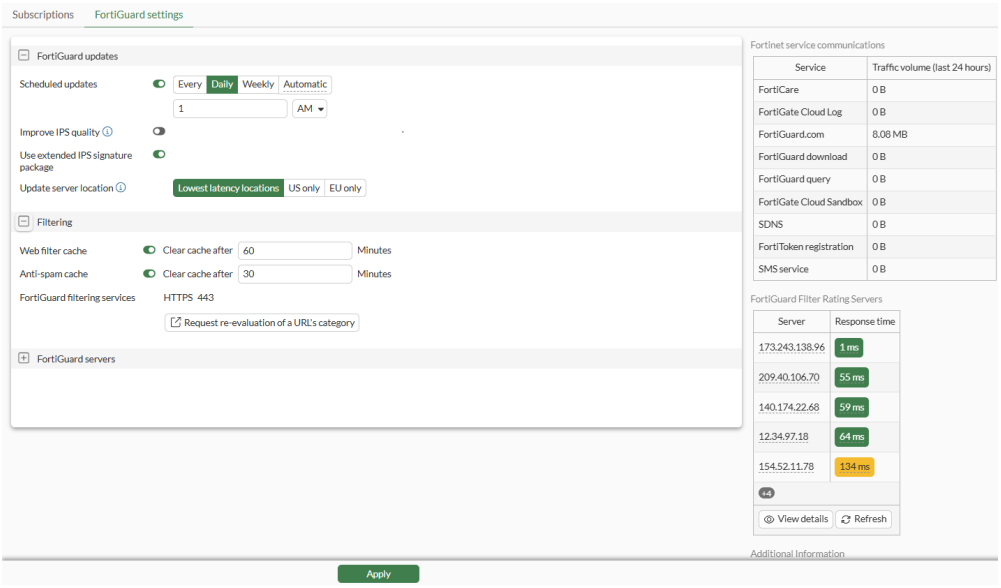
FortiGuard filter rating servers NEW

FortiGuard rating servers are selected based on fastest round-trip time (RTT).

On the *System > Settings > FortiGuard > FortiGuard settings* page, the *FortiGuard Filter Ratings Servers* table communicates connection status to filter rating servers. Click the *View details* button to view the same information available with the `diagnose debug rating` command.

To view connectivity status:

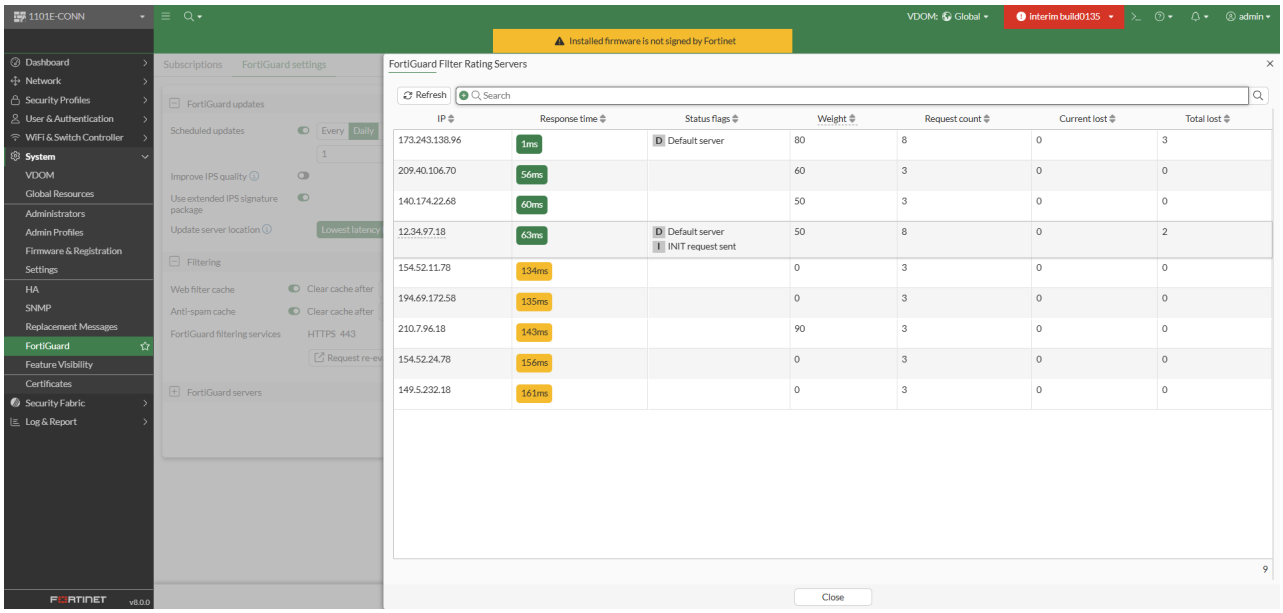
1. Go to *System* > *FortiGuard* > *FortiGuard settings*.



2. In the right-side gutter, the *FortiGuard Filter Rating Servers* displays connectivity information for each server in the following columns:
   - *Server*: Displays the IP address for each server.
   - *Response time*: Displays a color-coded round-trip time.

   The colors in the *Response time* column indicate latency:
   - Green: Indicates less than or equal to 100 ms.
   - Yellow: Indicates 101 - 200 ms.
   - Red: Indicates 201 ms and higher.
3. Click *Refresh* for the latest connectivity information.
4. Click *View Details*. The following columns of information are displayed:

   The same information can be viewed with the `diagnose debug rating` command.



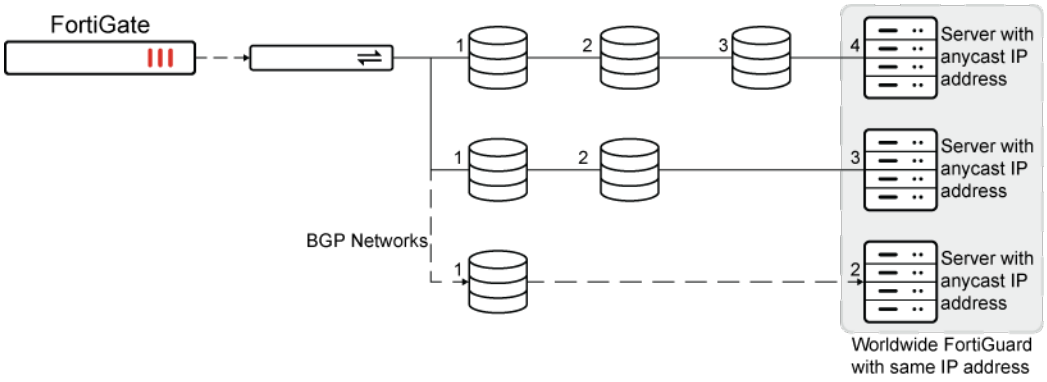
Column	Description
IP	Displays the IP address for each server.
Response time	Displays a color-coded round-trip time. Used to select a server.
Status flags	Indicates the server status: D: Indicates the server was found through the DNS lookup of the hostname. If the hostname returns more than one IP address, all of the IP addresses are D I: Indicates the server to which the last INIT request was sent. F: The server has not responded to requests and is considered to have failed. T: The server is currently being timed.
Weight	Displays the calculated weight of the server. The starting weight of a server is 0 (in Anycast setups) or the time zone difference #10 (in unicast setups). The weight goes up when the requests fail and down when they succeed, and the starting weight is used as the minimum for that IP.
Request count	The number of requests from FortiGate to the server.
Current lost	The number of recent and consecutive queries without reply.
Total lost	The historical total number of queries without reply. The values reset when the device restarts.

5. Click *Close*.

Anycast

FortiGuard servers use Anycast addresses in order to optimize and distribute traffic across many servers. Anycast is the default access mode for FortiGates when connecting to FortiGuard which by default utilizes HTTPS and port 443.

Each type of FortiGuard servers and services have a FortiGuard domain name that resolves to a single Anycast IP address. Regardless of where the FortiGate is located, the resolution is still the same. Fortinet maintains the network in the background to ensure routes to the FortiGuard servers are optimized. In the below diagram, several servers have the same Anycast IP, but the FortiGate will connect to the one with the least hops.



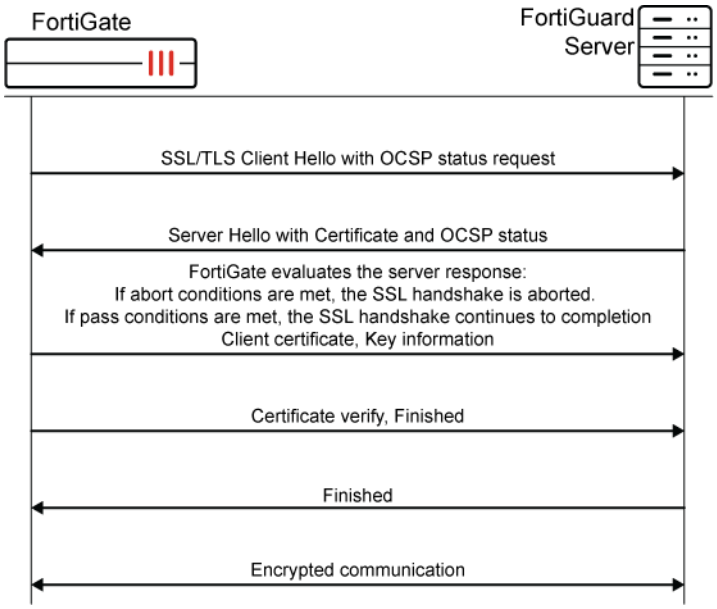
Connection and OCSP stapling

When the FortiGate connects to a FortiGuard server, it is important for it to validate the server is indeed a real FortiGuard server. Hence, FortiGuard servers provide the following security:

- The domain name of each FortiGuard service is the common name in that service's certificate, which is signed by a third-party intermediate CA.
- The FortiGuard server also applies Online Certificate Status Protocol (OCSP) stapling check, in which it attaches a time-stamped OCSP status of the server certificate from the OCSP server to the TLS response.

This ensures FortiGate can validate the FortiGuard server certificate efficiently during the TLS handshake.

The following illustrates the connection process:



FortiGate will only complete the TLS handshake with an anycast server when abort conditions are not met. Abort conditions include:

- The CN in the server's certificate does not match the domain name resolved from the DNS.
- The OCSP status is revoked or unknown.
- The issuer-CA is revoked by the root-CA.

To configure the anycast FortiGuard access mode:

```
config system fortiguard
    set fortiguard-anycast enable
end
```

If FortiGuard is not reachable via Anycast, choose between the following options to work around this issue:

1. Switch to other Anycast servers:

```
config system fortiguard
    set fortiguard-anycast enable
    set fortiguard-anycast-source aws
end
```

2. Disable Anycast and use HTTPS:

```
config system fortiguard
    set fortiguard-anycast disable
    set protocol https
    set port 8888
end
```

3. Disable Anycast and use UDP:

```
config system fortiguard
    set fortiguard-anycast disable
    set protocol udp
    set port 53
end
```

Anycast and unicast services

The following services are accessed by FortiGate:

Service	Non-Anycast FQDN addresses	Anycast Domain name
FortiGuard Object download	update.fortiguard.net	globalupdate.fortinet.net
Querying service (web-filtering, anti-spam ratings) over HTTPS	securewf.fortiguard.net	globalguardservice.fortinet.net
Querying service (web-filtering, anti-spam ratings) over UDP	service.fortiguard.net	Service only in Unicast
Device info Collection	Service only in Anycast	globaldevcollect.fortinet.net
Device info Query	Service only in Anycast	globaldevquery.fortinet.net

Service	Non-Anycast FQDN addresses	Anycast Domain name
FortiGate Cloud logging	logctrl1.fortinet.com	globallogctrl.fortinet.net
FortiGate Cloud management	mgrctrl1.fortinet.com	globalmgrctrl.fortinet.net
FortiGate Cloud messaging	msgctrl1.fortinet.com	globalmsgctrl.fortinet.net
FortiGate Cloud sandbox	aptctrl1.fortinet.com	globalaptctrl.fortinet.net
FortiCare registration	directregistration.fortinet.com	globalregistration.fortinet.net
Secure DNS	sdns.fortinet.net	globalsdns.fortinet.net
FortiCloud FortiClient	forticlient.fortinet.net	globalfctupdate.fortinet.net
FortiMobile Tokens	directregistration.fortinet.com	globalftm.fortinet.net
EMS cloud	forticlient-emsproxy.forticloud.com	Service only in Unicast
DDNS	ddns.fortinet.net	globalddns.fortinet.net
GeoIP	gip.fortinet.net	globalgip.fortinet.net
IP blocklist	ipbl.fortinet.net	N/A

Online security tools

FortiGuard Labs provides a number of online security tools, including but not limited to:

- **URL lookup**
  Enter a website address to see if it has been rated and what category and classification it is filed as. If you find a site that has been wrongly categorized, use this page to request that the site be re-evaluated: https://www.fortiguard.com/webfilter
- **Threat Encyclopedia**
  Browse FortiGuard Labs extensive encyclopedia of threats. Search for viruses, botnet C&C, IPS, endpoint vulnerabilities, and mobile malware: https://www.fortiguard.com/encyclopedia
- **Application Control**
  Browse FortiGuard Labs extensive encyclopedia of applications: https://www.fortiguard.com/appcontrol

Feature visibility

Feature visibility is used to control which features are visible in the GUI. This allows features that are not in use to be hidden. Some features are also invisible by default and must be made visible before they can be configure in the GUI.

The visibility of a feature does not affect its functionality or configuration. Invisible features can still be configured using the CLI.

To change the visibility of features:

1. Go to *System* > *Feature Visibility*.
2. Change the visibility of the features as required.
   For information about what settings each option affects, click on the + icon to the right of the feature name.
   Changes are listed on the right side of the content pane.
3. Click *Apply*.

Certificates

This section contains topics about uploading certificates and provides examples of how certificates may be used to encrypt and decrypt communications, and represent the identity of the FortiGate. This sections assumes that you have a high level understanding of the public key infrastructure (PKI) system, particularly how entities leverage trusted certificate authorities (CAs) to verify the authenticating party, and how public and private certificate keys work to secure communications.

The certificates feature is hidden by default in FortiOS. In the GUI, go to *System* > *Feature Visibility* and enable *Certificates*.

Common certificate uses in FortiOS

Type	How to generate/import certificate	FortiGate Use Examples	Private Key[1]
Local Certificate	ACME, Self Sign, CSR, File Upload (PKCS #12 & CER+PEM)	Client certificates: • Site-to-Site VPN • Dialup SSL VPN • Dialup IPsec VPN Server certificates: • HTTPS admin access • Protecting an SSL Server	Yes
Remote Certificate	File Upload	SAML • SP: Security Fabric > Fabric Connectors > Security Fabric Setup > Advanced Options > SP certificate • IdP: Security Fabric > Fabric Connectors > Security Fabric Setup > Edit/create a SP > IdP certificate	No
Local CA (and sub-CA)	Online SCEP, File Upload	SSL Inspection Multiple Clients Connecting to Multiple Servers • Security Profiles > SSL/SSH Inspection > *profile_name* > CA certificate	Yes

Type	How to generate/import certificate	FortiGate Use Examples	Private Key[1]
Remote CA	Online SCEP, File Upload	Enable FortiGate to trust certificates signed by the remote CA. Example uses: • LDAPS connection • User authentication (policy & admin access) • FortiAnalyzer OFTP tunnel	No

[1]Certificates with a private key are uploaded in the following common formats:

- Certificate and private key in one file (PKCS #12)
  - .PFX
- Certificate and private key in separate files
  - Certificate: .CER, .DER
  - Key: .PEM
- When using Certificate Signing Request (CSR), the private key remains on the FortiGate and the signed CSR is returned to the FortiGate to complete the cert+key pair.
  - .CER

  See Import a certificate on page 3962 for more details.

For additional capabilities and enhanced certificate management, please review the FortiAuthenticator Administration Guide and Cookbook. FortiManager can integrate with FortiAuthenticator to provide large scale FortiGate certificate deployment and management; see FortiManager Examples.

The following topics provide information about certificates:

- Automatically provision a certificate on page 3955
- Generate a new certificate on page 3960
- Regenerate default certificates on page 3961
- Import a certificate on page 3962
- Generate a CSR on page 3964
- CA certificate on page 3967
- Remote certificate on page 3968
- Certificate revocation list on page 3968
- Export a certificate on page 3969
- Uploading certificates using an API on page 3970
- Enrollment over Secure Transport for automatic certificate management on page 3990
- ACME External Account Binding support on page 4001

The following topics provide examples of how to use certificates:

- Procuring and importing a signed SSL certificate on page 3974
- Microsoft CA deep packet inspection on page 3977
- Administrative access using certificates on page 3982
- Creating certificates with XCA on page 3982

- Basic site-to-site VPN with digital certificate on page 2867
- Configuring FortiClient EMS on page 4322
- Agentless VPN with certificate authentication on page 3392
- Agentless VPN with LDAP-integrated certificate authentication on page 3398
- Configuring certificates for SAML SSO on page 4460
- Protecting an SSL server on page 2700
- Using the default certificate for HTTPS administrative access on page 3819

Automatically provision a certificate

The Automated Certificate Management Environment (ACME), as defined in RFC 8555, is used by the public Let's Encrypt certificate authority (https://letsencrypt.org) to provide free SSL server certificates. The FortiGate can be configured to use certificates that are managed by Let's Encrypt, and other certificate management services, that use the ACME protocol. The server certificates can be used for secure administrator log in to the FortiGate.

ACME certificates do not support loopback interfaces.

- The FortiGate must have a public IP address and a hostname in DNS (FQDN) that resolves to the public IP address.
- The configured ACME interface must be publicly reachable, and port TCP/443 or TCP/80 must not be used by other services.
- By default, the challenge is sent on TCP/80. By default, port 80 on the FortiGate redirects to TCP/443 for security purposes.
- If TCP/443 is in use by a process on the FortiGate (such as HTTPS daemon), the ACME daemon will fall back to TCP/80 for the challenge.
- If TCP/80 is also used by another service, the ACME process will fail.
- If a VIP is in use on any of these ports, then the incoming ACME challenge will be processed by the VIP rather than the system/ACME daemon.

  A VIP can be intentionally used to allow servers behind the FortiGate to participate in the ACME process, and can also be used to forward ACME challenges to a management VDOM that is not directly reachable by the ACME service.
- The Subject Alternative Name (SAN) field is automatically filled with the FortiGate DNS hostname. It cannot be edited, wildcards cannot be used, and multiple SANs cannot be added.

Security

FortiOS supports two forms of ACME challenge for Let's Encrypt: TLS-ALPN-01 (on TCP/443) and HTTP-01 (on TCP/80). By default, it uses the TLS-ALPN-01 challenge. See Challenge Types for an overview of the Let's Encrypt challenge types.

TLS-ALPN-01

This challenge requires the FortiGate to provide a self-signed certificate that includes specific requirements for ACME, such as SAN and acmeIdentifier. This self-signed certificate is often identified by security scans as an issue, but this is expected behavior for the TLS-ALPN-01 challenge. See RFC 8737 for details.

HTTP-01

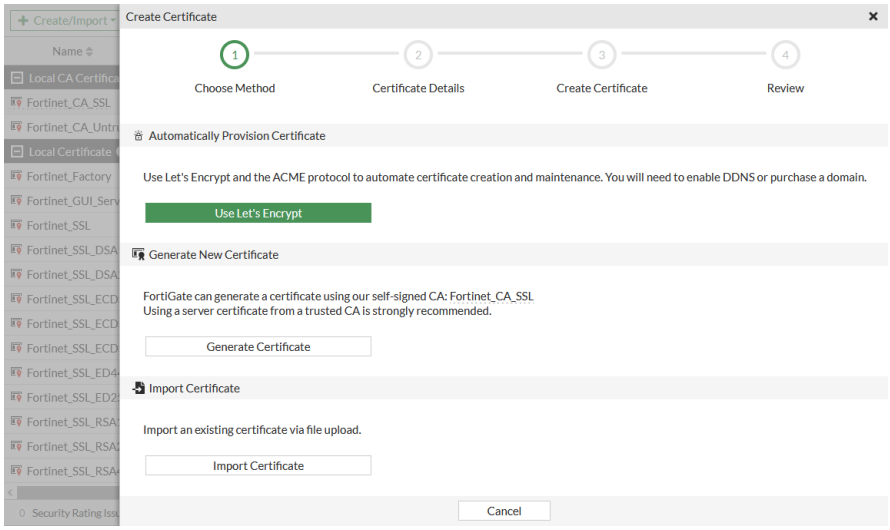
As indicated by the lack of an *s* in the protocol name, HTTP-01 uses an unencrypted protocol (HTTP) to complete the challenge on TCP/80. This is often identified by security scans as an issue, but this is expected behavior.

Example

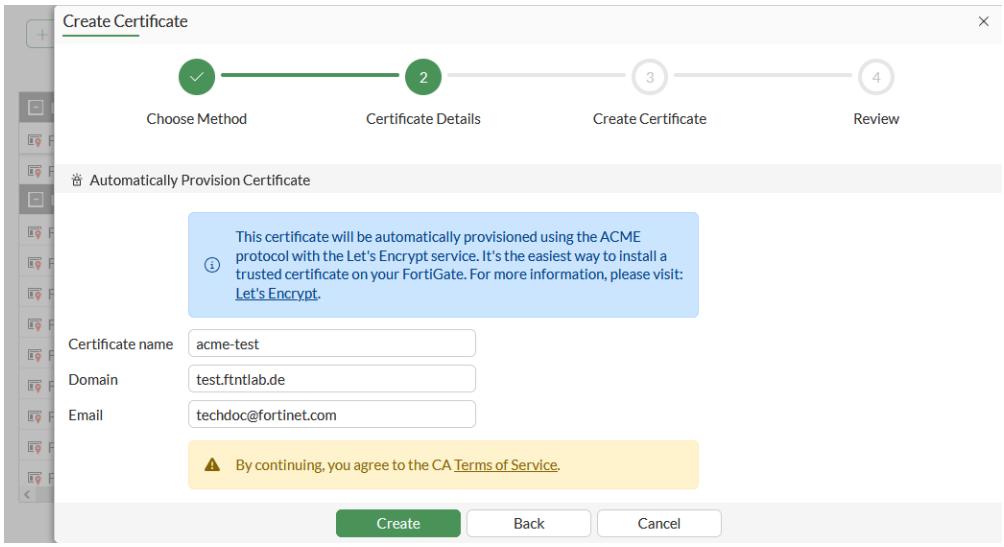
This example shows how to import an ACME certificate from Let's Encrypt, and use it for secured remote administrator access to the FortiGate.

To generate a certificate using ACME and Let's Encrypt:

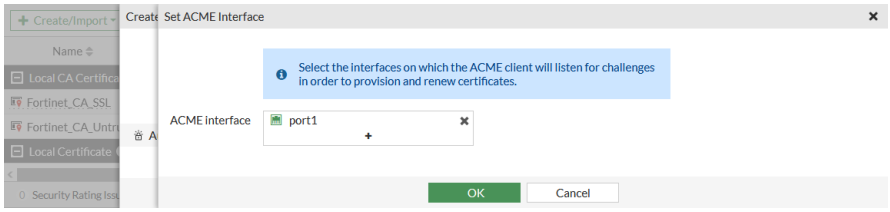
1. Go to *System* > *Certificates* and click *Create/Import* > *Certificate*.



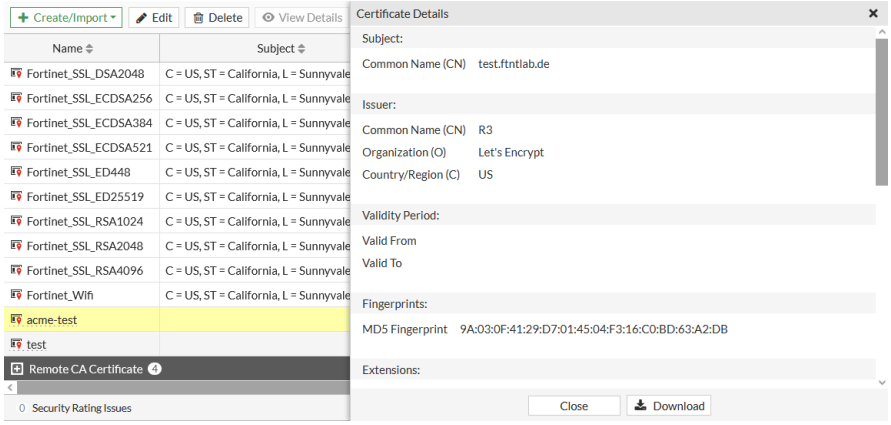
2. Click *Use Let's Encrypt*.
3. Set *Certificate name* to an appropriate name for the certificate. This is what is referenced when using the certificate in FortiGate configurations.
4. Set *Domain* to the public FQDN of the FortiGate.
5. Set *Email* to a valid email address. The email is not used during the enrollment process.



6. Click *Create*.
7. Set the *ACME interface*, on which the ACME client will listen for challenges in order to provision and renew certificates. The challenge is how the certificate signing request is validated by Let's Encrypt.



8. Click *OK*. Let's Encrypt provisions the certificate and the certificate is added to the certificate list in the *Local Certificates* section.
9. Click *View Details* to verify that the FortiGate's FQDN is in the certificate's *Subject: Common Name (CN)*.



To import an ACME certificate in the CLI:

1. Set the interface that the FortiGate communicates with Let's Encrypt on:

```
config system acme
    set interface "port1"
end
```

2. Make sure that the FortiGate can contact the Let's Encrypt enrollment server:

```
# execute ping acme-v02.api.letsencrypt.org
PING ca80a1adb12a4fbdac5ffcbc944e9a61.pacloudflare.com (172.65.32.248): 56 data bytes
64 bytes from 172.65.32.248: icmp_seq=0 ttl=60 time=2.0 ms
64 bytes from 172.65.32.248: icmp_seq=1 ttl=60 time=1.7 ms
64 bytes from 172.65.32.248: icmp_seq=2 ttl=60 time=1.7 ms
64 bytes from 172.65.32.248: icmp_seq=3 ttl=60 time=2.1 ms
64 bytes from 172.65.32.248: icmp_seq=4 ttl=60 time=2.0 ms

--- ca80a1adb12a4fbdac5ffcbc944e9a61.pacloudflare.com ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 1.7/1.9/2.1 ms
```

3. Configure the local certificate request:

```
config vpn certificate local
    edit "acme-test"
        set enroll-protocol acme2
        set acme-domain "test.ftntlab.de"
        set acme-email "techdoc@fortinet.com"
    next
By enabling this feature you declare that you agree to the Terms of Service at https://acme-
v02.api.letsencrypt.org/directory
Do you want to continue? (y/n)y
end
```

4. Verify that the enrollment was successful:

```
# get vpn certificate local details acme-test
path=vpn.certificate, objname=local, tablename=(null), size=2632
== [ acme-test ]
        Name:        acme-test
        Subject:     CN = test.ftntlab.de
        Issuer:      C = US, O = Let's Encrypt, CN = R3
        Valid from:  2021-03-11 17:43:04  GMT
        Valid to:    2021-06-09 17:43:04  GMT
        Fingerprint: 9A:03:0F:41:29:D7:01:45:04:F3:16:C0:BD:63:A2:DB
        Serial Num:  03:d3:55:80:d2:e9:01:b4:ca:80:3f:2e:fc:24:65:ad:7c:0c
ACME details:
        Status: The certificate for the managed domain has been renewed successfully and can
be used (valid since Thu, 11 Mar 2021 17:43:04 GMT).
        Staging status: Nothing in staging
```

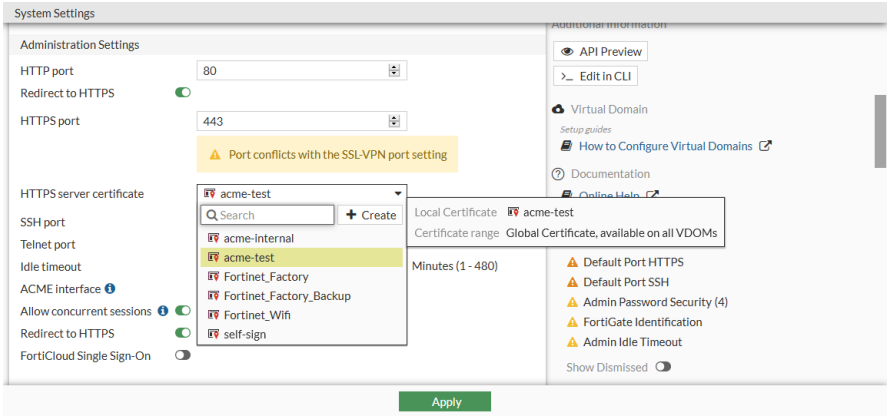
5. Check the ACME client full status log for the CN domain:

```
# diagnose sys acme status-full test.ftntlab.de
{
  "name": "test.ftntlab.de",
  "finished": true,
  "notified": false,
  "last-run": "Thu, 11 Mar 2021 18:43:02 GMT",
  "valid-from": "Thu, 11 Mar 2021 17:43:04 GMT",
```

```
  "errors": 0,
  "last": {
    "status": 0,
    "detail": "The certificate for the managed domain has been renewed successfully and can be
used (valid since Thu, 11 Mar 2021 17:43:04 GMT). A graceful server restart now is
recommended.",
    "valid-from": "Thu, 11 Mar 2021 17:43:04 GMT"
  },
  "log": {
    "entries": [
      {
        "when": "Thu, 11 Mar 2021 18:43:05 GMT",
        "type": "message-renewed"
      },
      ...
      {
        "when": "Thu, 11 Mar 2021 18:43:02 GMT",
        "type": "starting"
      }
    ]
  }
}
```


To exchange the default FortiGate administration server certificate for the new public Let's Encrypt server certificate in the GUI:

1. Go to *System* > *Settings* > *Access*.
2. Set *HTTPS server certificate* to the new certificate.



3. Click *Apply*.
4. Log in to the FortiGate using an administrator account from any internet browser. There should be no warnings related to non-trusted certificates, and the certificate path should be valid.

To exchange the default FortiGate administration server certificate for the new public Let's Encrypt server certificate in the CLI:

```
config system global
    set admin-server-cert "acme-test"
```

```
end
```

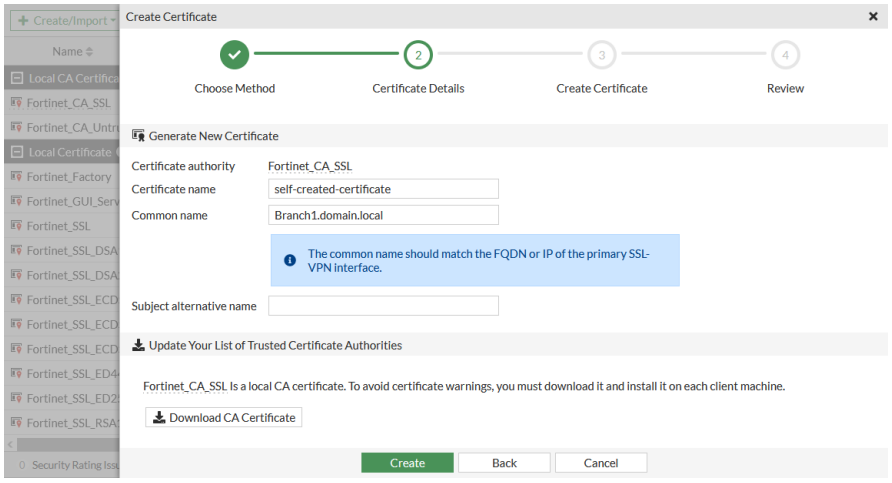
When you log in to the FortiGate using an administrator account there should be no warnings related to non-trusted certificates, and the certificate path should be valid.

Generate a new certificate

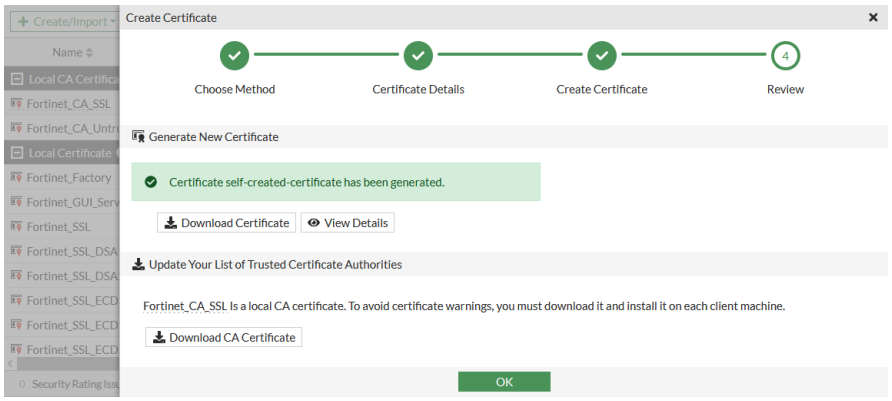
The FortiGate can generate a certificate using a pre-loaded, self-signed CA certificate: *Fortinet_CA_SSL*, instead of generating a CSR and providing it to a CA for signing. It is recommended that a server certificate from a well-known and trusted CA is used.

To generate a new certificate:

1. Go to *System* > *Certificates* and select *Create/Import* > *Certificate*.
2. Click *Generate Certificate*.
3. Set *Certificate name* to the name of the certificate. This is what is referenced when using the certificate in FortiGate configurations.
4. Set the *Common name* (CN) for the certificate. The common name should match the FQDN or IP of the primary Agentless VPN interface.
5. Optionally, set the *Subject alternative name*.
6. Click *Download CA Certificate* to download the CA certificate so that it can be installed or imported to all the machines that need to trust this certificate.



7. Click *Create*.
8. After the certificate is created, click *Download Certificate* to download the certificate. Click *View Details* to review the certificate details.



9. Click *OK*.

Regenerate default certificates

The FortiGate includes default certificates that are generated the first time that the FortiGate is booted up. In some circumstances, it can be necessary to regenerate these certificates, such as when they are nearing expiry, or if the key becomes compromised.

To regenerate default certificates:

```
# execute vpn certificate local generate default-gui-mgmt-cert
```

```
# execute vpn certificate local generate default-ssl-ca
```

```
# execute vpn certificate local generate default-ssl-ca-untrusted
```

```
# execute vpn certificate local generate default-ssl-key-certs
```

```
# execute vpn certificate local generate default-ssl-serv-key
```

`default-gui-mgmt-cert`	Regenerate the default GUI management admin-server (Fortinet_GUI_Server) certificate.
`default-ssl-ca`	Regenerate the default CA certificate (Fortinet_CA_SSL) used by SSL Inspection.
`default-ssl-ca-untrusted`	Regenerate the default untrusted CA certificate (Fortinet_CA_Untrusted) used by SSL Inspection.
`default-ssl-key-certs`	Regenerate the default RSA, DSA, ECDSA, and EdDSA key certificates for SSL resign: • Fortinet_SSL_DSA1024 • Fortinet_SSL_DSA2048 • Fortinet_SSL_ECDSA256 • Fortinet_SSL_ECDSA384 • Fortinet_SSL_ECDSA521

	• Fortinet_SSL_ED448 • Fortinet_SSL_ED25519 • Fortinet_SSL_RSA1024 • Fortinet_SSL_RSA2048 • Fortinet_SSL_RSA4096
`default-ssl-serv-key`	Regenerate the default server key (Fortinet_SSL) used by SSL Inspection.

Import a certificate

You can upload a certificate to the FortiGate that was generated on its own. This is typical of wildcard certificates (*.domain.tld) where the same certificate is used across multiple devices (FGT.domain.tld, FAZ.domain.tld, and so on), but can also be used for individual certificates as long as the information provided to the signing CA matches that of the FortiGate.

Any certificate uploaded to a VDOM is only accessible to that VDOM. Any certificate uploaded to the Global VDOM is globally accessible by all VDOMs.

A signed certificate that is created using a CSR that was generated by the FortiGate does not include a private key, and can be imported to the FortiGate from a the management computer or a TFTP file server.

There are three options:

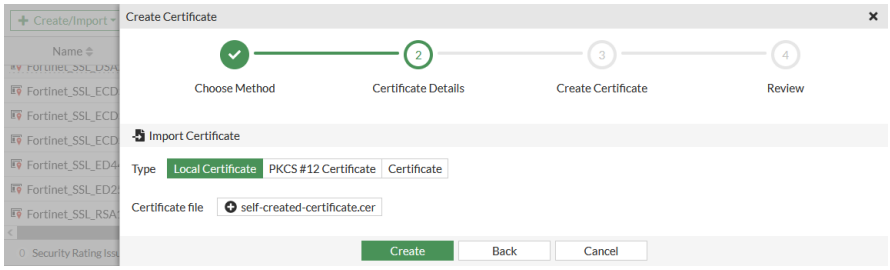
- Local certificate on page 3962
- PKCS #12 certificate on page 3963
- Certificate on page 3963

Local certificate

This option allows you to upload a single file and no key. Use it when you have created a CSR on the FortiGate (Generate a CSR on page 3964), as the key is generated as part of the CSR process and remains on the FortiGate. You must upload a .CER file.

To import a local certificate in the GUI:

1. Go to *System* > *Certificates* and select *Create/Import* > *Certificate*.
2. Click *Import Certificate*.
3. Set *Type* to *Local Certificate*.
4. Click *Upload*, and locate the certificate on the management computer.



5. Click *Create*, then click *OK* on the confirmation page.

To import a local certificate in the CLI:

```
execute vpn certificate local import tftp <filename> <tftp_IP> cer
```

PKCS #12 certificate

This option takes a specific certificate file type that contains the private key. The certificate is encrypted and a password must be supplied with the certificate file. PKCS #12 certificates are .PFX files.

To import a PKCS #12 certificate in the GUI:

1. Go to *System* > *Certificates* and select *Create/Import* > *Certificate*.
2. Click *Import Certificate*.
3. Set *Type* to *PKCS #12 Certificate*.
4. Click *Upload*, and locate the certificate on the management computer.
5. Enter the password, then confirm the password.
6. Optionally, customize the *Certificate name*.
7. Click *Create*, then click *OK* on the confirmation page.

To import a PKCS #12 certificate in the CLI:

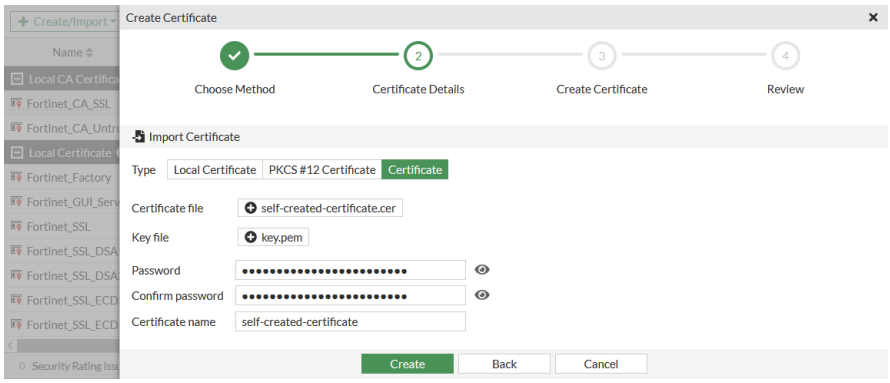
```
execute vpn certificate local import tftp <filename> <tftp_IP> p12 <password>
```

Certificate

This option is intended for certificates that were generated without using the FortiGate's CSR. Because the certificate private key is being uploaded, a password is required. This option is similar to PKCS #12 certificate, but the certificate and key file are separate files, usually .CER and .PEM files.

To import a certificate in the GUI:

1. Go to *System* > *Certificates* and select *Create/Import* > *Certificate*.
2. Click *Import Certificate*.
3. Set *Type* to *Certificate*.
4. In the *Certificate* field, click *Upload*, and locate the certificate on the management computer.
5. In the *Key file* field, click *Upload*, and locate the key file on the management computer.
6. Enter the password, then confirm the password.



7. Optionally, customize the *Certificate name*.
8. Click *Create*, then click *OK* on the confirmation page.

To import a certificate that requires a private key to a VDOM, or when VDOMs are disabled:

```
config vpn certificate {local | ca | remote | ocsp-server | crl}
```

Refer to the FortiOS CLI Reference for detailed options for each certificate type (local, CA, remote, OSCP server, CRL).

To import a global certificate that requires a private key when VDOMs are enabled:

```
config certificate {local | ca | remote | crl}
```

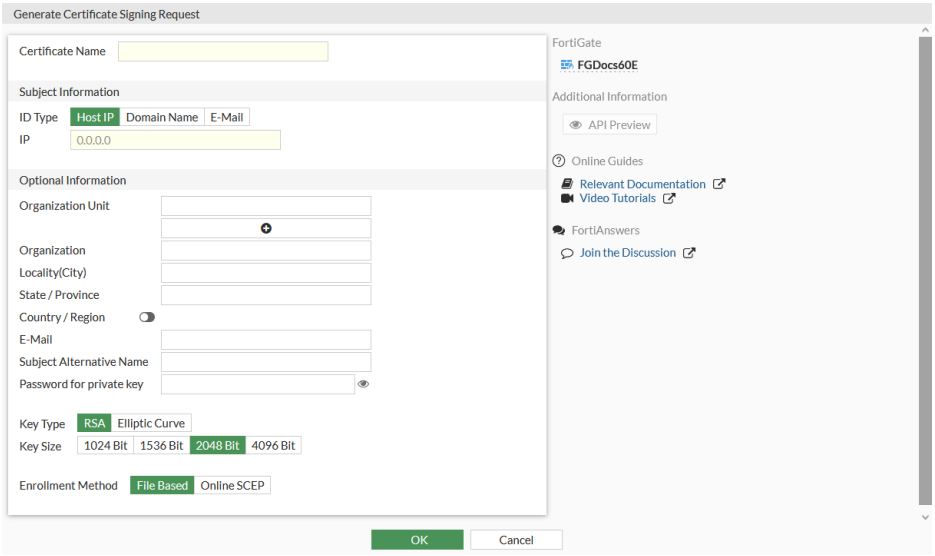
This command is only available when VDOMs are enabled. For details, see the FortiOS CLI Reference.

Generate a CSR

Certificate signing requests (CSRs) are used to generate a certificate which is then signed by a CA to create a chain of trust. The CSR includes details about the FortiGate and its public key. A CSR is not strictly necessary; some CAs allow you to provide the details of the FortiGate manually, but a CSR helps streamline the process.

To generate a CSR in the GUI:

1. Go to *System* > *Certificates* and select *Create/Import* > *Generate CSR*.



2. Enter the following information:

Certificate Name	Enter the certificate name; this is how it will appear in the *Local Certificates* list.
Subject Information	Specify an ID type: host IP address, domain name (FQDN), or email address.
Optional Information	Although listed as optional, we recommended entering the information for each field in this section. If you are generating a CSR for a third-party CA, you need to insure that these values reflect those listed for your company or organization at said certificate authority. If you are generating a certificate for a Microsoft CA, you need to check with the administrator regarding these values.
Organization Unit	Enter the name of the organizational unit under which the certificate will be issued.
Organization	Enter the overall name of the organization.
Locality(City)	Enter the city where the SSL certificate is located.
State / Province	Some issuers will reject a CSR that has an abbreviated state or province, so enter the full name of the state or province.
Country / Region	Enable the option and select the country from the dropdown.
E-Mail	Enter the email address of the technical contact for the SSL certificate that is being requested.

Subject Alternative Name	This field allows multiple domains to be used in an SSL certificate. Select from email addresses, IP addresses, URIs, DNS names, and so on. You can use multiple of the same type, or a variety of different types at the same time.
	Examples:
	`DNS:example.fortinet.com`
	`IP Address:192.168.99.100`
	`email:user1@fortinet.com`
	`IP:192.168.99.101, email:user2@fortinet.com`

Password for private key	If supplied, this is used as an encryption password for the private key file.
Key Type	Select *RSA* or *Elliptic Curve*.
Key Size	When *Key Type* is *RSA*, select 1024, 1536, 2048, or 4096 for bit-size/strength. We recommend using at least 2048 if your CA can issue certificates of that size.
Curve Name	When *Key Type* is *Elliptic Curve*, select the elliptic curve type: secp256r1, secp384r1, or secp521r1.

Enrollment Method	Select one of the following methods that determines how the CSR will be signed.
	• *File Based*: this will generate a certificate in the certificate menu under *Local Certificate*, which differs from the existing ones because it has no *Subject*, *Comments*, *Issuer*, or *Expires* values in the table. It will also show a *Pending* status because it is only a CSR at the moment and cannot function as a certificate just yet. You can download the CSR to provide to a CA for signing. If you open the CSR file, it should look similar to this:
	`-----BEGIN CERTIFICATE REQUEST-----`
	`MIIC7jCCAdYCAQAwgZUxCzAJBgNVBAYT (… )HEKjDX+Hg==`
	`-----END CERTIFICATE REQUEST-----`
	Next. the CSR file is supplied to a CA for signing and the returned file from the CA should be in .CER format. This file is then uploaded to the FortiGate by going to *System > Certificates > Import > Local Certificate* and uploading the CER file.
	• *Online SCEP*: the Simple Certificate Enrollment Protocol (SCEP) allows devices to enroll for a certificate by using a URL and a password. The SCEP server works as a proxy to forward the FortiGate's request to the CA and returns the result to the FortiGate (setting up an SCEP server is beyond the scope of this topic). Once the request is approved by the SCEP server, the

	FortiGate will have a signed certificate containing the details provided in the CSR.

3. Click *OK*.
   The CSR generated, and can be downloaded from the local certificate list.

To generate a CSR in the CLI:

```
# execute vpn certificate local generate cmp-ec <certificate_name> <key_size> <server> <path>
<server_certificate> <auth_certificate> <user> <password> <subject> [SANs] [source_IP]
```

```
# execute vpn certificate local generate cmp-rsa <certificate_name> <key_size> <server> <path>
<server_certificate> <auth_certificate> <user> <password> <subject> [SANs] [source_IP]
```

```
# execute vpn certificate local generate ec <certificate_name> <curve_name> <subject> <country>
<state/province> <city> <organization> <OU> <email> [SANs] [options]
```

```
# execute vpn certificate local generate rsa <certificate_name> <key_size> <subject> <country>
<state/province> <city> <organization> <OU> <email> [SANs] [options]
```

cmp-ec	Generate an ECDSA certificate request over CMPv2. To enable SSL/TLS, append `https://` to the server address.
cmp-rsa	Generate a RSA certificate request over CMPv2. To enable SSL/TLS, append `https://` to the server address.
ec	Generate an elliptic curve certificate request.
rsa	Generate a RSA certificate request.

CA certificate

FortiGates come with many CA certificates from well-known certificate authorities pre-installed, just as most modern operating systems like Windows and MacOS. Use this option to add private CA certificates to the FortiGate so that certificates signed by this private CA are trusted by the FortiGate.

For example, a private CA can be used when two FortiGates are establishing a site-to-site VPN tunnel using a certificate not signed by a public or trustworthy CA, or for your LDAPS connection to your corporate AD server that also uses a certificate signed with a private CA in your domain. It is very common to upload a private CA when using PKI user authentication, since most PKI user certificates will be signed by an internal CA.

To import a CA certificate in the GUI:

1. Go to *System* > *Certificates* and select *Create/Import* > *CA Certificate*.
2. Set the *Type* to *Online SCEP* or *File*.
   - *Online SCEP*: Enter the *URL of the SCEP server* and optionally, the *Optional CA Identifier*. The FortiGate contacts an SCEP server to request the CA certificate.
   - *File*: Upload the CA certificate file directly from the management computer.
3. Click *OK*.

To import a CA certificate in the CLI:

```
# execute vpn certificate ca import auto <CA_server> [identifier] [source_ip] [fingerprint]
```

```
# execute vpn certificate ca import bundle <filename> <tftp_IP>
```

```
# execute vpn certificate ca import tftp <filename> <server_address>
```

```
# execute vpn certificate ems_ca import tftp <filename> <server_address>
```

Command	Description
`auto`	Import CA certificate via SCEP.
`bundle`	Import certificate bundle from a TFTP server.
`tftp`	Import CA certificate from a TFTP server.

In proxy mode, when building a certificate chain, WAD will consider user-installed CA certificates before falling back to searching CA certificates from FortiGuard or built-in certificates.

Remote certificate

Remote certificates are public certificates and contain only the public key. They are used to identify a remote device. For example, when configuring your FortiGate for SAML authentication with the FortiGate as an identity provider (IdP), you can optionally specify the service provider (SP) certificate. However, when configuring your FortiGate as a SP, you must specify the certificate used by the IdP. Both these certificates can be uploaded to the FortiGate as a remote certificate, since the private key is not necessary for its implementation.

To upload a remote certificate in the GUI:

1. Go to *System* > *Certificates* and select *Create/Import* > *Remote Certificate*.
2. Upload the remote certificate file directly from the management computer.
3. Click *OK*.

To upload a remote certificate in the CLI:

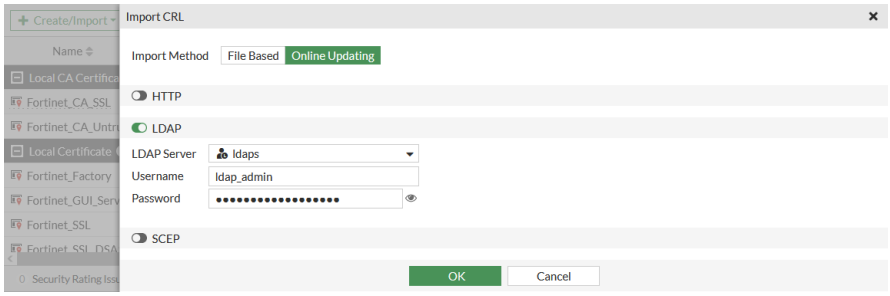
```
# execute vpn certificate remote import tftp <file_name> <server_address>
```

Certificate revocation list

Because it is not possible to recall a certificate, the certificate revocation list (CRL) details certificates signed by valid CAs that should no longer be trusted. Certificates may be revoked for many reasons, such as if the certificate was issued erroneously or if the private key of a valid certificate has been compromised.

To import a CRL in the GUI:

1. Go to *System* > *Certificates* and select *Create/Import* > *CRL*.
2. Set the *Import Method* to *File Based* or *Online Updating*.
   - *File Based*: Upload the CRL file directly from the management computer. CAs publish files containing the list of certificates that should no longer be trusted.
   - *Online Updating*: This is the preferred method to keep the list of revoked certificates up to date. Configure the protocols as required.
     - *HTTP*: Enter the *URL of the HTTP server*.
     - *LDAP*: Select the *LDAP Server* and enter the *Username* and *Password*.
     - *SCEP*: Select the *Certificate* and enter the *URL of the SCEP server*.



3. Click *OK*.

To import a CRL in the CLI:

```
# execute vpn certificate crl import auto <CRL_name>
```

Export a certificate

Certificates can be downloaded to the management computer in the GUI and exported to a TFTP server in the CLI.

To back up a certificate in the GUI:

1. Go to *System* > *Certificates*.
2. Select the certificate from the list.
3. Click *Download*.
4. Save the file to the management computer.

To export the certificate in the CLI:

```
# execute vpn certificate ca export tftp <certificate_name> <filename> <tftp_IP>
```

```
# execute vpn certificate local export tftp <certificate_name> <file_type> <filename> <tftp_
server>
```

```
# execute vpn certificate remote export tftp <certificate_name> <filename> <tftp_server>
```

Uploading certificates using an API

There are several API methods to upload a certificate based on the type and purpose of the certificate. The parameters of each method are available options, and some methods do not require all parameters to upload the certificate.

When uploading a certificate to the FortiGate using API, the certificate must be provided to the FortiGate in Base64 encoding. You must create a REST API user to authenticate to the FortiGate and use the generated API token in the request.

api/v2/monitor/vpn-certificate/ca/import

```
{
  "import_method": "[file|scep]",
  "scep_url": "string",
  "scep_ca_id": "string",
  "scope": "[vdom*|global]",
  "file_content": "string"
}
```

api/v2/monitor/vpn-certificate/crl/import

```
{
  "scope": "[vdom*|global]",
  "file_content": "string"
}
```

api/v2/monitor/vpn-certificate/local/import

```
{
  "type": "[local|pkcs12|regular]",
  "certname": "string",
  "password": "string",
  "key_file_content": "string",
  "scope": "[vdom*|global]",
  "acme-domain": "string",
  "acme-email": "string",
  "acme-ca-url": "string",
  "acme-rsa-key-size": 0,
  "acme-renew-window": 0,
  "file_content": "string"
}
```

api/v2/monitor/vpn-certificate/remote/import

```
{
  "scope": "[vdom*|global]",
```

```
  "file_content": "string"
}
```

api/v2/monitor/vpn-certificate/csr/generate

```
{
  "certname": "string",
  "subject": "string",
  "keytype": "[rsa|ec]",
  "keysize": [1024|1536|2048|4096],
  "curvename": "[secp256r1|secp384r1|secp521r1]",
  "orgunits": [
    "string"
  ],
  "org": "string",
  "city": "string",
  "state": "string",
  "countrycode": "string",
  "email": "string",
  "sub_alt_name": "string",
  "password": "string",
  "scep_url": "string",
  "scep_password": "string",
  "scope": "[vdom*|global]"
}
```

Example

In this example, a PKCS 12 certificate is uploaded as a local certificate using Postman as the API client. PowerShell is used for the Base64 encoding.

To upload a PKCS 12 certificate using an API:

1. In PowerShell , encode the PKCS 12 certificate to Base64:

   ```
   cd C:\users\username\desktop
   $pkcs12cert = get-content 'C:\users\path\to\certificate\certificatename.p12' -Encoding Byte
   [System.Convert]::ToBase64String($pkcs12cert) | Out-File ‘base12encodedcert.txt’
   ```

   These three lines of code do the following:

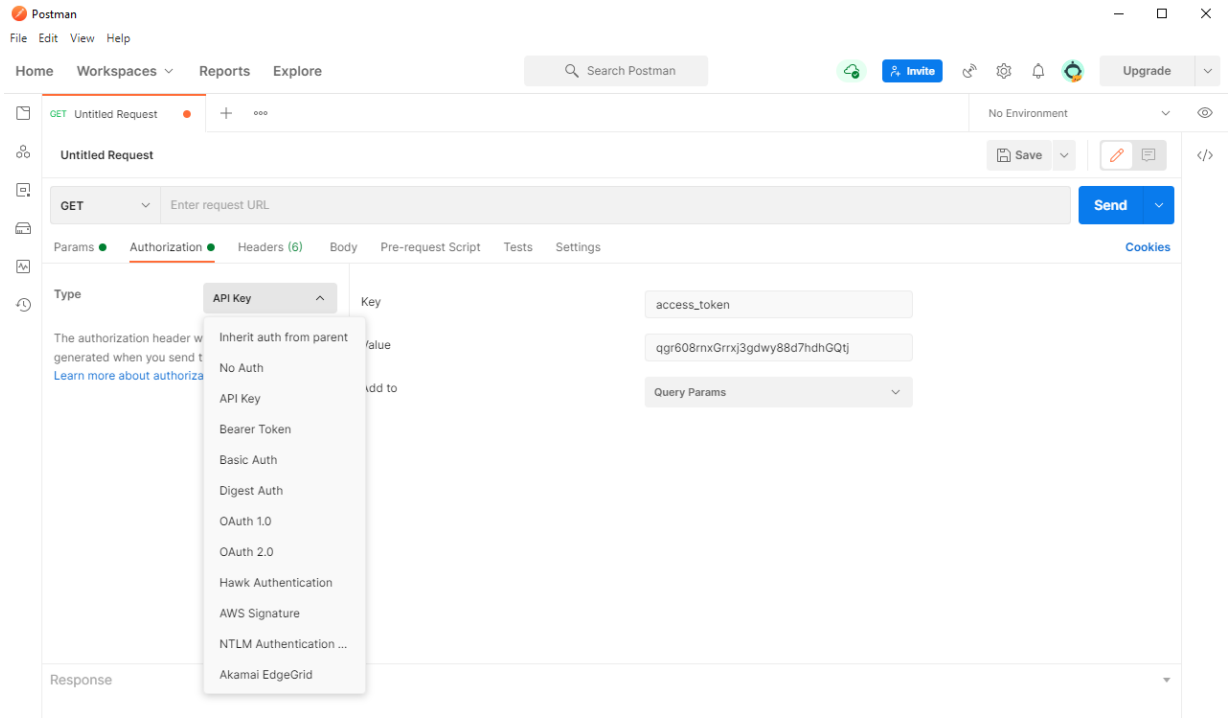
   a. Changes to working directory to the location where the encoded certificate will be created. In this example, it is the desktop.

   b. Creates a variable called `pkcs12cert` and defines it as the certificate file by specifying the full path to the certificate.

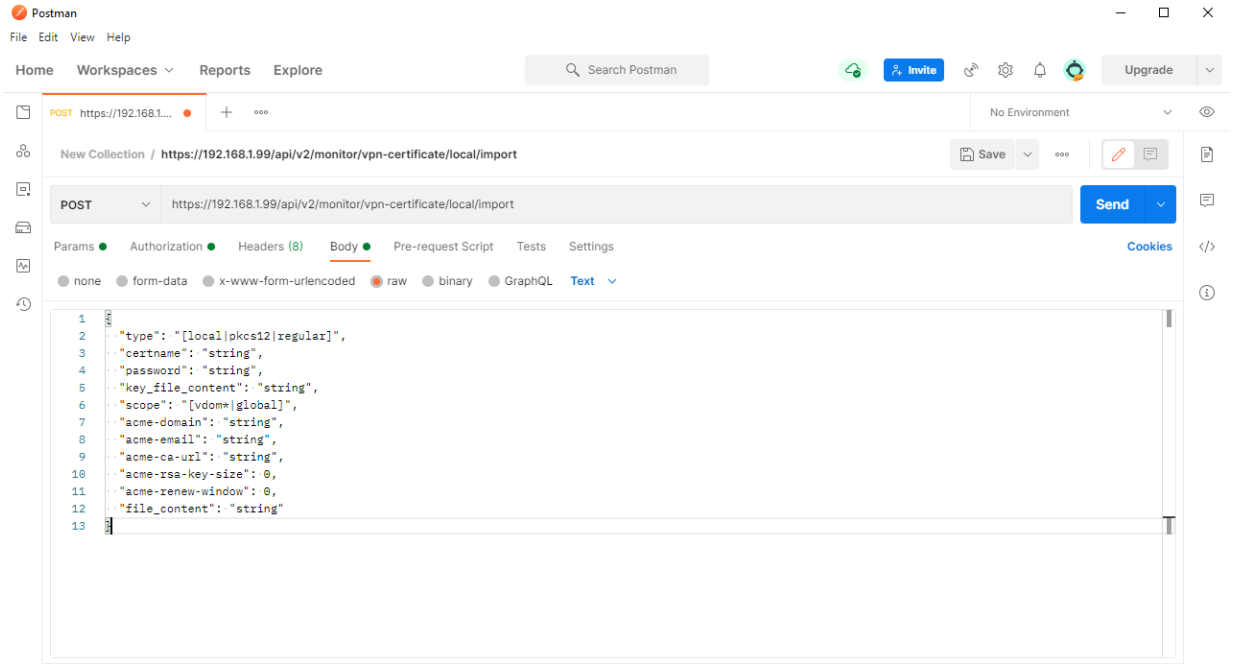
   c. Creates a text file called `base12encodedcert` at the location specified in the first step. You will copy and paste the contents of this as `file_content` later in Postman.

2. Generate an API token on the FortiGate by creating a REST API user. See Generate an API token on the Fortinet Developer Network. A subscription to the Fortinet Developer Network is required to view this topic.
3. Open Postman and create a new request:

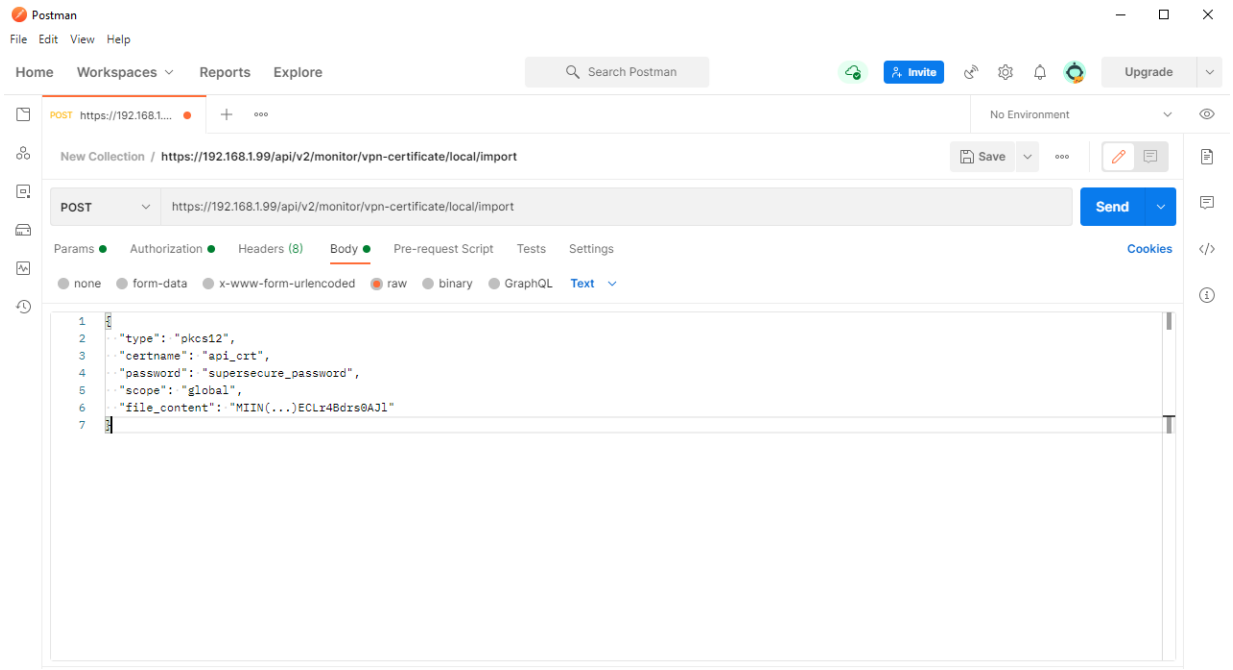
a. Click the *+*.
b. Click the *Authorization* tab and in the *Type* dropdown, select *API Key*.
c. For *Key*, enter *access_token* and enter the *Value* for the API user.
d. For *Add to*, select *Query Params*.



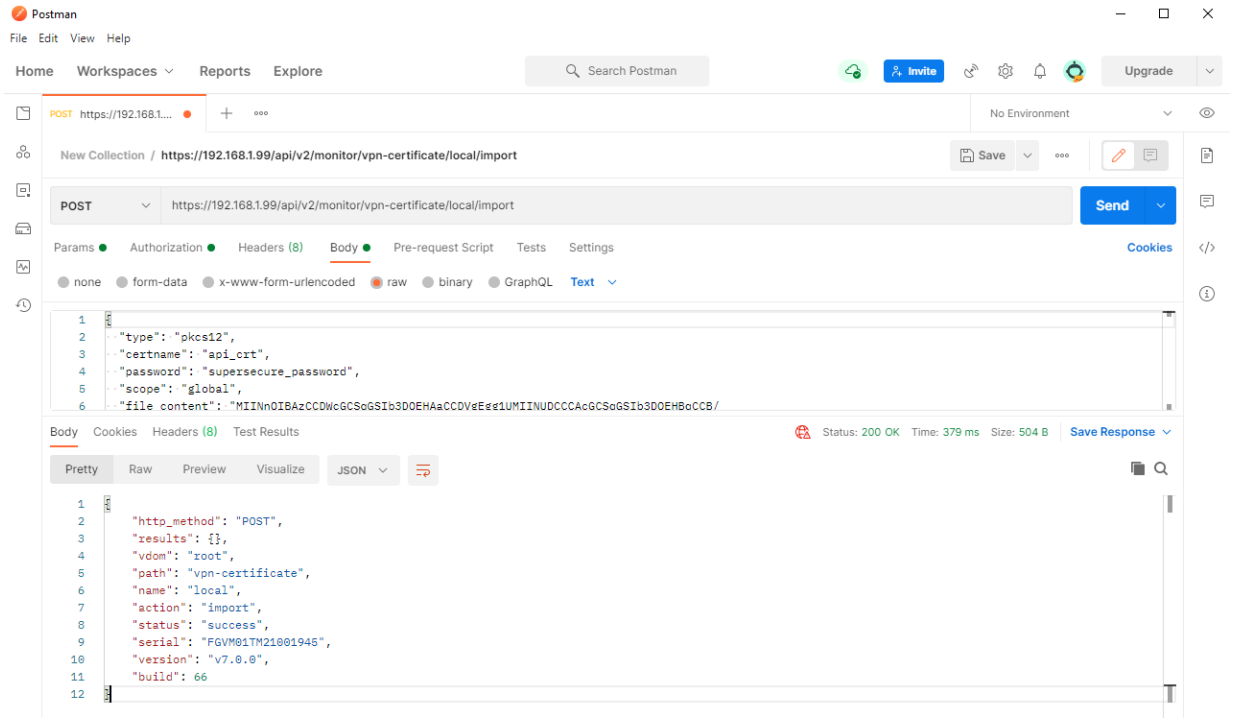
4. In the HTTP request dropdown, change the request from *GET* to *POST*, and enter the FortiGate's IP address and the URL of the API call.
5. Click the *Body* tab, and copy and paste the API parameters.



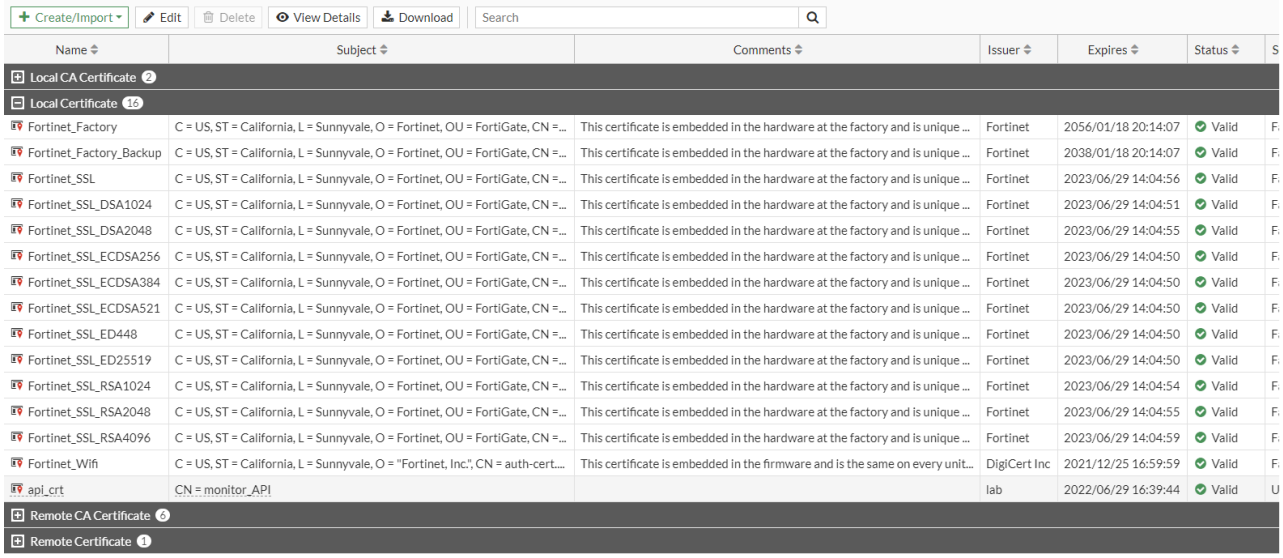
6. Remove unnecessary parameters (ACME related parameters and `key_file_content`) and enter the correct settings for your certificate. Copy and paste the contents of the file generated by PowerShell earlier into `file_content`.



7. Click *Send*. The lower window will return the results.



8. In FortiOS, go to *System* > *Certificates* and verify that the uploaded certificate is shown in the table (*api_crt*).



To debug using the HTTPS daemon:

```
# diagnose debug reset
# diagnose debug enable
# diagnose debug application httpsd -1
<output>
# diagnose debug disable
```

Procuring and importing a signed SSL certificate

A signed SSL certificate can be used when configuring Agentless VPN, for administrator GUI access, and for other functions that require a certificate.

Before creating a certificate, you must have a registered domain. With a valid FortiGuard subscription, FortiDDNS can be used to register a domain; see DDNS on page 385 for more information.

Follow these instructions to purchase, import, and use a signed SSL certificate:

- Obtain, setup, and download an SSL certificate package from a certificate authority
- Generate a CSR
- Import the signed certificate into your FortiGate
- Configure your FortiGate to use the signed certificate

Obtain, setup, and download an SSL certificate package from a certificate authority

SSL certificate packages can be purchased from any Certificate Authority (CA), such as DigiCert, GoDaddy, or GlobalSign.

Let's Encrypt can be used to generate a free, trusted SSL certificate. See Automatically provision a certificate on page 3955 for more information.

A third party CA might not sign a certificate with an intranet name or IP address. For details, see Can I request a certificate for an intranet name or IP address?

The process for purchasing, setting up, and downloading a certificate will vary depending on the CA that is used, and if a CSR must be generated on the FortiGate.

To purchase a certificate package:

1. Create an account with your chosen vendor, or use the account that you used to purchase your domain.
2. Locate the SSL Certificates page.
3. Purchase a basic SSL certificate for domain validation only. If required, a more secure SSL certificate can be purchased.
4. If required, load the CSR, either by uploaded the text file or copying and pasting the contents into the requisite text box. See Generate a CSR on page 3975 for information on generating the CSR on the FortiGate.
5. If required, set the server type to *Other*.
6. Verify the certificate per the requirements of the CA.
7. Download the signed certificate to your computer.
8. Import the signed certificate into your FortiGate; see Import the signed certificate into your FortiGate on page 3976.

Generate a CSR

Some CAs can auto-generate the CSR during the signing process, or provide tools for creating CSRs. If necessary, a CSR can be created in your FortiGate device's GUI.

To generate a CSR on your FortiGate:

1. Go to *System* > *Certificates* and select *Create/Import* > *Generate CSR*.
2. Configure the CSR:
   - Ensure that the certificate has a unique name.
   - Set the *ID Type* to *Domain Name* and enter a *Domain Name*.
   - An email address is required.
   - Ensure that the *Key Size* is set to *2048 Bit*.
   - Set the *Enrollment Method* to *File Based*.

3. Click *OK*.
   The CSR will be added to the certificate list with a status of *PENDING*.
4. In the certificate list, select the new CSR then click *Download* to save the CSR to your computer.
   The CSR file can be opened in any text editor, and will resemble the following:

```
-----BEGIN CERTIFICATE REQUEST-----
MIICuTCCAaECAQAwSzEcMBoGA1UEAxMTZm9ydGlzc2x2cG5kZW1vLmNvbTErMCkG
CSqGSIb3DQEJARYcZm9ydGlzc2x2cG5kZW1vQGZvcnRpbmV0LmNvbTCCASIwDQYJ
KoZIhvcNAQEBBQADggEPADCCAQoCggEBAMtnpNoR20NH2+UEX/NsyCmZhQqc4af3
Be1u9iOoNbo9Fk42gw47r71moAN+1jTL/Tcp3hRhXtpgoI7Zh3vjZnBbD2wwU8Ow
U7d1h5MULyMehR9r4T6OAJl4KbKPt5u90r5SpIb6mM1OIKvzMncuRS66rW1St0KP
mp/f6QjpjMrthnyJkCejgyTA1YwWNuT9BcO6PTkxBqVMLaRP6TUH6He9uhOx1Cj/
5tzvSdAozZIr2moMieQy0lNd6oQcgpdzaB9QN41+cZOlUXRCMPoH7E4KUe3/Gnis
+NMdQ8rIBijvWCXrKj20wb6sUEjAGJkcXlqVHWYCKWXl6Owejmc4ipkCAwEAAaAp
MCcGCSqGSIb3DQEJDjEaMBgwCQYDVR0TBAIwADALBgNVHQ8EBAMCBaAwDQYJKoZI
hvcNAQELBQADggEBAJKhtz2BPIKeHH9HcJKnfBKL+a6vu1l+1sW+YqnyD+3oR9ec
0eCmLnPxyyxsVel/tRsUg4DTfmooLNDhOjgfMsWxAGUQgrDH2k87cw6kiDAPCqv1
b+hFPNKZQSd09+HXAvOpXrMlrw5YdSaoRnau6Q02yUIYennKTIzFIscgh1mk4FSe
mb12DhPF+QydDCGDgtqnQbfxlDC0WmDcmxwa/0ZktoQhhhEbYgJ2O7l4TMqOxs/q
AZgwJlSNGBALLA2AxkIRUMKUteDdXz0QE8xNrvZpLTbWCNIpYJdRRqSd5C1w2VF4
CFgugTjFaJ13kYmBimeMRQsFtjLV5AxN+bUUsnQ=
-----END CERTIFICATE REQUEST-----
```

Import the signed certificate into your FortiGate

To import the signed certificate into your FortiGate:

1. Unzip the file downloaded from the CA.
   There should be two CRT files: a CA certificate with *bundle* in the file name, and a local certificate.
2. Import the local certificate:
   a. Go to *System* > *Certificates* and select *Create/Import* > *Certificate*.
   b. Click *Import Certificate*.
   c. Set *Type* to *Local Certificate*, upload the local certificate file, then click *Create*. See Local certificate on page 3962 for more information.
      The status of the certificate will change from *PENDING* to *OK*.
3. Import the CA certificate:
   a. Go to *System* > *Certificates* and select *Create/Import* > *CA Certificate*.
   b. Set the *Type* to *File*, upload the CA certificate file, then click *OK*. See CA certificate on page 3967 for more information.
      The CA certificate will be listed in the *CA Certificates* section of the certificates list.

Configure your FortiGate to use the signed certificate

After the signed certificates have been imported, you can use it when configuring Agentless VPN, for administrator GUI access, and for other functions that require a certificate.

To configure your FortiGate to use the signed certificate for Agentless VPN:

1. Go to *VPN* > *Agentless VPN Settings*.
2. Set *Server Certificate* to the new certificate.
3. Configure other settings as needed.
4. Click *Apply*.

For more information on configuring Agentless VPN, see Agentless VPN on page 3349.

To configure using the certificate for administrator GUI access in the CLI:

```
config system global
    set admin-server-cert fortisslvpndemo
end
```

To change the certificate that is used for administrator GUI access in the GUI:

1. Go to *System* > *Settings* > *Access*.
2. In the *Administration settings* section, change *HTTPS server certificate* as needed.
3. Click *Apply*. You will be logged out of FortiOS.

Microsoft CA deep packet inspection

In most production environments, you want to use a certificate issued be your own PKI for deep packet inspection (DPI). This is due to the trust relationship involved with certificates and the access (trust) they are given. As such, it is extremely important to take precautions that these certificates are not misused and in the event that they are compromised, you retain the ability to revoke their validity.

Microsoft PKI best practice utilizes a two-tier system where a Root CA signs a sub-CA (or Issuing CA, named after its role), and this Sub-CA is used to sign (sometimes referred to as *generate*) end-entity certificates, such as the certificate used by FortiGates for DPI. The purpose of this two tier system is to keep the Root CA offline so that the private key of the Root CA is better protected from being compromised.

If a sub-CA becomes compromised, it can be removed from the PKI. If the Root CA becomes compromised, the entire PKI must be replaced. Therefore, the following steps take place on a sub-CA of the Root CA for your organization.

Complete the following steps to create your own sub-CA certificate and use it for DPI:

1. Create a Microsoft sub-CA certificate
2. Export the certificate and private key
3. Import the certificate and private key into the FortiGate
4. Configure a firewall policy for DPI
5. Verify that the sub CA certificate is being used for DPI

The FortiGate firewall uses information in the original web server certificate (e.g. www.google.com), then issues a new certificate signed by the DPI certificate. The FortiGate then sends this new certificate along with the issuing DPI certificate to the client's web browser when the SSL session is being established.

The browser verifies that the web server certificate (e.g.www.google.com) was issued by a valid CA, then looks for the issuing CA of the DPI certificate in its local trusted Root-CA store to complete the path to trusted Root-CA. In this case, the issuing CA is a sub-CA of the Root-CA. The end device will need a trust chain to ensure it understands that this sub-CA is approved by the Root-CA that the device trusts.

The Root-CA certificate is normally deployed to all client PCs in the Windows domain, so the client can complete the certificate path up to a trusted root CA. Note that this is the public key of the Root-CA. The FortiGate now controls and can inspect the two HTTPS sessions: one with the external web server, and one with the client PC.

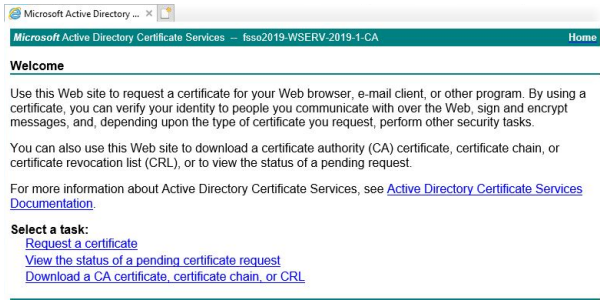
Create a Microsoft sub-CA certificate

A Microsoft sub-CA certificate can be created directly on a Microsoft sub-CA server, or remotely using a web browser.

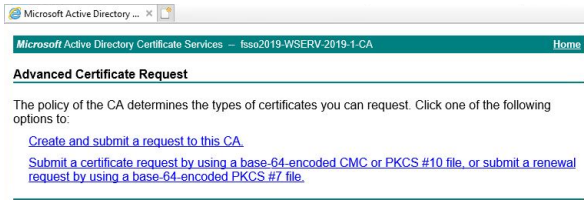
Creating a certificate remotely requires that the web enrollment option is configured on the Microsoft sub-CA server. Remote certificate requests require HTTPS; requests are not allowed with HTTP.

To create a Microsoft sub CA certificate remotely:

1. Open a web browser and go to one of the following URLs:
   - *https://<FQDN-CA-server>/CertSrv*
   - *https://<IP-CA-server>/CertSrv*
2. Log in to a domain administrator account that has web enrollment rights.

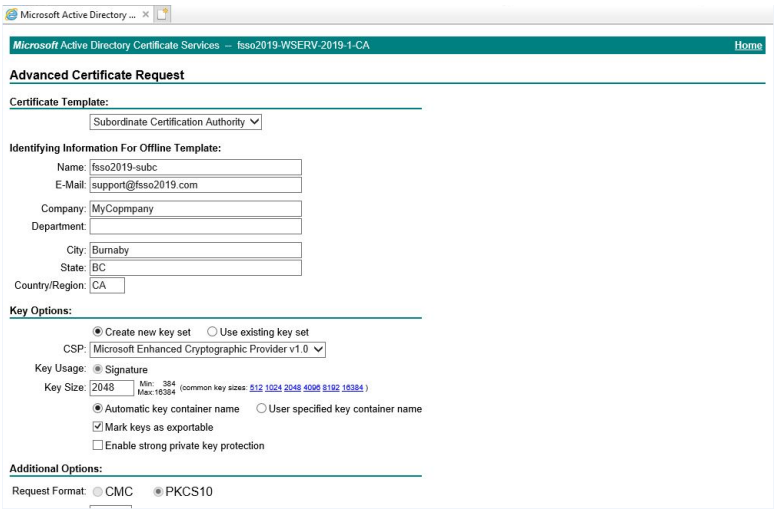


3. Click *Request a certificate*.
4. Click *advanced certificate request*.



5. Click *Create and submit a request to this CA*, then click *Yes* in the *Web Access Confirmation* warning.
6. For the *Certificate Template*, select *Subordinate Certification Authority*.
7. Enable *Mark keys as exportable*.

8. Fill out the remaining information according to your security policy.

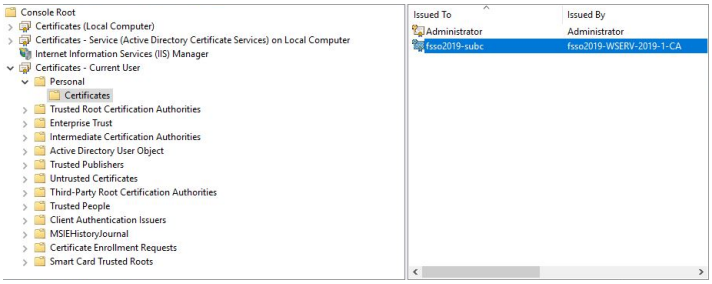


9. Submit the request.
10. Click *Yes* in the *Web Access Confirmation* warning.
11. Click *Install this certificate*.
    The certificate and private key are located in the current user's certificate store.

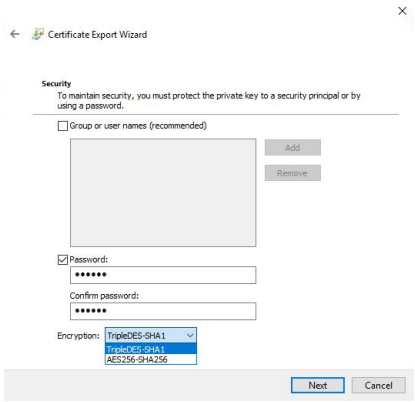
Export the certificate and private key

To export the certificate and private key:

1. Open the Microsoft Management Console (MMC) and add the *Certificate Snap-in*.
2. Go to the user's certificate store to locate the sub CA certificate that you just installed.



3. Right-click the certificate and select *All Tasks* > *Export*.
4. Click *Next* to start the *Microsoft Certificate Export Wizard*.
5. Follow the steps in the wizard:
   - When asked, select *Yes, export the private key*.
   - Only the PKCS #12 (.PFX) format is available, and it requires a password.
   - When selecting the encryption type, select *TripleDES-SHA1* if you are using an older version of FortiOS (5.6.9 and earlier). Otherwise, select *AES256-SHA256*.



6. Complete the wizard, and save the DPI certificate to a local folder.

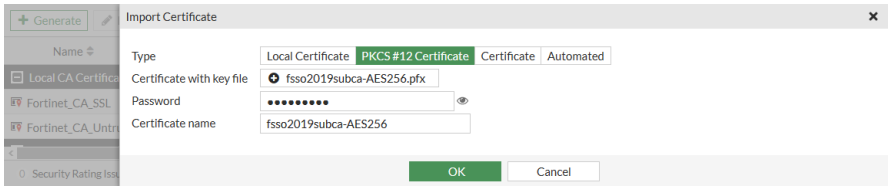
Import the certificate and private key into the FortiGate

The certificate can be imported from the local computer using the GUI, or from a TFTP server using the CLI.

After importing the certificate, you can view it in the GUI to verify that it was successfully imported.

To import the certificate and private key into the FortiGate in the GUI:

1. Go to *System* > *Certificates* and select *Create/Import* > *Certificate*.
2. Click *Import Certificate*.
3. Set *Type* to *PKCS #12 Certificate*.
4. Click *Upload*, and locate the certificate on the management computer.
5. Enter the password, then confirm the password.
6. Optionally, customize the *Certificate name*.



7. Click *OK*.

To import the certificate and private key into the FortiGate in the CLI:

```
execute vpn certificate local import tftp fsso2019subca-AES256.pfx <tftp_IP> p12 <password>
```

To verify that the certificate was imported:

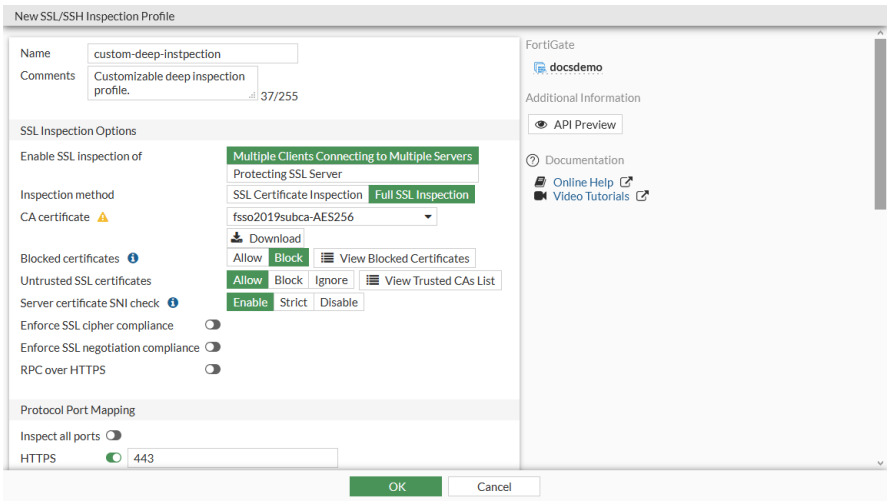
1. Go to *System* > *Certificates*.
2. Locate the newly imported certificate in the table.
3. Select the certificate and click *View Details* to view the certificate details.

Configure a firewall policy for DPI

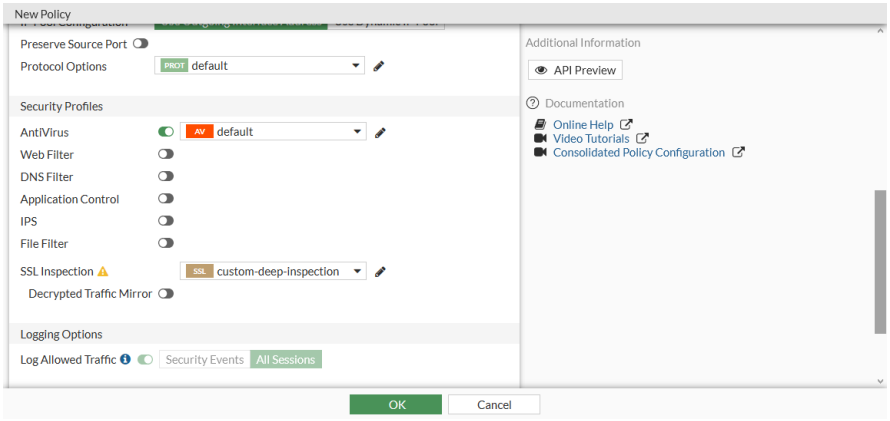
The certificate is used in an SSL/SSH inspection profile that is then used in a firewall policy.

To configure a firewall policy for DPI:

1. Go to *Security Profiles* > *SSL/SSH Inspection* and click *Create New*.
2. Configure the inspection profile, selecting the new certificate



3. Click *Apply*.
4. Go to *Policy & Objects* > *Firewall Policy*.
5. Create a new policy, or edit an existing policy.
6. In the *SSL Inspection* field, select the new SSL inspection profile.



7. Configure the remaining settings as needed.
8. Click *OK*.

Verify that the sub CA certificate is being used for DPI

You can verify that the certificate is being used for resigning web server certificates when a user connects to an external HTTPS website.

To verify that the certificate is being used:

1. On a client PC that is behind the FortiGate, go to an external HTTPS website.
   When connecting to the website, no certificate warning should be shown.
2. In your web browser, view the certificate and certificate path.
   The methods for doing this vary depending on the browser. See your browsers documentation for information.

Administrative access using certificates

Certificates can be used for administrative authentication.

Generated key pairs can also be used for this authentication. See Public key SSH access on page 3725 for information about generating a key pair.

To log in to the FortiGate with a certificate private key:

1. On the PC, generate a certificate.
2. In FortiOS, import the PEM file for the remote certificate:

```
# execute vpn certificate remote import tftp certificate.pem 172.16.200.55
```

3. Display the imported remote certificate:

```
config certificate remote
    edit "REMOTE_Cert_1"
    next
end
```

4. Apply the remote certificate to the administrative user:

```
config system admin
    edit "admin1"
        set accprofile "prof_admin"
        set vdom "root"
        set ssh-certificate "REMOTE_Cert_1"
        set password ************
    next
end
```

5. On the PC, verify that the administrator can log in to the FortiGate with the SSH certificate:

```
root@PC05:~# ssh -i certificate-private.pem admin1@172.16.200.1
FortiGate-101F $ get system status
Version: FortiGate-101F v7.0.2,build0234,211019 (GA)
```

Creating certificates with XCA

This topic explains how to generate various certificates to be used in conjunction with a FortiGate, including:

- CA certificate
  - Signing server and client certificates
  - Issuing subordinate CAs for deep inspection
- Server certificate
  - SSL/TLS web administration authentication
  - VPN authentication
  - Internal SSL server protection
- Client certificate
  - End user authentication for SSL or IPsec VPN

XCA is an x509 certificate generation tool that handles RSA, DSA, and EC keys, as well as certificate signing requests (PKCS #10) and CRLs.

There are several options for generating and managing certificates. This topic covers basic certificate generation for XCA. It is not a comprehensive guide to its application and does not explore all options available when generating a certificate.

Creating the XCA database

Before creating any certificates, you must create an XCA database to group the certificates in. You should use a different database for each PKI you create.

To create the database:

1. Go to *File* > *New Database*.
2. Select a directory to store the created certificates and keys.
3. Enter a name. The provided password encrypts the private keys and is used to access the XCA database in the future.

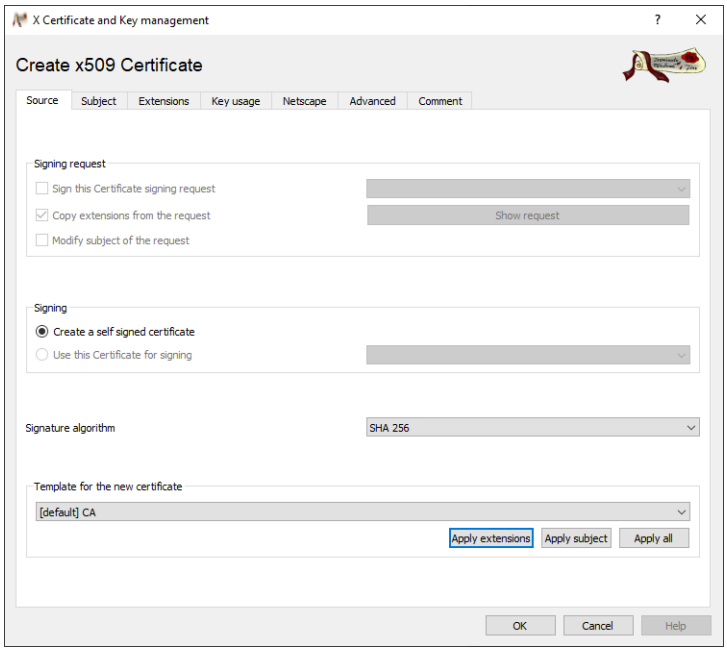
The remaining procedures in this topic assume you are using this XCA database.

Creating a CA certificate

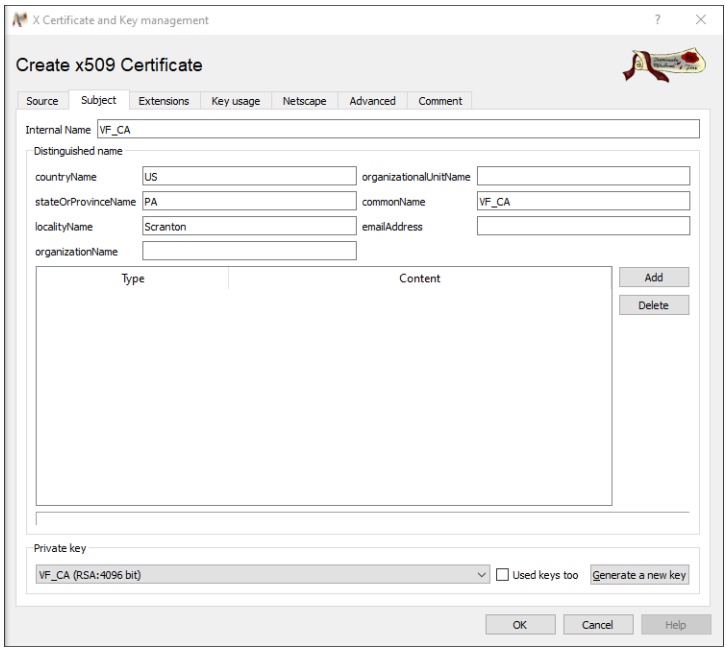
A CA certificate marks the root of a certificate chain. If this CA certificate is trusted by an end entity, any certificates signed by the CA certificate are also trusted.

To create a CA certificate:

1. Click the *Certificates* tab, then click *New Certificate*.
2. Edit the *Source* tab:
   a. Set *Template for the new certificate* to *[default] CA*.
   b. Click *Apply extensions*.



3. Edit the *Subject* tab:
   a. Enter an *Internal Name* to reference this certificate within XCA.
   b. Enter a *commonName*.
   c. Optionally, click *Add* to add other distinguished name fields.
   d. Since this XCA database does not contain any keys yet, click *Generate a new key*. The *Private key* field is now populated.



4. Optionally, edit the *Extensions* tab:
   a. Adjust the *Time range* if needed.
   b. Click *Apply*.
5. Click *OK*.

Issuing a subordinate CA certificate for deep inspection

Subordinate CA certificates are similar to CA certificates because they are used to sign other certificates to establish trust of the signed certificate's content. This trust of the signed certificate is only valid if the subordinate CA is also trusted by the client.

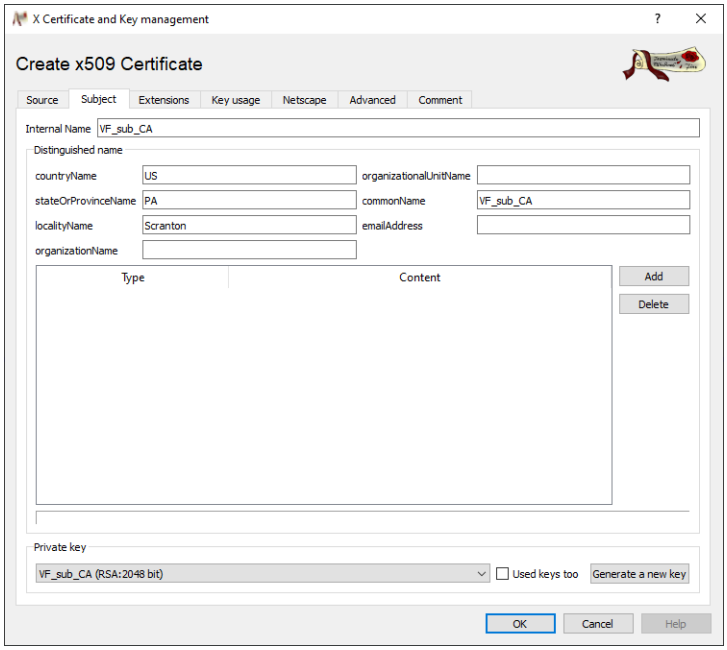
When performing deep inspection on a FortiGate, the FortiGate proxies the connection between the endpoint and the server. This is done transparently so that the end user believes they are communicating with the server, and the server with the client. To do this, when the webpage is requested by a client, the FortiGate must present a certificate that matches the requested website and is trusted by the client.

The certificate presented by the FortiGate is generated on-demand to match the requested website and is signed by this subordinate CA to establish trust with the requesting endpoint. The subordinate CA must be installed on the ForitGate (with the private key) and on the client device (without the private key).

A subordinate CA is used in place of a CA so that it may be revoked as necessary. This is critical since the subordinate CA's private key is exported and becomes susceptible of being compromised. If the CA private key becomes compromised, you would be forced to re-create your entire PKI with a new root CA because root CAs cannot be revoked. See Microsoft CA deep packet inspection on page 3977 for more information about using subordinate CA certificates.

To issue a subordinate CA certificate for deep inspection:

1. Click the *Certificates* tab, then click *New Certificate*.
2. Edit the *Source* tab:
   a. Set *Use this Certificate for signing* to the CA created previously.
   b. Set *Template for the new certificate* to *[default] CA*.
   c. Click *Apply extensions*.
3. Edit the *Subject* tab:
   a. Enter an *Internal Name* to reference this certificate within XCA.
   b. Enter a *commonName*.
   c. Optionally, click *Add* to add other distinguished name fields.
   d. Click *Generate a new key* to create a new private key for the subordinate CA.



4. Optionally, edit the *Extensions* tab:
   a. Adjust the *Time range* if needed.
   b. Click *Apply*.
5. Click *OK*.

Creating a server host certificate

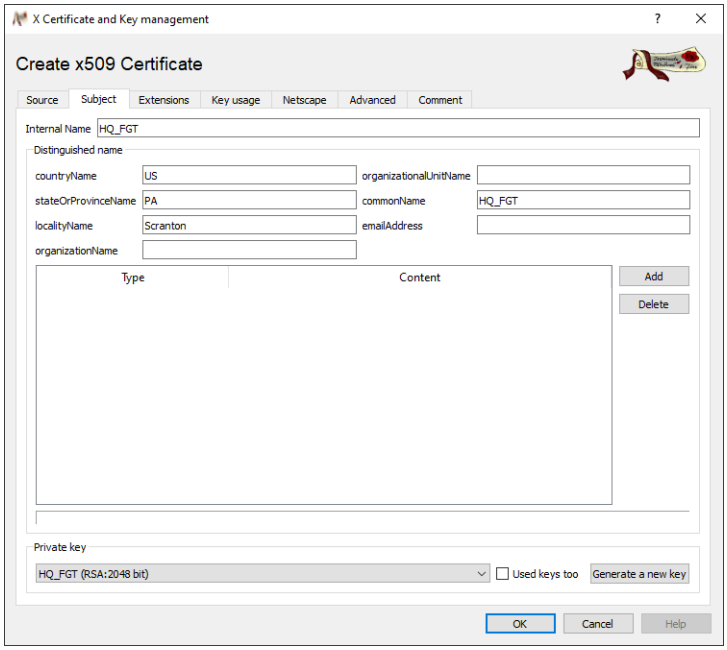
When a CA signs a host certificate, that CA is vouching for the credentials in the certificate. These credentials are what identifies the host.

Some endpoints can generate a certificate signing request (CSR). A CSR is a certificate outline that specifies the details of the endpoint, including its public key. This allows the CA to review the details and sign the request if they are true. This request is then returned or uploaded to the generating endpoint to be used.

Since some endpoints cannot generate their own CSR, you can create the certificate manually in XCA. If you already have a CSR, use the *Certificate signing requests* tab to import and then sign it.

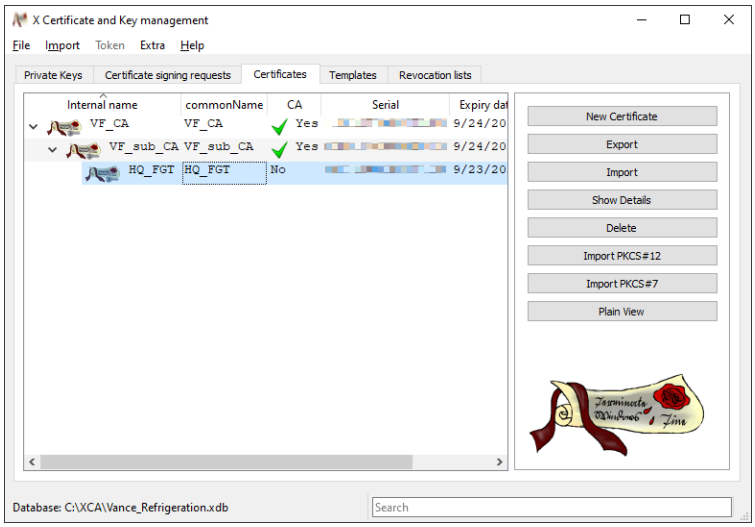
To create a server host certificate:

1. Click the *Certificates* tab, then click *New Certificate*.
2. Edit the *Source* tab:
   a. Set *Template for the new certificate* to *[default] TLS_server*.
   b. Click *Apply extensions*.
   c. In the *Signing* section, select *Use this Certificate for signing* and select the subordinate CA certificate.
3. Edit the *Subject* tab:
   a. Enter an *Internal Name* to reference this certificate within XCA.
   b. Enter the distinguished name fields as needed.
   c. Click *Generate a new key*.



4. Edit the *Extensions* tab:
   a. For *X509v3 Subject Alternative Name*, enter *email:user@domain.tld*.
5. Click *OK*.

6. Click the *Certificates* tab to view the certificate.



This certificate may be used to identify an SSL or TLS server by uploading the certificate and key pair to the server, such as when the FortiGate presents the administrative webpage or for Agentless VPN authentication (see Configure your FortiGate to use the signed certificate on page 3976). Another use case for a server host certificate is to enable SSL server protection so the FortiGate simulates the real server and brokers the connection (see Protecting an SSL server on page 2700).

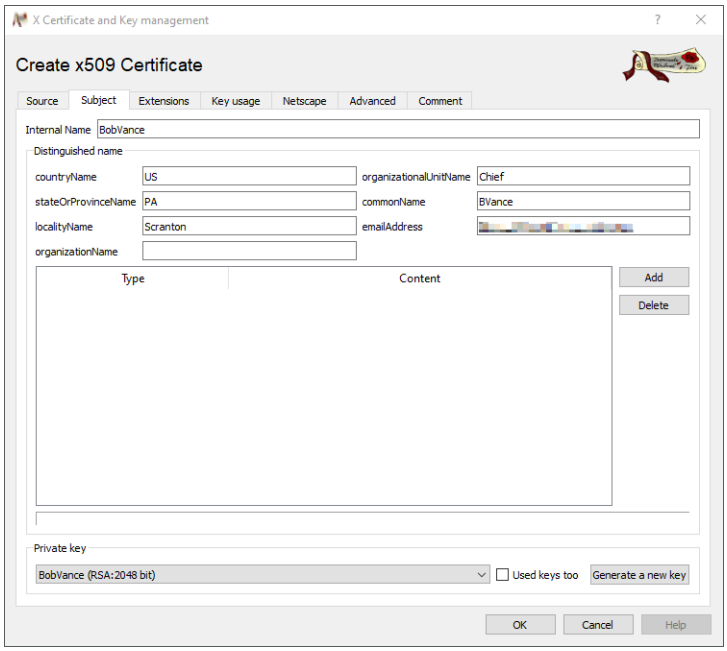
Creating a client host certificate

A client host certificate is used to identify an end entity in a more secure way than a username and password. Once the client host certificate is generated, see Agentless VPN with certificate authentication on page 3392 for more information about using the certificate.

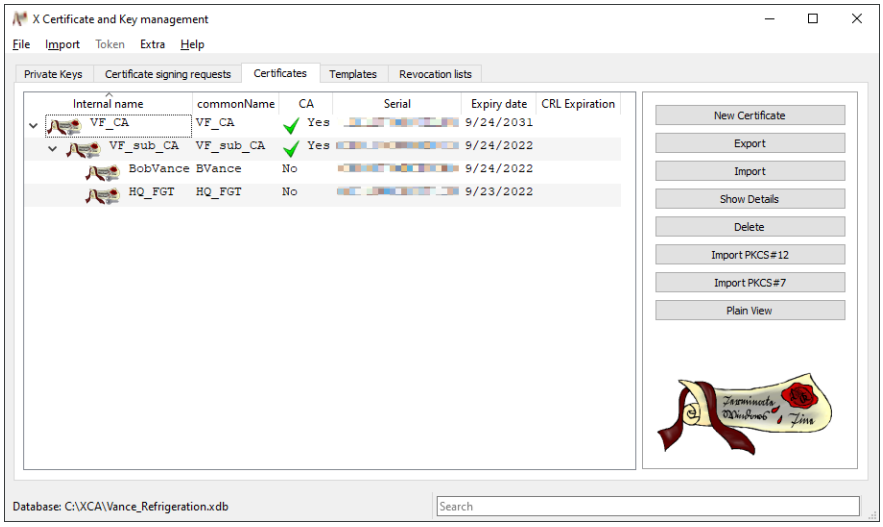
To create a client host certificate:

1. Click the *Certificates* tab, then click *New Certificate*.
2. Edit the *Source* tab:
   a. In the *Signing* section, select *Use this Certificate for signing* and select the CA or subordinate CA.
   b. Set *Template for the new certificate* to *[default] TLS_client*.
   c. Click *Apply extensions*.
3. Edit the *Subject* tab:
   a. Enter an *Internal Name* to reference this certificate within XCA.
   b. Enter the distinguished name fields as needed.

c. Click *Generate a new key*.



4. Click *OK*.
5. Click the *Certificates* tab. The FortiGate and client certificates are listed under the signing CA certificate and are ready to be exported.



6. Select a certificate and click *Export*.
7. Enter the file name and select an export format.
8. Click *OK*.

Certificate formats

Certificate file formats indicate what is contained in the file, how it is formatted, and how it is encoded. See Import a certificate on page 3962 for more information about which formats the FortiGate expects for a given certificate type.

Enrollment over Secure Transport for automatic certificate management

The FortiGate supports Enrollment over Secure Transport (EST) and the RFC 7030 standards when generating a new CSR request, performing automatic renewals, or manually regenerating a certificate. EST provides more security for automatic certificate management than Simple Certificate Enrollment Protocol (SCEP), which is commonly used for certificate enrollment.

Background

SCEP helps automate and simplify the process for obtaining a digital certificate from a certificate authority (CA). However, SCEP does not natively support secure connections, and instead relies on the underlying transport protocol to provide security. EST was developed, which uses TLS to establish a secure communication channel over which subsequent certificate management protocol messages like initial certificate enroll and certificate renewal messages are exchanged.

On the FortiGate, when generating a certificate signing request (CSR), you can use the SCEP method to send the request to an SCEP server, or use EST to send the request to an EST server to be signed by a CA.

To configure the enrollment protocol settings for a local certificate:

```
config vpn certificate local
    edit <name>
        set enroll-protocol est
        set est-server <string>
        set est-ca-id <string>
        set est-http-username <string>
        set est-http-password <string>
        set est-client-cert <certificate>
        set est-server-cert <certificate>
        set est-srp-username <string>
        set est-srp-password <string>
    next
end
```

`est-server <string>`	Enter the address and port for EST server (such as https://example.com:1234).
`est-ca-id <string>`	Enter the CA identifier of the CA server for signing with EST.
`est-http-username <string>`	Enter the HTTP Authentication username for signing with EST.
`est-http-password <string>`	Enter the HTTP Authentication password for signing with EST.
`est-client-cert <certificate>`	Enter the certificate used to authenticate this FortiGate to the EST server.
`est-server-cert <certificate>`	Enter the EST server's certificate that has to be verifiable by the specified certificate on the FortiGate.

`est-srp-username <string>`	Enter the EST SRP authentication username.
`est-srp-password <string>`	Enter the EST SRP authentication password.

To manually generate a CSR for the EST server to be signed by a CA:

```
# execute vpn certificate local generate est {reqired_1} {reqired_2} {reqired_3} [options]
```

option 1 (required)	Name of the local server certificate.
option 2 (required)	Cryptography algorithm: `rsa-1024`, `rsa-1536`, `rsa-2048`, `rsa-4096`, `ec-secp256r1`, `ec-secp384r1`, or `ec-secp521r1`.
option 3 (required)	URL and listening port of the remote EST responder.
option 4 (optional)	Server certificate subject in the certificate enroll request. Separate fields by a comma (,).
option 5 (optional)	Subject Alternative Name (SAN). This can be an FQDN and/or IP. Use DNS:<FQDN>,IP:<IP_address> for example. If the issuing CA does not support SAN, this option will be ignored. Separate fields by a comma (,).
option 6 (optional)	HTTP authentication username.
option 7 (optional)	HTTP authentication password.
option 8 (optional)	CA identifier.
option 9 (optional)	CA certificate used to verify the remote EST responder server certificate and certificates issued by a remote PKI.
option 10 (optional)	Password for the private key.
option 11 (optional)	Client certificate.
option 12 (optional)	Source IP for communications to the CA server.
option 13 (optional)	TLS-SRP username.
option 14 (optional)	TLS-SRP password.

Example 1: enrolling for a new FortiGate server certificate with EST

To enroll for a new FortiGate server certificate with EST:

1. Verify that the FortiGate can communicate with remote EST responder (testrfc7030.com):

```
# execute ping testrfc7030.com
PING testrfc7030.com (54.70.32.33): 56 data bytes
64 bytes from 54.70.32.33: icmp_seq=0 ttl=31 time=13.6 ms
64 bytes from 54.70.32.33: icmp_seq=1 ttl=31 time=19.1 ms
64 bytes from 54.70.32.33: icmp_seq=2 ttl=31 time=16.5 ms
^C
--- testrfc7030.com ping statistics ---
```

```
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 13.6/16.4/19.1 ms
```

2. Start running debugs to track the progress of the enrollment:

```
# diagnose debug application est -1
# diagnose debug enable
```

3. Create a new server CSR file locally and send it to the remote EST responder:

```
# execute vpn certificate local generate est est-test101 ec-secp256r1
https://testrfc7030.com:8443 CN=firewall-portal1,DC=local,DC=COM DNS:firewall-
portal1.local.ca,IP:172.18.60.184 estuser estpwd G_CA_Cert_1
```

The CA certificate (G_CA_Cert_1) is used to verify the remote EST responder server certificate and certificates issued by a remote PKI.

testrfc7030.com is a self-signed CA, which by default is not in the local trusted root store and must be imported prior to enrollment.

If the CA that issues the server certificate is not in the local root store, an error would appear in the debug messages:

```
# diagnose debug application est -1
# diagnose debug enable
...
[1795] est_curl_req: Error buf: SSL certificate problem: self-signed certificate in
certificate chain,
[2402] est_simple_enroll: Failed to get ca certs: -1.
...
```

4. If the enrollment was successful, in a few seconds, a `Done` message appears. Verify the debugs to view the enrollment process.
   a. The remote CA's certificate is retrieved and stored locally in the EST configuration after being verified with the CA in the trusted root store:

```
[1962] __est_curl_set_auth: trace
[2046] __est_curl_set_auth: HTTP Authentication username is set
[2050] __est_curl_set_auth: HTTP Authentication password is set
[2075] __est_get_ca_certs: ============STARTED============
[1728] est_curl_req: URL: https://testrfc7030.com:8443/.well-known/est/cacerts
[1776] est_curl_req: HTTP GET
[143] __curl_ssl_ctx_finalizer: global CAs are loaded.
[165] __curl_ssl_ctx_finalizer: SSL_CTX ex data is set.
[1651] curl_header_debug_func: Header received:HTTP/1.1 200 OK
```

   b. The debug displays the CA used by the remote EST responder:

```
[1191] save_pkcs7_certs: Saving pkcs7 response
[505] est_print_pkcs7: Certs: (1 in total)
[507] est_print_pkcs7: Cert 1:
[427] est_print_x509:         Version: 3 (0x2)
```

```
        Serial Number:
            ab:e8:32:e1:f6:6a:6b:43
        Issuer: CN=estExampleCA
        Subject: CN=estExampleCA
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:TRUE
            X509v3 Subject Key Identifier:
                1A:DF:39:84:C2:56:E6:6C:CF:2A:B4:26:A5:FD:0C:D2:43:F5:3D:3E

[1220] save_pkcs7_certs: Received 1 certs
[1228] save_pkcs7_certs: Saving cert(s):
        is_global:1
        est_url:https://testrfc7030.com:8443
        source_ip:NULL
        ca_identifier:NULL
```

c. The CA certificate is imported. FortiOS sends a query to learn about the attributes supported by the CA in the certificate request and will then create the CSR accordingly:

```
[1288] save_pkcs7_certs: CA certs imported!
[2101] __est_get_csr_attrs: ============STARTED============
[1728] est_curl_req: URL: https://testrfc7030.com:8443/.well-known/est/csrattrs
[1776] est_curl_req: HTTP GET
[1651] curl_header_debug_func: Header received:HTTP/1.1 200 OK
[1651] curl_header_debug_func: Header received:Status: 200 OK
[1651] curl_header_debug_func: Header received:Content-Type: application/csrattrs
[1651] curl_header_debug_func: Header received:Content-Transfer-Encoding: base64
[1651] curl_header_debug_func: Header received:Content-Length: 57
[1651] curl_header_debug_func: Header received:
[1787] est_curl_req: Response 200
[1788] est_curl_req: Buffer:MCYGBysGAQEBARYGCSqGSIb3DQEJAQYFK4EEACIGCWCGSAFlAwQCAg==
[1439] decode_csrattrs_callback: Decoding csrattrs, resp->len: 57
[1474] decode_csrattrs_callback: Object: 1.3.6.1.1.1.1.22 undefined
[1474] decode_csrattrs_callback: Object: 1.2.840.113549.1.9.1 emailAddress
[1474] decode_csrattrs_callback: Object: 1.3.132.0.34 secp384r1
[1474] decode_csrattrs_callback: Object: 2.16.840.1.101.3.4.2.2 sha384
```

d. The CSR information is generated, which is sent to the remote EST responder:

```
        est_ctx: is_global:1
        vfid:0
        svr_original_url:https://testrfc7030.com:8443
        svr_hostinfo:Exists
        ca_identifier:(null)
        http_username:estuser
        http_password:*
        clt_cert:(null)
        svr_cert:(null)
        srp_username:(null)
        srp_password:(null)
        source_ip:(null)
        need_pop:0
```

```
        newcert_name:est-test101
        passwd:(null)
        rsa_keysize:0
        ec_curvename:secp256r1
        subject:CN=firewall-portal1,DC=local,DC=COM
        sub_alt_name:DNS:firewall-portal1.local.ca,IP:172.18.60.184
        svr_cert_x509:NULL
        csr_attrs:Exists
        csr:NULL
        pkey:NULL
        header_ptr:NULL
        tmp_p10:NULL
[2259] __est_simple_enroll: ============STARTED============
```

e. The CSR is sent to the EST responder:

```
[1728] est_curl_req: URL: https://testrfc7030.com:8443/.well-known/est/simpleenroll
[1753] est_curl_req: HTTP POST
[1651] curl_header_debug_func: Header received:HTTP/1.1 200 OK
[1651] curl_header_debug_func: Header received:Status: 200 OK
[1651] curl_header_debug_func: Header received:Content-Type: application/pkcs7-mime;
smime-type=certs-only
[1651] curl_header_debug_func: Header received:Content-Transfer-Encoding: base64
[1651] curl_header_debug_func: Header received:Content-Length: 585
[1651] curl_header_debug_func: Header received:
```

f. The CA issues the certificate and sends it back in a PKCS #7 structure:

```
[1787] est_curl_req: Response 200
[1788] est_curl_req:
Buffer:MIIBqwYJKoZIhvcNAQcCoIIBnDCCAZgCAQExADALBgkqhkiG9w0BBwGgggGAMIIB
fDCCASOgAwIBAgIDB0aXMAoGCCqGSM49BAMCMBcxFTATBgNVBAMTDGVzdEV4YW1w
...
```

g. The FortiGate decodes and displays the attributes of the certificate, then saves the certificate:

```
[1191] save_pkcs7_certs: Saving pkcs7 response
[505] est_print_pkcs7: Certs: (1 in total)
[507] est_print_pkcs7: Cert 1:
[427] est_print_x509:         Version: 3 (0x2)
        Serial Number: 476823 (0x74697)
        Issuer: CN=estExampleCA
        Subject: CN=firewall-portal1
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Key Usage:
                Digital Signature
            X509v3 Subject Key Identifier:
                9B:F8:39:D5:21:E6:FF:49:FF:AC:02:57:5B:FC:4C:1A:8B:1E:5D:8F
            X509v3 Authority Key Identifier:
                1A:DF:39:84:C2:56:E6:6C:CF:2A:B4:26:A5:FD:0C:D2:43:F5:3D:3E
```

```
[1220] save_pkcs7_certs: Received 1 certs
[1228] save_pkcs7_certs: Saving cert(s):
        is_global:1
        est_url:https://testrfc7030.com:8443
        source_ip:NULL
        ca_identifier:NULL

[1246] save_pkcs7_certs: Received 1 cert(s)
[427] est_print_x509:         Version: 3 (0x2)
        Serial Number: 476823 (0x74697)
        Issuer: CN=estExampleCA
        Subject: CN=firewall-portal1
        X509v3 extensions:
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Key Usage:
                Digital Signature
            X509v3 Subject Key Identifier:
                9B:F8:39:D5:21:E6:FF:49:FF:AC:02:57:5B:FC:4C:1A:8B:1E:5D:8F
            X509v3 Authority Key Identifier:
                1A:DF:39:84:C2:56:E6:6C:CF:2A:B4:26:A5:FD:0C:D2:43:F5:3D:3E

[827] est_cmdb_update_cert: Cert est-test101 updated in CMDB
[1276] save_pkcs7_certs: The cert is saved!
[592] est_ctx_clear_tmp_data: trace
[2408] est_simple_enroll: POST ret:0
[592] est_ctx_clear_tmp_data: trace
Done.
```

Example 2: automatically renewing a FortiGate server certificate with EST

When the time for certificate renewal is up, the FortiGate will use the existing EST parameters to perform an automatic renewal. This example demonstrates the renewal process through debugs.

To automatically renew a FortiGate server certificate with EST:

1. Verify the current local certificate configuration:

```
config vpn certificate local
(local) # get est-test101
name                : est-test101
password            : *
comments            :
private-key         : *
certificate         :
        Subject:     CN = firewall-portal1
        Issuer:      CN = estExampleCA
        Valid from:  2023-04-06 22:37:34  GMT
        Valid to:    2024-04-05 22:37:34  GMT
```

```
        Fingerprint: AE:67:11:CF:7D:F9:57:A4:09:8B:55:0A:F1:B1:7A:CF
...
state               : OK
range               : global
source              : user
source-ip           : 0.0.0.0
ike-localid-type    : asn1dn
enroll-protocol     : est
est-server          : https://testrfc7030.com:8443
est-ca-id           :
est-http-username   : estuser
est-http-password   : *
est-client-cert     :
est-server-cert     :
est-srp-username    :
est-srp-password    :
auto-regenerate-days: 0
auto-regenerate-days-warning: 0
```

Note that the current `Valid to` date and time is `2024-04-05 22:37:34 GMT`, which is one year from the issue date.

2. Start running debugs to track the progress of the renewal:

```
# diagnose debug application est -1
# diagnose debug enable
```

3. For demonstration purposes, update the `auto-regenerate-days` setting to 364 days to trigger the automatic renewal on the FortiGate:

```
config vpn certificate local
    edit est-test101
        set auto-regenerate-days 364
    next
end
```

4. Verify the debugs to confirm that the certificate was renewed.
   a. The FortiGate uses the content of the current certificate to create a new CSR. User credentials used for the initial enrollment are stored in local certificate configuration, but they are not used for renewal:

```
[1024] reconstruct_est_ctx: Reconstruction succeeded
est_ctx:        is_global:1
        vfid:0
        svr_original_url:https://testrfc7030.com:8443
        svr_hostinfo:NULL
        ca_identifier:
        http_username:estuser
        http_password:*
        clt_cert:
        svr_cert:
        srp_username:
        srp_password:
```

```
        source_ip:(null)
        need_pop:0
        newcert_name:est-test101
        passwd:f51da8548af5fef820edfe6267b0c178e76f7c3eae40ee0900318fc77ab6bd
        rsa_keysize:0
        ec_curvename:(null)
        subject:(null)
        sub_alt_name:(null)
        svr_cert_x509:NULL
        csr_attrs:NULL
        csr:NULL
        pkey:NULL
        header_ptr:NULL
        tmp_p10:NULL
```

b. The FortiGate sends the current server certificate for authentication/authorization and not the username/password used for initial enrollment:

```
[2453] est_simple_reenroll: Try to use est-test101 as client cert to authenticate
[1962] __est_curl_set_auth: trace
[2011] __est_curl_set_auth: Warning: cert est-test101 may not have the correct key usage
for TLS client authentication
[2014] __est_curl_set_auth: Will use cert est-test101 to prove my identity
...
[1651] curl_header_debug_func: Header received:
[1787] est_curl_req: Response 200
[1788] est_curl_req: Buffer:MCYGBysGAQEBARYGCSqGSIb3DQEJAQYFK4EEACIGCWCGSAFlAwQCAg==
[1439] decode_csrattrs_callback: Decoding csrattrs, resp->len: 57
[1474] decode_csrattrs_callback: Object: 1.3.6.1.1.1.1.22 undefined
[1474] decode_csrattrs_callback: Object: 1.2.840.113549.1.9.1 emailAddress
[1474] decode_csrattrs_callback: Object: 1.3.132.0.34 secp384r1
[1474] decode_csrattrs_callback: Object: 2.16.840.1.101.3.4.2.2 sha384
est_ctx:        is_global:1
        vfid:0
        svr_original_url:https://testrfc7030.com:8443
        svr_hostinfo:Exists
        ca_identifier:
        http_username:estuser
        http_password:*
        clt_cert:est-test101
        svr_cert:
        srp_username:
        srp_password:
        source_ip:(null)
        need_pop:0
        newcert_name:est-test101
        passwd:f51da8548af5fef820edfe6267b0c178e76f7c3eae40ee0900318fc77ab6bd
        rsa_keysize:0
        ec_curvename:(null)
        subject:(null)
        sub_alt_name:(null)
        svr_cert_x509:NULL
```

```
        csr_attrs:Exists
        csr:NULL
        pkey:NULL
        header_ptr:NULL
        tmp_p10:NULL
[2274] __est_simple_reenroll: ============STARTED============
```

c. The CSR for renewal is successfully generated:

```
[965] est_generate_csr_from_cert: Successfully generated CSR for est-test101
[2200] __est_simple_post: Data to be posted:
|||MIIBQDCB5gIBAjAbMRkwFwYDVQQDDBBmaXJld2FsbC1wb3J0YWwxMFkwEwYHKoZI
zj0CAQYIKoZIzj0DAQcDQgAEQoJQmPedxPNUcfCyRvpqyt1oiiJX/me+TdButUSu
8hg+9nPF6+xNf+5LmtG/YKHeXyCKG6xB9OmJf255Zmx+5qBpMGcGCSqGSIb3DQEJ
DjFaMFgwCQYDVR0TBAIwADALBgNVHQ8EBAMCB4AwHQYDVR0OBBYEFJv4OdUh5v9J
/6wCV1v8TBqLHl2PMB8GA1UdIwQYMBaAFBrfOYTCVuZszyq0JqX9DNJD9T0+MAoG
CCqGSM49BAMCA0kAMEYCIQCK3Li51F7fXsyKZwtIcYMFvDobY3cKKTTDixtN7QZ2
jwIhAKUkqfWPAzwcxQaNQw6pyYvo18ymB9aEheeIXZfGI+tV
|||

[1728] est_curl_req: URL: https://testrfc7030.com:8443/.well-known/est/simplereenroll
[1753] est_curl_req: HTTP POST
[1651] curl_header_debug_func: Header received:HTTP/1.1 200 OK
[1651] curl_header_debug_func: Header received:Status: 200 OK
[1651] curl_header_debug_func: Header received:Content-Type: application/pkcs7-mime;
smime-type=certs-only
[1651] curl_header_debug_func: Header received:Content-Transfer-Encoding: base64
[1651] curl_header_debug_func: Header received:Content-Length: 590
[1651] curl_header_debug_func: Header received:
[1787] est_curl_req: Response 200
```

d. The new certificate is received in PKCS #7 and is saved:

```
[1788] est_curl_req:
Buffer:MIIBrQYJKoZIhvcNAQcCoIIBnjCCAZoCAQExADALBgkqhkiG9w0BBwGgggGCMIIB
fjCCASOgAwIBAgIDB0aYMAoGCCqGSM49BAMCMBcxFTATBgNVBAMTDGVzdEV4YW1w
...
[1191] save_pkcs7_certs: Saving pkcs7 response
[505] est_print_pkcs7: Certs: (1 in total)
...
[1220] save_pkcs7_certs: Received 1 certs
[1228] save_pkcs7_certs: Saving cert(s):
        is_global:1
        est_url:https://testrfc7030.com:8443
        source_ip:NULL
        ca_identifier:

[1246] save_pkcs7_certs: Received 1 cert(s)
[427] est_print_x509:         Version: 3 (0x2)
        Serial Number: 476824 (0x74698)
        Issuer: CN=estExampleCA
        Subject: CN=firewall-portal1
        X509v3 extensions:
```

```
            X509v3 Basic Constraints:
                CA:FALSE
            X509v3 Key Usage:
                Digital Signature
            X509v3 Subject Key Identifier:
                9B:F8:39:D5:21:E6:FF:49:FF:AC:02:57:5B:FC:4C:1A:8B:1E:5D:8F
            X509v3 Authority Key Identifier:
                1A:DF:39:84:C2:56:E6:6C:CF:2A:B4:26:A5:FD:0C:D2:43:F5:3D:3E

[827] est_cmdb_update_cert: Cert est-test101 updated in CMDB
[1276] save_pkcs7_certs: The cert is saved!
[592] est_ctx_clear_tmp_data: trace
[2477] est_simple_reenroll: POST ret:0
[592] est_ctx_clear_tmp_data: trace
```

5. Verify the renewed local certificate configuration:

```
config vpn certificate local
(local) # get est-test101
name                : est-test101
password            : *
comments            :
private-key         : *
certificate         :
        Subject:     CN = firewall-portal1
        Issuer:      CN = estExampleCA
        Valid from:  2023-04-06 22:55:09  GMT
        Valid to:    2024-04-05 22:55:09  GMT
        Fingerprint: D9:51:6C:EF:04:E9:79:8D:A0:EE:10:23:4A:F4:46:B7
        Root CA:     No
        Version:     3
        Serial Num:
                07:46:a5
        Extensions:
                Name:     X509v3 Basic Constraints
                Critical: no
                Content:
```

Note that the `Valid to` date and time is now `2024-04-05 22:55:09 GMT`.

Example 3: manually regenerating a local certificate with EST

Note that manually regenerating the certificate will not generate a new server key pair.

To manually regenerate a local certificate with EST:

1. Run the following command:

```
# execute vpn certificate local generate est est-test101
Certificate 'est-test101' already exists, re-generate will ignore all the options you have
provided.
```

```
Are you sure to re-generate the certificate?
Do you want to continue? (y/n) y
```

2. Verify the debugs to confirm that the certificate was generated:

```
# diagnose debug application est -1
# diagnose debug enable
...
[1024] reconstruct_est_ctx: Reconstruction succeeded
est_ctx:        is_global:1
        vfid:0
        svr_original_url:https://testrfc7030.com:8443
        svr_hostinfo:NULL
        ca_identifier:
        http_username:estuser
        http_password:*
        clt_cert:
        svr_cert:
        srp_username:
        srp_password:
        source_ip:(null)
        need_pop:0
        newcert_name:est-test101
        passwd:f51da8548af5fef820edfe6267b0c178e76f7c3eae40ee0900318fc77ab6bd
        rsa_keysize:0
        ec_curvename:(null)
        subject:(null)
        sub_alt_name:(null)
        svr_cert_x509:NULL
        csr_attrs:NULL
        csr:NULL
        pkey:NULL
        header_ptr:NULL
        tmp_p10:NULL
[2453] est_simple_reenroll: Try to use est-test101 as client cert to authenticate
[1962] __est_curl_set_auth: trace
...
```

3. Once the certificate is saved, verify the local certificate configuration:

```
config vpn certificate local
(local) # get est-test101
name                : est-test101
password            : *
comments            :
private-key         : *
certificate         :
        Subject:     CN = firewall-portal1
        Issuer:      CN = estExampleCA
        Valid from:  2023-04-13 17:23:40  GMT
        Valid to:    2024-04-12 17:23:40  GMT
        Fingerprint: 4A:96:E1:73:6D:D3:64:FE:A3:A8:28:56:1D:39:05:37
```

```
        Root CA:     No
        Version:     3
        Serial Num:
                07:47:02
        Extensions:
                Name:     X509v3 Basic Constraints
                Critical: no
                Content:
                CA:FALSE

                Name:     X509v3 Key Usage
                Critical: no
                Content:
                Digital Signature

                Name:     X509v3 Subject Key Identifier
                Critical: no
                Content:
                9B:F8:39:D5:21:E6:FF:49:FF:AC:02:57:5B:FC:4C:1A:8B:1E:5D:8F

                Name:     X509v3 Authority Key Identifier
                Critical: no
                Content:
                1A:DF:39:84:C2:56:E6:6C:CF:2A:B4:26:A5:FD:0C:D2:43:F5:3D:3E


state               : OK
range               : global
source              : user
source-ip           : 0.0.0.0
ike-localid-type    : asn1dn
enroll-protocol     : est
est-server          : https://testrfc7030.com:8443
est-ca-id           :
est-http-username   : estuser
est-http-password   : *
est-client-cert     :
est-server-cert     :
est-srp-username    :
est-srp-password    :
auto-regenerate-days: 0
auto-regenerate-days-warning: 0
```

The Subject Key Identifier is the same, so no new key pair was generated.

ACME External Account Binding support

ACME External Account Binding (EAB) is supported as defined in RFC 8555 section 7.3.4.

EAB is a way to associate an ACME account with an existing non-ACME account, such as a CA customer database, by adding additional information in newAccount requests. This additional information is used by the

CA operating the ACME server to verify domain ownership by the requester, without the need for human users to follow interactive, natural-language instructions from the CA. Domain ownership verification is done when you register for EAB with your CA.

```
config vpn certificate local
    edit < name>
        set acme-eab-key-id <key>
        set acme-eab-key-hmac <HMAC>
    next
end
```

Command	Description
`acme-eab-key-id <key>`	External Account Binding Key ID (optional setting).
`acme-eab-key-hmac <HMAC>`	External Account Binding HMAC Key (URL-encoded base64).

A user obtains EAB from ACME CA or creates it using their web account access provided by ACME CA. Note that this feature is not supported by all CAs; for example, Let's Encrypt CA does not currently support EAB. Once created, EAB can be utilized for ACME certificate enrollment. Some ACME CAs allow the use of EAB as an authentication method, bypassing the standard online verification of domain ownership during the ACME certificate enrollment process via HTTP.

Example

In this example, public ZeroSSL CA is used (zerossl.com) as it supports EAB and allows registered accounts to create an EAB online. The server is an Azure VM with a public IP address and DNS.

To configure and verify ACME EAB:

1. Create an account with zeroSSL and create the EAB.
   Securely save both the `eab-key-id` and `acme-eab-key-hmac`. They are not stored in your account.
2. On the FortiGate, set the ACME interface to the port that is used for external communication:

```
config system acme
    set interface port1
end
```

3. Create a local VPN certificate using ACME as the enrollment protocol:

```
config vpn certificate local
    edit "test-acme-zeroSSL"
        set enroll-protocol acme2
        set acme-ca-url "https://acme.zerossl.com/v2/DV90"
        set acme-email owahn@fortinet.com
        set acme-eab-key-id "ZSx3bMEaa99RRt7wIjaRrw"
        set acme-eab-key-hmac "DeGr0jpQkZ1hqMVskqpe99dsyyPUM-SS77qqQQTTMM88-
yy331kf2DOzmG6dg96aM3-HuHi_OVELPsBFQSLNJw"
        set acme-domain qa-fgt-acme-test-vkonddcbssww2.westus.cloudapp.azure.com
    next
```

```
By enabling this feature you declare that you agree to the Terms of Service at
https://acme.zerossl.com/v2/DV90
Do you want to continue? (y/n)y
end
```

4. Verify the results:

```
config vpn certificate local
    edit "test-acme-zeroSSL"
        get
name                : test-acme-zeroSSL
password            : *
comments            : Renewed with ACME on Tue Mar  4 22:43:35 2025 (UTC)
private-key         : *
certificate         :
        Subject:     CN = qa-fgt-acme-test-vkonddcbssww2.westus.cloudapp.azure.com
        Issuer:      C = AT, O = ZeroSSL, CN = ZeroSSL RSA Domain Secure Site CA
        Valid from:  2025-03-04 00:00:00  GMT
        Valid to:    2025-06-02 23:59:59  GMT
        Fingerprint:
3F:9A:A8:1F:3A:C4:AB:44:15:66:FD:83:EA:D0:58:01:4E:73:0B:52:69:22:9F:A0:1D:0A:17:FE:6A:7E:33:4
2
        Root CA:     No
        Version:     3
        Serial Num:
                e8:90:0e:9f:0b:b7:76:3b:76:42:1b:1a:7a:81:02:e6
        Extensions:
                Name:     X509v3 Authority Key Identifier
                Critical: no
                Content:
                C8:D9:78:68:A2:D9:19:68:D5:3D:72:DE:5F:0A:3E:DC:B5:86:86:A6

                Name:     X509v3 Subject Key Identifier
                Critical: no
                Content:
                4C:B6:A3:DD:20:A4:33:2C:21:8A:B8:BA:96:A3:4E:FD:A3:2B:E3:BA

                Name:     X509v3 Key Usage
                Critical: yes
                Content:
                Digital Signature, Key Encipherment

                Name:     X509v3 Basic Constraints
                Critical: yes
                Content:
                CA:FALSE

                Name:     X509v3 Extended Key Usage
                Critical: no
                Content:
                TLS Web Server Authentication, TLS Web Client Authentication
```

```
                Name:     X509v3 Certificate Policies
                Critical: no
                Content:
                Policy: 1.3.6.1.4.1.6449.1.2.2.78
                CPS: https://sectigo.com/CPS
                Policy: 2.23.140.1.2.1

                Name:     Authority Information Access
                Critical: no
                Content:
                CA Issuers -
URI:http://zerossl.crt.sectigo.com/ZeroSSLRSADomainSecureSiteCA.crt
                OCSP - URI:http://zerossl.ocsp.sectigo.com

                Name:     CT Precertificate SCTs
                Critical: no
                Content:
                Signed Certificate Timestamp:
    Version   : v1 (0x0)
    Log ID    : CF:11:56:EE:D5:2E:7C:AF:F3:87:5B:D9:69:2E:9B:E9:
                1A:71:67:4A:B0:17:EC:AC:01:D2:5B:77:CE:CC:3B:08
    Timestamp : Mar  4 22:43:32.069 2025 GMT
    Extensions: none
    Signature : ecdsa-with-SHA256
                30:45:02:20:42:95:F3:A6:8D:23:8D:B7:FC:AC:69:E5:
                82:78:D7:AA:B6:15:A1:3B:2F:C2:57:66:36:E8:96:63:
                C4:16:F2:09:02:21:00:E3:59:20:04:3A:34:8B:0F:25:
                04:A3:3B:52:AA:F

                Name:     X509v3 Subject Alternative Name
                Critical: no
                Content:
                DNS:qa-fgt-acme-test-vkonddcbssww2.westus.cloudapp.azure.com

state               : OK
range               : global
source              : user
source-ip           : 0.0.0.0
ike-localid-type    : asn1dn
enroll-protocol     : acme2
acme-ca-url         : https://acme.zerossl.com/v2/DV90
acme-domain         : qa-fgt-acme-test-vkonddcbssww2.westus.cloudapp.azure.com
acme-email          : owahn@fortinet.com
acme-eab-key-id     : ZSxXXXXXXXXXXXXXXXXXXIjaRrw

acme-eab-key-hmac   : DeGr0XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXBFQSLNJw
acme-rsa-key-size   : 2048
acme-renew-window   : 30
```

If the certificate details are not shown, check the detailed status and error messages for the enrollment process:

```
#  diagnose sys acme status-full <acme-domain>
```

Security

The following topics provide information about security:

- BIOS-level signature and file integrity checking on page 4005
- Real-time file system integrity checking on page 4009
- Running a file system check automatically on page 4013
- Built-in entropy source on page 4013
- FortiGate VM unique certificate on page 4015
- Closed network VM license security on page 4016
- Encrypt configuration files in the eCryptfs file system on page 4018
- FIPS-CC mode and OpenSSL FIPS provider on page 4018
- Enhanced administrator password security on page 4019
- Black box for FortiGate on page 4022
- DoS protection on page 4025

BIOS-level signature and file integrity checking

The BIOS-level signature and integrity checking includes several checks that occur during different stages.

Stage	Checks
BIOS-level signature and integrity check during file upload	Dually-signed images such as the firmware image, AV engine file and IPS engine file are verified during file upload while FortiOS is running.
BIOS-level signature and integrity check during the boot process	Dually-signed images such as the firmware image, AV engine file and IPS engine file are verified during the boot process before the kernel is mounted.
BIOS-level file integrity check during bootup as files are mounted	Signed hashes of important files related to the kernel, filesystems and AV/IPS engines and executables are verified during bootup as they are mounted and loaded into user space.

Each FortiOS GA firmware image, AV engine file, and IPS engine file are dually-signed by the Fortinet CA and a third-party CA.

Signature checking occurs when the FortiOS firmware, AV, and IPS engine files are uploaded. This allows the FortiGate to either warn users of potential risks involved with uploading an unauthenticated file, or block the file upload depending on the BIOS security level.

During the boot process before the kernel is loaded, the BIOS also verifies that each file matches their secure hash as indicated by their certificates. Users are warned when there is a failed integrity check, and the system may be prevented from booting depending on the severity and the BIOS security level.

Once the signature check passes, important files are extracted, mounted and loaded into user space during the bootup. All the important files are verified against their signed hashes to validate the integrity of the files before

they can be mounted or loaded into user space. The hash file containing hashes of all executables and shared libraries is also verified to ensure the integrity of the file before the individual hashes are loaded into memory.

When the system is started, real-time protection kicks in. See Real-time file system integrity checking on page 4009 for more details.

BIOS-level signature and integrity check on firmware images

The outcome of the signature and integrity check during file upload and boot process depends on the security level configured in BIOS and the certificate authority that signed the file.

The security levels on the BIOS are:

FortiOS level	Behavior
High	FortiOS and BIOS only accept certified images.
Low	FortiOS and BIOS only accept certified images without a warning and un-certified images with a warning

The following table summarizes the use cases and the potential outcome based on the security level.

Use case	Certificate signed by		Outcome based on security level	
	Fortinet CA	Third-party CA	Level High	Level Low
GA-Certified (GA firmware, Beta firmware, Special Technical Support final builds)	Yes	Yes	Accept	Accept
Non-GA certified (Special builds: Special Technical Support and NPI quick builds)	Yes	No	Warning	Accept
Interim and Dev builds, or unknown build	No	Yes or No	Reject	Warning

On newer device models, there is a Signed Firmware LED light indicating High or Low BIOS security level.

On FortiGates without supported BIOS security levels, the device acts like security level High. For example, on a FortiGate-VM that does not have BIOS, the security level is defaulted to level High.

Platforms with old BIOS versions will support security levels 0, 1, and 2, while FortiOS will support levels High and Low. BIOS level 2 will correspond to the behaviors in Level High, and BIOS level 0 and 1 will correspond to behaviors in Level Low.

Security levels can be verified using the command `get system status`.

Examples of BIOS-level signature and integrity check during file upload

The following examples outline the different use cases when upgrading firmware and AV files on a FortiGate model that supports BIOS security levels, and a FortiGate model that does not support BIOS security levels.

For more information, see the Firmware & Registration on page 3741 section and Manual updates on page 3928.

Upgrading on a device with BIOS security levels

The following use cases are applicable when upgrading firmware and AV files on a FortiGate with BIOS security levels. Firmware is upgraded using the *System > Firmware & Registration* page, and AV files are upgraded using the *System > FortiGuard* page.

Security Level	Use case	Behavior
High	Load certified GA image in TFTP in boot menu	FortiGate boots up without warning messages.
High	Restore certified GA image in CLI	FortiGate boots up without warning messages.
High	Load certified non-GA image in TFTP in boot menu	FortiGate boots up with a warning message: `Warning: Non GA FOS image!`

High	Restore certified non-GA image in CLI	FortiGate displays a warning upon upload:
		`Warning: This firmware image is no GA certified!`
		FortiGate boots up with a warning message:
		`Warning: Non GA FOS image!`

High	Load un-certified interim image in TFTP in boot menu	The upload is blocked.
		A warning is displayed:
		`Checking image… This firmware image is not certified!`
		`Aborting firmware installation. Please power cycle.`
		`System halted.`

High	Restore un-certified interim image in CLI	The upload is blocked. A warning is displayed: `Image verification failed!` …
Low	Load certified GA or non-GA image in TFTP in boot menu	FortiGate boots up without warning messages.

Security Level	Use case	Behavior
Low	Restore certified GA or non-GA image in CLI	FortiGate boots up without warning messages.

Low	Load un-certified interim image in TFTP in boot menu	FortiGate outputs a warning message, but the upload is allowed to proceed:
		`Warning: Image decode failed. Try to continue under security level 1…`
		`OK`
		`This firmware image is not certified!`
		`Save as Default firmware/Backup firmware/Run image without saving [D/B/R]?`
		After boot up:
		`System file integrity init check failed!`

Low	Restore un-certified interim image in CLI	FortiGate outputs a warning message, but the upload is allowed to proceed:
		`Image verification failed!`
		`...`
		`Please continue only if you understand and are willing to accept the risks.`
		`Do you want to continue? (y/n)`
		During boot up:
		`Warning: FOS is not authenticated! Continue booting under security level 1...`
		`Initializing firewall...`
		After boot up:
		`System file integrity init check failed!`

Upgrading on a device without BIOS security levels

The following use cases are applicable when upgrading firmware and AV files on a FortiGate without BIOS security levels. Firmware is upgraded using the *System > Firmware & Registration* page, and AV files are upgraded using the *System > FortiGuard* page. A FortiGate 60F is used in these examples and acts like it has security level 1.

When upgrading from 7.2.4 to 7.4.0 with a dually-signed firmware image, FortiOS verifies the certificates and accepts the image.

When upgrading from 7.2.4 to 7.4.0 with an unsigned firmware image in the GUI, FortiOS is unable to verify the certificates and the image fails verification. A warning dialog is displayed indicating that *This firmware failed signature validation*, but the user can click *Continue* to use the firmware.

When running 7.4.0 and uploading an unsigned AV engine file on the *System > FortiGuard* page, FortiOS is unable to verify the certificates and the file fails verification. A warning dialog is displayed indicating that *This package file has no signature for validation*, but the user can click *OK* to use the file.

BIOS-level file integrity check on important file-system and object files

During bootup, the kernel is required to verify the signed hashes of important file-system and object files. This prevents unauthorized changes to file-systems to be mounted and other unauthorized objects to be loaded into user space on bootup.

This verification does not depend on the security level of the device. The verification will always run when the firmware image type is a GA, SA, Beta, or Top3 image. If the signed hash verification fails, the system will halt during bootup.

Example

Upon detection of an altered IPS library file upon bootup, the system will halt as follows:

```
FortiGate-60F (18:03-01.27.2017)
Ver:05000012
Serial number: FGT60FTK1804xxxx
CPU: 1000MHz
Total RAM: 2 GB
Initializing boot device...
Initializing MAC... nplite#0
Please wait for OS to boot, or press any key to display configuration menu......
```

```
Booting OS...
Reading boot image... 2891501 bytes.
Initializing firewall...
fos_ima: System Integrity check failed....
CPU3: stopping
CPU1: stopping
CPU0: stopping
```

The exact display in the CLI may vary depending on the device model, security level, or reasons for the failed verification.

Real-time file system integrity checking

Real-time file system integrity checking has two main purposes:

- Prevent unauthorized modification of important binaries.
- Detect unauthorized binaries and prevent them from running.

How it works

A hash of all executable binary files and shared libraries are taken during image build time. The file containing these hashes, called the executable hash, is also hashed. This new hash is signed together with other important files like the FortiOS firmware, AV and IPS engine files.

When the FortiGate boots, the system performs a BIOS level integrity check on the firmware image, the AV engine file, and the IPS engine file. These files are signed by the process described in BIOS-level signature and file integrity checking on page 4005, and the BIOS verifies their signature against their certificates. The hash of the executable hash will also follow the BIOS-level signature and integrity check during the boot process, to prevent any unauthorized changes from being loaded.

Once these files are verified to be authentic, the BIOS can extract and boot the root filesystem and other executables and libraries, which then go through another file integrity check.

Once the file integrity checks passes, the corresponding hashes of every executable and shared library can be extracted from the executable hash and loaded into memory. The system is started and real-time protection begins.

The important executables and binaries are protected from write access and any modifications. It also blocks the kernel from loading any modules. Any unauthorized loading of modules is blocked. If violations are found, logs are triggered.

When each executable and shared library is initialized, it will be verified against its respective hashes to ensure integrity. If there is a hash mismatch when attempting to run a binary, that binary is blocked from running, and the system is rebooted. A log will be generated with ID 20234.

If there is a missing hash when attempting to run a binary, then the system is rebooted. A log will be generated with ID 20223.

The system also runs a periodic check to verify the integrity of important binaries and AV and IPS engines.

Log summary

The following logs are recorded when specific actions take place.

Log	Description
20230 - LOG_ID_SYS_ SECURITY_WRITE_VIOLATION 432	The root filesystem is read only. Any modification triggers this log.
20231 - LOG_ID_SYS_ SECURITY_HARDLINK_ VIOLATION 432	An attacker trying to replace symlink triggers this log.
20232 - LOG_ID_SYS_ SECURITY_LOAD_MODULE_ VIOLATION 433	Only the kernel can load modules. Any unusual loading of modules triggers this log.

Log	Description
20233 - LOG_ID_SYS_SECURITY_FILE_HASH_MISSING 434	File hashes are generated for legitimate files during bootup. If a hash cannot be found, the file may be suspicious as it could be a new routine inserted by an attacker. The binary is blocked.
20234 - LOG_ID_SYS_SECURITY_FILE_HASH_MISMATCH 434	File hashes are generated for legitimate files during bootup. If a hash does not match when the file is exercised, it is an indication that it could have been modified by an attacker. The system is rebooted.

Detection examples

Example 1: system reboots due to mismatched hash

```
fos_ima: fos_process_appraise 110: Executable File(/bin/node) doesn't match previous hash, it has
been changed
Restarting system.
…
fos_ima: fos_process_appraise 110: Executable File(/lib/libc.so.6) doesn't match previous hash, it
has been changed
Restarting system.
…
```

Logs similar to the following are captured:

```
date="2023-06-16" time="12:01:44" id=7245222014288399309 bid=471609558 dvid=6533 itime=1686909705
euid=3 epid=3 dsteuid=3 dstepid=3 logver=604132092 logid="0100020234" type="event"
subtype="system" level="alert" msg="Hash of executable file(/bin/init) doesn't match the
previous." logdesc="Integrity check of Run/loading Excutable File failed without Integrity
measure" severity="alert" eventtime=1686909705825483706 tz="+0200" devid="xxxxxxxxx" vd="root"
devname="xxxxxxxxx"

date="2023-06-15" time="09:57:54" id=7244819017507013700 bid=470303007 dvid=1431 itime=1686815875
euid=3 epid=3 dsteuid=3 dstepid=3 logver=604132092 logid="0100020234" type="event"
subtype="system" level="alert" msg="Hash of executable file(/lib/libc.so.6) doesn't match the
previous." logdesc="Integrity check of Run/loading Excutable File failed without Integrity
measure" severity="alert" eventtime=1686815874936267770 tz="+0200" devid=" xxxxxxxxx " vd="root"
devname=" xxxxxxxxx”
```

Example 2: suspected compromise due to an observed indicator of compromise (IoC)

```
fos_ima: fos_process_appraise 99: Suspicous Executable File(/data2/libcrashpad.so) is missing hash
…
fos_ima: fos_process_appraise 99: Suspicous Executable File(/data2/flatkc_info) is missing hash
…
```

No logs are found.

Corrective action

In the previous examples where a mismatched or missing hash occurs, alert technical support straight away so that they may gather information to start a forensic analysis with our internal PSIRT team. There are two possible outcomes:

1. The firewall is reporting a false positive, in which a bug causes a mismatched or missing hash.
   Once verified by technical support, the corrective action may to be upgrade to a newer build where the bug is fixed
2. An actual compromise has occurred, or is occurring.
   The system could be blocking an offending binary that causes the system to malfunction, or the system could reboot to protect itself from compromise.

In either case, contact technical support for further forensic analysis. If an IoC is detected and it is determined that the persistent threat resides on the FortiGate, a reflash and reload of the firmware may be recommended.

Unauthorized firmware modification attempt reporting

In the rare event that unauthorized modification is detected in the firmware, the system will immediately log and report the modification attempt to FortiGuard through a secure channel. Payloads are encrypted to ensure the security of the transferred information. Information about the attempted modification of firmware helps Fortinet Inc. proactively investigate the incident and protect future malicious attempts at compromising the system.

After reporting the modification attempt, the FortiGate real-time file system integrity checking feature continues with the required actions based on the assessed threat. This may involve reverting the change and rebooting the firewall to mitigate the threat.

Example

This example demonstrates when an attempt to alter files in the 'bin' directory was made by a threat actor.

Captured log:

```
1: date=2024-02-16 time=18:29:15 eventtime=1708136955710925685 tz="-0800" logid="0100020230"
type="event" subtype="system" level="alert" vd="vd1" logdesc="Write Permission Violation" msg="
[Write Violation: try to write readonly file](/bin/lspci)."
```

The FortiGate sends an encrypted report to FortiGuard with information about the affected platform and the Modification Attempt such as:

- FortiGate serial number
- Model number
- FortiOS firmware
- Type of modification attempt (such as *Write violation*)
- File path (such as */bin/lspci*)
- File size
- Time of access and modification

Running a file system check automatically

There is an option in FortiOS to enable automatic file system checks if the FortiGate shuts down ungracefully.

By default, the automatic file system check is disabled. When an administrator logs in after an ungraceful shutdown, a warning message appears advising them to manually run a file system check. A warning also appears in the CLI:

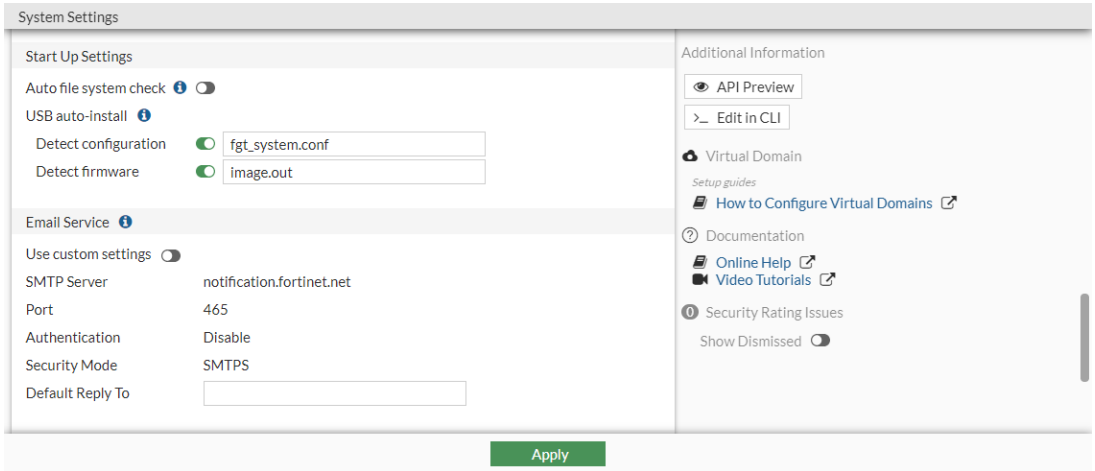
```
WARNING: File System Check Recommended! Unsafe reboot may have caused inconsistency in disk drive.
It is strongly recommended that you check file system consistency before proceeding.
Please run 'execute disk scan 17'
Note: The device will reboot and scan during startup. This may take up to an hour
```

Enabling automatic file system checks

You can enable automatic file system checks in both the GUI and CLI.

To enable automatic file system checks in the GUI:

1. Go to *System* > *Settings* > *Device*.
2. In the *Start Up Settings* section, enable *Auto file system check*.
3. Click *Apply*.



To enable automatic file system checks using the CLI:

```
config system global
   set autorun-log-fsck enable
end
```

Built-in entropy source

FortiOS includes a built-in entropy source, which eliminates the need for a physical USB entropy token when booting up in FIPS mode on any platform. This enhancement continues to meet the requirements of FIPS 140-3

Certification by changing the source of entropy to CPU jitter entropy.

> The `entropy-token` parameter under `config system fips-cc` is removed if the FortiGate is a SoC3, SoC4, or CP9 device.

To verify that jitter entropy is used:

1. Enable FIPS-CC mode, which will cause the FortiGate to reboot:

```
config system fips-cc
    set status enable
end

Please enter admin administrator password:********
Please re-enter admin administrator password:********

Warning: most configuration will be lost,
do you want to continue?(y/n) y
The system is going down NOW !!

Please stand by while rebooting the system.
Restarting system.
...
Reading boot image 2919154 bytes.
Initializing firewall...
System is starting...

FIPS-CC mode: Starting self-tests.
Running Configuration/VPN Bypass test...      passed
Running AES test...                           passed
Running SHA1-HMAC test...                     passed
Running SHA256-HMAC test...                   passed
Running SHA384/512-HMAC test...               passed
Running RSA test...                           passed
Running ECDSA test...                         passed
Running TLS1.1-KDF test...                    passed
Running TLS1.2-KDF test...                    passed
Running SSH-KDF test...                       passed
Running IKEv1-KDF test...                     passed
Running IKEv2-KDF test...                     passed
Running Primitive-Z test...                   passed
Running Firmware integrity test...            passed
Running RBG-instantiate test...               passed
Running RBG-reseed test...                    passed
Running RBG-generate test...                  passed
Self-tests passed
```

2. Verify the entropy token user event logs:

```
# execute log filter category event
# execute log filter field logid 0102038012
```

```
# execute log display

3 logs found.
3 logs returned.

1: date=2023-07-18 time=20:27:56 eventtime=1689737275853093806 tz="-0700" logid="0102038012"
type="event" subtype="user" level="notice" vd="root" logdesc="Seeding from entropy source"
user="system" action="reseeding" msg="Reseeding PRNG from JitterEnt entropy"

2: date=2023-07-18 time=20:26:56 eventtime=1689737146847643497 tz="-0700" logid="0102038012"
type="event" subtype="user" level="notice" vd="root" logdesc="Seeding from entropy source"
user="system" action="seeding" msg="Seeding PRNG from JitterEnt entropy"

3: date=2023-07-18 time=19:29:25 eventtime=1689733702417108422 tz="-0700" logid="0102038012"
type="event" subtype="user" level="notice" vd="root" logdesc="Seeding from entropy source"
user="system" action="seeding" msg="Seeding PRNG from JitterEnt entropy"
```

FortiGate VM unique certificate

To safeguard against certificate compromise, FortiGate VM and FortiAnalyzer VM use the same deployment model as FortiManager VM where the license file contains a unique certificate tied to the serial number of the virtual device.

A hardware appliance usually comes with a BIOS certificate with a unique serial number that identifies the hardware appliance. This built-in BIOS certificate is different from a firmware certificate. A firmware certificate is distributed in all appliances with the same firmware version.

Using a BIOS certificate with a built-in serial number provides a high trust level for the other side in X.509 authentication.

Since a VM appliance has no BIOS certificate, a signed VM license can provide an equivalent of a BIOS certificate. The VM license assigns a serial number in the BIOS equivalent certificate. This gives the certificate an abstract access ability, which is similar to a BIOS certificate with the same high trust level.

Sample configurations

Depending on the firmware version and VM license, the common name (CN) on the certificate will be configured differently.

License	Firmware	
	6.0	6.2 and later
6.0	CN = FortiGate	CN = FortiGate
6.2 and later	CN = FortiGate	CN = serial number

To view validated certificates:

1. Go to *System* > *Certificates*.
2. Double-click on a VM certificate. There are two VM certificates:

- *Fortinet_Factory*
- *Fortinet_Factory_Backup*

The *Certificate Detail Information* window displays.

Closed network VM license security

The CMS signature is verified immediately after the license is loaded. This ensures that the license is from FortiCare and confirms the authenticity of the license's contents and contracts, enhancing license integrity and customer trust.

If a valid offline license with a CMS signature is loaded:

```
FGVM2VTM22222222# get system status
Version: FortiGate-VM64-KVM v7.6.0,build9999,240524
First GA patch build date: 230509
Security Level: 0
Firmware Signature: not-certified
...
Serial-Number: FGVM2VTM22222222
License Status: Valid
License Expiration Date: 2024-08-02
VM Resources: 1 CPU/2 allowed, 1992 MB RAM
...
```

```
FGVM2VTM22222222# diagnose debug vm-print-license
SerialNumber: FGVM2VTM22222222
CreateDate: Wed Mar 20 18:47:49 2024
License expires: Fri Aug  2 00:00:00 2024
Expiry: 134
UUID: fdbf7aa999999999a9999aa578127e67
Default Contract: FMWR:6:20230802:20240804,ENHN:20:20230802:20240804,...,IPMC:6:20230802:20240804
Key: yes
Cert: yes
Key2: yes
Cert2: yes
Signature: yes
Model: 2V (6)
CPU: 2
MEM: 2147483647
VDOM license:
  permanent: 2
  subscription: 0
```

If an old offline license without a CMS signature is loaded:

```
FGVM2VTM22222222# get system status
Version: FortiGate-VM64-KVM v7.6.0,build9999,240524
First GA patch build date: 230509
```

```
Security Level: 0
Firmware Signature: not-certified
...
Serial-Number: FGVM2VTM22222222
License Status: Invalid
License Expiration Date: 2024-08-02
VM Resources: 1 CPU/2 allowed, 1992 MB RAM
...
```

```
FGVM2VTM22222222# diagnose debug vm-print-license
SerialNumber: FGVM2VTM22222222
CreateDate: Wed Aug  2 20:18:51 2023
License expires: Fri Aug  2 00:00:00 2024
Expiry: 365
UUID: fdbf7aa999999999a9999aa578127e67
Default Contract: FMWR:6:20230802:20240804,ENHN:20:20230802:20240804,...,IPMC:6:20230802:20240804
Key: yes
Cert: yes
Key2: yes
Cert2: yes
Model: 2V (6)
CPU: 2
MEM: 2147483647
VDOM license:
```

If an offline license with a modified CMS signature is loaded the license is invalid and there is no signature:

```
FGVM2VTM22222222# get system status
Version: FortiGate-VM64-KVM v7.6.0,build9999,240524
First GA patch build date: 230509
Security Level: 0
Firmware Signature: not-certified
...
Serial-Number: FGVM2VTM22222222
License Status: Invalid
License Expiration Date: 2024-08-02
VM Resources: 1 CPU/2 allowed, 1992 MB RAM
...
```

```
FGVM2VTM22222222# diagnose debug vm-print-license
SerialNumber: FGVM2VTM22222222
CreateDate: Wed Mar 20 18:47:49 2024
License expires: Fri Aug  2 00:00:00 2024
Expiry: 134
UUID: fdbf7aa999999999a9999aa578127e67
Default Contract: FMWR:6:20230802:20240804,ENHN:20:20230802:20240804,...,IPMC:6:20230802:20240804
Key: yes
Cert: yes
Key2: yes
Cert2: yes
```

```
Model: 2V (6)
CPU: 2
MEM: 2147483647
VDOM license:
  permanent: 2
  subscription: 0
```

Encrypt configuration files in the eCryptfs file system

Configuration files are encrypted in the eCryptfs file system when the system reboots or shuts down, and are decrypted when the system boots up and has to load the configuration to CMDB.

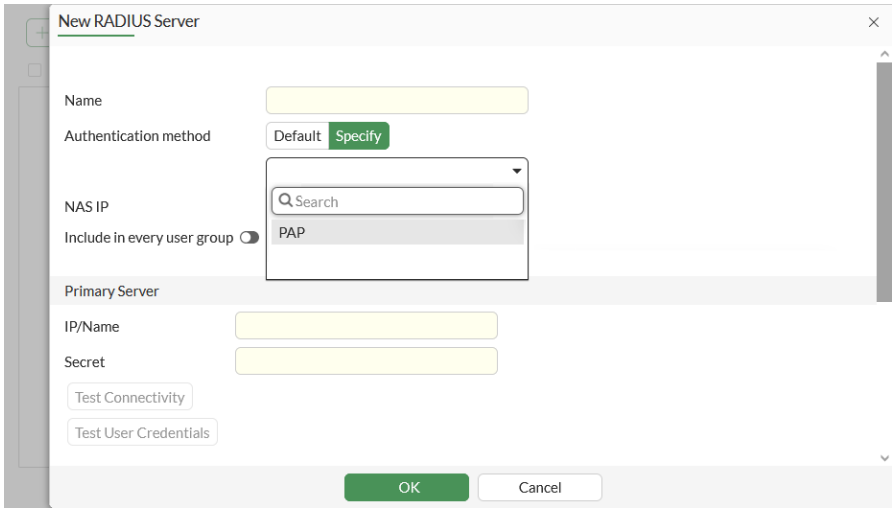
If the device supports TPM, the 32 byte eCryptfs encryption key is randomly generated and stored in the TPM, like the private-data-encryption key. If the device does not support TPM, the key is generated by the cryptographically secure pseudorandom number generator (CSPRNG) and stored on the disk. See Trusted platform module support on page 3830 and TPM support for FortiGate-VM on page 4925 for more information.

FIPS-CC mode and OpenSSL FIPS provider

When the device is in FIPS-CC mode, the OpenSSL FIPS provider is installed globally at startup, ensuring that any OpenSSL application is automatically compliant with FIPS regulations. An OpensSSL FIPS provider test (`ossl-fips-provider-test`) is added, and the self-test runs automatically at startup. The TLSv1.1 KDF self-test is removed.

The system defaults to the more secure TLSv1.2 and TLSv1.3 protocols instead of SSL3.0 and TLS1.1, and only Diffie-Hellman parameters of 2048 bits or higher are permitted, ensuring a robust security posture and aligning with industry standards.

When configuring a RADIUS server, *Authentication method* (`auth-type`) can only be set to *PAP* (password authentication protocol), and `transport-protocol` can only be set to `tls`.



To manually run the OpenSSL FIPS provider self-test:

```
# execute fips kat ossl-fips-provider-test
HMAC              : KAT_Integrity        : Pass
HMAC              : Module_Integrity     : Pass
SHA1              : KAT_Digest           : Pass
SHA2              : KAT_Digest           : Pass
SHA3              : KAT_Digest           : Pass
AES_GCM           : KAT_Cipher           : Pass
AES_ECB_Decrypt   : KAT_Cipher           : Pass
RSA               : KAT_Signature        : Pass
ECDSA             : KAT_Signature        : Pass
ECDSA             : KAT_Signature        : Pass
DSA               : KAT_Signature        : Pass
TLS13_KDF_EXTRACT : KAT_KDF              : Pass
TLS13_KDF_EXPAND  : KAT_KDF              : Pass
TLS12_PRF         : KAT_KDF              : Pass
PBKDF2            : KAT_KDF              : Pass
SSHKDF            : KAT_KDF              : Pass
KBKDF             : KAT_KDF              : Pass
HKDF              : KAT_KDF              : Pass
SSKDF             : KAT_KDF              : Pass
X963KDF           : KAT_KDF              : Pass
X942KDF           : KAT_KDF              : Pass
HASH              : DRBG                 : Pass
CTR               : DRBG                 : Pass
HMAC              : DRBG                 : Pass
DH                : KAT_KA               : Pass
ECDH              : KAT_KA               : Pass
RSA_Encrypt       : KAT_AsymmetricCipher : Pass
RSA_Decrypt       : KAT_AsymmetricCipher : Pass
RSA_Decrypt       : KAT_AsymmetricCipher : Pass
Running OSSL-FIPS-TEST test...                 passed
```

Enhanced administrator password security

The PBKDF2 hashing scheme with randomized salts is now used to store system administrator passwords on the FortiGate to enhance security. Previously the SHA256 hashing algorithm was used.

With this change, a new command is available to maintain FortiOS downgrade:

```
config system password-policy
    set login-lockout-upon-weaker-encryption {enable | disable}
end
```

Command	Description
`login-lockout-upon-weaker-encryption {enable | disable}`	Enable/disable system administrator login lockout after downgrade to FortiOS firmware that does not support safer passwords (default = disable).

Command	Description
	When disabled, system administrator passwords with SHA256 hash are kept after successfully converting to PBKDF2 hash. SHA256 hashed passwords are used after downgrading to a firmware version that does not support PBKDF2 hashed passwords. When enabled, system administrator passwords that are converted to PBKDF2 hash will immediately remove the SHA256 hashed password. Upon downgrading the FortiOS firmware to a lower version, where safer passwords are unsupported, the administrators will be locked out.

When creating a new administrative user in FortiOS 7.6.1 or later, the PBKDF2 hashing scheme is used to store the password. When displayed in the CLI, the password is encoded with a prefix of PB2:

```
# config  system  admin
(admin)# edit  admin2
new entry 'admin2' added
(admin2)# set accprofile  super_admin
(admin2)# set vdom  root
(admin2)# set password 123456
(admin2)# show
config system admin
    edit "admin2"
        set accprofile "super_admin"
        set vdom "root"
        set password ENC
PB20jOHUbfWEI7eKFTnKkE/qBgE5md70OjpXqiHhHTxo78GnIZshrIq67dT1IUouJSxgEZ6NYeAdAS9vLHpZGlcpBmtfUbfFOa
6qKaVAN+F9CM=
    next
end
```

Upgrade

To view system administrator passwords before and after upgrade to FortiOS 7.6.1:

1. Before upgrade to FortiOS 7.6.1 or later, view the encoded password. The encoded password shows a SH2 prefix because it was hashed with the SHA256 algorithm:

   ```
   # show  system  admin
   config system admin
       edit "admin1"
           set accprofile "prof_admin"
           set vdom "root"
           set password ENC SH2RHqyB8gaEKC10dzpgU6lZg7YSpnb21wWLFOtqzMlpyKJZyyq3ISFYPyIL/s=
       next
   end
   ```

2. Upgrade to FortiOS 7.6.1 or later. Each system administrator password hashed with SHA256 is stored on FortiGate until each system administrator successfully logs in to FortiOS.

If a system administrator does not log in to FortiGate after upgrading to FortiOS 7.6.1 or later, their password remains saved as the SHA256 hashed password.

3. Log in to FortiOS. The password is converted to a PBKDF2 hashed password.
4. View the encoded password. The encoded password shows a `PB2` prefix because it was hashed with the PBKDF2 algorithm:

```
FortiGate login: admin1
Password:
Verifying password...
$ show  system  admin
config system admin
    edit "admin1"
        set accprofile "prof_admin"
        set vdom "root"
        set password ENC
PB2a2X8D3DIt0gXbBXVdknLZb48BKrGTD50z//UrbpWAD5kpFdwqBKie0h8STxL6Db//wQ2ZWN/5FW3+DkX3+0nBE1RNbe
TKSVi18WcFmSPDQM=
    next
end
```

Downgrade

To support downgrading to an older version that does not support the PBKDF2 hashed password, by default, the old SHA256 hashed password is still stored in the system after being converted to PBKDF2. This is controlled by the following setting that is disabled by default:

```
config system password-policy
    set login-lockout-upon-weaker-encryption disable
end
```

Downgrading will successfully restore the SHA256 hashed password and operations will continue uninterrupted.

FortiGate supports downgrading from 8.0 to 7.6. However, it does not support direct downgrade from 8.0 to 7.4 or below. Step through each minor version when downgrading (8.0 > 7.6 > 7.4).

If the `login-lockout-upon-weaker-encryption` option is enabled:

```
config system password-policy
    set login-lockout-upon-weaker-encryption enable
end
```

The SHA256 hashed password will be removed from the system as soon as it is converted to a PBKDF2 hashed password upon a successful login.

During a downgrade operation, the system will display the following warning:

```
# execute restore  image  ftp FGT_2601F-v7.6.0.F-build3401-FORTINET.out 172.16.106.105 username
password
This operation will replace the current firmware version and reboot the system!
```

```
Do you want to continue? (y/n)y
Please wait...
Connect to ftp server 172.16.106.105 ...
Get image from ftp server OK.
Verifying the signature of the firmware image.
Image verification OK!
Warning: Installing image v7.6.0 from v7.6.1 is not a recommended upgrade path. Continuing with
the upgrade may result in loss of configuration. Do you want to proceed?
Do you want to continue? (y/n)y
…
You are downgrading to a version that does not support safer passwords.
After downgrade, some administrative user (e.g., admin1) no longer will be able to login.
Do you want to continue? (y/n)
```

Administrators can choose to proceed or abort this downgrade.

Finally, if an administrator wants to restore the SHA256 hashed password for a downgrade, they can do the following:

1. Disable the `login-lockout-upon-weaker-encryption` option.
2. Log out the current administrator.
3. For each system administrator, log in to the FortiGate to generate the SHA256 hashed password by running the SHA256 hashing algorithm.

> ⚠ After administrator passwords are converted to PBKDF2 hashed passwords, loading the config file to an older version that does not support safer passwords will lock out the administrators.

Black box for FortiGate

The black-box feature allows supported FortiGate models to capture and save information about the system at a regular interval inside encrypted debug logs stored on an NVMe drive on the FortiGate. The debug logs capture system statuses, which can be used to troubleshoot unexpected problems, if they occur. The logs are encrypted with AES256 using a generated symmetric key that is stored in the TPM.

In order to decrypt the debug logs, you must export from FortiGate an encrypted version of the symmetric key, which is encrypted with the factory private key of the FortiGate. Only Fortinet Technical Support can decrypt the symmetric key using the public key known to Fortinet.

This feature is available only on FortiGate models with an NVMe drive and TPM, including:

- FortiGate 400G
- FortiGate 401G
- FortiGate 700G
- FortiGate 701G
- FortiGate 3000G
- FortiGate 3001G
- FortiGate 3500G
- FortiGate 3501G

CLI syntax

```
config system global
  set black-box {enable* | disable}
  set black-box-interval <interval>
end
```

Option	Description
black-box {enable* | disable}	Enable or disable the black-box feature (default = enabled).
black-box-interval <interval>	Black-box recording interval, in seconds (default = 20).

To check if the black box is running:

```
# diagnose debug black-box info
    Black box status: running
    Black box recording interval: 20 seconds.
```

To upload black-box logs through TFTP:

```
# diagnose debug black-box upload tftp <ip address> <log date> <log time>
```

Where:

- `ip address`: Address of the TFTP server
- `log date`: date in the format YYYYMMDD. For example, 2025/04/12 is 20250412
- `log time`: the hour of the log and can be excluded. For example, 10:00am is 10.
  When `log time` is excluded, FortiGate uploads all available blackbox logs for the `log date` time.

To upload black-box logs through SFTP:

```
# diagnose debug black-box upload sftp <ip address> <user> <password> <destination folder>
<yyyymmdd>
```

To upload black-box logs through SCP:

```
# diagnose debug black-box upload scp <destination string> <yyyymmdd>
```

To export the black-box key:

```
# diagnose debug black-box get-key
```

To reset the black box:

Resetting the black-box deletes all files generated by the black box.

1. Disable the black box.

```
config system global
  set black-box disable
end
```

2. Reset the black box.

```
# diagnose debug black-box reset
```

To format the black-box log partition:

Format the black-box log partition if the file system is corrupt.

1. Disable the black box.

```
config system global
  set black-box disable
end
```

2. Reset the black box.

```
# diagnose debug black-box format
```

Example

This example shows how you can collect black-box information and provide it to Fortinet Technical Support for decryption and further analysis when you encounter the following issues:

- Occasional high CPU utilization
- Network speed slowdown
- Issues happening at irregular intervals

To collect black-box information:

1. Ensure the black-box feature is enabled and configured to a preferred interval:.

```
config system global
    set black-box enable
    set black-box-interval 20
end
```

2. Check that the black box is running:

```
# diagnose debug black-box info
Black box status: running
Black box recording interval: 20 seconds.
```

When the black box is running, FortiGate records debug logs to its NVMe drive at the regular interval. Logs are encrypted with AES256 encryption. Files are stored in a folder by date and rolled over every hour into a new file.

For example, debugs recorded on 2025/04/12 are stored in the `20250412/` folder with log files named:

```
-rw-r--r--    1 0        0       Sat Apr 12 01:00:27 2025          5945090 20250412-00.log.gz
-rw-r--r--    1 0        0       Sat Apr 12 02:00:21 2025          5905150 20250412-01.log.gz
-rw-r--r--    1 0        0       Sat Apr 12 10:10:08 2025          5388410 20250412-02.log.gz
-rw-r--r--    1 0        0       Sat Apr 12 11:00:27 2025         13612383 20250412-10.log.gz
-rw-r--r--    1 0        0       Sat Apr 12 12:00:20 2025          5894165 20250412-11.log.gz
-rw-r--r--    1 0        0       Sat Apr 12 13:00:14 2025          5891399 20250412-12.log.gz
-rw-r--r--    1 0        0       Sat Apr 12 14:00:08 2025          5896160 20250412-13.log.gz
-rw-r--r--    1 0        0       Sat Apr 12 15:00:28 2025          5940129 20250412-14.log.gz
-rw-r--r--    1 0        0       Sat Apr 12 16:00:22 2025          5893402 20250412-15.log.gz
-rw-r--r--    1 0        0       Sat Apr 12 17:00:16 2025          5898253 20250412-16.log.gz
...
```

3. If an issue occurs, estimate the approximate date and time that the issue occurred, and back up the black-box debug logs to the TFTP server.

   In this example, the logs for 2025/04/12 10:00 am are backed up to the TFTP server:

```
# diagnose debug black-box upload tftp 192.168.1.58 20250412 10
Uploading file 20250412-10.log.gz...
##############
Send file to tftp server OK.
```

4. Generate an encrypted version of the FortiGate symmetric key and provide it to Fortinet Technical Support.

   Fortinet Technical Support requires the symmetric key to decrypt the debug logs. The symmetric key is protected by the FortiGate's TPM. Technical Support will use an internal tool to decrypt the generated key, which will then be used to decrypt and open the log files.

```
# diagnose debug black-box get-key
    d2gil8gVP1gUYfC8HVgrH5L3Bz0wPhhnsajDeRQ0DL40AY6qPH+b+EmujHPxVjSu
    7cllGnGykCQQCkibvG9yU/3py6BedUqFGygJMhFHZ6jTpzlgkwgUoO/a/MhW9GP1
    Ph3s9PTB0+73lJMMdacwWzoVa4/DgeR5aGdzZFUGYzLCZezh4PixD8rREPVO5Alm
    +zo3W3ImZVhflIG2hCHJZWhDPGNdM+2PZSkMKd7O/VS7m8xTY3W9tTvF8kV9pPUK
    OozzUWSEGw3zZYtEW/kfRSKJi2grE2F79mG2FKWVsqAGJ5z9sjmKM+Q/p4si6H7S
    oYDwJriXP6dPYeXTBhsGGRf8WumDtCVxFGwOy3TV2yI+Mxmxc+X1BkLWEcbEWg5v
    ERa6HAJqBgWOfG0/S2n2juovLyWiVsJ6inKYAdbk0kLENHlkpqvVZ0TrWkv42hzU
    aYYD2DqnnpCYhJ1/Oer/8xZzazY4jxpN/Yw9ks9Ia+ua7dCUosiG50/3N7WHnUW9
    X9A7VbsO+++GyA2xaigDH9+/vqslcT8e9KXeuzaX/mVDev1MNHKqXu+D0jBiGIlK
    /SODP6rEsP9xAXnRx+sE1XEHl1GwTVla8lOInfYk1e6wXGRxn47EcEJw+lYV7dTM
    ksirrzBuMJz5fNsrF3MGrYWo6g4OAGPkxq8JS2CRUHU=
```

DoS protection

FortiOS includes built-in protection mechanisms in the Node.js–based web service to help secure the administrative GUI from denial-of-service (DoS) attacks. These mechanisms mitigate common attack patterns such as Slowloris and brute-force request flooding. DoS protection is provided through the following components:

- a request monitoring engine
- an HTTP rate-limiting system

Both components apply to all Node.js web service interfaces, including HTTP, HTTPS, and HTTP/2.

By enforcing request-handling timeouts and limiting excessive connection attempts, FortiOS improves the resilience of the GUI service, preventing malicious clients from consuming resources and ensuring that administrators can maintain access during periods of high load or attempted attacks.

HTTP request monitoring

The request monitor tracks all incoming HTTP connections and enforces configurable timeout thresholds for receiving complete HTTP headers and bodies. Connections that do not complete within the configured timeframe are discarded.

This mechanism prevents Slowloris-style attacks, where a client attempts to exhaust server resources by sending headers or body data extremely slowly.

CLI configuration:

Use the following commands to configure header and body timeout values:

```
config system global
    set admin-http-request-header-timeout <integer>
    set admin-http-request-body-timeout <integer>
    set admin-http-unauthenticated-request-body-timeout <integer>
end
```

Option	Description
`admin-http-request-header-timeout <integer>`	HTTP request header timeout, in milliseconds (0 will disable the timeout, default = 200).
`admin-http-request-body-timeout <integer>`	Authenticated HTTP request body timeout, in milliseconds (0 will disable the timeout, default = 1800000).
`admin-http-unauthenticated-request-body-timeout <integer>`	Unauthenticated HTTP request body timeout, in milliseconds, before authentication (0 will disable the timeout, default = 1000).

HTTP rate limiting

The HTTP rate limiter controls how many connections and requests a client IP can send to the GUI over time. Each client receives a token pool that regenerates gradually. Requests consume tokens. If a client exhausts its token pool, further requests are rejected until tokens recover. Rate-limiting applies to both authenticated and unauthenticated clients but can be configured to exempt authenticated administrators.

This mechanism helps control brute-force behavior and reduces the load caused by automated attacks.

CLI configuration:

Use the following commands to configure rate limiting and exemption:

```
config system global
    set admin-http-rate-limit-max-requests <integer>
```

```
    set admin-http-rate-limit-exempt-auth {enable | disable}
end
```

Option	Description
admin-http-rate-limit-max-requests <integer>	Maximum number of HTTP requests that are allowed to be made in a second by a single client (0 will disable the rate limiting, default = 100).
admin-http-rate-limit-exempt-auth {enable | disable}	Enable/disable exemption of authenticated administrator sessions from rate limiting. (default = disable).

Troubleshooting

FortiOS provides several diagnostic commands to inspect request monitor and rate limiter status.

View Node.js HTTP service status:

Displays status, configured timeouts, and rate limiter tokens.

```
#  diagnose web-svc server http state
Service: Request Monitor
   status              : enabled
   lastMonitored       : Wed, 18 Mar 2026 16:55:11 GMT
   headerTimeout       : 200ms
   bodyTimeout         : 1800000ms
   unauthBodyTimeout   : 1000ms
Service: HTTP Rate Limiter
   status              : enabled
   lastMonitored       : Wed, 18 Mar 2026 16:55:11 GMT
   maxTokens           : 5000
   exemptAuthed        : disabled
```

View active HTTP connections:

Displays the list of currently active HTTP requests and their associated timeout information. Requests are removed from the list automatically when they either time out or complete successfully. This command is available only to super_admin users.

```
#  diagnose web-svc server http client-connection
IP | Type | Creation | Timeout
   172.19.43.22 | 0 | Wed, 18 Mar 2026 16:55:11 GMT | Wed, 18 Mar 2026 16:55:11 GMT
```

View HTTP rate limiter token state:

Displays the list of tracked client IP addresses along with their token levels and current rate-limiter status. Clients that have full tokens and no violations are automatically removed from the list. This command is available only to super_admin users.

```
# diagnose web-svc server http rate-limit details
 IP                  | Tokens | Blocked | Bursting | Violations | Last Refill

 172.16.200.254      | 5000   | NO      | NO       | 0          | Wed, 18 Mar 2026 16:55:41 GMT
```

Clear rate limiter state:

Clears the specified IP address from the rate-limiter database, resetting all rate-limit state for that client. If no IP address is provided, the command clears the entire rate-limiter database.

```
# diagnose web-svc server http rate-limit clear <ip>
```

Configuration scripts

Configuration scripts are text files that contain CLI command sequences. They can be created using a text editor or copied from a CLI console, either manually or using the *Record CLI Script* function.

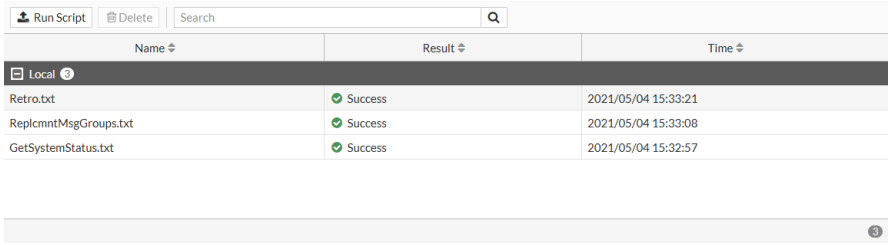
Scripts can be used to run the same task on multiple devices. For example, if your devices use the same security policies, you can enter or record the commands to create those policies in a script, and then run the script on each device. You could also create the policies in the GUI, and then copy and paste the CLI commands from the *CLI Console* using the *show* command.

If the FortiGate is managed by FortiManager, scripts can be uploaded to FortiManager and then run on any other FortiGates that are managed by that FortiManager. See Scripts in the FortiManager Administration Guide.

A comment line in a script starts with the number sign (#). Comments are not executed.

To run a script using the GUI:

1. Click on your username and select *Configuration* > *Scripts*.
2. Click *Run Script*.
3. Select the text file containing the script on your management computer, then click *OK*.
   The script runs immediately, and the *Script Execution History* table is updated, showing if the script ran successfully.



Workspace mode

Workspace mode allows administrators to make a batch of changes that are not implemented until the transaction is committed. Prior to committing, the changes can be reverted or edited as needed without impacting current operations.

When an object is edited in workspace mode it is locked, preventing other administrators from editing that object. A warning message will be shown to let the administrator know that the object is currently being configured in another transaction.

All administrators can use workspace mode; their permissions in workspace mode are the same as defined in their account profile.

A workspace mode transaction times out after five minutes if there is no activity. When a transaction times out, all changes are discarded. A warning message will be shown to let the administrator know that a timeout is imminent, or has already happened:

```
config transaction id=1 will expire in 30 seconds
config transaction id=1 will expire in 20 seconds
config transaction id=1 will expire in 10 seconds
config transaction id=1 has expired
```

The following commands are not changeable in a workspace transaction:

```
config system console
config system resource-limits
config system elbc
config system global
    set split-port
    set vdom-admin
    set management-vdom
    set wireless-mode
    set internal-switch-mode
end
config system settings
    set opmode
end
config system npu
config system np6
config system wireless
    set mode
end
config system vdom-property
config system storage
```

The `execute batch` command cannot be used in or to start workspace mode.

To use workspace mode:

1. Start workspace mode:

   ```
   execute config-transaction
   ```

Once in workspace mode, the administrator can make configuration changes, all of which are made in a local CLI process that is not viewable by other processes.

2. Commit configuration changes:

```
execute config-transaction commit
```

After performing the commit, the changes are available for all other processes, and are also made in the kernel.

3. Abort configuration changes:

```
execute config-transaction abort
```

If changes are aborted, no changes are made to the current configuration or the kernel.

Diagnose commands

- `diagnose sys config-transaction show txn-meta`

  Show config transaction meta information. For example:

```
# diagnose  sys  config-transaction  show txn-meta
txn_next_id=8, txn_nr=2
```

- `diagnose sys config-transaction show txn-info`

  Show config transaction information. For example:

```
# diagnose  sys  config-transaction  show txn-info
current_jiffies=680372

txn_id=6, expire_jiffies=706104, clicmd_fpath='/dev/cmdb/txn/6_EiLl9G.conf'
txn_id=7, expire_jiffies=707427, clicmd_fpath='/dev/cmdb/txn/7_UXK6wY.conf'
```

- `diagnose sys config-transaction show txn-entity`

  Show config transaction entity. For example:

```
# diagnose  sys  config-transaction  show txn-entity
vd='global', cli-node-oid=37(system.vdom), txn_id=7. location: fileid=0, storeid=0, pgnr=0,
pgidx=0
vd='global', cli-node-oid=46(system.interface), txn_id=7. location: fileid=3, storeid=0,
pgnr=0, pgidx=0
```

- `diagnose sys config-transaction show txn-lock`

  Show transaction lock status. For example:

```
# diagnose  sys  config-transaction  show txn-lock
type=-1, refcnt=0, value=256, pid=128
```

- `diagnose sys config-transaction status`

Show the transaction status in the current CLI.

RAID

Most FortiGate devices with multiple disk drives (SSD or HHD) can be configured to use RAID.

> ⚠ Enabling or disabling RAID, and changing the RAID level, erases all data on the log disk and reboots the device.

To verify that the FortiGate has multiple disks:

- List disk devices and partitions:

```
# execute disk list

Disk SSD1 ref: 255 223.6GiB type: SSD [ATA INTEL SSDSC2KB24] dev: /dev/sda
  partition ref: 1 220.1GiB, 219.0GiB free mounted: Y label: LOGUSEDXA707476A dev: /dev/sda1
start: 2048

Disk SSD2 ref: 16 223.6GiB type: SSD [ATA INTEL SSDSC2KB24] dev: /dev/sdb
  partition ref: 17 62.7GiB, 62.4GiB free mounted: Y label: WANOPTXX1FEBBFA1 dev: /dev/sdb1
start: 2048
  partition ref: 18 63.7GiB, 63.7GiB free mounted: N label: dev: /dev/sdb2 start: 133625856
  partition ref: 19 85.0GiB, 85.0GiB free mounted: N label: dev: /dev/sdb3 start: 267249664
```

- Display information about all of the disks:

```
# diagnose hardware deviceinfo disk

Disk SSD1 ref: 255 223.6GiB type: SSD [ATA INTEL SSDSC2KB24] dev: /dev/sda
  partition ref: 1 220.1GiB, 219.0GiB free mounted: Y label: LOGUSEDXA707476A dev: /dev/sda1
start: 2048

Disk SSD2 ref: 16 223.6GiB type: SSD [ATA INTEL SSDSC2KB24] dev: /dev/sdb
  partition ref: 17 62.7GiB, 62.4GiB free mounted: Y label: WANOPTXX1FEBBFA1 dev: /dev/sdb1
start: 2048
  partition ref: 18 63.7GiB, 63.7GiB free mounted: N label: dev: /dev/sdb2 start: 133625856
  partition ref: 19 85.0GiB, 85.0GiB free mounted: N label: dev: /dev/sdb3 start: 267249664

Disk SYSTEM(boot) 14.9GiB type: SSD [ATA 16GB SATA Flash] dev: /dev/sdc
  partition 247.0MiB, 155.0MiB free mounted: N label: dev: /dev/sdc1(boot) start: 1
  partition 247.0MiB, 154.0MiB free mounted: Y label: dev: /dev/sdc2(boot) start: 524289
  partition ref: 35 14.2GiB, 14.0GiB free mounted: Y label: dev: /dev/sdc3 start: 1048577

Disk USB-6(user-usb) ref: 48 28.6GiB type: USB [SanDisk Ultra] dev: /dev/sdd <<<<<<===this
info for usb disk because i have usb disk on FGT301E
  partition ref: 49 28.6GiB, 28.6GiB free mounted: Y label: dev: /dev/sdd1 start: 0
```

```
Total available disks: 4
Max SSD disks: 2 Available storage disks: 2
```

To check the RAID status:

- RAID enabled:

```
# execute disk raid status
RAID Level: Raid-1
RAID Status: OK (Background-Synchronizing) (9%)
RAID Size: 239GB

Disk 1: OK Used 228GB
Disk 2: OK Used 228GB
```

- RAID disabled:

```
# execute disk raid status
RAID Level: Unavailable
RAID Status: Unavailable
RAID Size: 0GB

Disk 1: OK Not-Used 228GB
Disk 2: OK Not-Used 228GB
```

To enable RAID:

```
# execute disk raid enable
This will erase all data on the log disk, and system will reboot!
Do you want to continue? (y/n)y

Dependent storage SSD2 removed.
Dependent storage SSD1 removed.
Raid-0 created with 2 disks.

Performing raid on the requested disk(s) and rebooting, please wait.. .

Configuring raid...
- unmounting /data2 : ok
- unmounting /var/log : ok
- unmounting /usb : ok
- unmounting /var/storage/SSD2-WANOPTXX0EA0EF17 : ok

Formatting the disk...
- unmounting /usb : ok
Formatting /dev/md0 ... done

The system is going down NOW !!

Please stand by while rebooting the system.
Restarting system.
```

To rebuild the RAID:

```
# execute disk raid rebuild
```

To rebuild the RAID to another level:

1. Check the supported RAID levels:

```
# execute disk raid rebuild-level
<RAID level> supported: Raid-0, Raid-1
```

2. Rebuild the RAID to the required level:

```
# execute disk raid rebuild-level Raid-1
This will erase all data on the log disk, and system will reboot!
Do you want to continue? (y/n)y

Dependent storage RAID removed.
Raid-1 created with 2 disks.

Performing raid on the requested disk(s) and rebooting, please wait...

Configuring raid...
- unmounting /data2 : ok
- unmounting /var/log : ok
- unmounting /usb : ok

Formatting the disk...
- unmounting /usb : ok
Formatting /dev/md0 ... done


The system is going down NOW !!

Please stand by while rebooting the system.
Restarting system.
```

To disable RAID:

```
# execute disk raid disable
This will erase all data on the log disk, and system will reboot!
Do you want to continue? (y/n)y

Dependent storage RAID removed.

Performing format on the requested disk(s) and rebooting, please wait...

Configuring raid...
- unmounting /data2 : ok
- unmounting /var/log : ok
- unmounting /usb : ok
```

```
Formatting the disk...
Partitioning and formatting /dev/sda label LOGUSEDX3D36836D ... done
Partitioning and formatting /dev/sdb label WANOPTXX1FEBBFA1 ...
Sending request for partno=0 start=2048 stop=133624230
Sending request for partno=1 start=133625856 stop=267248460
Sending request for partno=2 start=267249664 stop=445414150
done

The system is going down NOW !!

Please stand by while rebooting the system.
Restarting system.
FortiGate-301E (11:11-04.30.2018)
.
Reading boot image 3017355 bytes.
Initializing firewall...
System is starting...
```

FortiGate encryption algorithm cipher suites

FortiGates use SSL/TLS encryption for HTTPS and SSH administrative access, and Agentless VPN remote access. When establishing an SSL/TLS or SSH connection, you can control the encryption level and the ciphers that are used in order to control the security level.

HTTPS access

HTTP administrative access encryption is controlled using the following commands:

```
config system global
    set strong-crypto {enable | disable}
    set admin-https-ssl-versions {tlsv1-1 tlsv1-2 tlsv1-3}
    set admin-https-ssl-ciphersuites {<cipher_1> ... <cipher_n>}
    set admin-https-ssl-banned-ciphers {<cipher_1> ... <cipher_n>}
end
```

When strong encryption is enabled, only TLS 1.2 and TLS 1.3 are allowed. If strong encryption is then disabled, TLS 1.1 has to be manually enabled.

Setting `admin-https-ssl-ciphersuites` controls which cipher suites are offered in TLS 1.3. TLS 1.2 and lower are not affected by this command. To disable all TLS 1.3 cipher suites, remove TLS1-3 from `admin-https-ssl-versions`.

Setting `admin-https-ssl-banned-ciphers` controls which cipher technologies will not be offered for TLS 1.2 and lower.

Specific cipher suites are supported by each TLS version:

TLS version	Supported cipher suites	
TLS 1.1[1]	ECDHE-RSA-AES256-SHA[1]	AES256-SHA[1]
	ECDHE-RSA-AES128-SHA[1]	AES128-SHA[1]
TLS 1.2	ECDHE-RSA-AES256-GCM-SHA384	AES256-GCM-SHA384[1]
	ECDHE-RSA-AES128-GCM-SHA256	AES128-GCM-SHA256[1]
	ECDHE-RSA-CHACHA20-POLY1305	AES256-SHA256
	ECDHE-RSA-AES256-SHA384	AES128-SHA256
	ECDHE-RSA-AES128-SHA256	AES256-SHA[1]
	ECDHE-RSA-AES256-SHA[1]	AES128-SHA[1]
	ECDHE-RSA-AES128-SHA[1]	
TLS 1.3	TLS-AES-128-GCM-SHA256	TLS-AES-128-CCM-8-SHA256[1]
	TLS-AES-256-GCM-SHA384	TLS-CHACHA20-POLY1305-SHA256
	TLS-AES-128-CCM-SHA256[1]	

[1] Disabled if strong encryption (`strong-crypto`) is enabled.

SSH access

SSH access encryption is controlled using the following command:

```
config system global
    set admin-ssh-v1 {enable | disable}
    set strong-crypto {enable | disable}
end
```

```
config system ssh-config
    set ssh-enc-algo <algo_1> [<algo_2> ... <algo_n>]
    set ssh-hsk-algo <algo_1> [<algo_2> ... <algo_n>]
    set ssh-kex-algo <algo_1> [<algo_2> ... <algo_n>]
    set ssh-mac-algo <algo_1> [<algo_2> ... <algo_n>]
end
```

The algorithms available when configuring `set ssh-enc-algo` are affected by `set strong-crypto` as follows:

Strong encryption setting	Supported ciphers	
Enabled	aes256-gcm@openssh.com	aes256-ctr

Strong encryption setting	Supported ciphers	
Disabled	chacha20-poly1305@openssh.com	aes128-ctr
	aes192-ctr	aes256-ctr
	arcfour256	arcfour128
	aes128-cbc	3des-cbc
	blowfish-cbc	cast128-cbc
	aes192-cbc	aes256-cbc
	arcfour	rijndael-cbc@lysator.liu.se
	aes128-gcm@openssh.com	aes256-gcm@openssh.com

The following options are available for the `ssh-hsk-algo` algorithm based on the strong encryption setting:

Strong encryption setting	Supported ciphers	
Enabled	ecdsa-sha2-nistp521	ecdsa-sha2-nistp384
	ecdsa-sha2-nistp256	rsa-sha2-256
	rsa-sha2-512	ssh-ed25519
Disabled	ssh-rsa	ecdsa-sha2-nistp521
	ecdsa-sha2-nistp384	ecdsa-sha2-nistp256
	rsa-sha2-256	rsa-sha2-512
	ssh-ed25519	

The following options are available for the `ssh-kex-algo` algorithm based on the strong encryption setting:

Strong encryption setting	Supported ciphers	
Enabled	diffie-hellman-group14-sha256	diffie-hellman-group16-sha512
	diffie-hellman-group18-sha512	diffie-hellman-group-exchange-sha256
	curve25519-sha256@libssh.org	ecdh-sha2-nistp256
	ecdh-sha2-nistp384	ecdh-sha2-nistp521

Strong encryption setting	Supported ciphers	
Disabled	diffie-hellman-group14-sha1	diffie-hellman-group14-sha256
	diffie-hellman-group16-sha512	diffie-hellman-group18-sha512
	diffie-hellman-group-exchange-sha1	diffie-hellman-group-exchange-sha256
	curve25519-sha256@libssh.org	ecdh-sha2-nistp256
	ecdh-sha2-nistp384	ecdh-sha2-nistp521

The following options are available for the `ssh-mac-algo` algorithm based on the strong encryption setting:

Strong encryption setting	Supported ciphers	
Enabled	hmac-sha2-256	hmac-sha2-256-etm@openssh.com
	hmac-sha2-512	hmac-sha2-512-etm@openssh.com
Disabled	hmac-md5	hmac-md5-etm@openssh.com
	hmac-md5-96	hmac-md5-96-etm@openssh.com
	hmac-sha1	hmac-sha1-etm@openssh.com
	hmac-sha2-256	hmac-sha2-256-etm@openssh.com
	hmac-sha2-512	hmac-sha2-512-etm@openssh.com
	hmac-ripemd160	hmac-ripemd160@openssh.com
	hmac-ripemd160-etm@openssh.com	umac-64@openssh.com
	umac-128@openssh.com	umac-64-etm@openssh.com
	umac-128-etm@openssh.com	

Agentless VPN

For Agentless VPN connections, the TLS versions and cipher suites are controlled using the following commands:

```
config vpn ssl setting
    set algorithm {high | medium | low}
    set ssl-max-proto-ver {tls1-0 | tls1-1 | tls1-2 | tls1-3}
    set ssl-min-proto-ver {tls1-0 | tls1-1 | tls1-2 | tls1-3}
    set ciphersuite {TLS-AES-128-GCM-SHA256 TLS-AES-256-GCM-SHA384 TLS-CHACHA20-POLY1305-SHA256
TLS-AES-128-CCM-SHA256 TLS-AES-128-CCM-8-SHA256}
end
```

Only cipher suites supported by TLS 1.3 can be set using the `set ciphersuite` command.

When the Agentless VPN security level (`algorithm`) is set to high, only high levels are allowed. When it is set to medium, high and medium levels are allowed. When it is set to low, any level is allowed.

The strong encryption (`strong-crypto`) command has no effect on the Agentless VPN encryption level or ciphers.

Specific cipher suites are supported by each TLS version:

TLS version	Supported cipher suites	
TLS 1.0	ECDHE-RSA-AES256-SHA	DHE-RSA-CAMELLIA128-SHA
	DHE-RSA-AES256-SHA	AES128-SHA
	DHE-RSA-CAMELLIA256-SHA	SEED-SHA[1]
	AES256-SHA	CAMELLIA128-SHA
	CAMELLIA256-SHA	ECDHE-RSA-DES-CBC3-SHA[1*]
	ECDHE-RSA-AES128-SHA	EDH-RSA-DES-CBC3-SHA[1*]
	DHE-RSA-AES128-SHA[1]	DES-CBC3-SHA[1*]
	DHE-RSA-SEED-SHA	
TLS 1.1	ECDHE-RSA-AES256-SHA	DHE-RSA-CAMELLIA128-SHA
	DHE-RSA-AES256-SHA	AES128-SHA
	DHE-RSA-CAMELLIA256-SHA	SEED-SHA[1]
	AES256-SHA	CAMELLIA128-SHA
	CAMELLIA256-SHA	ECDHE-RSA-DES-CBC3-SHA[1*]
	ECDHE-RSA-AES128-SHA	EDH-RSA-DES-CBC3-SHA[1*]
	DHE-RSA-AES128-SHA	DES-CBC3-SHA[1*]
	DHE-RSA-SEED-SHA[1]	

TLS version	Supported cipher suites	
TLS 1.2	ECDHE-RSA-AES256-GCM-SHA384	ECDHE-RSA-AES128-SHA
	ECDHE-RSA-AES256-SHA384	DHE-RSA-AES128-GCM-SHA256
	ECDHE-RSA-AES256-SHA	DHE-RSA-AES128-CCM8
	DHE-RSA-AES256-GCM-SHA384	DHE-RSA-AES128-CCM
	ECDHE-RSA-CHACHA20-POLY1305	AES128-CCM8
	DHE-RSA-CHACHA20-POLY1305	AES128-CCM
	DHE-RSA-AES256-CCM8	DHE-RSA-AES128-SHA256
	DHE-RSA-AES256-CCM	DHE-RSA-AES128-SHA
	DHE-RSA-AES256-SHA256	ECDHE-RSA-CAMELLIA128-SHA256
	DHE-RSA-AES256-SHA	DHE-RSA-CAMELLIA128-SHA256
	ECDHE-RSA-CAMELLIA256-SHA384	DHE-RSA-SEED-SHA[1]
	DHE-RSA-CAMELLIA256-SHA256	DHE-RSA-CAMELLIA128-SHA
	DHE-RSA-CAMELLIA256-SHA	AES128-GCM-SHA256
	AES256-GCM-SHA384	AES128-SHA256
	AES256-CCM8	AES128-SHA
	AES256-CCM	CAMELLIA128-SHA256
	AES256-SHA256	SEED-SHA[1]
	AES256-SHA	CAMELLIA128-SHA
	CAMELLIA256-SHA256	ARIA128-GCM-SHA256
	CAMELLIA256-SHA	DHE-RSA-ARIA128-GCM-SHA256
	ARIA256-GCM-SHA384	ECDHE-ARIA128-GCM-SHA256
	DHE-RSA-ARIA256-GCM-SHA384	ECDHE-RSA-AES256-GCM-SHA384
	ECDHE-ARIA256-GCM-SHA384	ECDHE-RSA-DES-CBC3-SHA[1*]
	ECDHE-RSA-AES128-GCM-SHA256	EDH-RSA-DES-CBC3-SHA[1*]
	ECDHE-RSA-AES128-SHA256	DES-CBC3-SHA[1*]
TLS 1.3	TLS-AES-256-GCM-SHA384	TLS-AES-128-CCM-SHA256
	TLS-CHACHA20-POLY1305-SHA256	TLS-AES-128-CCM-8-SHA256
	TLS-AES-128-GCM-SHA256	

[1] This cipher is not available when the Agentless VPN security level (`algorithm`) is set to high.

[*] This cipher is not available when the Agentless VPN security level (`algorithm`) is set to medium.

Additional features

Security hardening for other FortiOS features can also be achieved by managing the encryption level or ciphers. See SSL algorithm security level on page 4042 for more information about different levels of security.

An asterisk (*) represents the default values for each feature.

Some of these features are described next.

WAN optimization

For WAN optimization tunnel connections, the encryption algorithm is controlled using the following commands:

```
config wanopt settings
    set tunnel-ssl-algorithm {high* | medium | low}
end
```

Explicit FTP proxy

For explicit FTP proxy, the encryption algorithm is controlled using the following commands:

```
config ftp-proxy explicit
    set ssl-algorithm { high* | medium | low}
end
```

Explicit web proxy

For explicit web proxy, the encryption algorithm is controlled using the following commands:

```
config web-proxy explicit
    set ssl-algorithm {high | medium | low*}
end
```

SSL Server

For SSL server, the TLS versions and the encryption algorithm are controlled using the following commands:

```
config firewall ssl-server
    edit <name>
        set ssl-algorithm {high* |medium | low}
        set ssl-max-version {tls-1.0* |tls-1.1 | tls-1.2 | tls-1.3}
        set ssl-min-version {tls-1.0 |tls-1.1 | tls-1.2 | tls-1.3*}
    next
end
```

VIP

For VIP, the TLS versions and the encryption algorithm are controlled using the following commands:

```
config firewall vip
        set ssl-max-version {ssl-3.0|tls-1.1 | tls1-2 | tls-1.3* | client}
        set ssl-min-version {ssl-3.0|tls-1.1* | tls1-2 | tls-1.3 |client}
        set ssl-algorithm {high* | medium | low | custom}
        config ssl-cipher-suites
            edit <priority>
                set cipher {TLS-AES-128-GCM-SHA256 | TLS-AES-256-GCM-SHA384|...}
                set versions {option1}, {option2}, ...
            next
        end
        set ssl-server-max-version [ssl-3.0|tls-1.1 | tls1-2 | tls-1.3 | client*}
        set ssl-server-min-version [ssl-3.0|tls-1.1 | tls1-2 | tls-1.3 | client*}

        set ssl-server-algorithm {high | medium | low | custom | client* }
        config ssl-server-cipher-suites
            edit <priority>
                set cipher {TLS-AES-128-GCM-SHA256 | TLS-AES-256-GCM-SHA384|...}
                set versions {option1}, {option2}, ...
            next
        end
    next
end
```

The command `config ssl-cipher-suites` is available only under certain conditions:

- When `set type` is set to either `server-load-balance` or `access-proxy`
- When `set ssl-algorithm` is set to `custom`

Similarly, the command `config ssl-server-cipher-suites` is available only under certain conditions:

- When `set type` is set to `server-load-balance`
- When `set ssl-mode` is set to `full`
- When `set ssl-algorithm` is set to `custom`

VoIP

For VoIP, the TLS versions and the encryption algorithm are controlled using the following commands:

```
config voip profile
      edit <name>
     config sip
        set ssl-algorithm {high* |medium | low}
        set ssl-max-version {ssl-3.0 | tls-1.0* |tls-1.1 | tls-1.2 | tls-1.3}
        set ssl-min-version {ssl-3.0 | tls-1.0 |tls-1.1 | tls-1.2 | tls-1.3*}
```

```
    next
end
```

SSL algorithm security level

Option	Description
high	High encryption. Allow only AES and ChaCha.
medium	Medium encryption. Allow AES, ChaCha, 3DES, and RC4.
low	Low encryption. Allow AES, ChaCha, 3DES, RC4, and DES.
*custom	Custom encryption. Use `ssl-server-cipher-suites` to select the cipher suites that are allowed.
*client	Use the same encryption algorithms for both client and server sessions.

The SSL algorithm security levels marked with an asterisk (*) are not supported across different FortiOS features.

Other Products

The security level of communication to and from FortiOS can be managed by controlling the encryption level and ciphers used. See Encryption algorithm security level on page 4044 for more information about different levels of security.

An asterisk (*) represents the default value for each product.

Some products that commonly interact with the FortiGate device are listed next.

syslog server

For syslog server, the TLS versions and the encryption algorithm are controlled using the following commands:

```
config log syslogd setting
    set enc-algorithm {high-medium | high | low | disable*}
    set ssl-min-proto-version {default* | SSLv3 | TLSv1 | TLSv1-1 | TLSv1-2}
end
config log syslogd override-setting
    set enc-algorithm {high-medium | high | low | disable*}
    set ssl-min-proto-version {default* | SSLv3 | TLSv1 | TLSv1-1 | TLSv1-2}
end
```

The command `ssl-min-proto-version` set to `default` means that the system global setting will be followed.

FortiCloud

For logging to FortiCloud, the TLS versions and the encryption algorithm are controlled using the following commands:

```
config log fortiguard setting
    set enc-algorithm {high-medium | high* | low}
    set ssl-min-proto-version {default* | SSLv3 | TLSv1 | TLSv1-1 | TLSv1-2}
end
```

FortiAnalyzer Cloud

For FortiAnalyzer Cloud, the TLS versions and the encryption algorithm are controlled using the following commands:

```
config log fortianalyzer-cloud setting
    set enc-algorithm {high-medium | high* | low}
    set ssl-min-proto-version {default* | SSLv3 | TLSv1 | TLSv1-1 | TLSv1-2 | TLSv1-3}
end
```

FortiAnalyzer

For FortiAnalyzer, the TLS versions and the encryption algorithm are controlled using the following commands:

```
config log fortianalyzer setting
    set enc-algorithm {high-medium | high* | low}
    set ssl-min-proto-version {default* | SSLv3 | TLSv1 | TLSv1-1 | TLSv1-2 | TLSv1-3}
end
config log fortianalyzer override-setting
    set enc-algorithm {high-medium | high* | low}
    set ssl-min-proto-version {default* | SSLv3 | TLSv1 | TLSv1-1 | TLSv1-2 | TLSv1-3}
end
```

FortiSandbox

For FortiSandbox, the TLS versions and the encryption algorithm are controlled using the following commands:

```
config system fortisandbox
    set enc-algorithm {default* | high | low}
    set ssl-min-proto-version {default* | SSLv3 | TLSv1 | TLSv1-1 | TLSv1-2}
end
```

FortiManager

For central management, the TLS versions and the encryption algorithm are controlled using the following commands:

```
config system central-management
    set enc-algorithm {default | high* | low}
end
```

Encryption algorithm security level

Option	Description
*high-medium/ default	SSL communication with high and medium encryption algorithms.
high	SSL communication with high encryption algorithm
low	SSL communication with low encryption algorithms.
*disable	Disable SSL communication.

Encryption algorithm security levels marked with an asterisk (*) are not supported across different products.

Conserve mode

Each FortiGate model has a specific amount of memory that is shared by all operations. If most or all of that memory is in use, system operations can be affected in unexpected ways. To control how FortiOS functions when the available memory is very low, FortiOS enters conserve mode. This causes functions, such as antivirus scanning, to change how they operate to reduce the functionality and conserve memory without compromising security.

Three memory thresholds can be configured:

```
config system global
    set memory-use-threshold-extreme <integer>
    set memory-use-threshold-green <integer>
    set memory-use-threshold-red <integer>
end
```

`memory-use-threshold-extreme <integer>`	The threshold at which memory usage is considered extreme and new sessions are dropped, in percent of total RAM (70 - 97, default = 95).
`memory-use-threshold-green <integer>`	The threshold at which memory usage forces the FortiGate to leave conserve mode, in percent of total RAM (70 - 97, default = 82).

`memory-use-threshold-red <integer>`	The threshold at which memory usage forces the FortiGate to enter conserve mode, in percent of total RAM (70 - 97, default = 88).

Proxy inspection in conserve mode

The FortiGate's proxy-based inspection behavior while in conserve mode is configured with the antivirus failopen command.

```
config system global
    set av-failopen {pass | off | one-shot}
end
```

`pass`	This is the default settings. Bypass the antivirus proxy and allow traffic to continue to its destination. Because traffic bypasses the proxy, security profiles that require the antivirus proxy are also bypassed. Security profiles that do not use the antivirus proxy continue to function normally. Use this setting when access is more important than security while the issue is resolved.
`off`	Force the FortiGate to stop all traffic that uses the antivirus proxy. New sessions are blocked, but active sessions continue to be processed normally unless they request more memory and are then terminated. If a security policy is configured to use antivirus scanning, then the traffic that it permits is blocked while in conserve mode. So, a policy with only IPS scanning enabled will continue normally, but a policy with both IPS and antivirus scanning is blocked because antivirus scanning requires the antivirus proxy. Use this setting when security is more important than access while the issue is resolved.
`one-shot`	Continue to bypass the antivirus proxy after the FortiGate is out of conserve mode, until the failopen setting is changed or the FortiGate is restarted.

Flow inspection in conserve mode

The FortiGate's flow-based inspection behavior while in conserve mode is configured with the IPS failopen command.

```
config ips global
    set fail-open {enable | disable}
end
```

- When disabled (default), the IPS engine drops all new sessions that require flow-based inspection.
- When enabled, the IPS engine does not perform any scans and allows new packets.

Diagnostics

When in conserve mode, FortiOS generates conserve mode log messages and SNMP traps, and a conserve mode banner is shown in the GUI.

To view current information about memory conservation status:

```
# diagnose hardware sysinfo conserve
memory conserve mode:                          on
total RAM:                                        997 MB
memory used:                                      735 MB   73% of total RAM
memory freeable:                                  173 MB   17% of total RAM
memory used + freeable threshold extreme:         947 MB   95% of total RAM
memory used threshold red:                        877 MB   88% of total RAM
memory used threshold green:                      817 MB   82% of total RAM
```

To view logs:

1. Go to *Log & Report* > *System Events* in the GUI.
2. If historical FortiView is enabled, select the *Logs* tab.
3. If the GUI is unresponsive due to high memory usage, making the logs inaccessible, they can be viewed in the CLI:

```
# execute log filter category 1
# execute log display
```

```
1: date=2022-11-02 time=16:58:37 eventtime=1667433517502192693 tz="-0700" logid="0100022011"
type="event" subtype="system" level="critical" vd="root" logdesc="Memory conserve mode
entered" service="kernel" conserve="on" total=997 MB used=707 MB red="877 MB" green="698 MB"
msg="Kernel enters memory conserve mode
```

To view the crash log in the CLI:

```
# diagnose debug crashlog read
```

```
1: 2022-10-27 14:22:36 service=kernel conserve=on total="997 MB" used="720 MB" red="877 MB"
2: 2022-10-27 14:22:36 green="817 MB" msg="Kernel enters memory conserve mode"
```

Using APIs

Administrators can use API calls to a FortiGate to:

- Retrieve, create, update, and delete configuration settings
- Retrieve system logs and statistics
- Perform basic administrative actions, such as a reboot or shut down through programming scripts.

Token-based authentication

FortiGate supports only token-based authentication for API calls. Token-based authentication requires the administrator to generate a token, which is then included in each API request for authentication. A token is automatically generated when a new API administrator is created in FortiOS.

> ⚠ Once the API administrator is created and the token displays, there is no way for the FortiGate to provide this token again. Ensure you record the token, and store it in a safe location; otherwise, you will have to generate a new token.

Creating the API administrator and generating the API token

When creating an API administrator, it is best practice to provide this account (and the associated token) with the minimum permissions required to complete the function. For example, if you only plan to use API calls to retrieve statistics or information from the FortiGate, the account should have read permissions.

> 💡 The API administrator account used in this topic's examples has full permissions strictly to illustrate various call types and does not adhere to the preceding recommendation.

See REST API administrator on page 3712 for detailed steps to create a REST API administrator.

Best Practices

Using API tokens with a request header

The API token can be included in any REST API request in either the request header or URL parameter. For added security, it is strongly recommended to use API tokens in the request header or transition your applications to include the API token in the request header instead of the URL parameter.

To pass the API token in the request header, the user needs to explicitly add the following field to the request header:

```
Authorization: Bearer <YOUR-API-TOKEN>
```

If the API token must be added to the URL query parameter, then enable the following global setting:

```
config system global
    set rest-api-key-url-query enable
end
```

Configuring security options when creating a REST API administrator

In addition to using API tokens in the request header, for added security when creating a new REST API administrator, one or more of the following fields should be configured, listed in order of configuration difficulty from easy to difficult:

- **Trusted Hosts**
  To ensure that only trusted hosts/subnets can access the FortiGate REST API, you should configure the *Trusted Hosts* field when creating a new REST API administrator. You need your *Source Address* to create the trusted host.
- **CORS Allow Origin**
  Cross Origin Resource Sharing (CORS) allows third-party web apps to make API requests to the FortiGate using the token. For this field, specify the `Access-Control-Allow-Origin` on API responses. If possible, avoid using '*'.
- **PKI Group**
  Configure a PKI group to enable peer authentication using certificate matching, which provides an extra layer of security. Both the client certificate and token must match to be granted access to the API. The PKI group is typically specified as the name of a peer group defined under `config user group` which has PKI members. See Configuring a PKI user on page 3662 for details.

Configuring token expiry

It is possible and a good practice to configure an expiry time for an access token. To do this, after a REST API administrator is created, use the following `execute` command to regenerate the API token and specify an expiry time from 1 - 10080 minutes:

```
# execute api-user generate-key <API username> [expiry in minutes]
```

Making an API call to retrieve information from the FortiGate

The newly created API token is used to query the FortiGate for all firewall addresses. Many applications can be used for this query; this example uses the curl and Postman clients to demonstrate the functionality.

To ensure maximum security when using API tokens, HTTPS is enforced. HTTP cannot be used.

Replace the placeholders below with values for your FortiGate:

- `<FortiGate_address>` is the IP address or hostname of your FortiGate as well as the HTTPS port number (default = 443 and does not need to be explicitly specified).
- `<API-TOKEN>` is the token you generated.

General API call

One of the simplest API calls is `api/v2/cmdb/firewall/address`, which returns all information about all firewall addresses.

To make a general API call using curl:

```
curl --insecure \
-H "Accept: application/json" \
-H "Authorization: Bearer <API-TOKEN>" https://<FortiGate_address>/api/v2/cmdb/firewall/address
```

The backslash (\) allows for multiline commands in curl. You can choose to enter the backslashes or enter the commands into a single wrapping line.

To make a general API call using Postman:

1. Open the Postman client.
2. Go to *Settings* > *General* and turn off *SSL certificate verification*.
3. Click on *New* and select *HTTP*
4. In the new request, click on the *Authorization* tab, select *Type* as *Bearer Token* and enter `<YOUR-API-TOKEN>` in the *Token* field.
5. Enter a URL like the one below:

   ```
   https://<YOUR-FORTIGATE-ADDRESS>/api/v2/cmdb/firewall/address
   ```

6. Click *Send*.

Results of the general API call:

curl and Postman display the output similar to the following (output shortened for brevity):

```
{
  "http_method": "GET",
  "size": 17,
  "limit_reached": false,
  "matched_count": 17,
  "next_idx": 16,
  "revision": "bd002ee1735120907182831e7528dc8b",
  "results": [
      {
          "name": "EMS_ALL_UNKNOWN_CLIENTS",
          "q_origin_key": "EMS_ALL_UNKNOWN_CLIENTS",
          "uuid":"********-****-****-****-************"
          "type": "dynamic",
          "route-tag": 0,
          "sub-type": "ems-tag",
          "clearpass-spt": "unknown",
          "macaddr": [],
          "country": "",
          "cache-ttl": 0,
```

```
            "sdn": "",
            "fsso-group": [],
            "interface": "",
            "obj-tag": "",
            "obj-type": "ip",
            "tag-detection-level": "",
            "tag-type": "",
            "dirty": "clean",
            "hw-vendor": "",
            "hw-model": "",
            "os": "",
            "sw-version": "",
            "comment": "",
            "associated-interface": "",
            "color": 0,
            "filter": "",
            "sdn-addr-type": "private",
            "node-ip-only": "disable",
            "obj-id": "",
            "list": [],
            "tagging": [],
            "allow-routing": "disable",
            "fabric-object": "disable"
        },
        {
            "name": "EMS_ALL_UNMANAGEABLE_CLIENTS",
            "q_origin_key": "EMS_ALL_UNMANAGEABLE_CLIENTS",
            "uuid":"********-****-****-****-************",
```


Formatting an API call

Since a general API call for address objects returns a large amount of information, it may be beneficial to format the API call to display certain information using the `format` parameter. In this example, the `format` parameter is used to display the name and comment for each firewall address.

To use the format parameter in an API call using curl:

```
curl --insecure \
-H "Accept: application/json" \
-H "Authorization: Bearer <API-TOKEN>" https://<FortiGate_
address>/api/v2/cmdb/firewall/address?format="name|comment"
```

The backslash (\) allows for multiline commands in curl. You can choose to enter the backslashes or enter the commands into a single wrapping line.

To use the format parameter in an API call using Postman:

1. Open the Postman client.
2. Go to *Settings* > *General* and turn off *SSL certificate verification*.
3. Click on *New* and select *HTTP*

4. In the new request, click on the *Authorization* tab, select *Type* as *Bearer Token* and enter `<YOUR-API-TOKEN>` in the *Token* field.
5. Enter a URL like the one below:

```
https://<FortiGate_address>/api/v2/cmdb/firewall/address/? format=name|comment.
```

6. Click *Send*.

Results of the formatted API call:

curl and Postman display the output similar to the following (output shortened for brevity):


```
{
  "http_method": "GET",
  "size": 17,
  "limit_reached": false,
  "matched_count": 17,
  "next_idx": 16,
  "revision": "bd002ee1735120907182831e7528dc8b",
  "results": [
      {
          "name": "EMS_ALL_UNKNOWN_CLIENTS",
          "q_origin_key": "EMS_ALL_UNKNOWN_CLIENTS",
          "comment": ""
      },
      {
          "name": "EMS_ALL_UNMANAGEABLE_CLIENTS",
          "q_origin_key": "EMS_ALL_UNMANAGEABLE_CLIENTS",
          "comment": ""
      },
      {
          "name": "FABRIC_DEVICE",
          "q_origin_key": "FABRIC_DEVICE",
          "comment": "IPv4 addresses of Fabric Devices."
      },
```


Filtering an API call

The filter parameter can be used to specify a field and a keyword to limit what results match and are returned by a call. In this example, the preceding call is used with a filter to return only names and comments for address objects with the word `Sales` in the name.

To use the filter parameter in an API call using curl:

```
curl --insecure \
-H "Accept: application/json" \
-H "Authorization: Bearer <API-TOKEN>" https://<FortiGate_
address>/api/v2/cmdb/firewall/address?format="name|comment&filter=name=@SSLVPN"
```

The backslash (\) allows for multiline commands in curl. You can choose to enter the backslashes or enter the commands into a single wrapping line.

To use the filter parameter in an API call using Postman:

1. Open the Postman client.
2. Go to *Settings* > *General* and turn off *SSL certificate verification*.
3. Click on *New* and select *HTTP*
4. In the new request, click on the *Authorization* tab, select *Type* as *Bearer Token* and enter `<YOUR-API-TOKEN>` in the *Token* field.
5. Enter a URL like the one below:

```
https://<FortiGate_address>/api/v2/cmdb/firewall/address/?
format=name|comment&filter=name=@SSLVPN.
```

6. Click *Send*.

Results of the formatted API call:

curl and Postman display the output similar to the following (output shortened for brevity):

```
{
  "http_method": "GET",
  "size": 17,
  "limit_reached": false,
  "matched_count": 1,
  "next_idx": 5,
  "revision": "bd002ee1735120907182831e7528dc8b",
  "results": [
      {
          "name": "SSLVPN_TUNNEL_ADDR1",
          "q_origin_key": "SSLVPN_TUNNEL_ADDR1",
          "comment": ""
      }
  ],
  "vdom": "root",
  "path": "firewall",
  "name": "address",
  "status": "success",
  "http_status": 200,
  "serial": "****************",
  "version": "******",
  "build": ****}
```

For a complete list of API calls, see the Fortinet Development Network (FNDN). A subscription is required to access the FNDN.

Configuration backups and reset

Once you successfully configure the FortiGate, it is extremely important that you back up the configuration. In some cases, you may need to reset the FortiGate to factory defaults or perform a TFTP upload of the firmware, which will erase the existing configuration. In these instances, the configuration on the device must be recreated, unless a backup can be used to restore it.

You can use the GUI or CLI to back up the configuration in FortiOS or YAML format. You have the option to save the configuration file in FortiOS format to various locations including the local PC, USB key, FTP, and TFTP server. FTP and TFTP are only configurable through the CLI. In YAML format, configuration files can be backed up or restored on an FTP or TFTP server through the CLI.

This topic includes the following information:

- Backing up and restoring configurations from the GUI on page 4053
- Backing up and restoring configurations from the CLI on page 4056
- Configuration revision on page 4060
- Restore factory defaults on page 4061
- Secure file copy on page 4062

Administrators can back up a configuration file when using an admin profile with access permissions for *System* set to *Read/Write*. Therefore, administrators using admin profiles with access permissions for *System* set to *Read* cannot back up a config file from the FortiGate or through SCP.

For more granularity within the admin profile, set access permission for *System* to *Custom* to access additional fields. When *System* is set to *Custom*, the *Administrator Users* field dictates whether the config file can be backed up in the GUI. Whereas the *Configuration* field dictates whether the config file can be backed up in the CLI.

Backing up and restoring configurations from the GUI

Configurations can be backed up using the GUI to your PC or a USB disk.

Field	Description
Scope	When the FortiGate is in multi-vdom mode and a user is logged in as a global administrator.
Backup to	You can choose where to save the configuration backup file. • *Local PC*: Save the configuration file to your PC. • *USB Disk*: Save the configuration file to an external USB disk. This option is not available if there is no USB drive inserted in the USB port. You can also back up to FortiManager using the CLI.
File format	The configuration file can be saved in FortiOS or YAML format.

Field	Description
Password mask	Use password masking when sending a configuration file to a third party. When password masking is enabled, passwords and secrets will be replaced in the configuration file with `FortinetPasswordMask`.
Encryption	Enable *Encryption* to encrypt the configuration file. A configuration file cannot be restored on the FortiGate without a set password. Encryption is performed using AES-GCM algorithm.

To back up the configuration in FortiOS format using the GUI:

1. Click on the user name in the upper right-hand corner of the screen and select *Configuration* > *Backup*.
2. Direct the backup to your *Local PC* or to a *USB Disk*.
   The *USB Disk* option will not be available if no USB drive is inserted in the USB port. You can also backup to the FortiManager using the CLI.
3. If VDOMs are enabled, indicate whether the scope of the backup is the entire FortiGate configuration (*Global*) or only a specific VDOM configuration (*VDOM*).
   If backing up a VDOM configuration, select the VDOM name from the list.
4. Enable *Encryption*.

   This is recommended to secure your backup configurations and prevent unauthorized parties from reloading your configuration.

5. Enter a password, and enter it again to confirm it. This password will be required to restore the configuration.
6. Click *OK*.
7. When prompted, select a location on the PC or USB disk to save the configuration file. The configuration file will have a .conf extension.

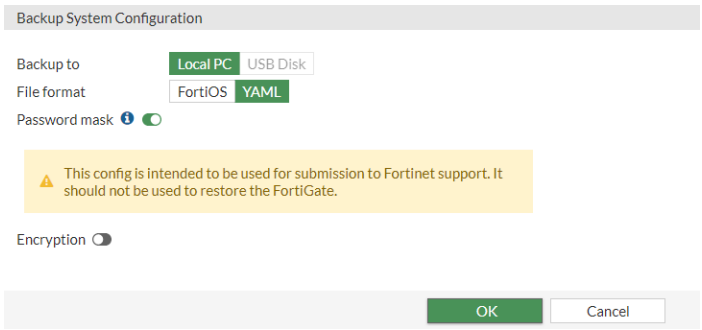
To back up the configuration in YAML format using the GUI:

1. Click on the user name in the upper right-hand corner of the screen and select *Configuration* > *Backup*.
2. Direct the backup to your *Local PC* or to a *USB Disk*.
3. Select *YAML* for the *File format*.
4. Click *OK*.

When backing up a configuration that will be shared with a third party, such as Fortinet Inc. Support, passwords and secrets should be obfuscated from the configuration to avoid information being unintentionally leaked. Password masking can be completed in the *Backup System Configuration* page and in the CLI. When password masking is enabled, passwords and secrets will be replaced in the configuration file with `FortinetPasswordMask`.

To mask passwords in the GUI:

1. Click on the username in the upper right-hand corner of the screen and select *Configuration* > *Backup*.
2. Select *YAML* as the *File format*.
3. Enable *Password mask*. A warning message is displayed.



4. Click *OK*. The configuration file is saved to your computer with passwords and secrets obfuscated.

The following is an example of output with password masking enabled:

```
config system admin
    edit "1"
        set accprofile "prof_admin"
        set vdom "root"
        set password FortinetPasswordMask
    next
end
config vpn ipsec phase1-interface
    edit "vpn-1"
        set interface "port1"
        set peertype any
        set net-device disable
        set proposal aes128-sha256 aes256-sha256 aes128-sha1 aes256-sha1
        set comments "VPN: vpn-1 (Created by VPN wizard)"
        set wizard-type static-fortigate
        set remote-gw 172.16.200.55
        set psksecret FortinetPasswordMask
    next
end
config wireless-controller vap
    edit "ssid-1"
        set passphrase FortinetPasswordMask
        set schedule "always"
    next
end
```

Restoring configuration files from the GUI

Configuration files can be used to restore the FortiGate to a previous configuration in the *Restore System Configuration* page.

To restore the FortiGate configuration using the GUI:

1. Click on the user name in the upper right-hand corner of the screen and select *Configuration* > *Restore*.
2. Identify the source of the configuration file to be restored: your *Local PC* or a *USB Disk*.
   The *USB Disk* option will not be available if no USB drive is inserted in the USB port. You can restore from the FortiManager using the CLI.

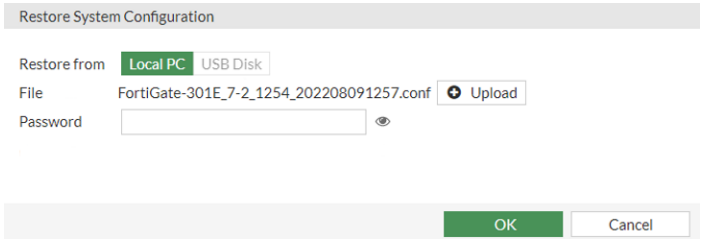
3. Click *Upload*, locate the configuration file, and click *Open*.
4. Enter the password if required.
5. Click *OK*.

When restoring a configuration file that has password masking enabled, obfuscated passwords and secrets will be restored with the password mask.

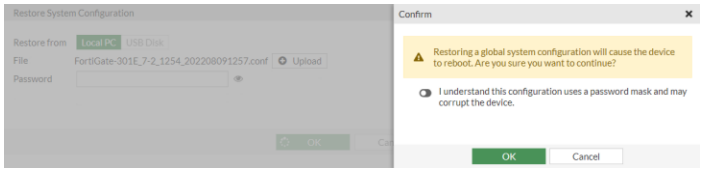
Restoring the FortiGate with a configuration with passwords obfuscated is not recommended.

To restore an obfuscated YAML configuration using the GUI:

1. Click on the user name in the upper right-hand corner of the screen and select *Configuration* > *Restore*.
2. Click *Upload*. The File Explorer is displayed.
3. Navigate to the configuration file and click *Open*.



4. (Optional) Enter the file password in the *Password* field.
5. Click *OK*. The *Confirm* pane is displayed with a warning.



6. Toggle the acknowledgment.
7. Click *OK*.

Backing up and restoring configurations from the CLI

Configuration backups in the CLI are performed using the `execute backup` commands and can be backed up in FortiOS and YAML format.

Configuration files can be backed up to various locations depending on the command:

- `flash`: Backup the configuration file to the flash drive.
- `ftp`: Backup the configuration file to an FTP server.
- `management-station`: Backup the configuration file to a management station, such as FortiManager or FortiGate Cloud.
- `sftp`: Backup the configuration file to a SFTP server.
- `tftp`: Backup the configuration file to a TFTP server.
- `usb`: Backup the configuration file to an external USB drive.
- `usb-mode`: Backup the configuration file for USB mode.

Command	Description
`# execute backup config`	Back up the configuration in FortiOS format. Backup your configuration file to: • `flash` • `ftp` • `management-station` • `sftp` • `tftp` • `usb` • `usb-mode`
`# execute backup full-config`	Backup the configuration, including backups of default configuration settings. Backup your configuration file to: • `ftp` • `sftp` • `tftp` • `usb` • `usb-mode`
`# execute backup yaml-config`	Backup the configuration in YAML format. Backup your configuration file to: • `ftp` • `tftp`
`# execute backup obfuscated-config`	Backup the configuration with passwords and secrets obfuscated. Backup your configuration file to: • `ftp` • `management-station` • `sftp` • `tftp` • `usb`
`# execute backup obfuscated-full-config`	Backup the configuration (including default configuration settings) with passwords and secrets obfuscated. Backup your configuration file to: • `ftp` • `sftp` • `tftp` • `usb`
`# execute backup obfuscated-yaml-config`	Backup the configuration in YAML format with passwords and secrets obfuscated. Backup your configuration file to: • `ftp` • `tftp`

To back up the configuration in FortiOS format using the CLI:

For FTP, note that port number, username are optional depending on the FTP site:

```
# execute backup config ftp <backup_filename> <ftp_server>[<:ftp_port>] [<user_name>] [<password>]
[<backup_password>]
```

or for TFTP:

```
# execute backup config tftp <backup_filename> <tftp_servers> [<backup_password>]
```

or for SFTP:

```
# execute backup config sftp <backup_filename> <sftp_server>[<:sftp_port>] <user> <password>
[<backup_password>]
```

or:

```
# execute backup config management-station <comment>
```

or:

```
# execute backup config usb <backup_filename> [<backup_password>]
```

Use the same commands to backup a VDOM configuration by first entering the commands:

```
config vdom
    edit <vdom_name>
```

See Backing up and restoring configurations in multi-VDOM mode on page 3849 for more information.

When backing up a configuration in YAML format, if it is not already specified in the file name, `.yaml` will be appended to the end. For example, if the file name entered is `301E.conf`, the name will become `301E.conf.yaml` after the configuration is backed up.

To back up the configuration in YAML format using the CLI:

```
# execute backup yaml-config {ftp | tftp} <filename> <server> [username] [password]
```

For example:

```
# execute backup yaml-config  tftp  301E.conf 172.16.200.55
    Please wait...
    The suffix '.yaml' will be appended to the filename if user does not add it specifically.
    Connect to tftp server 172.16.200.55 ...
    #
    Send config file to tftp server OK.
```

Configuration files can be configured with obfuscated passwords and secrets to not unintentionally leak information when sharing configuration files with third parties.

To mask passwords in a configuration backup in the CLI:

```
# execute backup obfuscated-config {ftp | management-station | sftp | tftp | usb}
```

To mask passwords in the full configuration backup in the CLI:

```
# execute backup obfuscated-full-config {ftp | sftp | tftp | usb}
```

To mask passwords in a configuration backup with YAML formatting in the CLI:

```
# execute backup obfuscated-yaml-config {ftp | tftp}
```

If a configuration is being backed up on a server, server information must be included with the command. Other information that may be required with an `execute backup` command includes file names, passwords, and comments.

Restoring configuration files from the CLI

Configuration files can be used to restore the FortiGate using the CLI.

Command	Description
`# execute restore config`	Restore a configuration that is in FortiOS or YAML format. The file format is automatically detected when it is being restored. Configurations can be loaded from: • `flash`: Load the configuration file from flash to firewall. • `ftp`: Load the configuration file from an FTP server. • `management-station`: Load the configuration from a management station. • `tftp`: Load the configuration from from a TFTP server. • `usb`: Load the configuration file from an external USB disk to firewall. • `usb-mode`: Load the configuration file from an external USB disk and reboot.

To restore the FortiGate configuration using the CLI:

For FTP, note that port number, username are optional depending on the FTP site:

```
# execute restore config ftp <backup_filename> <ftp_server>[<:port>] [<user_name>] [<password>]
[<backup_password>]
```

or for TFTP:

```
# execute restore config tftp <backup_filename> <tftp_server> [<backup_password>]
```

For restoring the configuration from FortiManager or FortiGate Cloud:

```
# execute restore config management-station normal <revision ID>
```

or:

```
# execute restore config usb <backup_filename> [<backup_password>]
```

The FortiGate will load the configuration file and restart. Once the restart has completed, verify that the configuration has been restored.

Troubleshooting

When restoring a configuration, errors may occur, but the solutions are usually straightforward.

Error message	Reason and Solution
Configuration file error	This error occurs when attempting to upload a configuration file that is incompatible with the device. This may be due to the configuration file being for a different model or being saved from a different version of firmware. **Solution**: Upload a configuration file that is for the correct model of FortiGate device and the correct version of the firmware.
Invalid password	When the configuration file is saved, it can be protected by a password. The password entered during the upload process is not matching the one associated with the configuration file. **Solution**: Use the correct password if the file is password protected.

Configuration revision

You can manage multiple versions of configuration files on models that have a 512MB flash memory and higher. Revision control requires either a configured central management server or the local hard drive, if your FortiGate has this feature. Typically, configuration backup to local drive is not available on lower-end models.

Central management server

The central management server can either be a FortiManager unit or FortiGate Cloud.

If central management is not configured on your FortiGate, a message appears instructing you to either enable central management, or obtain a valid license.

To enable central management from the GUI:

1. Go to *Security Fabric > Fabric Connectors* and double-click the *Central Management* card.
2. Set the *Status* to *Enabled* and select a *Type*.
3. Click *OK*.

To enable central management from the CLI:

```
config system central-management
    set type {fortimanager | fortiguard}
    set mode backup
    set fmg <IP address>
end
```

To backup to the management server:

```
# execute backup config management-station <comment>
```

To view a backed up revision:

```
# execute restore config management-station normal 0
```

To restore a backed up revision:

```
# execute restore config management-station normal <revision ID>
```

Backing up to a local disk

When revision control is enabled on your FortiGate unit, and configuration backups have been made, a list of saved revisions of those backed-up configurations appears.

Configuration backup occurs by default with firmware upgrades but can also be configured to occur every time you log out.

To configure configuration backup when logging out:

```
config system global
  set revision-backup-on-logout enable
end
```

To manually force backup:

```
# execute backup config flash <comment>
```

Configuration revisions are viewed by clicking on the user name in the upper right-hand corner of the screen and selecting *Configuration* > *Revisions*.

To view a list of revisions backed up to the disk from the CLI:

```
# execute revision list config
```

To restore a configuration from the CLI:

```
# execute restore config flash <revision ID>
```

Restore factory defaults

There may be a need to reset the FortiGate to its original defaults; for example, to begin with a fresh configuration. There are two options when restoring factory defaults:

`# execute factoryreset`	Reset the device to factory default configuration. The firmware version and antivirus and IPS attack definitions are not changed.
`# execute factoryreset2`	Reset to factory default configuration without losing management access to the FortiGate. Interface and VDOM configurations, as well as the firmware version and antivirus and IPS attack definitions, are not changed.

Secure file copy

You can also back up and restore your configuration using Secure File Copy (SCP).

You enable SCP support using the following commands:

```
config system global
    set admin-scp enable
end
```

```
config system interface
    edit <interface>
        append allowaccess ssh
    next
end
```

For more information about this command and about SCP support, see config system global.

Once enabled, on your SCP client, run these commands (examples are based on Linux):

Command	Description
`scp -O <admin-user>@<FGT_IP>:sys_config <dst file>`	Backup a config file. Use the `-O` option for legacy SCP protocol.
`scp -O <admin-user>@<FGT_IP>:encrypted-config:<encryption_password> <dst file>`	Backup a config file with encryption password.
`scp -O <src config file> <admin-user>@<FGT_IP>:fgt-restore-config`	Restore config file to FortiGate. FortiGate will reboot immediately.
`scp -O <src config file> <admin-user>@<FGT_IP>:fgt-restore-config:<encryption_password>`	Restore config file with encryption password to FortiGate. FortiGate will reboot immediately.
`scp <firmware.out> <admin-user>@< FGT_IP>:fgt-image`	Restore a firmware. FortiGate will reboot immediately.

An example of an admin trying to backup a FortiGate configuration without a password may look like this:

```
$ scp -O admin@172.18.27.95:sys_config 95-nopasswd
sys_config 100% 416KB 1.3MB/s 00:00
```

An admin trying to restore a FortiGate configuration with a password may look like this:

```
$ scp -O Downloads/95-password admin@172.18.27.95:fgt-restore-config:123456
95-password 100% 77KB 32.7MB/s 00:00
```

For other operating systems and command examples, see How to download a FortiGate configuration file and upload a firmware file using secure copy (SCP).

Fortinet Support Tool for capturing incidents

The Fortinet Support Tool application is used to capture real-time debugging information through a REST API key generated directly on the FortiGate device. It can be installed on Windows from the Microsoft Store and on macOS from the App store.

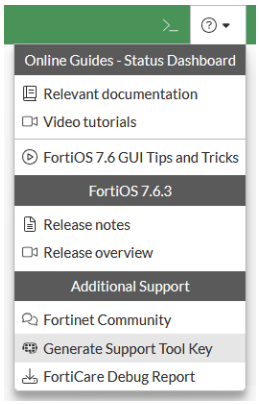
The program can run in the background for up to 48 hours. This increases the likelihood that it is active during an incident, allowing administrators to gather comprehensive logs and ensuring faster and more efficient troubleshooting.

To install the Fortinet Support Tool:

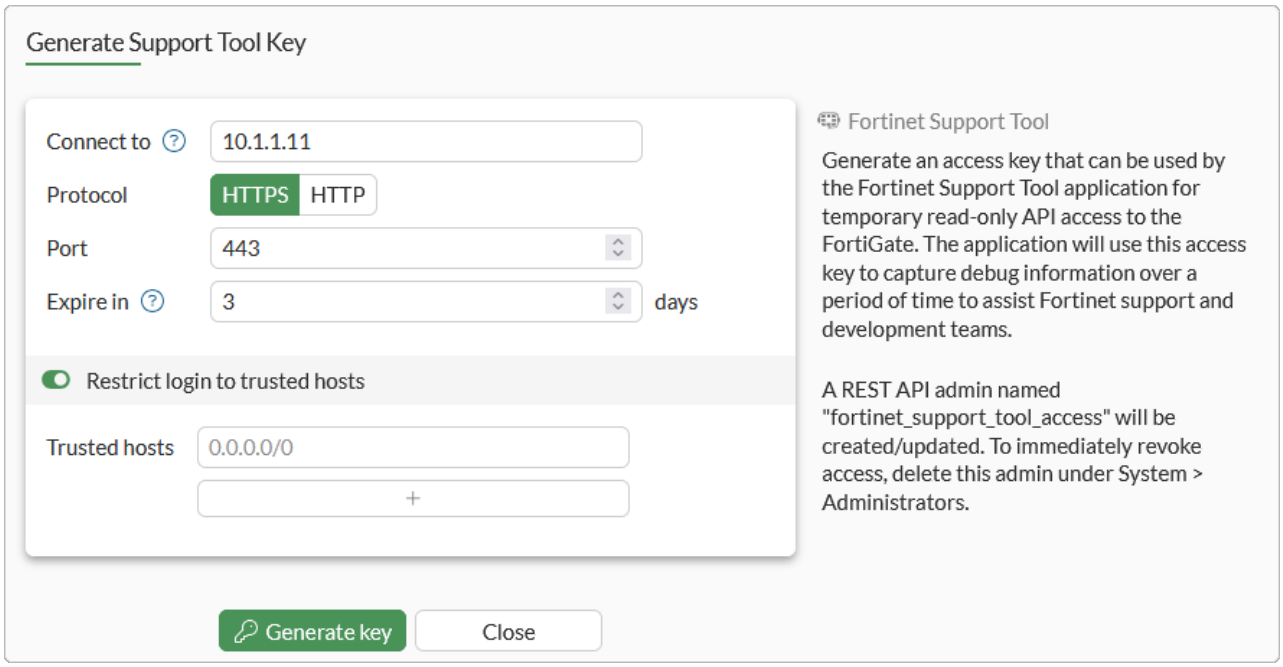
1. Open the Microsoft Store or the App Store.
2. Search for *Fortinet Support Tool*.
3. Install the tool.

To generate a support tool key:

1. In the FortiOS GUI, select *Generate Support Tool Key*.

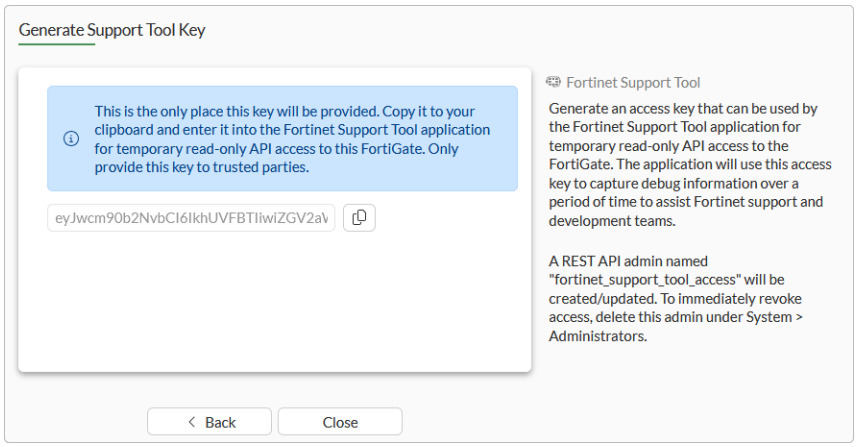


2. Configure the *Connect to* (IP/FQDN), *Protocol*, and *Port*, and set when the key will expire..



The Fortinet Support Tool will attempt to connect to this device using the specified IP/FQDN, protocol, and port.

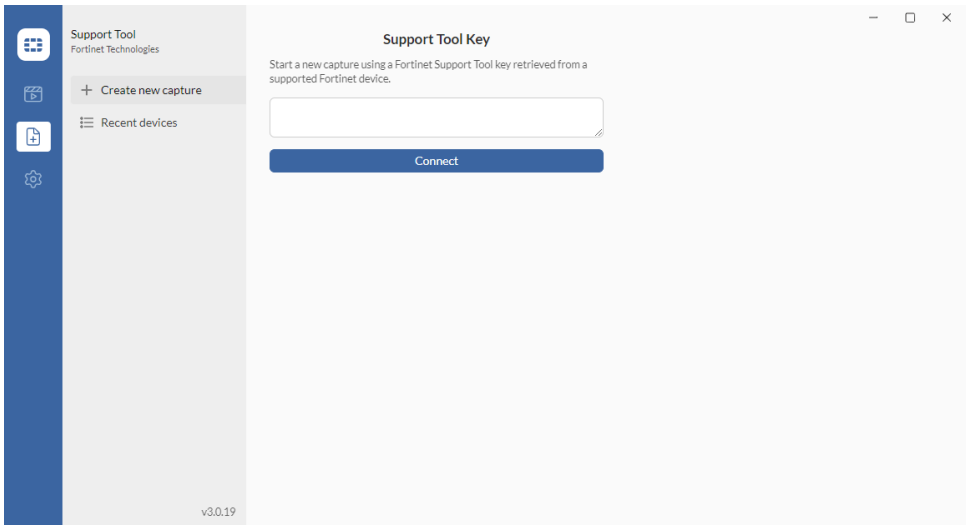
3. Optionally, configure trusted hosts.
4. Click *Generate key*.



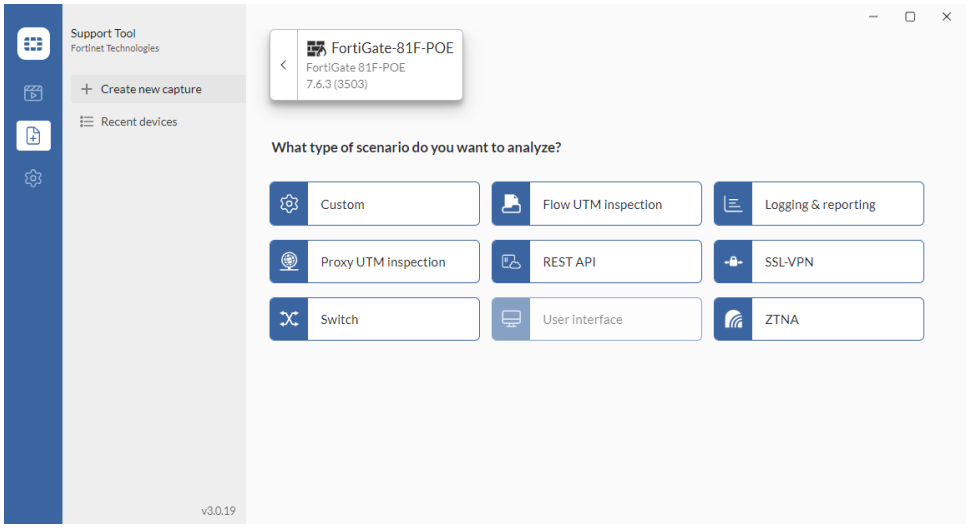
5. Copy the generated key, then click *Close*.

To create a new capture:

1. Open the Fortinet Support Tool and go to the capture tab.

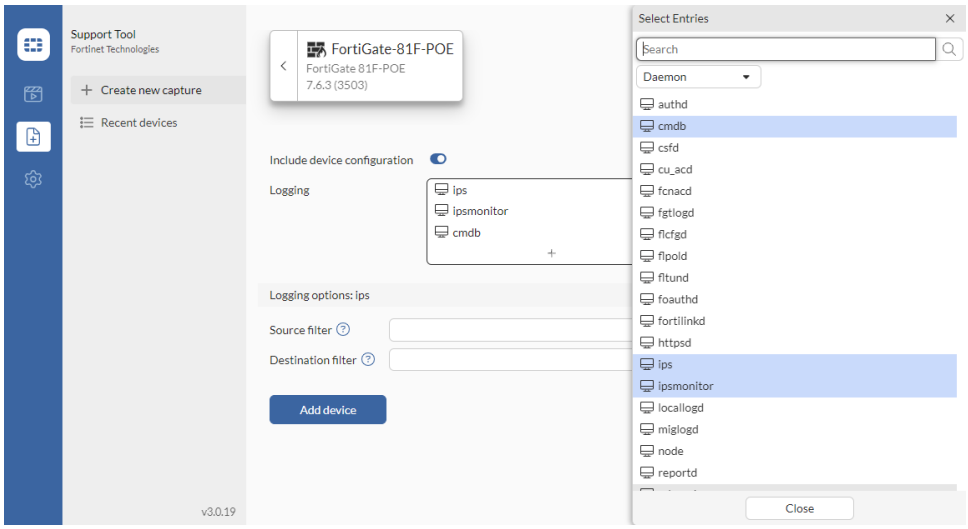


2. Paste the generated key into the field. The connection will start to be established immediately. You can also select *Recent devices* to try to reconnected to a previously used device.
3. Select the scenario to analyze.

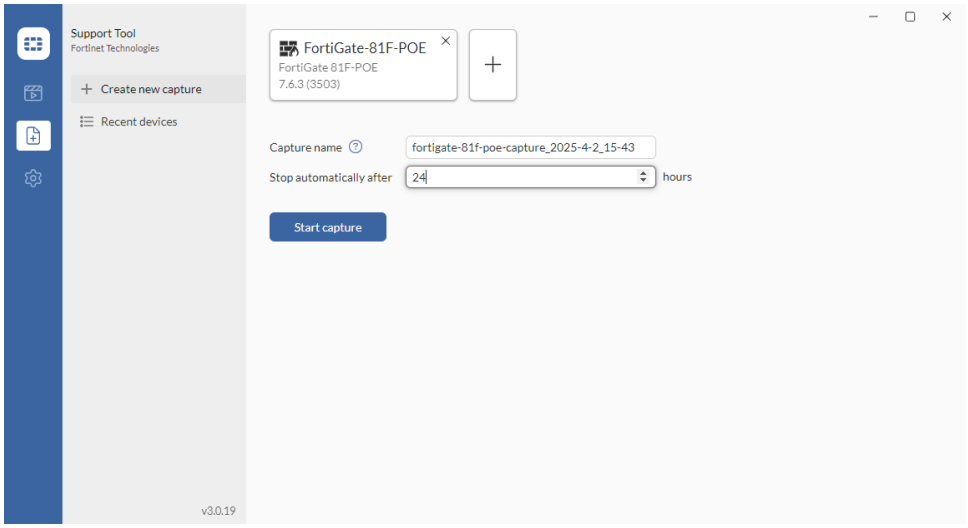


The user interface can only be captured using the Fortinet Support Tool Chrome extension.

4. Select the daemons to collect logs and apply filters as needed.



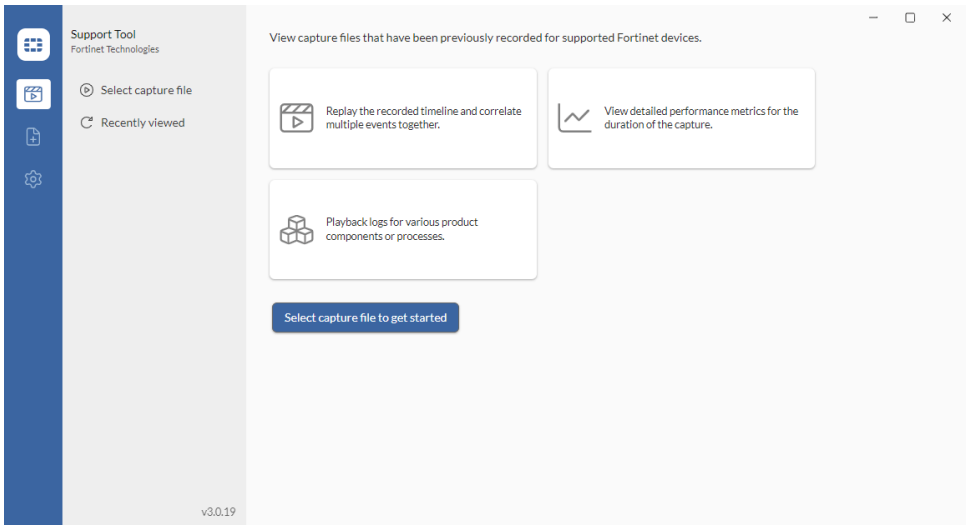
5. Specify the file name and length of the capture.



6. Click *Start capture*.
7. After the capture completes, or is manually stopped, save the capture file to the computer.

To view a capture:

1. Click *Select capture file to get started*.

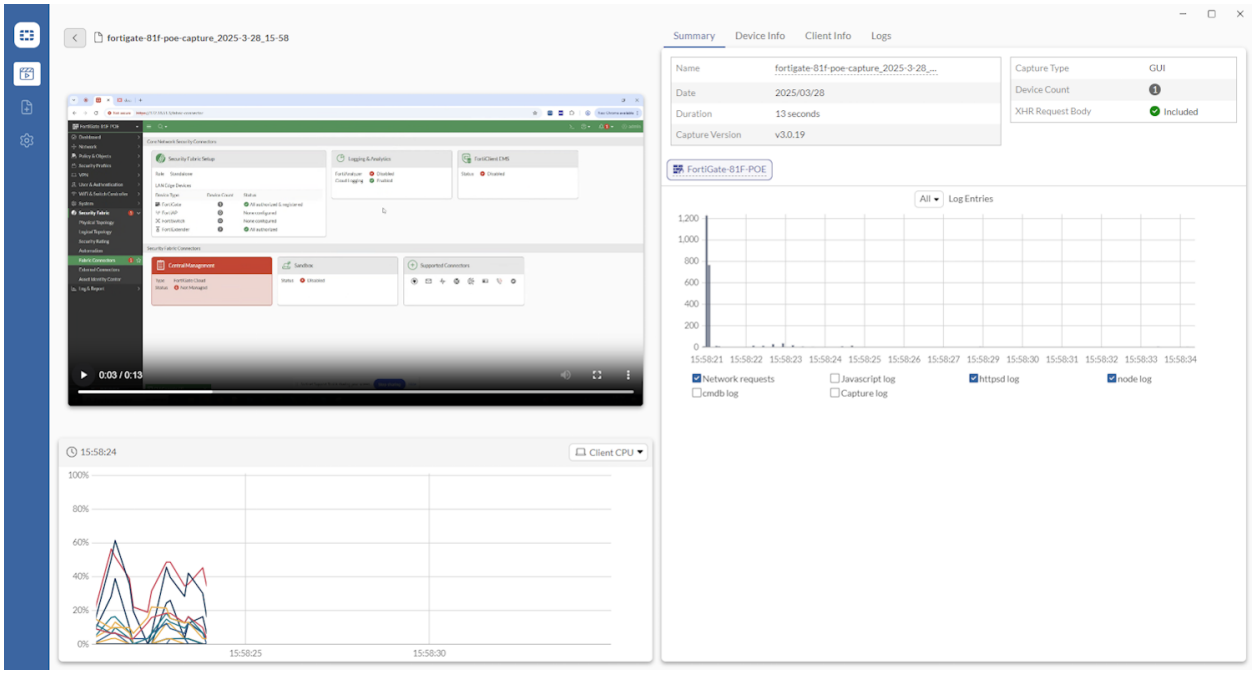


   You can also select *Recently viewed* to review a previously opened capture.

2. Select a capture file with the extensions .fgtcapture, .ftntguicap, or .ftntcap for review.

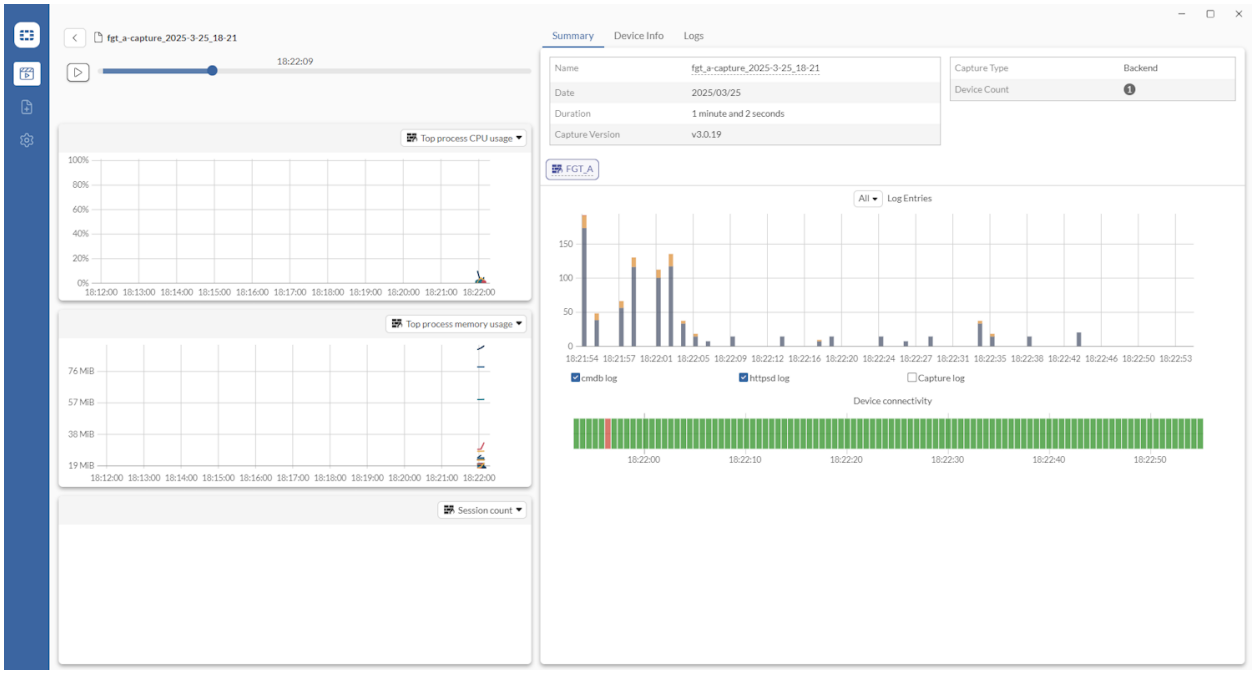
   When playing .fgtcapture or .ftntguicap files:

   - Top-left section: Video frame
   - Bottom-left section: Resource chart
   - Right section: Tabs including:
     - Summary
     - Device Info (Config, Crash log, Licenses, Table size, Profiling)
     - Client Info
     - Logs



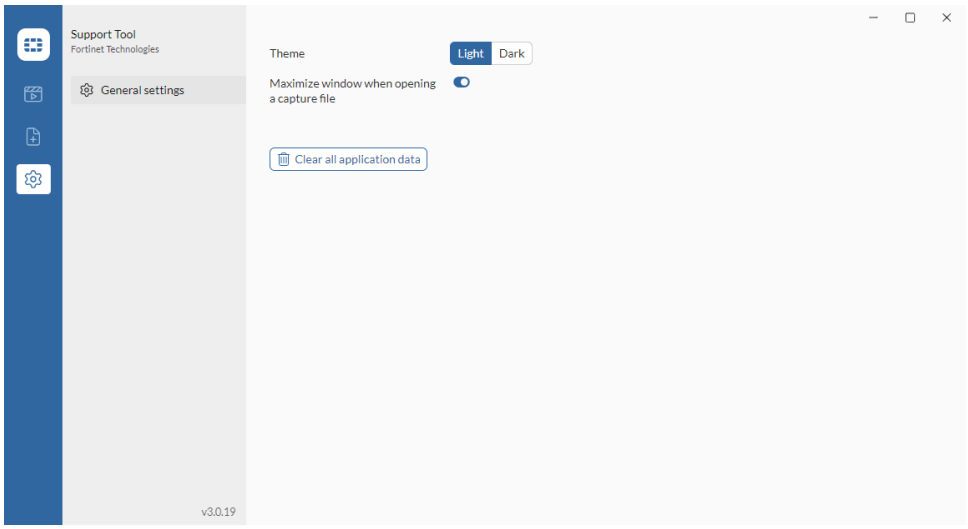
When playing .ftntcap files:

- Left section: Three resource charts
- Right section: Tabs including:
  - Summary
  - Device Info (General [Config, Crash log, Table size, Profiling], CLI diagnostics [System, Hardware, Filesystem, HA, Session, Update], REST API responses)
  - Logs



To configure the Fortinet Support Tool general settings:

1. Go to the *Settings* tab.



2. Adjust the *Theme* to *Dark* or *Light* (default).
3. Configure whether or not the application window is automatically maximized when viewing a capture file.

4. Click *Clear all application data* to clear all of the application data, such as recent devices or captures.

High Availability

Whether your FortiGate is used as a security gateway, an internal segmentation firewall, in the cloud, or in an MSSP environment, as long as there is critical traffic passing through it, there is risk of it being a single point of failure. Physical outages can occur due to power failures, physical link failures, transceiver failures, or power supply failures. Non-physical outages can be caused by routing, resource issues, or kernel panic.

Network outages cause disruptions to business operations, downtime, and frustration for users and in some situations may have financial setbacks. In designing your network and architecture, it is important to weigh the risks and consequences associated with unexpected outages.

There are many ways to build redundancy and resiliency. In a switching network, you can accomplish this by adding redundant links and switches in partial or full mesh topologies. Using redundant and aggregate links, you can avoid a single link failure causing a network to go down. Using SD-WAN, you can build redundant and intelligent WAN load balancing and failover architectures.

FortiGate HA offers several solutions for adding redundancy in the case where a failure occurs on the FortiGate, or is detected by the FortiGate through monitored links, routes, and other health checks. These solutions support fast failover to avoid lengthy network outages and disruptions to your traffic.

FortiGate Clustering Protocol (FGCP)

FGCP provides a solution for two key requirements of critical enterprise networking components: enhanced reliability and increased performance. Enhanced reliability is achieved through device failover protection, link failover protection, and remote link failover protection. Session failover protection for most IPv4 and IPv6 sessions also contributes to enhanced reliability. Increased performance is achieved though active-active HA load balancing.

See FGCP on page 4071.

FortiGate Session Life Support Protocol (FGSP)

In a network that already includes load balancing (either with load balancers or routers) for traffic redundancy, two entities (either standalone FortiGates or FGCP clusters) can be integrated into the load balancing configuration using the FortiGate Session Life Support Protocol (FGSP). The external load balancers or routers can distribute sessions among the FortiGates and the FGSP performs session synchronization of IPv4 and IPv6 TCP, SCTP, UDP, ICMP, expectation, and NAT sessions to keep the session tables of both entities synchronized. In the event of a failure, the load balancer can detect the failed unit and failover the sessions to other active members to continue processing the traffic.

See FGSP on page 4211.

VRRP

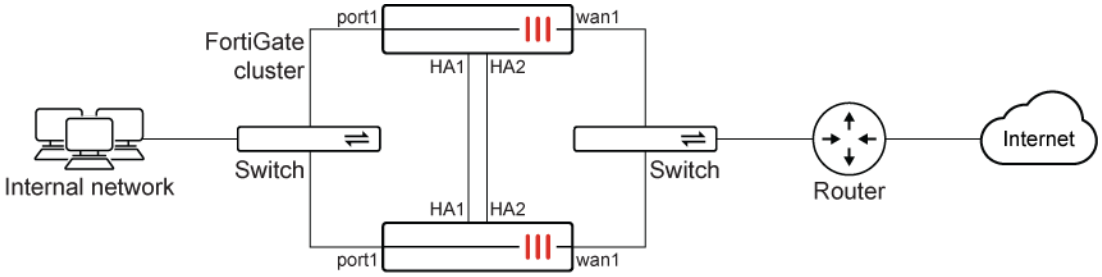
FortiGates can function as primary or backup Virtual Router Redundancy Protocol (VRRP) routers. The FortiGates can quickly and easily integrate into a network that has already deployed VRRP. A FortiGate can be integrated into a VRRP group with any third-party VRRP devices, and VRRP can provide redundancy between multiple FortiGates. FortiOS supports VRRP version 2 and 3.

See VRRP on page 4281.

FGCP

High availability (HA) is usually required in a system where there is high demand for little downtime. There are usually hot-swaps, backup routes, or standby backup units and as soon as the active entity fails, backup entities will start functioning. This results in minimal interruption for the users.

The FortiGate Clustering Protocol (FGCP) is a proprietary HA solution whereby FortiGates can find other member FortiGates to negotiate and create a cluster. A FortiGate HA cluster consists of at least two FortiGates (members) configured for HA operation. All FortiGates in the cluster must be the same model and have the same firmware installed. Cluster members must also have the same hardware configuration (such as the same number of hard disks). All cluster members share the same configurations except for their host name and priority in the HA settings. The cluster works like a device but always has a hot backup device.



All FortiGates that are in the same HA cluster must be registered under the same FortiCare account. Registering cluster members to different FortiCare accounts will result in licensing issues and potential downtime.

Critical cluster components

The following are critical components in an HA cluster:

- Identical heartbeat connections and interfaces: members will use this to communicate with each other. In general, a two-member cluster is most common. We recommend double back-to-back heartbeat connections (as demonstrated in the topology).

- Identical connections for internal and external interfaces: we recommend similar connections from each member to the switches for the cluster to function properly (as demonstrated in the topology).

The HA heartbeat interface communicates with each unit in the cluster using the same heartbeat interface for each member.

For example, if port1 and port2 are the heartbeat interfaces for the HA cluster, then in a cluster consisting of two members:

- port1 of the primary FortiGate should be connected to port1 of the secondary FortiGate.
- port2 of the primary FortiGate should be connected to port2 of the secondary FortiGate.

General operation

The following are best practices for general cluster operation:

- Ensure that heartbeat communication is present (see HA heartbeat interface on page 4098).
- Enable the session synchronization option in daily operation (see FGSP basic peer setup on page 4214).
- Monitor traffic flowing in and out of the interfaces.

Failover

FGCP provides failover protection in the following scenarios:

- The active device loses power.
- A monitored interface loses a connection.

After failover occurs, the user will not notice any difference, except that the active device has changed. See Failover protection on page 4110 for more information.

Synchronizing the configuration

FGCP uses a combination of incremental and periodic synchronization to make sure that the configuration of all cluster units is synchronized to that of the primary unit.

The following settings are not synchronized between cluster units:

- The FortiGate host name
- GUI Dashboard widgets
- HA override
- HA device priority
- The virtual cluster priority
- The system interface settings of the HA reserved management interface
- The HA default route for the reserved management interface, set using the `ha-mgmt-interface-gateway` option of the `config system ha` command

Most subscriptions and licenses are not synchronized, as each FortiGate must be licensed individually. FortiToken Mobile is an exception; they are registered to the primary unit and synchronized to the secondary units.

The primary unit synchronizes all other configuration settings, including the other HA configuration settings.

All synchronization activity takes place over the HA heartbeat link using TCP/703 and UDP/703 packets.

Topics

The following topics provide more information about FGCP:

Topic	Summary
Cluster operation	Defines how multiple units function as a single system using the FGCP protocol to negotiate roles and manage traffic. It includes the election logic for selecting a primary unit and configuring Active-Passive or Active-Active modes.
HA heartbeat	Explains the communication mechanism used to synchronize configurations and monitor peer health via "hello" packets. It details interface configuration, timing intervals, and how lost packets trigger a failover.
Session synchronization and failover	Describes the automatic transfer of traffic to a standby unit if the primary device fails. It focuses on session synchronization and pickup to preserve active user connections with minimal interruption.
Management access	Details how to establish reliable administrative connections to all cluster members using In-band or Out-of-band methods. It explains reserving dedicated interfaces to isolate management traffic from production data.
Maintenance	Covers procedures for upgrading firmware securely while minimizing downtime. It outlines standard upgrade paths as well as multi-version clusters for testing new firmware in a mixed environment.
Advanced operations	Explores complex scenarios like geographically distributed HA and clustering over managed switches. It addresses specialized needs such as Virtual MAC assignment and handling asymmetric routing or mixed power supplies.
Troubleshooting	Provides diagnostic steps and CLI commands to identify formation errors and synchronization discrepancies. It includes procedures for manually forcing failovers to test redundancy and verifying cluster health.
Best practices	Lists architectural standards to eliminate single points of failure, such as using redundant, isolated heartbeat links. It recommends performance optimizations like filtering session synchronization and ensuring hardware consistency.

Cluster operation

Cluster operation defines how multiple FortiGate units function together as a cohesive High Availability (HA) system using the FortiGate Clustering Protocol (FGCP). This operational framework governs how cluster members communicate, negotiate roles (Primary versus Secondary), and manage network traffic.

Key aspects of cluster operation include:

- **Operating Modes**: Defining whether standby units are passive or actively processing traffic.
- **Traffic Management**: utilizing load balancing schedules to distribute specific security tasks.
- **Virtualization**: Segmentation of the cluster into virtual clusters for granular VDOM control.
- **Election Logic**: The deterministic criteria used to select the primary unit during startup or failover events.

The following sections provide detailed configuration guidelines and behavioral insights for these components:

Topic	Summary
HA primary unit selection criteria	The strict logic and hierarchy the FortiGate uses to deterministically elect a Primary unit during startup or recovery.
HA active-passive cluster setup	The standard redundancy model where one unit handles all traffic while the other remains in standby, ready to take over instantly if the primary fails
HA active-active cluster setup	A performance-oriented configuration where both units actively process traffic, allowing you to leverage the resources of the secondary unit for specific security tasks.
Single FortiGuard license for FortiGate A-P HA cluster	Specific configuration details for sharing a single service license across cluster units using a Virtual Serial Number (vSN) on supported models.
HA virtual cluster setup	An advanced strategy using Virtual Domains (VDOMs) that allows different units to act as the Primary for different logical network segments, maximizing hardware utilization without the complexity of full Active-Active mode.

HA primary unit selection criteria

In a FGCP HA setup, cluster members must negotiate to determine who will become the primary unit upon connecting to the HA cluster. Once a primary unit is identified, all other members become subordinate (or secondary) members.

When does primary unit selection occur?

Primary unit selection occurs whenever a new unit joins the HA cluster or the primary unit leaves the HA cluster. It also occurs whenever a monitored interface status changes.

This can occur when:

- Two or more units initially form a new HA cluster.
- A new unit joins an existing HA cluster.
- A device failover takes place, where the primary unit fails due to a device failure.

- A link failover takes place, where a monitored interface on any unit either fails or is restored.

Relevant configurations

Configurations that can impact the HA primary unit section are listed below:

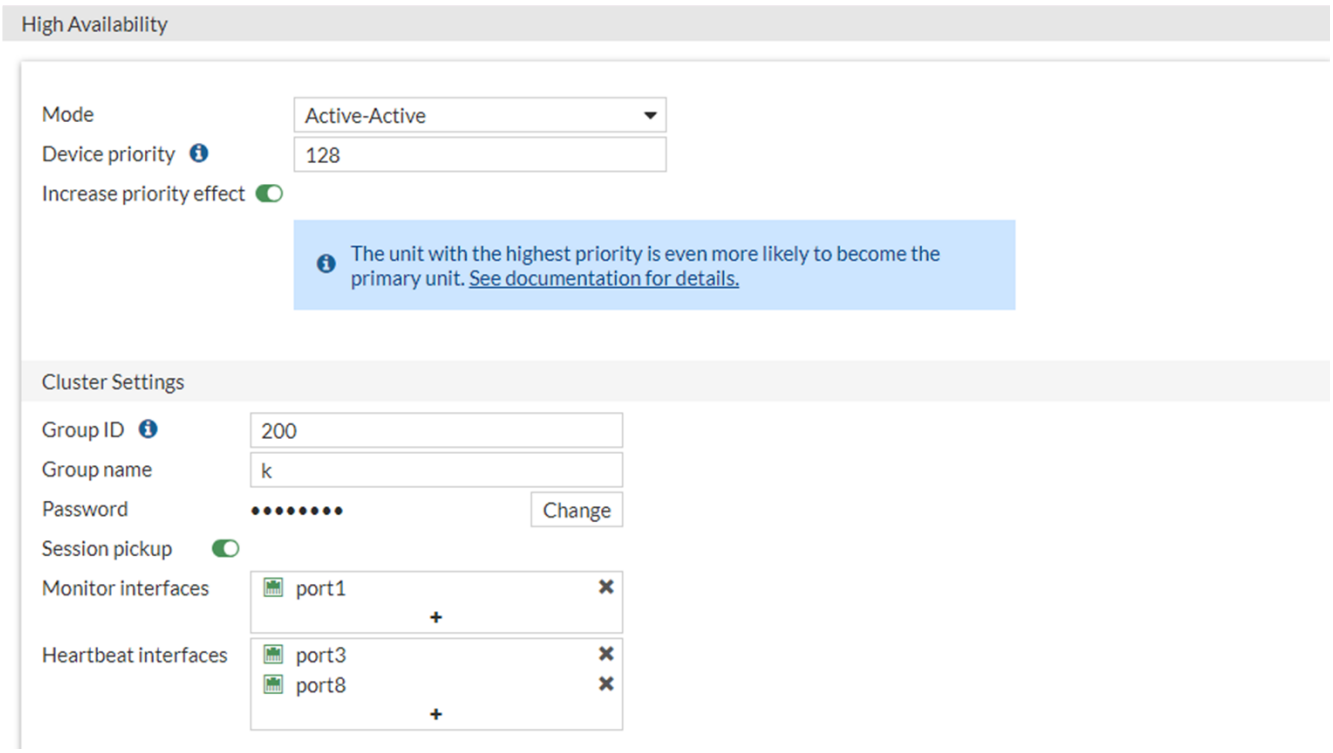
Priority	A value between 0-255 assigned to this unit. A higher number indicates higher priority. By default, priority is 128. Priority value does not get synchronized to other HA members.
Monitor	Interface(s) to check for a physical link failure. The following interface types are supported: • Physical interfaces • Aggregate interfaces • Redundant interfaces • Software-switch members
Override	Enable to prioritize priority value over uptime in HA primary unit selection. Disable to prioritize uptime over priority value. This setting is disabled by default. To ensure optimal performance and consistent behavior in your FortiGate HA cluster, it is strongly recommended that you standardize the override configuration across all cluster members. Specifically, override should be either uniformly enabled or disabled on all cluster units. Mixed configurations can lead to conflicts and unpredictable primary unit elections, which may impact cluster stability and manageability.

From CLI:

```
config system ha
    set priority <integer>
    set monitor <interface list>
    set override {enable | disable}
end
```

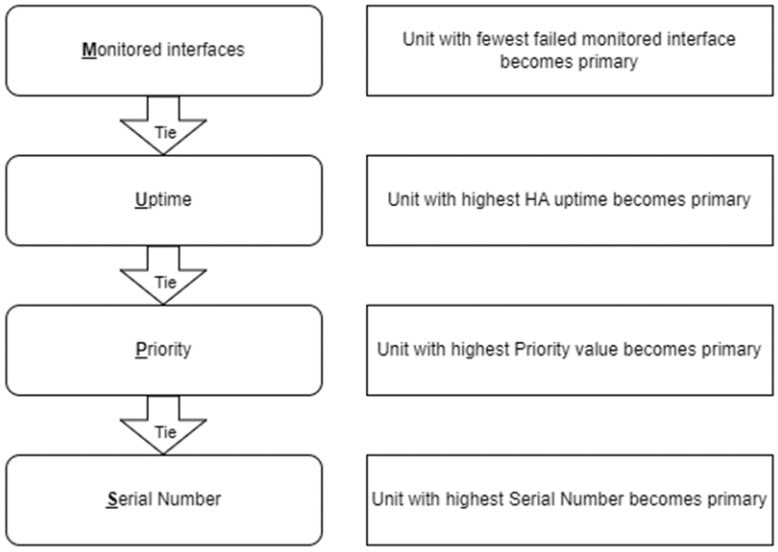
From GUI:

On the *System* > *HA* page:

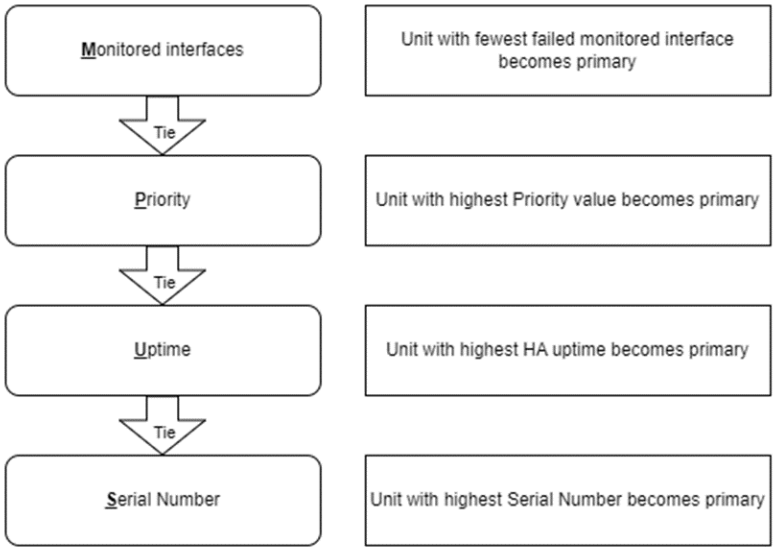


Primary unit selection criteria

If the HA override setting is *disabled* on all cluster members, the primary unit will be selected based on the following order:



If the HA override setting is *enabled* on all cluster members, the primary unit will be selected based on the following order:



For each criteria, if the value is the same, then it is considered a tie, and the next criteria is evaluated.

For the HA uptime criteria:

- If the difference between HA uptime is more than five (5) minutes (300 seconds), the cluster unit that is operating longer becomes the primary unit.
- If the difference between HA uptime is less than five (5) minutes (300 seconds), then the criteria is considered a tie.
- If a monitored interface fails on a HA unit, its HA uptime is reset to zero (0).
- If a cluster member restarts, the HA uptime is reset to zero (0).

In some documents, the terms MUPS and MPUS, which are based on the first letters of each criteria, are used to describe the order in which the criteria are considered during the HA primary unit selection process.

Viewing the role of the unit

After HA primary unit selection has completed, you can view the HA role of each unit in various ways.

- In the GUI, go to *System* > *HA* to view the members in the cluster and the role for each member.
- From the CLI, run `get system ha status`. The role of each unit is displayed:

```
# get system ha status
…
Primary: FG101FTK19xxxxx7, HA operating index = 0
Secondary: FG101FTK19xxxxx8, HA operating index = 1
```

- Similarly, from the CLI, run `diagnose sys ha status`. The role of each unit is displayed.

Viewing how the primary unit was selected

You can use the `get system ha status` command to see how the primary unit was selected. The output of this command contains a section called `Primary selected using` that shows a history of how the primary unit was selected.

```
# get system ha status
HA Health Status:
    WARNING: FG101FTK19xxxxx7 has hbdev down;
    WARNING: FG101FTK19xxxxx8 has hbdev down;
Model: FortiGate-101F
Mode: HA A-A
Group Name: FGT_HA
Group ID: 0
Debug: 0
Cluster Uptime: 5 days 8h:30m:57s
Cluster state change time: 2024-04-12 02:25:05
Primary selected using:
    <2024/04/12 02:25:05> vcluster-1: FG101FTK19xxxxx7 is selected as the primary because its
override priority is larger than peer member FG101FTK19xxxxx8.
    <2024/04/12 02:25:04> vcluster-1: FG101FTK19xxxxx7 is selected as the primary because it's the
only member in the cluster.
    <2024/04/12 02:13:34> vcluster-1: FG101FTK19xxxxx7 is selected as the primary because its
override priority is larger than peer member FG101FTK19xxxxx8.
    <2024/04/12 02:09:28> vcluster-1: FG101FTK19xxxxx7 is selected as the primary because it's the
only member in the cluster.
```

Comparing the HA uptime between cluster members

You can use the CLI command `diagnose sys ha dump-by group` to display the age difference of the units in a cluster. This command also displays information about a number of HA related parameters for each cluster unit.

For example, consider a cluster of two FortiGate units. Entering the `diagnose sys ha dump-by group` command from the primary unit CLI displays information similar to the following:

```
# diagnose sys ha dump-by group
...
vcluster_nr=1
vcluster-1: start_time=1712913904(2024-04-12 02:25:04), state/o/chg_time=2(work)/2
(work)/1712913904(2024-04-12 02:25:04)
        pingsvr_flip_timeout/expire=3600s/0s
        'FG101FTK19xxxxx8': ha_prio/o=1/1, link_failure=0, pingsvr_failure=0, flag=0x00000000,
mem_failover=0, uptime/reset_cnt=0/2
        'FG101FTK19xxxxx7': ha_prio/o=0/0, link_failure=0, pingsvr_failure=0, flag=0x00000001,
mem_failover=0, uptime/reset_cnt=189/2
```

The last two lines of the output display status information about each cluster unit including the `uptime`. The `uptime` is the age difference in seconds between the two units in the cluster.

In the example, the age of the subordinate unit is 189 seconds more than the age of the primary unit. The age difference is less than five (5) minutes (less than 300 seconds), so age has no effect on primary unit selection.

Changing the cluster age difference margin

You can change the cluster age difference margin using the following command:

```
config system ha
    set ha-uptime-diff-margin <margin>
end
```

Where the `<margin>` can be from 1 to 65535 seconds (default = 300).

Resetting the uptime of a unit

For debugging purpose, you may want to reset the HA member's uptime without restarting the unit or changing the status of a monitored interface.

To manually change the uptime:

```
# diagnose sys ha reset-uptime
```

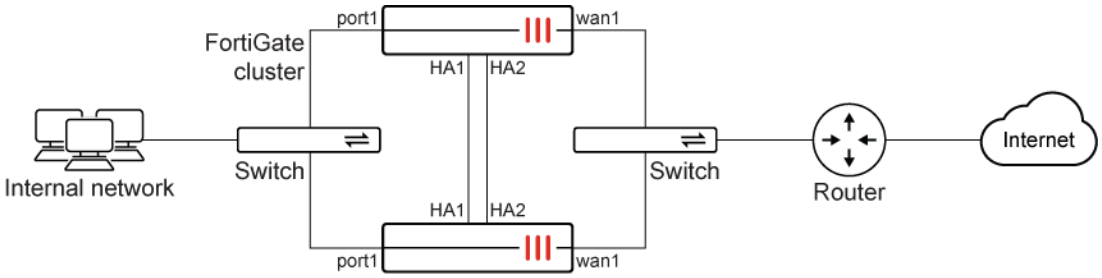
The command resets the HA age internally and does not affect the up time displayed for cluster units using the `diagnose sys ha dump-by all-vcluster` or `diagnose sys ha dump-by all-vcluster` command. It also does not affect the time displayed on the Dashboard or cluster members list.

HA active-passive cluster setup

An HA Active-Passive (A-P) cluster can be set up using the GUI or CLI.

FortiGate A-P HA cluster supports sharing a single FortiGuard service license for both cluster units. See Single FortiGuard license for FortiGate A-P HA cluster on page 4083.

This example uses the following network topology:

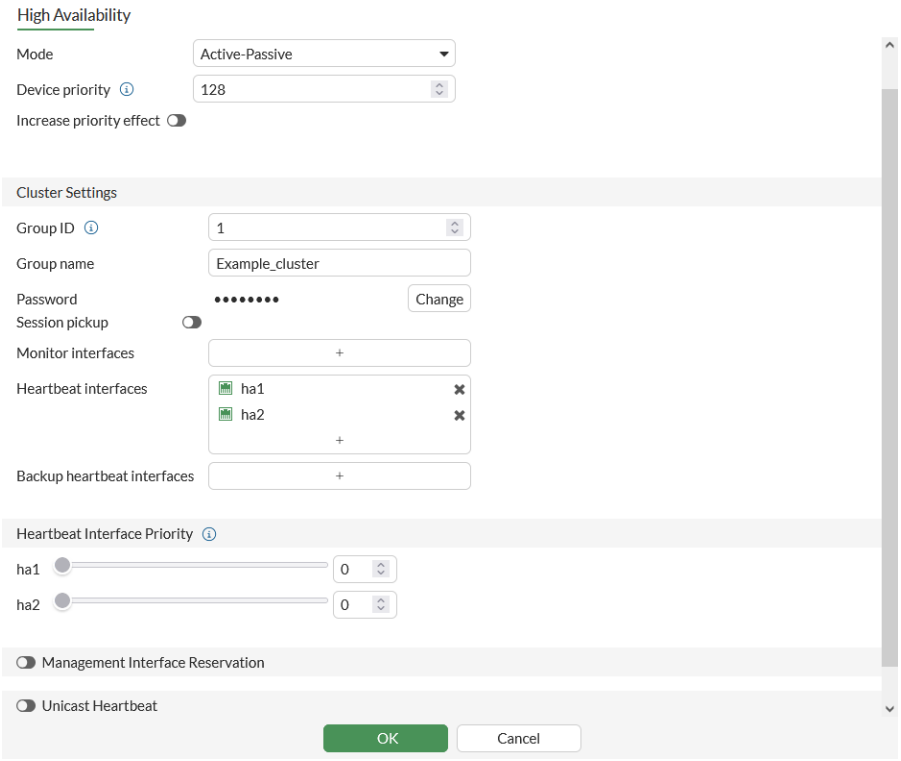


To set up an HA A-P cluster using the GUI:

1. Make all the necessary connections as shown in the topology diagram.
2. Log into one of the FortiGates.
3. Go to *System* > *HA* and set the following options:

Option	Value
Mode	Active-Passive
Device priority	128 or higher
Group ID	1 The group ID must be the same in all HA members in order to form a cluster. The group ID can impact the definition of the virtual MAC addresses of interfaces. See Determining VMAC addresses for more details.
Group name	Example_cluster
Password	********
Heartbeat interfaces	ha1 and ha2

Except for the device priority, these settings must be the same on all FortiGates in the cluster.



4. Leave the remaining settings as their default values. They can be changed after the cluster is in operation.
5. Click *OK*.
   The FortiGate negotiates to establish an HA cluster. Connectivity with the FortiGate may be temporarily lost as the HA cluster negotiates and the FGCP changes the MAC addresses of the FortiGate's interfaces.
6. Factory reset the other FortiGate that will be in the cluster, configure GUI access, then repeat steps 1 to 5, omitting setting the device priority, to join the cluster.

To set up an HA A-P cluster using the CLI:

1. Make all the necessary connections as shown in the topology diagram.
2. Log into one of the FortiGates.
3. Change the hostname of the FortiGate:

```
config system global
    set hostname Example1_host
end
```

   Changing the host name makes it easier to identify individual cluster units in the cluster operations.
4. Enable HA:

```
config system ha
    set mode a-p
    set group-id 1
    set group-name Example_cluster
    set password ********
    set hbdev ha1 10 ha2 20
end
```

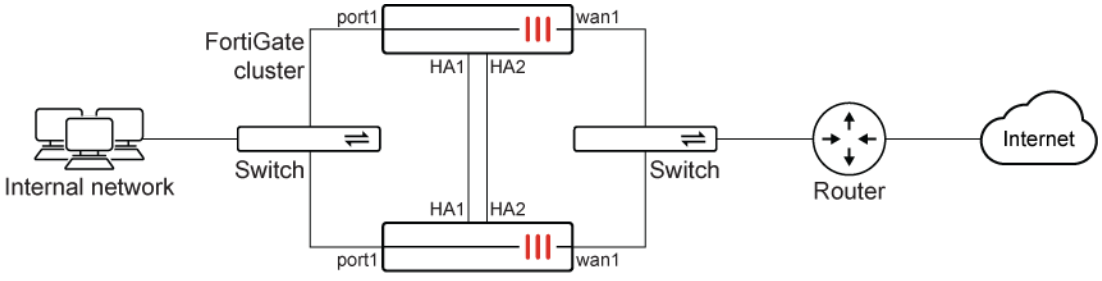
5. Leave the remaining settings as their default values. They can be changed after the cluster is in operation.
6. Repeat steps 1 to 5 on the other FortiGate devices to join the cluster, giving each device a unique hostname.

HA active-active cluster setup

An HA Active-Active (A-A) cluster can be set up using the GUI or CLI.

> An A-A cluster supports interfaces in DHCP mode, but not interfaces in PPPoE mode. If an interface is in PPPoE mode, then the Active-Active option will not appear in the *Mode* selection.
>
> FGCP in A-A mode cannot load balance any sessions that traverse NPU VDOM links or regular VDOM links. If Active-Active session load balancing between VDOMs is required, use an external router to handle the inter-VDOM routing.

This example uses the following network topology:



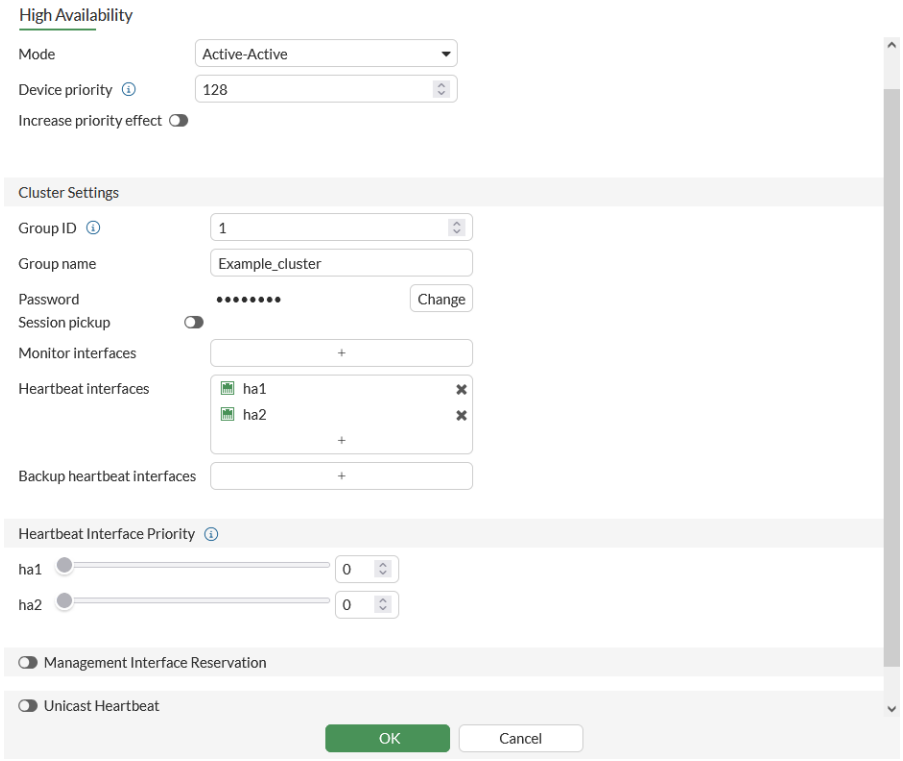
To set up an HA A-A cluster using the GUI:

1. Make all the necessary connections as shown in the topology diagram.
2. Log into one of the FortiGates.

3. Go to *System* > *HA* and set the following options:

Mode	Active-Active
Device priority	128 or higher
Group ID	1 The group ID must be the same in all HA members in order to form a cluster. The group ID can impact the definition of the virtual MAC addresses of interfaces. See Determining VMAC addresses for more details.
Group name	Example_cluster
Password	********
Heartbeat interfaces	ha1 and ha2

Except for the device priority, these settings must be the same on all FortiGates in the cluster.



4. Leave the remaining settings as their default values. They can be changed after the cluster is in operation.
5. Click *OK*.
   The FortiGate negotiates to establish an HA cluster. Connectivity with the FortiGate may be temporarily lost as the HA cluster negotiates and the FGCP changes the MAC addresses of the FortiGate's interfaces.
6. Factory reset the other FortiGate that will be in the cluster, configure GUI access, then repeat steps 1 to 5, omitting setting the device priority, to join the cluster.

To set up an HA A-A cluster using the CLI:

1. Make all the necessary connections as shown in the topology diagram.
2. Log into one of the FortiGates.
3. Change the hostname of the FortiGate:

```
config system global
    set hostname Example1_host
end
```

   Changing the host name makes it easier to identify individual cluster units in the cluster operations.
4. Enable HA:

```
config system ha
    set mode a-a
    set group-id 1
    set group-name Example_cluster
    set password ********
    set hbdev ha1 10 ha2 20
end
```

5. Leave the remaining settings as their default values. They can be changed after the cluster is in operation.
6. Repeat steps 1 to 5 on the other FortiGate devices to join the cluster.

Single FortiGuard license for FortiGate A-P HA cluster

FortiGate A-P HA cluster supports sharing a single FortiGuard service license for both cluster units for the following models:

- 40F and variants
- 60F and variants
- 70F and variants
- 80F and variants
- 100F and variants

When a customer purchases two units with the HA SKU (such as 2 x FG-40F-HA), they can further purchase a single order of the following subscriptions:

- Enterprise Protection
- Unified Threat Protection (UTP)
- Advanced Threat Protection (ATP)

The two FortiGate serial numbers will be associated together on FortiCare to create one virtual serial number (vSN). If multiple pairs of devices are ordered, each pair will be together in its own box to help identify the associated devices. The aforementioned services will then be registered to the vSN. A la carte SKUs are not supported, and cannot be registered to the vSN.

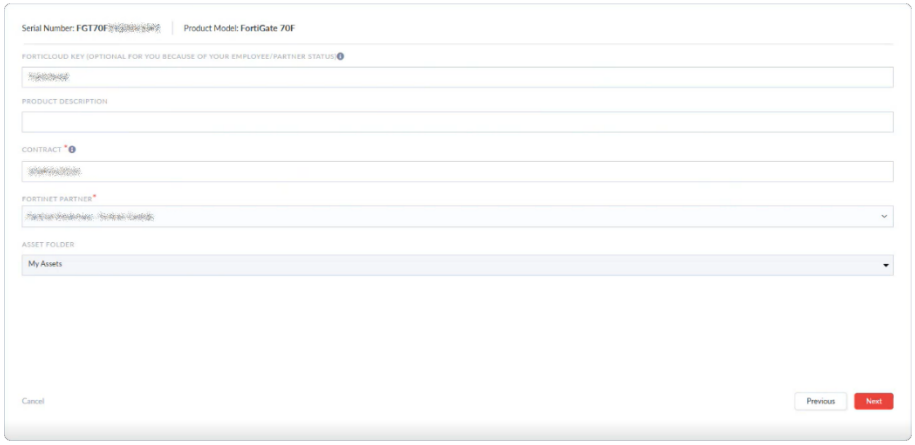
Deploying the FortiGates in HA to support vSN requires two steps:

1. Register the FortiGate and associated service contract
2. Provision the FortiGate HA configurations either through FortiGate Cloud or manually

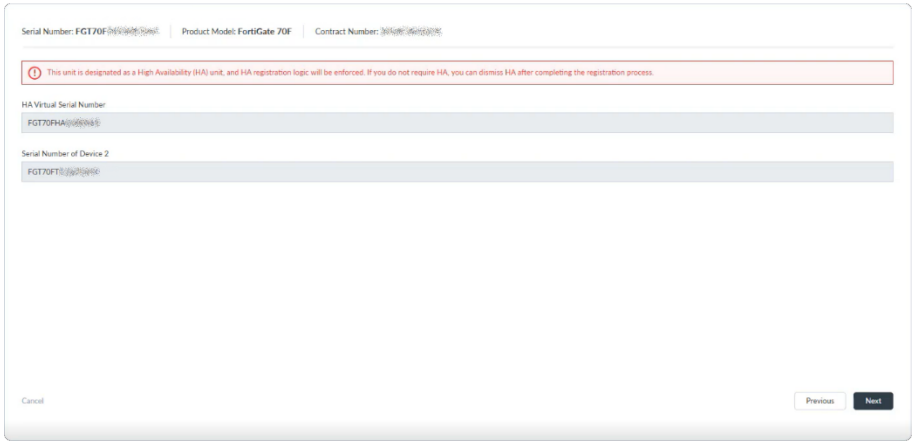
For information about RMAing the HA cluster, see Replacing a failed cluster unit on page 4208.

To register the FortiGates and associated contract:

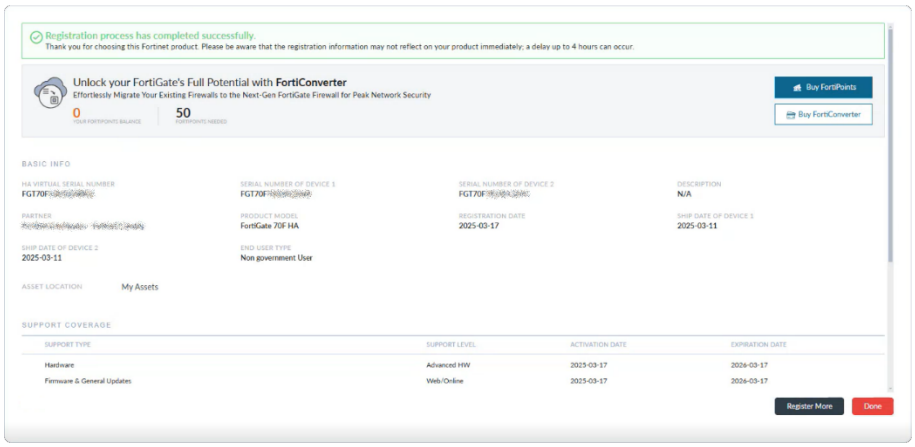
1. Log in to the FortiCloud support portal.
2. In the *Dashboard*, click *Register Now* to register a device and contract.
3. In the *Registration* field, enter the one of the FortiGate's serial numbers. Do not enter the service contract registration code, license certificate number, or asset transfer token.
4. Set the end user type, then click *Next*.
5. Enter the FortiCloud key from the FortiGate in the *FortiCloud Key* field, and the Registration Code from the service entitlement document in the *Contract* field.



6. Configure the remaining settings as needed then click *Next*.



7. Check the HA vSN and the serial number of the second device, then click *Next*.
8. Review the configuration, then click *Done* to complete the registration.



The vSN is registered in *Asset Management* with service entitlement. The individual FortiGates cannot be managed.

To configure the FortiGates in HA using FortiGate Cloud:

1. In the FortiGate Cloud portal, provision the FortiGate:
   a. In the FortiCloud support portal, go to *Services* > *FortiGate Cloud* to open the portal.
   b. Go to *Assets* > *Asset list* and click *Add FortiGate*.
   c. Select the FortiGate vSN from the inventory table and click *Provision* > *Provision to FortiGate Cloud*.
2. Unpack the boxes and connect the HA interfaces back to back using the highest physical port number that is not a fabric port (portA and portB) as indicated:

Model	HA interface
FortiGate 40F series	port3
FortiGate 60F series	port5
FortiGate 70F series	port4 and/or 5
FortiGate 80F series	port5 and/or 6
FortiGate 100F series	ha1 and/or ha2

   Some models have 2 HA interfaces. In these cases, both interfaces will be provisioned by FortiGate Cloud as heartbeat interfaces, but one or both of the interfaces can be connected. It is recommended to connect 2 heartbeat interfaces whenever possible for redundancy.
3. Connect the WAN interface to an upstream gateway that is providing DHCP service.
4. Connect internal interfaces to an internal switch as required.
5. Power on both FortiGates.
   Shortly after, the boxes will receive the vSN and their HA configuration from FortiGate Cloud, as follows:

```
config system ha
    set group-id <id>
    set group-name <group-name>
    set mode a-p
    set password ********
    set hbdev <HA interface 1> <priority 1> [HA interface 2] [priority 2]
```

```
    set override disable
    set logical-sn enable
end
```

To configure the FortiGates in HA manually using the CLI:

1. Unpack the two boxes, and connect to each unit through the CLI or the default management interface.
2. Configure the following basic HA settings on each unit:

```
config system ha
    set mode a-p
    set group-id <id>
    set group-name <group-name>
    set password ********
    set hbdev <HA interface 1> <priority 1> [HA interface 2] [priority 2]
    set logical-sn enable
end
```

3. Connect the HA interfaces back to back using your preferred interfaces.
4. Power on both FortiGates.
   Shortly after, the boxes will receive the vSN.

To verify the HA status and vSN (or Logical Serial) after the HA cluster registration is complete:

1. In the GUI, go to the *System* > *HA* page.
2. In the CLI use these commands:

```
# get system ha status
HA Health Status: OK
Model: FortiGate-80F
Mode: HA A-P
Group Name: Branch1-HA
Group ID: 100
Debug: 0
Cluster Uptime: 0 days 2h:33m:2s
Cluster state change time: 2024-11-19 13:57:31
Primary selected using:
    <2024/11/19 13:57:31> vcluster-1: FGT80FTK22023xxx is selected as the primary because its
override priority is larger than peer member FGT80FTK20000xxx.
    <2024/11/19 11:26:06> vcluster-1: FGT80FTK22023xxx is selected as the primary because it's
the only member in the cluster.
ses_pickup: enable, ses_pickup_delay=disable
override: enable
Configuration Status:
    FGT80FTK22023xxx(updated 1 seconds ago): in-sync
    FGT80FTK22023xxx chksum dump: 0e 4c b5 56 80 be bf 20 8e e5 ad d5 59 ea 5d b3
    FGT80FTK20000xxx(updated 0 seconds ago): out-of-sync
    FGT80FTK20000xxx chksum dump: d1 31 59 fc 0b 91 12 ca 92 69 62 d2 9f b7 a3 c3
System Usage stats:
    FGT80FTK22023xxx(updated 1 seconds ago):
```

```
        sessions=18, average-cpu-user/nice/system/idle=0%/0%/0%/100%, memory=26%
    FGT80FTK20000xxx(updated 0 seconds ago):
        sessions=4, average-cpu-user/nice/system/idle=6%/0%/6%/87%, memory=24%
HBDEV stats:
    FGT80FTK22023xxx(updated 1 seconds ago):
        internal3: physical/1000auto, up, rx-bytes/packets/dropped/errors=1492065/22100/0/0,
tx=20442845/47022/0/0
    FGT80FTK20000xxx(updated 0 seconds ago):
        internal3: physical/1000auto, up, rx-bytes/packets/dropped/errors=24954361/57802/0/0,
tx=1804396/27277/0/0
number of member: 2
80FASAAA        , FGT80FTK22023xxx, HA cluster index = 0
FGT-D           , FGT80FTK20000xxx, HA cluster index = 1
number of vcluster: 1
vcluster 1: work 169.254.0.1
Primary: FGT80FTK22023xxx, HA operating index = 0
Secondary: FGT80FTK20000xxx, HA operating index = 1
Logical Serial Number: FGT80FHA24090xxx
```

```
# diagnose system ha dump-by debug-zone
            HA information.
is_manage_primary=1,manage_vd=root,ip=169.254.0.1,num=2,nvcluster=1,jiffies=938038.
logical serial number is FGT80FHA24090xxx,
local serial number is FGT80FTK22023xxx,
member's serial number is FGT80FTK20000xxx
```

3. Furthermore, the service contract will be associated with the vSN and can be viewed on the *System* > *FortiGuard* page.

> ⚠ Do not change the HA mode from A-P to A-A when `logical-sn` is enabled. This will result in the FortiGate losing its vSN. Disabling `logical-sn` will also result in losing the vSN. As a result, service entitlements will no longer be registered to the HA cluster.

HA virtual cluster setup

Virtual clustering is an extension of FGCP HA to allow multiple clusters to be formed between your HA members. In effect, each cluster consists of the same HA members, with the option to prioritize different members as the primary unit. Each cluster operates as its on active-passive FGCP HA cluster, with different virtual domains residing in the virtual cluster. The following custom settings can be configured per cluster:

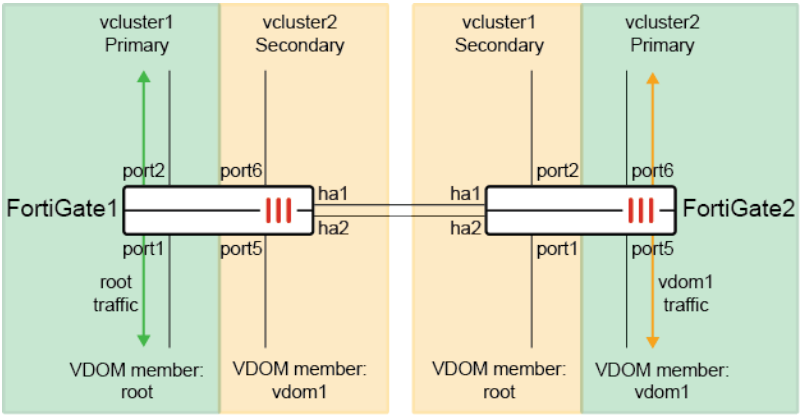
```
config system ha
    set vcluster-status enable
    config vcluster
        edit <id>
            set override {enable | disable}
            set priority <integer>
            set vdom <vdom_1>, ... [vdom_n]
            set monitor <interface_1>, ... [interface_n]
            set pingserver-monitor-interface <interface_1>, ... [interface_n]
```

```
        next
    end
end
```

Option	Description
`override {enable | disable}`	Enable/disable override and increase the priority of the unit that should always be the primary.
`priority <integer>`	Increase the priority to select the primary unit (0 - 255, default = 128).
`vdom <vdom_1>, ... [vdom_n]`	Set the virtual domains in the virtual cluster.
`monitor <interface_1>, ... [interface_n]`	Set the interfaces to check for port monitoring (or link failure).
`pingserver-monitor-interface <interface_1>, ... [interface_n]`	Set the interfaces to check for remote IP monitoring.

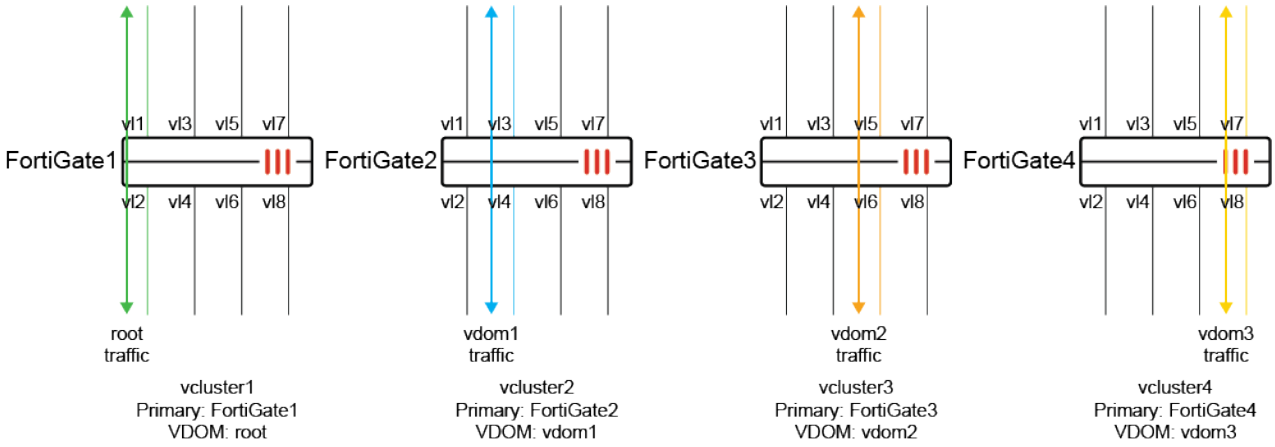
Active-passive virtual clustering uses VDOM partitioning to send traffic for some VDOMs to the primary FortiGate and traffic for other VDOMs to the secondary FortiGates. Traffic distribution between FortiGates can potentially improve throughput. If a failure occurs and only one FortiGate continues to operate, all traffic fails over to that FortiGate, similar to normal HA. If the failed FortiGates rejoin the cluster, the configured traffic distribution is restored.

In an active-passive virtual cluster of two FortiGates, the first and second FortiGates share traffic processing according to the VDOM partitioning configuration. The following is an example of two virtual clusters, with each member acting as primary for different vclusters.



If you add a third or fourth FortiGate, the first and second FortiGates process all traffic and the other one or two FortiGates operate in standby mode. If the first or second FortiGate fails, one of the other FortiGates becomes the new primary or secondary FortiGate and begins processing traffic.

For better load balancing, it is recommended to have as many vclusters as there are HA members. This way, each HA member can be a primary unit for each cluster, thereby processing traffic while standing by for the other vcluster as secondary. The following is an example of four FortiGates in an FGCP cluster, with four vclusters and four VDOMs. Each FortiGate is the primary unit for a vcluster and actively processes traffic as the primary member.



Virtual clustering and heartbeat interfaces

The HA heartbeat provides the same HA services in a virtual clustering configuration as in a standard HA configuration. One set of HA heartbeat interfaces provides HA heartbeat services for all of the VDOMs in the cluster. You do not have to add a heartbeat interface for each VDOM.

Special considerations for NPU-based VLANs in a virtual cluster

In an FGCP cluster, the primary FortiGate uses virtual MAC addresses when forwarding traffic, and the secondary uses the physical MAC addresses when forwarding traffic. In a virtual cluster, packets are sent with the cluster's virtual MAC addresses. However, in the case of NPU offloading on a non-root VDOM, traffic that leaves an NPU-based VLAN will use the physical MAC address of its parent interface rather than the virtual MAC address. If this behavior is not desired, disable `auto-asic-offload` in the firewall policy where the VLAN interface is used.

Support up to 30 virtual clusters

FortiOS supports up to 30 virtual clusters, which allows more VDOMs to be spread across different virtual clusters without overlapping. Each virtual cluster supports its own failover conditions. Prior to 7.2.0, only two virtual clusters were supported.

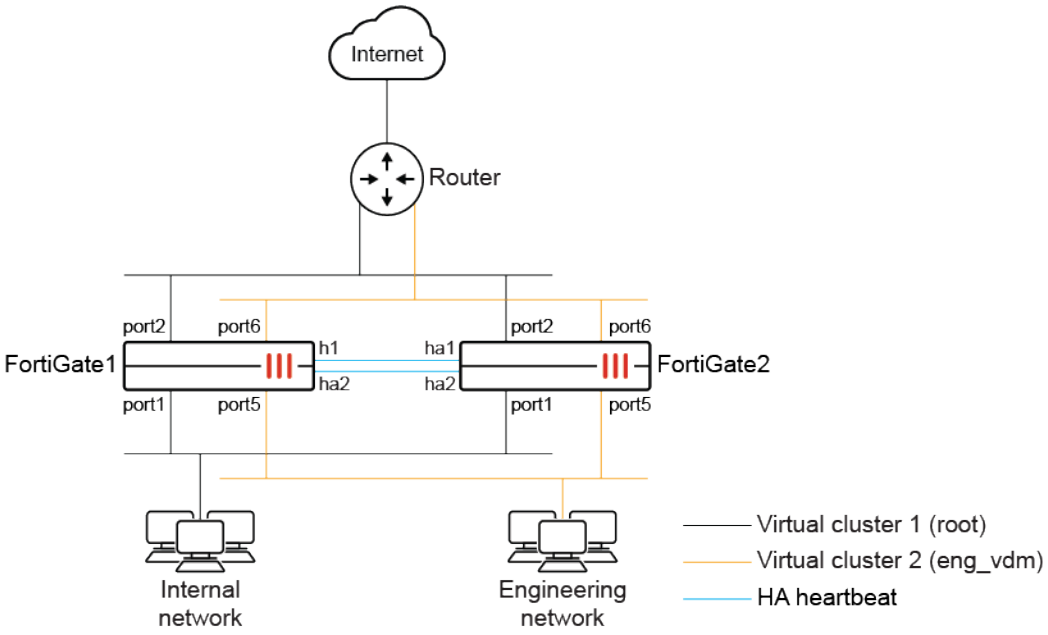
When configuring virtual clusters, the `group-id` is limited to a value from 0 to 7. If the HA `group-id` is greater than 7, use the command line first to change the `group-id` before enabling virtual clusters.

```
config system ha
    set group-id <integer>
end
```

⚠ When upgrading from 7.0 or earlier, old virtual clusters will be lost if the `group-id` is larger than 7.

Basic configuration

This example shows a virtual cluster configuration consisting of two FortiGates. The virtual cluster has two VDOMs, root and eng_vdm.



The root VDOM can only be associated with virtual cluster 1.

The VDOM that is assigned as the management VDOM can also only be associated with virtual cluster 1.

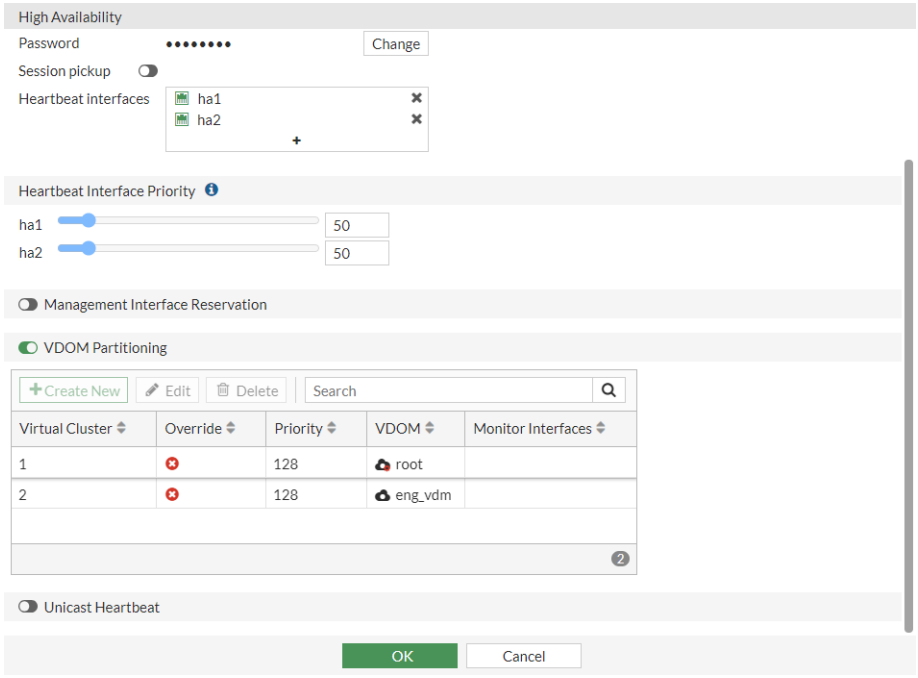
To set up an HA virtual cluster using the GUI:

1. Make all the necessary connections as shown in the topology diagram.
2. Configure a regular A-P cluster:
   a. Log in to one of the FortiGates.
   b. Go to *System* > *HA* and set the following options:

Mode	Active-Passive
Device priority	128 or higher
Group name	Example_cluster
Heartbeat interfaces	ha1 and ha2

   Except for the device priority, these settings must be the same on all FortiGates in the cluster.
   c. Leave the remaining settings as their default values. They can be changed after the cluster is in operation.
   d. Click *OK*.
   The FortiGate negotiates to establish an HA cluster. Connectivity with the FortiGate may be temporarily lost as the HA cluster negotiates and the FGCP changes the MAC addresses of the FortiGate's interfaces.
   e. Factory reset the other FortiGate that will be in the cluster, configure GUI access, then repeat step 2 (omitting setting the device priority) to join the cluster.
3. On the primary FortiGate, go to *System* > *Settings* > *General* and enable *Virtual domains*.

4. Click *Apply*. You will be logged out of the FortiGate.
5. Log back in to the FortiGate, and ensure that you are in the global VDOM.
6. Create the eng_vdm VDOM:
   a. Go to *System* > *VDOM* and click *Create New*. The *New Virtual Domain* pane opens.
   b. Enter the name in the *Virtual Domain* field, then click *OK*.
7. Implement a virtual cluster by moving the new VDOM to virtual cluster 2:
   a. Go to *System* > *HA* and enable *VDOM Partitioning*.
   b. In the table, click *Create New*. The *New Virtual Cluster* pane opens.
   c. Click the *+* and add the *eng_vdm* VDOM.
   d. Click *OK* to save the virtual cluster.



   e. Click *OK* to save the HA configuration.

To set up an HA virtual cluster using the CLI:

1. Make all the necessary connections as shown in the topology diagram.
2. Configure a regular A-P cluster. See HA active-passive cluster setup on page 4079.
3. Enable VDOMs:

```
config system global
    set vdom-mode multi-vdom
end
```

   You will be logged out of the FortiGate.
4. Create the eng_vdm VDOM:

```
config vdom
    edit eng_vdm
```

```
    next
end
```

5. Reconfigure the HA settings to be a virtual cluster:

```
config system ha
    set vcluster-status enable
    config vcluster
        edit 1
            set vdom root
            set override disable
        next
        edit 2
            set vdom eng_vdm
            set override disable
        next
    end
end
```

Configuration with 30 virtual clusters

In this example, there are 30 customers managed by an MSSP on an HA cluster, and each customer VDOM needs to failover independently of other customer VDOMs. Each customer is assigned to a different virtual cluster with its own virtual cluster configurations. This may include different monitored interfaces, ping servers, and priority for the primary and secondary cluster members. Each virtual cluster will fail over according to their own virtual cluster configurations.

This example assumes an A-P cluster and VDOMs have already been configured. See HA active-passive cluster setup on page 4079 and Virtual Domains on page 3835 for more information.

For each virtual cluster, this example assumes that unit 1 has an HA priority of 200, while unit 2 has an HA priority of 100. By default, unit 1 will be the primary cluster member of all the virtual clusters.

To configure multiple virtual clusters in the GUI:

1. Go to *System* > *HA* and enable *VDOM Partitioning*.
2. Create a virtual cluster:
   a. In the table, click *Create New*. The *New Virtual Cluster* pane opens.
   b. Set the *Device priority* to *200*.
   c. Click the *+* and add the *Virtual domains*.
   d. Optionally, click the *+* and add the *Monitor interfaces*.
   e. Click *OK*.
3. Repeat step 2 to create the remaining virtual clusters.
4. Click *OK* to save the HA configuration. The *HA* page summary displays the multiple virtual clusters, each with a *Primary* and *Secondary* HA member.
5. Edit the priority settings for the secondary members to be 100:
   a. Select the *Secondary* member in the table, and click *Edit*.
   b. Set the *Priority* to *100*.
   c. Click *OK*.

6. Repeat step 5 for the remaining secondary members.

To configure multiple virtual clusters in the CLI:

1. Configure the primary FortiGate:

```
config system ha
    set vcluster-status enable
    config vcluster
        edit 1
            set override disable
            set priority 200
            set vdom "vdom1"
        next
        edit 2
            set override disable
            set priority 200
            set vdom "vdom2"
        next
        ...
        edit 30
            set override disable
            set priority 200
            set vdom "vdom30"
        next
    end
end
```

2. Configure the secondary FortiGate:

```
config system ha
    set vcluster-status enable
    config vcluster
        edit 1
            set override disable
            set priority 100
            set vdom "vdom1"
        next
        edit 2
            set override disable
            set priority 100
            set vdom "vdom2"
        next
        ...
        edit 30
            set override disable
            set priority 100
            set vdom "vdom30"
        next
    end
end
```

HA and load balancing

FGCP active-active HA uses a technique similar to unicast load balancing where the primary unit is associated with the cluster HA virtual MAC addresses and cluster IP addresses. The primary unit is the only cluster unit that receives packets sent to the cluster. The primary unit uses a load balancing schedule to distribute sessions to all cluster units (including the primary unit). Subordinate unit interfaces retain their actual MAC addresses, and the primary unit communicates with the subordinate units using these MAC addresses. Packets exiting the subordinate units proceed directly to their destination and do not pass through the primary unit.

By default, active-active HA load balancing distributes proxy-based security profile processing to all cluster units. Proxy-based security profile processing is CPU and memory-intensive, so FGCP load balancing may result in higher throughput because resource-intensive processing is distributed among all cluster units.

The following proxy-based security profile processing is load balanced:

- Virus scanning
- Web filtering
- Email filtering
- Data Loss prevention (DLP) of HTTP, FTP, IMAP, IMAPS, POP3, POP3S, SMTP, SMTPS, IM, and NNTP sessions accepted by firewall policies

Other features enabled in firewall policies such as endpoint security, traffic shaping, and authentication have no effect on active-active load balancing.

Active-active HA load balancing does not support software switches. See Software switch on page 278.

The `load-balance-all` option can be enabled to have the primary unit load balance all TCP sessions. Load balancing TCP sessions increases overhead and may actually reduce performance. This setting is disabled by default.

To configure TCP session load balancing:

```
config system ha
    set load-balance-all {enable | disable}
end
```

NP6 and NP7 processors can offload and accelerate load balancing. See NP session offloading in HA active-active configuration for more information.

During active-active HA load balancing, the primary unit uses the configured load balancing schedule to determine which cluster unit will process a session. The primary unit stores the load balancing information for each load balanced session in the cluster load balancing session table. Using the information in this table, the primary unit can then forward all of the remaining packets in each session to the appropriate cluster unit. The load balancing session table is synchronized among all cluster units.

ICMP, multicast, and broadcast sessions are never load balanced and are always processed by the primary unit. The following sessions are only processed by the primary unit:

- IPS
- Application control
- Flow-based virus scanning
- Flow-based web filtering
- Flow-based DLP
- Flow-based email filtering
- VoIP
- IM
- P2P
- IPsec VPN
- Agentless VPN
- HTTP multiplexing
- SSL offloading
- WAN optimization
- Explicit web proxy
- WCCP

In addition to load balancing, active-active HA provides the same session, device, and link failover protection as active-passive HA. If the primary unit fails, a subordinate unit becomes the primary unit and resumes operating the cluster. Active-active HA maintains as many load balanced sessions as possible after a failover by continuing to process the load balanced sessions that were being processed by the cluster units that are still operating.

Load balancing schedules

The load balancing schedule controls how the primary unit distributes packets to all cluster units.

To configure the load balancing schedule:

```
config system ha
    set schedule {none | leastconnection | round-robin | weight-round-robin | random | ip |
ipport}
end
```

The following table outlines the load balancing schedule options.

Schedule	Description
None	Use no load balancing. Select this option when the cluster interfaces are connected to load balancing switches. The primary unit does not load balance traffic, and the subordinate units process incoming traffic that does not come from the primary unit. For all other load balancing schedules, all traffic is received first by the primary unit and then forwarded to the subordinate units. The subordinate units only receive and process packets sent from the primary unit.

Schedule	Description
Least connection	Distribute network traffic to the cluster unit currently processing the fewest connections.
Round robin	Distribute network traffic to the next available cluster unit.
Weighted round robin	This is similar to round robin, but weighted values are assigned to each cluster unit based on their capacity and how many connections they are currently processing. For example, the primary unit should have a lower weighted value because it handles scheduling and forwards traffic. Weighted round robin distributes traffic more evenly because units that are not processing traffic will be more likely to receive new connections than units that are very busy.
Random	Randomly distribute traffic to cluster units.
IP	Distribute traffic to cluster units based on the source and destination IP of the packet.
IP port	Distribute traffic to cluster units based on based on the source IP, source port, destination IP, and destination port of the packet.

Once a packet has been propagated to a subordinate unit, all packets are part of that same communication session are propagated to that same subordinate unit. Traffic is distributed according to the communication session, not just an individual packet.

Any subordinate unit that receives a forwarded packet processes it without applying load balancing. Note that subordinate units are still considered to be active because they perform routing, virus scanning, and other tasks on their share of the traffic. Active subordinate units share their session and link status information with all cluster units. Active subordinate units do not make load balancing decisions.

The primary unit is responsible for the load balancing process, and still performs other FortiGate tasks. Depending on the load balancing schedule used, the primary unit may assign itself a smaller share of the total load.

Active-active failover

If a subordinate unit fails, the primary unit redistributes the sessions that the subordinate was processing among the remaining active cluster members. If the primary unit fails, the subordinate units negotiate to select a new primary unit. The new primary unit continues to distribute packets among the remaining active cluster units.

Failover works in a similar way if the cluster consists of only two units. If the primary unit fails, the subordinate unit negotiates and becomes the new primary unit. If the subordinate unit fails, the primary unit processes all traffic. In both cases, the single remaining unit continues to function as a primary unit, maintaining the HA virtual MAC address for all of its interfaces.

HTTPS sessions and active-active load balancing

In proxy inspection mode, active-active HA does not load balance HTTPS sessions that have SSL deep packet scanning or certificate inspection enabled. For example, when proxy inspection is triggered for HTTPS traffic in web or AV filtering, the traffic must be processed by the primary HA unti in proxy inspection mode.

FortiGate identifies HTTPS sessions as all sessions received on the HTTPS TCP port. The default HTTPS port is 443. If the HTTPS port is changed in the SSL/SSH inspection profile applied in the firewall policy, FGCP stops load balancing all sessions that use the custom HTTPS port.

HTTPS traffic passing through a firewall policy that does not meet the proxy inspection mode criteria can still be load balanced when `load-balance-all` is enabled. HTTPS traffic passing through flow-based inspection does not have this limitation.

HA heartbeat

The HA heartbeat allows cluster units to communicate with each other. The heartbeat consists of hello packets that are sent at regular intervals by the heartbeat interface of all cluster units. The hello packets describe the state of the cluster unit (including communication sessions) and are used by other cluster units to keep the cluster synchronized. While the cluster is operating, the HA heartbeat confirms that all cluster units are functioning normally.

Key aspects of the HA Heartbeat include:

- **Synchronization**: The heartbeat link transports session information, configuration changes, kernel routing tables, and IPsec keys to ensure all units are aligned.
- **Communication Protocols**: Standard heartbeats utilize specific Layer 2 Ethernet frames (EtherTypes 0x8890, 0x8891, and 0x8893) to identify different types of synchronization traffic.
- **Addressing**: The cluster automatically creates virtual interfaces in a hidden VDOM (vsys_ha) and assigns them link-local IPv4 addresses (169.254.0.x) to facilitate communication.
- **Failover Detection**: The system monitors the heartbeat interval and lost threshold; if a peer stops sending packets for a specified duration, it is declared dead, triggering a failover.

The following sections provide detailed configuration guidelines and behavioral insights for HA heartbeat:

Topic	Summary
HA heartbeat interface	The standard configuration for physical appliances where Layer 2 Ethernet frames are used on dedicated interfaces to broadcast heartbeat and synchronization data.
HA for cloud and virtual environments	An alternative configuration for cloud and virtual environments that do not support Layer 2 broadcasts, instead utilizing Layer 3 IP packets to maintain the cluster connection. Once Unicast communication is established, the focus shifts to synchronization; because cluster members often reside in different physical locations or availability zones, they require unique settings that must be excluded from synchronization.
Advanced HA heartbeat settings	Tailors cluster behavior to specific network environments by adjusting interval thresholds for precise failover control, modifying startup wait times for stability in high-latency sites, and applying encryption to prevent injection attacks
Heartbeat bandwidth requirements	The throughput capacity needed for the heartbeat link, which is primarily driven by session synchronization traffic and can surge significantly during failover events or when a new unit joins the cluster.

Topic	Summary
Heartbeat packet EtherTypes	The use of specific, non-standard Layer 2 protocol identifiers (such as 0x8890 through 0x8893) to distinguish between different HA functions like hello packets, session sync, and configuration management.
FGCP internal communication and IP assignment	The automatic assignment of virtual Link-Local IPv4 addresses (169.254.0.x) to hidden internal interfaces, allowing cluster members to route management traffic and logs between units despite using Layer 2 transport.
Troubleshooting heartbeat packets	The diagnostic process of verifying cluster communication, often involving packet sniffing for specific EtherTypes to ensure that intermediate switches are not blocking the Layer 2 frames required for HA.

HA heartbeat interface

HA heartbeat packets are Layer 2 Ethernet frames that use EtherType values of 0x8890 and 0x8891 rather than 0x0800 for normal 802.3 IP packets. The default time interval between HA heartbeats is 200 ms.

As a best practice, it is recommended to isolate the heartbeat devices from the user networks by connecting the heartbeat devices to a dedicated switch that is not connected to any network. The heartbeat packets contain sensitive information about the cluster configuration and may use a considerable amount of network bandwidth. If the cluster consists of two FortiGates, connect the heartbeat device interfaces back-to back using a crossover cable. If there are more than two FortiGates, each heartbeat interface should be connected to a dedicated switch. For example, in a four-member HA cluster with two heartbeat interfaces, there would be two switches (one switch dedicated to each interface).

Upon starting up, a FortiGate configured for HA broadcasts HA heartbeat hello packets from its HA heartbeat interface to find other FortiGates configured to operate in HA mode. If two or more FortiGates operating in HA mode connect with each other, they compare HA configurations (mode, password, and group ID). If the HA configurations match, then the units negotiate to form a cluster.

The HA heartbeat interface communicates with each unit in the cluster using the same heartbeat interface for each member.

For example, if port1 and port2 are the heartbeat interfaces for the HA cluster, then in a cluster consisting of two members:

- port1 of the primary FortiGate should be connected to port1 of the secondary FortiGate.
- port2 of the primary FortiGate should be connected to port2 of the secondary FortiGate.

Configuring an HA heartbeat interface

A heartbeat interface is an Ethernet network interface in a cluster that is used by the FGCP for HA heartbeat communications between cluster units.

By default, two interfaces are configured to be heartbeat interfaces on most FortiGate models. The heartbeat interface configuration can be changed to select an additional or different heartbeat interface. It is possible to select only one heartbeat interface; however, this is not a recommended configuration (see Split brain scenario on page 4198).

Another important setting in the HA configuration is the heartbeat interface priority. In all cases, the heartbeat interface with the highest priority is used for all HA heartbeat communication. If the interface fails or becomes disconnected, then the selected heartbeat interface with the next highest priority handles all HA heartbeat communication.

If more than one heartbeat interface has the same priority, the heartbeat interface with the highest priority that is also highest in the heartbeat interface list is used for all HA heartbeat communication. If this interface fails or becomes disconnected, then the selected heartbeat interface with the highest priority that is next highest in the list handles all heartbeat communication (see Selecting heartbeat packets and interfaces on page 4099).

The default heartbeat interface configuration sets the priority of both heartbeat interfaces to 50, and the range is 0 to 512. When selecting a new heartbeat interface, the default priority is 0. The higher the number, the higher the priority.

In most cases, the default heartbeat interface configuration can be maintained as long the heartbeat interfaces are connected. Configuring HA heartbeat interfaces is the same for virtual clustering and for standard HA clustering. Up to eight heartbeat interface can be selected. This limit only applies to FortiGates with more than eight physical interfaces.

Heartbeat communications can be enabled on physical interfaces, but not on switch ports, VLAN subinterfaces, IPsec VPN interfaces, redundant interfaces, or 802.3ad aggregate interfaces.

To change the heartbeat interfaces in the GUI:

1. Go to *System* > *HA* and select a *Mode*.
2. Click the *+* in the *Heartbeat interfaces* field to select an interface.
3. Click *OK*.

To configure two interfaces as heartbeat interfaces with the same priority in the CLI:

```
config system ha
    set hbdev port4 150 port5 150
end
```

In this example, port4 and port5 are configured as the HA heartbeat interfaces and they both have a priority of 150.

To configure two interfaces as heartbeat interfaces with different priorities in the CLI:

```
config system ha
    set hbdev port4 100 port1 50
end
```

In this example, port4 and port1 are configured as the HA heartbeat interfaces. The priority for port4 is higher (100) than port1 (50), so port4 is the preferred HA heartbeat interface.

Selecting heartbeat packets and interfaces

HA heartbeat hello packets are sent constantly by all of the enabled heartbeat interfaces. Using these hello packets, each cluster unit confirms that the other cluster units are still operating. The FGCP selects one of the

heartbeat interfaces to be used for communication between the cluster units. This interface is used for heartbeat communication and is based on the linkfail states of the heartbeat interfaces, the heartbeat interface priority, and the interface index. The connected heartbeat interface with the highest priority is selected for heartbeat communication.

If more than one connected heartbeat interface has the highest priority, then the FGCP selects the heartbeat interface with the lowest interface index. The interface index order is visible in the CLI by running the `diagnose netlink interface list` command.

If the interface that is processing heartbeat traffic fails or becomes disconnected, the FGCP uses the same criteria to select another heartbeat interface for heartbeat communication. If the original heartbeat interface is fixed or reconnected, the FGCP selects this interface again for heartbeat communication.

The HA heartbeat interface communicates cluster session information, synchronizes the cluster configuration, synchronizes the cluster kernel routing table, and reports individual cluster member statuses. The HA heartbeat constantly communicates HA status information to make sure that the cluster is operating properly.

Separating heartbeat interfaces from traffic ports

As a best practice, it is recommended to isolate the heartbeat devices from the user networks by connecting the heartbeat devices to a dedicated switch that is not connected to any network. The heartbeat packets contain sensitive information about the cluster configuration and may use a considerable amount of network bandwidth. If the cluster consists of two FortiGates, connect the heartbeat device interfaces back-to back using a crossover cable. If there are more than two FortiGates, each heartbeat interface should be connected to a dedicated switch. For example, in a four-member HA cluster with two heartbeat interfaces, there would be two switches (one switch dedicated to each interface).

HA heartbeat and data traffic is supported on the same cluster interface. While these configurations are allowable, they are not recommended.

- **NAT mode**: if the heartbeat interfaces are used for processing network traffic, then the interface can be assigned any IP address. The IP address does not affect HA heartbeat traffic.
- **Transparent mode**: the heartbeat interface can be connected to the network with management access enabled on the same interface. A management connection would then be established to the interface using the transparent mode management IP address. This configuration does not affect HA heartbeat traffic.

Adding a backup heartbeat interface

At least one heartbeat interface must be selected for the HA cluster to function correctly. This interface must be connected to all the units in the cluster. If heartbeat communication is interrupted and cannot fail over to a second heartbeat interface, then the cluster units will not be able to communicate with each other and more than one cluster unit may become a primary unit. As a result, the cluster stops functioning normally because multiple devices on the network may be operating as primary units with the same IP and MAC addresses creating a split brain scenario. See Split brain scenario on page 4198 for more information.

A backup heartbeat interface is available to help prevent split-brain scenarios.

The backup heartbeat can be configured in the GUI on the *System > HA* page, or in the CLI:

```
config system ha
    set backup-hbdev <interface(s)>
end
```

The backup heartbeat is a dedicated interface that is automatically used when a secondary unit detects no heartbeats from the primary unit through the heartbeat interface(s). The backup heartbeat interface is no longer used when the secondary unit detects a heartbeat again.

Consider the following when using a backup heartbeat interface:

- A split-brain happens specifically when the secondary unit cannot detect heartbeats from the primary unit, and it promotes itself to primary. Therefore, the `backup-hbdev` is used only when the secondary unit cannot detect heartbeats. When the `backup-hbdev` is in use, the setting cannot be changed.
- The backup heartbeat interface does not bind to the virtual `port_ha` interface. Its main purpose is to operate efficiently to maintain the HA cluster and continue the flow of traffic. Therefore, some functions are not available by design.
- Configuration changes are not synchronized to the secondary member in the HA cluster while the backup heartbeat interface is in use.
- Without using `session-sync-dev`, the `session-sync` and `session-pickup` events will not occur.

HA for cloud and virtual environments

In virtual machine (VM) and cloud environments that do not support heartbeat communication with Layer 2 Ethernet frames (see HA heartbeat interface on page 4098), you can set up a Layer 3 unicast HA heartbeat when configuring HA and VDOM Exceptions to manage necessary configuration differences between the cluster members.

Unicast HA heartbeat

This method allows the HA cluster to form by sending heartbeat packets directly to a specified Peer IP address. This requires enabling the feature and adding a peer IP address. The peer IP address is the IP address of the HA heartbeat interface of the other FortiGate VM in the HA cluster.

Prerequisites and constraints:

- Unicast HA is only supported between two FortiGate VMs in active-passive (A-P) mode.
- The heartbeat interfaces must be connected to the same network, and the IP addresses must be added to these interfaces.

In the following example, unicast HA heartbeat is enabled over the port3 interface.

To enable unicast HA heartbeat in the GUI:

1. Go to *System* > *HA*.
2. Enable *Unicast Heartbeat* and enter the *Peer IP*, such as *172.30.3.12*.
3. Click *OK*.

To enable unicast HA heartbeat in the CLI:

```
config system ha
    set hbdev port3 50
    set unicast-hb enable
```

```
    set unicast-hb-peerip 172.30.3.12
end
```

VDOM exceptions

VDOM exceptions are settings that can be selected for specific VDOMs or all VDOMs that are not synchronized to other HA members. This can be required when cluster members are not in the same physical location, subnets, or availability zones in a cloud environment.

Some examples of possible use cases include:

- You use different source IP addresses for FortiAnalyzer logging from each cluster member. See Override FortiAnalyzer and syslog server settings on page 4171 for more information.
- You need to keep management interfaces that have specific VIPs or local subnets that cannot transfer from being synchronized.
- In a unicast HA cluster in the cloud, you use NAT with different IP pools in different subnets, so IP pools must be exempt.
- In a unicast HA cluster in the cloud, when HA members have different interface IPs, the local gateway (`local-gw`) used to define the local end of the VPN tunnel may need to be specified individually for IPsec tunnel failover to occur.

When a VDOM exception is configured, the object will not be synchronized between the primary and secondary devices when the HA forms. Different options can be configured for every object.

When VDOM mode is disabled, the configured object is excluded for the entire device. To define a scope, VDOM mode must be enabled and the object must be configurable in a VDOM.

VDOM exceptions are synchronized to other HA cluster members.

To configure VDOM exceptions:

```
config global
    config system vdom-exception
        edit 1
            set object <object name>
            set scope {all* | inclusive | exclusive}
            set vdom <vdom name>
        next
    end
end
```

Option	Description
`object`	The name of the configuration object that can be configured independently for some or all of the VDOMs. See Objects on page 4103 for a list of available settings and resources.
`scope`	Determine if the specified object is configured independently for all VDOMs or a subset of VDOMs. • `all`: Configure the object independently on all VDOMs. • `inclusive`: Configure the object independently only on the specified

Option	Description
	VDOMs. • `exclusive`: Configure the object independently on all of the VDOMs that are not specified.
`vdom`	The names of the VDOMs that are included or excluded.

Objects

The following settings and resources can be exempt from synchronization in an HA cluster:

log.fortianalyzer.setting	user.radius
log.fortianalyzer.override-setting	system.interface*
log.fortianalyzer2.setting	vpn.ipsec.phase1-interface*
log.fortianalyzer2.override-setting	vpn.ipsec.phase2-interface*
log.fortianalyzer3.setting	router.bgp*
log.fortianalyzer3.override-setting	router.route-map*
log.fortianalyzer-cloud.setting	router.prefix-list*
log.fortianalyzer-cloud.override-setting	firewall.ippool*
log.syslogd.setting	firewall.ippool6*
log.syslogd.override-setting	router.static*
log.syslogd2.setting	router.static6*
log.syslogd2.override-setting	firewall.vip*
log.syslogd3.setting	firewall.vip6*
log.syslogd3.override-setting	system.sdwan*
log.syslogd4.setting	system.saml*
log.syslogd4.override-setting	router.policy*
system.central-management	router.policy6*
system.csf	
system.snmp.sysinfo	

* This setting can only be configured on cloud VMs.

Advanced HA heartbeat settings

Administrators can use advanced heartbeat settings to fine-tune cluster performance.

Configure the following settings for your environment as needed:

Topic	Summary
Modifying heartbeat timing	The customization of interval settings and loss thresholds to control how fast the cluster detects a peer failure allowing administrators to balance between sub-second failover speeds and tolerance for network latency.

Topic	Summary
Changing the time to wait in the hello state	A configuration setting that adjusts how long a unit searches for peers during startup or recovery before initializing ensuring stability in environments with communication delays between sites.
Configuring HA heartbeat encryption and authentication	A security feature that applies AES-128 encryption and SHA1 authentication to heartbeat packets protecting cluster credentials and preventing the injection of false status messages on shared networks.

Modifying heartbeat timing

The heartbeat interval and heartbeat lost threshold are two variables that dictate the length of time one cluster unit will wait before determining a peer is dead.

```
config system ha
    set hb-interval <integer>
    set hb-interval-in-milliseconds {100 | 10}
    set hb-lost-threshold <integer>
end
```

Option	Description
`hb-interval <integer>`	Set the time between sending heartbeat packets; increase to reduce false positives (1 - 20, default = 2).
`hb-interval-in-milliseconds {100 | 10}`	Set the number of milliseconds for each heartbeat interval (100 or 10, default = 100).
`hb-lost-threshold <integer>`	Set the number of lost heartbeats to signal a failure; increase to reduce false positives (1 - 60, default = 20).

Heartbeats are sent out every 2 × 100 ms, and it takes 20 consecutive lost heartbeats for a cluster member to be detected as dead. Therefore, it takes by default 2 × 100 ms × 20 = 4000 ms, or 4 seconds, for a failure to be detected.

Sub-second heartbeat failure detection can be achieved by lowering the interval and threshold or lowering the heartbeat interval unit of measurement from 100 ms to 10 ms.

If the primary unit does not receive a heartbeat packet from a subordinate unit before the heartbeat threshold expires, the primary unit assumes that the subordinate unit has failed.

If a subordinate unit does not receive a heartbeat packet from the primary unit before the heartbeat threshold expires, the subordinate unit assumes that the primary unit has failed. The subordinate unit then begins negotiating to become the new primary unit.

The HA heartbeat packets consume more bandwidth if the heartbeat interval is short. But if the heartbeat interval is very long, the cluster is not as sensitive to topology and other network changes. Therefore, gauge your settings based on the amount of traffic and CPU usage sustainable by the cluster units versus the tolerance for an outage when the primary unit fails. Avoid using the heartbeat interfaces as traffic ports to prevent congesting the interfaces.

Changing the time to wait in the hello state

The hello state hold down time is the number of seconds that a cluster unit waits before changing from hello state to work state. After a failure or when starting up, cluster units operate in the hello state to send and receive heartbeat packets so that all the cluster units can find each other and form a cluster. A cluster unit should change from the hello state to work state after it finds all the other FortiGates to form a cluster with.

If all cluster units cannot find each other during the hello state, then some cluster units may join the cluster after it has formed. This can cause disruptions to the cluster and affect how it operates. A delay could occur if the cluster units are located at different sites or if communication is delayed between the heartbeat interfaces. If delays occur, increase the cluster units wait time in the hello state.

```
config system ha
    set hello-holddown <integer>
end
```

Option	Description
`hello-holddown <integer>`	Set the time to wait before changing from hello to work state, in seconds (5 - 300, default = 20).

Configuring HA heartbeat encryption and authentication

HA heartbeat encryption and authentication to encrypt and authenticate HA heartbeat packets can be enabled. HA heartbeat packets should be encrypted and authenticated if the cluster interfaces that send HA heartbeat packets are also connected to the networks. HA heartbeat encryption and authentication are disabled by default. Note that enabling these settings could reduce cluster performance.

```
config system ha
    set authentication {enable | disable}
    set encryption {enable | disable}
end
```

If HA heartbeat packets are not encrypted, the cluster password and changes to the cluster configuration could be exposed. An attacker may be able to sniff HA packets to get cluster information. Enabling HA heartbeat message authentication prevents an attacker from creating false HA heartbeat messages. False HA heartbeat messages could affect the stability of the cluster.

HA authentication and encryption uses AES-128 for encryption and SHA1 for authentication. Heartbeat messages are encrypted and encapsulated in ESP packets for transfer in an IPsec tunnel between the cluster members.

Heartbeat bandwidth requirements

The heartbeat interface by default carries heartbeat packets and session synchronization packets. When session pickup is enabled, session synchronization traffic will consume a lot of bandwidth on the heartbeat interface. Other heartbeat interface traffic required to synchronize IPsec states, IPsec keys, routing tables, configuration changes, and so on is usually negligible.

The amount of traffic required for session synchronization depends on the connections per second (CPS) that the cluster is processing, since only new sessions (and session table updates) need to be synchronized.

Another factor to consider is that if session pickup is enabled, the traffic on the heartbeat interface surges during a failover or when a unit joins or re-joins the cluster. When one of these events occurs, the entire session table needs to be synchronized. Lower throughput HA heartbeat interfaces may increase failover time if they cannot handle the higher demand during these events.

The amount of heartbeat traffic can also be reduced by:

- Turning off session pickup if it is not needed
- Enabling `session-pickup-delay` to reduce the number of sessions that are synchronized
- Using the `session-sync-dev` option to move session synchronization traffic off of the heartbeat link

See Improving session sync performance for more information

Heartbeat packet EtherTypes

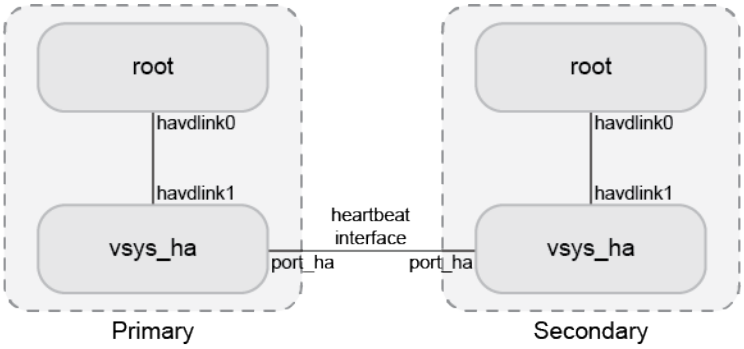
Normal 802.3 IP packets have an EtherType field value of 0x0800. EtherType values other than 0x0800 are understood as Layer 2 frames rather than IP packets.

HA heartbeat packets use the following EtherTypes:

Field value	Function	Description
0x8890	Heartbeat	Heartbeat packets are used by cluster units to find other cluster units, and to verify the status of other cluster units while the cluster is operating. Use the `ha-eth-type` option to change the EtherType.
0x8891	Traffic redistribution from primary to subordinate	These are used when the HA primary needs to redistribute traffic packets and the corresponding session information to the subordinate units in A-A mode. Use the `hc-eth-type` option to change the EtherType.
0x8892	Session synchronization	Session synchronization uses the heartbeat interfaces for communication, unless session synchronization devices are specified.
0x8893	HA Telnet sessions (configuration synchronization)	The Telnet sessions are used to synchronize the cluster configurations, and to connect from one cluster unit's CLI to another when an administrator uses the `execute ha manage` command. Use the `l2ep-eth-type` option to change the EtherType.

FGCP internal communication and IP assignment

An FGCP cluster communicates heartbeat packets using Layer 2 frames over the physical heartbeat interface, but it also communicates other synchronization traffic, logs, and locally generated traffic from subordinate devices over Layer 3 IP packets. Additional virtual interfaces are created in the hidden vsys_ha VDOM, which need to be addressed with IPv4 addresses.



The FGCP uses link-local IPv4 addresses (see RFC 3927) in the 169.254.0.x range for the virtual HA heartbeat interface (port_ha) and for the inter-VDOM link interfaces between the vsys_ha and management VDOM. When members join an HA cluster, each member's heartbeat interface (port_ha) is assigned an IP address from the range of 169.254.0.1 to 169.254.0.63/26. HA inter-VDOM link interfaces (havdlink0 and havdlink1) are assigned IP address from the range of 169.254.0.65 to 169.254.0.66/26.

The IP address that is assigned to a virtual heartbeat interface depends on the serial number priority of the member. Higher serial numbers have a higher priority, and therefore a lower `serialno_prio` number, for example:

```
# diagnose sys ha status
...
FGVM08TM20002002:    Secondary, serialno_prio=0, usr_priority=128, hostname=FGVM08TM20002002
FGVM08TM19003001:      Primary, serialno_prio=1, usr_priority=128, hostname=FGVM08TM19003001
```

The member with `serialno_prio=0` is assigned IP address 169.254.0.1, `serialno_prio=1` is assigned 169.254.0.2, and so forth.

To view the HA heartbeat interface IP address of the primary unit:

```
# get system ha status
...
vcluster 1: work 169.254.0.2
...
```

To view all the assigned IP addresses of a device:

```
# diagnose ip address list
IP=172.16.151.84->172.16.151.84/255.255.255.0 index=3 devname=port1
IP=192.168.2.204->192.168.2.204/255.255.255.0 index=6 devname=port2
IP=10.10.10.1->10.10.10.1/255.255.255.0 index=9 devname=port3
IP=127.0.0.1->127.0.0.1/255.0.0.0 index=13 devname=root
```

```
IP=127.0.0.1->127.0.0.1/255.0.0.0 index=16 devname=vsys_ha
IP=169.254.0.2->169.254.0.2/255.255.255.192 index=17 devname=port_ha
IP=127.0.0.1->127.0.0.1/255.0.0.0 index=18 devname=vsys_fgfm
IP=169.254.0.65->169.254.0.65/255.255.255.192 index=19 devname=havdlink0
IP=169.254.0.66->169.254.0.66/255.255.255.192 index=20 devname=havdlink1
```

When generating traffic from a subordinate unit, traffic will be routed to the primary unit's port_ha virtual heartbeat interface. From there, if traffic is destined to another network, the traffic is routed from the vsys_ha VDOM to the management VDOM by the havdlink interfaces.

Use the `execute traceroute` command on the subordinate unit to display HA heartbeat IP addresses and the HA inter-VDOM link IP addresses.

To trace the route to an IP address on a subordinate unit:

```
# execute ha manage 1
# execute traceroute 172.20.20.10
traceroute to 172.20.20.10 (172.20.20.10), 32 hops max, 72 byte packets
1 169.254.0.1 0 ms 0 ms 0 ms
2 169.254.0.66 0 ms 0 ms 0 ms
3 172.20.20.10 0 ms 0 ms 0 ms
```

To run a sniffer trace on the primary unit to view the traffic flow:

```
# diagnose sniffer packet any 'net 169.254.0.0/24' 4 0 l
```

Troubleshooting heartbeat packets

Understanding the different types of heartbeat packets will ease troubleshooting. Heartbeat packets are recognized as Layer 2 frames. The switches and routers on the heartbeat network that connect to heartbeat interfaces must be configured to allow them to pass through. If Layer 2 frames are dropped by these network devices, then the heartbeat traffic will not be allowed between the cluster units.

For example, some third-party network equipment may not allow EtherType 0x8893. The unit can still be found in the HA cluster, but you would be unable to run `execute ha manage` to manage the other unit. Use the following settings to change the EtherTypes of the HA heartbeat packets, if they require changing them for the traffic to be forwarded on the connected switch.

```
config system ha
    set ha-eth-type <hex_value>
    set hc-eth-type <hex_value>
    set l2ep-eth-type <hex_value>
end
```

To change the EtherType values of the heartbeat and HA Telnet session packets:

```
config system ha
    set ha-eth-type 8895
```

```
    set l2ep-eth-type 889f
end
```

For troubleshooting issues with packets sent or received on the HA heartbeat ports, use the following diagnostic command to sniff the traffic by EtherType.

```
# diagnose sniffer packet any 'ether proto <EtherType_in_hex>' 6 0 1
```

To sniff the traffic on EtherType 0x8890:

```
# diagnose sniffer packet any 'ether proto 0x8890' 6 0 l
Using Original Sniffing Mode
interfaces=[any]
filters=[ether proto 0x8890]
2022-10-19 16:22:26.512813 port5 out Ether type 0x8890 printer hasn't been added to sniffer.
0x0000   0000 0000 0000 000c 293b e61c 8890 5201        ...
```

Session synchronization and failover

The FortiGate Clustering Protocol (FGCP) provides failover protection, meaning that a cluster can continue to provide FortiGate services even when one of the devices in the cluster encounters a problem that would result in the complete loss of connectivity for a stand-alone FortiGate unit. See Failover protection on page 4110 for more information.

To ensure this continuity extends to active user data and to enable Session Failover, the cluster relies on the session synchronization. It achieves this by synchronizing the session table from the primary to the secondary device. Session failover (also called session-pickup) is not enabled by default for FortiGate. See Session pickup on page 4112 for more information.

Session failover means that after a primary unit fails, communications sessions resume on the new primary unit with minimal or no interruption. Because the sessions were synchronized, the new primary unit recognizes the open sessions that were being handled by the cluster and continues to process them according to their last known state. See Session handling on page 4114 for more information.

Session-pickup has some limitations. For example, session failover is not supported for sessions being scanned by proxy-based security profiles. Session failover is supported for sessions being scanned by flow-based security profiles; however, flow-based sessions that fail over are not inspected after they fail over. For more limitations, see Pass-through sessions on page 4114.

Additionally, using the `session-sync-dev` option, you can select one or more FortiGate interfaces to use for synchronizing sessions as required for session pickup. See Improving session sync performance on page 4118 for more information.

To understand the depth of session failover, consider these specific examples of what is preserved during a failover:

Topic	Summary
Resume IPS scanning of ICCP traffic after HA failover	Security does not stop during a failover. The new primary is able to pick up the traffic and immediately resume IPS scanning on ICCP streams ensuring no malicious data slips through during the transition.
Session synchronization during HA failover for ZTNA proxy sessions	For ZTNA, the cluster synchronizes the proxy sessions. This means that users connected using ZTNA do not lose their verified trust status or connection when the primary device fails.

Failover protection

The FortiGate Clustering Protocol (FGCP) provides failover protection, meaning that a cluster can provide FortiGate services even when one of the devices in the cluster encounters a problem that would result in the complete loss of connectivity for a stand-alone FortiGate unit. Failover protection provides a backup mechanism that can be used to reduce the risk of unexpected downtime, especially in mission-critical environments.

FGCP supports failover protection in four ways:

1. If a link fails.
2. If a device loses power.
3. If an SSD fails.
4. If memory utilization exceeds the threshold for a specified amount of time.

When session-pickup is enabled in the HA settings, existing TCP session are kept, and users on the network are not impacted by downtime as the traffic can be passed without reestablishing the sessions.

When and how the failover happens

1. **Link fails**

   Before triggering a failover when a link fails, the administrator must ensure that monitor interfaces are configured. Normally, the internal interface that connects to the internal network, and an outgoing interface for traffic to the internet or outside the network, should be monitored. Any of those links going down will trigger a failover.
2. **Loss of power for active unit**

   When an active (primary) unit loses power, a backup (secondary) unit automatically becomes the active, and the impact on traffic is minimal. There are no settings for this kind of fail over.
3. **SSD failure**

   An HA failover can be triggered by an SSD failure.

   To enable an SSD failure triggering HA fail over:

   ```
   config system ha
       set ssd-failover enable
   end
   ```
4. **Memory utilization**

An HA failover can be triggered when memory utilization exceeds the threshold for a specific amount of time.

Memory utilization is checked at the configured sample rate (`memory-failover-sample-rate`). If the utilization is above the threshold (`memory-failover-threshold`) every time that it is sampled for the entire monitor period (`memory-failover-monitor-period`), then a failover is triggered.

If the FortiGate meets the memory utilization conditions to cause failover, but the last memory triggered failover happened within the timeout period (`memory-failover-flip-timeout`), then the failover does not occur. Other HA cluster members can still trigger memory based failovers if they meet the criteria and have not already failed within the timeout period.

After a memory based failover from FortiGate A to FortiGate B, if the memory usage on FortiGate A goes down below the threshold but the memory usage on FortiGate B is still below the threshold, then a failover is triggered, and FortiGate A becomes the primary device.

When you disable memory based failover, a new HA primary selection occurs to determine the primary device.

To configure memory based HA failover:

```
config system ha
    set memory-based-failover {enable | disable}
    set memory-failover-threshold <integer>
    set memory-failover-monitor-period <integer>
    set memory-failover-sample-rate <integer>
    set memory-failover-flip-timeout <integer>
end
```

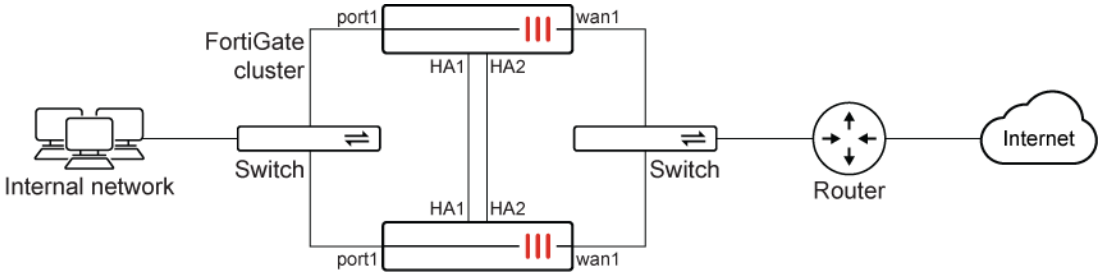
`memory-based-failover {enable | disable}`	Enable/disable memory based failover (default = disable).
`memory-failover-threshold <integer>`	The memory usage threshold to trigger a memory based failover, in percentage (0 - 95, 0 = use the conserve mode threshold, default = 0).
`memory-failover-monitor-period <integer>`	The duration of the high memory usage before a memory based failover is triggered, in seconds (1 - 300, default = 60).
`memory-failover-sample-rate <integer>`	The rate at which memory usage is sampled in order to measure memory usage, in seconds (1 - 60, default = 1).
`memory-failover-flip-timeout <integer>`	The time to wait between subsequent memory based failovers, in minutes (6 - 2147483647, default = 6).

Configuring HA failover time

On supported models, the HA heartbeat interval unit can be changed from the 100ms default to 10ms. This allows for a failover time of less than 50ms, depending on the configuration and the network.

```
config system ha
    set hb-interval-in-milliseconds {100ms | 10ms}
end
```

In this example, the HA heartbeat interval unit is changed from 100ms to 10ms. As the default heartbeat interval is two, this means that a heartbeat is sent every 20ms. The number of lost heartbeats that signal a failure is also changed to two. So, after two consecutive heartbeats are lost, a failover will be detected in 40ms.



To configure the HA failover time:

```
config system ha
    set group-id 240
    set group-name "300D"
    set mode a-p
    set password **********
    set hbdev "port3" 50 "port5" 100
    set hb-interval 2
    set hb-interval-in-milliseconds 10ms
    set hb-lost-threshold 2
    set override enable
    set priority 200
end
```

Session pickup

When session-pickup is enabled, the FGCP synchronizes the primary unit's TCP session table to all cluster units. As soon as a new TCP session is added to the primary unit's session table, that session is synchronized to all cluster units. This synchronization happens as quickly as possible to ensure the session tables remain synchronized.

If the primary unit fails, the new primary unit uses its synchronized session table to resume all TCP sessions that were being processed by the former primary unit, resulting in only minimal interruption. Under ideal conditions, all TCP sessions should be resumed. However, this is not guaranteed, and under less than ideal conditions, some TCP sessions may need to be restarted.

Session-pickup has some limitations. For example, session failover is not supported for sessions being scanned by proxy-based security profiles. Session failover is supported for sessions being scanned by flow-based security profiles; however, flow-based sessions that fail over are not inspected after they fail over. For more limitations, see Pass-through sessions on page 4114.

To enable session pickup in the GUI:

1. Go to *System* > *HA*.
2. Select the Primary FortiGate and click *Edit*.

3. Under *Cluster Settings*, enable *Session pickup*.
4. Click *OK* to save the setting.

To enable session pickup in the CLI:

```
config system ha
    set session-pickup enable
end
```

Enabling UDP, ICMP and broadcast packet session failover

By default, the FGCP does not maintain a session table for UDP, ICMP, or broadcast packets, even when session pickup is enabled. This means that the cluster does not specifically support the failover of these types of packets. However, it is possible to enable session pickup for UDP and ICMP packets. To do this, you must first enable session pickup for TCP sessions. After that, you can enable session pickup for connectionless sessions:

```
config system ha
    set session-pickup enable
    set session-pickup-connectionless enable
end
```

This configuration causes the cluster units to synchronize UDP and ICMP session tables and if a failover occurs UDP and ICMP sessions are maintained.

Enabling multicast session failover

HA supports IPv4 multicast route and session failover to ensure minimal interruption to multicast traffic. To adjust how often to synchronize the multicast routes to the secondary unit, use the following command to change the multicast TTL timer. The recommended setting to support multicast session failover is 120 seconds (2 minutes). The default setting is 600 seconds (10 minutes).

```
config system ha
    set multicast-ttl 120
end
```

The multicast TTL timer controls how long to keep synchronized multicast routes on the backup unit, ensuring they are present on the backup unit when it becomes the new primary unit after a failover. If you set the multicast TTL lower, the multicast routes on the backup unit are refreshed more often, and are therefore more likely to be accurate. However reducing this time causes route synchronization to happen more often, which could affect performance.

IPv4 multicast route and session failover can be supported in multicast-forwarding mode or multicast-routing mode.

Disabling session pickup

If you leave session pickup disabled, the cluster doesn't track sessions, and active sessions must be restarted or resumed after a failover. This is usually handled by TCP/IP communications.

The `session-pickup` setting does not affect session failover for sessions terminated by the cluster.

Disabling session pickup can reduce CPU and network bandwidth usage, especially if your cluster is mainly used for unsynchronized traffic. However, if session pickup is not enabled, sessions won't resume after a failover, causing a brief interruption. Most protocols can restart sessions with minimal data loss. For instance, web users can refresh their browsers to resume browsing, but large file downloads may need to be restarted. Some protocols may require manual session restarts, like FTP file downloads.

Session handling

FGCP handles session synchronization differently depending on whether the traffic is traversing the network or connecting directly to the cluster itself. This section outlines which protocols maintain state and which require re-establishment after an HA failover for two primary traffic categories:

Topic	Summary
Pass-through sessions	Sessions passing through the cluster
Terminated sessions	Sessions terminated by the cluster

Pass-through sessions

This section contains information about session failover for communication sessions passing through the cluster. In general, if session pickup is enabled, session failover is supported for most TCP traffic.

Protocol	Session failover
Most TCP sessions	Supported if session-pickup is enabled. See TCP session failover on page 4115 for more information.
Multicast sessions	Supported if multicast session-pickup is enabled. See Enabling multicast session failover on page 4113 for more information.
IPv6, NAT64, and NAT66	Supported if session-pickup is enabled.
Proxy-based security profile sessions	Not Supported; sessions have to be restarted. Proxy-based features require the FortiGate to maintain very large amounts of internal state information for each session. The FGCP does not synchronize this internal state information. As a result, proxy-based sessions are not failed over. Active-active clusters can resume some of these sessions after a failover. See Resume active-active HA subordinate units sessions after failover on page 4116 for more information.
Flow-based security profile sessions	Supported if session-pickup is enabled; however, internal state information is not synchronized so flow-based sessions that fail over are not inspected after they fail over. If both flow-based and proxy-based security profile features are applied to a TCP session, that session will not resume after a failover.

Protocol	Session failover
UDP, ICMP, or broadcast sessions	Supported if connectionless session-pickup is enabled. See Enabling UDP, ICMP and broadcast packet session failover on page 4113 for more information.
GPRS Tunneling Protocol (GTP)	Supported with limitations. See FortiOS Carrier GTP session failover on page 4115 for more information.
SIP	Supported for active-passive HA only. See SIP session failover on page 4115 for more information.
SIMPLE or SCCP signal session	Not supported; sessions have to be restarted.
SSL offloading and HTTP multiplexing	Not supported; sessions have to be restarted. See SSL offloading and HTTP multiplexing session failover on page 4116 for more information.

TCP session failover

TCP sessions that are not being processed by security profile features resume after a failover even if these sessions are accepted by security policies with security profiles. Only TCP sessions that are actually being processed by these security profile features do not resume after a failover.

- TCP sessions that are not virus scanned, web filtered, spam filtered, content archived, or are not SIP, SIMPLE, or SCCP signal traffic resume after a failover, even if they are accepted by a security policy with security profile options enabled. For example, SNMP TCP sessions through the FortiGate resume after a failover because FortiOS does not apply any security profile options to SNMP sessions.
- TCP sessions for a protocol for which security profile features have not been enabled resume after a failover even if they are accepted by a security policy with security profile features enabled. For example, if you have not enabled any antivirus or content archiving settings for FTP, FTP sessions resume after a failover.

SIP session failover

If session pickup is enabled, the FGCP supports SIP session failover (also called stateful failover) for active passive HA.

SIP session failover replicates SIP states to all cluster units. If an HA failover occurs, all in-progress SIP calls (setup complete) and their RTP flows are maintained and the calls will continue after the failover with minimal or no interruption.

SIP calls being set up at the time of a failover may lose signaling messages. In most cases the SIP clients and servers should use message retransmission to complete the call setup after the failover has completed. As a result, SIP users may experience a delay if their calls are being set up when an HA a failover occurs. But in most cases the call setup should be able to continue after the failover.

FortiOS Carrier GTP session failover

FortiOS Carrier HA supports GTP session failover. Once the GTP tunnel setup is completed, the primary unit synchronizes the GTP tunnel state across all cluster units. Although HA does not synchronize UDP sessions used by GTP, the new primary unit retains the GTP tunnel state information after a failover. This allows GTP UDP sessions using the same tunnel to continue to flow, albeit with certain limitations.

The limitation on packets continuing to flow is that there has to be a security policy to accept the packets. For example, if the FortiOS Carrier unit has an internal to external security policy, GTP UDP sessions using an established tunnel that are received by the internal interface are accepted by the security policy and can continue to flow. However, GTP UDP packets for an established tunnel that are received at the external interface cannot flow until packets from the same tunnel are received at the internal interface.

If you have bi-directional policies that accept GTP UDP sessions then traffic in either direction that uses an established tunnel can continue to flow after a failover without interruption.

SSL offloading and HTTP multiplexing session failover

SSL offloading and HTTP multiplexing requires the FortiGate to maintain very large amounts of internal state information for each session. Sessions accepted by security policies containing virtual IPs or virtual servers with SSL offloading or HTTP multiplexing enabled do not resume after a failover.

Resume active-active HA subordinate units sessions after failover

In an active-active cluster, subordinate units process sessions. After a failover, all cluster units that are still operating may be able to continue processing the sessions that they were processing before the failover. These sessions are maintained because after the failover the new primary unit uses the HA session table to continue to send session packets to the cluster units that were processing the sessions before the failover. Cluster units maintain their own information about the sessions that they are processing and this information is not affected by the failover. In this way, the cluster units that are still operating can continue processing their own sessions without loss of data.

The cluster keeps processing as many sessions as it can. But some sessions can be lost. Depending on what caused the failover, sessions can be lost in the following ways:

- A cluster unit (subordinate unit) fails. All sessions that were being processed by that cluster unit are lost.
- A link failure occurs. All sessions that were being processed through the network interface that failed are lost.

Terminated sessions

This section contains information about session failover for communication sessions terminated by the cluster. Sessions terminated by the cluster include management sessions as well as IPsec and Agentless VPN, WAN Optimization and so on between the cluster and a client.

In general, most sessions terminated by the cluster have to be restarted after a failover. There are some exceptions though. For example, the FGCP provides failover for IPsec and Agentless VPN sessions terminated by the cluster.

The session pickup setting does not affect session failover for sessions terminated by the cluster. Also other cluster settings such as active-active or active-passive mode do not affect session failover for sessions terminated by the cluster.

Protocol	Session failover
Administrative or management connections such as connecting to the GUI or CLI, SNMP, syslog, communication with FortiManager, FortiAnalyzer and so on	Not supported, sessions have to be restarted.
Explicit web proxy, WCCP, WAN Optimization and Web Caching	Not supported, sessions have to be restarted. See Explicit web proxy, explicit FTP proxy, WCCP, WAN optimization and Web Caching session failover on page 4117 for more information.
IPsec VPN tunnels terminating at the FortiGate	Supported. Security associations (SAs) and related IPsec VPN tunnel data is synchronized to cluster members. See IPsec VPN SA sync on page 4117 for more information.
Agentless VPN tunnels terminating at the FortiGate	Partially supported. Sessions are not synchronized and have to be restarted. Authentication failover and cookie failover are supported for Agentless VPN sessions. See Agentless VPN authentication failover on page 4118 for more information.
PPTP and L2TP VPN terminating at the FortiGate	Not supported; sessions have to be restarted. See PPTP and L2TP VPN sessions on page 4118 for more information.

Explicit web proxy, explicit FTP proxy, WCCP, WAN optimization and Web Caching session failover

Explicit web proxy, explicit FTP proxy, WCCP, WAN optimization and web caching sessions all require the FortiGate to maintain very large amounts of internal state information for each session. This information is not maintained and these sessions do not resume after a failover.

The active-passive HA clustering is recommended for WAN optimization. All WAN optimization sessions are processed by the primary unit only. Even if the cluster is operating in active-active mode, HA does not load-balance WAN optimization sessions.

Web cache and byte cache databases are only stored on the primary unit. These databases are not synchronized to the cluster. So, after a failover, the new primary unit must rebuild its web and byte caches. The new primary unit cannot connect to a SAS partition that the failed primary unit used.

Rebuilding the byte caches can happen relatively quickly because the new primary unit gets byte cache data from the other FortiGates that it is participating with in WAN optimization tunnels.

IPsec VPN SA sync

The FGCP synchronizes IPsec SAs between cluster members so that if a failover occurs, the cluster can resume IPsec sessions without having to establish new SAs. The result is improved failover performance because IPsec sessions are not interrupted to establish new SAs. Also, establishing a large number of SAs can reduce cluster performance.

Agentless VPN authentication failover

Authentication failover is supported for Agentless VPN sessions. This means that after a failover, Agentless VPN sessions can re-establish the Agentless VPN session between the Agentless VPN client and the FortiGate without having to authenticate again.

All sessions inside the Agentless VPN tunnel that were running before the failover are stopped and have to be restarted. For example, file transfers that were in progress would have to be restarted. As well, any communication sessions with resources behind the FortiGate that are started by an Agentless VPN session have to be restarted.

To support Agentless VPN cookie failover, when an Agentless VPN session starts, the FGCP distributes the cookie created to identify the Agentless VPN session to all cluster units.

PPTP and L2TP VPN sessions

PPTP and L2TP VPNs are supported in HA mode. For a cluster you can configure PPTP and L2TP settings and you can also add security policies to allow PPTP and L2TP pass through. However, the FGCP does not provide session failover for PPTP or L2TP. After a failover, all active PPTP and L2TP sessions are lost and must be restarted.

Improving session sync performance

Two HA configuration options are available to reduce the performance impact of enabling session-pickup:

- Reducing the number of sessions that are synchronized.
- Using more FortiGate interfaces for session synchronization.

Reducing the number of sessions that are synchronized

When session pickup is enabled, new sessions are synced across cluster units. To reduce the number of synced sessions, enable the `session-pickup-delay` option, which only syncs sessions active for more than 30 seconds. This can reduce syncs for clusters with many short sessions, like HTTP traffic.

Use the following commands to enable a 30-second delay:

```
config system ha
    set session-pickup-delay enable
end
```

This may result in more sessions not resuming after a failover, but most short sessions can restart with minor interruption.

Using multiple FortiGate interfaces for session synchronization

The `session-sync-dev` option allows you to choose one or more FortiGate interfaces for session synchronization, which is necessary for session pickup. Typically, session synchronization takes place over the HA heartbeat link. However, with this HA option, only the chosen interfaces are used for session synchronization, not the HA heartbeat link. If multiple interfaces are selected, the session synchronization traffic is load balanced among the selected interfaces.

Shifting session synchronization away from the HA heartbeat interface can reduce the bandwidth needed for HA heartbeat traffic, potentially enhancing the cluster's efficiency and performance. This is particularly true if the cluster is synchronizing a large volume of sessions. Load balancing session synchronization across multiple interfaces can further boost performance and efficiency when dealing with a large number of sessions.

To perform cluster session synchronization using the port10 and port12 interfaces:

```
config system ha
    set session-sync-dev port10 port12
end
```

The interfaces chosen for session synchronization must be interconnected, either directly with the appropriate cable (if the cluster only contains two units) or through switches. If one of the interfaces becomes disconnected the cluster uses the remaining interfaces for session synchronization. If all of the session synchronization interfaces become disconnected, session synchronization reverts back to using the HA heartbeat link. All session synchronization traffic is between the primary unit and each subordinate unit.

Since large amounts of session synchronization traffic can increase network congestion, it is recommended to isolate this traffic from your network by using dedicated connections.

Resume IPS scanning of ICCP traffic after HA failover

After HA failover occurs, the IPS engine will resume processing ICCP sessions and keep the traffic going on the new primary unit. `session-pickup` must be enabled in an active-passive cluster to pick up the ICCP sessions.

The following example uses an active-passive cluster. See HA active-passive cluster setup on page 4079 for more information.

To configure HA:

```
config system ha
    set group-name "HA-APP"
    set mode a-p
    set password ************
    set hbdev "port3" 100
    set session-pickup enable
    set override enable
end
```

Session states before failover

When HA is working, the ICCP session information is stored in the HA session cache on the secondary FortiGate.

To verify the HA session cache on the secondary FortiGate:

```
# diagnose ips share list
 HA Session Cache
  client=10.1.100.178:57218 server=172.16.200.177:102
    service=39, ignore_app_after=0, last_app=76919, buffer_len=32
```

```
    stock tags: nr=981, hash=e68dc8120970448
    custom tags: nr=0, hash=1a49b996b6a42aa2
    tags [count=2]: s-737, s-828,
```

The ICCP session information can be found in the IPS session list and the session table on the primary FortiGate.

To verify the IPS session information on the primary FortiGate:

```
# diagnose ips session list
SESSION id:1 serial:35487 proto:6 group:6 age:134 idle:1 flag:0x800012a6
        feature:0x4 encap:0 ignore:0,0 ignore_after:204800,0
        tunnel:0 children:0 flag:..s.-....-....
  C-10.1.100.178:57218, S-172.16.200.177:102
  state: C-ESTABLISHED/13749/0/0/0/0, S-ESTABLISHED/48951/0/0/0/0 pause:0, paws:0
  expire: 3599
  app: unknown:0 last:44684 unknown-size:0
  cnfm: cotp
  set: cotp
  asm: cotp
```

To verify the system information on the primary FortiGate:

```
# diagnose sys session list
session info: proto=6 proto_state=11 duration=209 expire=3585 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=5
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty ndr npu syn_ses app_valid
statistic(bytes/packets/allow_err): org=11980/104/1 reply=57028/164/1 tuples=3
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=10->9/9->10 gwy=172.16.200.177/10.1.100.178
hook=post dir=org act=snat 10.1.100.178:57218->172.16.200.177:102(172.16.200.4:57218)
hook=pre dir=reply act=dnat 172.16.200.177:102->172.16.200.4:57218(10.1.100.178:57218)
hook=post dir=reply act=noop 172.16.200.177:102->10.1.100.178:57218(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=2 auth_info=0 chk_client_info=0 vd=1
serial=00008a9f tos=ff/ff app_list=2003 app=44684 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id=00000000 rpdb_svc_id=0 ngfwid=n/a
npu_state=0x003c94 ips_offload
npu info: flag=0x81/0x81, offload=8/8, ips_offload=1/1, epid=71/71, ipid=134/132,
vlan=0x0000/0x0000
vlifid=134/132, vtag_in=0x0000/0x0000 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=10/10
```

Sample log on current primary FortiGate:

```
# execute log display
304 logs found.
```

```
10 logs returned.
28.8% of logs has been searched.

1: date=2021-06-04 time=16:54:40 eventtime=1622850881110547135 tz="-0700" logid="1059028704"
type="utm" subtype="app-ctrl" eventtype="signature" level="information" vd="vd1" appid=44684
srcip=10.1.100.178 dstip=172.16.200.177 srcport=57218 dstport=102 srcintf="port2"
srcintfrole="undefined" dstintf="port1" dstintfrole="undefined" proto=6 service="tcp/102"
direction="incoming" policyid=2 sessionid=35487 applist="test" action="pass" appcat="Industrial"
app="ICCP_Transfer.Reporting" incidentserialno=61868187 msg="Industrial: ICCP_Transfer.Reporting,"
apprisk="elevated"
```

Session states after failover

After HA failover, the IPS engine on the new primary picks up the related ICCP sessions and continues passing the traffic. The HA session cache disappears on the new primary. The ICCP session now appears on the IPS session list and session table on the new primary.

To verify the IPS session information on the new primary FortiGate:

```
# diagnose ips session list
SESSION id:1 serial:35487 proto:6 group:6 age:90 idle:2 flag:0x820012a3
        feature:0x4 encap:0 ignore:1,0 ignore_after:204800,0
        tunnel:0 children:0 flag:....-....-..i.
  C-10.1.100.178:57218, S-172.16.200.177:102
  state: C-ESTABLISHED/9114/0/0/0/0, S-ESTABLISHED/0/0/0/0/0 pause:0, paws:0
  expire: 28
  app: unknown:0 last:44684 unknown-size:0
```

The server and client IPs, ports, and protocols remain the same.

To verify the system information on the primary FortiGate:

```
# diagnose sys session list
session info: proto=6 proto_state=11 duration=569 expire=3577 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=5
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty ndr npu syn_ses app_valid
statistic(bytes/packets/allow_err): org=38629/308/1 reply=160484/483/1 tuples=3
tx speed(Bps/kbps): 158/1 rx speed(Bps/kbps): 1139/9
orgin->sink: org pre->post, reply pre->post dev=10->9/9->10 gwy=172.16.200.177/10.1.100.178
hook=post dir=org act=snat 10.1.100.178:57218->172.16.200.177:102(172.16.200.4:57218)
hook=pre dir=reply act=dnat 172.16.200.177:102->172.16.200.4:57218(10.1.100.178:57218)
hook=post dir=reply act=noop 172.16.200.177:102->10.1.100.178:57218(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=2 auth_info=0 chk_client_info=0 vd=1
serial=00008a9f tos=ff/ff app_list=2003 app=44684 url_cat=0
sdwan_mbr_seq=0 sdwan_service_id=0
rpdb_link_id=00000000 rpdb_svc_id=0 ngfwid=n/a
```

```
npu_state=0x003c94 ips_offload
npu info: flag=0x81/0x81, offload=8/8, ips_offload=1/1, epid=71/71, ipid=134/132,
vlan=0x0000/0x0000
vlifid=134/132, vtag_in=0x0000/0x0000 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=10/10
```

The server and client IPs, ports, and NPU state remain the same.

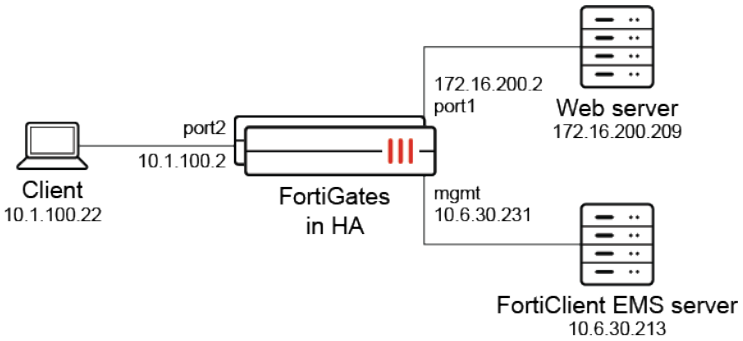
Sample log on new primary FortiGate:

```
# execute log display
653 logs found.
10 logs returned.
65.8% of logs has been searched.

1: date=2021-06-04 time=17:05:20 eventtime=1622851521364635480 tz="-0700" logid="1059028704"
type="utm" subtype="app-ctrl" eventtype="signature" level="information" vd="vd1" appid=44684
srcip=10.1.100.178 dstip=172.16.200.177 srcport=57218 dstport=102 srcintf="port2"
srcintfrole="undefined" dstintf="port1" dstintfrole="undefined" proto=6 service="tcp/102"
direction="incoming" policyid=2 sessionid=35487 applist="test" action="pass" appcat="Industrial"
app="ICCP_Transfer.Reporting" incidentserialno=198181218 msg="Industrial: ICCP_
Transfer.Reporting," apprisk="elevated"
```

Session synchronization during HA failover for ZTNA proxy sessions

User information and TLS sessions are synchronized between HA members for ZTNA proxy sessions. When a failover occurs, the new primary unit will continue allowing sessions from the logged in users without asking for the client certificate and re-authentication again.

In this example, a FortiGate HA pair is acting as a ZTNA access proxy. Clients that are trying to access the web server on qa.test.com are proxied by the ZTNA access proxy. Remote clients must be registered to the EMS server, and pass a client certificate check and user authentication in order to connect. Upon HA member failure, a failover occurs and the new primary unit will continue to allow connections without requesting client certificate check and user authentication for existing users and devices.



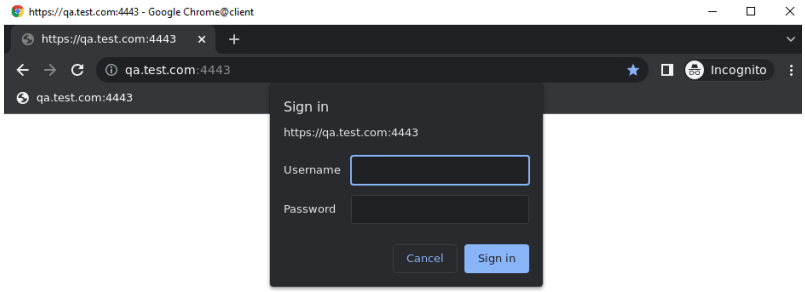
This example assumes ZTNA and EMS server settings are already configured.

To configure the HA settings:

```
config system ha
    set group-name "501E"
    set mode a-p
    set password **********
    set hbdev "ha" 0
    set session-pickup enable
    set override disable
    set monitor "port1" "port2"
end
```

To verify that the proxy sessions are synchronized between HA members:

1. On the client, access the web server. The ZTNA access proxy challenges the user for a client certificate and user authentication.



2. On the primary FortiGate, verify that the user information and TLS sessions are synchronized between HA members.
   a. Verify the list of proxy users:

```
501E-primary # diagnose wad user list
ID: 1, VDOM: root, IPv4: 10.1.100.22
  user name    : localuser1
  worker       : 0
  duration     : 8
  auth_type    : IP
  auth_method  : Basic
  pol_id       : 1
  g_id         : 0
  user_based   : 1
  expire       : 597
  LAN:
    bytes_in=2093 bytes_out=5753
  WAN:
    bytes_in=2024 bytes_out=1235
```

   b. Apply a filter to WAD debug to diagnose the wad informer process:

```
501E-primary # diagnose test application wad 2400
Set diagnosis process: type=informer index=0 pid=305
```

   c. Show the user cache from the WAD informer. Verify that the localuser1 entry exists:

```
501E-primary # diagnose test application wad 110
users:
[1]     localuser1@10.1.100.22:0 upn_domain= from:worker worker:6 vf:0 ref:1 stale=0
ntlm:0, has_fsae:0, guest:0
                user_node:(0x7fe18dcf0048) user:1[max=65536](0x7fe18dd08048) ip:1
(0x7fe18dd00048) scheme:0 outofsync:0(0) id:1
...
```

d. Verify using WAD real-time debugs on the secondary FortiGate. The user information is synchronized to the secondary FortiGate:

```
501E-secondary # diagnose wad debug enable category all
501E-secondary # diagnose wad debug enable level verbose
501E-secondary # diagnose debug enable
[I][p:296]              wad_proc_informer_ha_dgram_on_read:2811  Got HA msg: type=0,
sizeof(msg)=8, dlen=80, sz=88
[I][p:296]              wad_proc_informer_on_ha_user_add  :1493  reader:
ip=10.1.100.22:45852 vf=0 seq=0 grp_type=0 scheme=0 is_ntlm=0 has_fsae=0 concur_user=65536
domain=''
[I][p:296]              wad_informer_update_user_ext      :782   ip=10.1.100.22:45852
name=localuser1 from=worker
[I][p:296]              wad_informer_find_user_ip_entries :621   find=false(1) vf=0
ip=10.1.100.22:45852 pr=(nil)
mapping user_node:0x7fc1c84dd048, user_ip:0x7fc1c84ed048(0), user:0x7fc1c84f5048(0).
```

3. Verify the user cache from the WAD informer:

```
501E-secondary # diagnose test application wad 110
users:
[1]     localuser1@10.1.100.22:0 upn_domain= from:worker worker:-126 vf:0 ref:1 stale=0
ntlm:0, has_fsae:0, guest:0
               user_node:(0x7fa3eb07d048) user:1[max=65536](0x7fa3eb095048) ip:1
(0x7fa3eb08d048) scheme:0 outofsync:0(0) id:7
...
```

If the client tries to access the web server again after failover occurs, the client certificate check and authentication prompt does not appear. ZTNA allows the traffic to pass.

The ZTNA logs for both FortiGates contain the same user information.

Primary FortiGate log:

```
1: date=2022-03-23 time=11:49:57 eventtime=1648061397548444970 tz="-0700" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.1.100.22 srcname="client"
srcport=45826 srcintf="port2" srcintfrole="lan" dstcountry="Reserved" srccountry="Reserved"
dstip=172.16.200.209 dstport=443 dstintf="port1" dstintfrole="lan" sessionid=4786 service="HTTPS"
proto=6 action="accept" policyid=1 policytype="proxy-policy" poluuid="ea7a8a04-a56e-51ec-9d7b-
90d24b3a28e9" policyname="ztna" duration=5 user="localuser1" gatewayid=1 vip="ztna"
accessproxy="ztna" clientdeviceid="EF73C831C3FE4FF195A5B2030B******"
clientdevicetags="FCTEMS8821000000_all_registered_clients/MAC_FCTEMS8821000000_all_registered_
clients/FCTEMS8821000000_ZT_FILE_CERTFILE" wanin=2024 rcvdbyte=2024 wanout=1325 lanin=1511
sentbyte=1511 lanout=1075 fctuid="EF73C831C3FE4FF195A5B2030B******" unauthuser="fosqa"
unauthusersource="forticlient" appcat="unscanned"
```

Secondary FortiGate log:

```
1: date=2022-03-23 time=11:55:01 eventtime=1648061701628425041 tz="-0700" logid="0005000024"
type="traffic" subtype="ztna" level="notice" vd="root" srcip=10.1.100.22 srcname="client"
srcport=45830 srcintf="port2" srcintfrole="lan" dstcountry="Reserved" srccountry="Reserved"
dstip=172.16.200.209 dstport=443 dstintf="port1" dstintfrole="lan" sessionid=676 service="HTTPS"
proto=6 action="accept" policyid=1 policytype="proxy-policy" poluuid="ea7a8a04-a56e-51ec-9d7b-
90d24b3a28e9" policyname="ztna" duration=5 user="localuser1" gatewayid=1 vip="ztna"
accessproxy="ztna" clientdeviceid="EF73C831C3FE4FF195A5B2030B******"
clientdevicetags="FCTEMS8821000000_all_registered_clients/MAC_FCTEMS8821000000_all_registered_
clients/FCTEMS8821000000_ZT_FILE_CERTFILE" wanin=2024 rcvdbyte=2024 wanout=1325 lanin=1511
sentbyte=1511 lanout=1075 fctuid="EF73C831C3FE4FF195A5B2030B******" unauthuser="fosqa"
unauthusersource="forticlient" appcat="unscanned"
```

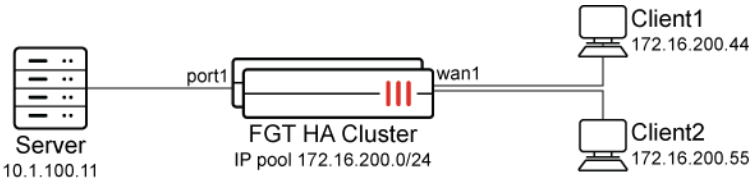
FGCP and FGSP synchronizes full-cone expectation sessions when session-pickup-expectation enabled

FGCP or FGSP HA configured to synchronize expectation sessions, now also synchronizes full cone NAT UDP expectation sessions. For information about full cone NAT, see Full cone NAT for fixed port range IP pools on page 2028.

HA configuration to synchronize UDP and expectation sessions:

```
config system ha
    set session-pickup enable
    set session-pickup-connectionless enable
    set session-pickup-expectation enable
end
```

Example configuration



Example HA configuration with `session-pickup`, `session-pickup-connectionless`, and `session-pickup-expectation` enabled.

```
config system ha
    set group-id 240
    set group-name FG_HA
    set mode a-p
    set password <password>
    set hbdev ha1 50 ha2 100
    set session-pickup enable
    set session-pickup-connectionless enable
    set session-pickup-expectation enable
```

```
    set override enable
    set priority 255
end
```

Example Full cone NAT IP pool:

```
config firewall ippool
    edit full_cone_nat
        set type fixed-port-range
        set startip 172.16.200.155
        set endip 172.16.200.165
        set startport 5000
        set endport 5087
        set source-startip 10.1.100.10
        set source-endip 10.1.100.23
        set port-per-user 32
        set permit-any-host enable
    end
```

Example full cone NAT Firewall policy:

```
config firewall policy
    edit 10
        set name FW
        set uuid a2efc630-015a-51f1-a27b-aa72f6406517
        set srcintf port1
        set dstintf wan1
        set action accept
        set srcaddr all
        set dstaddr all
        set schedule always
        set service ALL
        set auto-asic-offload disable
        set nat enable
        set ippool enable
        set poolname full_cone_nat
    end
```

In the example, the server connects to Client2 and this opens up full cone NAT ports on the FortiGate cluster wan1 interface that allow CLient1 and Client2 to connect through the FortiGate to the server.

When the server connects to Client2, the following session is created on the primary FortiGate in the HA cluster:

```
diagnose sys session list

session info: proto=17 proto_state=00 duration=218 expire=179 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=5
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty synced f16
```

```
statistic(bytes/packets/allow_err): org=29145088/19456/1 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 133569/1068 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=11->7/7->11 gwy=172.16.200.55/0.0.0.0
hook=post dir=org act=snat 10.1.100.11:5050->172.16.200.55:5001(172.16.200.155:5050)
hook=pre dir=reply act=dnat 172.16.200.55:5001->172.16.200.155:5050(10.1.100.11:5050)
misc=0 policy_id=10 pol_uuid_idx=667 auth_info=0 chk_client_info=0 vd=0
serial=000018fd tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
hrx info: valid=1/0, qid=7/0, npuid=1/0, sublink=0/0
```

Because of full cone NAT, the following expectation sessions are also created on the primary FortiGate:

```
diagnose sys session list expectation

session info: proto=17 proto_state=00 duration=251 expire=-221 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=2
origin-shaper=
reply-shaper=
per_ip_shaper=
master session: proto=17 10.1.100.11:5050->172.16.200.55:5001
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=255/255
state=new f31
statistic(bytes/packets/allow_err): org=0/0/0 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=11->7/7->11 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=dnat 10.1.100.11:0->172.16.200.155:5050(10.1.100.11:0)
hook=pre dir=org act=noop 0.0.0.0:0->0.0.0.0:0(0.0.0.0:0)
misc=0 policy_id=10 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=0
serial=000018fd tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
npu_state=0x4000000
no_ofld_reason:  new
hrx info: valid=0/0, qid=0/0, npuid=0/0, sublink=0/0

session info: proto=17 proto_state=00 duration=251 expire=-221 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=2
origin-shaper=
reply-shaper=
per_ip_shaper=
master session: proto=17 10.1.100.11:5050->172.16.200.55:5001
class_id=0 ha_id=0 policy_dir=1 tunnel=/ vlan_cos=255/255
state=new f31
statistic(bytes/packets/allow_err): org=0/0/0 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=0->11/11->0 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=dnat 0.0.0.0:0->172.16.200.155:5050(10.1.100.11:5050)
hook=pre dir=org act=noop 0.0.0.0:0->0.0.0.0:0(0.0.0.0:0)
misc=0 policy_id=10 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=0
serial=000018fd tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
npu_state=0x4000000
```

```
no_ofld_reason:  new
hrx info: valid=0/0, qid=0/0, npuid=0/0, sublink=0/0
```

HA synchronizes the following session to the backup or secondary FortiGate:

```
diagnose sys session list

session info: proto=17 proto_state=00 duration=338 expire=111 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=5
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=dirty may_dirty syn_ses
statistic(bytes/packets/allow_err): org=0/0/0 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=11->7/7->11 gwy=0.0.0.0/0.0.0.0
hook=post dir=org act=snat 10.1.100.11:5050->172.16.200.55:5001(172.16.200.155:5050)
hook=pre dir=reply act=dnat 172.16.200.55:5001->172.16.200.155:5050(10.1.100.11:5050)
misc=0 policy_id=10 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=0
serial=000018fd tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
npu_state=0x4000000
no_ofld_reason:  npu-flag-off
hrx info: valid=0/0, qid=0/0, npuid=0/0, sublink=0/0
```

As well, HA synchronizes the following expectation sessions to the backup or secondary FortiGate:

```
diagnose sys session list expectation

session info: proto=17 proto_state=00 duration=107 expire=-77 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=2
origin-shaper=
reply-shaper=
per_ip_shaper=
master session: proto=17 10.1.100.11:5050->172.16.200.55:5001
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=255/255
state=new synced f31
statistic(bytes/packets/allow_err): org=0/0/0 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=11->7/7->11 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=dnat 10.1.100.11:0->172.16.200.155:5050(10.1.100.11:0)
hook=pre dir=org act=noop 0.0.0.0:0->0.0.0.0:0(0.0.0.0:0)
misc=0 policy_id=10 pol_uuid_idx=665 auth_info=0 chk_client_info=0 vd=0
serial=000018fd tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
npu_state=0x000001 no_offload
no_ofld_reason:  new disabled-by-policy
hrx info: valid=0/0, qid=0/0, npuid=0/0, sublink=0/0

session info: proto=17 proto_state=00 duration=107 expire=-77 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=2
origin-shaper=
```

```
reply-shaper=
per_ip_shaper=
master session: proto=17 10.1.100.11:5050->172.16.200.55:5001
class_id=0 ha_id=0 policy_dir=1 tunnel=/ vlan_cos=255/255
state=new synced f31
statistic(bytes/packets/allow_err): org=0/0/0 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=0->11/11->0 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=dnat 0.0.0.0:0->172.16.200.155:5050(10.1.100.11:5050)
hook=pre dir=org act=noop 0.0.0.0:0->0.0.0.0:0(0.0.0.0:0)
misc=0 policy_id=10 pol_uuid_idx=665 auth_info=0 chk_client_info=0 vd=0
serial=000018fd tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
npu_state=0x000001 no_offload
no_ofld_reason:  new disabled-by-policy
hrx info: valid=0/0, qid=0/0, npuid=0/0, sublink=0/0
```

Full cone NAT also allows Client1 to connect to the server through the FortiGate cluster. After Client1 connects, the primary FortiGate includes the following sessions:

```
diagnose sys session list

session info: proto=17 proto_state=00 duration=454 expire=1724 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=4
origin-shaper=
reply-shaper=
per_ip_shaper=
master session: proto=17 10.1.100.11:5050->172.16.200.55:5001
class_id=0 ha_id=0 policy_dir=1 tunnel=/ vlan_cos=255/255
state=synced intree
statistic(bytes/packets/allow_err): org=22717170/15165/1 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 55604/444 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=7->11/11->7 gwy=10.1.100.11/172.16.200.44
hook=pre dir=org act=dnat 172.16.200.44:5020->172.16.200.155:5050(10.1.100.11:5050)
hook=post dir=reply act=snat 10.1.100.11:5050->172.16.200.44:5020(172.16.200.155:5050)
misc=0 policy_id=10 pol_uuid_idx=665 auth_info=0 chk_client_info=0 vd=0
serial=000018fd tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
hrx info: valid=1/1, qid=4/6, npuid=1/1, sublink=0/0

session info: proto=17 proto_state=00 duration=454 expire=104 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=6
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=dirty may_dirty synced f16
statistic(bytes/packets/allow_err): org=263648/176/1 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=11->7/7->11 gwy=172.16.200.55/0.0.0.0
hook=post dir=org act=snat 10.1.100.11:5050->172.16.200.55:5001(172.16.200.155:5050)
```

```
hook=pre dir=reply act=dnat 172.16.200.55:5001->172.16.200.155:5050(10.1.100.11:5050)
misc=0 policy_id=10 pol_uuid_idx=665 auth_info=0 chk_client_info=0 vd=0
serial=000018fd tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
npu_state=0x000001 no_offload
no_ofld_reason:  disabled-by-policy
hrx info: valid=1/0, qid=7/0, npuid=1/0, sublink=0/0
total session: 2
```

And these sessions are synchronized to the backup or secondary FortiGate:

```
diag sys session list


session info: proto=17 proto_state=00 duration=443 expire=1463 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=4
origin-shaper=
reply-shaper=
per_ip_shaper=
master session: proto=17 10.1.100.11:5050->172.16.200.55:5001
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=255/255
state=syn_ses intree
statistic(bytes/packets/allow_err): org=0/0/0 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=7->11/11->7 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=dnat 172.16.200.44:5020->172.16.200.155:5050(10.1.100.11:5050)
hook=post dir=reply act=snat 10.1.100.11:5050->172.16.200.44:5020(172.16.200.155:5050)
misc=0 policy_id=10 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=0
serial=000018fd tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
npu_state=0x4000000
no_ofld_reason:  npu-flag-off
hrx info: valid=0/0, qid=0/0, npuid=0/0, sublink=0/0

session info: proto=17 proto_state=00 duration=443 expire=6 timeout=0 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=5
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=dirty may_dirty syn_ses
statistic(bytes/packets/allow_err): org=0/0/0 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=11->7/7->11 gwy=0.0.0.0/0.0.0.0
hook=post dir=org act=snat 10.1.100.11:5050->172.16.200.55:5001(172.16.200.155:5050)
hook=pre dir=reply act=dnat 172.16.200.55:5001->172.16.200.155:5050(10.1.100.11:5050)
misc=0 policy_id=10 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=0
serial=000018fd tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
npu_state=0x4000000
no_ofld_reason:  npu-flag-off
hrx info: valid=0/0, qid=0/0, npuid=0/0, sublink=0/0
total session: 2
```

Because these sessions are synchronized, after an HA failover, Client1 and Client2 can continue communicating with the server.

Management access

Effective management of an HA cluster requires establishing reliable administrative access to both the primary and secondary units. Depending on your network topology and security requirements, you can configure management access using shared data interfaces or dedicated administrative ports.

This section covers the following methods for configuring and optimizing cluster management:

Topic	Summary
Out-of-band management with reserved management interfaces	This method involves reserving a dedicated physical interface exclusively for management traffic. This allows for direct access to individual cluster units (including the secondary unit) and segregates administrative traffic, such as SNMP, logging, and remote authentication, from production data traffic.
In-band management	This method utilizes existing network interfaces for administrative access. It does not require a dedicated physical port, allowing management traffic to flow through the same subnets as user traffic. This section details how to configure management IPs on cluster interfaces.
Routing NetFlow data over the HA management interface	This advanced configuration utilizes the reserved management interface to offload bandwidth-intensive telemetry data. This ensures that NetFlow traffic does not consume bandwidth on production data ports.

Out-of-band management with reserved management interfaces

As part of an HA configuration, you can reserve up to four management interfaces to provide direct management access to all cluster units. For each reserved management interface, you can configure a different IP address, administrative access, and other interface settings, for each cluster unit. By connecting these interfaces to your network, you can separately manage each cluster unit from different IP addresses.

- Reserved management interfaces provide direct management access to each cluster unit, and give each cluster unit a different identity on your network. This simplifies using external services, such as SNMP, to monitor separate cluster units.
- Reserved management interfaces are not assigned HA virtual MAC addresses. They retain the permanent hardware address of the physical interface, unless you manually change it using the `config system interface` command.
- Reserved management interfaces and their IP addresses should not be used for managing a cluster using FortiManager. To manage a FortiGate HA cluster with FortiManager, use the IP address of one of the cluster unit interfaces.
- Configuration changes to a reserved management interface are not synchronized to other cluster units. Other configuration changes are automatically synchronized to all cluster units.

You can configure an in-band management interface for a cluster unit. See In-band management on page 4138 for information. In-band management does not reserve the interface exclusively for HA management.

Management interface

Enable HTTPS or HTTP administrative access on the reserved management interfaces to connect to the GUI of each cluster unit. On secondary units, the GUI has the same features as the primary unit, except for unit specific information, for example:

- The System Information widget on the Status dashboard shows the secondary unit's serial number.
- In the cluster members list at *System > HA*, you can change the HA configuration of the unit that you are logged into. You can only change the host name and device priority of the primary and other secondary units.
- The system events logs show logs for the device that you are logged into. Use the HA device drop down to view the log messages for other cluster units, including the primary unit.

Enable SSH administrative access on the reserved management interfaces to connect to the CLI of each cluster unit. The CLI prompt includes the host of the cluster unit that you are connected to. Use the `execute ha manage` command to connect to other cluster unit CLIs.

Enable SNMP administrative access on a reserved management interface to use SNMP to monitor each cluster unit using the interface's IP address. Direct management of cluster members must also be enabled, see Configuration examples on page 4133.

Reserved management interfaces are available in both NAT and transparent mode, and when the cluster is operating with multiple VDOMs.

FortiCloud, FortiSandbox, and other management services

By default, management services such as FortiCloud, FortiSandbox, SNMP, remote logging, and remote authentication, use a cluster interface. This means that communication from each cluster unit will come from a cluster interface of the primary unit, and not from the individual cluster unit's interface.

You can configure HA reserved management interfaces to be used for communication with management services by enabling the `ha-direct` option. This separates management traffic for each cluster unit, and allows each unit to be individually managed. This is especially useful when cluster units are in different physical locations.

The following management features will then use the HA reserved management interface:

- SSH, HTTP, HTTPS administration
- Remote logging, including syslog, FortiAnalyzer, and FortiCloud
- SNMP queries and traps
- Remote authentication and certificate verification using LDAP, RADIUS, and TACACS+

> Remote authentication and certificate verification using LDAP, RADIUS, and TACACS+ follow specific routing behaviors based on configuration. The `interface-selection` option (specify/sdwan) takes the highest priority and overrides `ha-direct`. Meanwhile, setting a `source-ip` disables `ha-direct`, but `source-ip` and `interface-selection` operate independently without overriding or interacting with each other. On an HA secondary unit, if either `source-ip` or `interface-selection` is configured, traffic is routed through the HA primary unit, and `ha-direct` is ignored.

- Communication with FortiSandbox
- NetFlow and sflow, see Routing NetFlow data over the HA management interface on page 4139 for information.

Any other management function not explicitly listed above is not supported, such as Security Fabric connectivity and new device registration.

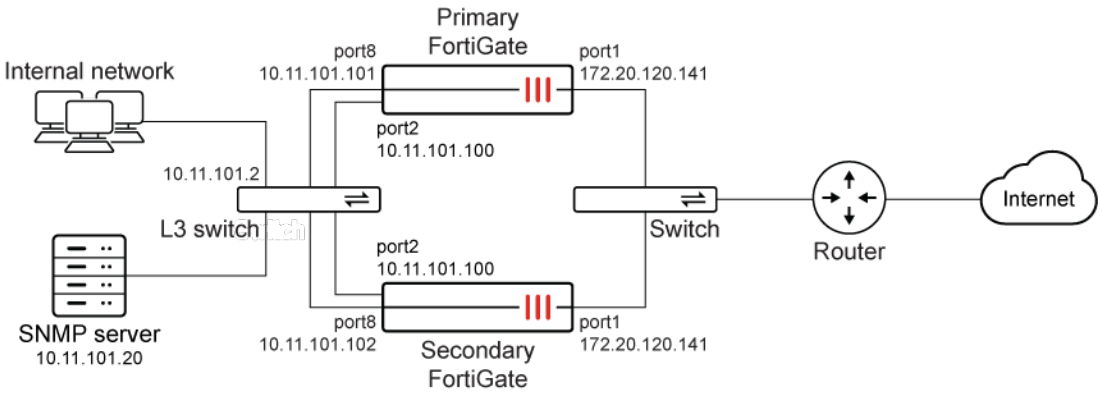
Syntax for HA reserved management interfaces is as follows:

```
config system ha
    set ha-mgmt-status enable
    config ha-mgmt-interfaces
        edit 1
            set interface <interface>
            set dst <destination IP>
            set gateway <IPv4 gateway>
            set gateway6 <IPv6 gateway>
        next
    end
    set ha-direct enable
end
```

> The `ha-direct` option is a pre-requisite for allowing communication on each HA reserved management interface for various management services listed above. Once enabled, all `source-ip` settings will be unset from log related, netflow and sflow management services.
>
> SNMP requires `ha-direct` to be configured under SNMP settings only. See below for more configuration options.

Configuration examples

The configuration examples below will use the following topology:



Two FortiGate units are already operating in a cluster. On each unit, port8 is connected to the internal network through a switch and configured as an out-of-band reserved management interface.

Configuration changes to the reserved management interface are not synchronized to other cluster units.

Administrative access and default route for HA management interface

To configure the primary unit's reserved management interface, configure an IP address and management access on port8. Then, configure the necessary HA settings to enable the HA reserved management interface and its route. To configure the secondary unit's reserved management interface, access the unit's CLI through the primary unit, and configure an IP address, management access on port8, and the necessary HA settings. Configuration changes to the reserved management interface are not synchronized to other cluster units.

To configure the primary unit reserved management interface to allow HTTPS, SSH, and ICMP access:

1. From a computer on the internal network, connect to the CLI at 10.11.101.100 on port2.
2. Change the port8 IP address and management access:

```
config system interface
    edit port8
        set ip 10.11.101.101/24
        set allowaccess https ping ssh
    next
end
```

3. Configure the HA settings for the HA reserved management interface by defining a default route to route to the gateway 10.11.101.2:

```
config system ha
    set ha-mgmt-status enable
    config ha-mgmt-interfaces
        edit 1
            set interface port8
            set gateway 10.11.101.2
        next
```

```
    end
end
```

You can now log into the primary unit's GUI by browsing to https://10.11.101.101. You can also log into the primary unit's CLI by using an SSH client to connect to 10.11.101.101.

To configure secondary unit reserved management interfaces to allow HTTPS, SSH, and ICMP access:

1. From a computer on the internal network, connect to the primary unit's CLI.
2. Connect to the secondary unit with the following command:

```
execute ha manage <unit id> <username> <password>
```

3. Change the port8 IP address and management access:

```
config system interface
    edit port8
        set ip 10.11.101.102/24
        set allowaccess https ping ssh
    next
end
```

```
exit
```

4. Configure the HA settings for the HA reserved management interface by defining a default route to route to the gateway 10.11.101.2:

```
config system ha
    set ha-mgmt-status enable
    config ha-mgmt-interfaces
        edit 1
            set interface port8
            set gateway 10.11.101.2
        next
    end
end
```

You can now log into the secondary unit's GUI by browsing to https://10.11.101.102. You can also log into the secondary unit's CLI by using an SSH client to connect to 10.11.101.102.

SNMP monitoring

The SNMP server can get status information from the cluster members. To use the reserved management interfaces, you must add at least one HA direct management host to an SNMP community. If the SNMP configuration includes SNMP users with user names and passwords, HA direct management must be enabled for the users.

To configure the cluster for SNMP management using the reserved management interfaces in the CLI:

1. Allow SNMP on port8 on both primary and secondary units:

```
config system interface
    edit port8
        append allowaccess snmp
    next
end
```

2. Add an SNMP community with a host for the reserved management interface of each cluster member. The host includes the IP address of the SNMP server.

```
config system snmp community
    edit 1
        set name "Community"
        config hosts
            edit 1
                set ip 10.11.101.20 255.255.255.255
                set ha-direct enable
            next
        end
    next
end
```

Enabling `ha-direct` in a non-HA environment will make SNMP unusable.

3. Add an SNMP user for the reserved management interface:

```
config system snmp user
    edit "1"
        set notify-hosts 10.11.101.20
        set ha-direct enable
    next
end
```

The SNMP configuration is synchronized to all cluster units.

To get CPU, memory, and network usage information from the SNMP manager for each cluster unit using the reserved management IP addresses:

1. Connect to the SNMP manager CLI.
2. Get resource usage information for the primary unit using the MIB fields:

```
snmpget -v2c -c Community 10.11.101.101 fgHaStatsCpuUsage
snmpget -v2c -c Community 10.11.101.101 fgHaStatsMemUsage
snmpget -v2c -c Community 10.11.101.101 fgHaStatsNetUsage
```

3. Get resource usage information for the primary unit using the OIDs:

```
snmpget -v2c -c Community 10.11.101.101 1.3.6.1.4.1.12356.101.13.2.1.1.3.1
snmpget -v2c -c Community 10.11.101.101 1.3.6.1.4.1.12356.101.13.2.1.1.4.1
snmpget -v2c -c Community 10.11.101.101 1.3.6.1.4.1.12356.101.13.2.1.1.5.1
```

4. Get resource usage information for the secondary unit using the MIB fields:

```
snmpget -v2c -c Community 10.11.101.102 fgHaStatsCpuUsage
snmpget -v2c -c Community 10.11.101.102 fgHaStatsMemUsage
snmpget -v2c -c Community 10.11.101.102 fgHaStatsNetUsage
```

5. Get resource usage information for the primary unit using the OIDs:

```
snmpget -v2c -c Community 10.11.101.102 1.3.6.1.4.1.12356.101.13.2.1.1.3.1
snmpget -v2c -c Community 10.11.101.102 1.3.6.1.4.1.12356.101.13.2.1.1.4.1
snmpget -v2c -c Community 10.11.101.102 1.3.6.1.4.1.12356.101.13.2.1.1.5.1
```

Firewall local-in policies for the reserved management interface

Enabling `ha-mgmt-intf-only` applies the local-in policy only to the VDOM that contains the reserved management interface. The incoming interface is set to match any interface in the VDOM.

To add local-in policies for the reserved management interface:

```
config firewall local-in-policy
    edit 0
        set ha-mgmt-intf-only enable
        set intf any
        set srcaddr internal-net
        set dstaddr mgmt-int
        set action accept
        set service HTTPS
        set schedule weekdays
    next
end
```

NTP over reserved management interfaces

When NTP is enabled in an HA cluster, the primary unit will always be the unit to contact the NTP server and synchronize system time to the secondary units over the HA heartbeat interface. However, in the event that the primary should contact the NTP server over the HA reserved management interface, then the `ha-direct` option should be enabled under the `config system ha` settings.

```
config system interface
    edit port5
        set ip 172.16.79.46 255.255.255.0
    next
end
```

```
config system ha
    set group-name FGT-HA
```

```
    set mode a-p
    set password **********
    set ha-mgmt-status enable
    config ha-mgmt-interfaces
        edit 1
            set interface port5
            set gateway 172.16.79.1
        next
    end
    set ha-direct enable
end
```

```
config system ntp
    set ntpsync enable
    set syncinterval 5
end
```

In-band management

In-band management IP addresses are an alternative to reserved HA management interfaces, and do not require reserving an interface exclusively for management access. They can be added to multiple interfaces on each cluster unit.

The in-band management IP address is accessible from the network that the cluster interface is connected to. It should be in the same subnet as the interface that you are adding it to. It cannot be in the same subnet as other interface IP addresses.

In-band management interfaces support ping, HTTP, HTTPS, and SNMP administrative access options.

Primary and secondary units can respond on the management IP to traffic from different networks by using the routing table. The secondary unit uses the kernel routing table synchronized from the primary to route the traffic.

In-band management IP address configuration is not synchronized to other cluster units.

To add an in-band management IP address to port23 with HTTPS, SSH, and SNMP access:

```
config system interface
    edit port23
        set ip 172.25.12.1/24
        set management-ip 172.25.12.5/24
        set allowaccess https ssh snmp
    next
end
```

Routing NetFlow data over the HA management interface

In an HA environment, the `ha-direct` option allows data from services such as syslog, FortiAnalyzer, SNMP, and NetFlow to be routed over the outgoing interface.

The following example shows how NetFlow data can be routed over the HA management interface mgmt1.

To route NetFlow data over the HA management interface:

1. On the primary unit (FortiGate A), configure the HA and mgmt1 interface settings:

```
(global) # config system ha
    set group-name "test-ha"
    set mode a-p
    set password *********
    set hbdev "port6" 50
    set hb-interval 4
    set hb-lost-threshold 10
    set session-pickup enable
    set ha-mgmt-status enable
    config ha-mgmt-interfaces
        edit 1
            set interface "mgmt1"
        next
    end
    set override enable
    set priority 200
    set ha-direct enable
end
```

```
(global) # config system interface
    edit "mgmt1"
        set ip 10.6.30.111 255.255.255.0
        set allowaccess ping https ssh http telnet fgfm
        set type physical
        set dedicated-to management
        set role lan
        set snmp-index 1
    next
end
```

2. On the secondary unit (FortiGate B), configure the HA and mgmt1 interface settings:

```
(global) # config system ha
    set group-name "test-ha"
    set mode a-p
    set password *********
    set hbdev "port6" 50
    set hb-interval 4
    set hb-lost-threshold 10
    set session-pickup enable
    set ha-mgmt-status enable
```

```
    config ha-mgmt-interfaces
        edit 1
            set interface "mgmt1"
        next
    end
    set override enable
    set priority 100
    set ha-direct enable
end
```

```
(global) # config system interface
    edit "mgmt1"
        set ip 10.6.30.112 255.255.255.0
        set allowaccess ping https ssh http telnet fgfm
        set type physical
        set dedicated-to management
        set role lan
        set snmp-index 1
    next
end
```

3. On the primary unit (FortiGate A), configure the NetFlow setting:

```
(global) # config system netflow
    set collector-ip 10.6.30.59
end
```

4. Verify that NetFlow uses the mgmt1 IP:

```
(global) # diagnose test application sflowd 3
```

5. Verify that the NetFlow packets are being sent by the mgmt1 IP:

```
(vdom1) # diagnose sniffer packet any 'udp and port 2055' 4
interfaces=[any]
filters=[udp and port 2055]
8.397265 mgmt1 out 10.6.30.111.1992 -> 10.6.30.59.2055: udp 60
23.392175 mgmt1 out 10.6.30.111.1992 -> 10.6.30.59.2055: udp 188
23.392189 mgmt1 out 10.6.30.111.1992 -> 10.6.30.59.2055: udp 60
...
3 packets received by filter
0 packets dropped by kernel
```

6. On the secondary device (FortiGate B), change the priority so that it becomes the primary:

```
(global) # config system ha
    set priority 250
end
```

7. Verify the NetFlow status on FortiGate A, which is using the new primary's mgmt1 IP:

```
(global) # diagnose test application sflowd 3
```

8. Verify that the NetFlow packets use the new source IP on FortiGate B:

```
(vdom1) # diagnose sniffer packet any 'udp and port 2055' 4
interfaces=[any]
filters=[udp and port 2055]
7.579574 mgmt1 out 10.6.30.112.3579 -> 10.6.30.59.2055: udp 60
22.581830 mgmt1 out 10.6.30.112.3579 -> 10.6.30.59.2055: udp 60
29.038336 mgmt1 out 10.6.30.112.3579 -> 10.6.30.59.2055: udp 1140
^C
3 packets received by filter
0 packets dropped by kernel
```

Maintenance

Maintaining firmware levels in a High Availability (HA) cluster is critical for security and stability. FortiOS provides flexible options for upgrading clusters, ranging from fully automated, zero-downtime procedures to manual, staged upgrades that allow for granular verification.

This section covers the two primary methods for HA firmware maintenance:

Topic	Summary
Upgrading FortiGates in an HA cluster	The standard approach using default or simultaneous upgrade modes.
FGCP multi-version cluster upgrade	An advanced method that places the cluster into a temporary mixed-version state for testing and verification.

Upgrading FortiGates in an HA cluster

You can upgrade the firmware on an HA cluster in the same way as on a standalone FortiGate. During a firmware upgrade, the cluster upgrades the primary unit and all of the subordinate units to the new firmware image.

⚠ Before upgrading a cluster, back up your configuration (Configuration backups and reset on page 4053), schedule a maintenance window, and make sure that you are using a supported upgrade path (https://docs.fortinet.com/upgrade-tool).

Uninterrupted upgrade

An uninterrupted upgrade occurs without interrupting communication in the physical or virtual cluster.

To upgrade the cluster firmware without interrupting communication, use the following steps. These steps are transparent to the user and the network, and might result in the cluster selecting a new primary unit.

1. The administrator uploads a new firmware image using the GUI or CLI. See Upgrading individual devices on page 3749 for details.
2. The firmware is upgraded on all of the subordinate units.
3. A new primary unit is selected from the upgraded subordinates.

4. The firmware is upgraded on the former primary unit.
5. Primary unit selection occurs, according to the standard primary unit selection process.

If all of the subordinate units crash or otherwise stop responding during the upgrade process, the primary unit will continue to operate normally, and will not be upgraded until at least one subordinate rejoins the cluster.

> ⚠ Uninterrupted upgrade does not guarantee that reboots will stagger after changes to CLI settings that require a reboot. Changing settings in the CLI that require a reboot will typically display a warning, such as:
>
> ```
> The configuration will take effect after system reboot.
> Do you want to continue? (y/n)
> ```
>
> This can result in all of the cluster units rebooting at the same time.

Interrupted upgrade

An interrupted upgrade upgrades all cluster members at the same time. This takes less time than an uninterrupted upgrade, but it interrupts communication in the cluster. Interrupted upgrade is disabled by default.

To enable interrupted upgrade:

```
config system ha
    set upgrade-mode simultaneous
end
```

FGCP multi-version cluster upgrade

The FGCP multi-version cluster (MVC) upgrade mode allows manual control over the cluster member that is being upgraded. HA members can temporarily run in an MVC while administrators perform tests to confirm traffic can pass through the upgraded member smoothly.

The syntax of the existing upgrade mode has been updated starting in 7.4.1:

```
config system ha
    set upgrade-mode {simultaneous | uninterruptible | local-only | secondary-only}
end
```

`upgrade-mode {simultaneous | uninterruptible | local-only | secondary-only}`	Set the mode to upgrade a cluster. • `simultaneous`: all HA members upgrade at the same time (`set uninterruptible-upgrade disable` in 7.4.0 and earlier). • `uninterruptible`: secondary HA members are upgraded first, followed by the primary member (`set uninterruptible-upgrade enable` in 7.4.0 and earlier). • `local-only`: only upgrade the local member in which the firmware is uploaded. • `secondary-only`: only upgrade the secondary members.

The `local-only` and `secondary-only` upgrade options are advanced configurations that should only be used to temporarily put the HA cluster in MVC operation mode. While in this operation, states and sessions (such as the session table and routing table) are synchronized, but configuration changes are not synchronized between cluster members in different builds. If more than two members are in the cluster, the configurations between members in the same builds will be synchronized. The configurations for the entire cluster will be synchronized once the upgrade process has completed.

How it works

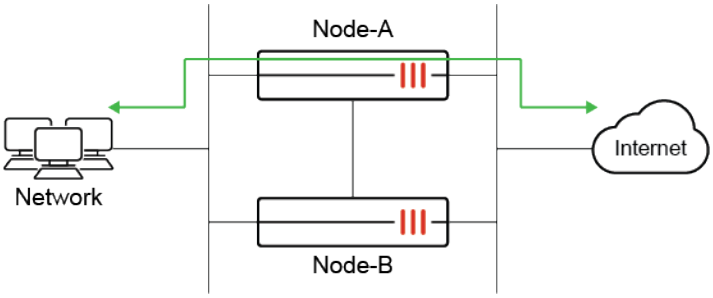
In `local-only` and `secondary-only` modes, the specific cluster member is upgraded and sessions are synchronized to it. The following tables show which members are upgraded based on the mode and where the upgrade is initiated.

local-only		
Upgrade method	**Outcome**	**Recommendation**
Initiate the upload or upgrade on the primary.	The primary member is upgraded.	Not recommended.
Initiate the upload or upgrade on the secondary member.	The secondary member where the image is uploaded is upgraded.	Recommended when selecting a specific HA member to upgrade.

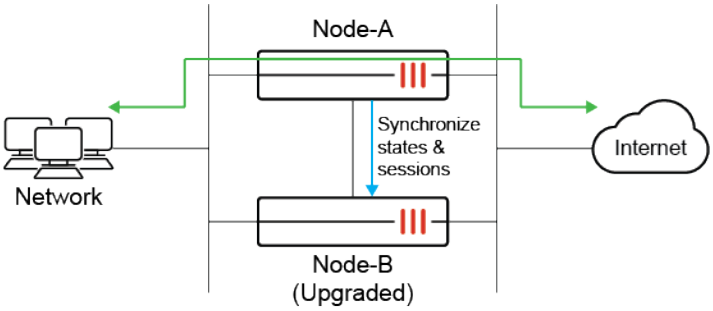
secondary-only		
Upgrade method	**Outcome**	**Recommendation**
Initiate the upload or upgrade on the primary.	All non-primary members are upgraded.	Recommended for scenarios where there is more than one secondary HA member.
Initiate the upload or upgrade on the secondary member.	The secondary member where the image is uploaded is upgraded.	Same result as initiating an upgrade on a secondary member in `local-only` mode.

This can apply to any HA clusters with two or more members. Administrators can initiate an upgrade on a secondary member by using its CLI console or accessing the device's GUI from its HA management interface.

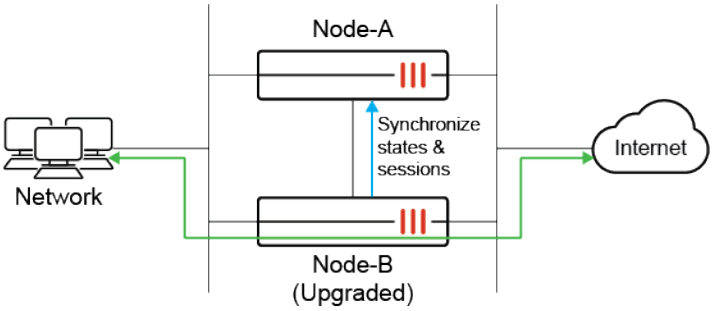
Initially, when you prepare an HA cluster in A-P mode for upgrade, traffic passes through the primary unit (Node-A) as the secondary unit (Node-B) sits on standby.



After the upgrade is completed on a secondary unit, states and sessions are synchronized. The members are now operating in MVC mode; however, traffic continues to pass through Node-A.



Administrators can manually trigger failover to make Node-B the new primary when ready. This can be done by resetting the HA uptime or changing device priorities, whichever method is desired. Traffic now passes through Node-B.

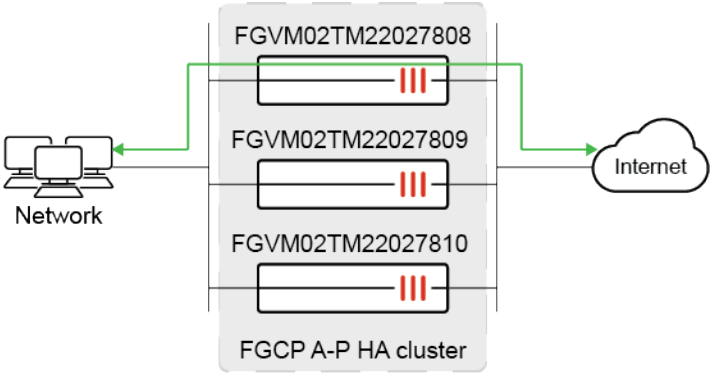


The upgraded system (Node-B) can be tested to verify that traffic can pass smoothly. If verification fails, administrators can trigger a failover to fail back to Node-A to avoid any downtime.

If verification is successful, administrators can manually trigger an upgrade on Node-A to bring the HA member up to the same version as Node-B to complete the HA upgrade procedure. This can be performed by accessing Node-A's GUI from its HA management interface or using its CLI console.

Example 1: upgrade a single secondary member using the local-only upgrade option

In this example, three HA members are running in an FGCP A-P HA cluster.



The member FGVM02TM22027808 is acting as the primary and forwarding traffic. The member FGVM02TM22027810 is chosen for upgrade.

The cluster is originally running build 2456. The secondary unit is upgraded to build 2461. Fictitious build numbers are used in this example to demonstrate functionality of the feature.

To configure the HA cluster:

```
config system ha
    set group-id 260
    set group-name "hagroup"
    set mode a-p
    set password **********
    set hbdev "port3" 0
    set session-pickup enable
    set upgrade-mode local-only
end
```

To perform the upgrade:

1. On the secondary member (FGVM02TM22027810), log in to the CLI console.
2. Execute a TFTP upgrade:

```
FGVM02TM22027810 # execute restore image tftp /home/Images/FortiOS/v7.00/images/build2461/FGT_
VM64-v7-build2461-FORTINET.out 172.16.100.71
This operation will replace the current firmware version!
Do you want to continue? (y/n)y

Please wait...

Connect to ftp server 172.16.100.71 ...
Get image from ftp server OK.
Verifying the signature of the firmware image.

Please wait for system to restart.
```

3. After the upgrade is complete, verify the version running on the secondary member:

```
FGVM02TM22027810 # get system status
Version: FortiGate-VM64 v7.4.1,build2461,230828 (interim)
…
```

4. On the primary unit, verify that HA is still formed between the three members:

```
FGVM02TM22027808 # diagnose sys ha dump-by group
<hatalk> vcluster_1: ha_prio=0(primary), state/chg_time/now=2(work)/1692750721/1693262149
            HA information.
group-id=260, group-name='hagroup'
has_no_aes128_gcm_sha256_member=0

gmember_nr=3
'FGVM02TM22027808': ha_ip_idx=2, hb_packet_version=10, last_hb_jiffies=0, linkfails=0,
weight/o=0/0, support_aes128_gcm_sha256=1
'FGVM02TM22027809': ha_ip_idx=1, hb_packet_version=12, last_hb_jiffies=51142842, linkfails=3,
weight/o=0/0, support_aes128_gcm_sha256=1
```

```
        hbdev_nr=1: port3(mac=000c..de, last_hb_jiffies=51142842, hb_lost=0),
'FGVM02TM22027810': ha_ip_idx=0, hb_packet_version=4, last_hb_jiffies=51142858, linkfails=3,
weight/o=0/0, support_aes128_gcm_sha256=1
        hbdev_nr=1: port3(mac=000c..1a, last_hb_jiffies=51142858, hb_lost=0),

vcluster_nr=1
vcluster-1: start_time=1692750718(2023-08-22 17:31:58), state/o/chg_time=2(work)/2
(work)/1692750721(2023-08-22 17:32:01)
        pingsvr_flip_timeout/expire=3600s/0s
        mondev: port1(prio=50,is_aggr=0,status=1) port7(prio=50,is_aggr=0,status=1) port8
(prio=50,is_aggr=0,status=1)
        'FGVM02TM22027808': ha_prio/o=0/0, link_failure=0, pingsvr_failure=0, flag=0x00000001,
mem_failover=0, uptime/reset_cnt=510868/0
        'FGVM02TM22027809': ha_prio/o=1/1, link_failure=0, pingsvr_failure=0, flag=0x00000000,
mem_failover=0, uptime/reset_cnt=510857/0
        'FGVM02TM22027810': ha_prio/o=2/2, link_failure=0, pingsvr_failure=0, flag=0x00000000,
mem_failover=0, uptime/reset_cnt=0/0
```

5. Fail over the HA cluster so that the secondary member, FGVM02TM22027810, becomes the primary. Since override is not enabled and the HA primary is determined by uptime, you can reset the HA uptime on the units that were not upgraded:

```
# diagnose sys ha reset-uptime
```

6. Once verification on the upgraded member is successful, repeat step 2 to perform upgrades on the remaining units.

Example 2: upgrade multiple secondary members using the secondary-only upgrade option

Using the same topology as example 1, the three HA cluster members are originally running build 2456. Both secondary units are upgraded using the `secondary-only` upgrade option. Fictitious build numbers are used in this example to demonstrate functionality of the feature.

To configure the HA cluster:

```
config system ha
    set group-id 260
    set group-name "hagroup"
    set mode a-p
    set password **********
    set hbdev "port3" 0
    set session-pickup enable
    set upgrade-mode secondary-only
end
```

To perform the upgrade:

1. On the primary unit (FGVM02TM22027808), log in to the CLI console.
2. Execute a TFTP upgrade:

```
FGVM02TM22027808 # execute restore image tftp /home/Images/FortiOS/v7.00/images/build2461/FGT_
VM64-v7-build2461-FORTINET.out 172.16.100.71
```

3. After the upgrade is complete, verify the version running on the secondary members.
   a. Member 1:

```
FGVM02TM22027809 # get system status
Version: FortiGate-VM64 v7.4.1,build2461,230828 (interim)
…
```

   b. Member 2:

```
FGVM02TM22027810 # get system status
Version: FortiGate-VM64 v7.4.1,build2461,230828 (interim)
…
```

4. On the primary unit, verify that HA is still formed between the three members:

```
FGVM02TM22027808 # diagnose sys ha dump-by group
            HA information.
group-id=260, group-name='hagroup'
has_no_aes128_gcm_sha256_member=0

gmember_nr=3
'FGVM02TM22027808': ha_ip_idx=2, hb_packet_version=19, last_hb_jiffies=0, linkfails=0,
weight/o=0/0, support_aes128_gcm_sha256=1
'FGVM02TM22027809': ha_ip_idx=1, hb_packet_version=4, last_hb_jiffies=51358055, linkfails=3,
weight/o=0/0, support_aes128_gcm_sha256=1
        hbdev_nr=1: port3(mac=000c..de, last_hb_jiffies=51358055, hb_lost=0),
'FGVM02TM22027810': ha_ip_idx=0, hb_packet_version=5, last_hb_jiffies=51358057, linkfails=3,
weight/o=0/0, support_aes128_gcm_sha256=1
        hbdev_nr=1: port3(mac=000c..1a, last_hb_jiffies=51358057, hb_lost=0),

vcluster_nr=1
vcluster-1: start_time=1692750718(2023-08-22 17:31:58), state/o/chg_time=2(work)/2
(work)/1692750721(2023-08-22 17:32:01)
        pingsvr_flip_timeout/expire=3600s/0s
        mondev: port1(prio=50,is_aggr=0,status=1) port7(prio=50,is_aggr=0,status=1) port8
(prio=50,is_aggr=0,status=1)
        'FGVM02TM22027808': ha_prio/o=0/0, link_failure=0, pingsvr_failure=0, flag=0x00000001,
mem_failover=0, uptime/reset_cnt=512775/0
        'FGVM02TM22027809': ha_prio/o=2/2, link_failure=0, pingsvr_failure=0, flag=0x00000000,
mem_failover=0, uptime/reset_cnt=0/0
        'FGVM02TM22027810': ha_prio/o=1/1, link_failure=0, pingsvr_failure=0, flag=0x00000000,
mem_failover=0, uptime/reset_cnt=1/0
```

Advanced operations

This section details complex deployment scenarios and specialized configurations for HA clusters. These advanced operations extend beyond standard setups, allowing for geographically distributed redundancy, customized synchronization behaviors, and integration with specific network infrastructures.

This section covers the following advanced topics and configurations:

Topic	Summary
Distributed HA clusters	Deploying cluster units across different physical locations to achieve geo-redundancy.
HA between remote sites over managed FortiSwitches	Utilizing the switch layer (MCLAG) to carry heartbeat signals between sites instead of dedicated fiber connections.
Cluster virtual MAC addresses	Managing how virtual MAC addresses are assigned to prevent conflicts and ensure reliable failover.
Disabling stateful SCTP inspection	Preventing packet drops in asymmetric routing scenarios within multi-homing topologies.
Manual failover of FortiGates deployed in an A-P architecture with VWP and using wildcard VLAN	Ensuring that non-native VLANs correctly update their paths during a failover in Virtual Wire Pair setups.
HA using a hardware switch to replace a physical switch	Deploying clusters without external switches by utilizing internal hardware switch configurations.
Override FortiAnalyzer and syslog server settings	Configuring secondary units to send logs to different destinations or use different source IPs than the primary unit.
FGCP HA between FortiGates of the same model with different AC and DC PSUs	Establishing redundancy across different power grids by clustering units with mixed power supply types.
Querying autoscale clusters for FortiGate VM	Verifying the status and synchronization of autoscale environments from secondary VM members.
Abbreviated TLS handshake after HA failover	Enabling faster session resumption for TLS traffic after a failover event.
HA with 802.3ad aggregate interfaces	Combine multiple physical links into a single logical interface.
HA with redundant interfaces	Combining two or more physical interfaces into a single redundant interface.
DHCP and PPPoE behavior in FortiGate HA clusters on page 4194	How FortiGate HA clusters handle scenarios where units obtain their own IP addresses using DHCP or PPPoE, and how they operate when functioning as a DHCP server or DHCP relay.
Monitoring VLAN interfaces on page 4196	Using VLAN heartbeat monitoring for HA clusters with VLAN interfaces.

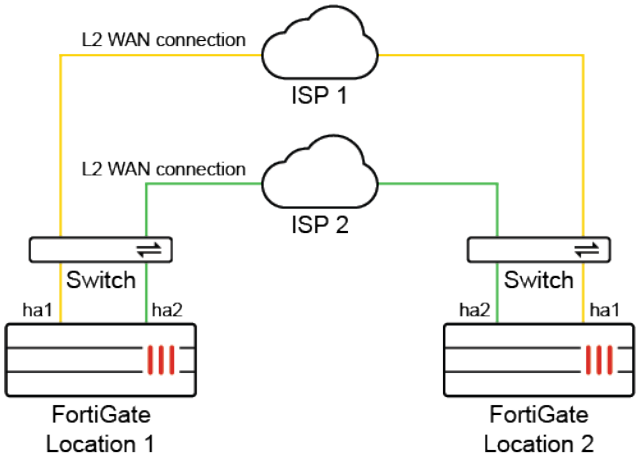
Distributed HA clusters

FGCP HA supports cluster units installed in different physical locations to achieve geo-redundancy. This may be desirable in large enterprises that deploy multiple data centers and network infrastructure to prevent interruptions caused by downtime in one location or region. Distributed clusters (or geographically distributed

clusters) can have cluster units in different rooms in the same building, different buildings in the same location, or different geographical regions (cities, countries, or continents). When disruption is detected in one location, traffic can be routed to another location and failed over to the HA unit in the same cluster to prevent major downtime.

Just like any FGCP HA cluster, distributed clusters require heartbeat communication between cluster units over a Layer 2 network. In a distributed cluster, this heartbeat communication can take place over a dedicated lease-line, MPLS, or other L2 WAN solutions. Most Data Center Interconnect (DCI) or MPLS-based solutions that support Layer 2 extensions between the remote data centers should also support HA heartbeat communication between the FortiGates in the distributed locations.

For more information about FGCP HA heartbeats, see HA heartbeat interface on page 4098.



Because of the possible distance between the cluster members, it may take longer for heartbeat packets to be transmitted between cluster units. If the time it takes and the possible latency and packet losses cause the configured heartbeat lost threshold to be exceeded, then a split brain scenario can occur (see Split brain scenario).

To avoid this, you can increase the heartbeat interval (the time between the sending of heartbeat packets) so that the cluster expects extra time between heartbeat packets. A general rule is to configure the failover time to be longer than the maximum latency. You could also increase the `hb-lost-threshold`, which is the number of lost heartbeats to signal a failure, in order to tolerate losing more heartbeat packets if the network connection is less reliable.

To configure the heartbeat interval and lost threshold:

```
config system ha
    set hb-interval <integer>
    set hb-lost-threshold <integer>
end
```

A longer interval and threshold can lead to slower failover time, and a shorter interval and threshold may lead to false positives. Therefore, these settings should be fine-tuned based on individual network scenarios. Additional options include:

- Using multiple heartbeat interfaces and different link paths for heartbeat packets to optimize HA heartbeat communication.

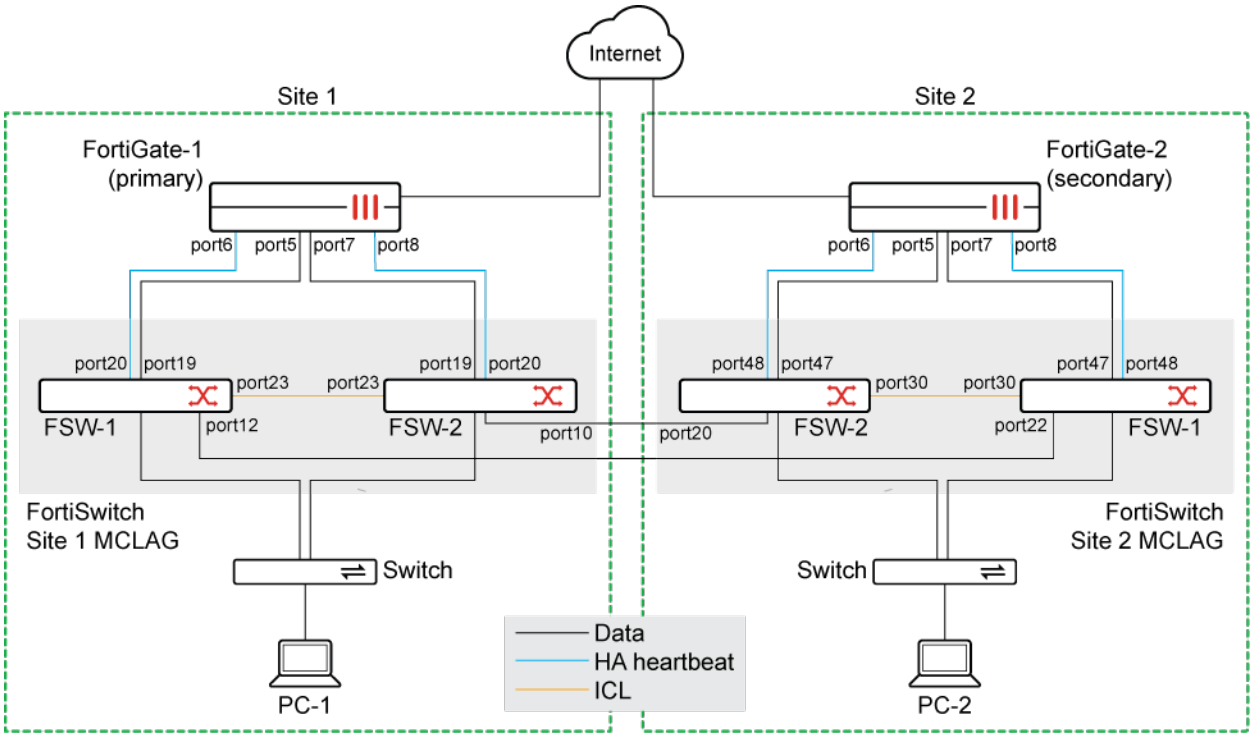
- Configuring QoS on the links used for HA heartbeat traffic to make sure heartbeat communication has the highest priority.

For information about changing the heartbeat interval and other heartbeat related settings, see Modifying heartbeat timing.

HA between remote sites over managed FortiSwitches

In a multi-site FortiGate HA topology that uses managed FortiSwitches in a multi-chassis link aggregation group (MCLAG) to connect between sites, HA heartbeat signals can be sent through the switch layer of the FortiSwitches, instead of through back-to-back links between the heartbeat interfaces. This means that two fiber connections can be used, instead of four (two back-to-back heartbeat fiber connections and two connections for the FortiSwitches). The FortiSwitches can be different models, but must all support MCLAG and be running version 6.4.2 or later.

This example shows how to configure heartbeat VLANs to assign to the access ports that the heartbeat interfaces connect to, passing over the trunk between the FortiSwitches on the two sites.

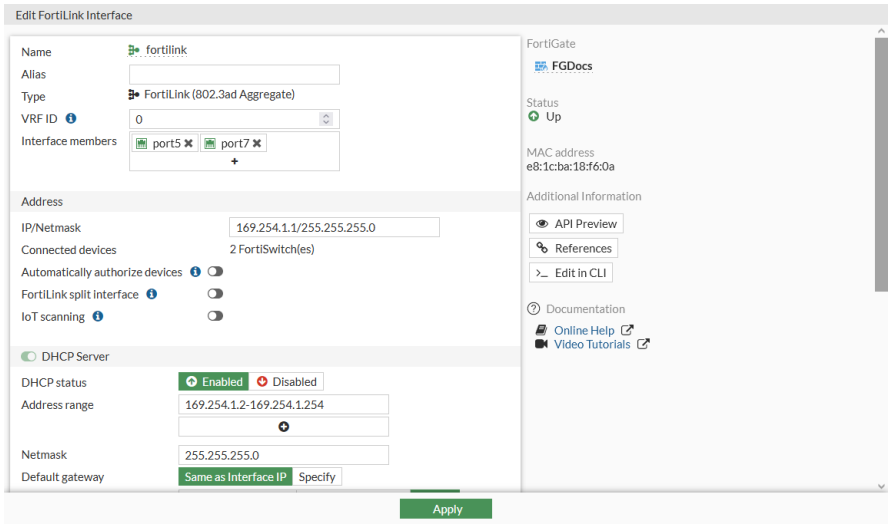


FortiGate HA is with two FortiGates in separate locations and the switch layer connection between the FortiSwitches is used for the heartbeat signal.

To configure the example:

1. Disconnect the physical connections between Site 1 and Site 2:
   - Disconnect the cable on Site 1 FSW-1 port 12.
   - Disconnect the cable on Site 1 FSW-2 port 10.
2. Configure Site 1:

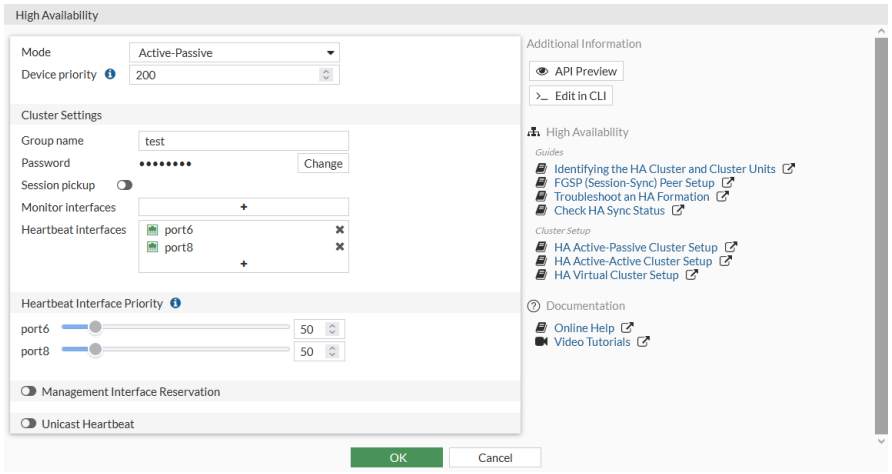
a. On the FortiGate, go to *WiFi & Switch Controller* > *FortiLink Interface* and configure FortiLink:



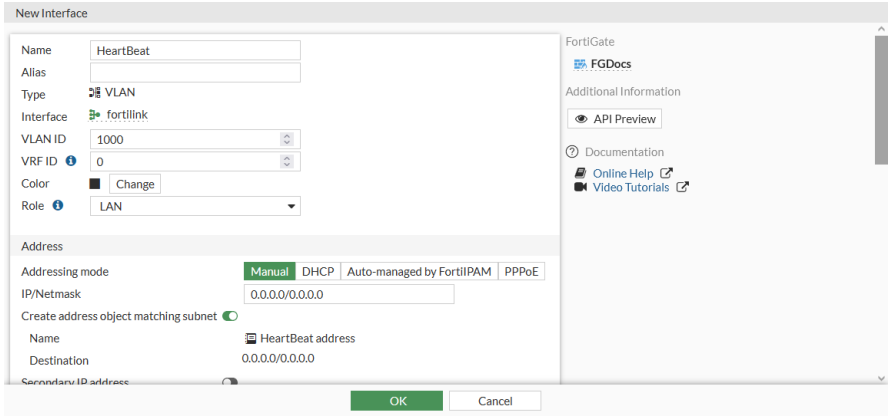
b. Go to *System* > *HA* and configure HA:

i. Set the heartbeat ports to the ports that are connected to FortiSwitch.

ii. Adjust the priority and enable override so that this FortiGate becomes the primary.



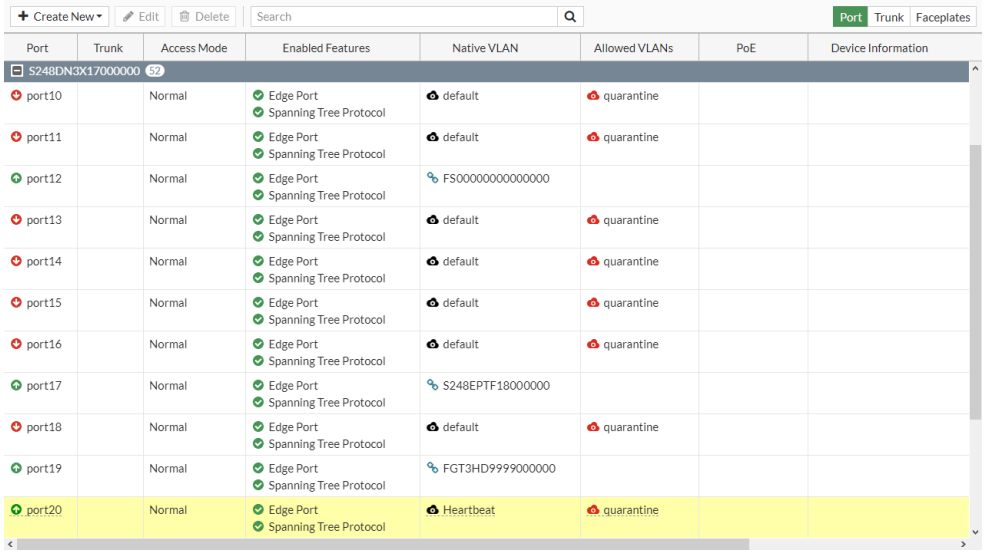
c. Go to *WiFi & Switch Controller* > *FortiSwitch VLANs* and create switch VLANs that are dedicated to each FortiGate HA heartbeat interface between the two FortiGates: Heartbeat VLAN 1000 and Heartbeat VLAN 1100.



d. Assign the native VLAN of the switch ports that are connected to the heartbeat ports to the created VLAN. Each HA heartbeat should be in its own VLAN.

i. Go to *WiFi & Switch Controller* > *FortiSwitch Ports*.

ii. In the *Native VLAN* column for the heartbeat port that is connected to FSW-1, click the edit icon and select the *Heartbeat* VLAN.



iii. In the *Native VLAN* column for the heartbeat port that is connected to FSW-2, click the edit icon and select the *Heartbeat2* VLAN.

e. On each FortiSwitch, enable MCLAG-ICL on the trunk port:

```
config switch trunk
    edit D243Z17000032-0
        set mclag-icl enable
    next
end
```

3. Configure Site 2 the same as Site 1, except set the HA priority so that the FortiGate becomes the secondary.
4. Disconnect the physical connections for FortiGate HA and FortiLink interfaces on Site 2:
   - Disconnect the cable on Site 2 FSW-1 ports 47 and 48.
   - Disconnect the cable on Site 2 FSW-2 ports 47 and 48.
5. Connect cables between the FortiSwitch MCLAG in Site 1 and Site 2:
   - Connect a cable from Site 1 FSW-1 port 12 to Site 2 FSW-1 port 22.
   - Connect a cable from Site 1 FSW-2 port 10 to Site 2 FSW-2 port 20.
6. On all of the FortiSwitches, configure the `auto-isl-port-group`. The group must match on both sides.

a. Site 1 FSW-1:

Set `members` to the port that is connected to Site 2 FSW-1:

```
config switch auto-isl-port-group
    edit 1
        set members port12
    next
end
```

b. Site 1 FSW-2:

Set `members` to the port that is connected to Site 1 FSW-1:

```
config switch auto-isl-port-group
    edit 1
        set members port22
    next
end
```

c. Site 2 FSW-1:

Set `members` to the port that is connected to Site 2 FSW-2:

```
config switch auto-isl-port-group
    edit 1
        set members port10
    next
end
```

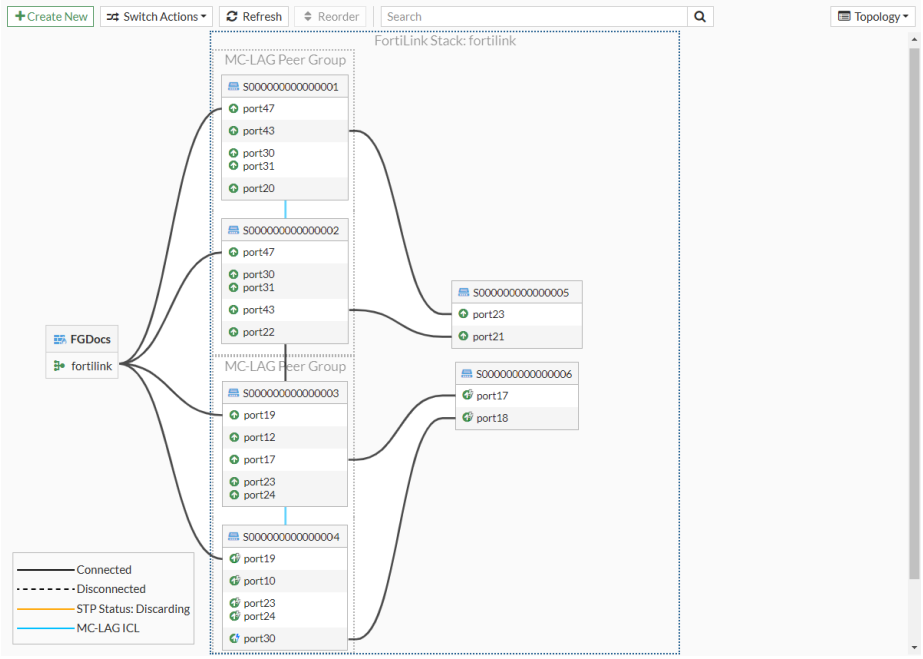
d. Site 2 FSW-2:

Set `members` to the port that is connected to Site 1 FSW-2:

```
config switch auto-isl-port-group
    edit 1
        set members port20
    next
end
```

7. Connect the FortiGate HA and FortiLink interface connections on Site 2.
8. Configure a firewall policy and route for traffic so that the client can reach the internet.
9. Wait for HA to finish synchronizing and for all of the FortiSwitches to come online, then on FortiGate-1, go to *WiFi & Switch Controller > Managed FortiSwitches* and select the Topology view from the drop-down on the right.

   The page should look similar to the following:



To test the configuration to confirm what happens when there is a failover:

1. On both PC-1 and PC-2, access the internet and monitor traffic. The traffic should be going through the primary FortiGate.
2. Perform a continuous ping to an outside IP address, then reboot any one of the FortiSwitches.
   Traffic from both Site 1 and Site 2 to the internet should be recovered in approximately five seconds.
3. Perform a continuous ping to an outside IP address, then force an HA failover (see Force HA failover for testing and demonstrations on page 4199).
   Traffic from both Site 1 and Site 2 to the internet should be recovered in approximately five seconds.
4. After an HA failover, on the new primary FortiGate, go to *WiFi & Switch Controller* > *Managed FortiSwitch*.
   The switch layer tiering will be changed so that the directly connected FortiSwitches are at the top of the topology.

Cluster virtual MAC addresses

In a cluster, each primary device interface requires a virtual MAC (VMAC) address. HA uses VMAC addresses during failover. If a failover occurs, the new primary device interfaces will have the same VMAC addresses and IP addresses as the failed primary device. As a result, most network equipment will identify the new primary device as the same device as the failed primary device and continue to communicate with the cluster.

The following methods are available to assign VMAC addresses to interfaces:

- Manual assignment
- Automatic assignment based on MAC address
- Automatic assignment based on vcluster ID, group ID, and physical index

See Determining VMAC addresses on page 4156 for more information.

If a cluster is operating in NAT mode with an automatic VMAC assignment method, FGCP assigns a different VMAC address to each primary device interface. VLAN subinterfaces are assigned the same VMAC address as the physical interface that the VLAN subinterface is added to. Redundant or 802.3ad aggregate interfaces are assigned the VMAC address of the first interface in the redundant or aggregate list.

If a cluster is operating in transparent mode with an automatic VMAC assignment method, FGCP assigns a VMAC address to the primary device's management IP address. Since you can connect to the management IP address from any interface, all FortiGate interfaces appear to have the same VMAC address.

The MAC address of a reserved management interface does not change to a VMAC address; it keeps its original MAC address.

Subordinate device MAC addresses do not change. Use `diagnose hardware deviceinfo nic <interface>` on the subordinate device to display the MAC addresses of each interface.

A MAC address conflict can occur when two clusters are operating on the same network using the same group ID (see Diagnosing packet loss). It is recommended that each cluster in the same network and broadcast domain uses a unique group

Failover

When the new primary device is selected after a failover, the primary device sends gratuitous ARP packets to update the devices connected to the cluster interfaces (usually layer 2 switches) with the VMAC addresses. This is sometimes called using gratuitous ARP packets (or GARP packets) to train the network. The gratuitous ARP packets sent from the primary unit are intended to make sure that the layer 2 switch forwarding databases (FDBs) are updated as quickly as possible.

Sending gratuitous ARP packets is not a requirement because connected devices will eventually learn of the new ports to forward the packets to. However, many network switches will update their FDBs more quickly after a failover if the new primary device sends gratuitous ARP packets.

Configuring ARP packet settings

The following settings can be configured.

```
config system ha
    set arps <integer>
    set arps-interval <integer>
    set gratuitous-arps {enable | disable}
    set link-failed-signal {enable | disable}
end
```

`arps <integer>`	Set the number of gratuitous ARPs; lower the value to reduce traffic, and increase the value to reduce failover time (1 - 60, default = 5).
`arps-interval <integer>`	Set the time between gratuitous ARPs; lower the value to reduce failover time, and increase the value to reduce traffic, in seconds (1 - 20, default = 8).
`gratuitous-arps {enable | disable}`	Enable/disable gratuitous ARPs (default = enable).

`link-failed-signal {enable | disable}`	Enable/disable shutting down all interfaces for one second after a failover. Use if gratuitous ARPs do not update the network (default = disable).

If you disable sending gratuitous ARP packets, it is recommended to enable the `link-failed-signal` setting. The `linked-fail-signal` alerts the connected switches of a failed link, which triggers them to react immediately to the changes.

For more information about gratuitous ARP packets see RFC 826 and RFC 3927.

Determining VMAC addresses

The following methods are available to assign virtual MAC addresses to interfaces, and they are listed in order of highest to lowest priority:

- Manual VMAC assignment on page 4156
- Automatic VMAC assignment based on hardware MAC address on page 4156
- Automatic VMAC assignment based on vcluster ID, group ID, and physical index on page 4157

Manual VMAC assignment

You can manually assign virtual MAC addresses to each interface, including physical, EMAC, or FortiExtender interfaces. Manual virtual MAC assignments override other virtual MAC address assignments on the interface.

```
config system interface
    edit <interface>
        set virtual-mac <mac_address>
    next
end
```

To manually assign a virtual MAC address to an interface:

```
config system interface
    edit "wan1"
        set ip 172.16.200.1 255.255.255.0
        set allowaccess ping https ssh snmp http telnet
        set virtual-mac 06:d5:90:04:f8:9c
        set type physical
        set snmp-index 3
        config ipv6
            set ip6-address 2000:172:16:200::1/64
            set ip6-allowaccess ping https ssh snmp http telnet
        end
    next
end
```

Automatic VMAC assignment based on hardware MAC address

FGCP can automatically assign virtual MAC addresses to physical interfaces based on hardware MAC address of the primary device with the locally administered bit (U/L bit) changed to `1`. For example, `00:xx:xx:xx:xx:xx` becomes `02:xx:xx:xx:xx:xx`. You must use `set auto-virtual-mac-interface` to the interfaces to use this feature.

```
config system ha
    set auto-virtual-mac-interface <interface> [interface(s)]
end
```

In a 48-bit MAC address, the U/L bit refers to the second least significant bit in the first octet of the hexadecimal MAC address. When this bit is 0, it indicates that the MAC address is Universal, meaning that it is assigned by a central authority. When this bit is 1, it indicates that the MAC address is Local, meaning that it is assigned locally.

For example, the first octet of 00 represented in binary is 00000000, where the U/L bit is 0. Whereas the first octet of 02 represented in binary is 00000010, where the U/L bit is set to 1.

To configure automatic virtual MAC address assignment:

```
config system ha
    set group-id 20
    set group-name "MMMMM"
    set mode a-p
    set password **********
    set hbdev "ha1" 50 "ha2" 100
    set auto-virtual-mac-interface "wan1" "port1" "port2" "ha1" "ha2" "port3" "port4" "port5"
"port6" "port7" "port8" "dmz"
    set upgrade-mode simultaneous
    set override enable
    set priority 200
end
```

Automatic VMAC assignment based on vcluster ID, group ID, and physical index

FGCP can automatically assign virtual MAC addresses to each primary device interface based on vcluster ID, group ID, and physical index. This automatic VMAC assignment method is used when the `set auto-virtual-mac-interface` option is disabled.

With this method, specific logic and variables determine VMAC addresses. The variables are:

- **Virtual cluster (vcluster) ID**: What is the vcluster ID that the interface belongs to?

  The vcluster ID used in the following logic is the number configured in the FortiGate minus 1. In this example the cluster ID is 0:

  ```
  config vcluster
      edit 1
  ```

- **Group ID**: What is the group ID that the interface belongs to?
- **Physical index**: What is the value of the physical index receiving the VMAC?

  The following command can be used to locate the physical index number:

  ```
  # diagnose sys ha dump-by debug-zone
  (…)
  <hatalk>              mgmt    ifindex=4       phyindex=0              mac=04.d5…
  ```

```
<hatalk>              ha      ifindex=3       phyindex=1              mac=04.d5…
<hatalk>              wan1    ifindex=17      phyindex=2              mac=04.d5…
<hatalk>              wan2    ifindex=18      phyindex=3              mac=04.d5…
```

The physical indexes used in this document are examples and may not match your interface indexes.

The logic uses the vcluster ID, group ID, and physical index variables to determine VMAC addresses as summarized in the following table:

Logic	Vcluster ID	Group ID	Physical index
Logic 1	0 or 1	Greater than 255	Less than 128
Logic 2	0 or 1	Less than 256	Less than 128
Logic 3	0 or 1	Less than 256	Greater than 127
Logic 4	Greater than 1	0-7	Less than 1024

Logic 1: vcluster ID 0 or 1, group ID > 255, and physical index < 128

The start of the VMAC address is always `e0:23:ff:--:--:--` with the last 24 bits defined as follows:

Preset bits	Group ID	Vcluster ID	Physical index
1 1 1 1 1 1	- - : - - - - - - - - :	-	- - - - - - -

This example uses group ID = 500 and vcluster ID = 0:

- Group ID
  - 500 – 256 = 244
  - $244_{10} = 011110100_2$
  - Group ID = 0011110100
- Vcluster ID
  - $0_{10} = 0_2$
  - Vcluster ID = 0
- Physical index
  - port7 physical index = 10
    - $10_{10} = 0001010_2$
  - port8 physical index = 12
    - $12_{10} = 0001100_2$
  - port9 physical index = 14
    - $14_{10} = 0001110_2$

Resulting in these VMAC addresses:

Interface	VMAC binary (last 24 bits)				VMAC hex (full)
	Preset bits	Group ID	Vcluster ID	Physical index	
port1	1 1 1 1 1 1	0 0 1 1 1 1 0 1 0 0	0	0 0 0 1 0 1 0	e0:23:ff:fc:f4:0a
port2	1 1 1 1 1 1	0 0 1 1 1 1 0 1 0 0	0	0 0 0 1 1 0 0	e0:23:ff:fc:f4:0c
port3	1 1 1 1 1 1	0 0 1 1 1 1 0 1 0 0	0	0 0 0 1 1 1 0	e0:23:ff:fc:f4:0e

Logic 2: vcluster ID 0 & 1, group ID < 256, and physical index < 128

The start of the VMAC address is always `00:09:0f:09:--:--` with the last 16 bits defined as follows:

Group ID	Vcluster ID	Physical index
- - - - - - - - : -	-	- - - - - - -

This example uses group ID = 200, vcluster ID = 1, and interfaces with physical indexes less than 128:

- Group ID:
  - $200_{10} = 11001000_2$
  - Group ID = 11001000
- Vcluster ID:
  - $1_{10} = 1_2$
  - Vcluster ID = 1
- Interfaces:
  - port25 physical index = 100
    - $100_{10} = 1100100_2$
  - port31 physical index = 120
    - $120_{10} = 1111000_2$
  - port38 physical index = 127
    - $127_{10} = 1111111_2$

Resulting in these VMAC addresses:

Interface	VMAC binary (last 16 bits)			VMAC hex (full)
	Group ID	Vcluster ID	Physical index	
port1	1 1 0 0 1 0 0 0	1	1 1 0 0 1 0 0	00:09:0f:09:c8:e4
port2	1 1 0 0 1 0 0 0	1	1 1 1 1 0 0 0	00:09:0f:09:c8:f8
port3	1 1 0 0 1 0 0 0	1	1 1 1 1 1 1 1	00:09:0f:09:c8:ff

Logic 3: vcluster ID 0 & 1, group ID < 256, and physical index > 127

The start of the VMAC address is always `70:4c:a5:--:--:--` with the last 24 bits defined as follows:

Physical index	Vcluster ID	Group ID
- - - - - - - - : - - - - - - - -	- :	- - - - - - - -

This example uses group ID = 25, vcluster ID = 1, and interfaces with physical indexes above 127:

- Group ID:
  - $25_{10} = 11001_2$
  - Group ID = 00011001
- Vcluster ID:
  - $1_{10} = 1_2$
  - Vcluster ID = 1
- Interfaces:
  - port40 physical index = 230
    - 230 – 128 = 102
    - $102_{10} = 1100110_2$
  - port45 physical index = 240
    - 240 – 128 = 112
    - $112_{10} = 1110000_2$
  - port50 physical index = 250
    - 250 – 128 = 122
    - $122_{10} = 1111010_2$

Resulting in these VMAC addresses:

Interface	VMAC binary (last 24 bits)			VMAC hex (full)
	Physical index	Vcluster ID	Group ID	
port1	0 0 0 0 0 0 0 0 1 1 0 0 1 1 0	1	0 0 0 1 1 0 0 1	70:4c:a5:00:cd:19
port2	0 0 0 0 0 0 0 0 1 1 1 0 0 0 0	1	0 0 0 1 1 0 0 1	70:4c:a5:00:e1:19
port3	0 0 0 0 0 0 0 0 1 1 1 1 0 1 0	1	0 0 0 1 1 0 0 1	70:4c:a5:00:f5:19

Logic 4: vcluster ID >= 2

When the vcluster ID is 2 or greater, the group ID must be between 0 and 7.

The start of the VMAC address is always `e0:23:ff:--:--:--` with the last 24 bits defined as follows:

Preset bits	Physical index	Vcluster ID	Group ID
1 1 1 1 1 1	- - : - - - - - - - - :	- - - - -	- - -

This example uses group ID = 6 and vcluster ID = 9:

- Group ID
  - $6_{10} = 110_2$
  - Group ID = 110

- Vcluster ID
  - $9_{10} = 1001_2$
  - Vcluster ID = 01001
- Interfaces
  - port1 physical index = 3_{10}
    - $3_{10} = 0000000011_2$
  - port2 physical index = 6_{10}
    - $6_{10} = 0000000110_2$
  - port3 physical index = 9_{10}
    - $9_{10} = 0000001001_2$

Resulting in these VMAC addresses:

Interface	VMAC binary (last 24 bits)				VMAC hex (full)
	Preset bits	Physical index	Vcluster ID	Group ID	
port1	1 1 1 1 1 1	0 0 0 0 0 0 0 0 1 1	0 1 0 0 1	1 1 0	e0:23:ff:fc:03:4e
port2	1 1 1 1 1 1	0 0 0 0 0 0 0 1 1 0	0 1 0 0 1	1 1 0	e0:23:ff:fc:06:4e
port3	1 1 1 1 1 1	0 0 0 0 0 0 1 0 0 1	0 1 0 0 1	1 1 0	e0:23:ff:fc:09:4e

Displaying VMAC addresses

Each FortiGate physical interface has two MAC addresses: the permanent and current hardware addresses. The permanent hardware address cannot be changed, as it is the actual MAC address of the interface hardware. The current hardware address can be changed, as it is the address seen by the network.

To change the current hardware address on a FortiGate not operating in HA:

```
config system interface
    edit <name>
        set macaddr <address>
    next
end
```

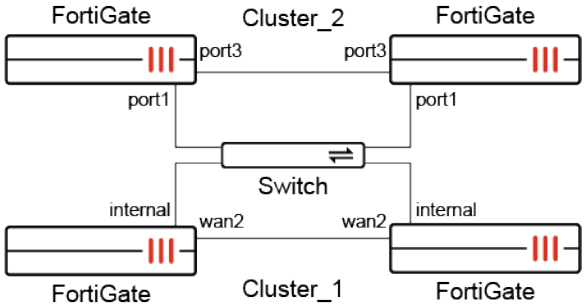
In an operating cluster, the current hardware address of each cluster device interface is changed to the HA virtual MAC address by the FGCP. The `macaddr` option is not available for a functioning cluster.

To display MAC addresses on a FortiGate operating in HA:

```
# diagnose hardware deviceinfo nic port1
...
Current_HWaddr 00:09:0f:09:ff:02
Permanent_HWaddr 08:5b:0e:72:3b:b2
```

Diagnosing packet loss

A network can experience packet loss when two FortiGate HA clusters are deployed in the same broadcast domain due to MAC address conflicts. You can resolve the MAC address conflict by changing the HA group ID (or cluster ID) configuration of the two clusters.



You can diagnose packet loss by pinging from one cluster to the other, or by pinging both of the clusters from a device within the broadcast domain.

To check for a MAC address conflict in a HA cluster:

1. On Cluster_1 and Cluster_2, check the VMAC address (`Current_HWaddr`) used in an interface on the primary device:

   ```
   # diagnose hardware deviceinfo nic <interface>
   ```

   If the group prefix and group hexadecimal ID are identical, there will be MAC address conflicts.
2. Change one of the clusters to use a different group ID:

   ```
   config system ha
       set group-id <integer>
   end
   ```

Disabling stateful SCTP inspection

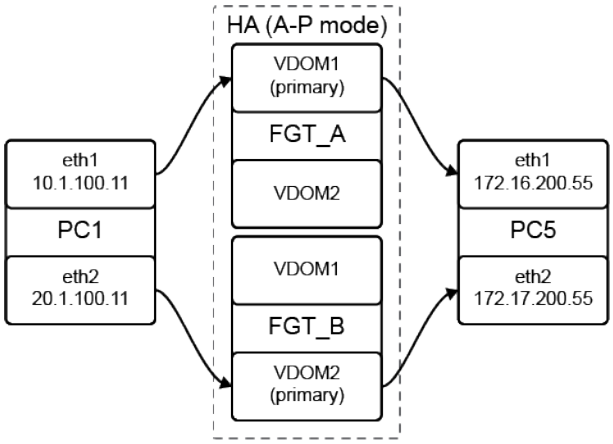
There is an option in FortiOS to disable stateful SCTP inspection. This option is useful when FortiGates are deployed in a high availability (HA) cluster that uses the FortiGate Clustering Protocol (FGCP) and virtual clustering in a multihoming topology. In this configuration, the primary stream control transmission protocol (SCTP) path traverses the primary FortiGate node by using its active VDOM (for example, VDOM1), and the backup SCTP path traverses the other passive FortiGate node by using its active VDOM (for example, VDOM2).

When stateful SCTP inspection is enabled, SCTP heartbeat traffic fails by means of the backup path because the primary path goes through a different platform and VDOM. Since there is no state sharing between VDOMs, the passive FortiGate is unaware of the original SCTP session and drops the heartbeats because of no associated sessions. When stateful SCTP inspection is disabled, the passive node permits the SCTP heartbeats to pass.

When set to `enable`, SCTP session creation without SCTP INIT is enabled. When set to `disable`, SCTP session creation without SCTP INIT is disabled (this is the default setting):

```
config system settings
    set sctp-session-without-init {enable | disable}
end
```

The following is an example topology and scenario:



In this example, FGT_A and FGT_B are in HA a-p mode with two virtual clusters. Two primaries exist on different FortiGate units. PC1 eth1 can access PC5 eth1 through VDOM1, and PC1 eth2 can access PC5 eth2 through VDOM2.

On PC5, to listen for an SCTP connection:

```
sctp_darn -H 172.16.200.55  -B 172.17.200.55 -P 2500 -l
```

On PC1, to start an SCTP connection:

```
sctp_darn -H 10.1.100.11 -B 20.1.100.11 -P 2600 -c 172.16.200.55 -c 172.17.200.55 -p 2500 -s
```

An SCTP four-way handshake is on one VDOM, and a session is created on that VDOM. With the default configuration, there is no session on any other VDOM, and the heartbeat on another path (another VDOM) is dropped. After enabling `sctp-session-without-init`, the other VDOM creates the session when it receives the heartbeat, and the heartbeat is forwarded:

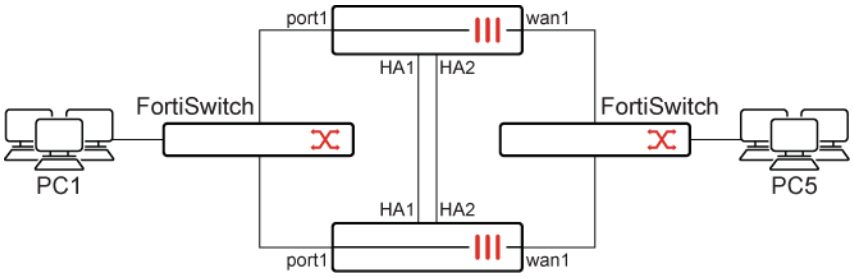
```
config system settings
  set sctp-session-without-init enable
end
```

Manual failover of FortiGates deployed in an A-P architecture with VWP and using wildcard VLAN

When the `bounce-intf-upon-failover` command is enabled, all monitored interfaces, includes aggregate interface members, are explicitly brought down during failover on the secondary device. This enhancement addresses a previous limitation where only the native VLAN received Gratuitous ARPs, leaving non-native VLANs unaware of the failover event.

```
config system ha
    set bounce-intf-upon-failover {enable | disable}
end
```

Example



1. Configure HA with `bounce-intf-upon-failover` enabled:

```
config system ha
    set group-id 222
    set group-name "FG_HA"
    set mode a-p
    set password **********
    set hbdev "ha1" 50 "ha2" 100
    set encryption enable
    set authentication enable
    set session-pickup enable
    set override disable
    set monitor "wan1" "port1"
    set ipsec-phase2-proposal aes256gcm
    set bounce-intf-upon-failover enable
end
```

2. Configure vlan20 on port1:

```
config system interface
    edit "vlan20"
        set vdom "root"
        set device-identification enable
        set snmp-index 36
        set interface "port1"
        set vlanid 20
    next
end
```

3. Configure VWP with members vlan20 and wan1:

```
config system virtual-wire-pair
    edit "sss"
        set member "vlan20" "wan1"
    next
end
```

4. Initiate a ping from PC1 to PC5 which go through the Virtual-Wire-Pair

5. Check the FDB table on the primary device:

```
diagnose netlink brctl name host sss_v.b
show bridge control interface sss_v.b host.
fdb: hash size=8192, used=10, num=10, depth=1, gc_time=4, ageing_time=300, version_
count=160109
Bridge sss_v.b host table
port no device  devname mac addr                ttl     attributes
  2     7       wan1    e0:23:ff:34:e0:90       1        Hit(1)
  2     7       wan1    00:09:0f:09:de:02       0       Local Static
  1     41      vlan20  00:0c:29:10:48:dc       93       Hit(93)
  2     7       wan1    e0:23:ff:fc:f4:02       2        Hit(2)
  2     7       wan1    02:42:ac:10:c8:2c       275
  2     7       wan1    02:42:ac:10:c8:37       0        Hit(0)
  1     41      vlan20  00:00:5e:00:01:01       0        Hit(0)
  2     7       wan1    00:0c:29:10:48:e6       93       Hit(93)
  1     41      vlan20  02:42:0a:01:64:16       0        Hit(0)
  1     41      vlan20  00:09:0f:09:de:06       0       Local Static
```

6. Check the FDB table on the secondary device:

```
diagnose netlink brctl name host sss_v.b
show bridge control interface sss_v.b host.
fdb: hash size=8192, used=10, num=10, depth=1, gc_time=4, ageing_time=300, version_count=0
Bridge sss_v.b host table
port no device  devname mac addr                ttl     attributes
  1     41      vlan20  00:03:e3:89:49:f2       150
  1     41      vlan20  00:0c:29:10:48:dc       145
  2     7       wan1    04:d5:90:04:f8:9c       0       Local Static
  2     7       wan1    02:42:ac:10:c8:2c       193
  1     41      vlan20  02:42:0a:01:64:0b       49
  2     7       wan1    02:42:ac:10:c8:37       189
  1     41      vlan20  00:00:5e:00:01:01       0
  2     7       wan1    00:0c:29:10:48:e6       49
  1     41      vlan20  02:42:0a:01:64:16       189
  1     41      vlan20  04:d5:90:04:f8:a0       0       Local Static
```

7. Check the MAC address table on the incoming switch:

```
# diagnose switch  mac-address list

MAC: 04:d5:90:04:f8:a0  VLAN: 20 Port: port1(port-id 1)
  Flags: 0x00000041 [ hit dynamic ]

MAC: e8:1c:ba:f2:a4:d6  VLAN: 20 Port: port2(port-id 2)
  Flags: 0x00000041 [ hit dynamic ]

MAC: 00:00:5e:00:01:01  VLAN: 20 Port: port3(port-id 3)
  Flags: 0x00000041 [ hit dynamic ]

MAC: e0:23:ff:fc:f4:02  VLAN: 20 Port: port2(port-id 2)
  Flags: 0x00000041 [ hit dynamic ]

MAC: 02:42:0a:01:64:16  VLAN: 20 Port: port21(port-id 21)
```

```
  Flags: 0x00000041 [ hit dynamic ]

MAC: 02:42:ac:10:c8:37  VLAN: 20 Port: port2(port-id 2)
  Flags: 0x00000041 [ hit dynamic ]

MAC: e0:23:ff:95:70:96  VLAN: 1 Port: internal(port-id 29)
  Flags: 0x00000060 [ static ]

MAC: 00:09:0f:09:6f:09  VLAN: 20 Port: port2(port-id 2)
  Flags: 0x00000041 [ hit dynamic ]

MAC: 00:0c:29:10:48:e6  VLAN: 20 Port: port2(port-id 2)
  Flags: 0x00000041 [ hit dynamic ]

MAC: 00:0c:29:10:48:dc  VLAN: 20 Port: port21(port-id 21)
  Flags: 0x00000041 [ hit dynamic ]

MAC: 00:03:e3:89:49:f2  VLAN: 20 Port: port19(port-id 19)
  Flags: 0x00000041 [ hit dynamic ]

Total Displayed: 11
```

8. Initiate HA failover:

```
execute ha failover set x
```

9. Confirm that the ping traffic is still going.
10. Recheck the FDB tables and note that the primary and secondary have switched.
11. Check the MAC address table on the incoming switch:

```
# diagnose switch mac-address list

MAC: 04:d5:90:04:f8:a0  VLAN: 20 Port: port1(port-id 1)
  Flags: 0x00000041 [ hit dynamic ]

MAC: e8:1c:ba:f2:a4:d6  VLAN: 20 Port: port2(port-id 2)
  Flags: 0x00000041 [ hit dynamic ]

MAC: 00:00:5e:00:01:01  VLAN: 20 Port: port3(port-id 3)
  Flags: 0x00000041 [ hit dynamic ]

MAC: e0:23:ff:fc:f4:02  VLAN: 20 Port: port1(port-id 1)
  Flags: 0x00000041 [ hit dynamic ]

MAC: 02:42:0a:01:64:16  VLAN: 20 Port: port21(port-id 21)
  Flags: 0x00000041 [ hit dynamic ]

MAC: 02:42:ac:10:c8:37  VLAN: 20 Port: port1(port-id 1)
  Flags: 0x00000041 [ hit dynamic ]

MAC: e0:23:ff:95:70:96  VLAN: 1 Port: internal(port-id 29)
  Flags: 0x00000060 [ static ]
```

```
MAC: e0:23:ff:34:e0:90  VLAN: 20 Port: port1(port-id 1)
  Flags: 0x00000041 [ hit dynamic ]

MAC: e0:23:ff:34:e0:8f  VLAN: 20 Port: port1(port-id 1)
  Flags: 0x00000041 [ hit dynamic ]

MAC: 00:0c:29:18:2d:90  VLAN: 20 Port: port1(port-id 1)
  Flags: 0x00000041 [ hit dynamic ]

MAC: 00:0c:29:10:48:e6  VLAN: 20 Port: port1(port-id 1)
  Flags: 0x00000041 [ hit dynamic ]

MAC: 00:0c:29:10:48:dc  VLAN: 20 Port: port21(port-id 21)
  Flags: 0x00000041 [ hit dynamic ]

MAC: 00:09:0f:09:de:03  VLAN: 20 Port: port1(port-id 1)
  Flags: 0x00000041 [ hit dynamic ]

MAC: 00:09:0f:09:de:02  VLAN: 20 Port: port1(port-id 1)
  Flags: 0x00000041 [ hit dynamic ]

MAC: 00:03:e3:89:49:f2  VLAN: 20 Port: port19(port-id 19)
  Flags: 0x00000041 [ hit dynamic ]

Total Displayed: 15
```

12. Check the FortiSwitch logs:

```
# execute log display
420 logs found.
10 logs returned.

1: 2025-07-10 17:35:14 log_id=0105008255 type=event subtype=spanning_tree pri=notice vd=root
user="stp_daemon" action=state-change unit="primary" switch.physical-port="port2"
instanceid="0" event="state migration" oldstate="discarding" newstate="forwarding"
msg="primary port port2 instance 0 changed state from discarding to forwarding"

2: 2025-07-10 17:35:12 log_id=0105008255 type=event subtype=spanning_tree pri=notice vd=root
user="stp_daemon" action=role-change unit="primary" switch.physical-port="port2"
instanceid="0" event="role migration" oldrole="disabled" newrole="designated" msg="primary
port port2 instance 0 changed role from disabled to designated"

3: 2025-07-10 17:35:12 log_id=0100001400 type=event subtype=link pri=information vd=root
unit="primary" switch.physical-port="port2" status="up" msg="primary switch port port2 has
come up"

4: 2025-07-10 17:35:09 log_id=0105008255 type=event subtype=spanning_tree pri=notice vd=root
user="stp_daemon" action=state-change unit="primary" switch.physical-port="port2"
instanceid="0" event="state migration" oldstate="forwarding" newstate="discarding"
msg="primary port port2 instance 0 changed state from forwarding to discarding"

5: 2025-07-10 17:35:09 log_id=0105008255 type=event subtype=spanning_tree pri=notice vd=root
```

```
user="stp_daemon" action=role-change unit="primary" switch.physical-port="port2"
instanceid="0" event="role migration" oldrole="designated" newrole="disabled" msg="primary
port port2 instance 0 changed role from designated to disabled"

6: 2025-07-10 17:35:09 log_id=0100001401 type=event subtype=link pri=information vd=root
unit="primary" switch.physical-port="port2" status="down" msg="primary switch port port2 has
gone down"

7: 2025-07-10 17:20:10 log_id=0103032001 type=event subtype=system pri=information vd=root
user="admin" ui=console action=login status=success reason=none profile="super_admin"
msg="Administrator admin logged in successfully from console"

8: 2025-07-10 17:19:30 log_id=0103032002 type=event subtype=system pri=alert vd=root
user="8TZWR/tWanTlR11x9Diizg==" ui=console action=login status=failed reason="name_invalid"
msg="Administrator 8TZWR/tWanTlR11x9Diizg== login failed from console because of invalid user
name"

9: 2025-07-10 11:10:33 log_id=0105008255 type=event subtype=spanning_tree pri=notice vd=root
user="stp_daemon" action=state-change unit="primary" switch.physical-port="port9"
instanceid="0" event="state migration" oldstate="discarding" newstate="forwarding"
msg="primary port port9 instance 0 changed state from discarding to forwarding"

10: 2025-07-10 11:10:32 log_id=0105008255 type=event subtype=spanning_tree pri=notice vd=root
user="stp_daemon" action=state-change unit="primary" switch.physical-port="port3"
instanceid="0" event="state migration" oldstate="discarding" newstate="forwarding"
msg="primary port port3 instance 0 changed state from discarding to forwarding"
```

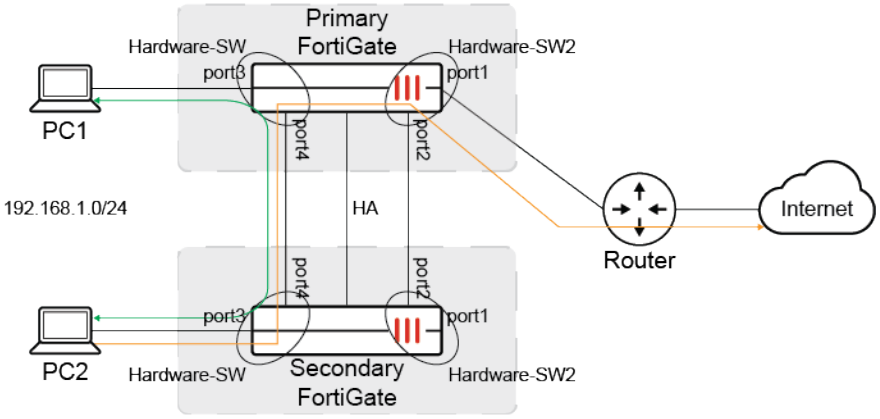
HA using a hardware switch to replace a physical switch

An HA cluster can be deployed without physical switches connecting the traffic interfaces on the primary and secondary members. This setup may be desirable in certain environments where the network infrastructure must be kept to a bare minimum.

Generally, using a hardware switch to replace a physical switch is not recommended, as it offers no redundancy or interface monitoring. If one FortiGate loses power, all of the clients connected to that FortiGate device cannot go to another device until that FortiGate recovers. A hardware switch cannot be used as a monitor interface in HA. Any incoming or outgoing link failures on hardware member interfaces will not trigger failover; this can affect traffic. Therefore, assess your environment thoroughly before applying this solution.

Examples

The examples use the following topology:



Traffic between hardware switches

When using Hardware switch in HA environment, a client device connected to the hardware switch on the primary FortiGate can communicate with client devices connected to the hardware switch on secondary FortiGates as long as there is a direct connection between the two switches.

To configure the FortiGate devices:

1. Connect the LAN side of the FortiGate cluster as shown in the topology diagram.
2. On each FortiGate, configure HA:

```
config system ha
    set mode a-a
    set group-name Example_cluster
    set password **********
    set hbdev ha1 10 ha2 20
end
```

3. On the primary FortiGate, configure the hardware switch:

```
config system virtual-switch
    edit Hardware-SW
        set physical-switch sw0
        config port
            edit port3
            next
            edit port4
            next
        end
    next
end
```

4. On each FortiGate, configure the IP addresses on the hardware switches:

```
config system interface
    edit Hardware-SW
        set ip 192.168.10.1 255.255.255.0
        set allowaccess ping ssh http https
    next
end
```

After configuring the hardware switches, PC1 and PC2 can now communicate with each other.

Traffic passes through FortiGate

If client device needs to send traffic through the FortiGate, additional firewall configuration on the FortiGate is required.

All traffic from the hardware switches on either the primary or secondary FortiGate reaches the primary FortiGate first. The traffic is then directed according to the HA mode and firewall configuration.

On the WAN side, in order for both HA members to reach the upstream router without connecting to a switch, a hardware switch must be configured with a direct connection between the cluster members.

To configure the FortiGate devices:

1. Connect the WAN side of the FortiGate cluster as shown in the topology diagram.
2. On the primary FortiGate, configure another hardware switch for the WAN connection:

```
config system virtual-switch
    edit Hardware-SW2
        set physical-switch sw0
        config port
            edit port1
            next
            edit port2
            next
        end
    next
end
```

3. On each FortiGate, configure the IP addresses on the hardware switch:

```
config system interface
    edit Hardware-SW2
        set ip 172.16.200.1 255.255.255.0
        set allowaccess ping ssh http https
    next
end
```

4. On each FortiGate, configure a firewall policy:

```
config firewall policy
    edit 1
        set srcintf Hardware-SW
        set dstintf Hardware-SW2
```

```
        set srcaddr all
        set dstaddr all
        set service ALL
        set action accept
        set schedule always
        set nat enable
    next
end
```

5. On each FortiGate, configure a static route:

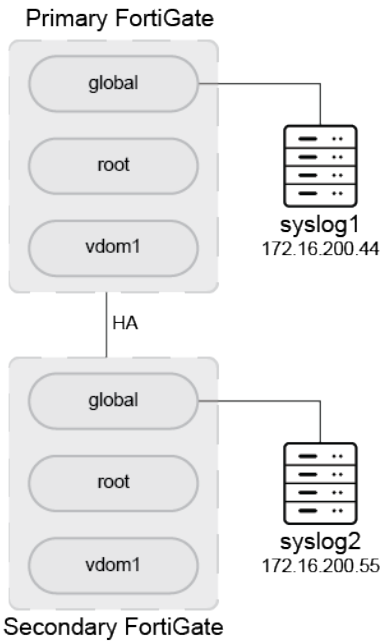
```
config router static
    edit 1
        set device Hardware-SW2
        set gateway 172.16.200.254
    next
end
```

Traffic from PC1 and PC2 can now reach destinations outside of the FortiGate cluster.

Override FortiAnalyzer and syslog server settings

In an HA cluster, secondary devices can be configured to use different FortiAnalyzer devices and syslog servers than the primary device. VDOMs can also override global syslog server settings.

Configure a different syslog server on a secondary HA device



To configure the primary HA device:

1. Configure a global syslog server:

```
config global
    config log syslog setting
        set status enable
        set server 172.16.200.44
        set facility local6
        set format default
    end
end
```

2. Set up a VDOM exception to enable setting the global syslog server on the secondary HA device:

```
config global
    config system vdom-exception
        edit 1
            set object log.syslogd.setting
        next
    end
end
```

To configure the secondary HA device:

1. Configure a global syslog server:

```
config global
    config log syslogd setting
        set status enable
        set server 172.16.200.55
        set facility local5
    end
end
```

2. After the primary and secondary device synchronize, generate logs on the secondary device.

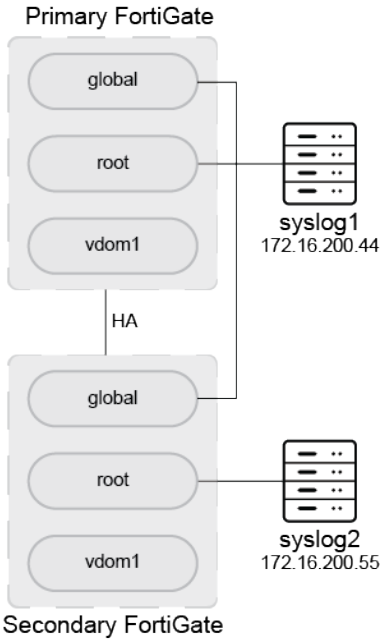
To confirm that logs are been sent to the syslog server configured on the secondary device:

1. On the primary device, retrieve the following packet capture from the secondary device's syslog server:

```
# diagnose sniffer packet any "host 172.16.200.55" 6
interfaces=[any]
filters=[host 172.16.200.55]

266.859494 port2 out 172.16.200.2.7434 -> 172.16.200.55.514: udp 278
0x0000   0000 0000 0000 0009 0f09 0004 0800 4500        ..............E.
0x0010   0132 f3c7 0000 4011 9d98 ac10 c802 ac10        .2....@.........
0x0020   c837 1d0a 0202 011e 4b05 3c31 3734 3e64        .7......K.<174>d
0x0030   6174 653d 3230 3230 2d30 332d 3134 2074        ate=2020-03-14.t
0x0040   696d 653d 3132 3a30 303a 3035 2064 6576        ime=12:00:05.dev
0x0050   6e61 6d65 3d22 466f 7274 6947 6174 652d        name="FGT-81E-Sl
```

```
0x0060   3831 455f 4122 2064 6576 6964 3d22 4647        ave-A".devid="FG
0x0070   5438 3145 3451 3136 3030 3030 3438 2220        T81E4Q16000048".
0x0080   6c6f 6769 643d 2230 3130 3030 3230 3032        logid="010002002
0x0090   3722 2074 7970 653d 2265 7665 6e74 2220        7".type="event".
0x00a0   7375 6274 7970 653d 2273 7973 7465 6d22        subtype="system"
0x00b0   206c 6576 656c 3d22 696e 666f 726d 6174        .level="informat
0x00c0   696f 6e22 2076 643d 2276 646f 6d31 2220        ion".vd="vdom1".
0x00d0   6576 656e 7474 696d 653d 3135 3834 3231        eventtime=158421
0x00e0   3234 3035 3835 3938 3335 3639 3120 747a        2405859835691.tz
0x00f0   3d22 2d30 3730 3022 206c 6f67 6465 7363        ="-0700".logdesc
0x0100   3d22 4f75 7464 6174 6564 2072 6570 6f72        ="Outdated.repor
0x0110   7420 6669 6c65 7320 6465 6c65 7465 6422        t.files.deleted"
0x0120   206d 7367 3d22 4465 6c65 7465 2031 206f        .msg="Delete.1.o
0x0130   6c64 2072 6570 6f72 7420 6669 6c65 7322        ld.report.files"
```

Configure a different syslog server in the root VDOM on a secondary HA device



To configure the primary HA device:

1. Configure a global syslog server:

```
config global
    config log syslog setting
        set status enable
        set server 172.16.200.44
        set facility local6
        set format default
    end
end
```

2. Set up a VDOM exception to enable `syslog-override` in the secondary HA device root VDOM:

```
config global
    config system vdom-exception
        edit 1
            set object log.syslogd.override-setting
            set scope inclusive
            set vdom root
        next
    end
end
```

3. In the VDOM, enable `syslog-override` in the log settings, and set up the override syslog server:

```
config root
    config log setting
        set syslog-override enable
    end
    config log syslog override-setting
        set status enable
        set server 172.16.200.44
        set facility local6
        set format default
    end
end
```

After `syslog-override` is enabled, an override syslog server must be configured, as logs will not be sent to the global syslog server.

To configure the secondary HA device:

1. Configure an override syslog server in the root VDOM:

```
config root
    config log syslogd override-setting
        set status enable
        set server 172.16.200.55
        set facility local5
        set format default
    end
end
```

2. After the primary and secondary device synchronize, generate logs in the root VDOM on the secondary device.

To confirm that logs are been sent to the syslog server configured for the root VDOM on the secondary device:

1. On the primary device, retrieve the following packet capture from the syslog server configured in the root VDOM on the secondary device:

```
# diagnose sniffer packet any "host 172.16.200.55" 6
interfaces=[any]
filters=[host 172.16.200.55]
```

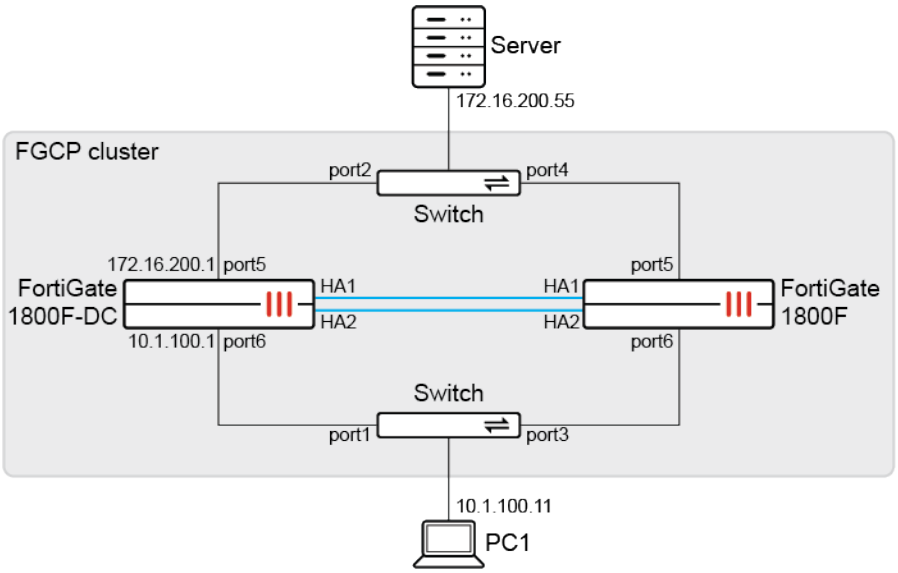
```
156.759696 port2 out 172.16.200.2.1165 -> 172.16.200.55.514: udp 277
0x0000   0000 0000 0000 0009 0f09 0004 0800 4500        ..............E.
0x0010   0131 f398 0000 4011 9dc8 ac10 c802 ac10        .1....@.........
0x0020   c837 048d 0202 011d af5f 3c31 3734 3e64        .7......._<174>d
0x0030   6174 653d 3230 3230 2d30 332d 3134 2074        ate=2020-03-14.t
0x0040   696d 653d 3131 3a33 353a 3035 2064 6576        ime=11:35:05.dev
0x0050   6e61 6d65 3d22 466f 7274 6947 6174 652d        name="FGT-81E-Sl
0x0060   3831 455f 4122 2064 6576 6964 3d22 4647        ave-A".devid="FG
0x0070   5438 3145 3451 3136 3030 3030 3438 2220        T81E4Q16000048".
0x0080   6c6f 6769 643d 2230 3130 3030 3230 3032        logid="010002002
0x0090   3722 2074 7970 653d 2265 7665 6e74 2220        7".type="event".
0x00a0   7375 6274 7970 653d 2273 7973 7465 6d22        subtype="system"
0x00b0   206c 6576 656c 3d22 696e 666f 726d 6174        .level="informat
0x00c0   696f 6e22 2076 643d 2272 6f6f 7422 2065        ion".vd="root".e
0x00d0   7665 6e74 7469 6d65 3d31 3538 3432 3130        venttime=1584210
0x00e0   3930 3537 3539 3334 3132 3632 2074 7a3d        905759341262.tz=
0x00f0   222d 3037 3030 2220 6c6f 6764 6573 633d        "-0700".logdesc=
0x0100   224f 7574 6461 7465 6420 7265 706f 7274        "Outdated.report
0x0110   2066 696c 6573 2064 656c 6574 6564 2220        .files.deleted".
0x0120   6d73 673d 2244 656c 6574 6520 3220 6f6c        msg="Delete.2.ol
0x0130   6420 7265 706f 7274 2066 696c 6573 22          d.report.files"
```

FGCP HA between FortiGates of the same model with different AC and DC PSUs

To improve power redundancy, FGCP HA clusters can support forming HA between units of the same model but with different AC PSU and DC PSU power supplies. This enables redundancy in a situation where power is completely lost on the AC grid, but traffic can fail over to a cluster member running on an independent DC grid.

The cluster members must be the same model with the same firmware installed, and must have the same hardware configuration other than the PSU.

In the following examples, there is an FGCP cluster with AC and DC PSU members: a FortiGate 1800F-DC (primary) and FortiGate 1800F (secondary).



Basic configuration

To configure the FGCP cluster in the GUI:

1. On the primary FortiGate (FG-1800F-DC), go to *System* > *HA*.
2. Configure the following settings:

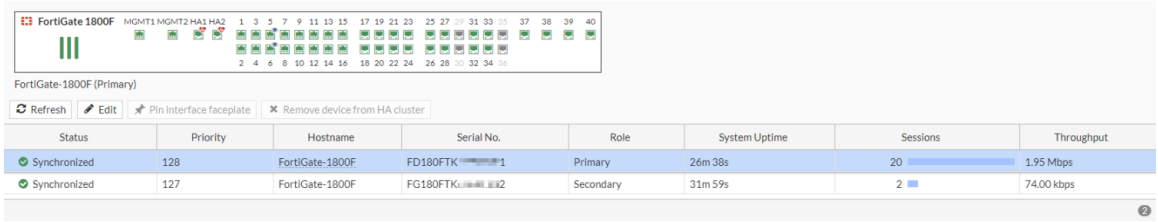
Mode	*Active-Passive*
Device priority	*128*
Group ID	*0*
Group name	*Example_cluster*
Password	Enter a password.
Session pickup	Enable this setting.
Monitor interfaces	Click the *+* to add *port5* and *port6*.
Heartbeat interfaces	Click the *+* to add *ha1* and *ha2*.

3. Click *OK*.
4. On the secondary FortiGate (FG-1800F), go to *System* > *HA*.
5. Configure the following settings:

Mode	*Active-Passive*
Device priority	*127*
Group ID	*0*
Group name	*Example_cluster*
Password	Enter a password.

Session pickup	Enable this setting.
Monitor interfaces	Click the *+* to add *port5* and *port6*.
Heartbeat interfaces	Click the *+* to add *ha1* and *ha2*.

6. Click *OK*.
7. Verify that the cluster status is *Synchronized*.



To configure the FGCP cluster in the CLI:

1. Configure the primary FortiGate (FG-1800F-DC):

```
config system ha
    set group-name "Example_cluster"
    set mode a-p
    set password **********
    set hbdev "ha2" 0 "ha1" 0
    set session-pickup enable
    set override disable
    set monitor "port5" "port6"
end
```

2. Configure the secondary FortiGate (FG-1800F):

```
config system ha
    set group-name "Example_cluster"
    set mode a-p
    set password **********
    set hbdev "ha2" 0 "ha1" 0
    set session-pickup enable
    set override disable
    set priority 127
    set monitor "port5" "port6"
end
```

3. Verify the cluster status on the primary FortiGate:

```
# get system ha status
HA Health Status: OK
Model: FortiGate-1800F
Mode: HA A-P
Group Name: Example_cluster
Group ID: 0
Debug: 0
Cluster Uptime: 0 days 0:56:11
```

```
Cluster state change time: 2023-05-29 19:11:14
Primary selected using:
    <2023/05/29 19:11:14> vcluster-1: FG180FTK*******1 is selected as the primary because its
uptime is larger than peer member FG180FTK*******2.
    <2023/05/29 18:59:45> vcluster-1: FG180FTK*******2 is selected as the primary because its
uptime is larger than peer member FG180FTK*******1.
    <2023/05/29 18:59:45> vcluster-1: FG180FTK*******1 is selected as the primary because its
override priority is larger than peer member FG180FTK*******2.
ses_pickup: enable, ses_pickup_delay=disable
override: disable
Configuration Status:
    FG180FTK*******1(updated 4 seconds ago): in-sync
    FG180FTK*******1 chksum dump: 95 4e 92 c3 39 75 8e 0e db 83 8d b7 b2 b1 9f 04
    FG180FTK*******2(updated 5 seconds ago): in-sync
    FG180FTK*******2 chksum dump: 95 4e 92 c3 39 75 8e 0e db 83 8d b7 b2 b1 9f 04
System Usage stats:
    FG180FTK*******1(updated 4 seconds ago):
        sessions=4, npu-sessions=0, average-cpu-user/nice/system/idle=0%/0%/0%/99%, memory=22%
    FG180FTK*******2(updated 5 seconds ago):
        sessions=0, npu-sessions=0, average-cpu-user/nice/system/idle=0%/0%/0%/99%, memory=22%
HBDEV stats:
    FG180FTK*******1(updated 4 seconds ago):
        ha2: physical/10000full, up, rx-bytes/packets/dropped/errors=18367581/33512/0/0,
tx=9563450/16609/0/0
        ha1: physical/10000full, up, rx-bytes/packets/dropped/errors=11543018/22166/0/0,
tx=12359673/22151/0/0
    FG180FTK*******2(updated 5 seconds ago):
        ha2: physical/10000full, up, rx-bytes/packets/dropped/errors=19133123/35087/0/0,
tx=10685583/18475/0/0
        ha1: physical/10000full, up, rx-bytes/packets/dropped/errors=17011332/25876/0/0,
tx=11919050/24991/0/0
MONDEV stats:
    FG180FTK*******1(updated 4 seconds ago):
        port5: physical/1000full, up, rx-bytes/packets/dropped/errors=988220/13742/0/0,
tx=106998000/73260/0/0
        port6: physical/1000full, up, rx-bytes/packets/dropped/errors=107084264/73624/0/0,
tx=953158/13611/0/0
    FG180FTK*******2(updated 5 seconds ago):
        port5: physical/1000full, up, rx-bytes/packets/dropped/errors=38194/128/0/0,
tx=0/0/0/0
        port6: physical/1000full, up, rx-bytes/packets/dropped/errors=99019/448/0/0,
tx=0/0/0/0
Primary     : FortiGate-1800F , FG180FTK*******1, HA cluster index = 1
Secondary   : FortiGate-1800F , FG180FTK*******2, HA cluster index = 0
number of vcluster: 1
vcluster 1: work 169.254.0.2
Primary: FG180FTK*******1, HA operating index = 0
Secondary: FG180FTK*******2, HA operating index = 1
```

4. Verify the cluster status on the secondary FortiGate:

```
# get system ha status
HA Health Status: OK
Model: FortiGate-1800F
Mode: HA A-P
Group Name: Example_cluster
Group ID: 0
Debug: 0
Cluster Uptime: 0 days 0:56:53
Cluster state change time: 2023-05-29 19:11:14
Primary selected using:
    <2023/05/29 19:11:14> vcluster-1: FG180FTK*******1 is selected as the primary because its
uptime is larger than peer member FG180FTK*******2.
    <2023/05/29 18:59:45> vcluster-1: FG180FTK*******2 is selected as the primary because its
uptime is larger than peer member FG180FTK*******1.
    <2023/05/29 18:55:03> vcluster-1: FG180FTK*******2 is selected as the primary because it's
the only member in the cluster.
    <2023/05/29 18:54:57> vcluster-1: FG180FTK*******2 is selected as the primary because SET_
AS_SECONDARY flag is set on peer member FG180FTK*******1.
ses_pickup: enable, ses_pickup_delay=disable
override: disable
...
Secondary   : FortiGate-1800F , FG180FTK*******2, HA cluster index = 0
Primary     : FortiGate-1800F , FG180FTK*******1, HA cluster index = 1
number of vcluster: 1
vcluster 1: standby 169.254.0.2
Secondary: FG180FTK*******2, HA operating index = 1
Primary: FG180FTK*******1, HA operating index = 0
```

Testing synchronization in the cluster

Based on the preceding example, the interface and firewall policy configurations are changed on the primary FortiGate. These configuration changes and sessions are synchronized to the secondary FortiGate. If the switch interface connected to the primary's port5 is down (port2), this triggers the monitor interface to be down, and the PC1 traffic will fail over to the secondary FortiGate.

To test configuration synchronization in the FGCP cluster:

1. Modify configurations on the primary FortiGate (FG-1800F-DC).
   a. Edit the interface settings:

```
config system interface
    edit "port5"
        set ip 10.1.100.1 255.255.255.0
        set allowaccess ping https ssh http telnet
        set alias "To_Client_PC"
        config ipv6
            set ip6-address 2000:10:1:100::1/64
            set ip6-allowaccess ping https ssh http
        end
    next
    edit "port6"
```

```
        set ip 172.16.200.1 255.255.255.0
        set allowaccess ping https ssh http fgfm
        set alias "To_Server"
        config ipv6
            set ip6-address 2000:172:16:200::1/64
            set ip6-allowaccess ping https ssh http
        end
    next
end
```

b. Edit the firewall policy settings:

```
config firewall policy
    edit 1
        set name "to_server_policy"
        set srcintf "port5"
        set dstintf "port6"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set logtraffic-start enable
    next
end
```

2. On the secondary FortiGate (FG-1800F), verify that the settings were synchronized.

a. Verify the interface settings:

```
show system interface
    config system interface
        ...
        edit "port5"
            set vdom "root"
            set ip 10.1.100.1 255.255.255.0
            set allowaccess ping https ssh http telnet
            set type physical
            set alias "To_Client_PC"
            set snmp-index 9
            config ipv6
                set ip6-address 2000:10:1:100::1/64
                set ip6-allowaccess ping https ssh http
            end
        next
        edit "port6"
            set vdom "root"
            set ip 172.16.200.1 255.255.255.0
            set allowaccess ping https ssh http fgfm
            set type physical
            set alias "To_Server"
            set snmp-index 10
            config ipv6
```

```
                set ip6-address 2000:172:16:200::1/64
                set ip6-allowaccess ping https ssh http
            end
        next
    end
```

b. Verify the firewall policy settings:

```
show firewall policy
    config firewall policy
            edit 1
            set name "to_server_policy"
            set uuid 82a05e78-fe90-51ed-eb16-ee7bdea60de0
            set srcintf "port5"
            set dstintf "port6"
            set action accept
            set srcaddr "all"
            set dstaddr "all"
            set schedule "always"
            set service "ALL"
            set logtraffic-start enable
        next
    end
```

c. Verify the HA checksum:

```
# diagnose sys ha checksum show
is_manage_primary()=0, is_root_primary()=0
debugzone
global: 4e 15 af c3 c6 87 32 f5 69 5c b7 33 b1 8b 27 12
root: 4a 52 e4 f1 6a 2b eb 7d 84 7d f1 48 50 93 fe d9
all: 95 4e 92 c3 39 75 8e 0e db 83 8d b7 b2 b1 9f 04

checksum
global: 4e 15 af c3 c6 87 32 f5 69 5c b7 33 b1 8b 27 12
root: 4a 52 e4 f1 6a 2b eb 7d 84 7d f1 48 50 93 fe d9
all: 95 4e 92 c3 39 75 8e 0e db 83 8d b7 b2 b1 9f 04
```

To test session synchronization in the FGCP cluster:

1. On PC1, verify the IP address and gateway:

```
root@pc1:~# ifconfig eth1
eth1      Link encap:Ethernet  HWaddr 00:0c:29:a0:60:d6
          inet addr:10.1.100.11  Bcast:10.1.100.255  Mask:255.255.255.0
          ...

root@pc1:~# route -n
Kernel IP routing table
Destination     Gateway         Genmask         Flags Metric Ref    Use Iface
0.0.0.0         10.1.100.1      0.0.0.0         UG    0      0        0 eth1
10.1.100.0      0.0.0.0         255.255.255.0   U     0      0        0 eth1
```

```
10.6.30.0       0.0.0.0         255.255.255.0   U     0      0        0 eth0
169.254.0.0     0.0.0.0         255.255.0.0     U     1000   0        0 eth0
```

2. Using Wget, initiate a large file download with HTTP that will maintain a long session:

```
root@pc1:~# wget http://172.16.200.55/big100MB.html   --keep-session-cookies --limit-rate=128k
--progress=dot -S -r --delete-after
--2023-05-29 14:55:33--  http://172.16.200.55/big100MB.html
Connecting to 172.16.200.55:80... connected.
HTTP request sent, awaiting response...
  HTTP/1.1 200 OK
  Date: Mon, 29 May 2023 21:55:41 GMT
  Server: Apache/2.4.18 (Ubuntu)
  Last-Modified: Thu, 01 Dec 2016 00:17:35 GMT
  ETag: "6126784-5428dbf967ad3"
  Accept-Ranges: bytes
  Content-Length: 101869444
  Vary: Accept-Encoding
  Keep-Alive: timeout=5, max=100
  Connection: Keep-Alive
  Content-Type: text/html
Length: 101869444 (97M) [text/html]
Saving to: '172.16.200.55/big100MB.html'

     0K ..........  ..........  ..........  ..........  ..........  0%  199K 8m18s
    50K ..........  ..........  ..........  ..........  ..........  0%  100K 12m26s
   100K ..........  ..........  ..........  ..........  ..........  0%  200K 11m3s
   150K ..........  ..........  ..........  ..........  ..........  0%  100K 12m25s
   200K ..........  ..........  ..........  ..........  ..........  0%  100K 13m14s
   250K ..........  ..........  ..........  ..........  ..........  0%  200K 12m24s
```

3. On the primary FortiGate (FG-1800F-DC), check the session information:

```
# diagnose sys session filter  dport  80
# diagnose sys session list

session info: proto=6 proto_state=01 duration=5 expire=3594 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=4
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=may_dirty npu synced log-start
statistic(bytes/packets/allow_err): org=112/2/1 reply=60/1/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=13->14/14->13 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 10.1.100.11:54752->172.16.200.55:80(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.200.55:80->10.1.100.11:54752(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=1 pol_uuid_idx=15767 auth_info=0 chk_client_info=0 vd=0
serial=00000d80 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
```

```
npu_state=0x4000c00 ofld-O ofld-R
npu info: flag=0x81/0x81, offload=9/9, ips_offload=0/0, epid=133/132, ipid=132/133,
vlan=0x0000/0x0000
vlifid=132/133, vtag_in=0x0000/0x0000 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=12/12
total session: 1
```

4. On the secondary FortiGate (FG-1800F), check that the session is synchronized:

```
# diagnose sys session filter dport 80
# diagnose sys session list

session info: proto=6 proto_state=01 duration=47 expire=3552 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=dirty may_dirty npu syn_ses
statistic(bytes/packets/allow_err): org=0/0/0 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=13->14/14->13 gwy=0.0.0.0/0.0.0.0
hook=pre dir=org act=noop 10.1.100.11:54752->172.16.200.55:80(0.0.0.0:0)
hook=post dir=reply act=noop 172.16.200.55:80->10.1.100.11:54752(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=1 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=0
serial=00000d80 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000000
npu info: flag=0x00/0x00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0, vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
no_ofld_reason:
total session: 1
```

To test failover in the FGCP cluster:

1. On the switch connected to port5 of the primary FortiGate, change port2's status to be down:

```
config switch physical-port
    edit port2
        set status down
    next
end
```

2. Check the HA status on the primary FortiGate (FG-1800F-DC), which now becomes the secondary device:

```
# get system ha status
HA Health Status:
    WARNING: FG180FTK*******1 has mondev down;
Model: FortiGate-1800F
Mode: HA A-P
Group Name: Example_cluster
Group ID: 0
```

```
Debug: 0
Cluster Uptime: 0 days 1:16:13
Cluster state change time: 2023-05-29 20:08:56
Primary selected using:
    <2023/05/29 20:08:56> vcluster-1: FG180FTK*******2 is selected as the primary because the
value 0 of link-failure + pingsvr-failure is less than peer member FG180FTK*******1.
    <2023/05/29 19:11:14> vcluster-1: FG180FTK*******1 is selected as the primary because its
uptime is larger than peer member FG180FTK*******2.
    <2023/05/29 18:59:45> vcluster-1: FG180FTK*******2 is selected as the primary because its
uptime is larger than peer member FG180FTK*******1.
    <2023/05/29 18:59:45> vcluster-1: FG180FTK*******1 is selected as the primary because its
override priority is larger than peer member FG180FTK*******2.
ses_pickup: enable, ses_pickup_delay=disable
override: disable
...
Secondary   : FortiGate-1800F , FG180FTK*******1, HA cluster index = 1
Primary     : FortiGate-1800F , FG180FTK*******2, HA cluster index = 0
number of vcluster: 1
vcluster 1: standby 169.254.0.1
Secondary: FG180FTK*******1, HA operating index = 1
Primary: FG180FTK*******2, HA operating index = 0
```

3. Check the HA status on the new primary FortiGate (FG-1800F):

```
# get system ha status
HA Health Status:
    WARNING: FG180FTK*******1 has mondev down;
Model: FortiGate-1800F
Mode: HA A-P
Group Name: Example_cluster
Group ID: 0
Debug: 0
Cluster Uptime: 0 days 1:19:9
Cluster state change time: 2023-05-29 20:08:56
Primary selected using:
    <2023/05/29 20:08:56> vcluster-1: FG180FTK*******2 is selected as the primary because the
value 0 of link-failure + pingsvr-failure is less than peer member FG180FTK*******1.
    <2023/05/29 19:11:14> vcluster-1: FG180FTK*******1 is selected as the primary because its
uptime is larger than peer member FG180FTK*******2.
    <2023/05/29 18:59:45> vcluster-1: FG180FTK*******2 is selected as the primary because its
uptime is larger than peer member FG180FTK*******1.
    <2023/05/29 18:55:03> vcluster-1: FG180FTK*******2 is selected as the primary because it's
the only member in the cluster.
ses_pickup: enable, ses_pickup_delay=disable
override: disable
...
Primary     : FortiGate-1800F , FG180FTK*******2, HA cluster index = 0
Secondary   : FortiGate-1800F , FG180FTK*******1, HA cluster index = 1
number of vcluster: 1
vcluster 1: work 169.254.0.1
Primary: FG180FTK*******2, HA operating index = 0
Secondary: FG180FTK*******1, HA operating index = 1
```

4. On PC1, verify that the HTTP traffic remains uninterrupted:

```
...
74700K .......... .......... .......... .......... .......... 75%  100K 3m13s
74750K .......... .......... .......... .......... .......... 75%  200K 3m13s
74800K .......... .......... .......... .......... .......... 75%  100K 3m12s
74850K .......... .......... .......... .......... .......... 75%  200K 3m12s
74900K .......... .......... .......... .......... .......... 75%  100K 3m12s
74950K .......... .......... .......... .......... .......... 75%  100K 3m11s
75000K .......... .......... .......... .......... .......... 75%  200K 3m11s
75050K .......... .......... .......... .......... .......... 75%  100K 3m10s
75100K .......... .......... .......... .......... .......... 75%  200K 3m10s
75150K .......... .......... .......... .......... .......... 75%  100K 3m10s
```

Querying autoscale clusters for FortiGate VM

When a FortiGate VM secondary device is added to a cluster, the new secondary member can query the cluster about its autoscale environment. FortiManager can then run this query on the new secondary member to update its autoscale record.

To view cluster information from a secondary member:

```
# diagnose sys ha checksum autoscale-cluster
```

Cluster information sample

Sample cloud topology:

```
FGT_BYOL; primary; 10.0.0.6; FGVM04TM00000066
FGT_BYOL; secondary; 10.0.0.7; FGVM00000000056
FGT_PAYG; secondary; 10.0.0.4; FGTAZ000000000CD
FGT_PAYG; secondary; 10.0.0.5; FGTAZ0000000003D
```

From the secondary device, you can see cluster checksums and the primary device:

```
(Secondary) # diagnose sys ha checksum autoscale-cluster
================== FGTAZ000000000CD ==================
is_autoscale_master()=0
debugzone
global: 56 49 b3 02 f2 b7 5b 82 ec 2d c2 1a ff 80 8c 79
root: bf 18 cf 83 1e 04 c3 04 4c e4 66 bc 38 fe 3a dc
all: 77 06 d0 89 6e 06 c0 86 17 98 53 72 33 85 ae ff
checksum
global: 56 49 b3 02 f2 b7 5b 82 ec 2d c2 1a ff 80 8c 79
root: bf 18 cf 83 1e 04 c3 04 4c e4 66 bc 38 fe 3a dc
all: 77 06 d0 89 6e 06 c0 86 17 98 53 72 33 85 ae ff
================== FGVM04TM00000066 ==================
is_autoscale_master()=1
debugzone
```

```
global: 56 49 b3 02 f2 b7 5b 82 ec 2d c2 1a ff 80 8c 79
root: bf 18 cf 83 1e 04 c3 04 4c e4 66 bc 38 fe 3a dc
all: 77 06 d0 89 6e 06 c0 86 17 98 53 72 33 85 ae ff
checksum
global: 56 49 b3 02 f2 b7 5b 82 ec 2d c2 1a ff 80 8c 79
root: bf 18 cf 83 1e 04 c3 04 4c e4 66 bc 38 fe 3a dc
all: 77 06 d0 89 6e 06 c0 86 17 98 53 72 33 85 ae ff
================== FGVM00000000056 ==================
is_autoscale_master()=0
debugzone
global: 56 49 b3 02 f2 b7 5b 82 ec 2d c2 1a ff 80 8c 79
root: bf 18 cf 83 1e 04 c3 04 4c e4 66 bc 38 fe 3a dc
all: 77 06 d0 89 6e 06 c0 86 17 98 53 72 33 85 ae ff
checksum
global: 56 49 b3 02 f2 b7 5b 82 ec 2d c2 1a ff 80 8c 79
root: bf 18 cf 83 1e 04 c3 04 4c e4 66 bc 38 fe 3a dc
all: 77 06 d0 89 6e 06 c0 86 17 98 53 72 33 85 ae ff
================== FGTAZ0000000003D ==================
is_autoscale_master()=0
debugzone
global: 56 49 b3 02 f2 b7 5b 82 ec 2d c2 1a ff 80 8c 79
root: bf 18 cf 83 1e 04 c3 04 4c e4 66 bc 38 fe 3a dc
all: 77 06 d0 89 6e 06 c0 86 17 98 53 72 33 85 ae ff
checksum
global: 56 49 b3 02 f2 b7 5b 82 ec 2d c2 1a ff 80 8c 79
root: bf 18 cf 83 1e 04 c3 04 4c e4 66 bc 38 fe 3a dc
all: 77 06 d0 89 6e 06 c0 86 17 98 53 72 33 85 ae ff
```

To get ha sync information from the secondary device:

```
(Secondary) # get test hasync 50
autoscale_count=69. current_jiffies=41235125
    10.0.0.6, timeo=31430, serial_no=FGVM04TM19001766
    10.0.0.7, timeo=31430, serial_no=FGVM04TM19008156
    10.0.0.5, timeo=31430, serial_no=FGTAZR7UZRKKNR3D
```

Abbreviated TLS handshake after HA failover

TLS sessions that pass through an HA A-A or A-P cluster can use an abbreviated TLS handshake instead of a full TLS handshake upon failover from a primary HA unit to a secondary HA unit. This reduces session pickup delays by reducing the time needed to renegotiate the TLS session, given that the TLS session ticket can be re-used.

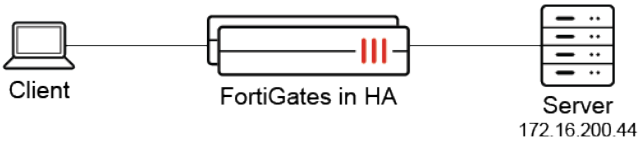
To accomplish this, FortiOS uses the web proxy global `ssl-ca-cert` to generate the key used in the TLS session ticket:

```
config web-proxy global
    set ssl-ca-cert "Fortinet_CA_SSL"
end
```

The certificate can be synchronized to the secondary HA unit, which allows the secondary unit to generate the same session key for a TLS session. When a TLS session reconnects after HA failover using the same session ticket as the first session, the new primary unit is able to generate the same key matching that session ticket and allow an abbreviated handshake.

Example

In this example, OpenSSL is used to create a TLS session between the client and the server through the primary FortiGate. The session ticket is outputted and saved. Upon failover, the same session ticket is reused to create a TLS session through the new primary unit. Because the new primary unit uses the same certificate to generate the key for the TLS session ticket, it allows the connection to be made using an abbreviated TLS handshake.



This example is for demonstration purposes only. In a normal failover, TLS sessions from clients will automatically be able to re-establish using an abbreviated handshake through the new primary unit.

To verify if an abbreviated TLS handshake is used after HA failover:

1. On the client using OpenSSL, open a new session to 172.16.200.44:443 and output the session ticket to a file called aaa.txt. This session will pass through the current HA primary unit:

```
# openssl s_client -connect 172.16.200.44:443 -sess_out aaa.txt
```

2. Fail over the primary unit to the secondary unit. The HA secondary unit starts handling the traffic.
3. On the client, try connecting to 172.16.200.44:443 using the same saved session ticket as before (aaa.txt):

```
# openssl s_client -connect 172.16.200.44:443 -sess_in aaa.txt
```

4. Verify whether the session succeeds in using the original session ticket:

```
Reused, TLSv1.3, Cipher is TLS_AES_256_GCM_SHA384
Server public key is 4096 bit
Secure Renegotiation IS NOT supported
Compression: NONE
Expansion: NONE
No ALPN negotiated
Early data was not sent
Verify return code: 19 (self signed certificate in certificate chain)
...
```

If the session is established using the same ticket, `Reused, TLSv1.3, Cipher is <name>` is displayed. If session is established using a new ticket, `New, TLSv1.3, Cipher is <name>` is displayed.

The new primary is able to use the web proxy global `ssl-ca-cert` to generate the same key as the old primary that was used in the session ticket. So, the second TLS connection that reuses the TLS session ticket from the first session can complete an abbreviated TLS handshake.

HA with 802.3ad aggregate interfaces

You can use 802.3ad Link Aggregation and Link Aggregation Control Protocol (LACP) to combine multiple physical links into a single logical interface. This increases throughput and provides network resiliency.

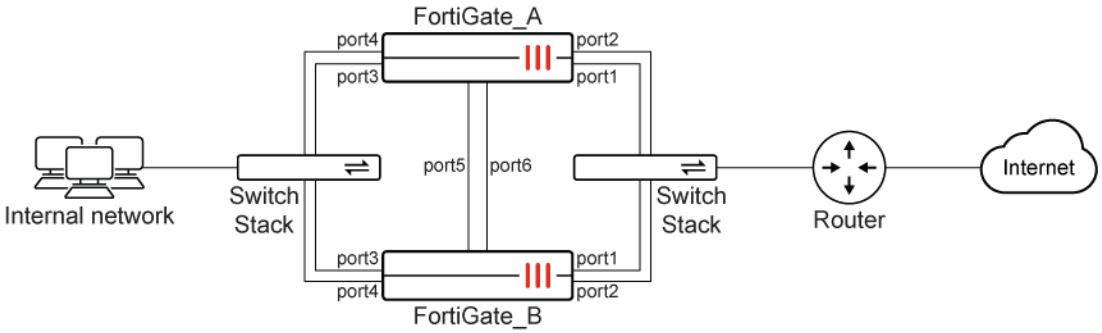
In an HA cluster, HA monitoring treats the aggregate link as a single interface. A failure is only registered if all physical members in the aggregate link fail. The aggregate interface uses a virtual MAC address derived from the first physical port in the bundle.

For optimal LACP and HA failover performance in Active-Passive or Active-Active modes, the connected switches should support multiple Link Aggregation Groups (LAGs). This allows the subordinate unit to participate in LACP negotiation during normal operation, reducing failover delay.

If the switch does not support multiple LAGs for the same MAC address (the virtual MAC shared by the cluster), you must prevent the subordinate unit from participating in LACP negotiation using the `set lacp-ha-slave disable` command to avoid cluster disruption. If the link-level redundancy is the primary requirement and load-balancing is not needed see HA with redundant interfaces on page 4191.

Disabling LACP on the subordinate unit prevents the use of Active-Active mode and may increase failover time. Additionally, using Static LACP may further reduce delay but risks failing to detect non-physical link issues due to the lack of LAC Protocol Data Unit (LACPDU) heartbeats.

In this example, two FortiGate units are configured in a HA cluster to ensure continuous network uptime. The units utilize Port 5 and Port 6 as dedicated heartbeat links to synchronize sessions and monitor device health. To prevent bottlenecks and provide redundancy, physical ports are combined into 802.3ad Link Aggregates, connecting the internal network and the internet router through redundant switches. If the primary FortiGate fails, the secondary unit instantly takes over the traffic, maintaining seamless connectivity for all users.



To configure the FortiGates for HA operation in the GUI:

1. On both FortiGates, go to *System* > *HA* and configure the following:

Option	Value
Mode	Active-Passive
Group Name	Pochiya
Password	**********
Heartbeat interfaces	port5 port6
Heartbeat Interface Priority	port5: 50 port6: 50

2. Click *OK*.

To configure the aggregate interfaces in the GUI:

1. Go to *Network* > *Static Routes* and temporarily delete any routes using member ports (port1 to port4).
2. Go to *Network* > *Interfaces* and click *Create new* > *Interface*.
3. Configure the WAN aggregate interface:

Option	Value
Name	WAN_Aggregate
Type	802.3ad Aggregate
Interface Members	port1 port2
IP/Netmask	172.20.120.141/24

4. Click *OK*.
5. Click *Create new* > *Interface* again to create the internal aggregate interface:

Option	Value
Name	Internal_Aggregate
Type	802.3ad Aggregate
Interface Members	port3 port4
IP/Netmask	10.11.101.100/24

6. Click *OK*.

To disable LACP on the subordinate unit (CLI only):

You must use the CLI to prevent the subordinate unit from sending LACP packets.

```
config system interface
    edit "WAN_Aggregate"
        set lacp-ha-slave disable
    next
    edit "Internal_Aggregate"
        set lacp-ha-slave disable
    next
end
```

To configure static routing in the GUI:

1. Go to *Network* > *Static Routes* and click *Create new*.
2. Configure the following:

Option	Value
Destination	0.0.0.0/0.0.0.0
Gateway Address	172.20.120.2
Interface	WAN_Aggregate

3. Click *OK*.

To configure HA port monitoring in the GUI:

1. Go to *System* > *HA*.
2. Edit the primary unit in the *Cluster Members* list.
3. Set *Monitor Interfaces* to WAN_Aggregate and Internal_Aggregate .
4. Click *OK*.

To configure HA with aggregate interfaces in the CLI:

1. Configure HA on both units:

```
config system ha
    set mode a-p
    set group-name Pochiya
    set password *********
    set hbdev port5 50 port6 50
end
```

2. Create aggregate interfaces with LACP slave participation disabled:

```
config system interface
    edit "WAN_Aggregate"
        set type aggregate
        set member port1 port2
        set lacp-ha-slave disable
        set ip 172.20.120.141 255.255.255.0
    next
    edit "Internal_Aggregate"
```

```
        set type aggregate
        set member port3 port4
        set lacp-ha-slave disable
        set ip 10.11.101.100 255.255.255.0
    next
end
```

3. Configure static routing:

```
config router static
    edit 1
        set gateway 172.20.120.2
        set device "WAN_Aggregate"
    next
end
```

4. Configure HA monitoring:

```
config system ha
    set monitor "WAN_Aggregate" "Internal_Aggregate"
end
```

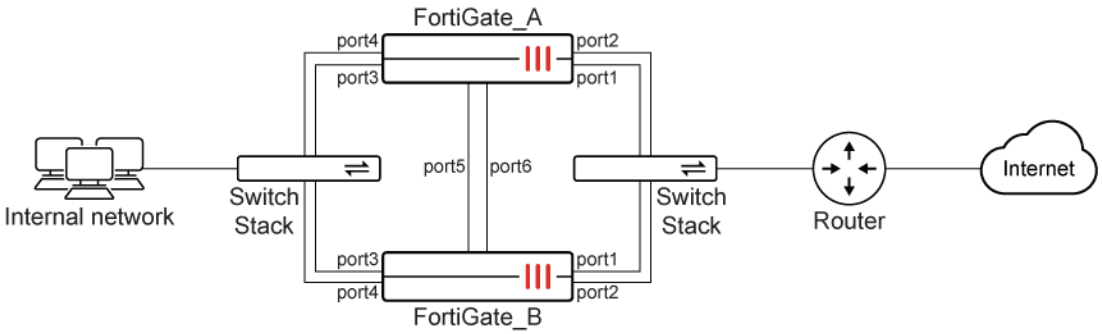
HA with redundant interfaces

You can combine two or more physical interfaces into a single redundant interface. Unlike aggregate interfaces (LACP), redundant interfaces do not increase throughput; instead, they provide a pure active-backup failover mechanism. Traffic is processed by a single port at a time, failing over to the next available member only if the active link goes down.

In an HA cluster, HA monitoring treats the redundant interface as a single logical entity. A cluster failover is triggered only if all physical ports within the redundant interface fail. The redundant interface uses a virtual MAC address derived from the first physical port added to its configuration.

While redundant interfaces provide physical link failover, they do not offer the load-balancing benefits of 802.3ad aggregation. See HA with 802.3ad aggregate interfaces on page 4188 for more information. Redundant interfaces are best suited for environments where switches do not support LACP or when simple link-level redundancy is the primary requirement.

In this example, two FortiGate units are configured in an HA cluster. Port 5 and Port 6 are used as heartbeat links. Redundant interfaces are used to connect to the internet (Port 1 and Port 2) and the internal network (Port 3 and Port 4). This ensures that if a cable or switch port fails on either unit, the redundant link takes over immediately without requiring a full cluster failover.



To configure the FortiGates for HA operation in the GUI:

1. On both FortiGates, go to *System* > *HA* and configure the following:

Option	Value
Mode	Active-Passive
Group Name	Pochiya
Password	**********
Heartbeat interfaces	port5 port6
Heartbeat Interface Priority	port5: 50 port6: 50

2. Click *OK*.

To configure the redundant interfaces in the GUI:

1. Go to *Network* > *Static Routes* and temporarily delete any routes using member ports (port1 to port4).
2. Go to *Network* > *Interfaces* and click *Create new* > *Interface*.
3. Configure the WAN redundant interface:

Option	Value
Name	WAN_Redundant
Type	Redundant Interface
Interface Members	port1 port2
IP/Netmask	172.20.120.141/24

4. Click *OK*.
5. Click *Create new* > *Interface* again to create the internal redundant interface:

Option	Value
Name	Internal_Redundant
Type	Redundant Interface
Interface Members	port3 port4
IP/Netmask	10.11.101.100/24

6. Click *OK*.

To configure static routing in the GUI:

1. Go to *Network* > *Static Routes* and click *Create new*.
2. Configure the following:

Option	Value
Destination	0.0.0.0/0.0.0.0
Gateway Address	172.20.120.2
Interface	WAN_Redundant

3. Click *OK*.

To configure HA port monitoring in the GUI:

1. Go to *System* > *HA*.
2. Edit the primary unit in the *Cluster Members* list.
3. Set *Monitor Interfaces* to WAN_Redundant and Internal_Redundant.
4. Click *OK*.

To configure HA with redundant interfaces in the CLI:

1. Configure HA on both units:

```
config system ha
    set mode a-p
    set group-name Pochiya
    set password *********
    set hbdev port5 50 port6 50
end
```

2. Create redundant interfaces:

```
config system interface
    edit "WAN_Redundant"
        set type redundant
        set member port1 port2
        set ip 172.20.120.141 255.255.255.0
```

```
    next
    edit "Internal_Redundant"
        set type redundant
        set member port3 port4
        set ip 10.11.101.100 255.255.255.0
    next
end
```

3. Configure static routing:

```
config router static
    edit 1
        set gateway 172.20.120.2
        set device "WAN_Redundant"
    next
end
```

4. Configure HA monitoring:

```
config system ha
    set monitor "WAN_Redundant" "Internal_Redundant"
end
```

DHCP and PPPoE behavior in FortiGate HA clusters

This topic outlines how FortiGate HA clusters handle scenarios where units obtain their own IP addresses using DHCP or PPPoE, and how they operate when functioning as a DHCP server or DHCP relay.

It includes the following operational modes:

- FortiGate as a DHCP or PPPoE client on page 4194
- DHCP relay operation in a FortiGate HA cluster on page 4195
- DHCP server operation in a FortiGate HA cluster on page 4195

Each subsection explains the expected behaviors, HA-specific requirements, and operational impacts to ensure consistency, reliability, and seamless failover across the cluster.

FortiGate as a DHCP or PPPoE client

A FortiGate interface can use DHCP or PPPoE to obtain its own IP address. These dynamic addressing methods are supported in HA deployments, but require specific configuration sequencing to ensure stable cluster operation.

HA support for DHCP and PPPoE

- FortiGate fully supports using DHCP or PPPoE on HA cluster interfaces.
- However, these features must be enabled after the HA cluster is formed.
- This ensures that the primary unit negotiates and maintains the correct dynamic IP address on behalf of the cluster.

Risks when enabled before HA formation

If an interface is already using DHCP or PPPoE before HA is enabled, the following issues may occur:

- Incorrect or inconsistent IP assignments across units.
- Failure to reach the DHCP or PPPoE server during HA negotiation.
- Possible disruption in WAN connectivity while the cluster becomes operational.

DHCP relay operation in a FortiGate HA cluster

A FortiGate HA cluster can function as a DHCP relay agent, forwarding DHCP client messages to an external DHCP server. In an HA environment, the relay process is tightly controlled to ensure consistent behavior across the cluster.

Primary unit responsibilities

In both active-passive and active-active HA modes, all DHCP relay sessions are handled exclusively by the primary unit. This ensures deterministic forwarding and prevents duplication of relay traffic.

Failover behavior

If an HA failover occurs:

- Active DHCP relay sessions are not resumed by the new primary. As a result, DHCP clients may need to retransmit their DHCP requests.
- The external DHCP server continues to manage all lease allocations, ensuring no lease conflicts occur.

DHCP server operation in a FortiGate HA cluster

A FortiGate HA cluster can function as a fully integrated DHCP server. In an HA setup, the primary unit is responsible for all DHCP related operations, ensuring consistent lease assignment and preventing conflicts.

Primary unit responsibilities

- Respond to all DHCP requests.
- Maintain the complete DHCP lease database.
- Automatically synchronize the lease database to all subordinate units.

Failover behavior

If an HA failover occurs:

The newly elected primary unit already holds the most up to date DHCP lease database. This prevents issues such as:

- Incorrect IP address assignments
- Duplicate lease allocations
- Lost or stale lease information

Cluster formation and lease database merging

When an HA cluster is first formed:

- Any unit joining as a subordinate might already contain an existing DHCP lease database from prior standalone operation.
- To ensure consistency across the cluster, all DHCP lease databases from all units are merged into a single unified database. This consolidated database is then synchronized to every cluster unit.

Monitoring VLAN interfaces

FortiGate HA clusters that use VLAN interfaces can take advantage of VLAN heartbeat monitoring to ensure VLAN connectivity is functioning correctly across all units. When this feature is enabled, the primary unit actively verifies communication with each subordinate unit over every configured VLAN interface. This mechanism also confirms that the upstream switch is correctly configured for those VLANs.

If the primary unit cannot reach a subordinate unit on a specific VLAN, the system records a link-monitor event (log ID 20099) indicating that the corresponding VLAN interface has gone down.

Use the following CLI command to enable monitoring VLAN interfaces:

```
config system ha-monitor
    set monitor-vlan {enable | disable}
    set vlan-hb-interval <interval_seconds>
    set vlan-hb-lost-threshold <vlan-lost-heartbeat-threshold>
end
```

Command	Description
`monitor-vlan {enable | disable}`	Enable/disable monitor VLAN interfaces (default = disable).
`vlan-hb-interval <interval_seconds>`	The VLAN heartbeat interval defines how often heartbeat packets are sent (1 - 30 seconds, default = 5 seconds).
`vlan-hb-lost-threshold <vlan-lost-heartbeat-threshold>`	The threshold is the number of consecutive VLAN heartbeat packets that must be missed before the VLAN is considered down (1 - 60 packets, default = 3). For example, if 3 heartbeat packets are not received, the VLAN is marked as down.

With a `vlan-hb-interval` of 5 seconds and a `vlan-hb-lost-threshold` of 3, the system detects a VLAN failure after 5 × 3 = 15 seconds.

To check VLAN heartbeat monitor status:

Run the following command to view the HA VLAN heartbeat monitor status:

```
# diagnose sys ha vlan-hb-monitor
```

Example output:

```
2026-03-12 11:03:44 peer #1: serial_no: FGTSERIALNUMBER123, last_hb: 0 sec ago, active
2026-03-12 11:03:44 vlan_intf #2: dmz1      <-- #<VLAN>: interface
2026-03-12 11:03:44 peer #1: serial_no: FGTSERIALNUMBER123, last_hb: 0 sec ago, active
2026-03-12 11:03:44 vlan_intf #3: internal  <-- #<VLAN>: interface
2026-03-12 11:03:44 peer #1: serial_no: FGTSERIALNUMBER123, last_hb: 0 sec ago, active
...
2026-03-12 11:03:44 peer #1: serial_no: FGTSERIALNUMBER123, last_hb: 4219777 sec ago, not active
2026-03-12 11:03:44 vlan_intf #9: servers   <-- #<VLAN>: interface
```

To monitor VLAN heartbeat messages from the HA monitor daemon in real time:

```
# diagnose debug application hamonitord -1
```

Example error message:

```
# 2026-03-12 11:11:26 [vlan_hb_timer_func:139] sendto() failed: 100(Network is down)
```

Troubleshooting

Ensuring the health and consistency of an HA cluster is essential for maintaining network uptime. When issues arise, prompt identification of formation errors, synchronization discrepancies, or failover behaviors is required to restore redundancy.

This section covers the necessary steps and commands for diagnosing and resolving common HA issues:

Topic	Summary
Troubleshoot an HA formation	Identifies the strict hardware and firmware prerequisites required for a cluster to form. This section also details how to diagnose and resolve critical situations.
Force HA failover for testing and demonstrations	Explains how to use the execute ha failover command to manually trigger a failover. This is useful for validating redundancy during maintenance windows or demonstrating failover capabilities to auditors without physically disconnecting cables.
Check HA synchronization status	Describes methods to verify that the configuration is identical across all cluster members. It covers how to interpret checksum mismatches in the GUI and CLI to identify specific configuration tables that are out of sync.
Replacing a failed cluster unit	Provide instructions for completing an RMA transfer in the FortiCloud portal.

Troubleshoot an HA formation

The following are requirements for setting up an HA cluster or FGSP peers.

Cluster members must have:

- The same model.
- The same hardware configuration.

- The same connections.
- The same generation.

> The requirement to have the same generation is done as a best practice as it avoids issues that can occur later on. If you are unsure if the FortiGates are from the same generation, please contact customer service.

Troubleshooting common HA formation errors

One member keeps shutting down during HA setup (hard drive failure)

If one member has a hard drive failure but the other does not, the one with the hard drive failure will be shut down during HA setup. In this case, RMA the member to resolve the issue.

Split brain scenario

A split brain scenario occurs when two or more members of a cluster cannot communicate with each other on the heartbeat interface, causing each member to think it is the primary. As a result, each member assumes the primary HA role and applies the same IP and virtual MAC addresses on its interfaces. This causes IP and MAC conflicts on the network, and causes flapping on L2 devices when they learn the same MAC address on ports connected to different FortiGates.

A split brain scenario is usually caused by a complete lost of the heartbeat link or links. This can be a physical connectivity issue, or less commonly, something blocking the heartbeat packets between the HA members. Another cause is congestion and latency in the heartbeat links that exceeds the heartbeat lost intervals and thresholds.

The following are common symptoms of a split brain scenario:

- The connections to the FortiGates in the cluster work intermittently when trying to connect with administrative access.
- Sessions cannot be established through the FortiGate, and the traffic drops.
- When logging in to the FortiGates using the console, `get system ha status` shows each FortiGate as the primary.

To resolve a split brain scenario:

- Be physically on-site with the FortiGates (recommended). If this is not possible, connect to the FortiGates using console access.
- Identify the heartbeat ports, and verify that they are physically connected and up.
- Verify that heartbeat packets are being sent and received on the heartbeat ports.
- Verify that the HA configurations match between the HA members. The HA `mode`, `group-name`, `group-id`, and `password` settings should be the same. Different `group-id` values will result in different virtual MAC addresses, which might not cause a MAC conflict. However, an IP conflict can still occur.
- If everything seems to be in working order, run `get system ha status` to verify that HA has formed successfully.

To avoid a split brain scenario:

- In a two-member HA configuration, use back-to-back links for heartbeat interface instead of connecting through a switch.

- Use redundant HA heartbeat interfaces.
- In a configuration where members are in different locations, ensure the heartbeat lost intervals and thresholds are longer than the possible latency in the links.

Force HA failover for testing and demonstrations

> ⚠ This command should only be used for testing, troubleshooting, maintenance, and demonstrations. Do not use it in a live production environment outside of an active maintenance window.

HA failover can be forced on an HA primary device. The device will stay in a failover state (secondary) regardless of the conditions. The only way to remove the failover status is by manually turning it off.

Syntax

```
execute ha failover set <cluster_id>
execute ha failover unset <cluster_id>
```

Variable	Description
<cluster_id>	The cluster ID is 1 for any cluster that is not in virtual cluster mode, and can be 1 or 2 if virtual cluster mode is enabled.

Example

To manually force an HA failover:

```
# execute ha failover set 1
Caution: This command will trigger an HA failover.
It is intended for testing purposes.
Do you want to continue? (y/n)y
```

To view the failover status:

```
# execute ha failover status
failover status: set
```

To view the system status of a device in forced HA failover:

```
# get system ha status
HA Health Status: OK
Model: FortiGate-300D
Group Name:
Group ID: 240
Debug: 0
Cluster Uptime: 0 days 2:11:46
Cluster state change time: 2020-03-12 17:38:04
```

```
Primary selected using:
    <2020/03/12 17:38:04> FGT3HD3914800153 is selected as the primary because EXE_FAIL_OVER flag
is set on peer member FGT3HD3914800069.
    <2020/03/12 15:27:26> FGT3HD3914800069 is selected as the primary because it has the largest
value of override priority.
ses_pickup: disable
override: enable
Configuration Status:
    FGT3HD3914800069(updated 4 seconds ago): in-sync
    FGT3HD3914800153(updated 3 seconds ago): in-sync
System Usage stats:
    FGT3HD3914800069(updated 4 seconds ago):
        sessions=5, average-cpu-user/nice/system/idle=0%/0%/0%/100%, memory=30%
    FGT3HD3914800153(updated 3 seconds ago):
        sessions=41, average-cpu-user/nice/system/idle=0%/0%/0%/99%, memory=30%
HBDEV stats:
    FGT3HD3914800069(updated 4 seconds ago):
        port3: physical/1000auto, up, rx-bytes/packets/dropped/errors=15914162/42929/0/0,
tx=15681840/39505/0/0
        port5: physical/1000auto, up, rx-bytes/packets/dropped/errors=17670346/52854/0/0,
tx=20198409/54692/0/0
    FGT3HD3914800153(updated 3 seconds ago):
        port3: physical/1000auto, up, rx-bytes/packets/dropped/errors=16636700/45544/0/0,
tx=15529791/39512/0/0
        port5: physical/1000auto, up, rx-bytes/packets/dropped/errors=20199928/54699/0/0,
tx=17672146/52862/0/0
Secondary: FortiGate-300D  , FGT3HD3914800069, HA cluster index = 1
Primary: FortiGate-300D  , FGT3HD3914800153, HA cluster index = 0
number of vcluster: 1
vcluster 1: standby 169.254.0.1
Secondary: FGT3HD3914800069, HA operating index = 1
Primary: FGT3HD3914800153, HA operating index = 0
```

To stop the failover status:

```
# execute ha failover unset 1
Caution: This command may trigger an HA failover.
It is intended for testing purposes.
Do you want to continue? (y/n)y
```

To view the system status of a device after forced HA failover is disabled:

```
# get system ha status
HA Health Status: OK
Model: FortiGate-300D
Mode: HA A-P
Group Name:
Group ID: 240
Debug: 0
Cluster Uptime: 0 days 2:14:55
Cluster state change time: 2020-03-12 17:42:17
```

```
Primary selected using:
    <2020/03/12 17:42:17> FGT3HD3914800069 is selected as the primary because it has the largest
value of override priority.
    <2020/03/12 17:38:04> FGT3HD3914800153 is selected as the primary because EXE_FAIL_OVER flag
is set on peer member FGT3HD3914800069.
    <2020/03/12 15:27:26> FGT3HD3914800069 is selected as the primary because it has the largest
value of override priority.
ses_pickup: disable
override: enable
Configuration Status:
    FGT3HD3914800069(updated 3 seconds ago): in-sync
    FGT3HD3914800153(updated 2 seconds ago): in-sync
System Usage stats:
    FGT3HD3914800069(updated 3 seconds ago):
        sessions=0, average-cpu-user/nice/system/idle=0%/0%/0%/100%, memory=30%
    FGT3HD3914800153(updated 2 seconds ago):
        sessions=38, average-cpu-user/nice/system/idle=0%/0%/0%/100%, memory=30%
HBDEV stats:
    FGT3HD3914800069(updated 3 seconds ago):
        port3: physical/1000auto, up, rx-bytes/packets/dropped/errors=16302442/43964/0/0,
tx=16053848/40454/0/0
        port5: physical/1000auto, up, rx-bytes/packets/dropped/errors=18161941/54088/0/0,
tx=20615650/55877/0/0
    FGT3HD3914800153(updated 2 seconds ago):
        port3: physical/1000auto, up, rx-bytes/packets/dropped/errors=17033009/46641/0/0,
tx=15907891/40462/0/0
        port5: physical/1000auto, up, rx-bytes/packets/dropped/errors=20617180/55881/0/0,
tx=18163135/54091/0/0
Primary: FortiGate-300D  , FGT3HD3914800069, HA cluster index = 1
Secondary: FortiGate-300D  , FGT3HD3914800153, HA cluster index = 0
number of vcluster: 1
vcluster 1: work 169.254.0.2
Primary: FGT3HD3914800069, HA operating index = 0
Secondary: FGT3HD3914800153, HA operating index = 1
```

Check HA synchronization status

The HA synchronization status can be viewed in the GUI through either a widget on the *Dashboard* or on the *System > HA* page. It can also be confirmed through the CLI.

When a cluster is out of synchronization, configuration changes may not propagate, leading to potential management and failover inconsistencies that affect configuration integrity.

When units are out of synchronization in an HA cluster, the GUI will compare the HA checksums and display the tables that caused HA to be out of synchronization. This can be visualized on the HA monitor page and in the HA status widget.

Initial verification

Before performing advanced troubleshooting, ensure that both cluster members are running the identical firmware version using the following command on both units:

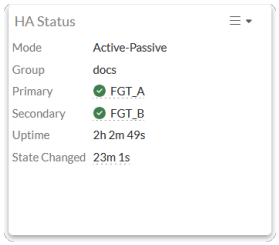
```
get system status
```

To access the secondary unit's CLI for verification, use the `execute ha manage <id> <username>` command from the primary unit.

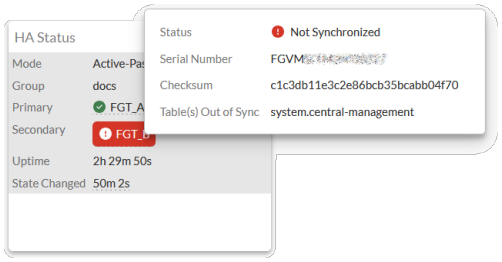
HA synchronization status in the GUI

Following HA setup, the *HA Status* widget can be added to the *Dashboard*. It shows the HA synchronization statuses of the members.

A green checkmark is shown next to each member that is in synchronization.

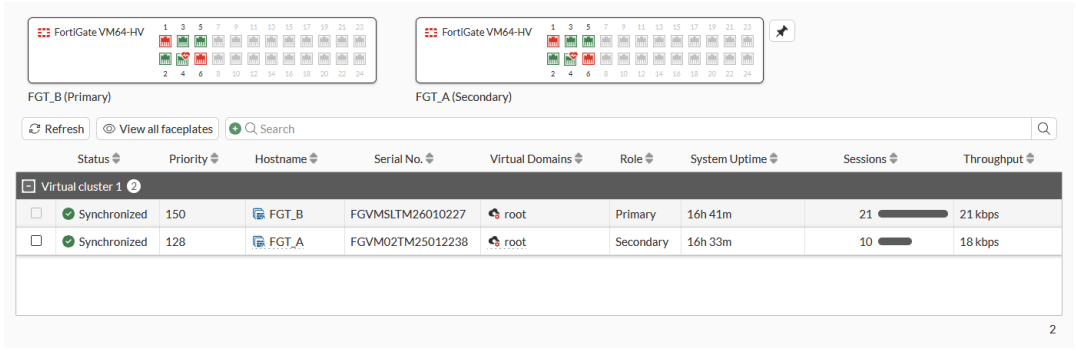


A member that is out of synchronization is highlighted in red. Hover the cursor over the *Not Synchronized* status to see a pop-up information panel showing the HA checksums and specific tables and objects that caused the cluster to the be out of synchronization.

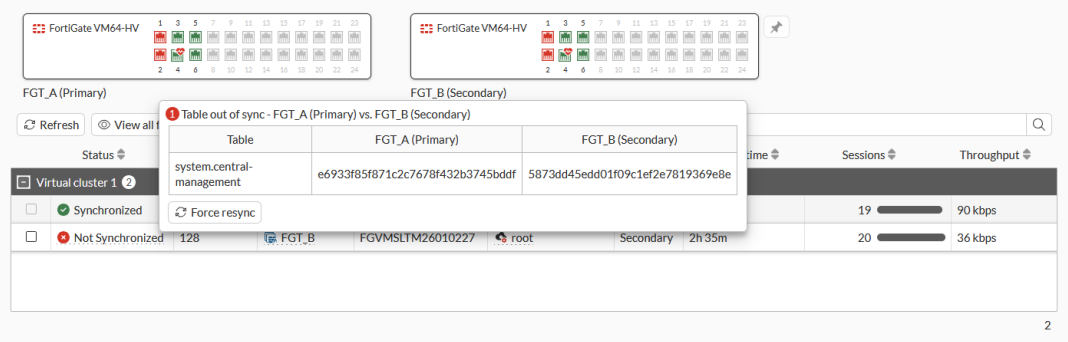


You can also go to *System > HA* to see the synchronization statuses of the members. A member that is out of synchronization will have a red icon next to its name. Hover the cursor over the not synchronized device to see the tables that are out of synchronization and the checksum values.

Synchronized:



Unsynchronized:



Click *Force resync* in the pop-up to force the not synchronized unit to try to synchronize with the Primary unit.

HA synchronization status in the CLI

In the CLI, run the `get system ha status` command to see if the cluster is in synchronization . The synchronization status is reported under `Configuration Status`.

Key output sections to monitor:

- `Primary selected using`: Indicates why the current primary was elected (such as uptime or priority).
- `Configuration Status`: Shows whether each serial number is in-sync or out-of-sync. It also displays the checksum dump; if these strings do not match between members, they are out of synchronization.
- `HBDEV / MONDEV stats`: Verifies that heartbeat and monitoring interfaces are up and passing traffic.

When both members are in synchronization:

```
# get system ha status
HA Health Status: OK
Model: FortiGate-VM64-HV
Mode: HA A-P
Group Name: docs
Group ID: 7
Debug: 0
Cluster Uptime: 0 days 16h:42m:32s
Cluster state change time: 2026-04-10 07:19:26
Primary selected using:
    <2026/04/10 07:19:26> vcluster-1: FGVMXXXX00000007 is selected as the primary because its
override priority is larger than peer member FGVMXXXX00000008.
    <2026/04/09 08:10:09> vcluster-1: FGVMXXXX00000007 is selected as the primary because it's the
only member in the cluster.
    <2026/04/08 12:21:00> vcluster-1: FGVMXXXX00000008 is selected as the primary because its
override priority is larger than peer member FGVMXXXX00000007.
    <2026/04/08 12:21:00> vcluster-1: FGVMXXXX00000007 is selected as the primary because it's the
only member in the cluster.
ses_pickup: disable
override: disable
unicast_hb: peerip=172.24.168.8, myip=172.24.168.8, hasync_port='port4'
Configuration Status:
    FGVMXXXX00000007(updated 5 seconds ago): in-sync
    FGVMXXXX00000007 chksum dump: 93 e1 51 4f f4 05 09 d6 ff 31 41 26 66 55 57 74
    FGVMXXXX00000008(updated 90237 seconds ago): out-of-sync
```

```
    FGVMXXXX00000008 chksum dump: 82 9b 70 77 88 01 e7 0a 21 ba c5 72 fc 9f 6d da
System Usage stats:
    FGVMXXXX00000007(updated 5 seconds ago):
        sessions=10, average-cpu-user/nice/system/idle=1%/0%/2%/96%, memory=70%
    FGVMXXXX00000008(updated 90237 seconds ago):
        sessions=13, average-cpu-user/nice/system/idle=1%/0%/0%/97%, memory=76%
HBDEV stats:
    FGVMXXXX00000007(updated 5 seconds ago):
        port4: physical/10000full, up, rx-bytes/packets/dropped/errors=105722696/253867/0/0,
tx=94453091/232594/0/0
    FGVMXXXX00000008(updated 90237 seconds ago):
        port4: physical/10000full, up, rx-bytes/packets/dropped/errors=56255256/179632/0/0,
tx=88333718/182761/0/0
number of member: 2
FGT_B           , FGVMXXXX00000007, HA cluster index = 0
FGT_A           , FGVMXXXX00000008, HA cluster index = 1
number of vcluster: 1
vcluster 1: work 172.24.168.8
Primary: FGVMXXXX00000007, HA operating index = 0
Secondary: FGVMXXXX00000008, HA operating index = 1
```

When one of the members is out of synchronization:

```
# get system ha status
HA Health Status: OK
Model: FortiGate-VM64-HV
Mode: HA A-P
Group Name: docs
Group ID: 7
Debug: 0
Cluster Uptime: 0 days 16h:30m:41s
Cluster state change time: 2026-04-10 07:19:26
Primary selected using:
    <2026/04/10 07:19:26> vcluster-1: FGVMXXXX00000007 is selected as the primary because its
override priority is larger than peer member FGVMXXXX00000008.
    <2026/04/09 08:10:09> vcluster-1: FGVMXXXX00000007 is selected as the primary because it's the
only member in the cluster.
    <2026/04/08 12:21:00> vcluster-1: FGVMXXXX00000008 is selected as the primary because its
override priority is larger than peer member FGVMXXXX00000007.
    <2026/04/08 12:21:00> vcluster-1: FGVMXXXX00000007 is selected as the primary because it's the
only member in the cluster.
ses_pickup: disable
override: disable
unicast_hb: peerip=172.24.168.8, myip=172.24.168.8, hasync_port='port4'
Configuration Status:
    FGVMXXXX00000007(updated 4 seconds ago): in-sync
    FGVMXXXX00000007 chksum dump: 93 e1 51 4f f4 05 09 d6 ff 31 41 26 66 55 57 74
    FGVMXXXX00000008(updated 89526 seconds ago): out-of-sync
    FGVMXXXX00000008 chksum dump: 82 9b 70 77 88 01 e7 0a 21 ba c5 72 fc 9f 6d da
System Usage stats:
    FGVMXXXX00000007(updated 4 seconds ago):
        sessions=10, average-cpu-user/nice/system/idle=0%/0%/0%/100%, memory=70%
    FGVMXXXX00000008(updated 89526 seconds ago):
```

```
        sessions=13, average-cpu-user/nice/system/idle=1%/0%/0%/97%, memory=76%
HBDEV stats:
    FGVMXXXX00000007(updated 4 seconds ago):
        port4: physical/10000full, up, rx-bytes/packets/dropped/errors=103951517/248888/0/0,
tx=92482453/228316/0/0
    FGVMXXXX00000008(updated 89526 seconds ago):
        port4: physical/10000full, up, rx-bytes/packets/dropped/errors=56255256/179632/0/0,
tx=88333718/182761/0/0
number of member: 2
FGT_B           , FGVMXXXX00000007, HA cluster index = 0
FGT_A           , FGVMXXXX00000008, HA cluster index = 1
number of vcluster: 1
vcluster 1: work 172.24.168.8
Primary: FGVMXXXX00000007, HA operating index = 0
Secondary: FGVMXXXX00000008, HA operating index = 1
```

Filtering the synchronization status by virtual cluster ID

The HA status information can be filtered for a specific virtual cluster in the CLI. The following example HA configuration has 10 virtual clusters, and the status information is filtered for virtual cluster 8.

To view the HA synchronization status of the entire cluster:

```
# get system ha status
HA Health Status:
    WARNING: FG101FTK19000007 has hbdev down;
    WARNING: FG101FTK19000009 has hbdev down;
Model: FortiGate-101F
Mode: HA A-P
Group Name: FGDocs
Group ID: 5
Debug: 0
Cluster Uptime: 0 days 0:8:29
Cluster state change time: 2023-05-24 15:00:56
Primary selected using:
  virtual cluster 1:
<2023/05/24 14:53:11> vcluster-1: FG101FTK19000007 is selected as the primary because its override
priority is larger than peer member FG101FTK19000009.
  virtual cluster 2:
<2023/05/24 14:58:49> vcluster-2: FG101FTK19000007 is selected as the primary because its override
priority is larger than peer member FG101FTK19000009.
  virtual cluster 3:
<2023/05/24 14:58:49> vcluster-3: FG101FTK19000007 is selected as the primary because its override
priority is larger than peer member FG101FTK19000009.
  ...
  virtual cluster 10:
<2023/05/24 15:00:56> vcluster-10: FG101FTK19000009 is selected as the primary because its
override priority is larger than peer member FG101FTK19000007.
<2023/05/24 15:00:55> vcluster-10: FG101FTK19000007 is selected as the primary because its
override priority is larger than peer member FG101FTK19000009.
ses_pickup: disable
```

```
override:
    vcluster_1 disable
    vcluster_2 disable
    vcluster_3 disable
    ...
    vcluster_10 disable
Configuration Status:
    FG101FTK19000007(updated 2 seconds ago): in-sync
    FG101FTK19000007 chksum dump: bb 79 03 7c bf 28 16 dc 14 99 23 9f 53 94 1a f0
    FG101FTK19000009(updated 1 seconds ago): out-of-sync
    FG101FTK19000009 chksum dump: 92 24 8c 43 db ed a3 f2 6f 6c cc 06 56 f5 0f 76
System Usage stats:
    FG101FTK19000007(updated 2 seconds ago):
        sessions=11, average-cpu-user/nice/system/idle=0%/0%/0%/99%, memory=33%
    FG101FTK19000009(updated 1 seconds ago):
        sessions=0, average-cpu-user/nice/system/idle=0%/0%/0%/99%, memory=32%
HBDEV stats:
    FG101FTK19000007(updated 2 seconds ago):
        ha1: physical/1000auto, up, rx-bytes/packets/dropped/errors=8490968/16615/0/0,
tx=3116622/7352/0/0
        ha2: physical/00, down, rx-bytes/packets/dropped/errors=0/0/0/0, tx=0/0/0/0
    FG101FTK19000009(updated 1 seconds ago):
        ha1: physical/1000auto, up, rx-bytes/packets/dropped/errors=7359328/16036/0/0,
tx=4369340/8183/0/0
        ha2: physical/00, down, rx-bytes/packets/dropped/errors=0/0/0/0, tx=0/0/0/0
Primary     : FortiGate-101F  , FG101FTK19000007, HA cluster index = 0
Secondary   : FortiGate-101F  , FG101FTK19000009, HA cluster index = 1
number of vcluster: 10
vcluster 1: work 169.254.0.1
Primary: FG101FTK19000007, HA operating index = 0
Secondary: FG101FTK19000009, HA operating index = 1
vcluster 2: work 169.254.0.1
Primary: FG101FTK19000007, HA operating index = 0
Secondary: FG101FTK19000009, HA operating index = 1
vcluster 3: work 169.254.0.1
Primary: FG101FTK19000007, HA operating index = 0
Secondary: FG101FTK19000009, HA operating index = 1
...
vcluster 10: standby 169.254.0.2
Secondary: FG101FTK19000007, HA operating index = 1
Primary: FG101FTK19000009, HA operating index = 0
```

To view the HA synchronization status of virtual cluster 8:

```
# get system ha status 8
HA Health Status:
    WARNING: FG101FTK19000007 has hbdev down;
    WARNING: FG101FTK19000009 has hbdev down;
Model: FortiGate-101F
Mode: HA A-P
Group Name: FGDocs
Group ID: 5
```

```
Debug: 0
Cluster Uptime: 0 days 0:8:48
Cluster state change time: 2023-05-24 15:00:56
Primary selected using:
  virtual cluster 8:
<2023/05/24 15:00:55> vcluster-8: FG101FTK19000007 is selected as the primary because its override
priority is larger than peer member FG101FTK19000009.
ses_pickup: disable
override:
    vcluster_8 disable
Configuration Status:
    FG101FTK19000007(updated 2 seconds ago): in-sync
    FG101FTK19000007 chksum dump: bb 79 03 7c bf 28 16 dc 14 99 23 9f 53 94 1a f0
    FG101FTK19000009(updated 1 seconds ago): out-of-sync
    FG101FTK19000009 chksum dump: 92 24 8c 43 db ed a3 f2 6f 6c cc 06 56 f5 0f 76
System Usage stats:
    FG101FTK19000007(updated 2 seconds ago):
        sessions=11, average-cpu-user/nice/system/idle=0%/0%/0%/99%, memory=33%
    FG101FTK19000009(updated 1 seconds ago):
        sessions=0, average-cpu-user/nice/system/idle=0%/0%/0%/99%, memory=32%
HBDEV stats:
    FG101FTK19000007(updated 2 seconds ago):
        ha1: physical/1000auto, up, rx-bytes/packets/dropped/errors=8721683/17144/0/0,
tx=3214512/7536/0/0
        ha2: physical/00, down, rx-bytes/packets/dropped/errors=0/0/0/0, tx=0/0/0/0
    FG101FTK19000009(updated 1 seconds ago):
        ha1: physical/1000auto, up, rx-bytes/packets/dropped/errors=7616471/16547/0/0,
tx=4440283/8383/0/0
        ha2: physical/00, down, rx-bytes/packets/dropped/errors=0/0/0/0, tx=0/0/0/0
Primary     : FortiGate-101F  , FG101FTK19000007, HA cluster index = 0
Secondary   : FortiGate-101F  , FG101FTK19000009, HA cluster index = 1
number of vcluster: 10
vcluster 8: work 169.254.0.1
Primary: FG101FTK19000007, HA operating index = 0
Secondary: FG101FTK19000009, HA operating index = 1
```

Advanced troubleshooting and hardware checks

In some cases, synchronization fails due to environmental factors or local database mismatches.

- ISDB Versions: Ensure the Internet Service Database (ISDB) versions match. Update the unit that has the lower version if they are different.
- Disk Matching: Use the `diagnose hardware device disk` command to verify that disk sizes and statuses match. If a disk shows `Need format` the unit must be removed from the cluster and the disk formatted.
- Config Error Log: Run the `diagnose debug config-error-log read` command to see if specific commands were rejected by the secondary unit during the synchronization process.
- Certificate Bundles: Always verify the Certificate Bundle versions using the `diagnose autoupdate versions | grep Certificate -A6` command. Mismatched CA bundles are a common cause of invisible synchronization failures.

Hard reset of secondary unit

If software troubleshooting fails, re-imaging the secondary unit with a primary unit backup is the most reliable recovery method:

1. Backup: Take a global configuration backup from the primary unit.
2. Isolate: Disconnect all data and heartbeat cables from the secondary unit.
3. Restore: Upload the primary unit's backup to the secondary unit using a console connection.
4. Modify: Change the hostname and lower the HA priority on the secondary unit to ensure that it does not trigger a failover upon reconnection. If a Reserved Management Interface is used, you must also update its unique IP address (and gateway if on a different subnet) to avoid conflicts and maintain out-of-band management.
5. Reconnect: Connect heartbeat cables first, verify synchronization with the `get system ha status` command, and then reconnect data cables.

Replacing a failed cluster unit

This procedure describes how to safely remove a failed unit from a FortiGate HA cluster and integrate a replacement (or a repaired) unit. These steps ensure the cluster remains stable and the new unit does not inadvertently take over as the primary unit before it is ready. This process does not interrupt cluster operations unless network cabling changes are required to accommodate the new hardware.

Prerequisites for the replacement unit

The replacement or repaired unit must meet these strict criteria to successfully join the cluster:

- Hardware consistency: You must maintain identical hardware configurations across all cluster members, including the same model, generation, RAM, and number of hard disks/SSDs.
- Firmware build: The replacement must run the exact same firmware build as the active cluster.
- Licensing: All units must have the same level of licensing.
- Operating mode: The unit must be in the same mode (NAT or Transparent) as the cluster.
- VDOM mode: The unit must be in single VDOM mode initially.
- Certificates: You may install third-party certificates on the primary unit before forming the cluster; these will synchronize to the replacement unit once it joins.

Step 1: Prepare for removal (failover check)

Before physically disconnecting the failed unit, you must confirm the status of the cluster:

- Check cluster health: Ensure the remaining unit is healthy and has taken over all traffic.
- Manual failover: If the "failed" unit is still partially responsive or acting as the primary unit, manually trigger a failover (by adjusting HA priority or disconnecting its heartbeat interface) to ensure the healthy unit is the primary before you remove the hardware. This prevents traffic drops during the physical swap.
- Physical disconnection: Once the healthy unit is confirmed as primary, disconnect the failed unit from both the cluster heartbeat and the network.

Step 2: RMA transfer and registration

If you are replacing a unit through RMA, you must transfer the registration and services so the new unit can access the correct licenses.

1. Log in to the FortiCloud Support portal.
2. Go to *Products* > *Product List* and select the *Serial Number* (or HA vSN) of the failed unit.
3. In the *Registration* widget, click *RMA Transfer*.
4. Enter the serial number of the new replacement unit and complete the wizard.
5. Register and apply any specific license keys to the replacement unit to match the cluster's licensing level.

Step 3: Configure the replacement unit (ensure secondary status)

To prevent the new unit from accidentally becoming the primary unit and over-writing your production config with a blank one, follow this safety-first configuration:

1. Isolate the unit: Perform the initial configuration using the Console or Management port while the unit is not connected to the network.
2. Set low priority: Set the HA priority of the replacement unit to a value significantly lower than the existing primary. (For example, set the primary to 200 and the replacement to 100.)
3. Match HA Settings: Configure the HA group ID, group name, and password to match the existing cluster.
4. Override disabled: Ensure `set override disable` is configured in the HA settings. This ensures the unit with the longest uptime (the existing primary) remains the leader.
5. Mode sync: If the cluster is in transparent mode, you must switch the replacement unit to transparent mode using the CLI/GUI before connecting it to the cluster to avoid broadcast loops or traffic interruptions.

Step 4: Connect the replacement

1. Cabling: Connect the heartbeat (HA) interfaces first.
2. Power on: Turn on the replacement unit.
3. Monitor synchronization: Use the command `get system ha status` on the primary to see the new unit join.
4. Data cables: Once the units are in sync (the *Configuration Status* shows as *in-sync*), connect the remaining data/network cables. The cluster will automatically synchronize the configuration from the primary to the replacement unit.

Best practices

FortiGate HA is designed to eliminate single points of failure in your network security infrastructure. By clustering devices using the FGCP, you ensure enhanced reliability through seamless failover and can achieve increased performance.

The following best practices focus on architectural design and operational standards to maximize cluster stability and uptime:

Architecture and topology

- **Redundant Heartbeat Design**: Design the cluster with at least two heartbeat connections to prevent "split-brain" scenarios where communication is lost and both units attempt to become the Primary.
- **Heartbeat Isolation**: Physically isolate heartbeat traffic. Ideally, connect the heartbeat interfaces directly back-to-back. If switches are required (for clusters with more than two units), use dedicated switches or VLANs strictly for heartbeat traffic to avoid network congestion affecting cluster stability.
- **Symmetrical Network Connectivity**: Ensure both cluster members have identical physical connections to the internal and external networks. This guarantees that if a failover occurs, the secondary unit has the exact same network access as the failed unit.
- **Full Mesh**: To ensure there is no single source of failure, use a full mesh design to connect your cluster members. This involves using 2 ingress and 2 egress interfaces, preferably in an aggregate or redundant interface configuration. Connect these interfaces to redundant switches.

Hardware and licensing alignment

- **Hardware Consistency**: Maintain identical hardware configurations across all cluster members, including the same model, generation, RAM, and number of hard disks/SSDs. Mismatched hardware can prevent the cluster from forming or functioning correctly.
- **Firmware Uniformity**: Ensure all units are running the exact same firmware version and build number.
- **Unified Licensing**: Register all cluster members under the same FortiCare account. This prevents potential licensing conflicts and downtime, especially regarding FortiGuard services. Also maintain the same subscriptions and expiration dates

Failover logic and monitoring

- **Strategic Interface Monitoring**: Configure the cluster to monitor only the most critical network paths (typically the internal LAN and external WAN interfaces). A failover should only be triggered if a link essential to traffic flow goes down.
- **Avoid Monitoring Heartbeats**: Do not configure the cluster to "monitor" the heartbeat interfaces themselves, as their status is handled separately by the HA protocol.
- **Consistent Override Policy**: Standardize the "Override" setting across all members. It should be either uniformly enabled (to prioritize a specific device) or disabled (to prioritize uptime) on all units to ensure predictable primary unit selection.
- **Heartbeat optimization**: Fine tune your heartbeat frequencies and intervals to failover within the timeframe that is within your expectations. If sub-second failover is required, change the heartbeat interval metric from 100ms to 10ms

Performance optimization

- **Session Synchronization Efficiency**: Use the session-pickup-delay feature to filter out short-lived sessions (like DNS or quick HTTP lookups) from being synchronized. This reduces CPU and bandwidth overhead on the heartbeat links.
- **Dedicated Session Sync Interface**: For high-traffic environments, offload session synchronization traffic from the heartbeat interface to a dedicated data interface (using session-sync-dev). This prevents heartbeat congestion during traffic spikes.
- **Heartbeat Bandwidth Management**: Avoid using heartbeat interfaces for regular traffic processing. Keeping these interfaces dedicated prevents user traffic from congesting the link and causing false

failovers.

Management and operations

- **Out-of-Band Management**: Utilize reserved management interfaces for cluster administration. This assigns a unique IP to each member, allowing for direct monitoring (via SNMP) and management of the secondary unit without affecting the primary's operation.
- **Synchronized Upgrades**: Plan for "Uninterrupted Upgrades" where the cluster manages the firmware update sequence (Secondary -> Primary) to minimize traffic disruption during maintenance windows.

Security

- **Heartbeat Encryption**: Enable authentication and encryption for heartbeat packets, especially if the heartbeat links pass through shared switches. This prevents unauthorized devices from joining the cluster or spoofing heartbeat signals.

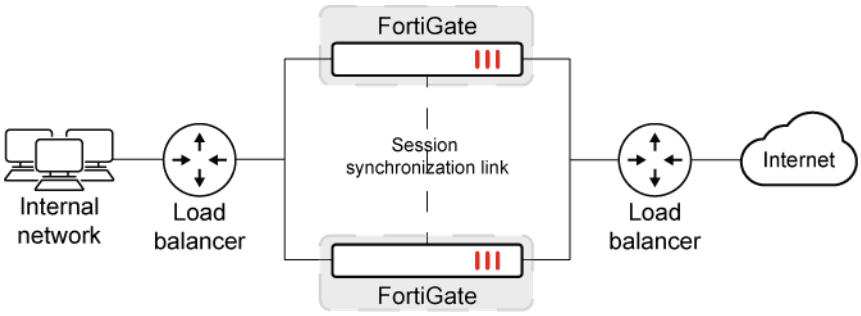
FGSP

Standalone FortiGates or FGCP clusters can be integrated into the load balancing configuration using the FortiGate Session Life Support Protocol (FGSP) in a network where traffic is load balanced by an upstream load balancer and scanned by downstream FortiGates. FGSP can perform session synchronization of IPv4 and IPv6 TCP, SCTP, UDP, ICMP, expectation, RSSO authenticated user logon information, and NAT sessions to keep the session tables synchronized on all entities. If one of the FortiGates fails, the upstream load balancer should detect the failed member and stop distributing sessions to it. Session failover occurs and active sessions fail over to the peers that are still operating. Traffic continues to flow on the new peer without data loss because the sessions are synchronized.

The FortiGates in FGSP operate as peers that process traffic and synchronize sessions. An FGSP deployment can include two to 16 standalone FortiGates, or two to 16 FortiGate FGCP clusters of two members each. Adding more FortiGates increases the CPU and memory required to keep all of the FortiGates synchronized, and it increases network synchronization traffic. Exceeding the numbers of members is not recommended and may reduce overall performance. By default, FGSP synchronizes all IPv4 and IPv6 TCP sessions, and IPsec tunnels. You can optionally add filters to control which sessions are synchronized, such as synchronizing packets from specific source and destination addresses, source and destination interfaces, or services.

FGSP is also compatible with FortiGate VRRP.

FGSP is primarily used instead of FGCP when external load balancers are part of the topology, and they are responsible for distributing traffic amongst the downstream FortiGates. FGSP provides the means to synchronize sessions between the FortiGate peers without needing a primary member to distribute the sessions like in FGCP active-active mode. If the external load balancers direct all sessions to one peer, the effect is similar to active-passive FGCP HA. If external load balancers balance traffic to both peers, the effect is similar to active-active FGCP HA. The load balancers should be configured so that all packets for any given session are processed by the same peer, including return packets whenever possible.



Session pickup

Session pickup is an optional setting that can be enabled to synchronize connectionless (UDP and ICMP) sessions, expectation sessions, and NAT sessions. If session pickup is not enabled, the FGSP does not share session tables for the particular session type, and sessions do not resume after a failover. All sessions are interrupted by the failover and must be re-established at the application level. Many protocols can successfully restart sessions with little, or no, loss of data. Others may not recover as easily. Enable session pickup for sessions that may be difficult to reestablish. Since session pickup requires FortiGate memory and CPU resources, only enable this feature for sessions that need to synchronize.

Session synchronization link

The session synchronization link is an optional configuration that allows peers to synchronize sessions over a dedicated interface instead of the interface in which the peer IP is routed. In this configuration, communications occur over L2 instead of L3. Configuring session synchronization links is recommended when you want to minimize traffic over the peering interface when there are many sessions that need to be synchronized.

Expectation sessions

FortiOS session helpers keep track of the communication of layer 7 protocols, such as FTP and SIP, that have control sessions and expectation sessions. The control sessions establish the link between the server and client, and negotiate the ports and protocols that will be used for data communications. The session helpers then create expectation sessions through the FortiGate for the ports and protocols negotiated by the control session.

The expectation sessions are the sessions that actually communicate data. For FTP, the expectation sessions transmit files being uploaded or downloaded. For SIP, the expectation sessions transmit voice and video data. Expectation sessions usually have a timeout value of 30 seconds. If the communication from the server is not initiated within 30 seconds, the expectation session times out and traffic will be denied.

By default, FGSP does not synchronize expectation sessions; if a failover occurs, the sessions will have to be restarted.

Expectation session synchronization also synchronizes full cone NAT sessions, see FGCP and FGSP synchronizes full-cone expectation sessions when session-pickup-expectation enabled on page 4125.

To synchronize expectation sessions so they continue after a failover:

```
config system ha
    set session-pickup enable
    set session-pickup-expectation enable
end
```

NAT session synchronization

NAT sessions are not synchronized by default. You can enable NAT session synchronization by entering the following command:

```
config system ha
    set session-pickup enable
    set session-pickup-nat enable
end
```

When deploying FGCP over FGSP, the `session-pickup-nat` setting is hidden and enabled by default. It cannot be modified. See FGCP over FGSP per-tunnel failover for IPsec on page 4260 for more information.

After a failover with this configuration, all sessions that include the IP addresses of interfaces on the failed FortiGate unit will have nowhere to go since the IP addresses of the failed FortiGate unit will no longer be on the network. If you want NAT sessions to resume after a failover you should not configure NAT to use the destination interface IP address, since the FGSP FortiGate units have different IP addresses. To avoid this issue, you should use IP pools with the type set to overload (which is the default IP pool type), as shown in this example:

```
config firewall ippool
    edit FGSP-pool
        set type overload
        set startip 172.20.120.10
        set endip 172.20.120.20
    next
end
```

In NAT mode, only sessions for route mode security policies are synchronized. FGSP is also available for FortiGate units or virtual domains operating in transparent mode. Only sessions for normal transparent mode policies are synchronized.

The following topics provide more information about FGSP:

- FGSP basic peer setup on page 4214
- Synchronizing sessions between FGCP clusters on page 4219
- Session synchronization interfaces in FGSP on page 4220
- UTM inspection on asymmetric traffic in FGSP on page 4223
- UTM inspection on asymmetric traffic on L3 on page 4226
- FGSP support for failover with asymmetric traffic and UTM on page 4228

- Monitor routing prefix for FGSP session failover on page 4230
- Encryption for L3 on asymmetric traffic in FGSP on page 4237
- Optimizing FGSP session synchronization and redundancy on page 4238
- Firmware upgrades in FGSP on page 4243
- FGSP session synchronization between different FortiGate models or firmware versions on page 4244
- Applying the session synchronization filter only between FGSP peers in an FGCP over FGSP topology on page 4245
- FGSP static site-to-site IPsec VPN setup on page 4252
- FGSP per-tunnel failover for IPsec on page 4254
- FGCP over FGSP per-tunnel failover for IPsec on page 4260
- Allow IPsec DPD in FGSP members to support failovers on page 4270

FGSP basic peer setup

The FortiGate Session Life Support Protocol (FGSP) is a proprietary HA solution for only sharing sessions between entities based on peer-to-peer communications. The entities could be standalone FortiGates or an FGCP cluster. Sessions are load balanced by an upstream load balancer. Each peer will synchronize its sessions with the other peers so that if a failure occurs, sessions will continue to flow as the load balancer redirects the traffic to the other peers.

Basic requirements and limitations

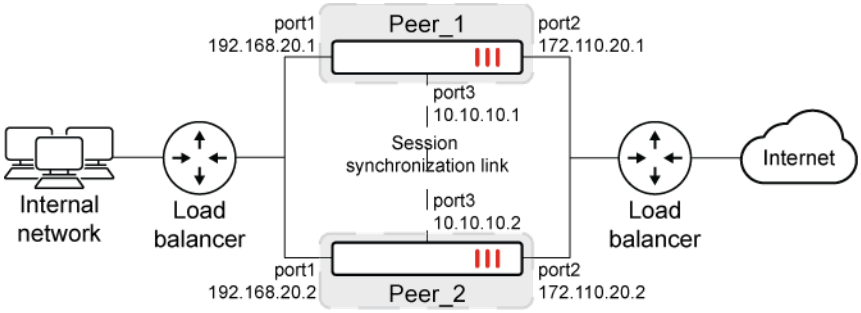
In most production environments, the following requirements should be met:

- The peers are FortiGates of the same model.
- The peers are running the same firmware version.
- There are 2 to 16 standalone FortiGates, or 2 to 16 FortiGate FGCP clusters of two members each.
- The configurations related to session tables should match. For example, the logical names used in firewall policies, IPsec interface names, VDOM names, firewall policy tables, and so on.

Two FortiGates must have similar capabilities so that data structures used in session synchronization will match, and are capable of delivering similar performance. Therefore, the same model and firmware version is highly recommended for most production deployments. The limitation on the number of FortiGates in FGSP also ensures that session synchronization will occur smoothly.

Example

This example uses two peer FortiGates. The load balancer is configured to send all sessions to Peer_1, and if Peer_1 fails, all traffic is sent to Peer_2.



To configure a basic FGSP peer setup:

These instructions assume that all FortiGates have been factory reset.

1. Make all the necessary connections as shown in the topology diagram.
2. On Peer_1, configure the peer IP in which this device will peer with:

```
config system standalone-cluster
    config cluster-peer
        edit 1
            set peerip 10.10.10.2
        next
    end
    set standalone-group-id 1
    set group-member-id 1
end
```

   If there are multiple peer IPs from the same peer, enter them as separate entries. If there are multiple peers, enter the IP of each peer in separate entries. See Optimizing FGSP session synchronization and redundancy on page 4238 for an example.

   Sessions by default will be synchronized over layer 3 on the interface in which the current unit connects to the peer's IP.
3. On Peer_2, configure session synchronization:

```
config system standalone-cluster
    config cluster-peer
        edit 1
            set peerip 10.10.10.1
        next
    end
    set standalone-group-id 1
    set group-member-id 2
end
```

4. Configure identical firewall policies on each peer, such as for traffic going from the same incoming interface (port1) to the outgoing interface (port2).

To test the FGSP peer setup:

1. Initiate TCP traffic (like HTTP access) to go through Peer_1.
2. Check the session information:

```
# diagnose sys session filter src <IP_address>
```

```
# diagnose sys session list
```

3. Enter the same commands on Peer_2 to verify if the same session information appears.

Optional filters

Filters can be added to synchronize certain types of sessions that meet the filter criteria.

To add filters for session synchronization:

```
config system standalone-cluster
    config cluster-peer
        edit <id>
            config session-sync-filter
                set srcintf <interface>
                set dstintf <interface>
                set srcaddr <IPv4_address>
                set dstaddr <IPv4_address>
                set srcaddr6 <IPv6_address>
                set dstaddr6 <IPv6_address>
            end
        next
    end
end
```

Filter examples

To synchronize only sessions with a particular source subnet:

```
config system standalone-cluster
    config cluster-peer
        edit 1
            config session-sync-filter
                set srcaddr 192.168.20.0/24
            end
        next
    end
end
```

To synchronize only sessions with a particular source address range:

```
config system standalone-cluster
    config cluster-peer
        edit 1
            config session-sync-filter
                set srcaddr 192.168.20.10 192.168.20.20
            end
        next
    end
end
```

To synchronize only sessions with a particular destination address range:

```
config system standalone-cluster
    config cluster-peer
        edit 1
            config session-sync-filter
                set dstaddr6 2001:db8:0:2::/64
            end
        next
    end
end
```

Session pickup

You can enable this setting to synchronize connectionless (UDP and ICMP) sessions, expectation sessions, and NAT sessions. If session pickup is not enabled, the FGSP does not share session tables for the particular session type, and sessions do not resume after a failover.

To enable UDP and ICMP session synchronization:

```
config system ha
    set session-pickup enable
    set session-pickup-connectionless enable
end
```

Session synchronization

You can specify interfaces used to synchronize sessions in L2 instead of L3 using the `session-sync-dev` setting. For more information about using session synchronization, see Session synchronization interfaces in FGSP on page 4220.

To configure session synchronization over redundant L2 connections:

```
config system standalone-cluster
    set session-sync-dev <interface 1> [<interface 2>] ... [<interface n>]
```

```
end
```

VDOM synchronization

When multi-VDOM mode is enabled, you can specify the peer VDOM and the synchronized VDOMs. The peer VDOM contains the session synchronization link interface on the peer unit. The synchronized VDOMs' sessions are synchronized using this session synchronization configuration.

To synchronize between VDOMs:

```
config system standalone-cluster
    config cluster-peer
        edit 1
            set peerip <IP address>
            set peervd <vdom>
            set syncvd <vdom 1> [<vdom 2>] ... [<vdom n>]
        next
    end
end
```

Configuring unique group and member ID

FGSP can function between standalone FortiGates or between FGCP clusters. In either case, the `standalone-group-id` must match between FGSP members, and the `group-member-id` must be unique for each FGCP cluster. This allows each member to actively process traffic without any conflict.

To configure FGSP peering between standalone FortiGates, follow the steps under To configure a basic FGSP peer setup.

To configure FGSP peering between different FGCP clusters:

These instructions assume Peer_1 and Peer_2 are in cluster 1, and Peer_3 and Peer_4 are in cluster 2.

1. On Peer_1, configure the first group ID:

   ```
   config system standalone-cluster
       set standalone-group-id 1
       set group-member-id 1
   end
   ```

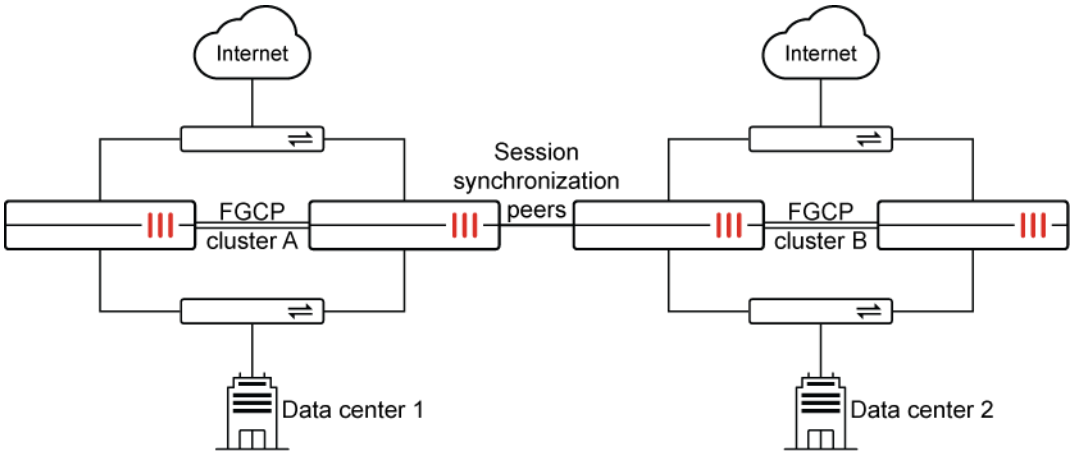
   This configuration will automatically be synchronized to Peer_2 by FGCP (cluster 1).
2. On Peer_3, configure the second group ID:

   ```
   config system standalone-cluster
       set standalone-group-id 1
       set group-member-id 2
   end
   ```

   This configuration will automatically be synchronized to Peer_4 by FGCP (cluster 2).

Synchronizing sessions between FGCP clusters

Synchronizing sessions between FGCP clusters is useful when data centers in different locations are used for load balancing, and traffic must be shared and flow freely based on demand.



There are some limitations when synchronizing sessions between FGCP clusters:

- All FortiGates must have the same model and generation, hardware configuration, and FortiOS version.
- A total of 16 clusters can share sessions.
- The configurations related to session tables should match. For example, the logical names used in firewall policies, IPsec interface names, VDOM names, firewall policy tables, and so on.

To configure session synchronization between two clusters:

1. Configure the two clusters (see HA active-passive cluster setup on page 4079 or HA active-active cluster setup on page 4081).
2. On cluster A, configure the peer IP for the interface:

```
config system interface
    edit "port5"
        set vdom "root"
        set ip 10.10.10.1 255.255.255.0
        set allowaccess ping https ssh snmp http telnet
    next
end
```

   In this example, cluster A uses port5 and its IP address, 10.10.10.1, is reachable from another cluster.
3. On cluster A, configure FGSP, including cluster and session synchronization:

```
config system standalone-cluster
    set standalone-group-id 1
    set group-member-id 0
    set session-sync-dev <interface>
    config cluster-peer
        edit 1
            set peerip 10.10.10.2
```

```
        next
    end
end
```

The `standalone-group-id` must match between FGSP members. The `group-member-id` is unique for each FGCP cluster. `session-sync-dev` is an optional command to specify the interfaces to sync sessions.

4. On cluster B, configure the peer IP for the interface:

```
config system interface
    edit "port5"
        set vdom "root"
        set ip 10.10.10.2 255.255.255.0
        set allowaccess ping https ssh snmp http telnet
    next
end
```

In this example, cluster B uses port5 and its IP address, 10.10.10.2, is reachable from another cluster.

5. On cluster B, configure FGSP, including cluster and session synchronization:

```
config system standalone-cluster
    set standalone-group-id 1
    set group-member-id 1
    set session-sync-dev <interface>
    config cluster-peer
        edit 1
            set peerip 10.10.10.1
        next
    end
end
```

Session synchronization interfaces in FGSP

When peering over FGSP, by default, the FortiGates or FGCP clusters share information over L3 between the interfaces that are configured with Peer IP addresses. When a session synchronization interface is configured and FGSP peers are directly connected on this interface, then session synchronization is done over L2, only falling back to L3 if the session synchronization interface becomes unavailable.

When FGSP peers are formed between standalone FortiGates, the session synchronization process is performed by the kernel with UDP encapsulation. When using a FGSP session synchronization interface, the synchronization process is also offloaded to the kernel, albeit more efficiently without the UDP encapsulation. Therefore, a fast, dedicated, and stable L2 connection is recommended for the session synchronization interface between the FGSP peers. For redundancy, multiple synchronization interfaces can be configured. The session synchronization interface or interfaces should always be the same on each FGSP peer.

The configurations related to session tables should match. For example, the logical names used in firewall policies, IPsec interface names, VDOM names, firewall policy tables, and so on.

To configure session-sync interfaces:

```
config system standalone-cluster
    set session-sync-dev <interface 1> [<interface 2>] ... [<interface n>]
    set layer2-connection {available | unavailable}
    set encryption {enable | disable}
end
```

The `layer2-connection` setting is for forwarded traffic between FGSP peers. Set it to `available` if the peer interface user for traffic forwarding is directly connected and supports L2 forwarding. See UTM inspection on asymmetric traffic in FGSP on page 4223 for more information.

Session synchronization in FGCP over FGSP

To provide full redundancy, FGCP clusters can be used in FGSP peering. This is called FGCP over FGSP. In these complex environments, as well as in high performance, low latency data centers, using the FGCP session synchronization interface is recommended, as it offloads the session synchronization process to the kernel. If FGCP uses session synchronization interfaces, FGSP must also be configured to use the same session synchronization interfaces.

To offload the session synchronization process to the kernel and synchronize sessions using connected interfaces directly:

```
config system ha
    set session-sync-dev <interface 1> [<interface 2>] ... [<interface n>]
end
```

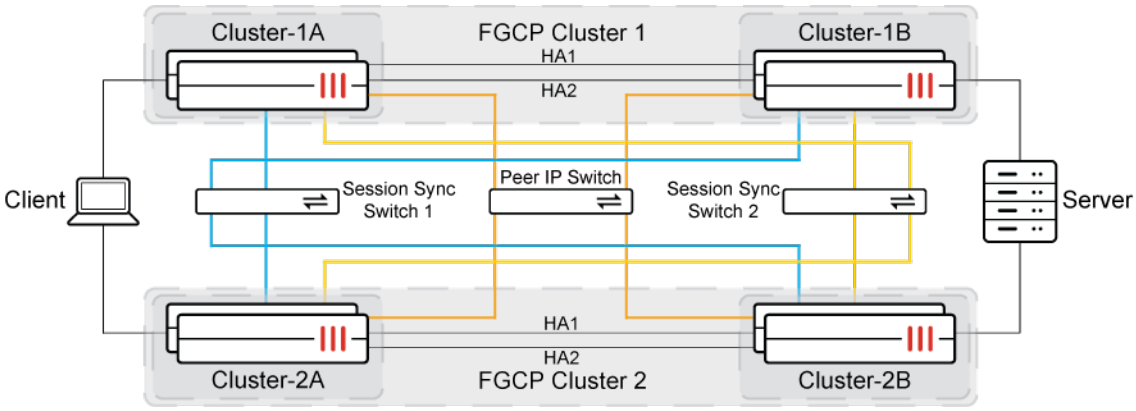
This is optimal when any of the following conditions apply:

- The session rate is high.
- Low network latency is required.
- There is a complex environment with FGCP clusters synchronizing over FGSP using L3 only in the presence of asymmetric traffic.

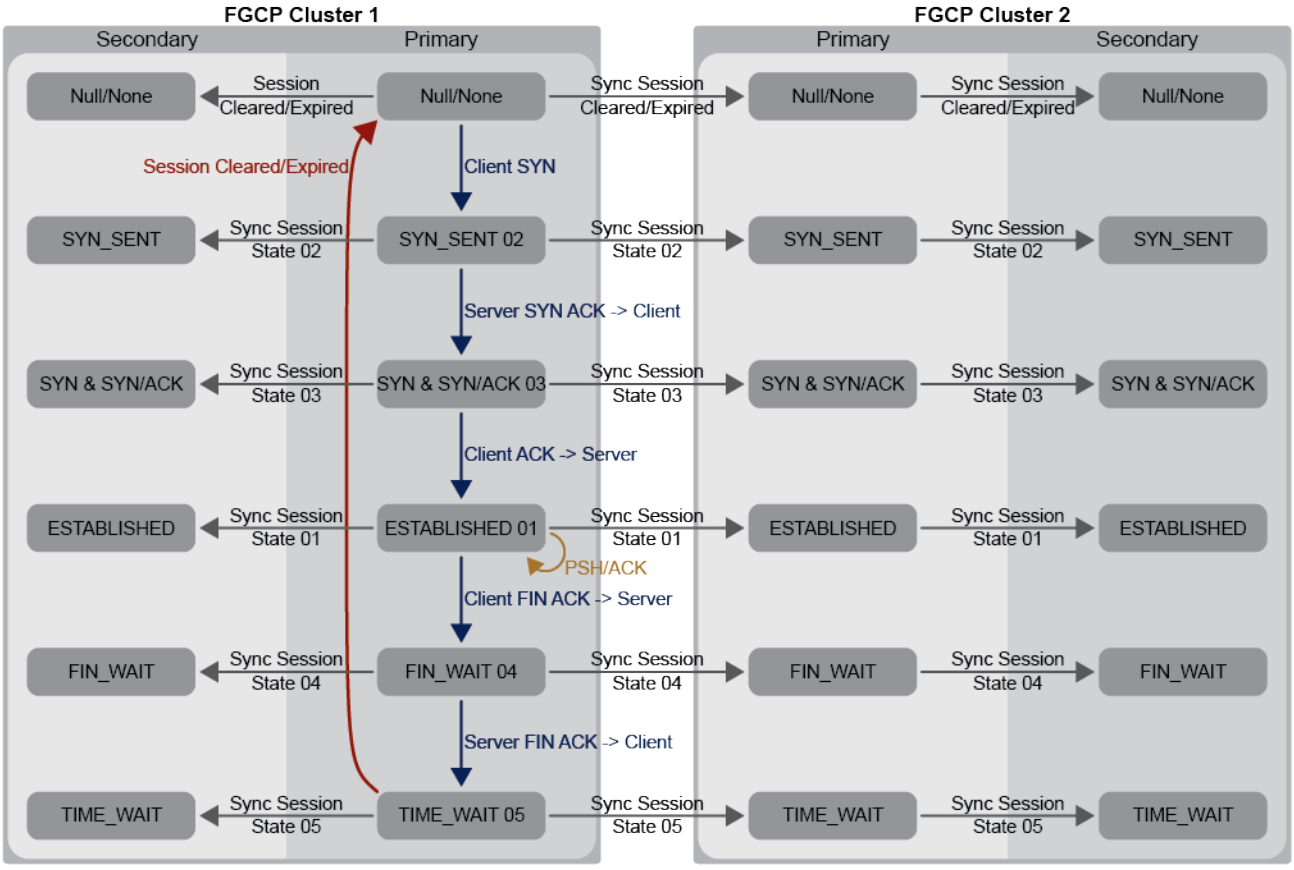
Configuring the FGCP `session-sync-dev` effectively offloads the session synchronization from the sessionsync daemon into the kernel and reduces the session synchronization latency.

When configuring both `system.standalone-cluster.session-sync-dev` and `system.ha.session-sync-dev` in FGCP over FGSP mode, the interfaces configured should be the same.

The following topology uses multiple session synchronization interfaces with a full mesh backbone to prevent any single point of failure.



The state diagram summarizes the session synchronization of a TCP session. It assumes that the session is connected over FGCP Cluster 1 and processed entirely by the primary unit, Cluster-1A.



1. The session starts with the Client SYN packet.
2. As the session is established, Cluster-1A synchronizes the session with Cluster-1B over the heartbeat interface, and with Cluster-2A over the session synchronization interface.
3. Cluster-2A then synchronizes the session with Cluster-2B over its heartbeat interface.
4. The process then repeats as it transitions to different states.

Session synchronization if links fail

In the previous topology, if any single session synchronization link fails on the primary member of each cluster, session synchronization will continue on the second link from the pair of session of session synchronization interfaces.

If the second link on the primary member of the same cluster then fails, L2 session synchronization over the session synchronization interface stops, and synchronization fails over to L3 between the peer IP links.

If the Peer IP link then fails, the FGSP peers are effectively disconnected, and no session synchronization will occur.

Fully synchronized sessions before bringing up interface

When a FGSP peer is brought up, it may be desirable to synchronize the sessions from peers before starting to process and forward traffic on the interfaces. The following command allows the named interfaces to be brought down during boot-up before existing sessions from peers are synchronized. If the sessionsync process does not start within 60 seconds of boot-up, then a timeout occurs and the interfaces are brought up.

```
config system standalone-cluster
    config cluster-peer
        edit <sync-id>
            set down-intfs-before-sess-sync <name1>, <name2>, ...
        next
    end
end
```

FGSP has protocols running between members to make sure that all existing sessions are synchronized at boot up. During the time when the interfaces are brought down, newly created sessions on remote FGSP members will also be synchronized in real-time.

Note that in the situation where FGSP peering is down, FGSP is considered broken. When FGSP peering comes up again, sessions will be automatically synchronized again but the interfaces will not be brought down during synchronization.

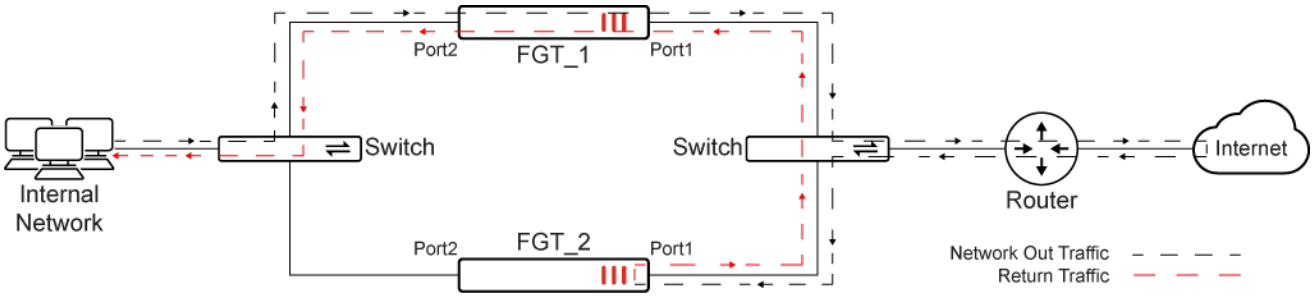
UTM inspection on asymmetric traffic in FGSP

When traffic passes asymmetrically through FGSP peers, UTM inspection can be supported by always forwarding traffic back to the session owner for processing. The session owner is the FortiGate that receives the first packet of the session.

In this example, traffic from the internal network first hits FGT_1, but the return traffic is routed to FGT_2. Consequently, traffic bounces from FGT_2 port1 to FGT_1 port1 using FGT_1's MAC address. Traffic is then inspected by FGT_1.

This example requires the following settings:

- The internal and outgoing interfaces of both FortiGates in the FGSP pair are in the same subnet.
- Both peers have layer 2 access with each other.



Due to the bouncing of traffic back to the session owner, performance degradation is expected. Starting in FortiOS 7.6.4, administrators can manually disable this behavior using the CLI:

```
config system standalone-cluster
    set helper-traffic-bounce disable
    set utm-traffic-bounce disable
end
```

Note that disabling these options may stop the asymmetric-traffic UTM features from working.

To configure FGT_1:

1. Configure FGSP cluster attributes, including setting the peer IP to the IP address of FGT_2:

```
config system standalone-cluster
    set standalone-group-id 1
    set group-member-id 0
    set layer2-connection available
    unset session-sync-dev
    config cluster-peer
        edit 1
            set peerip 10.2.2.2
        next
    end
end
```

2. Configure the firewall policy:

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set av-profile "default"
        set logtraffic all
        set nat enable
```

```
    next
end
```

To configure FGT_2:

1. Configure FGSP cluster attributes, including setting the peer IP to the IP address of FGT_1:

```
config system standalone-cluster
    set standalone-group-id 1
    set group-member-id 1
    set layer2-connection available
    unset session-sync-dev
    config cluster-peer
        edit 1
            set peerip 10.2.2.1
        next
    end
end
```

2. Configure the firewall policy:

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set av-profile "default"
        set logtraffic all
        set nat enable
    next
end
```

Results

Capture packets on FGT_2 to see that traffic bounced from FGT_2 to FGT_1 over the traffic interface.

```
FGT_2 # diagnose sniffer packet any 'host 10.1.100.15 and host 172.6.200.55' 4
interfaces=[any]
filters=[host 10.1.100.15 and host 172.16.200.55]
91.803816 port1 in 172.16.200.55.80 -> 10.1.100.15.40008: syn 2572073713 ack 261949279
92.800480 port1 in 172.16.200.55.80 -> 10.1.100.15.40008: syn 2572073713 ack 261949279
92.800486 port1 out 172.16.200.55.80 -> 10.1.100.15.40008: syn 2572073713 ack 261949279
92.800816 port1 in 172.16.200.55.80 -> 10.1.100.15.40008: syn 2572073713 ack 261949279
92.800818 port1 out 172.16.200.55.80 -> 10.1.100.15.40008: syn 2572073713 ack 261949279
```

When defining a logical interface such as a Link Aggregate interface, all FGSP peers must define the interface in the same configuration order. If the configuration order does not match, the `phyindex` used by the HA module may not match, causing asymmetric traffic destined for an interface to be dropped.

To view the `phyindex` number for an interface, use the following command:

```
# diagnose system ha standalone-peers
```

FortiGates running FGSP in transparent mode are also able to do UTM inspection on asymmetric TCP traffic. Other protocols such as UDP and ICMP are not supported.

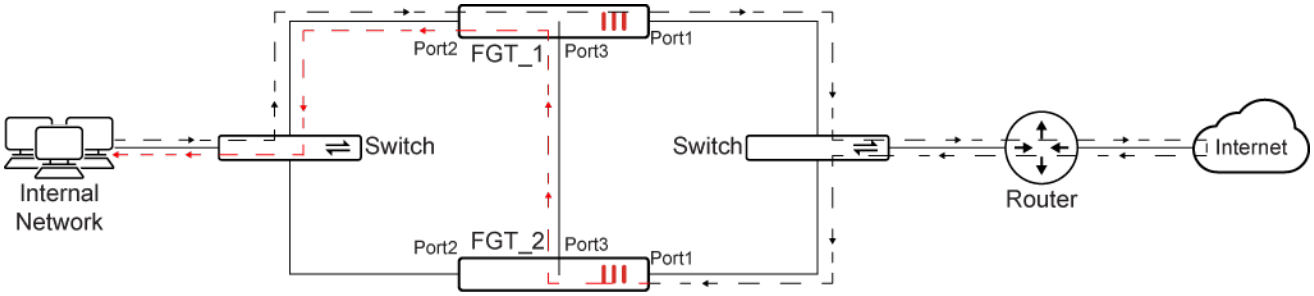
UTM inspection on asymmetric traffic on L3

When traffic passes asymmetrically through FGSP peers, UTM inspection can be supported by always forwarding traffic back to the session owner for processing. The session owner is the FortiGate that receives the first packet of the session.

For networks where L2 connectivity is not available, such as cloud environments, traffic bound for the session owner are forwarded through the peer interface using a UDP connection.

In this example, traffic from the internal network first hits FGT_1, but the return traffic is routed to FGT_2. Consequently, return traffic is packed and sent from FGT_2 to FGT_1 using UDP encapsulation between two peer interfaces (port 3). Traffic is then inspected by FGT_1.

Both of the FortiGates in this example are peering on interfaces in the same subnet, but the solution does not require the FortiGate peers to be on the same subnet or connected back-to-back.



Due to the bouncing of traffic back to the session owner, performance degradation is expected.

Starting in FortiOS 7.6.4, administrators can manually disable this behavior using the CLI:

```
config system standalone-cluster
    set helper-traffic-bounce disable
    set utm-traffic-bounce disable
end
```

Note that disabling these options may stop the asymmetric-traffic UTM features from working.

To configure FGT_1:

1. Configure FGSP cluster attributes, including setting the peer IP to the IP address of FGT_2:

```
config system standalone-cluster
    set standalone-group-id 1
    set group-member-id 0
    set layer2-connection unavailable
    unset session-sync-dev
    config cluster-peer
        edit 1
            set peerip 10.2.2.2
        next
    end
end
```

2. Configure the firewall policy:

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set av-profile "default"
        set logtraffic all
        set nat enable
    next
end
```

To configure FGT_2:

1. Configure FGSP cluster attributes, including setting the peer IP to the IP address of FGT_1:

```
config system standalone-cluster
    set standalone-group-id 1
    set group-member-id 1
    set layer2-connection unavailable
    unset session-sync-dev
    config cluster-peer
        edit 1
            set peerip 10.2.2.1
        next
    end
end
```

2. Configure the firewall policy:

```
config firewall policy
    edit 1
        set srcintf "port2"
        set dstintf "port1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set av-profile "default"
        set logtraffic all
        set nat enable
    next
end
```

FortiGates running FGSP in transparent mode are also able to do UTM inspection on asymmetric TCP traffic. Other protocols such as UDP and ICMP are not supported.

FGSP support for failover with asymmetric traffic and UTM

FortiGate can continue sessions on the active FGSP peers if the original FGSP peer, which initially received the session's first packet, becomes unavailable or unhealthy. It may go into this state due to:

- Device shutdown or reboot
- A monitored interface is down
- A ping-server monitor goes down

When the original peer for a session is unavailable or unhealthy and other peers receive the asymmetric traffic, this traffic does not bounce back to the original peer. Instead, it gets handled by the active FGSP peer. In the case of an unhealthy peer due to failed monitored interface or ping-server monitor, it will share its not-ready status over FGSP heartbeats so other peers will know not to bounce traffic back to it.

In the case of an unavailable peer, once the original FGSP peer is back online, it will perform session sync. While it is syncing, it will share a not-ready status over FGSP so peers will not bounce traffic back to it. Once session sync is complete, it will share a ready status and accept sessions to fail back to it.

UTM inspection will not occur during failover since the traffic must be scanned by the original peer only. Furthermore, during failback of the session to the original FGSP peer, the session will only be scanned if the policy is using flow-based inspection.

This ensures continuity and reliability of the network sessions, even if a device does not function as expected.

There are two options available in the `config system standalone-cluster` command on the CLI:

```
config system standalone-cluster
    set monitor-interface <interface name>
```

```
    set pingsvr-monitor-interface <interface name>
end
```

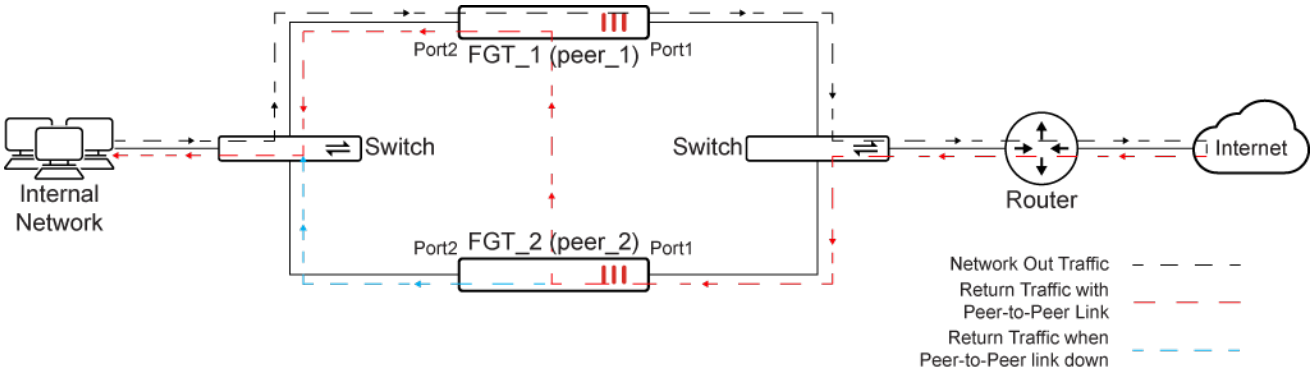
Example

In the following configurations, two peers are configured in FGSP, and a monitored interface and ping server monitor are configured.

```
config system link-monitor
    edit "1"
        set srcintf "wan1"
        set server "172.16.200.254"
    next
end
config system standalone-cluster
    set standalone-group-id 22
    set group-member-id 2
    config cluster-peer
        edit 1
            set peerip 10.2.2.2
            set hb-interval 20
            set hb-lost-threshold 60
        next
    end
    set monitor-interface "wan2"
    set pingsvr-monitor-interface "wan1"
end
```

Traffic originally passes through UTM inspection over *peer_1*. The return traffic is routed to *peer_2*, where it will bounce to *peer_1*, the original FGSP peer for inspection.

If *peer_1* becomes unavailable or unhealthy, traffic no longer bounces back to *peer_1*. Instead, it is failed over to *peer_2* for processing.



Monitor routing prefix for FGSP session failover

FGSP now supports another method for determining an FGSP peer as unhealthy, by monitoring the health of a routing prefix from RIP, OSPF or BGP. Using this method, FortiGate can prevent network isolation and blackholing by recognizing a critical data path is down. For instance, when peer 1's path to a certain network no longer exists in the routing table, it will share a not-ready status over FGSP heartbeats so other peers will know not to bounce traffic back to it.

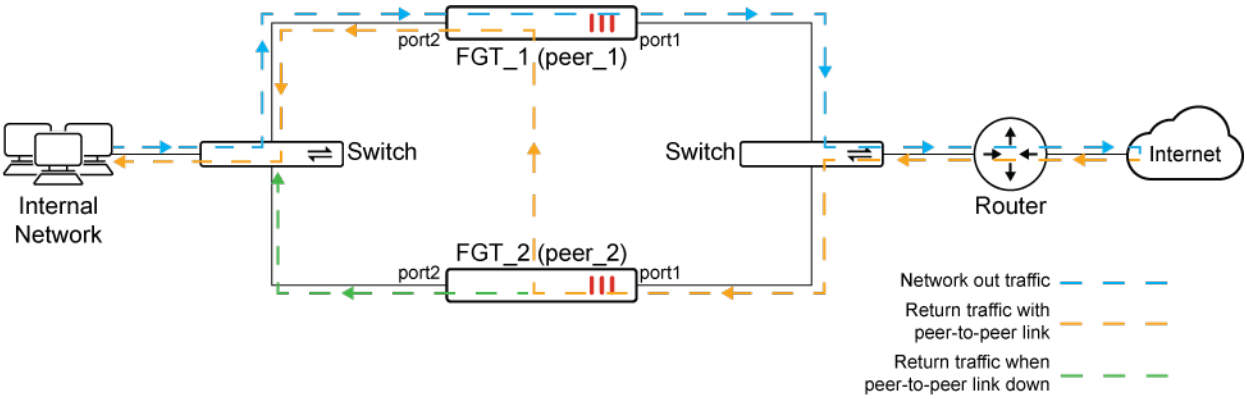
> Multiple prefixes can be monitored. However, a bad health status on a prefix will trigger the entire peer as not-ready, not only the specific path. Furthermore, if traffic is failed over to the peer that is not the original owner of the session, then UTM inspection will not apply.

Route monitoring to FGSP peers can be configured in the CLI:

```
config system standalone-cluster
    config monitor-prefix
        edit <ID>
            set vdom <VDOM name>
            set vrf <VRF ID>
            set prefix <ip address and netmask>
        next
    end
end
```

Example

In the following configurations, two peers are configured in FGSP. Two routing prefixes are monitored. In the diagram below, traffic travels asymmetrically, but is eventually bounced back to the peer (FGT_1) where the traffic was initiated. This allows the traffic to continue its UTM inspection through the peer.



In the scenario that one of the peers (FGT_1) no longer sees a route to a prefix, disrupting the flow of traffic, instead of causing network isolation and blackholing by bouncing the traffic back to the peer (FGT_1), traffic instead continues through the healthy peer (FGT_2).

Note that in this case, traffic does not get scanned by UTM on the healthy peer (FGT_2).

To configure routing prefix monitoring to FGSP:

1. Configure two peers in FGSP:

```
config system standalone-cluster
    set standalone-group-id 1
    config cluster-peer
        edit 1
            set peerip 10.2.2.2
        next
    end
    config monitor-prefix
        edit 1
            set vdom "root"
            set prefix 192.168.2.0 255.255.255.0
        next
        edit 2
            set vdom "root"
            set prefix 20.1.1.0 255.255.255.0
        next
    end
end
```

2. Verification:
   a. FGT_1 and FGT_2 both learn the network prefix of 20.1.1.0/24 over RIP from the upstream router.

```
FGT_1 #  get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       V - BGP VPNv4
       * - candidate default

Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 172.16.200.254, wan1, [1/0]
C       10.1.100.0/24 is directly connected, port1
C       10.2.2.0/24 is directly connected, ha1
R       20.1.1.0/24 [120/2] via 172.16.200.3, wan1, 00:01:46, [1/0]
C       172.16.200.0/24 is directly connected, wan1
C       192.168.2.0/24 is directly connected, mgmt
```

   b. As such, the health status on *peer_1* for both prefixes display healthy=1:

```
FGT_1 # diagnose test application sessionsync 1
HA is not enabled
sync context:
        sync-enabled=0, sync-tcp=1, sync-nat=0
        sync-other=1, sync-exp=1, standalone-sync=1, mtu=0
        ipsec-tun-sync=1, encrypt-enabled=0
fgsp-peers-num=1, kernel-filters-num=1
fgsp-peers:
```

```
        vdom=0, ip/port=10.2.2.2:708
fgsp_route_health=1
        mon_prefix: vdom=root vrf=0, prefix=192.168.2.0(255.255.255.0) healthy=1
        mon_prefix: vdom=root vrf=0, prefix=20.1.1.0(255.255.255.0) healthy=1
```

c. On *peer_2*, it sees the health status of *peer_1* as ready:

```
FGT_2 # diagnose sys ha standalone-peers
Group=1, ID=2
Detected-peers=1
Peer ready bitmap=0000000100000000
Kernel standalone-peers: num=1.
peer0: vfid=0, peerip:port = 10.2.2.1:708, standalone_id=0, ready=1
        session-type: send=0, recv=0
         packet-type: send=0, recv=0
…
```

d. Traffic originally passes through UTM inspection over *peer_1*. The return traffic is routed to *peer_2*, where it will bounce to *peer_1*, the original FGSP peer for inspection.

e. In the event that the network prefix 20.1.1.0/24 becomes unavailable from *peer_1*:

```
FGT_1 # get router info routing-table all
Codes: K - kernel, C - connected, S - static, R - RIP, B - BGP
       O - OSPF, IA - OSPF inter area
       N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
       E1 - OSPF external type 1, E2 - OSPF external type 2
       i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
       V - BGP VPNv4
       * - candidate default
Routing table for VRF=0
S*      0.0.0.0/0 [10/0] via 172.16.200.254, wan1, [1/0]
C       10.1.100.0/24 is directly connected, port1
C       10.2.2.0/24 is directly connected, ha1
C       172.16.200.0/24 is directly connected, wan1
C       192.168.2.0/24 is directly connected, mgmt
```

f. Its health status immediately changes to unhealthy (`healthy=0`).

```
# diagnose test application sessionsync 1
HA is not enabled
sync context:
        sync-enabled=0, sync-tcp=1, sync-nat=0
        sync-other=1, sync-exp=1, standalone-sync=1, mtu=0
        ipsec-tun-sync=1, encrypt-enabled=0
fgsp-peers-num=1, kernel-filters-num=1
fgsp-peers:
        vdom=0, ip/port=10.2.2.2:708
fgsp_route_health=0
        mon_prefix: vdom=root vrf=0, prefix=192.168.2.0(255.255.255.0) healthy=1
        mon_prefix: vdom=root vrf=0, prefix=20.1.1.0(255.255.255.0) healthy=0
```

g. *peer_2* also detects the not-ready status over the heartbeat packets (`ready=0`):

```
FGT_2 # diagnose sys ha standalone-peers
Group=1, ID=2
Detected-peers=1
Peer ready bitmap=0000000000000000
Kernel standalone-peers: num=1.
peer0: vfid=0, peerip:port = 10.2.2.1:708, standalone_id=0, ready=0
        session-type: send=0, recv=0
         packet-type: send=0, recv=0
```

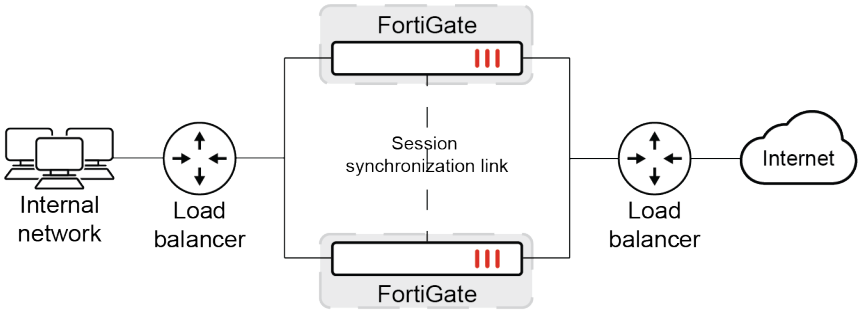
h. Upon *peer_1* becoming unavailable or unhealthy, traffic no longer bounces back to *peer_1*. Instead, it is failed over to *peer_2* for processing.

FGSP synchronizes firewall-authenticated users

FGSP synchronizes firewall-authenticated users between FGSP peers. If authenticated users sessions change to a different FGSP peer they do not have to re-authenticate. No special FGSP configuration is required to support this synchronization.

A firewall authenticated user is a user who has been authenticated by a user group that has been added to a firewall policy (see User groups.)

Example configuration



First FGSP peer configuration

This simple example includes setting up a basic FCSP configuration with two FGSP peers. The configuration uses local firewall users as the example authentication configuration.

```
config system standalone-cluster
    set standalone-group-id 222
    set group-member-id 8
    config cluster-peer
        edit 1
            set peerip 10.2.2.2
        next
    end
end
```

```
diagnose sys ha fgsp-zone
Local standalone-member-id: 8
Primary FGSP peer_num = 1, ver=3
        peer[1]: standalone-member-id=9, IP=10.2.2.2,vd=root, serial_no=FG101FTK19002788, prio=1,
rsso_sync_done=0
```

Second FGSP peer configuration

```
config system standalone-cluster
    set standalone-group-id 222
    set group-member-id 9
    config cluster-peer
        edit 1
            set peerip 10.2.2.1
        next
    end
end
```

```
diagnose sys ha fgsp-zone
Local standalone-member-id: 9
Primary FGSP peer_num = 1, ver=6
        peer[1]: standalone-member-id=8, IP=10.2.2.1,vd=root, serial_no=FG101FTK19003737, prio=1,
rsso_sync_done=0
```

Local user and firewall configuration

The following local user configuration must be the same for all FGSP peers:

```
config user local
    edit "tt"
        set type password
        set passwd *
    next
end
```

```
config user group
    edit "g1"
        set member "tt"
    next
end
```

```
config firewall policy
    edit 1
        set srcintf "port1"
        set dstintf "wan1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
```

```
        set schedule "always"
        set service "ALL"
        set groups "g1" <-------------------- local user group
    next
```

Generate HTTP traffic

The peer FortiGate that processes the traffic could have the following session information:

```
diagnose firewall auth list

10.1.100.24, tt
        src_mac: 00:0c:29:a7:3e:3b
        type: fw, id: 0, duration: 4984, idled: 8
        expire: 292, allow-idle: 300
        server: tt
        packets: in 2610 out 4784, bytes: in 321123 out 304220
        user_id: 16777218
        group_id: 2
        group_name: g1

----- 1 listed, 0 filtered ------
```

```
diagnose sys session list


session info: proto=6 proto_state=01 duration=9 expire=3590 timeout=3600 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=8:0 policy_dir=0 tunnel=/ vlan_cos=0/255
user=tt auth_server=tt state=may_dirty npu authed synced acct-ext
statistic(bytes/packets/allow_err): org=92/2/1 reply=52/1/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=11->7/7->11 gwy=172.16.200.55/10.1.100.24
hook=post dir=org act=snat 10.1.100.24:50851->172.16.200.55:80(172.16.200.1:50851)
hook=pre dir=reply act=dnat 172.16.200.55:80->172.16.200.1:50851(10.1.100.24:50851)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=1 pol_uuid_idx=4072 auth_info=2 chk_client_info=0 vd=0
serial=00005500 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
npu_state=0x000c00 ofld-O ofld-R
npu info: flag=0x81/0x81, offload=8/8, ips_offload=0/0, epid=66/70, ipid=70/66, vlan=0x0000/0x0000
vlifid=70/66, vtag_in=0x0000/0x0000 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=4/7, ha_divert=0/0
hrx info: valid=1/1, qid=5/8, npuid=1/1, sublink=0/0
total session: 1
```

Similar sessions appear on the other peer FortiGate:

```
diagnose firewall auth list

10.1.100.24, tt
        src_mac: 00:0c:29:a7:3e:3b
        type: fw, id: 0, duration: 81, idled: 88
        expire: 212, allow-idle: 300
        flag(400): ha
        server: tt
        packets: in 0 out 0, bytes: in 0 out 0
        user_id: 16777218
        group_id:
        group_name:

----- 1 listed, 0 filtered ------
```

```
diagnose sys session list

session info: proto=6 proto_state=01 duration=14 expire=3585 timeout=3600 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=8:0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=dirty may_dirty npu syn_ses
statistic(bytes/packets/allow_err): org=0/0/0 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=11->7/7->11 gwy=0.0.0.0/0.0.0.0
hook=post dir=org act=snat 10.1.100.24:50851->172.16.200.55:80(172.16.200.1:50851)
hook=pre dir=reply act=dnat 172.16.200.55:80->172.16.200.1:50851(10.1.100.24:50851)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=1 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=0
serial=00005500 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
npu_state=00000000
npu info: flag=0x00/0x00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0, vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0, ha_divert=0/0
no_ofld_reason:
hrx info: valid=0/0, qid=0/0, npuid=0/0, sublink=0/0
total session: 1
```

After a failover

You can simulate a failover by restarting the peer FortiGate that is processing traffic. The authenticated user can continue to send traffic through the second peer FortiGate without re-authenticating.

```
diagnose firewall auth list

10.1.100.24, tt
        src_mac: 00:0c:29:a7:3e:3b
        type: fw, id: 0, duration: 81, idled: 88
        expire: 212, allow-idle: 300
```

```
        flag(400): ha
        server: tt
        packets: in 0 out 0, bytes: in 0 out 0
        user_id: 16777218
        group_id:
        group_name:

----- 1 listed, 0 filtered ------
```

```
diagnose sys session list

session info: proto=6 proto_state=01 duration=14 expire=3585 timeout=3600 refresh_dir=both
flags=00000000 socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=8:0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=dirty may_dirty npu syn_ses
statistic(bytes/packets/allow_err): org=230/5/0 reply=52/1/1 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=11->7/7->11 gwy=0.0.0.0/0.0.0.0
hook=post dir=org act=snat 10.1.100.24:50851->172.16.200.55:80(172.16.200.1:50851)
hook=pre dir=reply act=dnat 172.16.200.55:80->172.16.200.1:50851(10.1.100.24:50851)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=1 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=0
serial=00005500 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfw_id=n/a duplicaton_id=0
npu_state=00000000
npu info: flag=0x00/0x00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0, vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0, ha_divert=0/0
no_ofld_reason:
hrx info: valid=0/0, qid=0/0, npuid=0/0, sublink=0/0
total session: 1
```

Encryption for L3 on asymmetric traffic in FGSP

In scenarios where asymmetric routing between FGSP members occurs, the return traffic can be encrypted and routed back to the session owner on Layer 3 (L3).

To encrypt L3 traffic in FGSP:

1. Run the following on both FortiGates:

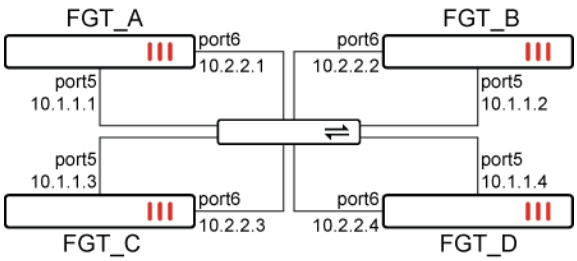
```
config system standalone-cluster
    set encryption enable
    set psksecret xxxxxxxxx
end
```

Optimizing FGSP session synchronization and redundancy

In this example where standalone FortiGates are peered in FGSP, using `session-sync-dev` optimizes session synchronization as it eliminates UDP encapsulation and offloads session synchronization processing to the kernel. FGSP session synchronization can be supported to handle heavy loads.

For more information about session synchronization, see Session synchronization interfaces in FGSP on page 4220.

Topology



In this topology, there are three FGSP peer groups for each FortiGate. Sessions are synchronized between each FortiGate and its peer groups. Redundancy is achieved by using two dedicated session sync device links for each peer setup. There are a total of six peer IPs for each session synchronization device link in each FGSP peer. When one link is fails, session synchronization is not affected.

For optimization, `sync-packet-balance` is enabled to distribute synchronization packets processing to multiple CPUs. The session synchronization process is offloaded to the kernel, and sessions are synchronized over layer 2 over the connected interfaces (`set session-sync-dev "port5" "port6"`). Jumbo frame MTU 9216 is configured on each session synchronization device link to reduce the number of packets; however, setting MTU to 9216 is entirely optional.

To configure FGT_A:

1. Configure HA:

```
config system ha
    set sync-packet-balance enable
    set session-pickup enable
    set session-pickup-connectionless enable
    set session-pickup-expectation enable
    set session-pickup-nat enable
end
```

2. Configure the layer 2 session synchronization links:

```
config system standalone-cluster
    set session-sync-dev "port5" "port6"
end
```

3. Configure the session TTL default timeout:

```
config system session-ttl
    set default 300
end
```

4. Configure the interfaces:

```
config system interface
    edit port5
        set ip 10.1.1.1/24
        set mtu-override enable
        set mtu 9216
    next
    edit port6
        set ip 10.2.2.1/24
        set mtu-override enable
        set mtu 9216
    next
end
```

5. Configure FGSP session synchronization:

```
config system standalone-cluster
    config cluster-peer
        edit 1
            set peerip 10.1.1.2
        next
        edit 2
            set peerip 10.2.2.2
        next
        edit 3
            set peerip 10.1.1.3
        next
        edit 4
            set peerip 10.2.2.3
        next
        edit 5
            set peerip 10.1.1.4
        next
        edit 6
            set peerip 10.2.2.4
        next
    end
end
```

To configure FGT_B:

1. Configure HA:

```
config system ha
    set sync-packet-balance enable
    set session-pickup enable
    set session-pickup-connectionless enable
```

```
    set session-pickup-expectation enable
    set session-pickup-nat enable
end
```

2. Configure the layer 2 session synchronization links:

```
config system standalone-cluster
    set session-sync-dev "port5" "port6"
end
```

3. Configure the session TTL default timeout:

```
config system session-ttl
    set default 300
end
```

4. Configure the interfaces:

```
config system interface
    edit port5
        set ip 10.1.1.2/24
        set mtu-override enable
        set mtu 9216
    next
    edit port6
        set ip 10.2.2.2/24
        set mtu-override enable
        set mtu 9216
    next
end
```

5. Configure FGSP session synchronization:

```
config system standalone-cluster
    config cluster-peer
        edit 1
            set peerip 10.1.1.1
        next
        edit 2
            set peerip 10.2.2.1
        next
        edit 3
            set peerip 10.1.1.3
        next
        edit 4
            set peerip 10.2.2.3
        next
        edit 5
            set peerip 10.1.1.4
        next
        edit 6
            set peerip 10.2.2.4
```

```
        next
    end
end
```

To configure FGT_C:

1. Configure HA:

```
config system ha
    set sync-packet-balance enable
    set session-pickup enable
    set session-pickup-connectionless enable
    set session-pickup-expectation enable
    set session-pickup-nat enable
end
```

2. Configure the layer 2 session synchronization links:

```
config system standalone-cluster
    set session-sync-dev "port5" "port6"
end
```

3. Configure the session TTL default timeout:

```
config system session-ttl
    set default 300
end
```

4. Configure the interfaces:

```
config system interface
    edit port5
        set ip 10.1.1.3/24
        set mtu-override enable
    set mtu 9216
    next
    edit port6
        set ip 10.2.2.3/24
        set mtu-override enable
        set mtu 9216
    next
end
```

5. Configure FGSP session synchronization:

```
config system standalone-cluster
    config cluster-peer
        edit 1
            set peerip 10.1.1.1
        next
        edit 2
            set peerip 10.2.2.1
```

```
        next
        edit 3
            set peerip 10.1.1.2
        next
        edit 4
            set peerip 10.2.2.2
        next
        edit 5
            set peerip 10.1.1.4
        next
        edit 6
            set peerip 10.2.2.4
        next
    end
end
```

To configure FGT_D:

1. Configure HA:

```
config system ha
    set sync-packet-balance enable
    set session-pickup enable
    set session-pickup-connectionless enable
    set session-pickup-expectation enable
    set session-pickup-nat enable
end
```

2. Configure the layer 2 session synchronization links:

```
config system standalone-cluster
    set session-sync-dev "port5" "port6"
end
```

3. Configure the session TTL default timeout:

```
config system session-ttl
    set default 300
end
```

4. Configure the interfaces:

```
config system interface
    edit port5
        set ip 10.1.1.4/24
        set mtu-override enable
        set mtu 9216
    next
    edit port6
        set ip 10.2.2.4/24
        set mtu-override enable
        set mtu 9216
```

```
    next
end
```

5. Configure FGSP session synchronization:

```
config system standalone-cluster
    config cluster-peer
        edit 1
            set peerip 10.1.1.1
        next
        edit 2
            set peerip 10.2.2.1
        next
        edit 3
            set peerip 10.1.1.2
        next
        edit 4
            set peerip 10.2.2.2
        next
        edit 5
            set peerip 10.1.1.3
        next
        edit 6
            set peerip 10.2.2.3
        next
    end
end
```

Firmware upgrades in FGSP

FGSP supports cluster members using different firmware versions with some limitations. This allows for cluster members to be upgraded without needing to remove them from the cluster or network. You can find details on the requirements to support different firmware versions in the cluster in Different FortiGate models on page 4244.

If your cluster's current and target firmware are supported, you may upgrade each member one by one without any need to disconnect the member.

Otherwise, the following steps are recommended to upgrade the firmware of FortiGates in an FGSP deployment. Follow these steps whether or not you have enabled standalone configuration synchronization.

This example FGSP deployment has two FortiGates, FGT-1 and FGT-2.

To upgrade the firmware in an FGSP deployment:

1. Switch all traffic to FGT-1:
   a. Configure the load balancer or router that distributes traffic between the FortiGates to send all traffic to FGT-1.
2. Disconnect FGT-2 from the network.

Make sure to also disconnect the interfaces that allow heartbeat and synchronization communication with FGT-1. This is to prevent FGT-2 from communicating with FGT-1.

3. Upgrade the firmware on FGT-2.
4. Reconnect the traffic interfaces on FGT-2, but not the interfaces used for heartbeat and synchronization communication with FGT-1.
5. Switch all traffic to the newly upgraded FGT-2:
   a. Configure the load balancer or router that distributes traffic between the FortiGates to send all traffic to FGT-2.
6. Upgrade the firmware on FGT-1 (while heartbeat and synchronization communication with FGT-2 remains disconnected).
7. Reconnect the FGT-2 interfaces that allow heartbeat and synchronization communication between FGT-1 and FGT-2.
8. Restore the original traffic distribution between FGT-1 and FGT-2:
   a. Configure the load balancer or router to distribute traffic to both FortiGates in the FGSP deployment.

FGSP session synchronization between different FortiGate models or firmware versions

FGSP HA deployments are generally meant for interoperating between FortiGates with the same model and firmware version. However, situations may arise where individual members or FGCP clusters running over FGSP use different models or firmware versions. For example, to avoid downtime while upgrading the members, some FGSP members or clusters may be upgraded first and then re-join the FGSP peers after a successful upgrade. Or while performing maintenance, sessions may need to be offloaded to a temporary member or FGCP cluster of a different model.

Being able to perform FGSP session synchronization between members of different models or firmware versions is helpful to transition the traffic smoothly and causes minimal disruptions. This topic outlines requirements to be aware of before assessing whether FGSP session synchronization may work between members with different models or firmware versions.

Different FortiGate models

The general guideline is to only use FortiGate models in a similar tier and family. Vastly different models have different performance and capabilities, which may not be compatible. The goal is for two models to have similar capabilities so that data structures used in session synchronization will match, and are capable of delivering similar performance.

When considering FGSP session synchronization between two FortiGates, ensure that:

- The FortiGates use the same 32-bit kernel or 64-bit kernel.
- The FortiGates use the same type of CPU (such as ARM or x86).
- For network interfaces:
  - The same type of physical interface should be used on each member.
  - The physical interfaces should be capable of the same speeds.

- The device memory should be similar in size. If the FortiGates have vastly different memory sizes, their performance may be different if one device supports more sessions than the other.
- The configurations related to session tables should match. For example, the logical names used in firewall policies, IPsec interface names, VDOM names, firewall policy tables, and so on.

Different firmware versions

When operating in FGSP, the firmware needs to have compatible data structures and session synchronization packet headers. The firmware is generally able to handle different data structures between old and new FortiOS sessions. Session synchronization packets are typically the same between versions.

Note the following exceptions and guidelines when assessing FGSP session synchronization compatibility between different firmware versions:

- FortiOS 7.0.2 added support for widening the HA virtual MAC address range. This change updated the session synchronization packet header structure.
  - FortiGates running 7.0.2 or later, and FortiGates running 7.0.1 or earlier will not accept session synchronization packets from each other.
- If the traffic uses a new feature only available in a newer FortiOS version, it may not work when synchronized to an older FortiOS version.
  - For example, PFCP (Packet Forwarding Control Protocol) support was added in 7.0.1, and a PFCP profile name was added to the sessions. When the sessions are synchronized to an older firmware version, the PFCP profile name will be lost and the sessions will not be able to handle the traffic as they would in 7.0.1.
- FortiOS 7.2.1 added `group-id` into the protocol header. This means that FortiGates running 7.2.1 and later cannot perform session synchronization with FortiGates running earlier versions.

Session synchronization interfaces

Session synchronization between FGSP members uses an L3 connection over the peer IP by default.

Session synchronization between FGSP members uses an L2 connection when a session synchronization interface (`session-sync-dev`) is used. The synchronization process is also offloaded to the kernel.

FGSP is also compatible with FortiGate VRRP.

Applying the session synchronization filter only between FGSP peers in an FGCP over FGSP topology

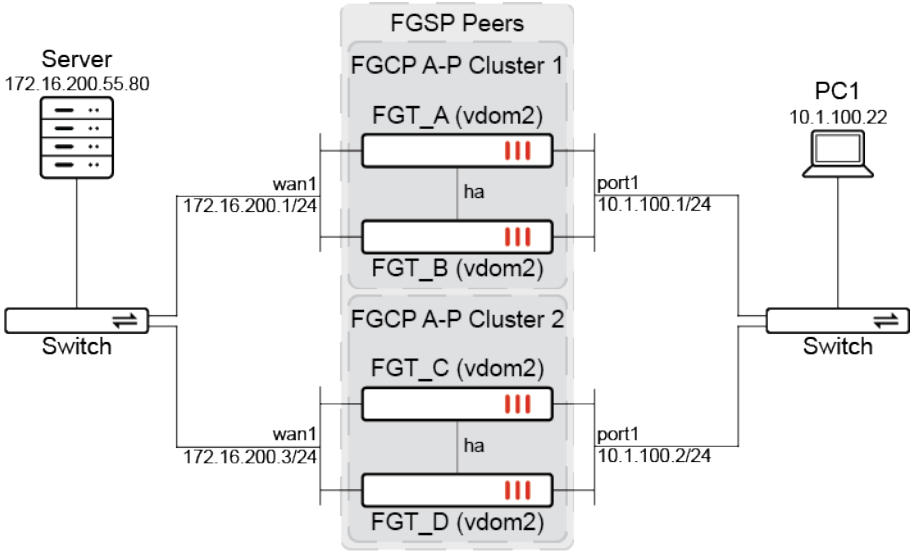
When the session synchronization filter is applied on FGSP, the filter will only affect sessions synchronized between the FGSP peers. When virtual clustering is used, sessions synchronized between each virtual cluster can also be synchronized to FGSP peers. All peers' `syncvd` must be in the same HA virtual cluster.

Example

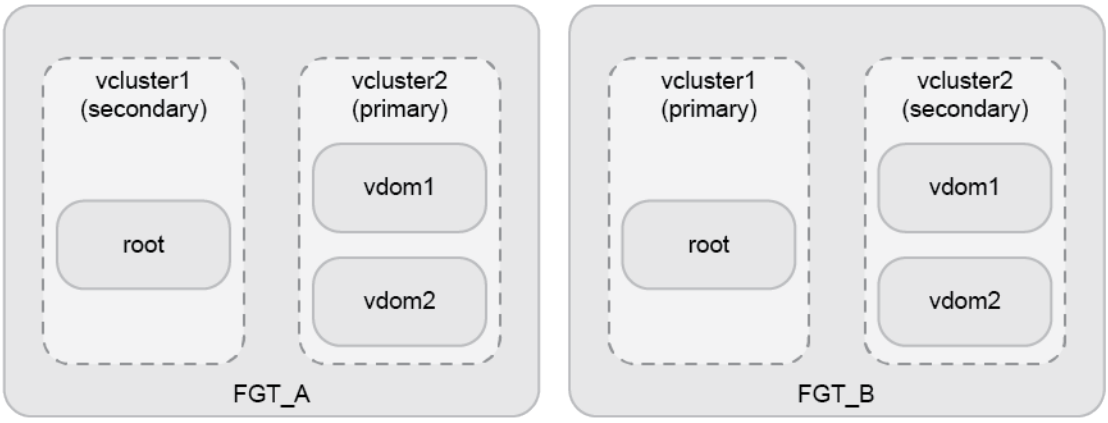
In this example, there is a simplified configuration where there is no router or load balancer performing balancing between the FGSP peers, but it demonstrates the following:

- When sessions pass through FGCP A-P Cluster 1, all sessions are synchronized between the FGT_A and FGT_B regardless of the session synchronization filter.
- Session synchronization between the FGSP peers (FGCP A-P Cluster 1 and 2) only occurs for the service specified in the filter, which is HTTP/80.
- The preceding behavior is applicable when virtual clustering is configured. This example focuses on vdom2, which belongs to vcluster2. FGT_A is the primary for vcluster2.

Each FGSP A-P cluster is connected on ha as the FGCP cluster heartbeat device. The FGSP peers are connected on mgmt over 10.1.1.1-2/24.



Virtual clustering between FGT_A and FGT_B:



Interface	FGT_A	FGT_B	FGT_C	FGT_D
wan1	172.16.200.1/24	172.16.200.1/24	172.16.200.3/24	172.16.200.3/24

Interface	FGT_A	FGT_B	FGT_C	FGT_D
port1	10.1.100.1/24	10.1.100.1/24	10.1.100.2/24	10.1.100.2/24
mgmt	10.1.1.1/24	10.1.1.1/24	10.1.1.2/24	10.1.1.2/24
ha	FGCP cluster heartbeat device		FGCP cluster heartbeat device	

To configure the HA clusters:

1. Configure FGCP A-P Cluster 1 (use the same configuration for FGT_A and FGT_B):

```
config system ha
    set group-id 146
    set group-name "FGT_HA1"
    set mode a-p
    set password **********
    set hbdev "wan2" 100 "ha" 50
    set session-pickup enable
    set session-pickup-nat enable
    set vcluster-status enable
    config vcluster
        edit 1
            set override enable
            set priority 25
            set monitor "wan1" "port1"
            set vdom "root"
        next
        edit 2
            set override disable
            set priority 150
            set monitor "wan1"
            set vdom "vdom2" "vdom1"
        next
    end
end
```

2. Configure FGCP A-P Cluster 2 (use the same configuration for FGT_C and FGT_D):

```
config system ha
    set group-id 200
    set group-name "FGT_HA2"
    set mode a-p
    set password **********
    set hbdev "wan2" 100 "ha" 50
    set session-pickup enable
    set session-pickup-nat enable
    set vcluster-status enable
    config vcluster
        edit 1
            set override enable
            set priority 120
            set monitor "wan1" "port1"
```

```
            set vdom "root"
        next
        edit 2
            set override disable
            set priority 150
            set monitor "wan1"
            set vdom "vdom2" "vdom1"
        next
    end
end
```

To configure the FGSP peers:

1. Configure FGT_A:

```
config system standalone-cluster
    set standalone-group-id 1
    set group-member-id 1
    config cluster-peer
        edit 1
            set peervd "vdom2"
            set peerip 10.1.1.2
            set syncvd "vdom2"
            config session-sync-filter
                config custom-service
                    edit 1
                        set dst-port-range 80-80
                    next
                end
            end
        next
    end
end
```

   The configuration is automatically synchronized to FGT_B.

2. Configure FGT_C:

```
config system standalone-cluster
    set standalone-group-id 1
    set group-member-id 2
    config cluster-peer
        edit 1
            set peervd "vdom2"
            set peerip 10.1.1.1
            set syncvd "vdom2"
            config session-sync-filter
                config custom-service
                    edit 1
                        set dst-port-range 80-80
                    next
                end
```

```
            end
        next
    end
end
```

The configuration is automatically synchronized to FGT_D.

To verify the configuration:

1. Verify the FGSP peer information on Cluster 1:

```
FGT_A (global) # diagnose sys ha fgsp-zone
Local standalone-member-id: 1
FGSP peer_num = 1
        peer[1]: standalone-member-id=2, IP=10.1.1.2, vd=vdom2, prio=1
```

2. Verify the FGSP peer information on Cluster 2:

```
FGT_C (global) # diagnose sys ha fgsp-zone
Local standalone-member-id: 1
FGSP peer_num = 1
        peer[1]: standalone-member-id=1, IP=10.1.1.1, vd=vdom2, prio=1
```

3. Initiate two sessions, HTTP and SSH.
4. Verify that the HTTP session is synchronized from Cluster 1 to Cluster 2.
   a. Verify the session list of vdom2 on FGT_A:

```
FGT_A (vdom2) # diagnose sys session list

session info: proto=6 proto_state=01 duration=693 expire=3593 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=4
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=1:0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty npu synced f00
statistic(bytes/packets/allow_err): org=87531/1678/1 reply=7413876/6043/1 tuples=2
tx speed(Bps/kbps): 134/1 rx speed(Bps/kbps): 11357/90
orgin->sink: org pre->post, reply pre->post dev=11->7/7->11 gwy=172.16.200.55/10.1.100.22
hook=post dir=org act=snat 10.1.100.22:44260->172.16.200.55:80(172.16.200.1:44260)
hook=pre dir=reply act=dnat 172.16.200.55:80->172.16.200.1:44260(10.1.100.22:44260)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=7 pol_uuid_idx=579 auth_info=0 chk_client_info=0 vd=2
serial=000a79df tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000c00 ofld-O ofld-R
npu info: flag=0x81/0x81, offload=8/8, ips_offload=0/0, epid=66/70, ipid=70/66,
vlan=0x0000/0x0000
vlifid=70/66, vtag_in=0x0000/0x0000 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=1/0

session info: proto=6 proto_state=01 duration=326 expire=3589 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=4
```

```
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=1:0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log may_dirty npu synced f00
statistic(bytes/packets/allow_err): org=4721/41/1 reply=5681/36/1 tuples=2
tx speed(Bps/kbps): 14/0 rx speed(Bps/kbps): 17/0
orgin->sink: org pre->post, reply pre->post dev=11->7/7->11 gwy=172.16.200.55/10.1.100.22
hook=post dir=org act=snat 10.1.100.22:50234->172.16.200.55:22(172.16.200.1:50234)
hook=pre dir=reply act=dnat 172.16.200.55:22->172.16.200.1:50234(10.1.100.22:50234)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=7 pol_uuid_idx=579 auth_info=0 chk_client_info=0 vd=2
serial=000a7d90 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000c00 ofld-O ofld-R
npu info: flag=0x81/0x81, offload=8/8, ips_offload=0/0, epid=66/70, ipid=70/66,
vlan=0x0000/0x0000
vlifid=70/66, vtag_in=0x0000/0x0000 in_npu=1/1, out_npu=1/1, fwd_en=0/0, qid=6/6
total session 2
```

b. Verify the session list of vdom2 on FGT_B:

```
FGT_B (vdom2) # diagnose sys session list

session info: proto=6 proto_state=01 duration=736 expire=3100 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=1:0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log dirty may_dirty npu f00 syn_ses
statistic(bytes/packets/allow_err): org=0/0/0 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=11->7/7->11 gwy=0.0.0.0/0.0.0.0
hook=post dir=org act=snat 10.1.100.22:44260->172.16.200.55:80(172.16.200.1:44260)
hook=pre dir=reply act=dnat 172.16.200.55:80->172.16.200.1:44260(10.1.100.22:44260)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=7 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=2
serial=000a79df tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000000
npu info: flag=0x00/0x00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0,
vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
no_ofld_reason:

session info: proto=6 proto_state=01 duration=369 expire=3230 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=1:0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log dirty may_dirty npu f00 syn_ses
```

```
statistic(bytes/packets/allow_err): org=0/0/0 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=11->7/7->11 gwy=0.0.0.0/0.0.0.0
hook=post dir=org act=snat 10.1.100.22:50234->172.16.200.55:22(172.16.200.1:50234)
hook=pre dir=reply act=dnat 172.16.200.55:22->172.16.200.1:50234(10.1.100.22:50234)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=7 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=2
serial=000a7d90 tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000000
npu info: flag=0x00/0x00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0,
vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
no_ofld_reason:
total session 2
```

c. Verify the session list of vdom2 on FGT_C:

```
FGT_C (vdom2) # diagnose sys session filter dst 172.16.200.55
FGT_C (vdom2) # diagnose sys session filter src 10.1.100.22
FGT_C (vdom2) # diagnose sys session list

session info: proto=6 proto_state=01 duration=837 expire=2762 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=1:0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log dirty may_dirty npu f00 syn_ses
statistic(bytes/packets/allow_err): org=0/0/0 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=11->7/7->11 gwy=0.0.0.0/0.0.0.0
hook=post dir=org act=snat 10.1.100.22:44260->172.16.200.55:80(172.16.200.1:44260)
hook=pre dir=reply act=dnat 172.16.200.55:80->172.16.200.1:44260(10.1.100.22:44260)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=7 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=2
serial=000a79df tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000000
npu info: flag=0x00/0x00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0,
vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
no_ofld_reason:
total session 1
```

d. Verify the session list of vdom2 on FGT_D:

```
FGT-D (vdom2) # diagnose sys session filter dst 172.16.200.55
FGT-D (vdom2) # diagnose sys session filter src 10.1.100.22
FGT-D (vdom2) # diagnose sys session list

session info: proto=6 proto_state=01 duration=902 expire=2697 timeout=3600 flags=00000000
socktype=0 sockport=0 av_idx=0 use=3
```

```
origin-shaper=
reply-shaper=
per_ip_shaper=
class_id=0 ha_id=1:0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=log dirty may_dirty npu f00 syn_ses
statistic(bytes/packets/allow_err): org=0/0/0 reply=0/0/0 tuples=2
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=11->7/7->11 gwy=0.0.0.0/0.0.0.0
hook=post dir=org act=snat 10.1.100.22:44260->172.16.200.55:80(172.16.200.1:44260)
hook=pre dir=reply act=dnat 172.16.200.55:80->172.16.200.1:44260(10.1.100.22:44260)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=7 pol_uuid_idx=0 auth_info=0 chk_client_info=0 vd=2
serial=000a79df tos=ff/ff app_list=0 app=0 url_cat=0
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x4000000
npu info: flag=0x00/0x00, offload=0/0, ips_offload=0/0, epid=0/0, ipid=0/0,
vlan=0x0000/0x0000
vlifid=0/0, vtag_in=0x0000/0x0000 in_npu=0/0, out_npu=0/0, fwd_en=0/0, qid=0/0
no_ofld_reason:
total session 1
```

Session synchronization filters are designed to be configured symmetrically on all of the FGSP peers. In cases where the filters are configured asymmetrically, note the following differences:

- In an FGCP over FGSP topology, session filtering will be applied on the FGSP peer that has the filtering configured and is receiving the session synchronization.
- In an FGSP topology between standalone peers, the filtering will be applied on the FGSP peer that has the filtering configured and is sending out the session synchronization.

FGSP static site-to-site IPsec VPN setup

When configuring static site-to-site IPsec VPN between FGSP FortiGates and a remote gateway, the FGSP peers must have the `passive-mode` setting enabled in the `vpn ipsec phase1-interface` configuration to function as an IPsec responder. This is a required configuration in this setup. If the FGSP peers act as initiators for tunnel setup when `passive-mode` is disabled and both FGSP peers initiate the tunnel with the same gateway IP, the remote IPsec gateway will be unable to process this, and the tunnel negotiation will fail. Likewise, when a failover occurs in FGSP and a new peer begins to initiate tunnel traffic, the remote IPsec gateway will be unable to handle the traffic initiated from the new peer.

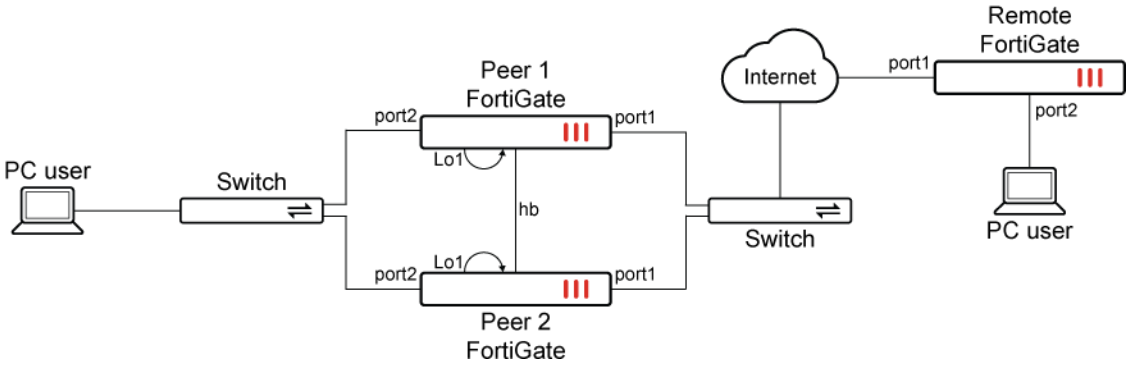
Enabling `passive-mode` ensures the FGSP peers only respond to tunnel initiations from the remote IPsec gateway and do not initiate tunnel negotiations. This way, the preceding situations will not occur.

For dynamic tunnel configuration examples on FGSP peers, see the following topics:

- FGSP per-tunnel failover for IPsec on page 4254
- FGCP over FGSP per-tunnel failover for IPsec on page 4260
- Allow IPsec DPD in FGSP members to support failovers on page 4270

Example

In this example, the FGSP peer has a loopback interface (Lo1) configured with the same IP address (192.168.202.31/32). All other interface IP addresses are different between the peers.



To configure IPsec on the FGSP peer FortiGates:

1. Configure the phase 1 settings:

```
config vpn ipsec phase1-interface
    edit "IPsec"
        set type static
        set interface "port1"
        set ike-version 2
        set local-gw 192.168.202.31
        set net-device disable
        set proposal aes256-sha256
        set dhgrp 14
        set passive-mode enable
        set remote-gw 10.10.100.100
    next
end
```

2. Configure the phase 2 settings:

```
config vpn ipsec phase2-interface
    edit "IPSec"
        set phase1name "IPsec"
        set proposal aes256-sha256
        set dhgrp 14
    next
end
```

To configure IPsec on the remote FortiGate:

1. Configure the phase 1 settings:

```
config vpn ipsec phase1-interface
    edit "IPsec"
```

```
        set type static
        set interface "port1"
        set ike-version 2
        set net-device disable
        set proposal aes256-sha256
        set dhgrp 14
        set remote-gw 192.168.202.31
    next
end
```

2. Configure the phase 2 settings:

```
config vpn ipsec phase2-interface
    edit "IPSec"
        set phase1name "IPsec"
        set proposal aes256-sha256
        set dhgrp 14
    next
end
```

FGSP per-tunnel failover for IPsec

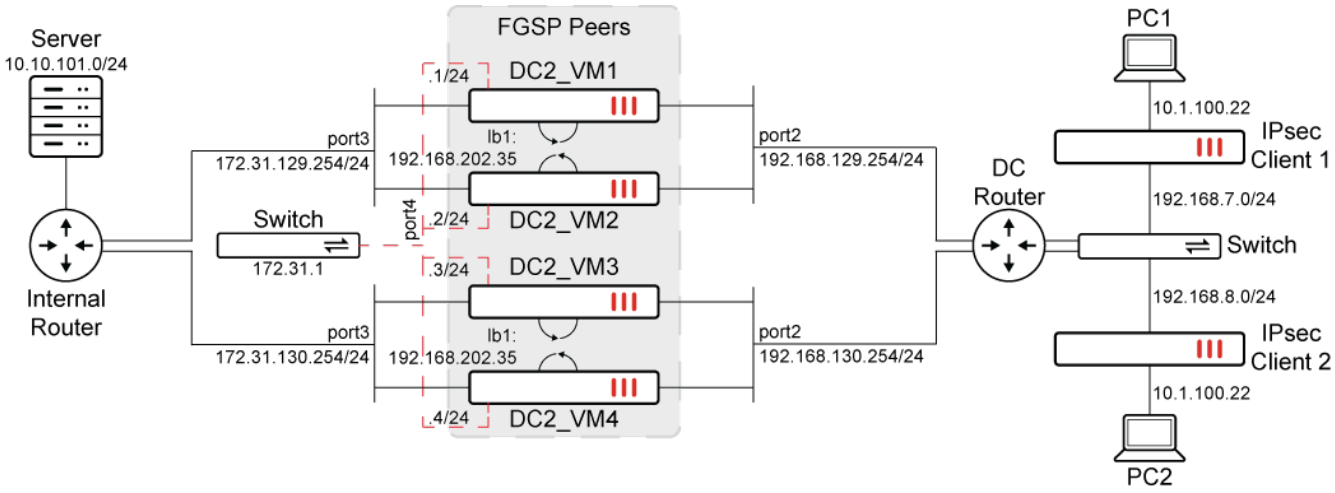
During FGSP per-tunnel failover for IPsec, the same IPsec dialup server configured on each FGSP member may establish tunnels with dialup clients as the primary gateway. The IPsec SAs are synchronized to all other FGSP peers that have FGSP synchronization for IPsec enabled. Other FGSP members may establish a tunnel with other clients on the same dialup server and synchronize their SAs to other peers.

Upon the failure of the FGSP member that is the primary gateway for a tunnel, the upstream router will fail over the tunnel traffic to another FGSP member. The other FGSP member will move from standby to the primary gateway for that tunnel and continue to forward traffic.

```
config vpn ipsec phase1-interface
    edit <name>
        set fgsp-sync {enable | disable}
    next
end
```

Example

In this example, the FGSP peers are connected on port4 over 172.31.1.1-4/24. Each peer has a loopback interface, lb1, with the same IP address. This loopback interface is used as the local gateway on each of the phase 1 connections to avoid each FGSP member having different IPs on port2. The DC Router uses ECMP to distribute traffic to each FGSP peer. It is assumed that the networking addresses are already configured properly.



Interface/setting	DC1_VM1	DC1_VM2	DC1_VM3	DC1_VM4
port2	192.168.125.254/24	192.168.126.254/24	192.168.127.254/24	192.168.128.254/24
port3	172.31.125.254/24	172.31.126.254/24	172.31.127.254/24	172.31.128.254/24
port4	172.31.1.1/24	172.31.1.2/24	172.31.1.3/24	172.31.1.4/24
lb1	192.168.202.31/32	192.168.202.31/32	192.168.202.31/32	192.168.202.31/32
`fgsp-sync`	Enabled	Enabled	Enabled	Disabled

Out of the four FGSP peers, DC1_VM1, DC1_VM2, and DC1_VM3 have `fgsp-sync` enabled in their IPsec phase 1 configurations. This allows the three FGSP members to synchronize IPsec SAs as clients establish dialup tunnels to them individually. DC1_VM4, which does not have `fgsp-sync` configured, will not participate in synchronizing IPsec SAs or establishing tunnels. The DC Router uses ECMP to route traffic to the destination 192.168.202.31 through each of the participating FGSP peers.

In a larger scale there may be many more IPsec dialup clients connecting, with each eligible FGSP peer being the primary gateway for a set of dialup tunnels, and is in standby for the rest of the tunnels. If an FGSP peer fails, traffic will fail over to other peers, and these peers will become primary gateways for the respective dialup tunnels.

To configure the FGSP peers (DC1_VM1):

The following steps are to configure DC1_VM1. The other peers have similar configurations based on the preceding table. In the `config vpn ipsec phase1-interface` settings, all peers should have the same local gateway external interface (192.168.202.31).

1. Configure the FGSP settings:

```
config system standalone-cluster
    set standalone-group-id 1
    set group-member-id 1
    config cluster-peer
        edit 1
```

```
            set peerip 172.31.1.2
        next
        edit 2
            set peerip 172.31.1.3
        next
        edit 3
            set peerip 172.31.1.4
        next
    end
end
```

2. Configure the VPN tunnel phase 1 settings:

```
config vpn ipsec phase1-interface
    edit "vpn1"
        set type dynamic
        set interface "port2"
        set ike-version 2
        set local-gw 192.168.202.31
        set keylife 90000
        set peertype one
        set net-device disable
        set proposal aes128-sha1
        set dpd on-idle
        set dhgrp 2
        set fgsp-sync enable
        set nattraversal disable
        set peerid "Nokia_Peer"
        set psksecret xxxxx
        set dpd-retryinterval 60
    next
end
```

3. Configure the VPN tunnel phase 2 settings:

```
config vpn ipsec phase2-interface
    edit "vpn1"
        set phase1name "vpn1"
        set proposal aes128-sha1
        set keylifeseconds 10800
    next
end
```

To verify the configuration:

1. Once the FGSP members establish peering with each other, verify the standalone peers on DC1_VM1:

```
DC1_VM1 # diagnose sys ha standalone-peers
Group=1, ID=1
Detected-peers=3
Kernel standalone-peers: num=3.
peer0: vfid=0, peerip:port = 172.31.1.2:708, standalone_id=2
```

```
        session-type: send=0, recv=0
         packet-type: send=0, recv=0
peer1: vfid=0, peerip:port = 172.31.1.3:708, standalone_id=3
        session-type: send=0, recv=0
         packet-type: send=0, recv=0
peer2: vfid=0, peerip:port = 172.31.1.4:708, standalone_id=4
        session-type: send=0, recv=0
         packet-type: send=0, recv=0
Kernel standalone dev_base:
        standalone_id=0:
        standalone_id=1:
                phyindex=0: mac=00:0c:29:22:00:6b, linkfail=1
                phyindex=1: mac=00:0c:29:22:00:75, linkfail=1
                phyindex=2: mac=00:0c:29:22:00:7f, linkfail=1
                phyindex=3: mac=00:0c:29:22:00:89, linkfail=1
                phyindex=4: mac=00:0c:29:22:00:93, linkfail=1
                phyindex=5: mac=00:0c:29:22:00:9d, linkfail=1
                phyindex=6: mac=00:0c:29:22:00:a7, linkfail=1
                phyindex=7: mac=00:0c:29:22:00:b1, linkfail=1
                phyindex=8: mac=00:0c:29:22:00:bb, linkfail=1
                phyindex=9: mac=00:0c:29:22:00:c5, linkfail=1
        standalone_id=2:
                phyindex=0: mac=00:0c:29:06:4e:d6, linkfail=1
                phyindex=1: mac=00:0c:29:06:4e:e0, linkfail=1
                phyindex=2: mac=00:0c:29:06:4e:ea, linkfail=1
                phyindex=3: mac=00:0c:29:06:4e:f4, linkfail=1
                phyindex=4: mac=00:0c:29:06:4e:fe, linkfail=1
                phyindex=5: mac=00:0c:29:06:4e:08, linkfail=1
                phyindex=6: mac=00:0c:29:06:4e:12, linkfail=1
                phyindex=7: mac=00:0c:29:06:4e:1c, linkfail=1
                phyindex=8: mac=00:0c:29:06:4e:26, linkfail=1
                phyindex=9: mac=00:0c:29:06:4e:30, linkfail=1
        standalone_id=3:
                phyindex=0: mac=00:0c:29:70:b9:6c, linkfail=1
                phyindex=1: mac=00:0c:29:70:b9:76, linkfail=1
                phyindex=2: mac=00:0c:29:70:b9:80, linkfail=1
                phyindex=3: mac=00:0c:29:70:b9:8a, linkfail=1
                phyindex=4: mac=00:0c:29:70:b9:94, linkfail=1
                phyindex=5: mac=00:0c:29:70:b9:9e, linkfail=1
                phyindex=6: mac=00:0c:29:70:b9:a8, linkfail=1
                phyindex=7: mac=00:0c:29:70:b9:b2, linkfail=1
                phyindex=8: mac=00:0c:29:70:b9:bc, linkfail=1
                phyindex=9: mac=00:0c:29:70:b9:c6, linkfail=1
        standalone_id=4:
                phyindex=0: mac=00:0c:29:5c:d3:23, linkfail=1
                phyindex=1: mac=00:0c:29:5c:d3:2d, linkfail=1
                phyindex=2: mac=00:0c:29:5c:d3:37, linkfail=1
                phyindex=3: mac=00:0c:29:5c:d3:41, linkfail=1
                phyindex=4: mac=00:0c:29:5c:d3:4b, linkfail=1
                phyindex=5: mac=00:0c:29:5c:d3:55, linkfail=1
                phyindex=6: mac=00:0c:29:5c:d3:5f, linkfail=1
```

```
                phyindex=7: mac=00:0c:29:5c:d3:69, linkfail=1
                phyindex=8: mac=00:0c:29:5c:d3:73, linkfail=1
                phyindex=9: mac=00:0c:29:5c:d3:7d, linkfail=1
        standalone_id=5:
        ...
        standalone_id=15:
```

2. Initiate a dialup tunnel connection from the IPsec Client 2 FortiGate (192.168.1.2).
3. Verify the tunnel list for vpn1_1 on each peer. The output shows the bi-directional SAs for that particular tunnel are synchronized to all participating FGSP peers.
   a. DC1_VM1:

```
DC1_VM1 # diagnose vpn tunnel list name vpn1_1
list ipsec tunnel by names in vd 0
------------------------------------------------------
name=vpn1_1 ver=2 serial=a4 192.168.202.31:0->192.168.1.2:0 tun_id=192.168.1.2 tun_
id6=::10.0.0.15 dst_mtu=1500 dpd-link=on weight=1
bound_if=6 lgwy=static/1 tun=intf/0 mode=dial_inst/3 encap=none/8840 options[2288]=npu
rgwy-chg frag-rfc  run_state=0 role=sync-primary accept_traffic=1 overlay_id=0

parent=vpn1 index=1
proxyid_num=1 child_num=0 refcnt=6 ilast=6 olast=6 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=1 idle=60000ms retry=3 count=0 seqno=20
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=vpn1 proto=0 sa=1 ref=2 serial=3 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.10.1.0-10.10.1.255:0
  SA:  ref=3 options=682 type=00 soft=0 mtu=1438 expire=10480/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=10788/10800
  dec: spi=a575b631 esp=aes key=16 5de449f75c7d70258f4972506dd164e2
       ah=sha1 key=20 7e65d641be6bc52655619ff542c67c61713de523
  enc: spi=10aa45b0 esp=aes key=16 65ad3b4849386deb4f3028079a657257
       ah=sha1 key=20 b5f1e1c6786f69482b5d271347a69a0cbb83ed58
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=192.168.1.2 npu_lgwy=192.168.202.31 npu_selid=b2 dec_npuid=0 enc_
npuid=0
```

   b. DC1_VM2:

```
DC1_VM2 # diagnose vpn tunnel list name vpn1_1
list ipsec tunnel by names in vd 0
------------------------------------------------------
name=vpn1_1 ver=2 serial=a3 192.168.202.31:0->192.168.1.2:0 tun_id=192.168.1.2 tun_
id6=::10.0.0.15 dst_mtu=0 dpd-link=on weight=1
bound_if=6 lgwy=static/1 tun=intf/0 mode=dial_inst/3 encap=none/8712 options[2208]=npu
frag-rfc  run_state=0 role=standby accept_traffic=1 overlay_id=0

parent=vpn1 index=1
```

```
proxyid_num=1 child_num=0 refcnt=6 ilast=43063501 olast=43063501 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=1 idle=60000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=vpn1 proto=0 sa=1 ref=2 serial=3 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.10.1.0-10.10.1.255:0
  SA:  ref=3 options=682 type=00 soft=0 mtu=1280 expire=10466/0B replaywin=2048
       seqno=10000001 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=10788/10800
  dec: spi=a575b631 esp=aes key=16 5de449f75c7d70258f4972506dd164e2
       ah=sha1 key=20 7e65d641be6bc52655619ff542c67c61713de523
  enc: spi=10aa45b0 esp=aes key=16 65ad3b4849386deb4f3028079a657257
       ah=sha1 key=20 b5f1e1c6786f69482b5d271347a69a0cbb83ed58
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=192.168.1.2 npu_lgwy=192.168.202.31 npu_selid=ab dec_npuid=0 enc_
npuid=0
```

c. DC1_VM3:

```
DC1_VM3 # diagnose vpn tunnel list name vpn1_1
list ipsec tunnel by names in vd 0
------------------------------------------------------
name=vpn1_1 ver=2 serial=ac 192.168.202.31:0->192.168.1.2:0 tun_id=192.168.1.2 tun_
id6=::10.0.0.15 dst_mtu=0 dpd-link=on weight=1
bound_if=6 lgwy=static/1 tun=intf/0 mode=dial_inst/3 encap=none/8712 options[2208]=npu
frag-rfc  run_state=0 role=standby accept_traffic=1 overlay_id=0

parent=vpn1 index=1
proxyid_num=1 child_num=0 refcnt=6 ilast=43063499 olast=43063499 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=1 idle=60000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=vpn1 proto=0 sa=1 ref=2 serial=2 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.10.1.0-10.10.1.255:0
  SA:  ref=3 options=682 type=00 soft=0 mtu=1280 expire=10462/0B replaywin=2048
       seqno=10000001 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=10788/10800
  dec: spi=a575b631 esp=aes key=16 5de449f75c7d70258f4972506dd164e2
       ah=sha1 key=20 7e65d641be6bc52655619ff542c67c61713de523
  enc: spi=10aa45b0 esp=aes key=16 65ad3b4849386deb4f3028079a657257
       ah=sha1 key=20 b5f1e1c6786f69482b5d271347a69a0cbb83ed58
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=192.168.1.2 npu_lgwy=192.168.202.31 npu_selid=b4 dec_npuid=0 enc_
npuid=0
```

d. DC1_VM4:

```
DC1_VM4 # diagnose vpn tunnel list name vpn1_1
list ipsec tunnel by names in vd 0
```

The IPsec tunnel `role=sync-primary`on DC1_VM1 indicates that the IPsec tunnel was established on the FortiGate and traffic is being forwarded. On DC1_VM2 and DC1_VM3, the IPsec tunnel `role=standby` indicates that they are synchronized from the FGSP peer and are in standby for traffic forwarding.

The IPsec SAs do not synchronize to DC1_VM4 because `fgsp-sync` is disabled.

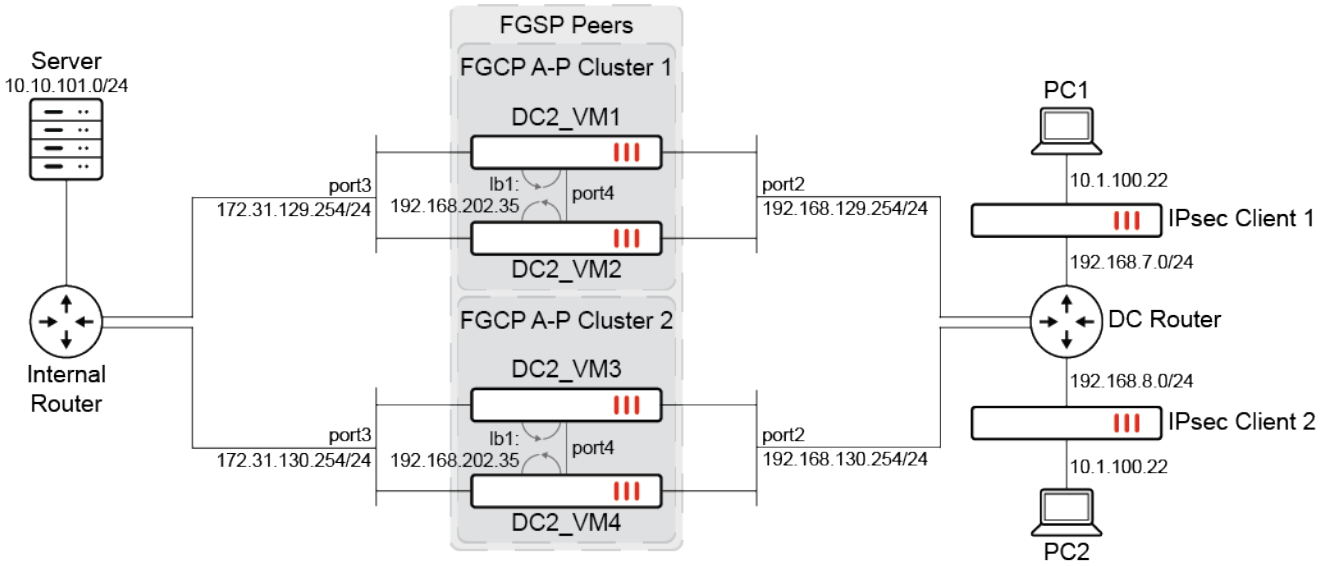
4. When a failure occurs on DC1_VM1, the tunnel traffic will fail over to either DC1_VM2 or DC1_VM3. Its tunnel role will become `role=sync-primary`.

FGCP over FGSP per-tunnel failover for IPsec

For additional redundancy, an FGCP cluster on one site may form FGSP peering with FGCP clusters on other sites. The FGCP over FGSP peers can still synchronize IPsec SAs and act as the primary gateway for individual tunnels for the same dialup servers. When failover happens within an FGCP cluster, tunnel traffic will failover to the other FGCP cluster member. When an FGCP cluster fails, tunnel traffic will failover to the other FGSP peer.

Example

In this example, each FGCP A-P cluster is connected on port4 as the heartbeat interface. The FGSP peers are connected on port5 over 172.31.2.1-2/24. Each FGSP peer and FGCP cluster has a loopback interface, lb1, with the same IP address. This loopback interface is used as the local gateway on each of the phase 1 connections to avoid each FGSP member having different IPs on port2. The DC Router uses ECMP to distribute traffic to each FGSP peer. It is assumed that the networking addresses are already configured properly.



Interface/setting	DC2_VM1	DC2_VM2	DC2_VM3	DC2_VM4
port2	192.168.129.254/24	192.168.129.254/24	192.168.130.254/24	192.168.130.254/24
port3	172.31.129.254/24	172.31.129.254/24	172.31.130.254/24	172.31.130.254/24
port4	FGCP HA heartbeat interface	FGCP HA heartbeat interface	FGCP HA heartbeat interface	FGCP HA heartbeat interface

Interface/setting	DC2_VM1	DC2_VM2	DC2_VM3	DC2_VM4
port5	172.31.2.1/24	172.31.2.1/24	172.31.2.2/24	172.31.2.2/24
lb1	192.168.202.35/32	192.168.202.35/32	192.168.202.35/32	192.168.205.35/32
fgsp-sync	Enabled	Enabled	Enabled	Enabled

There are two pairs of FGCP A-P HA clusters that form FGSP peering with each other. This is a typical FGCP over FGSP configuration used in large enterprises and service provider environments where high redundancy is needed. Each cluster uses the same loopback address for the local gateway. The DC Router uses ECMP to route traffic to the destination 192.168.202.31 through each of the participating FGSP peers.

In a larger scale there may be many more members in the FGCP clusters, more FGSP peers, and more IPsec dialup clients connecting. Each eligible FGSP peer will be the primary gateway for a set of dialup tunnels, and is in standby for the rest of the tunnels. When the FGCP cluster is configured in A-P mode, the tunnels will be established on the primary unit and synchronized to the standby unit.

The following configurations and example demonstrates PC1 initiating traffic to the Server. First, a dialup tunnel is formed between FortiGate IPsec Client 1 and DC2_VM1, which allows traffic to go through. IPsec SAs are synchronized to the FGCP standby unit, and to the FGSP peer. Upon failure of DC2_VM1, DC2_VM2 takes over as the primary of the HA cluster, and assumes the primary role for the failover tunnels.

If both DC2_VM1 and DC2_VM2 fail, the tunnels that were formed on this FGSP peer will now be re-routed to the other FGSP peer. The primary FGCP cluster member, DC2_VM3, will now pick up the tunnel traffic and assume the primary role for the failover tunnels.

To configure the HA clusters:

1. Configure FGCP A-P Cluster 1 (use the same configuration for DC2_VM1 and DC2_VM2):

```
config system ha
    set group-id 1
    set group-name "DC2_VM12"
    set mode a-p
    set password ********
    set hbdev "port4" 50
    set session-pickup enable
    set upgrade-mode simultaneous
    set override disable
    set priority 100
end
```

2. Configure FGCP A-P Cluster 2 (use the same configuration for DC2_VM3 and DC2_VM4):

```
config system ha
    set group-id 2
    set group-name "DC2_VM34"
    set mode a-p
    set password ********
    set hbdev "port4" 50
    set session-pickup enable
    set upgrade-mode simultaneou
    set override disable
```

```
    set priority 100
end
```

To configure the FGSP peers:

1. Configure DC2_VM1:

```
config system standalone-cluster
    set standalone-group-id 2
    set group-member-id 1
    config cluster-peer
        edit 1
            set peerip 172.31.2.2
        next
    end
end
```

   The configuration is automatically synchronized to DC2_VM2.

2. Configure DC2_VM3:

```
config system standalone-cluster
    set standalone-group-id 2
    set group-member-id 2
    config cluster-peer
        edit 1
            set peerip 172.31.2.1
        next
    end
end
```

   The configuration is automatically synchronized to DC2_VM4.

3. To configure the IPsec VPN settings (use the same configuration for DC2_VM1 and DC2_VM3).
   a. Configure the VPN tunnel phase 1 settings:

```
config vpn ipsec phase1-interface
    edit "vpn1"
        set type dynamic
        set interface "port2"
        set ike-version 2
        set local-gw 192.168.202.35
        set keylife 90000
        set peertype one
        set net-device disable
        set proposal aes128-sha1
        set add-route disable
        set dpd on-idle
        set dhgrp 2
        set fgsp-sync enable
        set nattraversal disable
        set peerid "Nokia_Peer"
        set psksecret ********
```

```
        set dpd-retryinterval 60
    next
end
```

b. Configure the VPN tunnel phase 2 settings:

```
config vpn ipsec phase2-interface
    edit "vpn1"
        set phase1name "vpn1"
        set proposal aes128-sha1
        set keylifeseconds 10800
    next
end
```

To verify the configuration:

1. The FGCP HA cluster and the FGSP peering have formed. Verify the respective HA statuses.
   a. Verify the FGCP cluster status on DC2_VM1:

```
DC2_VM1 # diagnose sys ha status

HA information
Statistics
        traffic.local = s:0 p:439253 b:89121494
        traffic.total = s:0 p:440309 b:89242174
        activity.ha_id_changes = 2
        activity.fdb  = c:0 q:0

Model=80006, Mode=2 Group=1 Debug=0
nvcluster=1, ses_pickup=1, delay=0

[Debug_Zone HA information]
HA group member information: is_manage_primary=1.
FGVM02TM22000002:      Primary, serialno_prio=0, usr_priority=100, hostname=DC2_VM2
FGVM02TM22000001:    Secondary, serialno_prio=1, usr_priority=200, hostname=DC2_VM1

[Kernel HA information]
vcluster 1, state=work, primary_ip=169.254.0.1, primary_id=0
FGVM02TM22000002:      Primary, ha_prio/o_ha_prio=0/0
FGVM02TM22000001:    Secondary, ha_prio/o_ha_prio=1/1
```

   b. Verify the FGSP peering status on DC2_VM1:

```
DC2_VM1 # diagnose sys ha standalone-peers
Group=2, ID=1
Detected-peers=1
Kernel standalone-peers: num=1.
peer0: vfid=0, peerip:port = 172.31.2.2:708, standalone_id=2
        session-type: send=3, recv=4
         packet-type: send=0, recv=0
Kernel standalone dev_base:
        standalone_id=0:
```

```
        standalone_id=1:
                phyindex=0: mac=00:0c:29:fc:a3:17, linkfail=1
                phyindex=1: mac=00:0c:29:fc:a3:21, linkfail=1
                phyindex=2: mac=00:0c:29:fc:a3:2b, linkfail=1
                phyindex=3: mac=00:0c:29:fc:a3:35, linkfail=1
                phyindex=4: mac=00:0c:29:fc:a3:3f, linkfail=1
                phyindex=5: mac=00:0c:29:fc:a3:49, linkfail=1
                phyindex=6: mac=00:0c:29:fc:a3:53, linkfail=1
                phyindex=7: mac=00:0c:29:fc:a3:5d, linkfail=1
                phyindex=8: mac=00:0c:29:fc:a3:67, linkfail=1
                phyindex=9: mac=00:0c:29:fc:a3:71, linkfail=1
        standalone_id=2:
                phyindex=0: mac=00:09:0f:09:02:00, linkfail=1
                phyindex=1: mac=00:09:0f:09:02:01, linkfail=1
                phyindex=2: mac=00:09:0f:09:02:02, linkfail=1
                phyindex=3: mac=00:09:0f:09:02:03, linkfail=1
                phyindex=4: mac=00:09:0f:09:02:04, linkfail=1
                phyindex=5: mac=00:09:0f:09:02:05, linkfail=1
                phyindex=6: mac=00:09:0f:09:02:06, linkfail=1
                phyindex=7: mac=00:09:0f:09:02:07, linkfail=1
                phyindex=8: mac=00:09:0f:09:02:08, linkfail=1
                phyindex=9: mac=00:09:0f:09:02:09, linkfail=1
        standalone_id=3:
        ...
        standalone_id=15:
```

c. Verify the FGCP cluster status on DC2_VM3:

```
DC2_VM3 # diagnose sys ha status
HA information
Statistics
        traffic.local = s:0 p:443999 b:89037989
        traffic.total = s:0 p:445048 b:89157373
        activity.ha_id_changes = 2
        activity.fdb  = c:0 q:0

Model=80006, Mode=2 Group=2 Debug=0
nvcluster=1, ses_pickup=1, delay=0

[Debug_Zone HA information]
HA group member information: is_manage_primary=1.
FGVM02TM22000004:      Primary, serialno_prio=0, usr_priority=100, hostname=DC2_VM4
FGVM02TM22000003:    Secondary, serialno_prio=1, usr_priority=200, hostname=DC2_VM3

[Kernel HA information]
vcluster 1, state=work, primary_ip=169.254.0.1, primary_id=0
FGVM02TM22000004:      Primary, ha_prio/o_ha_prio=0/0
FGVM02TM22000003:    Secondary, ha_prio/o_ha_prio=1/1
```

d. Verify the FGSP peering status on DC2_VM3:

```
DC2_VM3 # diagnose sys ha standalone-peers
Group=2, ID=2
```

```
Detected-peers=1
Kernel standalone-peers: num=1.
peer0: vfid=0, peerip:port = 172.31.2.1:708, standalone_id=1
        session-type: send=2, recv=6
         packet-type: send=0, recv=0
Kernel standalone dev_base:
        standalone_id=0:
        standalone_id=1:
                phyindex=0: mac=00:09:0f:09:01:00, linkfail=1
                phyindex=1: mac=00:09:0f:09:01:01, linkfail=1
                phyindex=2: mac=00:09:0f:09:01:02, linkfail=1
                phyindex=3: mac=00:09:0f:09:01:03, linkfail=1
                phyindex=4: mac=00:09:0f:09:01:04, linkfail=1
                phyindex=5: mac=00:09:0f:09:01:05, linkfail=1
                phyindex=6: mac=00:09:0f:09:01:06, linkfail=1
                phyindex=7: mac=00:09:0f:09:01:07, linkfail=1
                phyindex=8: mac=00:09:0f:09:01:08, linkfail=1
                phyindex=9: mac=00:09:0f:09:01:09, linkfail=1
        standalone_id=2:
                phyindex=0: mac=00:0c:29:bb:77:af, linkfail=1
                phyindex=1: mac=00:0c:29:bb:77:b9, linkfail=1
                phyindex=2: mac=00:0c:29:bb:77:c3, linkfail=1
                phyindex=3: mac=00:0c:29:bb:77:cd, linkfail=1
                phyindex=4: mac=00:0c:29:bb:77:d7, linkfail=1
                phyindex=5: mac=00:0c:29:bb:77:e1, linkfail=1
                phyindex=6: mac=00:0c:29:bb:77:eb, linkfail=1
                phyindex=7: mac=00:0c:29:bb:77:f5, linkfail=1
                phyindex=8: mac=00:0c:29:bb:77:ff, linkfail=1
                phyindex=9: mac=00:0c:29:bb:77:09, linkfail=1
        standalone_id=3:
        ...
        standalone_id=15:
```

2. Initiate traffic from PC1 to the Server. This initiates a tunnel from the IPsec Client 1 FortiGate to DC2_VM1.
3. Verify the tunnel list for vpn1_1 on each peer.
   a. DC2_VM1:

```
DC2_VM1 # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=vpn1_1 ver=2 serial=4 192.168.202.35:0->192.168.7.2:0 tun_id=192.168.7.2 tun_
id6=::10.0.0.4 dst_mtu=1500 dpd-link=on weight=1
bound_if=6 lgwy=static/1 tun=intf/0 mode=dial_inst/3 encap=none/8840 options[2288]=npu
rgwy-chg frag-rfc  run_state=0 role=sync-primary accept_traffic=1 overlay_id=0

parent=vpn1 index=1
proxyid_num=1 child_num=0 refcnt=5 ilast=41 olast=41 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=1 idle=60000ms retry=3 count=0 seqno=156
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
```

```
proxyid=vpn1 proto=0 sa=1 ref=2 serial=1
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.10.1.0-10.10.1.255:0
  SA:  ref=3 options=602 type=00 soft=0 mtu=1438 expire=1424/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=10791/10800
  dec: spi=37f426a1 esp=aes key=16 3671c9303b6295fc73b11765811bdf96
       ah=sha1 key=20 41b98cb541dc9c76311ddec4b23584ee35d31915
  enc: spi=10aa4d3a esp=aes key=16 cc8529ee16de6e4ac42b0ce506d7cdd1
       ah=sha1 key=20 0c2d9edd0fdbe45942cf718ac2ebb4d59c2760c6
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=192.168.7.2 npu_lgwy=192.168.202.35 npu_selid=1c dec_npuid=0 enc_
npuid=0
```

b. DC2_VM2:

```
DC2_VM2 # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=vpn1_1 ver=2 serial=4 192.168.202.35:0->192.168.7.2:0 tun_id=192.168.7.2 tun_
id6=::10.0.0.4 dst_mtu=0 dpd-link=on weight=1
bound_if=6 lgwy=static/1 tun=intf/0 mode=dial_inst/3 encap=none/8712 options[2208]=npu
frag-rfc  run_state=0 role=standby accept_traffic=1 overlay_id=0

parent=vpn1 index=1
proxyid_num=1 child_num=0 refcnt=5 ilast=42975898 olast=42975898 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=1 idle=60000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=vpn1 proto=0 sa=1 ref=2 serial=1
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.10.1.0-10.10.1.255:0
  SA:  ref=3 options=602 type=00 soft=0 mtu=1280 expire=1325/0B replaywin=2048
       seqno=10000001 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=10791/10800
  dec: spi=37f426a1 esp=aes key=16 3671c9303b6295fc73b11765811bdf96
       ah=sha1 key=20 41b98cb541dc9c76311ddec4b23584ee35d31915
  enc: spi=10aa4d3a esp=aes key=16 cc8529ee16de6e4ac42b0ce506d7cdd1
       ah=sha1 key=20 0c2d9edd0fdbe45942cf718ac2ebb4d59c2760c6
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=192.168.7.2 npu_lgwy=192.168.202.35 npu_selid=1c dec_npuid=0 enc_
npuid=0
```

c. DC2_VM3:

```
DC2_VM3 # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=vpn1_1 ver=2 serial=4 192.168.202.35:0->192.168.7.2:0 tun_id=192.168.7.2 tun_
id6=::10.0.0.4 dst_mtu=0 dpd-link=on weight=1
bound_if=6 lgwy=static/1 tun=intf/0 mode=dial_inst/3 encap=none/8712 options[2208]=npu
frag-rfc  run_state=0 role=standby accept_traffic=1 overlay_id=0
```

```
parent=vpn1 index=1
proxyid_num=1 child_num=0 refcnt=5 ilast=42975982 olast=42975982 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=1 idle=60000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=vpn1 proto=0 sa=1 ref=2 serial=1
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.10.1.0-10.10.1.255:0
  SA:  ref=3 options=602 type=00 soft=0 mtu=1280 expire=1215/0B replaywin=2048
       seqno=10000001 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=10791/10800
  dec: spi=37f426a1 esp=aes key=16 3671c9303b6295fc73b11765811bdf96
       ah=sha1 key=20 41b98cb541dc9c76311ddec4b23584ee35d31915
  enc: spi=10aa4d3a esp=aes key=16 cc8529ee16de6e4ac42b0ce506d7cdd1
       ah=sha1 key=20 0c2d9edd0fdbe45942cf718ac2ebb4d59c2760c6
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=192.168.7.2 npu_lgwy=192.168.202.35 npu_selid=1c dec_npuid=0 enc_
npuid=0
```

d. DC2_VM4:

```
DC2_VM4 # diagnose vpn  tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=vpn1_1 ver=2 serial=4 192.168.202.35:0->192.168.7.2:0 tun_id=192.168.7.2 tun_
id6=::10.0.0.4 dst_mtu=0 dpd-link=on weight=1
bound_if=6 lgwy=static/1 tun=intf/0 mode=dial_inst/3 encap=none/8712 options[2208]=npu
frag-rfc  run_state=0 role=standby accept_traffic=1 overlay_id=0

parent=vpn1 index=1
proxyid_num=1 child_num=0 refcnt=5 ilast=42975768 olast=42975768 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=1 idle=60000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=vpn1 proto=0 sa=1 ref=2 serial=1
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.10.1.0-10.10.1.255:0
  SA:  ref=3 options=602 type=00 soft=0 mtu=1280 expire=1433/0B replaywin=2048
       seqno=10000001 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=10791/10800
  dec: spi=37f426a1 esp=aes key=16 3671c9303b6295fc73b11765811bdf96
       ah=sha1 key=20 41b98cb541dc9c76311ddec4b23584ee35d31915
  enc: spi=10aa4d3a esp=aes key=16 cc8529ee16de6e4ac42b0ce506d7cdd1
       ah=sha1 key=20 0c2d9edd0fdbe45942cf718ac2ebb4d59c2760c6
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=192.168.7.2 npu_lgwy=192.168.202.35 npu_selid=1c dec_npuid=0 enc_
npuid=0
```

The IPsec tunnel `role=sync-primary`on DC2_VM1 indicates that it is being used to carry IPsec traffic. On DC2_VM2, DC2_VM3, and DC2_VM4, the IPsec tunnel `role=standby` indicates that they are in

standby for traffic forwarding.

To test failover scenarios:

1. Verify the sniffer trace on DC2_VM1 before FGCP HA failover:

```
DC2_VM1 # diagnose sniffer packet any icmp 4
Using Original Sniffing Mode
interfaces=[any]
filters=[icmp]
0.171753 vpn1 in 10.10.1.2 -> 10.10.101.2: icmp: echo request
0.171763 port3 out 10.10.1.2 -> 10.10.101.2: icmp: echo request
0.171941 port3 in 10.10.101.2 -> 10.10.1.2: icmp: echo reply
0.171947 vpn1 out 10.10.101.2 -> 10.10.1.2: icmp: echo reply
```

   Traffic passes through DC2_VM1.
2. Reboot the primary FortiGate, DC2_VM1.
3. Verify the sniffer trace on DC2_VM2 after FGCP HA failover:

```
DC2_VM2 # diagnose sniffer packet any icmp 4
Using Original Sniffing Mode
interfaces=[any]
filters=[icmp]
0.111107 vpn1 in 10.10.1.2 -> 10.10.101.2: icmp: echo request
0.111118 port3 out 10.10.1.2 -> 10.10.101.2: icmp: echo request
0.111293 port3 in 10.10.101.2 -> 10.10.1.2: icmp: echo reply
0.111298 vpn1 out 10.10.101.2 -> 10.10.1.2: icmp: echo reply
^C
16 packets received by filter
0 packets dropped by kernel
```

   Traffic passes through DC2_VM2.
4. Verify the tunnel list for vpn1_1 on DC2_VM2:

```
DC2_VM2 # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=vpn1_1 ver=2 serial=4 192.168.202.35:0->192.168.7.2:0 tun_id=192.168.7.2 tun_
id6=::10.0.0.4 dst_mtu=1500 dpd-link=on weight=1
bound_if=6 lgwy=static/1 tun=intf/0 mode=dial_inst/3 encap=none/8840 options[2288]=npu rgwy-
chg frag-rfc  run_state=0 role=sync-primary accept_traffic=1 overlay_id=0

parent=vpn1 index=1
proxyid_num=1 child_num=0 refcnt=5 ilast=0 olast=0 ad=/0
stat: rxp=58 txp=31 rxb=4872 txb=2604
dpd: mode=on-idle on=1 idle=60000ms retry=3 count=0 seqno=169
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=vpn1 proto=0 sa=1 ref=3 serial=3
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.10.1.0-10.10.1.255:0
  SA:  ref=3 options=602 type=00 soft=0 mtu=1438 expire=10730/0B replaywin=2048
```

```
      seqno=20 esn=0 replaywin_lastseq=0000003b qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=10790/10800
  dec: spi=37f426c1 esp=aes key=16 ef61b49078b6ab3e00a4d3a048d779f5
       ah=sha1 key=20 ee2e8de9c522d89b6481c37faa73a7bb54163645
  enc: spi=10aa4d58 esp=aes key=16 4cb95f12657ca8e269b9f8a25f9b19c1
       ah=sha1 key=20 326744c4e5b4a0758397725464593d94ba9390dc
  dec:pkts/bytes=116/9744, enc:pkts/bytes=62/7316
  npu_flag=00 npu_rgwy=192.168.7.2 npu_lgwy=192.168.202.35 npu_selid=1e dec_npuid=0 enc_
npuid=0
```

The role has changed to `role=sync-primary`.

5. Shut down DC2_VM1 and the DC2_VM2 IPsec uplink interface.
6. Verify the sniffer trace on DC2_VM3. As expected, traffic now passes through DC2_VM3:

```
DC2_VM3 # diagnose sniffer packet any icmp 4
Using Original Sniffing Mode
interfaces=[any]
filters=[icmp]
0.165088 vpn1 in 10.10.1.2 -> 10.10.101.2: icmp: echo request
0.165102 port3 out 10.10.1.2 -> 10.10.101.2: icmp: echo request
0.165294 port3 in 10.10.101.2 -> 10.10.1.2: icmp: echo reply
0.165301 vpn1 out 10.10.101.2 -> 10.10.1.2: icmp: echo reply
^C
14 packets received by filter
0 packets dropped by kernel
```

7. Verify the tunnel list for vpn1_1 on DC2_VM3:

```
DC2_VM3 # diagnose vpn tunnel list
list all ipsec tunnel in vd 0
------------------------------------------------------
name=vpn1_1 ver=2 serial=4 192.168.202.35:0->192.168.7.2:0 tun_id=192.168.7.2 tun_
id6=::10.0.0.4 dst_mtu=1500 dpd-link=on weight=1
bound_if=6 lgwy=static/1 tun=intf/0 mode=dial_inst/3 encap=none/8712 options[2208]=npu frag-
rfc  run_state=0 role=sync-primary accept_traffic=1 overlay_id=0

parent=vpn1 index=1
proxyid_num=1 child_num=0 refcnt=5 ilast=0 olast=0 ad=/0
stat: rxp=53 txp=53 rxb=4452 txb=4452
dpd: mode=on-idle on=1 idle=60000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=vpn1 proto=0 sa=1 ref=3 serial=3
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.10.1.0-10.10.1.255:0
  SA:  ref=3 options=602 type=00 soft=0 mtu=1438 expire=10347/0B replaywin=2048
       seqno=10000155 esn=0 replaywin_lastseq=000001b0 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=10790/10800
  dec: spi=37f426c1 esp=aes key=16 ef61b49078b6ab3e00a4d3a048d779f5
       ah=sha1 key=20 ee2e8de9c522d89b6481c37faa73a7bb54163645
  enc: spi=10aa4d58 esp=aes key=16 4cb95f12657ca8e269b9f8a25f9b19c1
       ah=sha1 key=20 326744c4e5b4a0758397725464593d94ba9390dc
```

```
  dec:pkts/bytes=88/7392, enc:pkts/bytes=88/10384
  npu_flag=00 npu_rgwy=192.168.7.2 npu_lgwy=192.168.202.35 npu_selid=1e dec_npuid=0 enc_
npuid=0
```

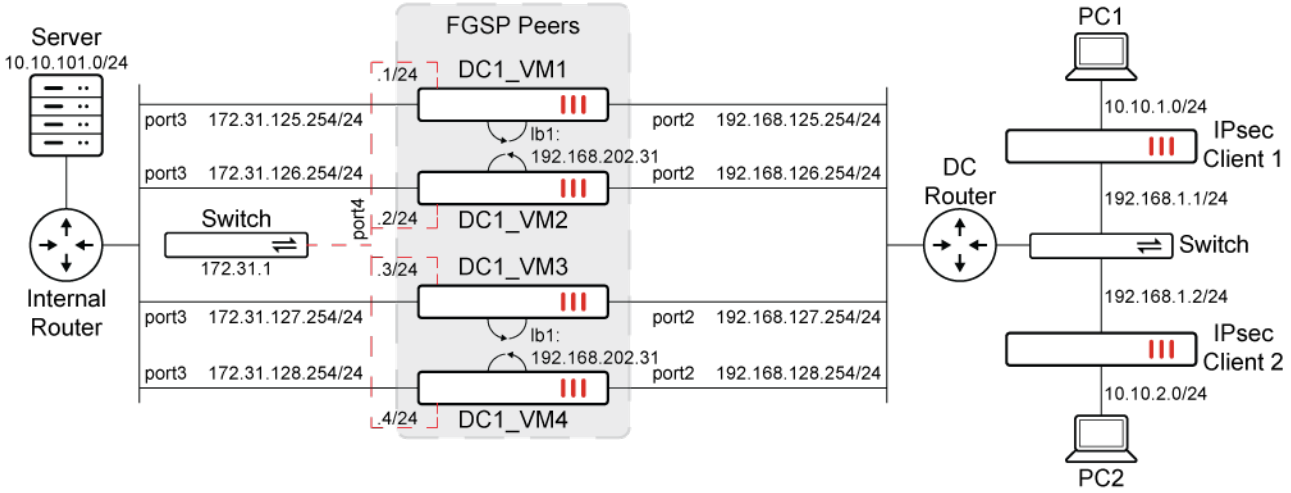
The role has changed to `role=sync-primary`.

Allow IPsec DPD in FGSP members to support failovers

In conjunction with support for FGSP per-tunnel failover for IPsec on page 4254, configuring DPD (dead peer detection) on an FGSP member is permitted. This allows a failed FGSP member to send out DPD probes during failover to detect unreachable remote peers and to flush the corresponding tunnels.

Example

In this example, using the same configuration as in FGSP per-tunnel failover for IPsec on page 4254, a tunnel can be established from one of the remote IPsec clients to one of the FGSP members (DC1_VM1). DPD can be set to `on-idle`, with a configured `dpd-retryinterval` of 60 seconds. When a client disappears, whether it is due to remote client failures or server-side routing failures, the FGSP member or gateway (DC1_VM1) will send out DPD probes for detection. Once the three iterations are complete and no responses are detected, the FGSP member will flush the tunnel and remove any routing to that peer.



Interface/setting	DC1_VM1	DC1_VM2	DC1_VM3	DC1_VM4
port2	192.168.125.254/24	192.168.126.254/24	192.168.127.254/24	192.168.128.254/24
port3	172.31.125.254/24	172.31.126.254/24	172.31.127.254/24	172.31.128.254/24
port4	172.31.1.1/24	172.31.1.2/24	172.31.1.3/24	172.31.1.4/24
lb1	192.168.202.31/32	192.168.202.31/32	192.168.202.31/32	192.168.202.31/32
`fgsp-sync`	Enabled	Enabled	Enabled	Disabled

To configure the FGSP peers (DC1_VM1):

The following steps are to configure DC1_VM1. The other peers have similar configurations based on the preceding table. In the `config vpn ipsec phase1-interface` settings, all peers should have the same local gateway external interface (192.168.202.31). For DC1_VM4, `fgsp-sync` is disabled in the VPN tunnel phase 1 settings.

1. Configure the FGSP settings:

```
config system standalone-cluster
    set standalone-group-id 1
    set group-member-id 1
    config cluster-peer
        edit 1
            set peerip 172.31.1.2
        next
        edit 2
            set peerip 172.31.1.3
        next
        edit 3
            set peerip 172.31.1.4
        next
    end
end
```

2. Configure the VPN tunnel phase 1 settings:

```
config vpn ipsec phase1-interface
    edit "vpn1"
        set type dynamic
        set interface "port2"
        set ike-version 2
        set local-gw 192.168.202.31
        set keylife 90000
        set peertype one
        set net-device disable
        set proposal aes128-sha1
        set dpd on-idle
        set dhgrp 2
        set fgsp-sync enable
        set nattraversal disable
        set peerid "Nokia_Peer"
        set psksecret xxxxx
        set dpd-retryinterval 60
    next
end
```

3. Configure the VPN tunnel phase 2 settings:

```
config vpn ipsec phase2-interface
    edit "vpn1"
        set phase1name "vpn1"
```

```
        set proposal aes128-sha1
        set keylifeseconds 10800
    next
end
```

To verify the configuration:

1. Once the FGSP members establish peering with each other, verify the standalone peers on DC1_VM1:

```
DC1_VM1 # diagnose sys ha standalone-peers
Group=1, ID=1
Detected-peers=3
Kernel standalone-peers: num=3.
peer0: vfid=0, peerip:port = 172.31.1.2:708, standalone_id=2
        session-type: send=0, recv=0
         packet-type: send=0, recv=0
peer1: vfid=0, peerip:port = 172.31.1.3:708, standalone_id=3
        session-type: send=0, recv=0
         packet-type: send=0, recv=0
peer2: vfid=0, peerip:port = 172.31.1.4:708, standalone_id=4
        session-type: send=0, recv=0
         packet-type: send=0, recv=0
Kernel standalone dev_base:
        standalone_id=0:
        standalone_id=1:
                phyindex=0: mac=00:0c:29:22:00:6b, linkfail=1
                phyindex=1: mac=00:0c:29:22:00:75, linkfail=1
                phyindex=2: mac=00:0c:29:22:00:7f, linkfail=1
                phyindex=3: mac=00:0c:29:22:00:89, linkfail=1
                phyindex=4: mac=00:0c:29:22:00:93, linkfail=1
                phyindex=5: mac=00:0c:29:22:00:9d, linkfail=1
                phyindex=6: mac=00:0c:29:22:00:a7, linkfail=1
                phyindex=7: mac=00:0c:29:22:00:b1, linkfail=1
                phyindex=8: mac=00:0c:29:22:00:bb, linkfail=1
                phyindex=9: mac=00:0c:29:22:00:c5, linkfail=1
        standalone_id=2:
                phyindex=0: mac=00:0c:29:06:4e:d6, linkfail=1
                phyindex=1: mac=00:0c:29:06:4e:e0, linkfail=1
                phyindex=2: mac=00:0c:29:06:4e:ea, linkfail=1
                phyindex=3: mac=00:0c:29:06:4e:f4, linkfail=1
                phyindex=4: mac=00:0c:29:06:4e:fe, linkfail=1
                phyindex=5: mac=00:0c:29:06:4e:08, linkfail=1
                phyindex=6: mac=00:0c:29:06:4e:12, linkfail=1
                phyindex=7: mac=00:0c:29:06:4e:1c, linkfail=1
                phyindex=8: mac=00:0c:29:06:4e:26, linkfail=1
                phyindex=9: mac=00:0c:29:06:4e:30, linkfail=1
        standalone_id=3:
                phyindex=0: mac=00:0c:29:70:b9:6c, linkfail=1
                phyindex=1: mac=00:0c:29:70:b9:76, linkfail=1
                phyindex=2: mac=00:0c:29:70:b9:80, linkfail=1
                phyindex=3: mac=00:0c:29:70:b9:8a, linkfail=1
                phyindex=4: mac=00:0c:29:70:b9:94, linkfail=1
```

```
                phyindex=5: mac=00:0c:29:70:b9:9e, linkfail=1
                phyindex=6: mac=00:0c:29:70:b9:a8, linkfail=1
                phyindex=7: mac=00:0c:29:70:b9:b2, linkfail=1
                phyindex=8: mac=00:0c:29:70:b9:bc, linkfail=1
                phyindex=9: mac=00:0c:29:70:b9:c6, linkfail=1
        standalone_id=4:
                phyindex=0: mac=00:0c:29:5c:d3:23, linkfail=1
                phyindex=1: mac=00:0c:29:5c:d3:2d, linkfail=1
                phyindex=2: mac=00:0c:29:5c:d3:37, linkfail=1
                phyindex=3: mac=00:0c:29:5c:d3:41, linkfail=1
                phyindex=4: mac=00:0c:29:5c:d3:4b, linkfail=1
                phyindex=5: mac=00:0c:29:5c:d3:55, linkfail=1
                phyindex=6: mac=00:0c:29:5c:d3:5f, linkfail=1
                phyindex=7: mac=00:0c:29:5c:d3:69, linkfail=1
                phyindex=8: mac=00:0c:29:5c:d3:73, linkfail=1
                phyindex=9: mac=00:0c:29:5c:d3:7d, linkfail=1
        standalone_id=5:
        ...
        standalone_id=15:
```

2. Initiate a dialup tunnel connection from the IPsec Client 2 FortiGate (192.168.1.2).
3. Verify the tunnel list for vpn1_1 on each peer.
   a. DC1_VM1:

```
DC1_VM1 # diagnose vpn tunnel list name vpn1_1
list ipsec tunnel by names in vd 0
------------------------------------------------------
name=vpn1_1 ver=2 serial=a4 192.168.202.31:0->192.168.1.2:0 tun_id=192.168.1.2 tun_
id6=::10.0.0.15 dst_mtu=1500 dpd-link=on weight=1
bound_if=6 lgwy=static/1 tun=intf/0 mode=dial_inst/3 encap=none/8840 options[2288]=npu
rgwy-chg frag-rfc  run_state=0 role=sync-primary accept_traffic=1 overlay_id=0

parent=vpn1 index=1
proxyid_num=1 child_num=0 refcnt=6 ilast=6 olast=6 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=1 idle=60000ms retry=3 count=0 seqno=20
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=vpn1 proto=0 sa=1 ref=2 serial=3 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.10.1.0-10.10.1.255:0
  SA:  ref=3 options=682 type=00 soft=0 mtu=1438 expire=10480/0B replaywin=2048
       seqno=1 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=10788/10800
  dec: spi=a575b631 esp=aes key=16 5de449f75c7d70258f4972506dd164e2
       ah=sha1 key=20 7e65d641be6bc52655619ff542c67c61713de523
  enc: spi=10aa45b0 esp=aes key=16 65ad3b4849386deb4f3028079a657257
       ah=sha1 key=20 b5f1e1c6786f69482b5d271347a69a0cbb83ed58
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=192.168.1.2 npu_lgwy=192.168.202.31 npu_selid=b2 dec_npuid=0 enc_
npuid=0
```

b. DC1_VM2:

```
DC1_VM2 # diagnose vpn tunnel list name vpn1_1
list ipsec tunnel by names in vd 0
------------------------------------------------------
name=vpn1_1 ver=2 serial=a3 192.168.202.31:0->192.168.1.2:0 tun_id=192.168.1.2 tun_
id6=::10.0.0.15 dst_mtu=0 dpd-link=on weight=1
bound_if=6 lgwy=static/1 tun=intf/0 mode=dial_inst/3 encap=none/8712 options[2208]=npu
frag-rfc  run_state=0 role=standby accept_traffic=1 overlay_id=0

parent=vpn1 index=1
proxyid_num=1 child_num=0 refcnt=6 ilast=43063501 olast=43063501 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=1 idle=60000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=vpn1 proto=0 sa=1 ref=2 serial=3 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.10.1.0-10.10.1.255:0
  SA:  ref=3 options=682 type=00 soft=0 mtu=1280 expire=10466/0B replaywin=2048
       seqno=10000001 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=10788/10800
  dec: spi=a575b631 esp=aes key=16 5de449f75c7d70258f4972506dd164e2
       ah=sha1 key=20 7e65d641be6bc52655619ff542c67c61713de523
  enc: spi=10aa45b0 esp=aes key=16 65ad3b4849386deb4f3028079a657257
       ah=sha1 key=20 b5f1e1c6786f69482b5d271347a69a0cbb83ed58
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=192.168.1.2 npu_lgwy=192.168.202.31 npu_selid=ab dec_npuid=0 enc_
npuid=0
```

c. DC1_VM3:

```
DC1_VM3 # diagnose vpn tunnel list name vpn1_1
list ipsec tunnel by names in vd 0
------------------------------------------------------
name=vpn1_1 ver=2 serial=ac 192.168.202.31:0->192.168.1.2:0 tun_id=192.168.1.2 tun_
id6=::10.0.0.15 dst_mtu=0 dpd-link=on weight=1
bound_if=6 lgwy=static/1 tun=intf/0 mode=dial_inst/3 encap=none/8712 options[2208]=npu
frag-rfc  run_state=0 role=standby accept_traffic=1 overlay_id=0

parent=vpn1 index=1
proxyid_num=1 child_num=0 refcnt=6 ilast=43063499 olast=43063499 ad=/0
stat: rxp=0 txp=0 rxb=0 txb=0
dpd: mode=on-idle on=1 idle=60000ms retry=3 count=0 seqno=0
natt: mode=none draft=0 interval=0 remote_port=0
fec: egress=0 ingress=0
proxyid=vpn1 proto=0 sa=1 ref=2 serial=2 add-route
  src: 0:0.0.0.0-255.255.255.255:0
  dst: 0:10.10.1.0-10.10.1.255:0
  SA:  ref=3 options=682 type=00 soft=0 mtu=1280 expire=10462/0B replaywin=2048
       seqno=10000001 esn=0 replaywin_lastseq=00000000 qat=0 rekey=0 hash_search_len=1
  life: type=01 bytes=0/0 timeout=10788/10800
  dec: spi=a575b631 esp=aes key=16 5de449f75c7d70258f4972506dd164e2
```

```
      ah=sha1 key=20 7e65d641be6bc52655619ff542c67c61713de523
  enc: spi=10aa45b0 esp=aes key=16 65ad3b4849386deb4f3028079a657257
      ah=sha1 key=20 b5f1e1c6786f69482b5d271347a69a0cbb83ed58
  dec:pkts/bytes=0/0, enc:pkts/bytes=0/0
  npu_flag=00 npu_rgwy=192.168.1.2 npu_lgwy=192.168.202.31 npu_selid=b4 dec_npuid=0 enc_
npuid=0
```

4. When a shut down occurs on the VPN client to vpn1_2, verify the IKE debug messages on DC1_VM2. There are three iterations of DPD probes:

```
DC1_VM2 # diagnose debug enable
DC1_VM2 # diagnose debug application ike -1
...
ike 0:vpn1_2: link is idle 6 192.168.202.31->192.168.4.2:0 dpd=1 seqno=72 rr=0
ike 0:vpn1_2:171: send IKEv2 DPD probe, seqno 114
ike 0:vpn1_2:158: sending NOTIFY msg
ike 0:vpn1_2:171:158: send informational
ike 0:vpn1_2:171: sent IKE msg (INFORMATIONAL): 192.168.202.31:500->192.168.4.2:500, len=76,
vrf=0, id=87458c81a3be17f9/c8db7d3f2c70e638:00000004
ike 0: comes 192.168.1.2:500->192.168.202.31:500,ifindex=6,vrf=0...
ike 0:vpn1_2: link is idle 6 192.168.202.31->192.168.4.2:0 dpd=1 seqno=72 rr=0
ike 0:vpn1_2:171: send IKEv2 DPD probe, seqno 114
ike 0:vpn1_2:158: sending NOTIFY msg
ike 0:vpn1_2:171:158: send informational
ike 0:vpn1_2:171: sent IKE msg (INFORMATIONAL): 192.168.202.31:500->192.168.4.2:500, len=76,
vrf=0, id=87458c81a3be17f9/c8db7d3f2c70e638:00000004
ike 0: comes 192.168.1.2:500->192.168.202.31:500,ifindex=6,vrf=0....
ike 0:vpn1_2: link is idle 6 192.168.202.31->192.168.4.2:0 dpd=1 seqno=72 rr=0
ike 0:vpn1_2:171: send IKEv2 DPD probe, seqno 114
ike 0: comes 192.168.1.2:500->192.168.202.31:500,ifindex=6,vrf=0....
ike 0:vpn1_2:171: 87458c81a3be17f9/c8db7d3f2c70e638 negotiation of IKE SA failed due to retry
timeout
ike 0:vpn1_2:171: expiring IKE SA 87458c81a3be17f9/c8db7d3f2c70e638
ike 0:vpn1_2: deleting
ike 0:vpn1_2: flushing
ike 0:vpn1_2: deleting IPsec SA with SPI 85700354
ike 0:vpn1_2:vpn1: deleted IPsec SA with SPI 85700354, SA count: 0
ike 0:vpn1_2: sending SNMP tunnel DOWN trap for vpn1
ike 0:vpn1_2: sending tunnel down event for addr 10.10.4.0
ike 0:vpn1_2:vpn1: delete
ike 0:vpn1:152: del route 10.10.4.0/255.255.255.0 tunnel 192.168.4.2 oif vpn1(21) metric 15
priority 1
ike 0:vpn1_2: flushed
ike 0:vpn1_2:171: HA send IKE SA del 87458c81a3be17f9/c8db7d3f2c70e638
ike 0:vpn1_2:171:159: send informational
ike 0:vpn1_2:171: sent IKE msg (INFORMATIONAL): 192.168.202.31:500->192.168.4.2:500, len=76,
vrf=0, id=87458c81a3be17f9/c8db7d3f2c70e638:00000005
ike 0:vpn1_2: delete dynamic
ike 0:vpn1_2: deleted
```

Standalone configuration synchronization

You can configure synchronization from one standalone FortiGate to another standalone FortiGate (`standalone-config-sync`). With the exception of some configurations that do not sync (settings that identify the FortiGate to the network), the rest of the configurations are synced, such as firewall policies, firewall addresses, and UTM profiles.

This option is useful in situations when you need to set up FGSP peers, or when you want to quickly deploy several FortiGates with the same configurations. You can set up `standalone-config-sync` for multiple members.

> ⚠ `standalone-config-sync` is an independent feature and should be used with caution as there are some limitations. We recommend disabling it once the configurations have been synced over.

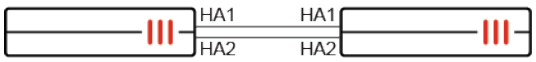
Limitations

When standalone configuration synchronization is enabled, there are some limitations, including but not limited to the following:

- Network interruptions occur during firmware upgrades: when upgrading the firmware, all members in the `standalone-config-sync` group are upgraded simultaneously. This creates downtime if the FortiGates are the only outgoing gateway in the network. We recommend disabling the option before upgrading firmware.
- Some unwanted configurations might be synced: the current design and implementation of `standalone-config-sync` is based on requirements from specific customers. Thus, some users may find that unwanted parts of the configurations are synced. Should this occur, we recommend disabling the option and modifying those configurations manually.
- The wrong primary device might be selected accidentally: `standalone-config-sync` is derived from the HA primary unit selection mechanism. All members in the group will join the selection process in the same way as a the HA cluster selection process. It is important to select the correct device as the primary, otherwise the wrong device could be selected and existing configurations could be overwritten.

Setting up standalone configuration synchronization

Two or more standalone FortiGates should be connected to each other with one or more heartbeat interfaces, either back-to-back or via a switch. In the following example, the device supplying the configurations is called "conf-prim," and the devices receiving the configurations are called "conf-secos."



To set up standalone configuration synchronization:

1. Configure the conf-prim device for the group:

```
config system ha
    set password **********
```

```
    set hbdev ha1 50 ha2 100
    set priority 255
    set override enable
    set standalone-config-sync enable
end
```

2. Configure the conf-prim device as needed to be functional.
3. Configure the other group members as conf-secos:

```
config system ha
    set password **********
    set standalone-config-sync enable
end
```

4. Wait 10–15 minutes for the configurations to sync over.
5. Verify the synchronization status:

```
# get system ha status
path=system, objname=ha, tablename=(null), size=5912
HA Health Status:
  WARNING: FG201E4Q17900771 has hbdev down;
  WARNING: FG201ETK19900991 has hbdev down;
Model: FortiGate-201E
Mode: ConfigSync
Group Name:
Group ID: 0
Debug: 0
Cluster Uptime: 0 days 0:0:51
Cluster state change time: 2019-09-03 17:46:07
Primary selected using:
  <2019/09/03 17:46:07> FG201ETK19900991 is selected as the primary because it has the largest
value of override priority.
ses_pickup: disable
override: disable
Configuration Status:
  FG201E4Q17900771(updated 3 seconds ago): out-of-sync
  FG201ETK19900991(updated 1 seconds ago): in-sync
System Usage stats:
  FG201E4Q17900771(updated 3 seconds ago):
    sessions=1, average-cpu-user/nice/system/idle=0%/0%/0%/100%, memory=16%
  FG201ETK19900991(updated 1 seconds ago):
    sessions=1, average-cpu-user/nice/system/idle=0%/0%/0%/100%, memory=16%
HBDEV stats:
  FG201E4Q17900771(updated 3 seconds ago):
    wan2: physical/1000auto, up, rx-bytes/packets/dropped/errors=114918/266/0/0,
tx=76752/178/0/0
    ha: physical/00, down, rx-bytes/packets/dropped/errors=0/0/0/0, tx=0/0/0/0
  FG201ETK19900991(updated 1 seconds ago):
    wan2: physical/1000auto, up, rx-bytes/packets/dropped/errors=83024/192/0/0,
tx=120216/278/0/0
    ha: physical/00, down, rx-bytes/packets/dropped/errors=0/0/0/0, tx=0/0/0/0
```

```
Secondary: FortiGate-201E, FG201E4Q17900771, HA cluster index = 1
Primary: FortiGate-201E, FG201ETK19900991, HA cluster index = 0
number of vcluster: 1
vcluster 1: work 169.254.0.1
Secondary: FG201E4Q17900771, HA operating index = 1
Primary: FG201ETK19900991, HA operating index = 0
```

If all members are `in-sync`, this means all members share the same configurations, except those that should not be synced. If any members are `out-of-sync`, this means the member failed to sync with the primary device.

Debugging is similar when a cluster is out of sync.

The following topic provides more information about standalone configuration synchronization:

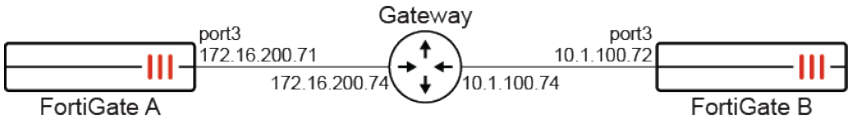
- Layer 3 unicast standalone configuration synchronization on page 4278

Layer 3 unicast standalone configuration synchronization

Unicast standalone configuration synchronization is supported on layer 3, allowing peers to be synchronized in cloud environments that do not support layer 2 networking. Configuring a unicast gateway allows peers to be in different subnets.

Example

In this example, two FortiGates in different subnets are connected through a unicast gateway. Both cluster members use the same port for the heartbeat interface.



To configure unicast synchronization between peers:

1. Configure FortiGate A:

```
config system ha
    set group-name "testcs"
    set password **********
    set hbdev "port3" 50
    set standalone-config-sync enable
    set unicast-status enable
    config unicast-peers
        edit 1
            set peer-ip 10.1.100.72
        next
    end
```

```
    set override enable
    set priority 200
    set unicast-gateway 172.16.200.74
end
```

2. Configure FortiGate B:

```
config system ha
    set group-name "testcs"
    set password **********
    set hbdev "port3" 50
    set standalone-config-sync enable
    set unicast-status enable
    config unicast-peers
        edit 1
            set peer-ip 172.16.200.71
        next
    end
    set override enable
    set priority 100
    set unicast-gateway 10.1.100.74
end
```

3. Check the HA status on FortiGate A:

```
# get system ha status
HA Health Status: OK
Model: FortiGate-VM64
Mode: ConfigSync
Group Name: testcs
Group ID: 0
Debug: 0
Cluster Uptime: 2 days 3:40:25
Cluster state change time: 2021-03-08 12:00:38
Primary selected using:
    <2021/03/08 12:00:38> FGVMSLTM00000001 is selected as the primary because its override
priority is larger than peer member FGVMSLTM00000002.
    <2021/03/06 11:50:35> FGVMSLTM00000001 is selected as the primary because it's the only
member in the cluster.
ses_pickup: disable
override: enable
Configuration Status:
    FGVMSLTM21000151(updated 5 seconds ago): in-sync
    FGVMSLTM21000152(updated 5 seconds ago): in-sync
System Usage stats:
    FGVMSLTM21000151(updated 5 seconds ago):
        sessions=7, average-cpu-user/nice/system/idle=0%/0%/0%/100%, memory=24%
    FGVMSLTM21000152(updated 5 seconds ago):
        sessions=5, average-cpu-user/nice/system/idle=0%/0%/0%/100%, memory=23%
HBDEV stats:
    FGVMSLTM21000151(updated 5 seconds ago):
        port3: physical/1000auto, up, rx-bytes/packets/dropped/errors=466060007/1049137/0/0,
```

```
tx=429538329/953028/0/0
    FGVMSLTM21000152(updated 5 seconds ago):
        port3: physical/1000auto, up, rx-bytes/packets/dropped/errors=48805199/85441/0/0,
tx=33470286/81425/0/0
Primary     : FGT-71          , FGVMSLTM00000001, HA cluster index = 1
Secondary   : FGT-72          , FGVMSLTM00000002, HA cluster index = 0
number of vcluster: 1
vcluster 1: work 0.0.0.0
Primary: FGVMSLTM00000001, HA operating index = 0
Secondary: FGVMSLTM00000002, HA operating index = 1
```

4. Check the HA checksums on FortiGate A:

```
# diagnose sys ha checksum cluster

================== FGVMSLTM00000001 ==================

is_manage_primary()=1, is_root_primary()=1
debugzone
global: 4f 2c a2 04 07 57 46 c4 47 28 ca d2 5a c5 98 ee
root: 16 af 5d a4 ac cf a5 4b b7 22 93 ce f9 02 68 bc
all: 6e 28 7f 8a 74 f7 37 43 8f 32 73 68 1e d6 ca cd

checksum
global: 4f 2c a2 04 07 57 46 c4 47 28 ca d2 5a c5 98 ee
root: 16 af 5d a4 ac cf a5 4b b7 22 93 ce f9 02 68 bc
all: 6e 28 7f 8a 74 f7 37 43 8f 32 73 68 1e d6 ca cd

================== FGVMSLTM00000002 ==================

is_manage_primary()=0, is_root_primary()=1
debugzone
global: 4f 2c a2 04 07 57 46 c4 47 28 ca d2 5a c5 98 ee
root: 16 af 5d a4 ac cf a5 4b b7 22 93 ce f9 02 68 bc
all: 6e 28 7f 8a 74 f7 37 43 8f 32 73 68 1e d6 ca cd

checksum
global: 4f 2c a2 04 07 57 46 c4 47 28 ca d2 5a c5 98 ee
root: 16 af 5d a4 ac cf a5 4b b7 22 93 ce f9 02 68 bc
all: 6e 28 7f 8a 74 f7 37 43 8f 32 73 68 1e d6 ca cd
```

5. Verify that configuration changes on the primary FortiGate are synchronized to the secondary FortiGate:
   a. Adjust the administrator timeout value on FortiGate A:

```
config system global
    set admintimeout 100
end
```

   b. Check the debug messages on FortiGate B:

```
# diagnose debug cli 7
Debug messages will be on for 30 minutes.
```

```
# diagnose debug enable

create pid=15639, clictxno=0, last=1615246288
0: conf sys global
0: set admintimeout 100
0: end
```

VRRP

A Virtual Router Redundancy Protocol (VRRP) configuration can be used as a high availability solution to ensure that a network maintains connectivity with the internet (or with other networks) even if the default router for the network fails. If a router or a FortiGate fails, all traffic to this device transparently fails over to another router or FortiGate that takes over the role of the failed device. If the failed device is restored, it will take over processing the network traffic.

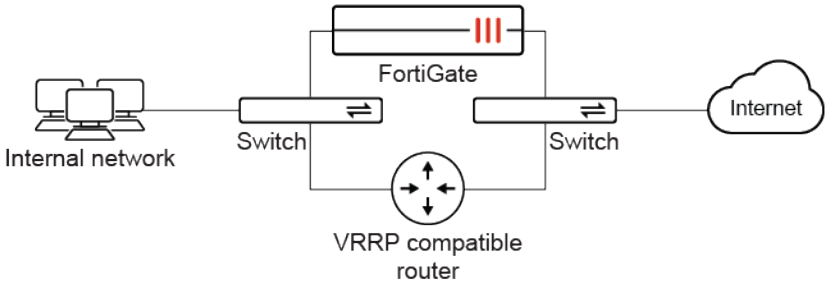
FortiOS supports VRRP versions 2 and 3. VRRP domains can be created, which can include multiple FortiGates and other VRRP-compatible routers. Different FortiGate models can be added to the same VRRP domain.

FortiOS supports IPv4 and IPv6 VRRP, so IPv4 and IPv6 VRRP virtual routers can be added to the same interface. FortiGates can quickly and easily integrate into a network that has already deployed VRRP.

Basic VRRP configuration

The most common VRRP application is to provide redundant default routers between an internal network and the internet. The default routers can be FortiGates or any routers that support VRRP.

Two or more FortiGate interfaces or routers must be configured with the same virtual router ID and IP address so they can automatically join the same VRRP domain. Priorities must be assigned to each FortiGate interface or router in the VRRP domain. All of the routers in the VRRP domain should have different priorities. One FortiGate interface or router must have the highest priority to become the primary router. The other FortiGates or routers in the domain are assigned lower priorities and become backups. If the primary router fails, VRRP automatically fails over to the router in the domain with the next highest priority.



To configure VRRP:

1. Add a virtual VRRP router to the internal interface of each FortiGate and/or router. This adds the FortiGates and routers to the same VRRP domain.
2. Set the VRRP IP address of the domain to the internal network default gateway IP address.
3. Set the priorities.

See Adding IPv4 and IPv6 virtual routers to an interface on page 4282 Single-domain VRRP example on page 4290, and Multi-domain VRRP example on page 4291 for configuration examples.

During normal operations, all traffic from the internal network to the internet passes through the primary VRRP router. The primary router also sends VRRP advertisement messages to the backup routers. A backup router will not attempt to become a primary router while receiving these messages. If the primary router fails, the backup router with the highest priority becomes the new primary router after a short delay. All packets sent to the default route are now sent to the new primary router. If the new primary router is a FortiGate, the network continues to benefit from FortiOS security features. If the new primary router is just a router, traffic continues to flow, but FortiOS security features are unavailable until the FortiGate is back online.

If the backup router is a FortiGate, during a VRRP failover as the FortiGate begins operating as the new primary router, it will not have session information for all of the failed over in-progress sessions. So, it would normally not be able to forward in-progress session traffic.

Adding IPv4 and IPv6 virtual routers to an interface

This topic describes to how to add IPv4 and IPv6 virtual routers to an interface. VRRP can only be configured on physical or VLAN interfaces. VRRP cannot be configured on hardware switch interfaces where multiple physical interfaces are combined into a hardware switch interface.

The hello timer for the VRRP dictates the rate at which VRRP advertisements are sent. With this enhanced control, you can ensure quick failover and high availability where necessary.

In VRRF v3, when the `adv-interval` is set, the interval will be rounded up to the closest 10ms (or centiseconds). VRRP v3 advertisements will be sent out at that interval and advertised in the advertisement interval (Adver Int) field of the VRRF packet as the rounded up interval.

In VRRF v2, when the `adv-interval` is set, the interval will be rounded up to the closet 10ms (or centiseconds). VRRP v2 advertisements will be sent out at that interval. However, when it advertises, the advertisement interval (Adver Int) field of the VRRF packet is rounded up to the nearest second. Therefore, if the adv-interval is set to 255ms, then the VRRP v2 advertisements will be sent out at an interval of 260ms and advertised within the VRRP v2 packet as 1sec.

IPv4 virtual router

In this example, an IPv4 VRRP router is added to port10 on the FortiGate. The VRRP virtual router has a virtual router ID of 200, uses IP address 10.31.101.200, and has a priority of 255. Since this is the highest priority in the configuration, this interface is configured to be the primary router of the VRRP domain.

To configure the interface settings:

```
config system interface
    edit port10
        config vrrp
            edit 200
                set vrip 10.31.101.200
                set priority 255
                set adv-interval 250
            next
        end
    next
end
```

IPv6 virtual router

In this example, an IPv6 VRRP router is added to port20 on the FortiGate. The VRRP virtual router has a virtual router ID of 220, uses IP address 2001:db8:1::12, and has a priority of 255. Since this is the highest priority in the configuration, this interface is configured to be the primary router of the VRRP domain.

To configure the interface settings:

```
config system interface
    edit port20
        config ipv6
            set vrip6_link_local <IPv6_address>
            config vrrp6
                edit 220
                    set vrip 2001:db8:1::12
                    set priority 255
                    set adv-interval 250
                next
            end
        end
    next
end
```

VRRP failover

VRRP routers in a VRRP domain periodically send VRRP advertisement messages to all routers in the domain to maintain one router as the primary router and the others as backup routers. The primary router has the highest priority. If the backup routers stop receiving these packets from the primary router, the backup router with the highest priority becomes the new primary router.

The primary router stops sending VRRP advertisement messages if it fails or becomes disconnected. Up to two VRRP destination addresses can be configured to be monitored by the primary router. As a best practice, the destination addresses should be remote addresses. If the primary router is unable to connect to these

destination addresses, it stops sending VRRP advertisement messages, and the backup router with the highest priority becomes the primary router.

To configure IPv4 VRRP with two destination addresses for monitoring:

```
config system interface
    edit port14
        config vrrp
            edit 12
                set vrdst 10.10.10.20 10.20.20.10
            next
        end
    next
end
```

To configure IPv6 VRRP with one destination address for monitoring:

```
config system interface
    edit port23
        config ipv6
            config vrrp6
                edit 223
                    set vrdst 2001:db8:1::12
                next
            end
        end
    next
end
```

IPv4 VRRP active failover

The `vrdst-priority` option can be used to reduce IPv4 VRRP failover times. This option causes the primary router to actively signal to the backup routers when the primary router cannot reach its configured destination addresses. The primary router sends a lower priority for itself in the VRRP advertisement messages. The backup router with the highest priority becomes the new primary router and takes over traffic processing.

In this example, the primary router is configured to have a priority of 255, so it should always become the primary router. The `vrdst-priority` is set to 10. If the primary router cannot connect to the 10.10.10.1 destination address, then the primary router informs the VRRP group that its priority is now 10.

To set the priority of the virtual router when the destination address is unreachable:

```
config system interface
    edit port10
        config vrrp
            edit 12
                set vrip 10.31.101.200
                set priority 255
                set vrdst 10.10.10.1
```

```
                set vrdst-priority 10
            next
        end
    next
end
```

IPv4 VIP and IP pool failover

The `proxy-arp` option can be used to map VIPs and IP pool address ranges to each router's VMAC (virtual MAC). After failover, the IP or ranges configured in the VRRP settings are routed to the new primary router's VMAC. In this example, a single IP and an address range are added for proxy ARP.

To configure the IP addresses for proxy ARP:

```
config system interface
    edit port5
        set vrrp-virtual-mac enable
        config vrrp
            edit 1
                config proxy-arp
                    edit 1
                        set ip 192.168.62.100-192.168.62.200
                    next
                    edit 2
                        set ip 192.168.62.225
                    next
                end
            next
        end
    next
end
```

Changing the advertisement message interval

By default, VRRP advertisement messages are sent once every second. The frequency can be changed with the `adv-interval` option to change the frequency of sending these messages (1 - 255 seconds).

The `adv-interval` also affects the period of time that a backup VRRP router waits before assuming the primary router has failed. The waiting period is three times the `adv-interval`. For example, if the `adv-interval` is set to 5, then the backup router waits for up to 15 seconds to receive a VRRP advertisement from the current primary router before taking over the role as the primary router.

To configure IPv4 VRRP to send advertisement messages every 10 seconds:

```
config system interface
    edit port14
        config vrrp
            edit 12
```

```
                set adv-interval 10
            next
        end
    next
end
```

To configure IPv6 VRRP to send advertisement messages every 20 seconds:

```
config system interface
    edit port23
        config ipv6
            config vrrp6
                edit 223
                set adv-interval 20
            next
        end
    next
end
```

Changing the VRRP startup time

The VRRP startup time is the time a backup or primary VRRP router waits before sending or receiving VRRP advertisements before potentially changing state (`start-time` in seconds, 1 - 255, default = 3). This timer is mainly visible when VRRP-monitored interfaces become up after previously been down. When this occurs, the device will wait for the time period before considering, and potentially changing its status.

There are some instances when the advertisement messages might be delayed. For example, some switches with spanning tree enabled may delay some of the advertisement message packets. If backup routers are attempting to become primary routers even though the primary router has not failed, extend the start time to ensure that the backup routers wait long enough for the advertisement messages.

To configure the IPv4 VRRP startup time to 10 seconds:

```
config system interface
    edit port14
        config vrrp
            edit 12
                set start-time 10
            next
        end
    next
end
```

To configure the IPv6 VRRP startup time to 15 seconds:

```
config system interface
    edit port23
        config ipv6
            config vrrp6
```

```
                edit 223
                set start-time 15
            next
        end
    next
end
```

VRRP groups

If VRRP routers are added to multiple interfaces of the same FortiGate, each router will be in a different VRRP domain. If one of the VRRP routers fails, it is useful if all of the VRRP routers added to the FortiGate also fail.

VRRP can only check the routers' status in a single VRRP domain and cannot track the status of routers in other domains. For multiple VRRP domains on a single FortiGate, only one can switch to being a backup, and the others remain operating normally. Using VRRP groups resolves this issue.

All the VRRP virtual routers on the FortiGate can be added to a VRRP group. If one of the virtual routers in a VRRP group switches to the backup, the VRRP group forces all members to switch to backups. All VRRP traffic being processed by the FortiGate fails over to other devices in the network.

The status of the virtual routers in a VRRP group only changes when one or more of the virtual routers in the group changes status. A VRRP group should not be used to manually change the status of the virtual routers in the group.

To configure two IPv4 VRRP routers in a VRRP group:

```
config system interface
    edit port10
        config vrrp
            edit 200
                set vrip 10.31.101.200
                set priority 255
                set vrgrp 10
            next
        end
    next
    edit port20
        config vrrp
            edit 100
                set vrip 10.23.1.223
                set priority 20
                set vrgrp 10
            next
        end
    next
end
```

To configure two IPv6 VRRP routers in a VRRP group:

```
config system interface
    edit port11
        config ipv6
            set vrip6_link_local <IPv6_address>
            config vrrp6
                edit 220
                    set vrip 2001:db8:1::12
                    set priority 255
                    set vrgrp 90
                next
            end
        end
    next
    edit port12
        config ipv6
            set vrip6_link_local <IPv6_address>
            config vrrp6
                edit 220
                    set vrip 2001:db8:1::14
                    set priority 100
                    set vrgrp 90
                next
            end
        end
    next
end
```

VRRP virtual MACs

The VRRP virtual MAC address (or virtual router MAC address) is a shared MAC address adopted by the primary router. If the primary router fails, the same virtual MAC address is picked up by the new primary router, allowing all devices on the network to transparently connect to the default route using the same virtual MAC address. This feature must be enabled on all members in a VRRP domain.

Each VRRP router has its own virtual MAC address. The last part octet is based on the VRRP router ID using the following format:

```
00-00-5E-00-01-<VRID_hex>
```

Where `<VRID_hex>` is the VRRP router ID in hexadecimal format in internet standard bit-order. For more information about virtual MAC formatting, see RFC 3768.

For example:

- If the VRRP router ID is 10, then the virtual MAC is `00-00-5E-00-01-0a`.
- If the VRRP router ID is 200, then the virtual MAC is `00-00-5E-00-01-c8`.

If the VRRP virtual MAC address feature is disabled (the default setting), the VRRP domain uses the MAC address of the primary router. On a FortiGate VRRP virtual router, this is the MAC address of the FortiGate interface that the VRRP router is added to. If the primary fails, when the new primary takes over, it sends

gratuitous ARPs to associate the VRRP router IP address with the MAC address of the new primary (or the FortiGate interface that became the new primary).

When a VRRP virtual MAC address is enabled, the new primary uses the same MAC address as the old primary.

Since devices on the LAN do not have to learn a new MAC address for a new VRRP router in the event of a failover, this feature can improve network efficiency, especially in large and complex networks.

To enable virtual MAC addresses in IPv4 VRRP:

```
config system interface
    edit <name>
        set vrrp-virtual-mac enable
    next
end
```

To enable virtual MAC addresses in IPv6 VRRP:

```
config system interface
    edit <name>
        config ipv6
            set vrrp-virtual-mac6 enable
        end
    next
end
```

Preempt mode

When preempt mode is enabled (the default setting), a higher priority backup router can preempt a lower priority primary router. This can happen if the primary router fails, the backup router becomes the primary router, and the failed primary router restarts. Since the restarted router has a higher priority, if preempt mode is enabled, the restarted router replaces the current primary router becoming the new primary router. If preempt mode is disabled, a restarted router that has a higher priority would not take over as the primary router.

Based on RFC 3768 Section 5.3.4, "The priority value for the VRRP router that owns the IP address (es) associated with the virtual router MUST be 255 (decimal). VRRP routers backing up a virtual router MUST use priority values between 1-254 (decimal)."

Therefore, in cases where preempt mode is disabled, but the priority is set to 255, the restarted unit will take over as the primary router.

To configure preempt mode in IPv4 VRRP:

```
config system interface
    edit <name>
        config vrrp
            edit <vrid>
                set preempt {enable | disable}
```

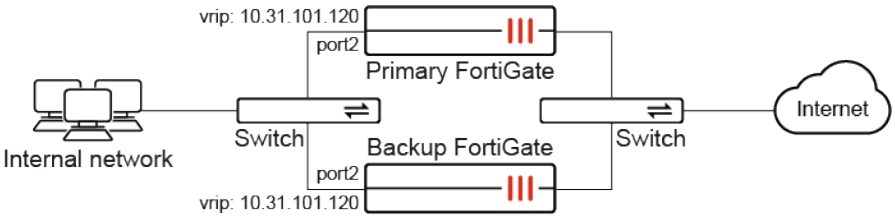
```
            next
        end
    next
end
```

To configure preempt mode in IPv6 VRRP:

```
config system interface
    edit <name>
        config ipv6
            config vrrp6
                edit <vrid>
                    set preempt {enable | disable}
                next
            end
        end
    next
end
```

Single-domain VRRP example

This example consists of a VRRP domain with two FortiGates that connect an internal network to the internet. The FortiGate port2 interfaces connect to the internal network, and a VRRP virtual router is added to each port2 interface with VRRP virtual MAC addresses enabled. The internal network default route is 10.31.101.120. Each FortiGate port2 interface has an IP address that is different from the virtual router IP address. Since `vrrp-virtual-mac` is enabled, upon failover, the new primary VRRP router will use the same VMAC as the previous router.



To configure the primary FortiGate:

```
config system interface
    edit port2
        set vrrp-virtual-mac enable
        config vrrp
            edit 5
                set vrip 10.31.101.120
                set priority 255
            next
        end
    next
end
```

To configure the backup FortiGate:

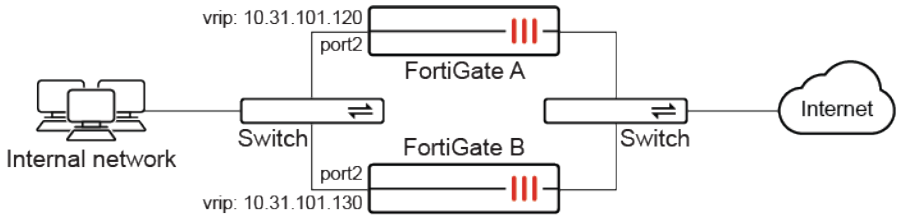
```
config system interface
    edit port2
        set vrrp-virtual-mac enable
        config vrrp
            edit 5
                set vrip 10.31.101.120
                set priority 50
            next
        end
    next
end
```

Multi-domain VRRP example

This example consists of two VRRP domains, and both FortiGates participate in the domains that connect an internal network to the internet. One FortiGate is the primary router of one domain and the other FortiGate is the primary router of the other domain. The network distributes traffic between two different default routes (10.31.101.120 and 10.31.101.130). One VRRP domain is configured with one of the default route IP addresses and the other VRRP domain gets the other default route IP address. During normal operation, both FortiGates process traffic, and the VRRP domains are used to load balance the traffic between the two FortiGates.

If one of the FortiGates fails, the remaining FortiGate becomes the primary router of both VRRP domains. The network sends all traffic for both default routes to this FortiGate. The result is a configuration that (under normal operational load) balances traffic between two FortiGates, but if one of the FortiGates fails, all traffic fails over to the FortiGate that is still operating.

VRRP virtual MAC address are enabled on both FortiGates' port2 interfaces so that the VRRP domains use their VRRP virtual MAC addresses.



Device	VRRP primary			VRRP backup		
	Virtual router IP	ID	Priority	Virtual router IP	ID	Priority
FortiGate A	10.31.101.120	50	255	10.31.101.130	100	50
FortiGate B	10.31.101.130	100	255	10.31.101.120	50	50

To configure FortiGate A:

```
config system interface
    edit port2
```

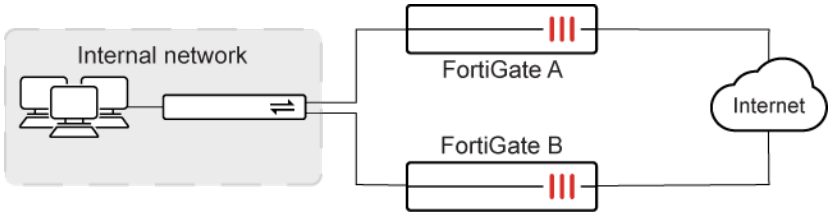
```
        set vrrp-virtual-mac enable
        config vrrp
            edit 50
                set vrip 10.31.101.120
                set priority 255
            next
            edit 100
                set vrip 10.31.101.130
                set priority 50
            next
        end
    next
end
```

To configure FortiGate B:

```
config system interface
    edit port2
        set vrrp-virtual-mac enable
        config vrrp
            edit 50
                set vrip 10.31.101.120
                set priority 50
            next
            edit 100
                set vrip 10.31.101.130
                set priority 255
            next
        end
    next
end
```

VRRP on EMAC-VLAN interfaces

Virtual Router Redundancy Protocol (VRRP) can be configured on EMAC-VLAN interfaces.



To configure the interfaces:

1. Configure FortiGate A:

```
config system interface
    edit "emac"
        set vdom "root"
        set ip 172.16.209.1 255.255.255.0
        set allowaccess ping https ssh snmp http telnet fgfm
        set type emac-vlan
        set vrrp-virtual-mac enable
        config vrrp
            edit 1
                set vrip 172.16.209.111
                set priority 200
            next
        end
        set snmp-index 61
        set interface "port1"
    next
end
```

2. Configure FortiGate B:

```
config system interface
    edit "emac"
        set vdom "root"
        set ip 172.16.209.2 255.255.255.0
        set allowaccess ping https ssh snmp http telnet fgfm
        set type emac-vlan
        set vrrp-virtual-mac enable
        config vrrp
            edit 1
                set vrip 172.16.209.111
                set priority 222
            next
        end
        set snmp-index 32
        set interface "port1"
    next
end
```

Check the VRRP information on the FortiGates:

Because FortiGate B has a higher priority, it is the primary device and FortiGate A is the backup.

1. FortiGate A:

```
# get router info vrrp
Interface: emac, primary IP address: 172.16.209.1
  UseVMAC: 1, SoftSW: 0, EmacVlan: 1 BrPortIdx: 0, PromiscCount: 0
  HA mode: primary (0:0:1) VRRP master number: 0
  VRID: 1 verion: 2
```

```
    vrip: 172.16.209.111, priority: 200 (200,0), state: BACKUP
    adv_interval: 1, preempt: 1, ignore_dft: 0 start_time: 3
    master_adv_interval: 100, accept: 1
    vrmac: 00:00:5e:00:01:01
    vrdst:
    vrgrp: 0
```

2. FortiGate B:

```
# get router info vrrp
Interface: emac, primary IP address: 172.16.209.2
  UseVMAC: 1, SoftSW: 0, EmacVlan: 1 BrPortIdx: 0, PromiscCount: 1
  HA mode: primary (0:0:1) VRRP master number: 1
  VRID: 1 verion: 2
    vrip: 172.16.209.111, priority: 222 (222,0), state: PRIMARY
    adv_interval: 1, preempt: 1, ignore_dft: 0 start_time: 3
    master_adv_interval: 100, accept: 1
    vrmac: 00:00:5e:00:01:01
    vrdst:
    vrgrp: 0
```

Ignore VRRP default route

Administrators can choose to exclude the default route from the calculation of available routes to the IPv6 VRRP destination, to better manage and control the VRRP states. Previously, the VRRP state would be `Primary` as long as any route, including the default route, could reach the IPv6 VRRP destination.

```
config system interface
    edit <name>
        config ipv6
            config vrrp6
                edit <id>
                    set ignore-default-route {enable | disable}
                next
            end
        end
    next
end
```

Command	Description
`ignore-default-route {enable | disable}`	Set the default route to be ignored: • `enable`: Ignore the default route when checking the VRRP destination. • `disable`: Include the default route when checking the VRRP destination (default).

Example

In this example, the IPv6 VRRP destination (`vrdst6`) is set with an IPv6 address of `2000:172:22:20::22`, and `ignore-default-route` is enabled for the destination. As long as non-default routes exist to the VRRP destination, the VRRP state is `Primary`. When only the default route to the VRRP destination exists, the VRRP state changes to `Backup`.

To ignore the default route when checking the IPv6 VRRP destination:

1. Enable the default route to be ignored for IPv6 VRRP.
   The IPv6 VRRP destination (`vrdst6`) is set with an IPv6 address of `2000:172:22:20::22`, and `ignore-default-route` is enabled for the destination.

```
config system interface
    edit "port2"
        config ipv6
            set vrrp-virtual-mac6 enable
            set vrip6_link_local fe80::926c:acff:2222:2222
            config vrrp6
                edit 100
                    set vrgrp 100
                    set vrip6 2000:10:1:100::222
                    set priority 200
                    set vrdst6 2000:172:22:20::22
                    set ignore-default-route enable
                next
            end
        end
    next
end
```

2. Check the route for IPv6 VRRP destination.
   The routing table shows an active route through `port1` to the IPv6 VRRP destination of `2000:172:22:20::22`. The active route is not a default route.

```
# get router info6 routing-table 2000:172:22:20::22
Routing entry for 2000:172:22:20::/80
  Known via "static", distance 10, metric 0
  Last update 00:00:15 ago
    via 2000:172:16:200::55, port1
```

3. Check VRRP group information for IPv6.
   The VRRP state is `Primary` because non-default routes to the IPv6 VRRP destination exist as shown in the previous step.

```
# get router info6 vrrp
Interface: port2, primary IPv6 address: 2000:10:1:100::1
link-local IPv6 address: fe80::96f3:92ff:fe15:1ecd
Virtual link-local IPv6 address: fe80::926c:acff:2222:2222
  UseVMAC: 1, SoftSW: 0, EmacVlan: 0 BrPortIdx: 0, PromiscCount: 1
  HA mode: primary (0:0:1)
```

```
  VRT primary count: 1
  VRID: 100 version: 3
    vrip: 2000:10:1:100::222, priority: 200, state: PRIMARY
    adv_interval: 1, preempt: 1, ignore_dft: 0, start_time: 3
    primary_adv_interval: 100, accept: 1
    vrmac: 00:00:5e:00:02:64
    vrdst: 2000:172:22:20::22
    vrgrp: 100
```

4. Delete the non-default routes to the IPv6 VRRP destination (`vrdst6`), and check the routes again.
   The routing table shows only the default route (`::/0`) is available to the IPv6 VRRP destination of `2000:172:22:20::22`.

```
# get router info6 routing-table 2000:172:22:20::22
Routing entry for ::/0
  Known via "static", distance 10, metric 0, best
  Last update 02:02:09 ago
  * via 2000:172:16:200::254, port1
```

5. Check VRRP group information for IPv6.
   The VRRP state is `Backup` because only the default route is available to the IPv6 VRRP destination as shown in the previous step.

```
#get router info6 vrrp
Interface: port2, primary IPv6 address: 2000:10:1:100::1
link-local IPv6 address: fe80::96f3:92ff:fe15:1ecd
Virtual link-local IPv6 address: fe80::926c:acff:2222:2222
  UseVMAC: 1, SoftSW: 0, EmacVlan: 0 BrPortIdx: 0, PromiscCount: 0
  HA mode: primary (0:0:1)
  VRT primary count: 0
  VRID: 100 version: 3
    vrip: 2000:10:1:100::222, priority: 0, state: BACKUP
    adv_interval: 1, preempt: 1, ignore_dft: 1, start_time: 3 but
    primary_adv_interval: 100, accept: 1
    vrmac: 00:00:5e:00:02:64
    vrdst: 2000:172:22:20::22
    vrgrp: 100
```

Fortinet Security Fabric

The Fortinet Security Fabric provides an intelligent architecture that interconnects discrete security solutions into an integrated whole to detect, monitor, block, and remediate attacks across the entire attack surface. It delivers broad protection and visibility into every network segment and device, be they hardware, virtual, or cloud based.

- The physical topology view shows all connected devices, including access layer devices. The logical topology view shows information about the interfaces that each device is connected to.
- Security rating checks analyze the Security Fabric deployment to identify potential vulnerabilities and highlight best practices to improve the network configuration, deploy new hardware and software, and increase visibility and control of the network.
- Fabric connectors provide integration with multiple SDN, cloud, and partner technology platforms to automate the process of managing dynamic security updates without manual intervention.
- Automation pairs an event trigger with one or more actions to monitor the network and take the designated actions automatically when the Security Fabric detects a threat.

As part of improvements to reducing memory usage, FortiGate models with 2 GB RAM can authorize up to five devices when serving as a Fabric root.

The affected models are the FortiGate 40F, 50G, and 60F series devices and their variants.

A maximum of 35 downstream FortiGates is recommended.

Components

As part of improvements to reducing memory usage, FortiGate models with 2 GB RAM can authorize up to five devices when serving as a Fabric root.

The affected models are the FortiGate 40F, 50G, and 60F series devices and their variants.

The Fortinet Security Fabric consists of different components that work together to secure you network.

The following devices are required to create a Security Fabric:

Device	Description
FortiGate	FortiGates are the core of the Security Fabric and can have one of the following roles: • **Root**: the root FortiGate is the main component in the Security Fabric. It is typically located on the edge of the network and connects the internal devices and networks to the internet through your ISP. From the root FortiGate, you can see information about the entire Security Fabric on the *Physical* and *Logical Topology*

Device	Description
	pages in the GUI. • **Downstream**: after a root FortiGate is installed, all other FortiGate devices in the Security Fabric act as Internal Segmentation Firewalls (ISFWs), located at strategic points in your internal network, rather than on the network edge. This allows extra security measures to be taken around key network components, such as servers that contain valuable intellectual property. ISFW FortiGates create network visibility by sending traffic and information about the devices that are connected to them to the root FortiGate. A maximum of 35 downstream FortiGates is recommended. See Configuring the root FortiGate and downstream FortiGates on page 4302 for more information about adding FortiGate devices in the Security Fabric. FortiGate documentation: https://docs.fortinet.com/product/fortigate
FortiAnalyzer*	FortiAnalyzer gives you increased visibility into your network, centralized monitoring, and awareness of threats, events, and network activity by collecting and correlating logs from all Security Fabric devices. This gives you a deeper and more comprehensive view across the entire Security Fabric. See Configuring FortiAnalyzer on page 4311 for more information about adding FortiAnalyzer devices in the Security Fabric. FortiAnalyzer documentation: https://docs.fortinet.com/product/fortianalyzer
Cloud Logging*	There are two options for cloud logging: FortiAnalyzer Cloud and FortiGate Cloud. Either can be used to enable the Security Fabric root device; however, if using FortiGate Cloud, all downstream devices must belong to the same FortiCloud account. See Configuring cloud logging on page 4314 for more information about configuring a Security Fabric with FortiGate Cloud. FortiGate Cloud documentation: https://docs.fortinet.com/product/fortigate-cloud

* FortiAnalyzer or Cloud Logging is a required component for the Security Fabric. Either FortiAnalyzer, FortiAnalyzer Cloud, or FortiGate Cloud can be used to meet this requirement.

The following devices are recommended:

Device	Description
FortiClient	FortiClient adds endpoint control to devices that are located in the Security Fabric, allowing only traffic from compliant devices to flow through the FortiGate. FortiClient compliance profiles are applied by the first FortiGate that a device's traffic flows through. Device registration and on-net status information for a device that is running FortiClient appears only on the FortiGate that applies the FortiClient profile to that device. FortiClient documentation: https://docs.fortinet.com/product/forticlient
FortiClient EMS	FortiClient EMS is used in the Security Fabric to provide visibility across your network, securely share information, and assign security profiles to endpoints. See Configuring FortiClient EMS on page 4322 for more information about adding FortiClient EMS devices in the Security Fabric. FortiClient EMS documentation: https://docs.fortinet.com/product/forticlient

Device	Description
FortiAP	Add FortiAP devices to extend the Security Fabric to your wireless devices. Devices connected to a FortiAP appear in the Physical and Logical Topology pages in the Security Fabric menu. See Configuring LAN edge devices on page 4347 for more information about adding FortiAP devices in the Security Fabric. FortiAP documentation: https://docs.fortinet.com/product/fortiap
FortiSwitch	A FortiSwitch can be added to the Security Fabric when it is managed by a FortiGate that is in the Security Fabric with the FortiLink protocol, and connected to an interface with *Security Fabric Connection* enabled. FortiSwitch ports to become logical extensions of the FortiGate. Devices connected to the FortiSwitch appear in the Physical and Logical Topology pages in the Security Fabric menu, and security features, such as FortiClient compliance profiles, are applied to them. See Configuring LAN edge devices on page 4347 for more information about adding FortiSwitch devices in the Security Fabric. FortiSwitch documentation: https://docs.fortinet.com/product/fortiswitch
FortiExtender	FortiExtender cellular gateways provide ultra-fast LTE and 5G wireless to connect and scale any WAN edge. See Configuring LAN edge devices on page 4347 for more information about adding FortiExtender devices in the Security Fabric. FortiExtender documentation: https://docs.fortinet.com/product/fortiextender
FortiManager	Add FortiManager to simplify the network management of devices in the Security Fabric by centralizing management access in a single device. This allows you to easily control the deployment of security policies, FortiGuard content security updates, firmware revisions, and individual configurations for devices in the Security Fabric. See Configuring central management on page 4349 for more information about adding FortiManager devices in the Security Fabric. FortiManager documentation: https://docs.fortinet.com/product/fortimanager
FortiSandbox	Add FortiSandbox to your Security Fabric to improve security with sandbox inspection. Sandbox integration allows FortiGate devices in the Security Fabric to automatically receive signature updates from FortiSandbox and add the originating URL of any malicious file to a blocked URL list. See Configuring sandboxing on page 4354 for more information about adding FortiSandbox devices in the Security Fabric. FortiSandbox documentation: https://docs.fortinet.com/product/fortisandbox
FortiADC	FortiADC devices optimize the availability, user experience, and scalability of enterprise application delivery. They enable fast, secure, and intelligent acceleration and distribution of even the most demanding enterprise applications. FortiADC documentation: https://docs.fortinet.com/product/fortiadc
FortiDDoS	FortiDDoS is a Network Behavior Anomaly (NBA) prevention system that detects and blocks attacks that intend to disrupt network service by overutilizing server resources.

Device	Description
	FortiDDoS documentation: https://docs.fortinet.com/product/fortiddos
FortiDeceptor	FortiDeceptor automatically lays out a layer of decoys and lures, which helps conceal sensitive and critical assets behind a fabricated deception surface to confuse and redirect attackers while revealing their presence on your network. See Configuring FortiDeceptor on page 4382 for more information about adding FortiDeceptor devices in the Security Fabric. FortiDeceptor documentation: https://docs.fortinet.com/product/fortideceptor
FortiMail	FortiMail antispam processing helps offload from other devices in the Security Fabric that would typically carry out this process. See Configuring FortiMail on page 4383 for more information about adding FortiMail devices in the Security Fabric. FortiMail documentation: https://docs.fortinet.com/product/fortimail
FortiMonitor	FortiMonitor is a holistic, SaaS-based digital experience and network performance monitoring solution. It facilitates deep analysis of both network health metrics and application performance to identify potential problem areas that impact user access. See Configuring FortiMonitor on page 4384for more information about adding FortiMonitor devices in the Security Fabric. FortiMonitor documentation: https://docs.fortinet.com/product/fortimonitor
FortiNAC	FortiNAC provides visibility to all administrators to see everything connected to their network, and the ability to control those devices and users, including dynamic, automated responses. See Configuring FortiNAC on page 4385 for more information about adding FortiNAC devices in the Security Fabric. FortiNAC documentation: https://docs.fortinet.com/product/fortinac
FortiNDR	FortiNDR uses artificial neural networks (ANN) that can deliver sub-second malware detection and a verdict. Add FortiNDR to your Security Fabric to automatically quarantine attacks. See Configuring FortiNDR on page 4387 for more information about adding FortiNDR devices in the Security Fabric. FortiNDR documentation: https://docs.fortinet.com/product/fortindr
FortiTester	FortiTester can be used for performance testing and validating network security infrastructure and services. It provides a comprehensive range of application test cases to evaluate equipment and right-size infrastructure. See Configuring FortiTester on page 4387 for more information about adding FortiTester devices in the Security Fabric. FortiTester documentation: https://docs.fortinet.com/product/fortitester
FortiWeb	FortiWeb defends the application attack surface from attacks that target application exploits. You can also configure FortiWeb to apply web application firewall features, virus scanning, and web filtering to HTTP traffic to help offload from other devices in the Security Fabric that would typically carry out these processes.

Device	Description
	See Configuring FortiWeb on page 4390 for more information about adding FortiWeb devices in the Security Fabric. FortiWeb documentation: https://docs.fortinet.com/product/fortiweb
FortiWLC	FortiWLC delivers seamless mobility and superior reliability with optimized client distribution and channel utilization. Both single and multi channel deployment options are supported, maximizing efficiency to make the most of available wireless spectrum. FortiWLC documentation: https://docs.fortinet.com/product/wireless-controller
FortiVoice	FortiVoice includes integrated high-definition voice, conferencing, and fax capabilities that enables organizations to communicate and collaborate easily and securely. See Configuring FortiVoice on page 4389 for more information about adding FortiVoice devices in the Security Fabric. FortiVoice documentation: https://docs.fortinet.com/product/fortivoice-enterprise
Other Fortinet products	Other Fortinet products can be added to the Security Fabric, including FortiAuthenticator, FortiToken, FortiCache, and FortiSIEM. Documentation: https://docs.fortinet.com/
Third-party products	Third-party products that belong to the Fortinet Fabric-Ready Partner Program can be added to the Security Fabric.

Security Fabric connectors

This section contains information about how to configure the following devices as part of the Fortinet Security Fabric:

- Configuring the root FortiGate and downstream FortiGates
- Configuring logging and analytics on page 4310
  - Configuring FortiAnalyzer on page 4311
  - Configuring cloud logging on page 4314
- Configuring FortiClient EMS on page 4322
- Synchronizing FortiClient ZTNA tags on page 4344
- Configuring LAN edge devices on page 4347
- Configuring central management on page 4349
- Configuring sandboxing on page 4354
- Connecting to FortiTelemetry agents on page 4364
- Connecting to FortiData on page 4369
- Configuring supported connectors on page 4373
  - Configuring FortiNDR on page 4387
  - Configuring FortiDeceptor on page 4382
  - Configuring FortiMail on page 4383
  - Configuring FortiMonitor on page 4384

  - Configuring FortiNAC on page 4385
  - Configuring FortiTester on page 4387
  - Configuring FortiVoice on page 4389
  - Configuring FortiWeb on page 4390
- Allowing FortiDLP Agent communication through the FortiGate on page 4392

System requirements

To set up the Security Fabric, the devices that you want to include must meet the Product Integration and Support requirements in the FortiOS Release Notes.

Some features of the Security Fabric are only available in certain firmware versions and models. Not all FortiGate models can run the FortiGuard Surface Attack Security Rating Service if they are the root FortiGate in a Security Fabric. For more information, see the Special Notices in the FortiOS Release Notes.

Prerequisites

- If devices are not already installed in your network, complete basic installation and configuration tasks by following the instructions in the device documentation.
- FortiGate devices must be operating in NAT mode.

Configuring the root FortiGate and downstream FortiGates

The following procedures include configuration steps for a typical Security Fabric implementation, where the edge FortiGate is the root FortiGate with other FortiGates that are downstream from the root FortiGate.

For information about the recommended number of downstream FortiGates, see the *FortiOS Best Practices*.

As part of improvements to reducing memory usage, FortiGate models with 2 GB RAM can authorize up to five devices when serving as a Fabric root.

The affected models are the FortiGate 40F, 50G, and 60F series devices and their variants.

Prerequisite

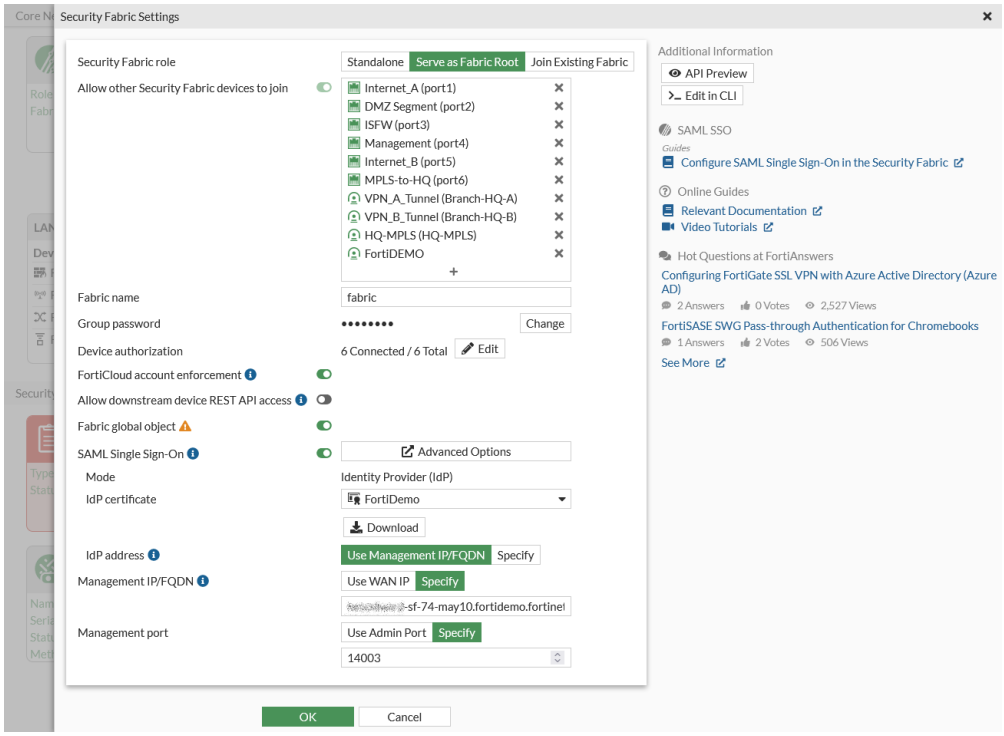
- The FortiGates must be operating in NAT mode.

Configuring the root FortiGate

The edge FortiGate is typically configured as the root FortiGate, as this allows you to view the full topology of the Security Fabric from the top down. The following steps describe how to add the FortiGate to serve as the root device.

To configure the root FortiGate:

1. On the root FortiGate, go to *Security Fabric > Fabric Connectors* and double-click the *Security Fabric Setup* card.
2. Set the *Security Fabric role* to *Serve as Fabric Root*.
3. Enter a *Fabric name*.
4. Ensure *Allow other Security Fabric devices to join* is enabled.
5. Select the interfaces that will be listening for device join requests. Enabling an interface here has the same effect as going to *Network > Interfaces*, editing an interface, and enabling *Security Fabric Connection* under *Administrative Access*.



6. Optionally, enable *Allow downstream device REST API access* to allow access to the REST API of the root FortiGate for API requests coming from downstream Security Fabric devices. This option must be enabled to use certain supported devices (such as FortiDeceptor, FortiMonitor, and FortiNAC) and the Fabric event trigger.
7. Click *OK*.

Using the root FortiGate with disk to store historic user and device information

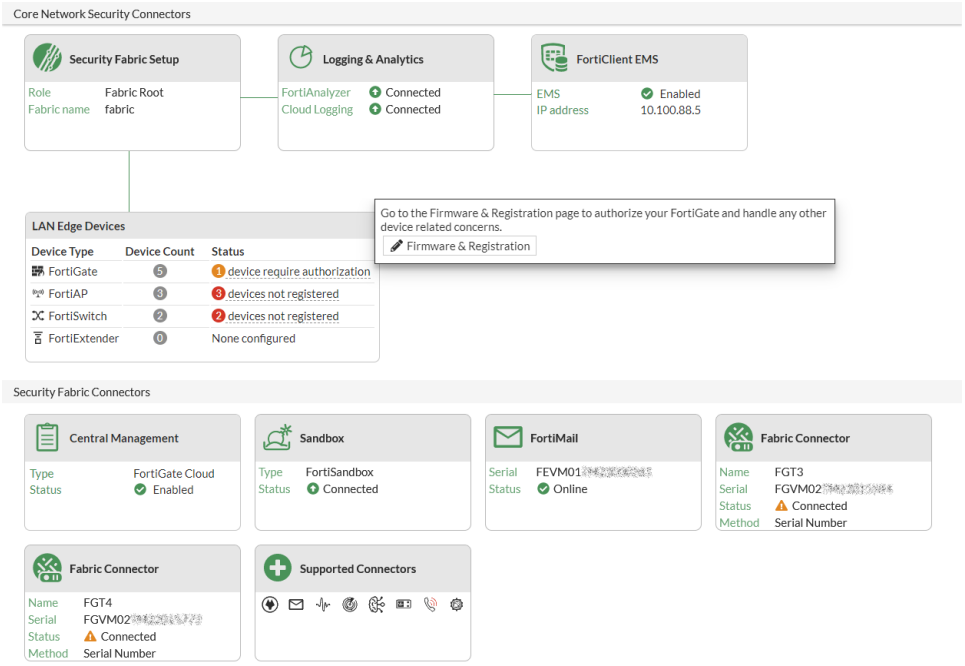
This backend implementation allows the root FortiGate in a Security Fabric to store historic user and device information in a database on its disk. This will allow administrators to visualize users and devices over a period of time.

The daemon, user_info_history, stores this data on the disk. The information source for the historical data will be the user_info daemon, which would be recorded on the disk when user_info notifies user_info_history that a user has logged out or the device is no longer connected.

Adding downstream devices

Downstream device serial numbers can be pre-authorized from the root FortiGate, or allowed to join by request. New authorization requests include the device serial number, IP address, and HA members. HA members can include up to four serial numbers and is used to ensure that, in the event of a fail over, the secondary FortiGate is still authorized. A downstream device's certificate can also be used to authorize the device by uploading the certificate to the root FortiGate.

The *LAN Edge Devices* card on the *Fabric Connectors* page displays a summary about the FortiGates, FortiAPs, FortiSwitches, and FortiExtenders in the Fabric. Information about the device type, number of devices, and number of unregistered and unauthorized devices is displayed. If there are devices that do not have a green checkmark in the *Status* column, hover over the status message to view the tooltip with required action. In this example, there is a downstream FortiGate that require authorization. The tooltip includes a link to the *System > Firmware & Registration* page to authorize the FortiGates.



The *Supported Connectors* card displays the icons of different Fortinet devices that support full Security Fabric integration. See Configuring supported connectors on page 4373 for more information about configuring supported Fabric connectors.

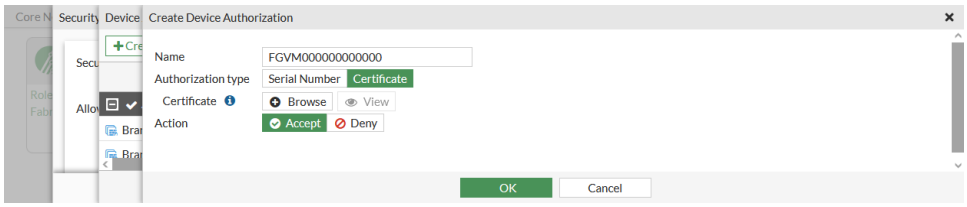
Pre-authorizing the downstream FortiGate

When a downstream Fortinet device's serial number or certificate is added to the trusted list on the root FortiGate, the device can join the Security Fabric as soon as it connects. After the new device is authorized, connected FortiAP and FortiSwitch devices are automatically included in the topology, where they can be authorized with one click.

The interface that connects to the downstream FortiGate must have *Security Fabric Connection* enabled.

To pre-authorize a FortiGate:

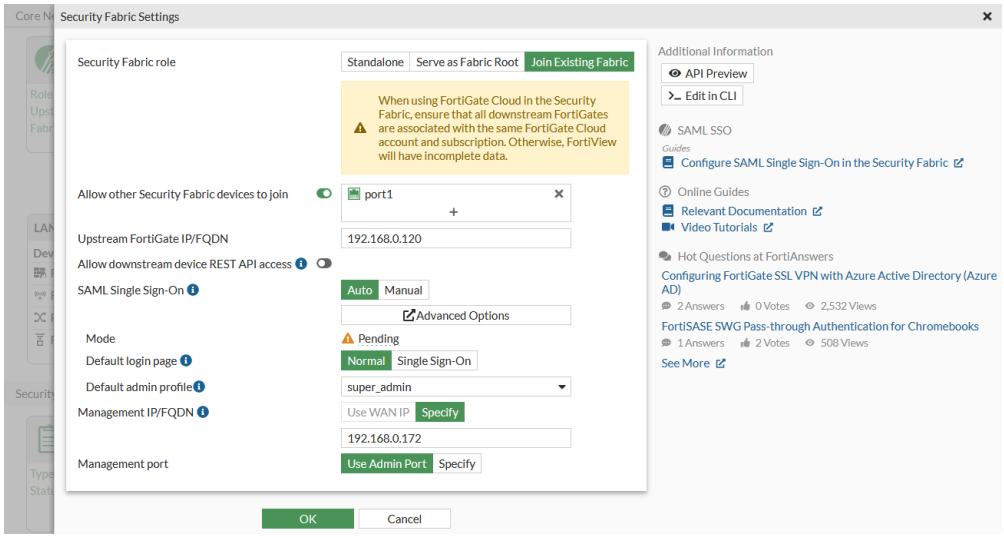
1. On the root FortiGate, go to *Security Fabric > Fabric Connectors* and double-click the *Security Fabric Setup* card.
2. In the *Device authorization* field and click *Edit*. The *Device Authorization* pane opens.
3. Click *Create New* to add a new device for pre-authorization.
4. Enter the device name in the *Name* field.
5. Select the *Authorization type*, either *Serial Number* or *Certificate*.



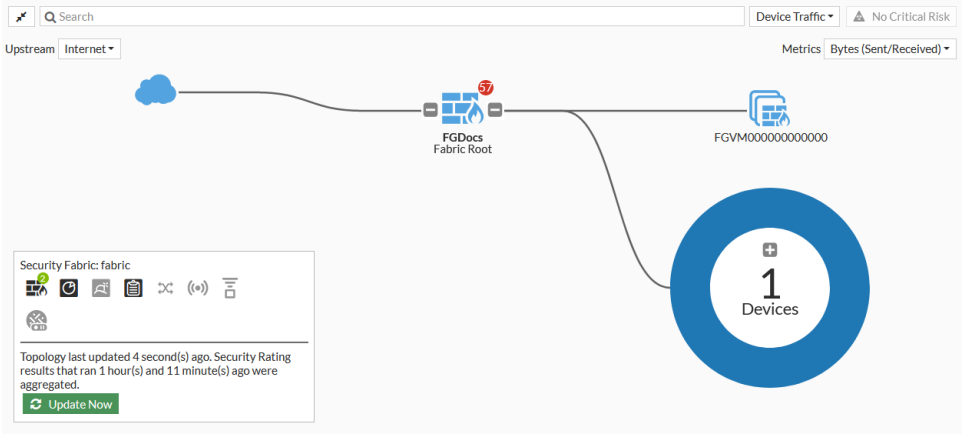
6. If *Certificate* is selected, click *Browse* to upload the downstream device's certificate from the management computer.
7. Set the *Action* to *Accept*.
8. Click *OK* and add more devices as required.
9. Click *OK*.

To configure a downstream FortiGate to connect to an upstream FortiGate:

1. Configure the downstream FortiGate:
   a. On the downstream FortiGate, go to *Security Fabric > Fabric Connectors* and double-click the *Security Fabric Setup* card.
   b. Set the *Security Fabric role* to *Join Existing Fabric*.
   c. Enter the IP address of the root FortiGate in the *Upstream FortiGate IP/FQDN* field.



   d. Click *OK*.
2. On the root FortiGate, go to *Security Fabric > Physical Topology* and verify that the downstream FortiGate that you added appears in the Security Fabric topology.

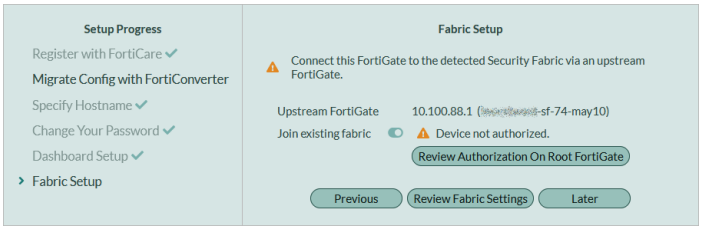


Authorizing a downstream FortiGate

When you log in to an unauthorized downstream FortiGate, the log in prompt includes the option to authorize the device on the root FortiGate.

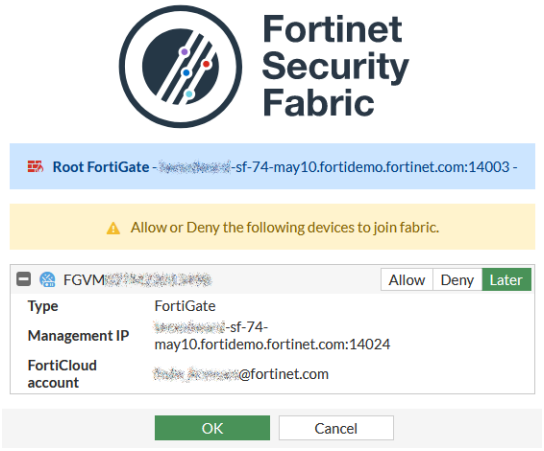
To authorize a downstream FortiGate:

1. Log in to the unauthorized, downstream device.



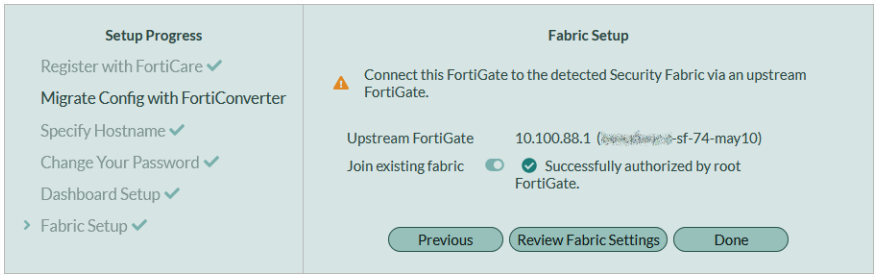
2. In the *Fabric Setup* step, click *Review Authorization on Root FortiGate*.
   A pop-up window opens to a log in screen for the root FortiGate.

3. Enter the log in credentials for the root FortiGate, then click *Login*.
   A list of pending authorizations is shown.



4. Select *Allow* and then click *OK* to authorize the downstream FortiGate. You can also select *Deny* to reject the authorization, or *Later* to postpone the decision to the next time that you log in.

   When authorization is allowed, the pop-up window closes, and the log in prompt shows that the downstream FortiGate has been authorized.



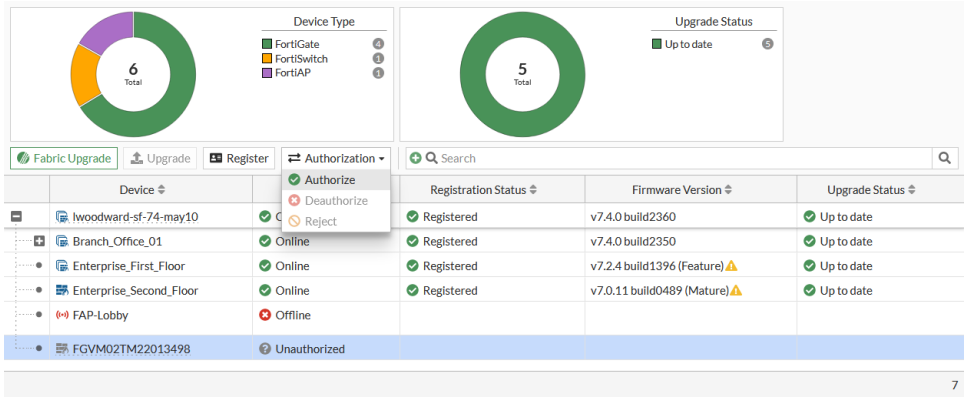
5. Click *Done* to log in to the downstream FortiGate.

Triggering authorization from the GUI

A downstream device can be authorized in the root FortiGate's GUI by using the *Firmware & Registration* page (see Authorizing devices on page 3773 for more information).

To authorize a downstream FortiGate from the Firmware & Registration page:

1. Go to *System* > *Firmware & Registration*.
2. Select the unauthorized device and click *Authorization* > *Authorize*.



A notification appears in the top-right corner once the device is authorized.

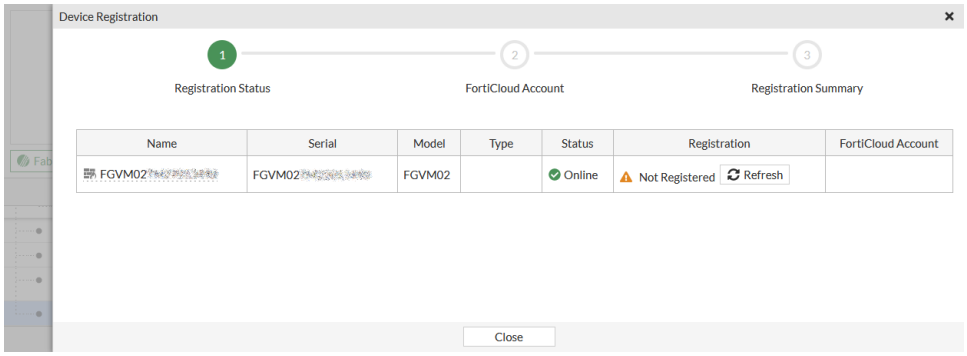
3. Refresh the page. The FortiGate's status is now *Online*.

Registering the downstream devices to FortiCloud

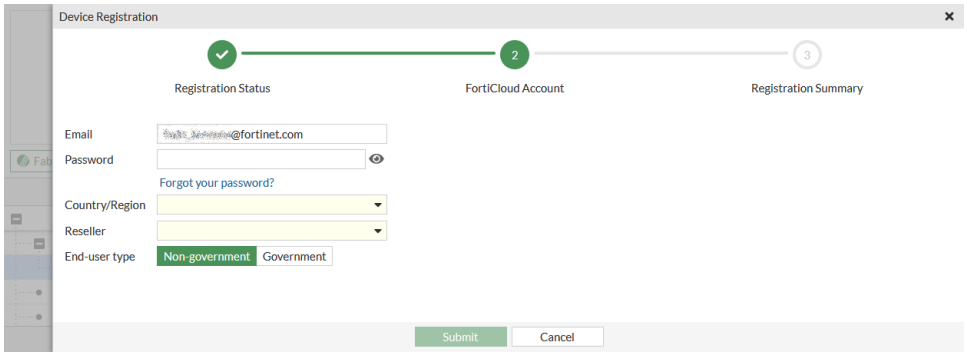
In this example, a downstream FortiGate has just been authorized, but it is not registered to a FortiCloud account. A device can be registered on the root FortiGate to a FortiCloud account.

To register the downstream FortiGate from the root:

1. Log in to the root FortiGate and go to *System > Firmware & Registration*
2. Select an unregistered device and click *Register*. The *Device Registration* pane opens.



3. Enter the required information (password, country/region, reseller, and end-user type).



4. Click *Submit*. The *Registration Summary* pane opens.
5. Click *Close*.

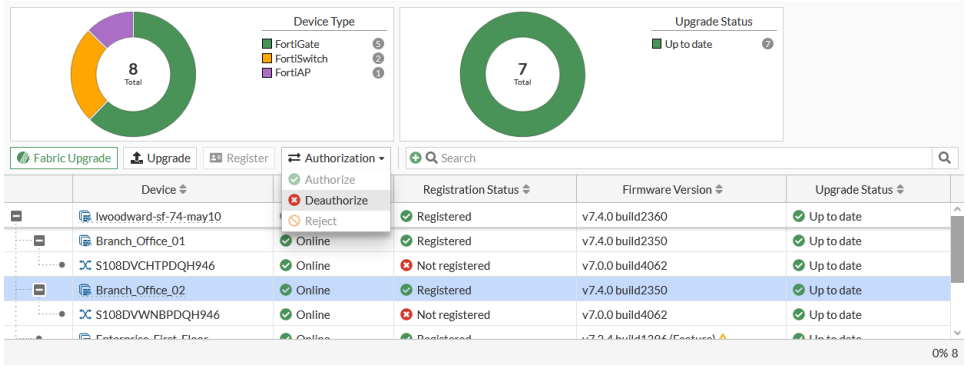
You can use IPAM to automatically assign subnets to downstream FortiGates to prevent duplicate IP addresses from overlapping within the same Security Fabric. See IPAM on page 524.

Deauthorizing a device

A device can be deauthorized to remove it from the Security Fabric in the root FortiGate's GUI by using the *Firmware & Registration* page (see Authorizing devices on page 3773 for more information).

To deauthorize a device from the Firmware & Registration page:

1. Go to *System* > *Firmware & Registration*.
2. Select the authorized device and click *Authorization* > *Deauthorize*.



A notification appears in the top-right corner once the device is deauthorized.

3. Refresh the page. The FortiGate is moved to the bottom of the list, and it's status is *Unauthorized*.

After a device is deauthorized, the serial number is saved in a trusted list that can be viewed in the CLI using the `show system csf` command. For example, this result shows a deauthorized FortiSwitch:

```
show system csf
    config system csf
        set status enable
        set group-name "Office-Security-Fabric"
        set group-password ************
        config trusted-list
            edit "FGT6HD391800000"
        next
            edit "S248DF3X1700000"
                set action deny
            next
        end
    end
```

CLI commands

Use the following commands to view, accept, and deny authorization requests, to view upstream and downstream devices, and to list or test Fabric devices:

Command	Description
`diagnose sys csf authorization pending-list`	View pending authorization requests on the root FortiGate.
`diagnose sys csf authorization accept <serial number> [name]`	Authorize a device to join the Security Fabric.

Command	Description
`diagnose sys csf authorization deny <serial number> [name]`	Deny a device from joining the Security Fabric.
`diagnose sys csf downstream`	Show connected downstream FortiGates.
`diagnose sys csf downstream-devices <device type>`	Show downstream fabric devices.
`diagnose sys csf upstream`	Show connected upstream devices.
`diagnose sys csf fabric-device list`	List all known Fabric devices.
`diagnose sys csf global`	Show a summary of all connected members in Security Fabric.

Desynchronizing settings

By default, the settings for FortiAnalyzer logging, central management, sandbox inspection, and FortiClient EMS are synchronized between all FortiGates in the Security Fabric.

To disable automatic synchronization:

```
config system csf
    set configuration-sync local
end
```

Configuring logging and analytics

FortiAnalyzer or Cloud Logging is a required component for the Security Fabric. Either FortiAnalyzer, FortiAnalyzer Cloud, or FortiGate Cloud can be used to meet this requirement. These settings are configured on the *Logging & Analytics* card on the *Security Fabric > Fabric Connectors* page. If there are multiple services enrolled on the FortiGate, the preference is: FortiAnalyzer Cloud logging, FortiAnalyzer logging, then FortiGate Cloud logging.

The following topics provide more information about configuring the logging and analytics connector:

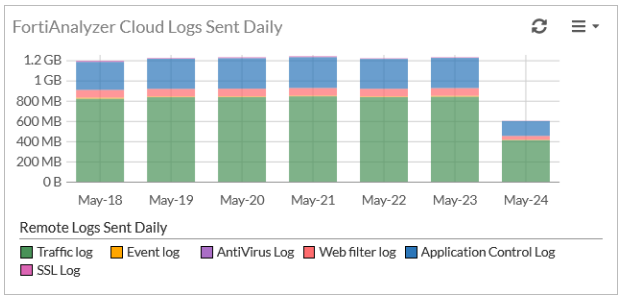
- Configuring FortiAnalyzer on page 4311
- Configuring cloud logging on page 4314

Logs Sent daily chart for remote logging sources

The *Logs Sent* widget displays a chart for a select remote logging source (FortiAnalyzer, FortiGate Cloud, and FortiAnalyzer Cloud). Once configured, the same data is available on the *FortiAnalyzer* and *Cloud Logging* tabs of the *Logging & Analytics* card.

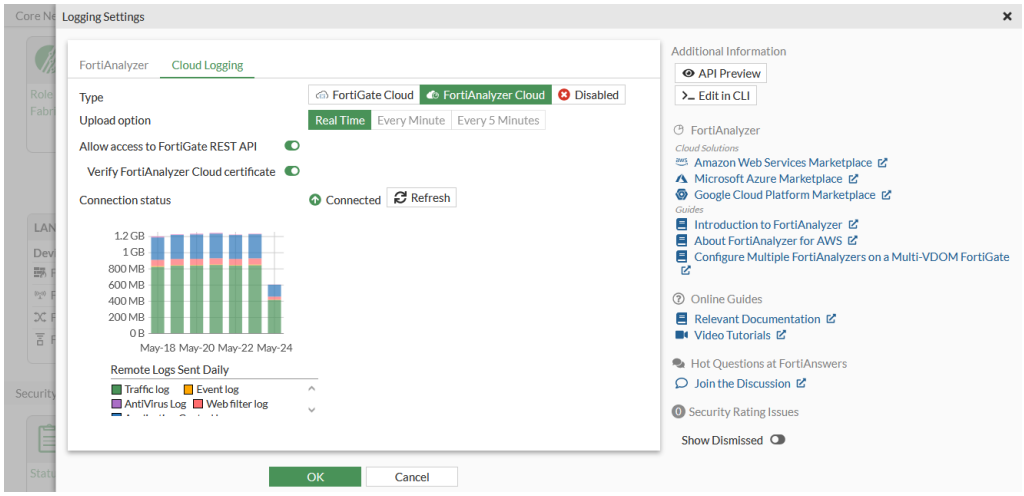
To add the Logs Sent widget:

1. Go to *Dashboard* > *Status* and click *Add Widget*.
2. Search for *Logs Sent* and click to add the widget to the dashboard.
3. Select a *Logging Source* (*FortiAnalyzer*, *FortiAnalyzer Cloud*, or *FortiGate Cloud*). *FortiAnalyzer Cloud* is used in this example.
4. Click *OK* and click *Close*. The *FortiAnalyzer Logs Sent Daily* widget is displayed in the dashboard.



To view the chart on the Logging & Analytics card:

1. Go to *Security Fabric* > *Fabric Connectors* and double-click the *Logging & Analytics* card.
   The *Logging Settings* pane is displayed.
2. Go to the *FortiAnalyzer* or *Cloud Logging* tabs to view the *Remote Logs Sent Daily* chart. *FortiAnalyzer Cloud* is used in this example.



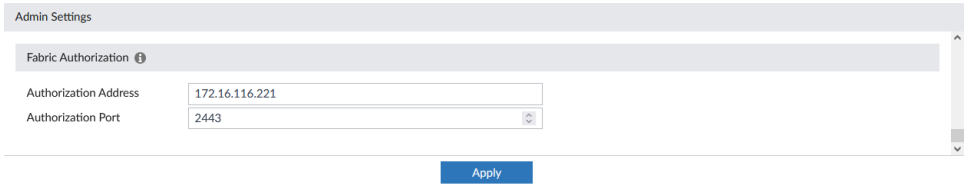
3. Click *OK* to close the pane.

Configuring FortiAnalyzer

FortiAnalyzer allows the Security Fabric to show historical data for the Security Fabric topology and logs for the entire Security Fabric. For more information about using FortiAnalyzer, see the FortiAnalyzer Administration Guide.

To authorize a FortiAnalyzer in the Security Fabric:

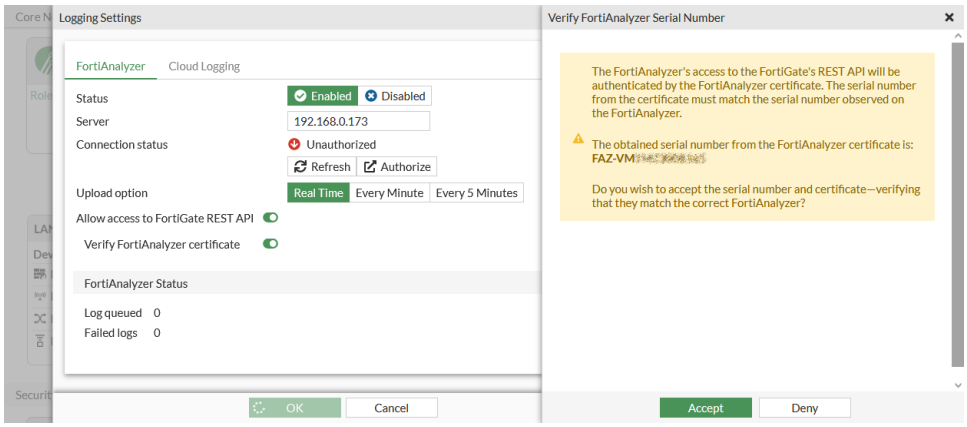
1. In FortiAnalyzer, configure the authorization address and port:
   a. Go to *System Settings* > *Settings*.
   b. In the *Fabric Authorization* section, enter an *Authorization Address* and *Authorization Port*. This is used to access the FortiAnalyzer login screen.



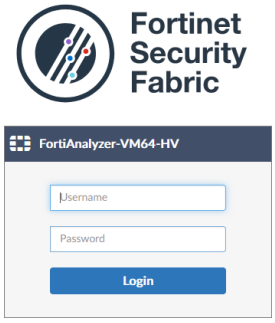
   c. Click *Apply*.
2. In FortiOS, go to *Security Fabric* > *Fabric Connectors* and double-click the *Logging & Analytics* card.
3. On the *FortiAnalyzer* tab, set the *Status* to *Enabled*.
4. Enter the FortiAnalyzer IP in the *Server* field.
5. Optionally, configure the remaining log settings:

Upload option	Select the frequency of log uploads to the remote device: • *Real Time*: logs are sent to the remote device in real time. • *Every Minute*: logs are sent to the remote device once every minute. This option is unavailable if the Security Fabric connection is configured. • *Every 5 Minutes*: logs are sent to the remote device once every five minutes. This is the default option. This option is unavailable if the Security Fabric connection is configured. • `store-and-upload`: store logs to a local disk before uploading to FortiAnalyzer or FortiManager at a scheduled time. This option is only available in the CLI.
Allow access to FortiGate REST API	Define access to FortiGate REST API: • *Enable*: the REST API accesses the FortiGate topology and shares data and results. • *Disable*: the REST API does not share data and results.
Verify FortiAnalyzer certificate	Define the FortiAnalyzer certificate verification process: • *Enable*: the FortiGate will verify the FortiAnalyzer serial number against the FortiAnalyzer certificate. When verified, the serial number is stored in the FortiGate configuration. • *Disable*: the FortiGate will not verify the FortiAnalyzer certificate against the serial number.

6. Click *OK*. The FortiAnalyzer *Connection status* is *Unauthorized* and a pane might open to verify the FortiAnalyzer's serial number.

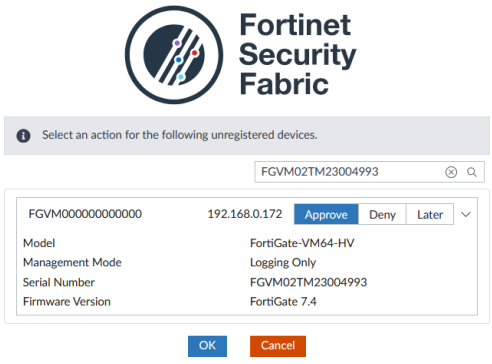


7. Click *Accept*.
8. Double-click the *Logging & Analytics* card again.
9. Click OK in the confirmation popup to open a window to authorize the FortiGate on the FortiAnalyzer.
   A login screen opens in a new browser window.
10. Enter the username and password, then click *Login*.

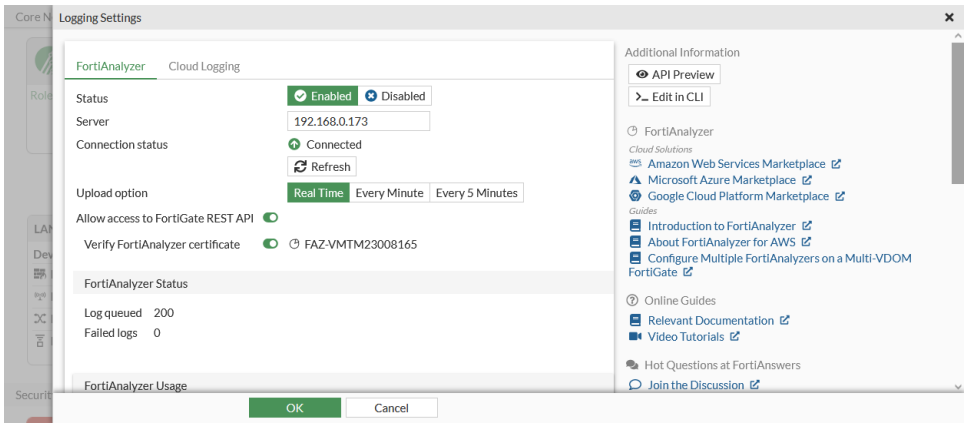


The authorization dialog opens.

11. Select *Approve* in the row for the FortiGate, and then click *OK* to authorize the FortiGate.



12. In FortiOS, *Connection status* is now *Authorized* on the *Logging Settings* pane.



FortiGates with a FortiAnalyzer Cloud license can send all logs to FortiAnalyzer Cloud.

Configuring cloud logging

There are two options available in the *Cloud Logging* tab of the *Logging & Analytics* connector card: *FortiGate Cloud* and *FortiAnalyzer Cloud*. If there are multiple services enrolled on the FortiGate, the preference is: FortiAnalyzer Cloud logging, FortiAnalyzer logging, then FortiGate Cloud logging.

This topic covers the following cloud logging aspects:

- Configuring FortiGate Cloud
  - Registration and activation
  - Enabling logging to FortiGate Cloud
  - Logging into the FortiGate Cloud portal
  - Configuring a Security Fabric with FortiGate Cloud logging
  - Cloud sandboxing
- Configuring FortiAnalyzer Cloud

Configuring FortiGate Cloud

FortiGate Cloud is a hosted security management and log retention service for FortiGate devices. It provides centralized reporting, traffic analysis, configuration management, and log retention without the need for additional hardware or software.

FortiGate Cloud offers a wide range of features:

- **Simplified central management**

  FortiGate Cloud provides a central GUI to manage individual or aggregated FortiGate and FortiWiFi devices. Adding a device to the FortiGate Cloud management subscription is straightforward. FortiGate Cloud has detailed traffic and application visibility across the whole network.
- **Hosted log retention with large default storage allocated**

  Log retention is an integral part of any security and compliance program, but administering a separate storage system is onerous. FortiGate Cloud takes care of this automatically and stores the valuable log information in the cloud. Different types of logs can be stored, including Traffic, System Events, Web, Applications, and Security Events.

- **Monitoring and alerting in real time**

  Network availability is critical to a good end-user experience. FortiGate Cloud enables you to monitor your FortiGate network in real time with different alerting mechanisms to pinpoint potential issues. Alerting mechanisms can be delivered via email.
- **Customized or pre-configured reporting and analysis tools**

  Reporting and analysis are your eyes and ears into your network's health and security. Pre-configured reports are available, as well as custom reports that can be tailored to your specific reporting and compliance requirements. The reports can be emailed as PDFs, and can cover different time periods.
- **Maintain important configuration information uniformly**

  The correct configuration of the devices within your network is essential for maintaining optimum performance and security posture. In addition, maintaining the correct firmware (operating system) level allows you to take advantage of the latest features.
- **Service security**

  All communication (including log information) between the devices and the cloud is encrypted. Redundant data centers are always used to give the service high availability. Operational security measures have been put in place to make sure your data is secure — only you can view or retrieve it.

For more information, refer to the FortiGate Cloud documentation.

When you run a function in FortiGate Cloud that applies to FortiGates, such as running a script, FortiGate Cloud may not pass the actual username of the user who performed the action to FortiOS:

- For remotely accessing a FortiGate from FortiGate Cloud, if the FortiGate Cloud administrator has enabled *Cloud Access Anonymous Mode*, FortiGate Cloud passes the username of the FortiGate Cloud user who performed the action as a randomized *@fortigatecloud.com* email address, such as *4aa567e55bc8@fortigatecloud.com*, to FortiOS. If the FortiGate Cloud admin has disabled *Cloud Access Anonymous Mode*, FortiGate Cloud passes the actual username of the FortiGate Cloud user who performed the action to FortiOS.
- For other management features that a user can perform from FortiGate Cloud, such as running a script, FortiGate Cloud passes the username of the FortiGate Cloud user who performed the action as *FortiGateCloud* to FortiOS.

Consequently, when viewing logs on the affected FortiGate, you may see *4aa567e55bc8@fortigatecloud.com* or *FortiGateCloud* as a username. For managed security service provider customers, this provides enhanced security by preventing subusers from seeing the primary account email address in the FortiGate logs.

For FortiGates managed by FortiGate Cloud, automatic firmware patch may be enabled depending on the FortiGate Cloud version and portal in use. See the Administration Guide for the applicable FortiGate Cloud version and portal:

- Standard Portal Administration Guide
- 25.1.a Portal (Beta) Administration Guide
- Premium Portal Administration Guide

Registration and activation

Before you can activate a FortiGate Cloud account, you must register your device first.

FortiGate Cloud accounts can be registered manually through the FortiGate Cloud website, https://www.forticloud.com, or you can easily register and activate your account directly from your FortiGate.

To activate your FortiGate Cloud account:

1. On your device, go to *Dashboard* > *Status*.
2. In the *FortiGate Cloud* widget, click the *Not Activated* > *Activate* button in the *Status* field.
3. A pane will open asking you to register your FortiGate Cloud account. Click *Create Account*, enter your information, view and accept the terms and conditions, and then click *OK*.
4. A second dialogue window opens, asking you to enter your information to confirm your account. This sends a confirmation email to your registered email. The dashboard widget then updates to show that confirmation is required.
5. Open your email, and follow the confirmation link it contains.

   A FortiGate Cloud page will open, stating that your account has been confirmed. The *Activation Pending* message on the dashboard will change to state the type of account you have, and will provide a link to the FortiGate Cloud portal.

Enabling logging to FortiGate Cloud

To enable logging to FortiGate Cloud:

1. Go to *Security Fabric* > *Fabric Connectors* and double-click the *Logging & Analytics* card.
2. On the *Cloud Logging* tab, set *Type* to *FortiGate Cloud*.
3. Select an upload option:
   - *Real Time*: logs are sent to the cloud device in real time.
   - *Every Minute*: logs are sent to the cloud device once every minute.
   - *Every 5 Minutes*: logs are sent to the cloud device once every five minutes (default).

   If the Security Fabric connection is configured, only the *Real Time* option is available.
4. Click *OK*.

Logging into the FortiGate Cloud portal

Once logging has been configured and you have registered your account, you can log into the FortiGate Cloud portal and begin viewing your logging results. There are two methods to reach the FortiGate Cloud portal:

- If you have direct network access to the FortiGate:
  a. Go to *Dashboard* > *Status*.
  b. In the *FortiGate Cloud* widget, in the *Status* field, click *Activated* > *Launch Portal*, or, in the *Licenses* widget, click *Support* > *Login to My Account*.
- If you do not have access to the FortiGate's interface, visit the FortiGate Cloud website (https://www.forticloud.com) and log in remotely, using your email and password. It will ask you to confirm

the FortiGate Cloud account you are connecting to and then you will be granted access.

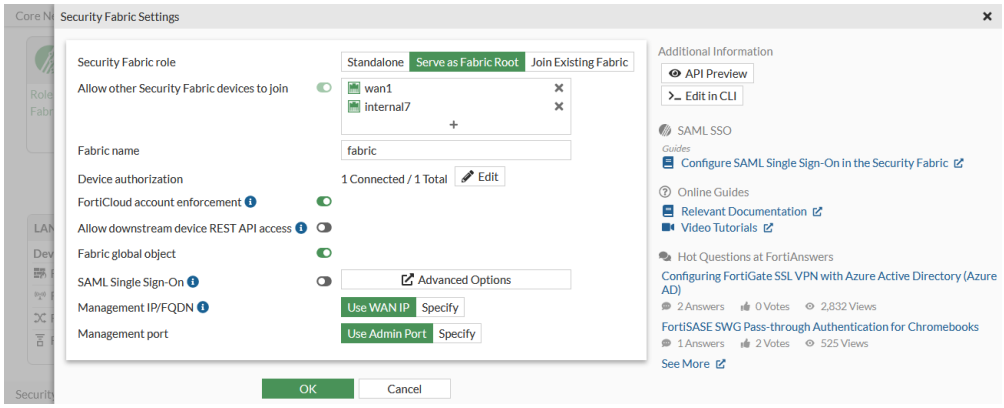
Configuring a Security Fabric with FortiGate Cloud logging

A Security Fabric can be created on the root device using FortiGate Cloud for cloud logging. When the FortiCloud account enforcement is enabled (by default), members joining the Fabric must be registered to the same FortiCloud account. Devices that are not activated with FortiCloud are also allowed.

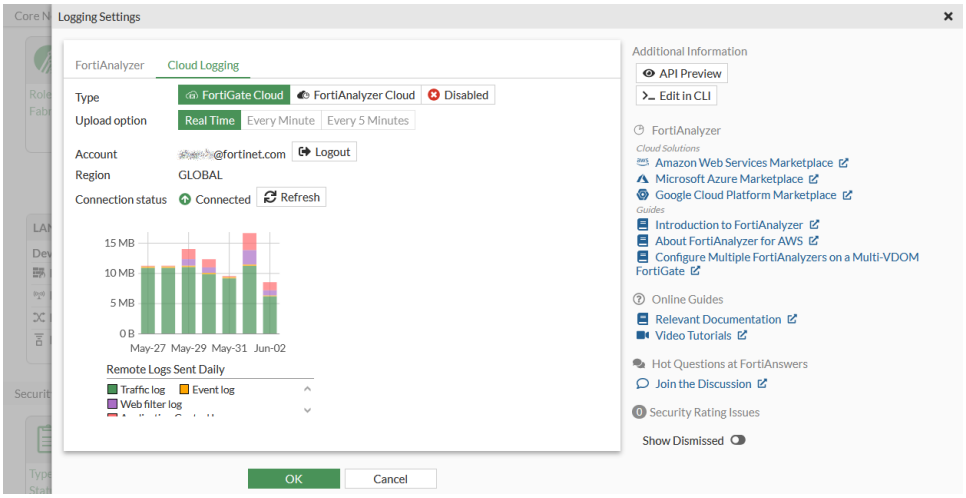
For example, the root FortiGate (FGT_10_101F) is configured with FortiGate Cloud logging. In the Security Fabric settings, the *FortiCloud account enforcement* option is enabled by default. The downstream FortiGate, FGT-F-VM, with the same FortiCloud account ID is able to join the Fabric.

To configure a Security Fabric with FortiCloud logging in the GUI:

1. Configure the Security Fabric settings on the root FortiGate (see Configuring the root FortiGate and downstream FortiGates on page 4302). The *FortiCloud account enforcement* setting is enabled by default.

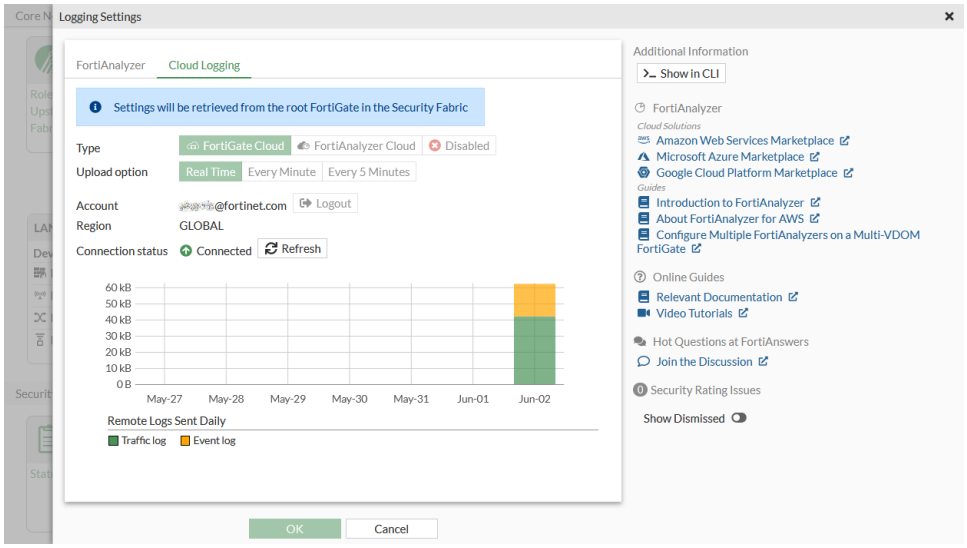


2. Configure FortiCloud logging on the root FortiGate:
   a. Go to *Security Fabric* > *Fabric Connectors* and double-click the *Logging & Analytics* card.
   b. On the *Cloud Logging* tab, set *Type* to *FortiGate Cloud*.



   c. Click *OK*.
3. Configure the FGT-F-VM to join the Security Fabric:

   a. Go to *Security Fabric* > *Fabric Connectors* and double-click the *Security Fabric Setup* card.
   b. Set *Security Fabric role* to *Join Existing Fabric*.
   c. Click *OK*. The FortiGate is authorized and successfully joins the Security Fabric.
4. Check the FortiCloud logging settings:
   a. Go to *Security Fabric* > *Fabric Connectors* and double-click the *Logging & Analytics* card.
   b. Go to the *Cloud Logging* tab. The settings are automatically retrieved from the root FortiGate and the *Account* is the same.



To configure a Security Fabric with FortiCloud logging in the CLI:

```
config log fortiguard setting
    set status enable
    set upload-option realtime
end
```

The FortiCloud account enforcement setting is enabled by default in the Security Fabric settings:

```
show system csf
    config system csf
        set status enable
        set group-name "CSF_101"
        set forticloud-account-enforcement enable
    end
```

Cloud sandboxing

FortiGate Cloud can be used for automated sample tracking, or sandboxing, for files from a FortiGate. This allows suspicious files to be sent to be inspected without risking network security. If the file exhibits risky behavior, or is found to contain a virus, a new virus signature is created and added to the FortiGuard antivirus signature database.

See Configuring sandboxing on page 4354 for instructions to configure FortiGate Cloud Sandbox. Sandboxing results are shown on the *Sandbox* tab in the FortiGate Cloud portal.

Configuring FortiAnalyzer Cloud

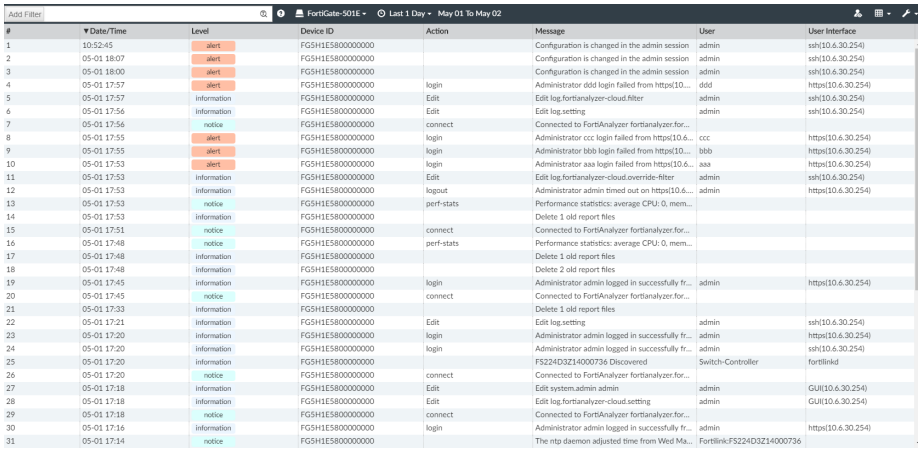
FortiAnalyzer Cloud differs from FortiAnalyzer in the following ways:

- You cannot enable FortiAnalyzer Cloud in `vdom override-setting` when global FortiAnalyzer Cloud is disabled.
- You must use the CLI to retrieve and display logs sent to FortiAnalyzer Cloud. The FortiOS GUI is not supported.
- You cannot enable FortiAnalyzer Cloud and FortiGate Cloud at the same time.

For more information, see Licensing in the FortiAnalyzer Cloud Deployment Guide.

In the *Security Fabric > Fabric Connectors > Logging & Analytics* card settings, *FortiAnalyzer Cloud* is grayed out when you do not have a FortiAnalyzer Cloud entitlement. When you have a FortiAnalyzer Cloud entitlement, *FortiAnalyzer Cloud* is available and you can authenticate by the certificate.

In FortiAnalyzer Cloud, you can view logs from FortiOS in the *Event > All Types* page.



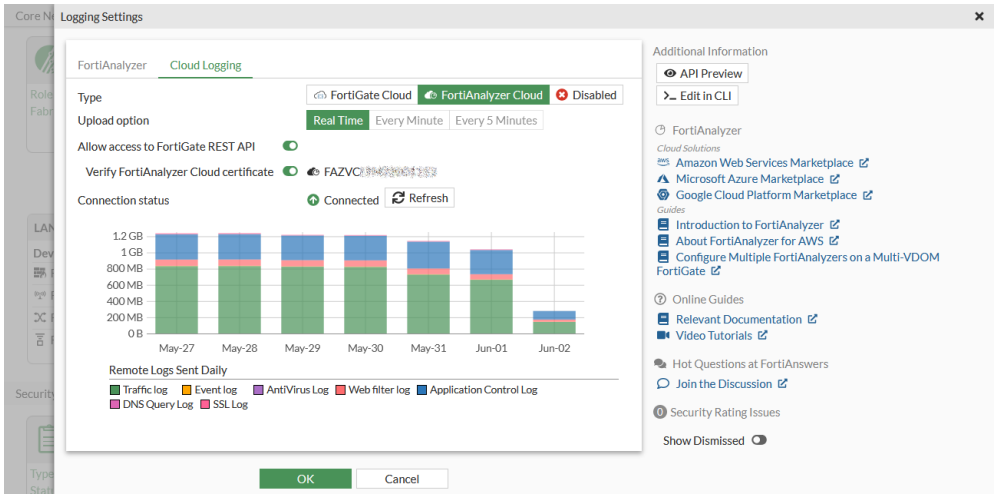
To configure FortiAnalyzer Cloud logging in the GUI:

1. Go to *Security Fabric > Fabric Connectors* and double-click the *Logging & Analytics* card.
2. Select the *Cloud Logging* tab and set the *Type* to *FortiAnalyzer Cloud*.
3. Optionally, configure the remaining log settings:

Upload option	Select the frequency of log uploads to the remote device: • *Real Time*: logs are sent to the remote device in real time. • *Every Minute*: logs are sent to the remote device once every minute. This option is unavailable if the Security Fabric connection is configured. • *Every 5 Minutes*: logs are sent to the remote device once every five minutes. This is the default option. This option is unavailable if the Security Fabric connection is configured.
Allow access to FortiGate REST API	Define access to FortiGate REST API: • *Enable*: the REST API accesses the FortiGate topology and shares data and results. • *Disable*: the REST API does not share data and results.

Verify FortiAnalyzer Cloud certificate	Define the FortiAnalyzer Cloud certificate verification process: • *Enable*: the FortiGate will verify the FortiAnalyzer Cloud serial number against the FortiAnalyzer certificate. When verified, the serial number is stored in the FortiGate configuration. • *Disable*: the FortiGate will not verify the FortiAnalyzer Cloud certificate against the serial number.

4. Click *OK*. A prompt appears to verify the FortiAnalyzer Cloud serial number.
5. Click *Accept*.
6. The verified FortiAnalyzer Cloud certificate appears in the settings.



To enable FortiAnalyzer Cloud logging in the CLI:

1. Configure the FortiAnalyzer Cloud settings:

```
config log fortianalyzer-cloud setting
    set status enable
    set ips-archive disable
    set certificate-verification enable
    set serial "FAZVCLTM19000000"
    set access-config enable
    set enc-algorithm high
    set ssl-min-proto-version default
    set conn-timeout 10
    set monitor-keepalive-period 5
    set monitor-failure-retry-period 5
    set upload-option realtime
end
```

2. Configure the FortiAnalyzer Cloud filters:

```
config log fortianalyzer-cloud filter
    set severity information
    set forward-traffic disable
    set local-traffic disable
```

```
    set multicast-traffic disable
    set sniffer-traffic disable
    set anomaly disable
    set voip disable
    set dlp-archive disable
end
```

To disable FortiAnalyzer Cloud logging for a specific VDOM in the CLI:

1. Enable override FortiAnalyzer in the general log settings:

```
config log setting
    set faz-override enable
end
```

2. Disable the override FortiAnalyzer Cloud setting:

```
config log fortianalyzer-cloud override-setting
    set status disable
end
```

To set FortiAnalyzer Cloud logging to filter for a specific VDOM in the CLI:

1. Enable override FortiAnalyzer in the general log settings:

```
config log setting
    set faz-override enable
end
```

2. Enable the override FortiAnalyzer Cloud setting:

```
config log fortianalyzer-cloud override-setting
    set status enable
end
```

3. Configure the override filters for FortiAnalyzer Cloud:

```
config log fortianalyzer-cloud override-filter
    set severity information
    set forward-traffic disable
    set local-traffic disable
    set multicast-traffic disable
    set sniffer-traffic disable
    set anomaly disable
    set voip disable
    set dlp-archive disable
end
```

To display FortiAnalyzer Cloud logs in the CLI:

```
# execute log filter device fortianalyzer-cloud
# execute log filter category event
# execute log display
```

Sample log

```
date=2019-05-01 time=17:57:45 idseq=60796052214644736 bid=100926 dvid=1027 itime="2019-05-01
17:57:48" euid=3 epid=3 dsteuid=0 dstepid=3 logver=602000890 logid=0100032002 type="event"
subtype="system" level="alert" srcip=10.6.30.254 dstip=10.6.30.9 action="login" msg="Administrator
ddd login failed from https(10.6.30.254) because of invalid user name" logdesc="Admin login
failed" sn="0" user="ddd" ui="https(10.6.30.254)" status="failed" reason="name_invalid"
method="https" eventtime=1556758666274548325 devid="FG5H1E5818900000" vd="root" dtime="2019-05-01
17:57:45" itime_t=1556758668 devname="FortiGate-501E"
```

```
date=2019-05-01 time=17:57:21 idseq=60796052214644736 bid=100926 dvid=1027 itime="2019-05-01
17:57:23" euid=3 epid=3 dsteuid=0 dstepid=3 logver=602000890 logid=0100044546 type="event"
subtype="system" level="information" action="Edit" msg="Edit log.fortianalyzer-cloud.filter "
logdesc="Attribute configured" user="admin" ui="ssh(10.6.30.254)" cfgtid=164757536
cfgpath="log.fortianalyzer-cloud.filter" cfgattr="severity[information->critical]"
eventtime=1556758642413367644 devid="FG5H1E5818900000" vd="root" dtime="2019-05-01 17:57:21"
itime_t=1556758643 devname="FortiGate-501E"
```

Configuring FortiClient EMS

The FortiGate Security Fabric root device can link to FortiClient Endpoint Management System (EMS) and FortiClient EMS Cloud (a cloud-based EMS solution) for endpoint connectors and automation. Multiple EMS servers can be added to the Security Fabric, including FortiClient EMS Cloud server. EMS settings are synchronized between all Fabric members.

FortiGates with FortiCare Premium or FortiCare Elite have 3 free seats to FortiClient EMS Cloud. See FortiCloud Subscriptions for details.

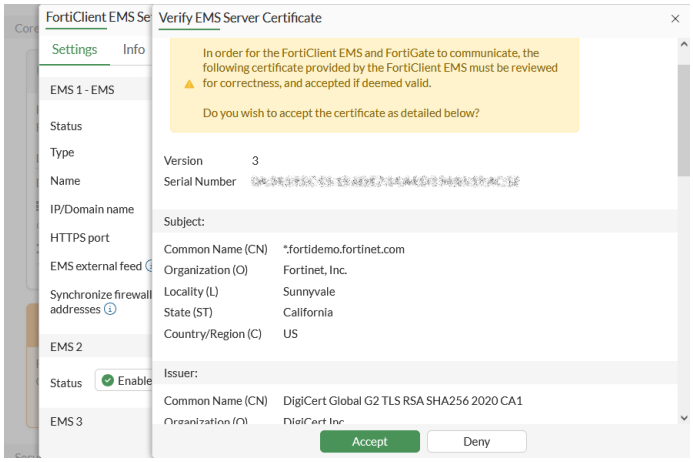
To enable cloud-based EMS services, the FortiGate must be registered to FortiCloud with an appropriate user account. The following examples presume that the EMS certificate has already been configured.

To add an on-premise FortiClient EMS server to the Security Fabric in the GUI:

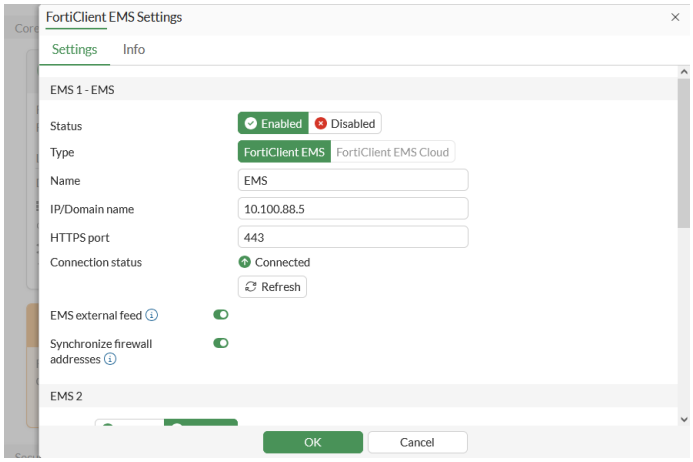
1. Go to *Security Fabric* > *Fabric Connectors* and double-click the *FortiClient EMS* card.
2. Enable an EMS, and set *Type* to *FortiClient EMS*.
3. Enter a name and IP address or FQDN.

   When connecting to a multitenancy-enabled EMS, Fabric connectors must use an FQDN to connect to EMS, where the FQDN hostname matches a site name in EMS (including "Default"). The following are examples of FQDNs to provide when configuring the connector to connect to the default site and to a site named SiteA, respectively: default.ems.yourcompany.com, sitea.ems.yourcompany.com. See Multitenancy.
4. Optionally, enable *EMS External Feed*. See Malware external feed from EMS on page 2287 for more information about using this setting in an AV profile.

5. Click *OK*.
   A window appears to verify the EMS server certificate:



6. Click *Accept*.
   The *Connection status* is now *Connected*:



7. If the device is not authorized, log in to the FortiClient EMS to authorize the FortiGate under *Administration > Fabric Devices*.
   In FortiClient 7.2.5 and above, approvals are performed under *Fabric & Connectors > Fabric Devices*.

To add a FortiClient EMS Cloud server to the Security Fabric in the GUI:

FortiClient EMS Cloud can only be configured when the FortiGate is registered to FortiCloud and the EMS Cloud entitlement is verified.

If the FortiCloud account does not pass the FortiClient EMS Cloud entitlement check, the option is not selectable in the FortiClient EMS connector settings.

1. Go to *Security Fabric > Fabric Connectors* and double-click the *FortiClient EMS* card.
2. Set the *Type* to *FortiClient EMS Cloud*.
3. Enter a name.
4. Click *OK*.
   A window appears to verify the EMS server certificate.

5. Click *Accept*.
   The *Connection status* is now *Connected*.

To test connectivity with the EMS server:

1. Go to *Security Fabric > Fabric Connectors* and double-click the *FortiClient EMS* card.
2. In the *Connection status* section, click *Refresh*.

To add an on-premise FortiClient EMS server to the Security Fabric in the CLI:

```
config endpoint-control fctems
    edit {1 | 2 | 3 | 4 | 5}
        set status {enable | disable}
        set name <string>
        set server <ip_address>
        set certificate <string>
        set https-port <integer>
        set source-ip <ip_address>
    next
end
```

The `https-port` is the EMS HTTPS access port number, and the `source-ip` is the REST API call source IP address.

To add a FortiClient EMS Cloud server to the Security Fabric in the CLI:

```
config endpoint-control fctems
    edit {1 | 2 | 3 | 4 | 5}
        set status {enable | disable}
        set name <string>
        set fortinetone-cloud-authentication enable
        set certificate <string>
    next
end
```

To verify the EMS Cloud entitlement in the CLI:

```
# diagnose test update info
```

To verify an EMS certificate in the CLI:

```
# execute fctems verify ems137

        Subject:     C = CA, ST = bc, L = burnaby, O = devqa, OU = top3, CN =
sys169.qa.fortinet.cm, emailAddress = xxxx@xxxxxxxx.xxx
        Issuer:      CN = 155-sub1.fortinet.com
        Valid from:  2017-12-05 00:37:57  GMT
        Valid to:    2027-12-02 18:08:13  GMT
        Fingerprint: D3:7A:1B:84:CC:B7:5C:F0:A5:73:3D:BB:ED:21:F2:E0
```

```
        Root CA:     No
        Version:     3
        Serial Num:
                01:86:a2
        Extensions:
                Name:     X509v3 Basic Constraints
                Critical: yes
                Content:
                CA:FALSE

                Name:     X509v3 Subject Key Identifier
                Critical: no
                Content:
                35:B0:E2:62:AF:9A:7A:E6:A6:8E:AD:CB:A4:CF:4D:7A:DE:27:39:A4

                Name:     X509v3 Authority Key Identifier
                Critical: no
                Content:
                keyid:66:54:0F:78:78:91:F2:E4:08:BB:80:2C:F6:BC:01:8E:3F:47:43:B1

DirName:/C=CA/ST=bc/L=burnaby/O=devqa/OU=top3/CN=fac155.fortinet.com/emailAddress=xyguo@fortinet.c
om
serial:01:86:A4

                Name:     X509v3 Subject Alternative Name
                Critical: no
                Content:
                DNS:sys169.qa.fortinet.cm

                Name:     X509v3 Key Usage
                Critical: no
                Content:
                Digital Signature, Non Repudiation, Key Encipherment, Data Encipherment, Key
Agreement, Certificate Sign, CRL Sign, Encipher Only, Decipher Only

                Name:     X509v3 Extended Key Usage
                Critical: no
                Content:
                TLS Web Server Authentication, TLS Web Client Authentication

EMS configuration needs user to confirm server certificate.
Do you wish to add the above certificate to trusted remote certificates? (y/n)y
```

FortiClient multi-tenancy

Multi-tenancy gives administrators the flexibility to deploy a single FortiGate with access to multiple FortiClient EMS servers, or a single FortiClient EMS with multiple tenants. The FortiGate can support up to seven EMS servers in a single VDOM. When multi-VDOM is enabled on the FortiGate, each VDOM can override the global EMS configurations to connect to their own EMS servers.

The override feature requires FortiClient EMS 7.2.1 and later, and FortiGate running FOS 7.4.0 or later. To use override with FortiClient EMS Cloud, a FortiGate must be running FOS 7.4.4 or later. To connect to a FortiClient Cloud instance registered under a FortiCloud sub-OU in the GUI, a FortiGate must be running FOS 7.6.1 or later.

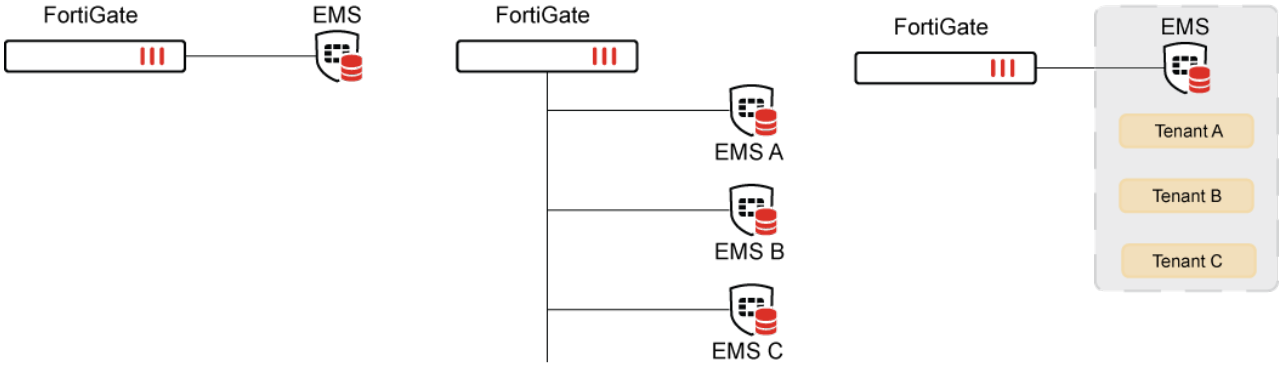
This functionality can be applied to MSSP (managed security service provider) configurations, and each VDOM has its own *FortiClient EMS* card for the EMS server or instance.

The following reference table provides a high-level view of single versus multi-tenancy scenarios, depending on the status of the FortiGate (whether VDOM is enabled or disabled) and FortiClient EMS:

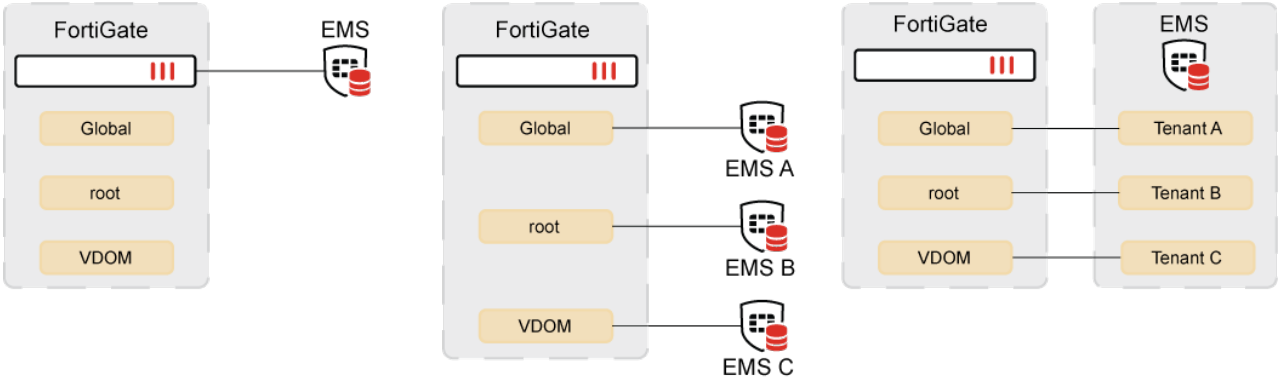
FortiGate	Single FortiClient EMS instance	Multiple FortiClient EMS instances	Single FortiClient EMS instance with multi-tenancy
No VDOM	Single tenant	Multi-tenancy without override	Multi-tenancy without override
VDOM	Global FortiClient EMS multi-tenancy	Multi-tenancy with override	Multi-tenancy with override

These scenarios can further be demonstrated as follows:

- No VDOM:



- VDOM:



Basic configurations

In a single VDOM configuration, you can configure multiple EMS servers.

To configure FortiClient EMS servers in a single VDOM set up:

1. Go to *Security Fabric > Fabric Connectors*.
2. Double-click on the *FortiClient EMS* card to edit.
3. For each EMS server, click *Enabled*, and fill in the configurations for that EMS server.
4. Click *OK* to save the settings.

In a multi-VDOM configuration, first configure the global EMS configurations, then configure override on each VDOM. If a VDOM does not enable override, it will inherit the global configurations.

To configure FortiClient EMS servers in a multi-VDOM set up:

1. In the Global VDOM, go to *Security Fabric > Fabric Connectors*.
2. Double-click on the *FortiClient EMS* card to edit.
3. For each EMS server, click *Enabled*, and fill in the configurations for that EMS server.
4. Click *OK* to save the settings.
5. Enter a VDOM.
6. From the CLI, edit the following settings:

```
config endpoint-control settings
    set override enable
end
```

7. Back in the GUI, go to *Security Fabric > Fabric Connectors*.
8. Configure each EMS server as needed.
9. Click *OK* to save.

Advanced configurations

FortiGate supports connecting to a FortiClient Cloud instance registered under a sub-OU in FortiCloud. Furthermore, a FortiGate can override FortiClient Cloud access key setting on a per-VDOM basis. With these enhancements, a FortiGate can support FortiClient Cloud in multi-tenancy scenarios.

This feature includes the following scope and limitations:

- The FortiGate will perform an entitlement check on the registered FortiCloud Account to verify a FortiClient Cloud entitlement exists on the root FortiCloud account. If the FortiGate has no FortiClient Cloud entitlement, you cannot select the FortiClient EMS Cloud type or input an access key.
- Using the FortiClient Cloud access key, a FortiGate can connect to a FortiClient Cloud instance belonging to a sub-OU in the same FortiCloud account or a different FortiCloud account.
- Within the same VDOM, the FortiGate can have an EMS connector connecting to multiple FortiClient Cloud instances.

The FortiClient Cloud access key can be implemented in the `cloud-authentication-access-key` parameter in the CLI.

```
config endpoint-control fctems-override
    edit 1
        set status enable
        set name <name>
        set fortinetone-cloud-authentication enable
        set cloud-authentication-access-key <key>
    next
end
```

Examples

Example 1: Enabling override on the root VDOM using the CLI

To enable override on the root VDOM in the CLI:

1. Enable override on the required VDOMs:

```
config endpoint-control settings
    set override enable
end
```

2. Configure the EMS server on the desired VDOM:

```
(root) config endpoint-control fctems-override
    edit 1
        set status enable
        set name "ems140_root"
        set server "172.16.200.140"
        set serial-number "FCTEMS8821******"
        set tenant-id "00000000000000000000000000000000"
        set capabilities fabric-auth silent-approval websocket websocket-malware push-ca-certs
common-tags-api tenant-id single-vdom-connector
    next
    edit 2
        set name "ems133_root"
        set server "172.16.200.133"
    next
end
```

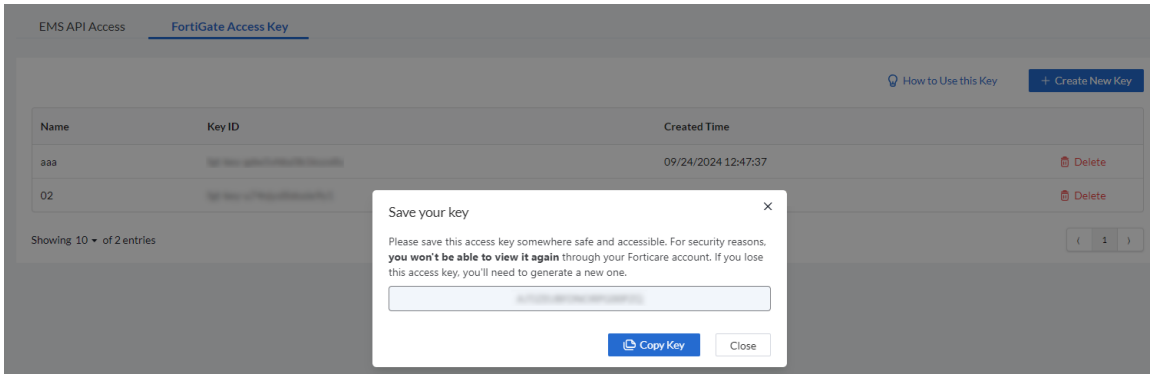
Example 2: Connecting to different FortiClient Cloud instances

In this example, a FortiGate will connect to different FortiClient Cloud instances between the Global EMS connector, root and vdom1.

To connect to different FortiClient Cloud instances in the CLI:

1. Obtain the access by from FortiClient Cloud by going to *FortiCloud > FortiClient Cloud*.
2. Click *Access Key* and switch to the *FortiGate Access Key* tab.

3. Click *Create New Key* to generate a new key.



4. Repeat this for another FortiClient Cloud instance to be applied to vdom1.
5. On the FortiGate with multi-VDOM enabled, configure the Global EMS connector:

```
config global
    config endpoint-control fctems
        edit 2
            set status enable
            set name "Cloud_EMS_Global"
            set fortinetone-cloud-authentication enable
            set serial-number "FCTEMSXXXXXXXXXX"
            set tenant-id "00000000000000000000000000000000"
        next
    end
end
```

6. Switch to and configure the root VDOM:

```
config vdom
    edit root
        config endpoint-control settings
            set override enable
        end
        config endpoint-control fctems-override
            edit 1
                set status enable
                set name "cloud_ems_root"
                set fortinetone-cloud-authentication enable
                set cloud-authentication-access-key "XXXXXXXXXXXXXXXXXXXX"
                set serial-number "FCTEMSXXXXXXXXXX"
                set tenant-id "00000000000000000000000000000000"
            next
        end
    next
end
```

7. Repeat the same steps for vdom1:

```
config vdom
    edit vdom1
```

```
        config endpoint-control settings
            set override enable
        end
        config endpoint-control fctems-override
            edit 1
                set status enable
                set name "cloud_vdom1"
                set fortinetone-cloud-authentication enable
                set cloud-authentication-access-key "XXXXXXXXXXXXXXXXXXX"
                set serial-number "FCTEMSXXXXXXXXXX"
                set tenant-id "00000000000000000000000000000000"
            next
        end
    next
end
```

8. From the CLI, run the following commands to troubleshoot.

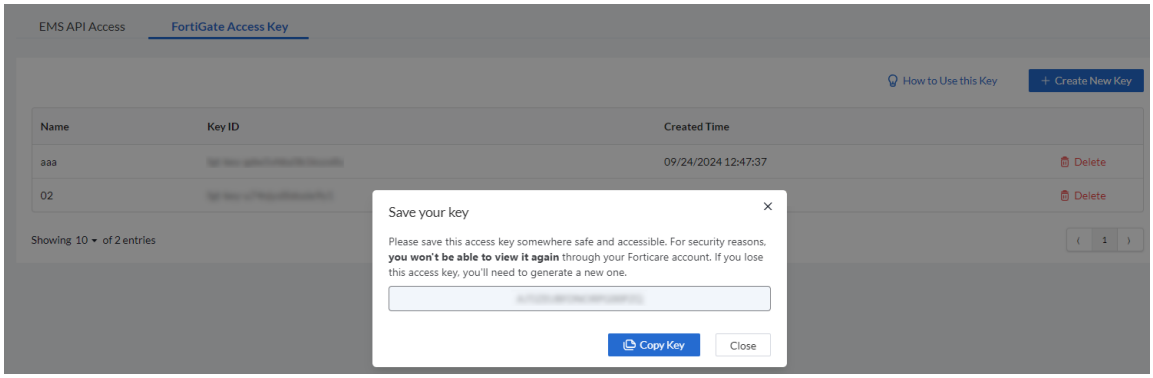
```
# diagnose endpoint filter show-large-data yes
# diagnose debug application fcnacd -1
# diagnose debug enable
```

A successful connection will look like the following:

```
…
[ec_ez_worker_base_prep_resolver:382] Outgoing interface index 0 for 1 (cloud_vdom1).
[ec_ez_worker_prep_data_url:190] Full URL: https://sf.00000-XXXXXXXXXXXXXXXXXXX.fortinet-
ca2.fortinet.com/api/v1/system/serial_number
[ec_ez_worker_base_prep_ssl:429] verify peer method: 3, current ssl_cb: (nil), new ssl_cb:
0x55c1163571b0
[ec_ems_context_submit_work:642] Call submitted successfully.
    obj-id: 0, desc: REST API to get EMS Serial Number., entry: api/v1/system/serial_number.
[__match_server_cert_key:462] verify_peer_method: 3
```

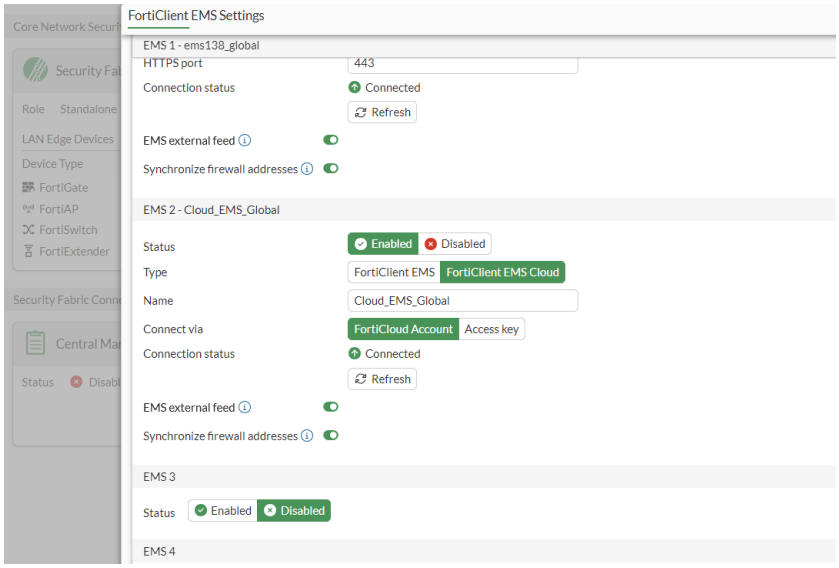
To connect to different FortiClient Cloud instances in the GUI:

1. Obtain the access by from FortiClient Cloud by going to *FortiCloud > FortiClient Cloud*.
2. Click *Access Key* and switch to the *FortiGate Access Key* tab.
3. Click *Create New Key* to generate a new key.

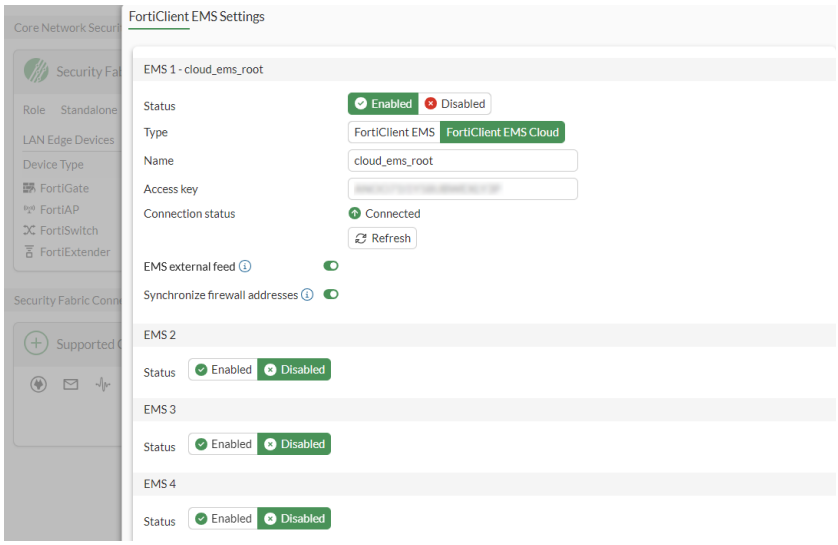


4. Repeat this for another FortiClient Cloud instance to be applied to vdom1.

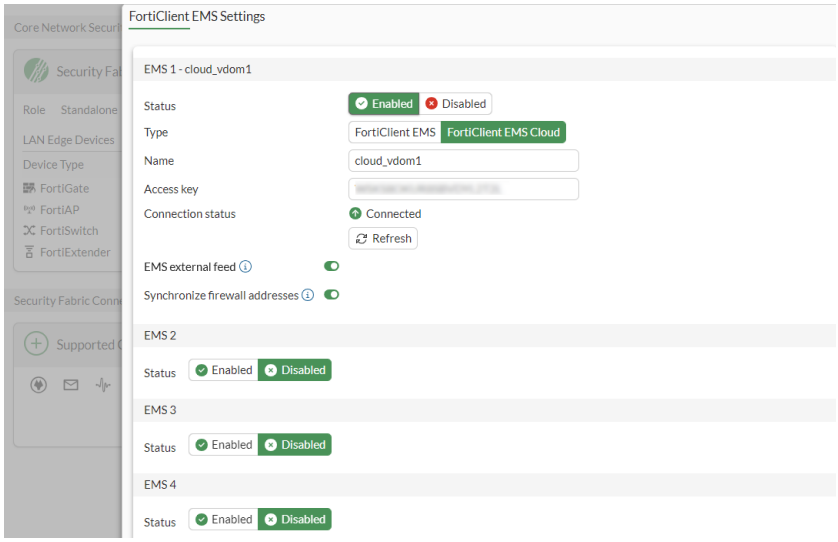
5. On the FortiGate with multi-vdom enabled, go to *Global*.
   a. Go to *Security Fabric* > *Fabric Connectors*.
   b. Edit the *FortiClient EMS* connector.
   c. Set *Status* to *Enabled*.
   d. Set *Type* to *FortiClient EMS Cloud*.
   e. Set *Name* to *Cloud_EMS_Global*.
   f. Set *Connect via* to *FortiCloud Account*.
   g. Click *OK* to save. Verify the certificate when prompted and continue saving the settings.



6. Switch to the root vdom.
   a. Go to *Security Fabric* > *Fabric Connectors*.
   b. Edit the *FortiClient EMS* connector.
   c. Set *Status* to *Enabled*.
   d. Set *Type* to *FortiClient EMS Cloud*.
   e. Set *Name* to *cloud_ems_root*.
   f. Set *Access key* to the key retrieved from FortiClient Cloud.
   g. Click *OK* to save.



7. Repeat the same steps above for vdom1.



FortiClient EMS capabilities

FortiClient EMS supports many capabilities that are integrated with the FortiGate though the EMS connector. New versions of FortiClient EMS may support new capabilities. The FortiGate can detect and synchronize the capabilities from each EMS server as it establishes the EMS connection.

Use the `config endpoint-control fctems` command on FortiGate to enable the following capabilities:

Capability	Description
fabric-auth	Allow this FortiGate to load the authentication page provided by EMS to authenticate itself with EMS.
silent-approval	Allow silent approval of non-root or FortiGate HA clusters on EMS in the Security Fabric.

Capability	Description
	For an example, see Using EMS silent approval in the Security Fabric on page 4333.
websocket	Enable/disable websockets for this FortiGate. Override behavior using websocket-override.
websocket-malware	Allow this FortiGate to request malware hash notifications over websocket.
push-ca-certs	Enable/disable syncing deep inspection certificates with EMS. For an example, see Allowing deep inspection certificates to be synchronized to EMS and distributed to FortiClient on page 4334.
common-tags-api	Can receive tag information from New Common Tags API from EMS.
tenant-id	Allow this FortiGate to retrieve Tenant-ID from EMS.
client-avatars	Allow this FortiGate to retrieve avatars from EMS by fingerprint.
single-vdom-connector	Allow this FortiGate to create a VDOM connector to EMS.
fgt-sysinfo-api	Allow this FortiGate to send additional info to EMS.
ztna-server-info	Allow this FortiGate to send VDOM's ZTNA server information to EMS.
used-tags	Allow this FortiGate to send used-tag information to EMS through the FortiOS REST API. Requires FortiClient EMS 7.4.4 or later. With the `used-tags` capability set, FortiGate uses the REST API (`fabric/tag_list`) to: • Send the updated used-tag list to EMS when a tag is added, updated, or removed from FortiGate • Send an empty used-tag list to EMS when a VDOM is added or deleted on FortiGate • Send all used tags from all VDOMs on FortiGate • Send all used tags to EMS after a connection between FortiGate and EMS is restored after being down.

Using EMS silent approval in the Security Fabric

FortiClient EMS with Fabric authorization and silent approval capabilities can approve the root FortiGate in a Security Fabric once, and then silently approve remaining downstream FortiGates in the Fabric. Similarly in an HA scenario, an approval only needs to be made once to the HA primary unit. The remaining cluster members are approved silently.

To use EMS silent approval:

1. Configure the EMS entry on the root FortiGate or HA primary:

```
config endpoint-control fctems
    edit 1
        set status enable
```

```
        set name "ems139"
        set fortinetone-cloud-authentication disable
        set server "172.16.200.139"
        set https-port 443
        set source-ip 0.0.0.0
        set pull-sysinfo enable
        set pull-vulnerabilities enable
        set pull-tags enable
        set pull-malware-hash enable
        unset capabilities
        set call-timeout 30
        set websocket-override disable
    next
end
```

When the entry is created, the capabilities are unset by default.

2. Authenticate the FortiGate with EMS:

```
# execute fctems verify ems_139
...
```

The FortiGate will enable the Fabric authorization and silent approval based on the EMS supported capabilities.

```
config endpoint-control fctems
    edit 1
        set server "172.18.62.12"
        set capabilities fabric-auth silent-approval websocket
    next
end
```

3. Configure a downstream device in the Security Fabric (see Configuring the root FortiGate and downstream FortiGates on page 4302 for more details). The downstream device will be silently approved.
4. Configure a secondary device in an HA system (see HA active-passive cluster setup on page 4079 and HA active-active cluster setup on page 4081 for more details). The secondary device will be silently approved.

Allowing deep inspection certificates to be synchronized to EMS and distributed to FortiClient

On FortiClient EMS versions that support `push CA certs` capability, the FortiGate will push CA certificates used in SSL deep inspection (see Deep inspection on page 2697 for more details) to the EMS server. On the EMS server, the CA certificates can be selected in the managed endpoint profiles so they can be installed on managed endpoints. FortiClient EMS 7.0.1 and later is required to use this feature.

To configure deep inspection certificate synchronization to EMS:

1. Configure the EMS Fabric connector:

```
config endpoint-control fctems
    edit 2
        set status enable
```

```
        set name "ems138"
        set fortinetone-cloud-authentication disable
        set server "172.16.200.138"
        set https-port 443
        set source-ip 0.0.0.0
        set pull-sysinfo enable
        set pull-vulnerabilities enable
        set pull-tags enable
        set pull-malware-hash enable
        set capabilities fabric-auth silent-approval websocket websocket-malware push-ca-certs
        set call-timeout 30
        set websocket-override disable
    next
end
```

2. Apply the certificate to an SSL/SSH profile for deep inspection:

```
config firewall ssl-ssh-profile
    edit "deep-inspection"
        set comment "Read-only deep inspection profile."
        config https
            set ports 443
            set status deep-inspection
        end
        ...
        set server-cert-mode re-sign
        set caname "Fortinet_CA_SSL"
        set untrusted-caname "Fortinet_CA_Untrusted"
    next
end
```

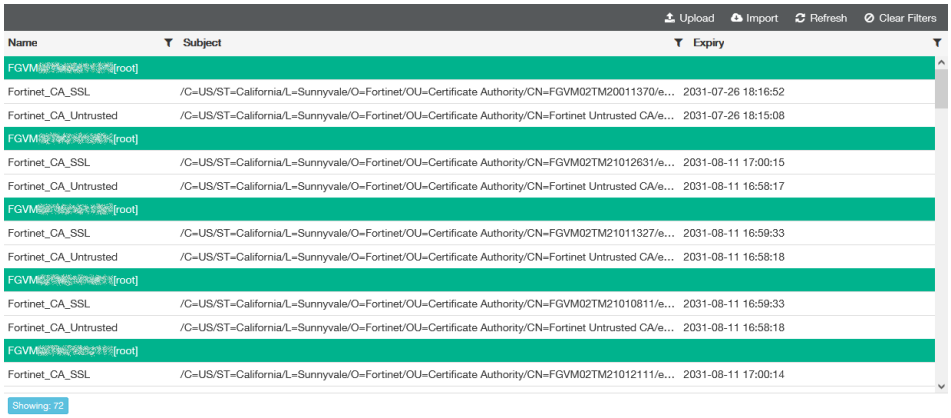
The default deep inspection profile, CA certificate, and untrusted CA certificates are used in this example.

3. Configure the firewall policy:

```
config firewall policy
    edit 1
        set name "deep-inspection"
        set srcintf "port14"
        set dstintf "port13"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "deep-inspection"
        set av-profile "default"
        set nat enable
    next
end
```

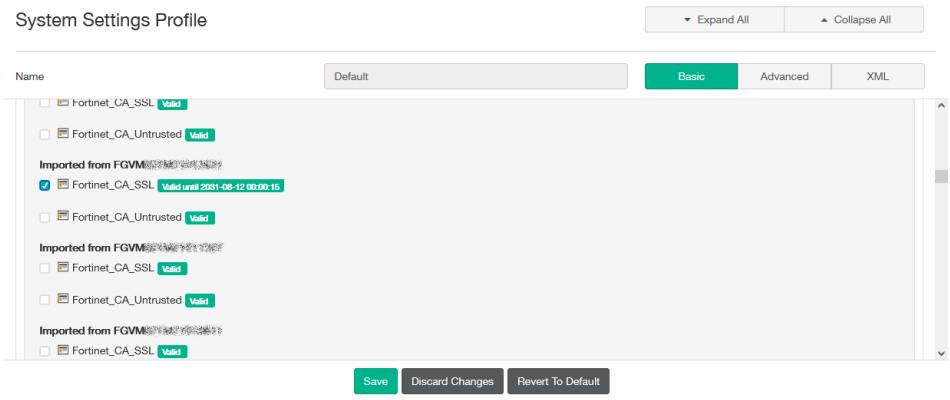
4. In EMS, verify that the CA certificate was pushed to EMS:

a. Go to *Endpoint Policy & Components > CA Certificates*.



b. Verify the certificate table to see that the EMS server received the CA certification from the different FortiGates.

5. Select the CA certificate in the endpoint profile:
   a. Go to *Endpoint Profiles > System Settings* and edit a profile. The default profile is used in this example.
   b. In the *Other* section, enable *Install CA Certificate on Client* and select the *Fortinet_CA_SSL* certificate for the desired endpoint.



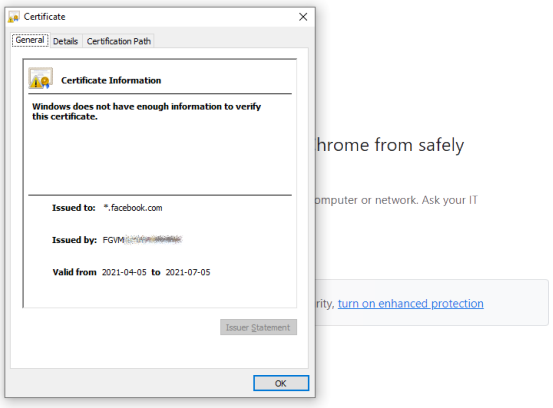
   c. Click *Save*.

      After the FortiClient endpoint is registered, it receives the CA certificate. When the FortiClient endpoint tries to access the internet through the FortiGate with the firewall policy that has deep inspection, no warning message is displayed. The server certificate is trusted with the installed CA certificate to complete the certificate chain.

Verification

Before configuring deep inspection certificate synchronization, a warning message is displayed when a FortiClient endpoint accesses the internet through the FortiGate with the firewall policy that has deep inspection. The FortiClient certificate store does not have the FortiGate's CA that is used in the deep inspection SSL/SSH profile.

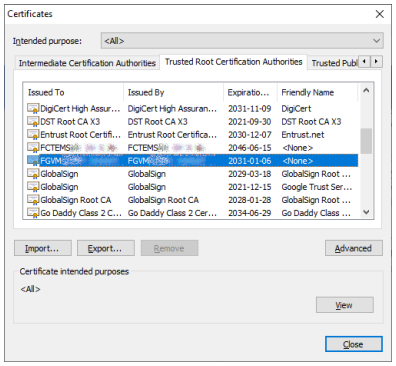
For example, accessing https://www.facebook.com in Chrome shows a warning. In the address bar, clicking *Not secure > Certificate* opens the *Certificate* dialog, which indicates that *Windows does not have enough information to verify the certificate.*



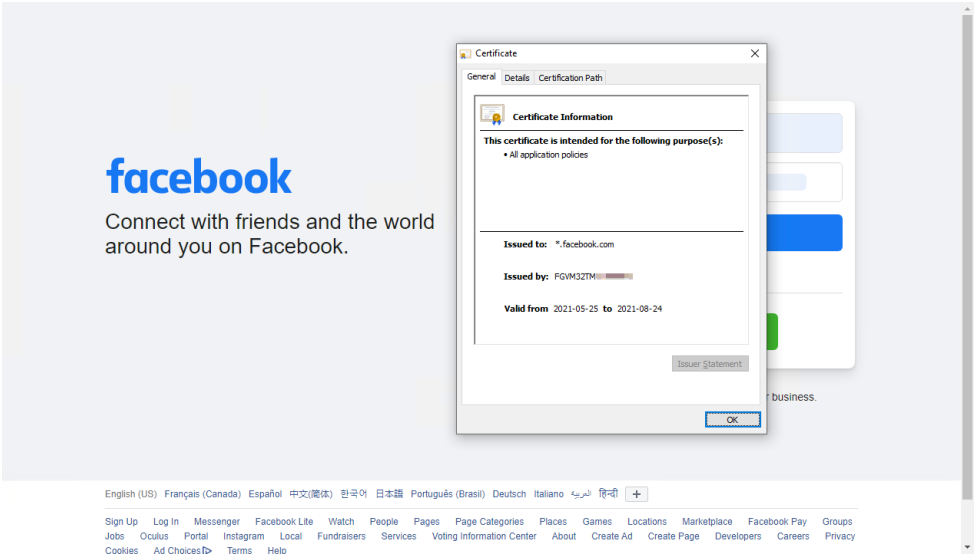
After the EMS profile is pushed to FortiClient endpoint, the expected FortiGate's certificate is shown in its certificate store.

To verify the deep inspection certificate synchronization:

1. In Chrome, go to *Settings* > *Privacy and security* and open *Manage certificates*.
2. Click the *Trusted Root Certification Authorities* tab. The FortiGate's certificate appears in the list.



3. On the FortiClient endpoint using Chrome, go to https://www.facebook.com. The website is displayed.
4. In the address bar, click the padlock, then click *Certificate*. The dialog displays the valid certificate information.



Diagnostics

Use the `diagnose endpoint fctems json deep-inspect-cert-sync` command in FortiOS to verify the certificate information. In the following example, there are multiple VDOMs with FortiGates in HA mode.

To verify the primary FortiGate:

```
FGT_EC_Primary (global) # diagnose endpoint fctems json deep-inspect-cert-sync
JSON:
"""
{
  "fortigates":[
    "FG2K5E39169*****",
    "FG2K5E39169*****"
  ],
  "vdoms":[
    {
      "vdom":"root",
      "certs":[
        {
          "name":"Fortinet_CA_SSL",
          "cert":"-----BEGIN CERTIFICATE-----\\nMIID5jCCAs6g...Sfu+Q8zE8Crmt6L1X\/bv+q\\n-----END
CERTIFICATE-----\\n"
        },
        {
          "name":"Fortinet_CA_Untrusted",
          "cert":"-----BEGIN CERTIFICATE-----\\nMIID8DCCAtig...3zBbfzP+nVUpC\\nZDPRZA==\\n-----END
CERTIFICATE-----"
        }
      ]
    },
    {
      "vdom":"vdom1",
```

```
      "certs":[
        {
          "name":"Fortinet_CA_SSL",
          "cert":"-----BEGIN CERTIFICATE-----\\nMIID5jCCAs6g...Sfu+Q8zE8Crmt6L1X\/bv+q\\n-----END
CERTIFICATE-----\\n"
        },
        {
          "name":"Fortinet_CA_Untrusted",
          "cert":"-----BEGIN CERTIFICATE-----\\nMIID8DCCAtig...3zBbfzP+nVUpC\\nZDPRZA==\\n-----END
CERTIFICATE-----"
        }
      ]
    }
  ]
}
"""
```

To verify the secondary FortiGate:

```
FGT_EC_Secondary(global) # diagnose endpoint fctems json deep-inspect-cert-sync
JSON:
"""
{
  "fortigates":[
    "FG2K5E39169*****",
    "FG2K5E39169*****"
  ],
  "vdoms":[
    {
      "vdom":"root",
      "certs":[
        {
          "name":"Fortinet_CA_SSL",
          "cert":"-----BEGIN CERTIFICATE-----\\nMIID5jCCAs6g...Sfu+Q8zE8Crmt6L1X\/bv+q\\n-----END
CERTIFICATE-----\\n"
        },
        {
          "name":"Fortinet_CA_Untrusted",
          "cert":"-----BEGIN CERTIFICATE-----\\nMIID8DCCAtig...3zBbfzP+nVUpC\\nZDPRZA==\\n-----END
CERTIFICATE-----"
        }
      ]
    },
    {
      "vdom":"vdom1",
      "certs":[
        {
          "name":"Fortinet_CA_SSL",
          "cert":"-----BEGIN CERTIFICATE-----\\nMIID5jCCAs6g...Sfu+Q8zE8Crmt6L1X\/bv+q\\n-----END
CERTIFICATE-----\\n"
        },
        {
```

```
          "name":"Fortinet_CA_Untrusted",
          "cert":"-----BEGIN CERTIFICATE-----\\nMIID8DCCAtig...3zBbfzP+nVUpC\\nZDPRZA==\\n-----END
CERTIFICATE-----"
        }
      ]
    }
  ]
}
"""
```

Allowing FortiClient EMS connectors to trust EMS server certificate renewals based on the CN field

When a FortiGate establishes a Fabric connection with FortiClient EMS, the FortiGate must trust the CA that signed the server certificate. Upon the approval of the EMS certificate, the FortiGate saves the CN field and will trust future certificates that are signed by the same CA and have the same CN field. This allows EMS servers to update their certificates at regular intervals without requiring re-authorization on the FortiGate side, as long as the CN field matches. This prevents interruptions to the EMS Fabric connection when a certificate is updated. Previously, upon the user's approval of the certificate, the certificate fingerprint was saved on the FortiGate, and this required the FortiGate to re-authorize the EMS connection each time the server certificate is updated.

```
config endpoint-control fctems
    edit <id>
        set trust-ca-cn {enable | disable}
    next
end
```

This feature is supported for EMS on-premise and cloud connections, and is the default setting. To authorize based on the certificate fingerprint, disable the `trust-ca-cn` setting. If the setting is changed back to be enabled at a later time, the user will have to re-approve the EMS certificate.

To configure the EMS Fabric connector to trust EMS server certificate renewals based on the CN field:

```
config endpoint-control fctems
    edit 1
        set status enable
        set name "ems133"
        set dirty-reason none
        set fortinetone-cloud-authentication disable
        set server "172.18.62.35"
        set https-port 443
        set serial-number "FCTEMS8822000000"
        set tenant-id "00000000000000000000000000000000"
        set source-ip 0.0.0.0
        set pull-sysinfo enable
        set pull-vulnerabilities enable
        set pull-tags enable
        set pull-malware-hash enable
        set capabilities fabric-auth silent-approval websocket websocket-malware push-ca-certs
```

```
common-tags-api tenant-id
        set call-timeout 30
        set out-of-sync-threshold 180
        set websocket-override disable
        set interface-select-method auto
        set trust-ca-cn enable
    next
end
```

To verify the configuration:

1. Download the FortiGate configuration file.
2. Verify the `ca-cn-info` entry, which lists the trusted CA certificate information. In this example, ems133 connector has `trust-ca-cn` enabled and ems138 connector has `trust-ca-cn` disabled. For ems138, the `ca-cn-info` entry does not appear, and there is a `certificate-fingerprint` field instead:

```
config endpoint-control fctems
    edit 1
        set status enable
        set name "ems133"
        set server "172.18.62.35"
        set serial-number "FCTEMS8822000000"
        set tenant-id "00000000000000000000000000000000"
        set capabilities fabric-auth silent-approval websocket websocket-malware push-ca-certs
common-tags-api tenant-id
        set ca-cn-info "C = CA, ST = BC, L = VANCOUVER, O = FTNT, OU = ReleaseQA, CN =
Release_QA, emailAddress = ********@fortinet.comRelease_QA"
    next
    edit 2
        set status enable
        set name "ems138"
        set server "172.18.62.18"
        set serial-number "FCTEMS8821000000"
        set tenant-id "00000000000000000000000000000000"
        set capabilities fabric-auth silent-approval websocket websocket-malware push-ca-certs
common-tags-api tenant-id
        set certificate-fingerprint
"18:51:76:67:EB:4C:31:A1:51:3F:74:F7:8E:1D:47:5C:18:0F:FE:45:DF:52:91:52:37:0B:27:E7:F1:85:5B:
01:8C:7D:FB:2D:C7:D2:CC:FE:4A:E3:0E:A9:2A:1C:27:4D:D2:A6:C5:87:B8:97:98:57:75:10:15:28:EF:A2:2
3:7C"
        set trust-ca-cn disable
    next
    ...
end
```

3. Run diagnostics to view the certificate information:

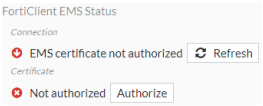
```
# diagnose test application fcnacd 96
ems_id 1, certificate authority and common name: C = CA, ST = BC, L = VANCOUVER, O = FTNT, OU
= ReleaseQA, CN = Release_QA, emailAddress = ********@fortinet.comRelease_QA
ems_id 1, fingerprint_sha512:
```

```
ems_id 2, certificate authority and common name:
ems_id 2, fingerprint_sha512:
18:51:76:67:EB:4C:31:A1:51:3F:74:F7:8E:1D:47:5C:18:0F:FE:45:DF:52:91:52:37:0B:27:E7:F1:85:5B:0
1:8C:7D:FB:2D:C7:D2:CC:FE:4A:E3:0E:A9:2A:1C:27:4D:D2:A6:C5:87:B8:97:98:57:75:10:15:28:EF:A2:23
:7C
```

FortiClient troubleshooting

Certificate not trusted

When configuring a new connection to an EMS server, the certificate might not be trusted.



When you click *Authorize*, a warning displays: *The server certificate cannot be authenticated with installed CA certificates. Please install its CA certificates on this FortiGate*.

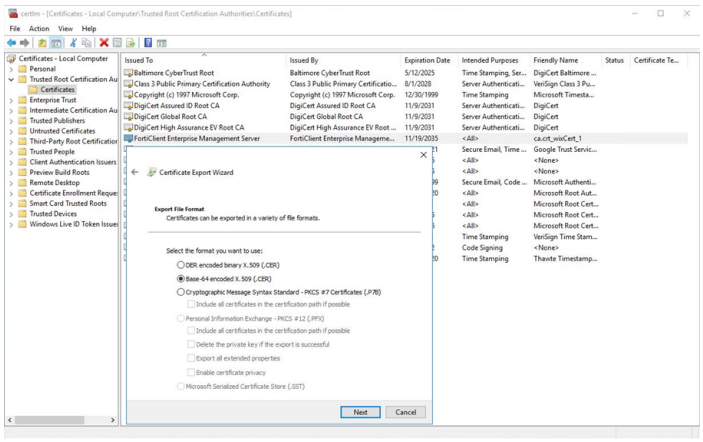
In the CLI, an error message displays when you try to verify the certificate:

```
# execute fctems verify Win2K16-EMS
certificate not configured/verified: 2
Could not verify server certificate based on current certificate authorities.
Error 1--92-60-0 in get SN call: EMS Certificate is not signed by a known CA.
```

The default FortiClient EMS certificate that is used for the SDN connection is signed by the CA certificate that is saved on the Windows server when FortiClient EMS is first installed. You can manually export and install it on the FortiGate.

To manually export and install the certificate on to the FortiGate:

1. Export the EMS certificate on the server that EMS is installed on:
   a. On the Windows server that EMS is installed on, go to *Settings > Manage computer certificates*.
   b. In the certificate management module, go to *Trusted Root Certification Authorities > Certificates*.
   c. Right click on the certificate issued by FortiClient Enterprise Management Server and select *All Tasks > Export*.
   d. The *Certificate Export Wizard* opens. Click *Next*.
   e. Select *Base-64 encoded X.509*, then click *Next*.



f. Enter a file name for the certificate and click *Browse* to select the folder where it will be located, then click *Next*.

g. Review the settings, then click *Finish*. The certificate is downloaded to the specified folder.

2. On the FortiGate, import the certificate:

a. Go to *System* > *Certificate*. By default, the *Certificate* option is not visible, see Feature visibility on page 3952 for information.

b. Click *Import* > *CA Certificate*.

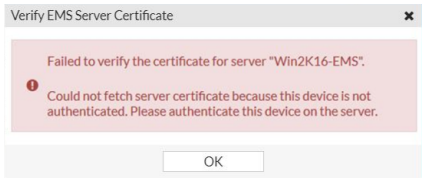
c. Set *Type* to *File*, and click *Upload* to import the certificate from the management computer.

d. Click *OK*. The imported certificate is shown in the *Remote CA Certificate* section of the certificate table.

3. Try to authorize the certificate on the FortiGate:

a. Go to *Security Fabric* > *Fabric Connectors* and edit the FortiClient EMS connector. The connection status should now say that the certificate is not authorized.

b. Click *Authorize*. The following warning is shown:



The warning can also be seen in the CLI:

```
# execute fctems verify Win2K16-EMS
failure in certificate configuration/verification: -4
Could not verify EMS. Error 1--94-0-401 in get SN call: Authentication denied.
```

4. Authorize the FortiGate on EMS:

a. Log in to the EMS server console and go to *Administration* > *Fabric Devices*.

b. Select the serial number of the FortiGate device, then click *Authorize*.

5. Try to authorize the certificate on the FortiGate again:

a. On the FortiGate, go to *Security Fabric* > *Fabric Connectors* and edit the *FortiClient EMS* card.

b. Click *Refresh*.

c. When presented with the EMS server certificate, click *Accept* to accept the certificate.
Your connection should now be successful and authorized.

d. Click *OK*.

Synchronizing FortiClient ZTNA tags

ZTNA tags (formerly FortiClient EMS tags in FortiOS 6.4 and earlier) are tags synchronized from FortiClient EMS as dynamic address objects on the FortiGate. FortiClient EMS uses zero-trust tagging rules to automatically tag managed endpoints based on various attributes detected by the FortiClient. When the FortiGate establishes a connection with the FortiClient EMS server through the EMS Fabric connector, it pulls zero-trust tags containing device IP and MAC addresses and converts them to read-only dynamic address objects. It also establishes a persistent WebSocket connection to monitor for changes in zero-trust tags, which keeps the device information current. These ZTNA tags can then be used in ZTNA rules, firewall rules, and NAC policies to perform security posture checks. ZTNA tags are displayed in the *Device Inventory* widget, *FortiClient* widget, and *Asset Identity Center* page.

By enabling ZTNA EMS tag checking in a firewall policy, you can include EMS tag information in the traffic log.

When primary and secondary ZTNA EMS tag checking is enabled using address groups, the *Primary EMS tag* and *Secondary EMS tag* fields will be included in the GUI traffic logs. Likewise, the `emstag` and `emstag2` fields will be included in the CLI traffic logs.

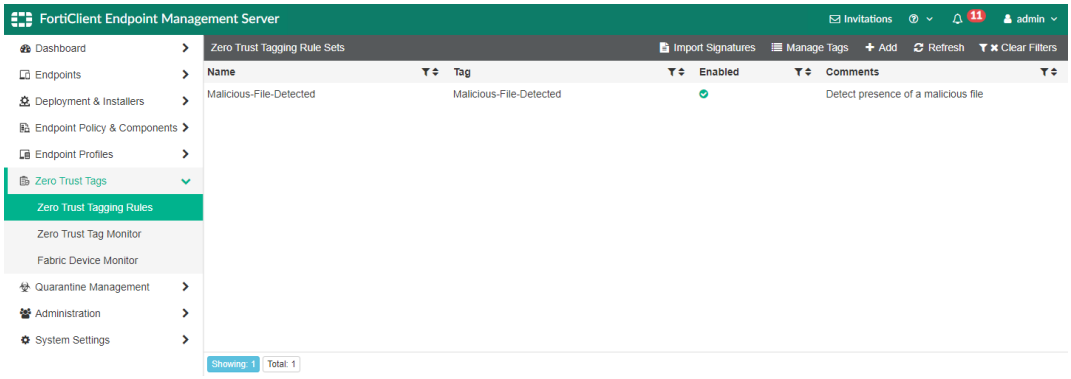
When using WebSocket, EMS pushes notifications to the corresponding FortiGate when there are updates to tags or other monitored attributes. The FortiGate then fetches the updated information using the REST API over TCP/8013. When WebSocket is not used (due to an override or unsupported EMS version), updates are triggered on demand from the FortiGate side over the REST API.

If the WebSocket capability is detected, the capabilities setting will automatically display the WebSocket option. You can use the `diagnose test application fcnacd 2` command to view the status of the WebSocket connection.

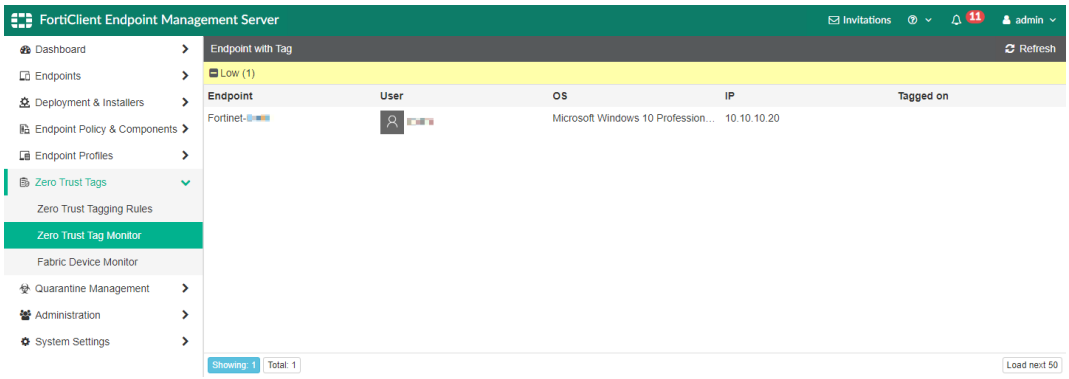
In the following example, the FortiGate connects to and retrieves ZTNA tags from a FortiClient EMS configured with tagging rules. It is assumed that zero-trust tags and rules are already created on the FortiClient EMS. For more information, see the Zero Trust Tags section of the EMS Administration Guide.

To verify zero-trust tags in FortiClient EMS:

1. Go to *Zero Trust Tags* > *Zero Trust Tagging Rules* to view the tags.

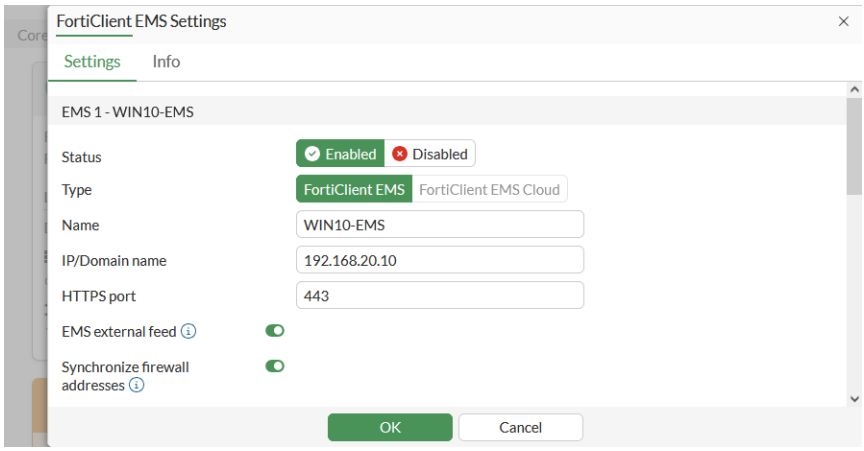


2. Go to *Zero Trust Tags* > *Zero Trust Tag Monitor* to view the registered users who match the defined tag.

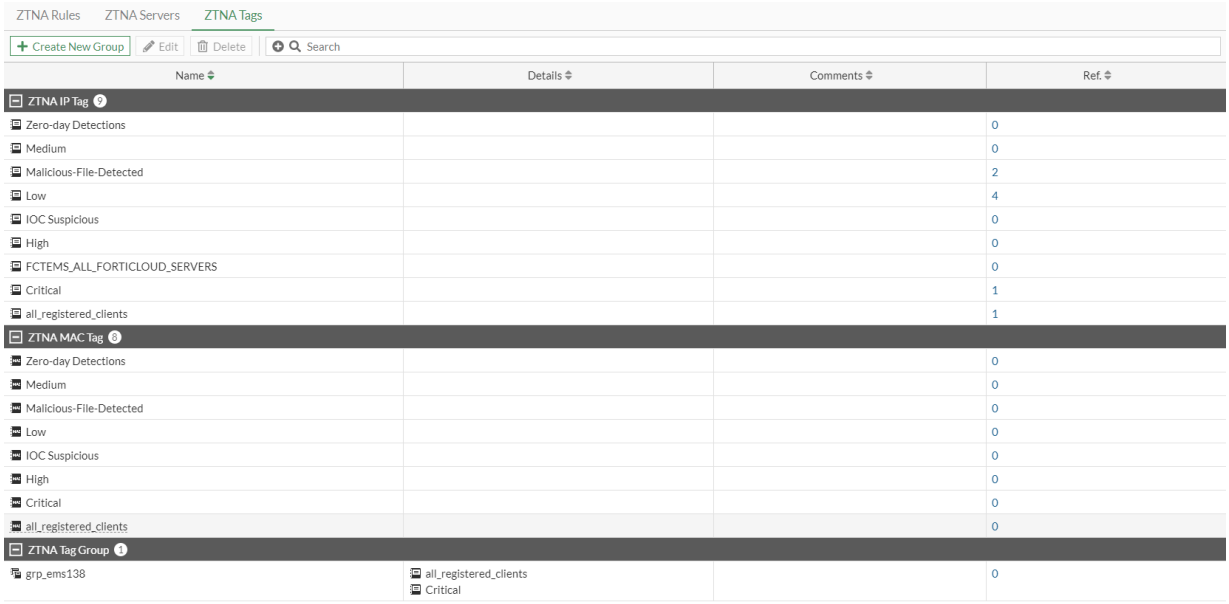


To configure the EMS Fabric connector to synchronize ZTNA tags in the GUI:

1. Configure the EMS Fabric connector:
   a. On the root FortiGate, go to *Security Fabric > Fabric Connectors* and double-click the *FortiClient EMS* card.
   b. Set the *Status* to *Enabled* for one of the EMS servers.
   c. Enable *Synchronize firewall addresses*.



   d. Configure the other settings as needed and validate the certificate.
   e. Click *OK*.
2. Enable ZTNA:
   a. Go to *System > Feature Visibility* and enable *Zero Trust Network Access*.
   b. Click *Apply*.
3. Go to *Policy & Objects > ZTNA* and select the *ZTNA Tags* tab. You will see the ZTNA IP and ZTNA MAC tags synchronized from the FortiClient EMS.



To configure the EMS Fabric connector to synchronize ZTNA tags in the CLI:

1. Configure the EMS Fabric connector on the root FortiGate:

```
config endpoint-control fctems
    edit 1
        set status enable
        set name "WIN10-EMS"
        set server "192.168.20.10"
        set https-port 443
        set pull-sysinfo enable
        set pull-vulnerabilities enable
        set pull-tags enable
        set pull-malware-hash enable
        set capabilities fabric-auth silent-approval websocket
    next
end
```

2. Verify which IPs the dynamic firewall address resolves to:

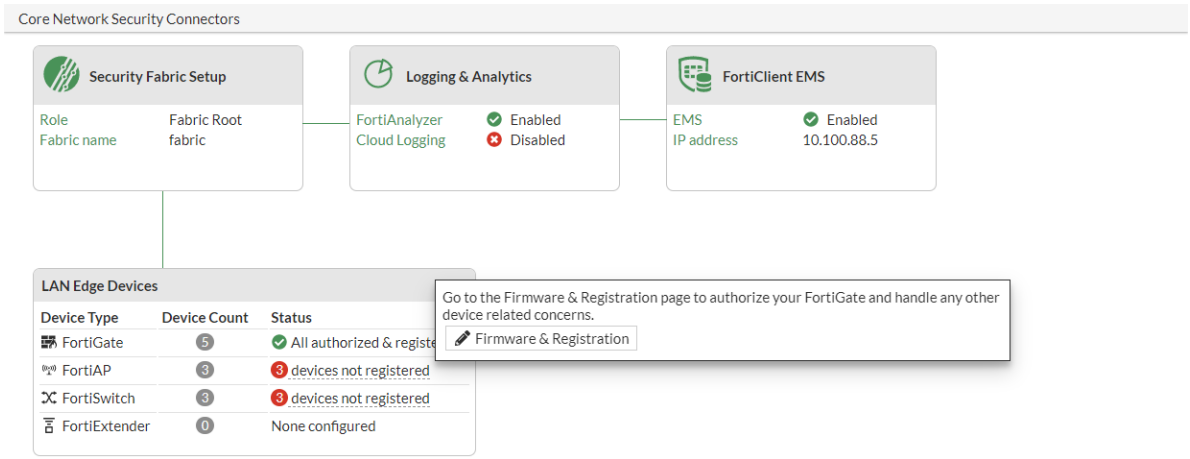
```
# diagnose firewall dynamic list
List all dynamic addresses:
FCTEMS0000100000_all_registered_clients: ID(51)
        ADDR(172.17.194.209)
        ADDR(10.10.10.20)
…

FCTEMS0000100000_Low: ID(78)
        ADDR(172.17.194.209)
        ADDR(10.10.10.20)
…

FCTEMS0000100000_Malicious-File-Detected: ID(190)
```

```
        ADDR(172.17.194.209)
        ADDR(10.10.10.20)
…
```

When running the FortiGate in multi-VDOM mode, by default, EMS is configured in the global VDOM. All ZTNA tags synchronized with the globally configured EMS are shared by all VDOMs. FortiOS 7.4 and later supports configuring EMS on a per-VDOM basis. See Configuring FortiClient EMS and FortiClient EMS Cloud on a per-VDOM basis for more information.

Configuring LAN edge devices

The *LAN Edge Devices* connector card displays a summary about the FortiGates, FortiAPs, FortiSwitches, and FortiExtenders in the Security Fabric. Information about the device type, number of devices, and number of unregistered and unauthorized devices is displayed. If there are devices that do not have a green checkmark in the *Status* column, hover over the status message to view the tooltip with required action. In this example, there are FortiAPs and FortiSwitches that are not registered. The tooltip includes a link to the *System > Firmware & Registration* page to register or authorize the devices. The tooltip for FortiExtenders includes a link to the *Network > FortiExtenders > Managed FortiExtenders* page to register or authorize the devices.



If the default `auto-auth-extension-device` settings on the FortiAP or FortiSwitch have been modified, manual authorization in the Security Fabric may not be required.

For more information about configuring FortiAPs, see Configuring the FortiGate interface to manage FortiAP units and Discovering, authorizing, and deauthorizing FortiAP units.

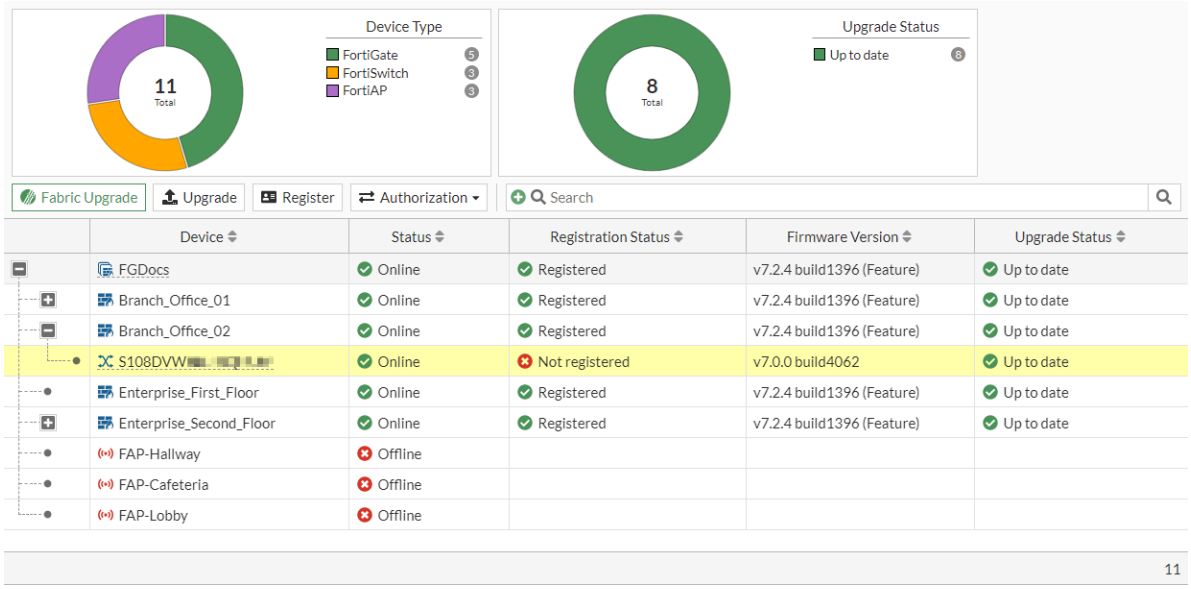
For more information about configuring FortiSwitches, see Discovering, authorizing, and deauthorizing FortiSwitch units.

For more information about configuring FortiExtenders, see Adding a FortiExtender on page 767 and FortiExtender and FortiGate integration.

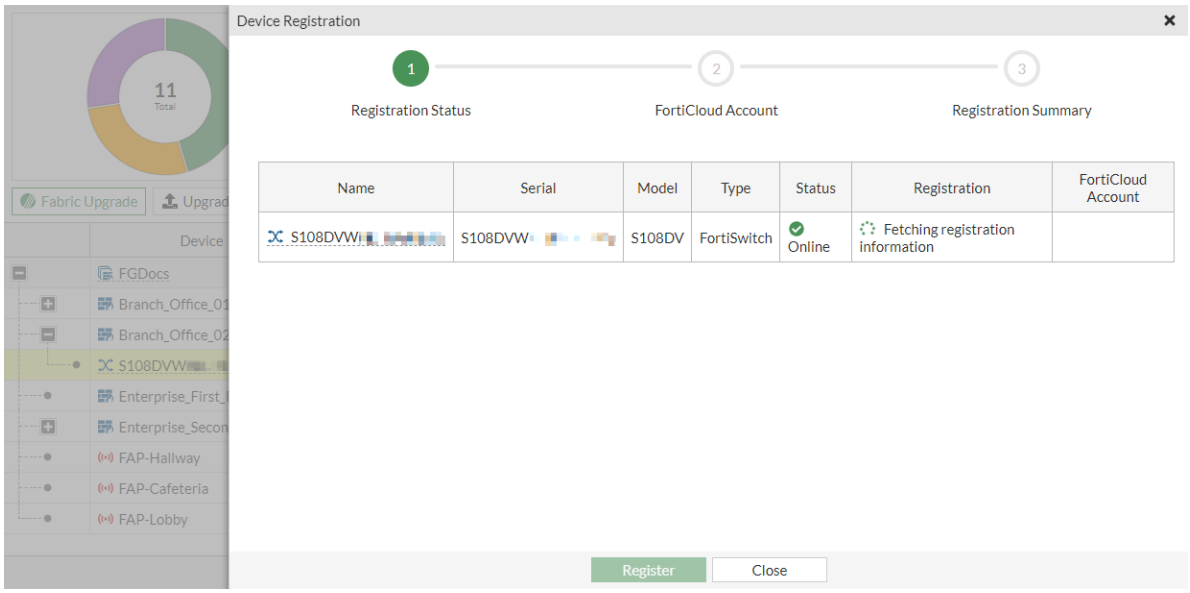
The following example shows how to register a FortiSwitch that has been authorized to join the Security Fabric. The procedure is similar for FortiAPs and FortiExtenders. If the device requires authorization, refer to Authorizing devices on page 3773.

To register a FortiSwitch:

1. Connect the device to the root FortiGate.
2. On the root FortiGate, go to *System* > *Firmware & Registration*. (Use the same page for a FortiAP, or *Network* > *FortiExtenders* > *Managed FortiExtenders* for a FortiExtender.)
3. Select the unregistered device and click *Register*.



The *Device Registration* pane opens and the FortiGate verifies the registration information.



4. Once the registration process is complete, click *Close*.

Configuring central management

The *Central Management* Fabric connector card on the root FortiGate is used to configure the FortiManager settings, which includes on-premises FortiManager, FortiGate Cloud, and FortiManager Cloud. After the *Central Management* connector is configured, it automatically synchronizes with any connected downstream devices.

This topic covers the following central management aspects:

- Configuring FortiManager
- Configuring FortiManager Cloud

Configuring FortiManager

Once the *Central Management* Fabric connector is configured, the root FortiGate pushes this configuration to downstream FortiGates. FortiManager provides remote management of FortiGate devices over TCP port 541. The FortiManager must have internet access for it to join the Security Fabric.

Once configured, the FortiGate can receive antivirus and IPS updates, and allows remote management through FortiManager or the FortiGate Cloud service. The FortiGate management option must be enabled so that the FortiGate can accept management updates to its firmware and FortiGuard services.

Adding a FortiManager device to the Security Fabric requires the following steps in FortiOS, which can be completed in the GUI or CLI:

- Specify the FortiManager IP address or domain name.
- Approve the FortiManager serial number returned by the FortiManager server certificate. This ensures that the administrator is connecting the FortiGate to the desired FortiManager.

After completing the steps in FortiOS, go to FortiManager to complete the process by authorizing the FortiGate.

To enable management of FortiGate VM devices in FortiManager:

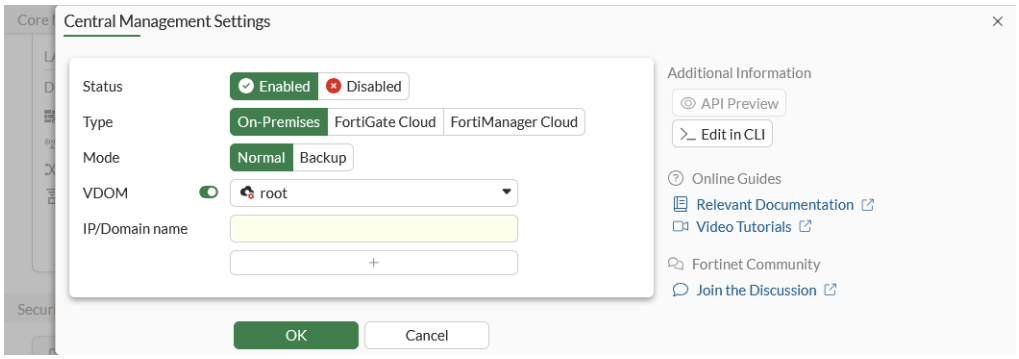
If you are using a FortiGate VM, enable recognition of FortiGate VM serial numbers in FortiManager. For FortiGate hardware models, skip this step. See Enabling management of VM devices in the FortiManager Administration Guide for more information.

In FortiManager, run the following command:

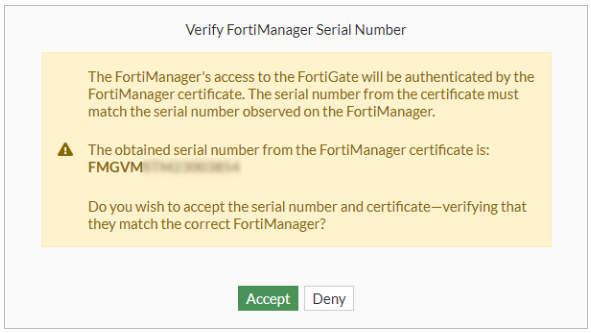
```
config sys global
    set fgfm-allow-vm enable
end
```

To add a FortiManager to the Security Fabric in the GUI:

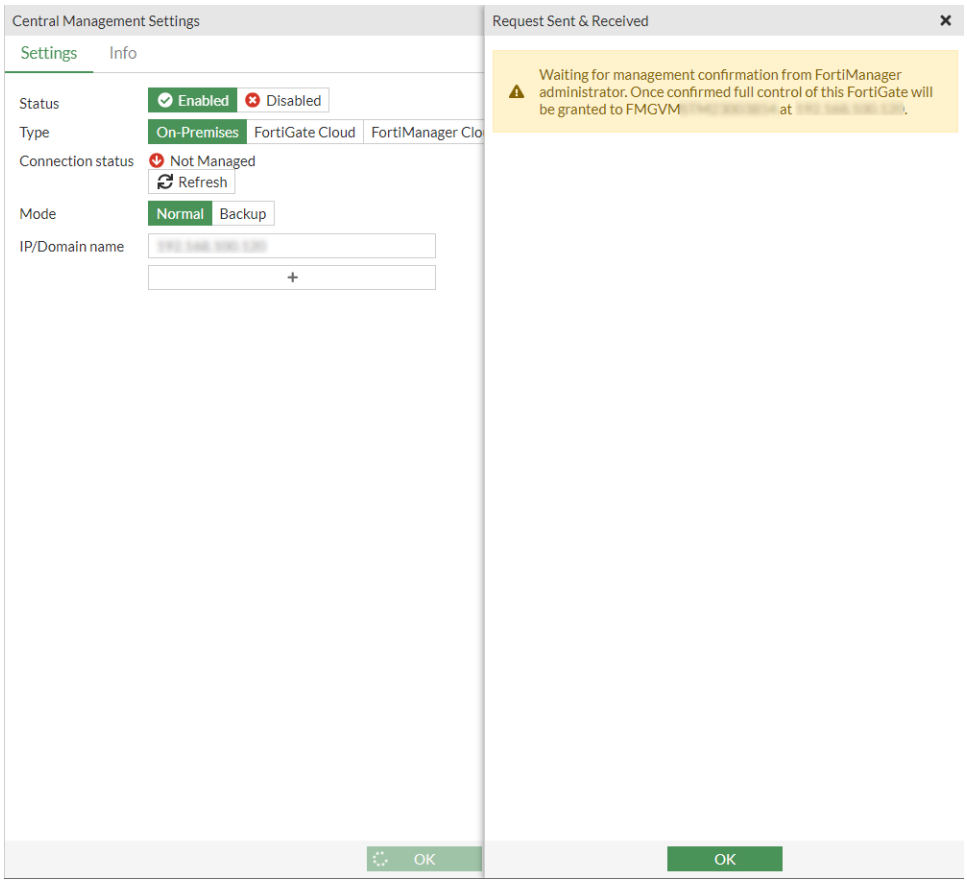
1. On the root FortiGate, go to *Security Fabric > Fabric Connectors* and double-click the *Central Management* card.
2. Set the *Status* to *Enabled*.
3. Set the *Type* to *On-Premises*.



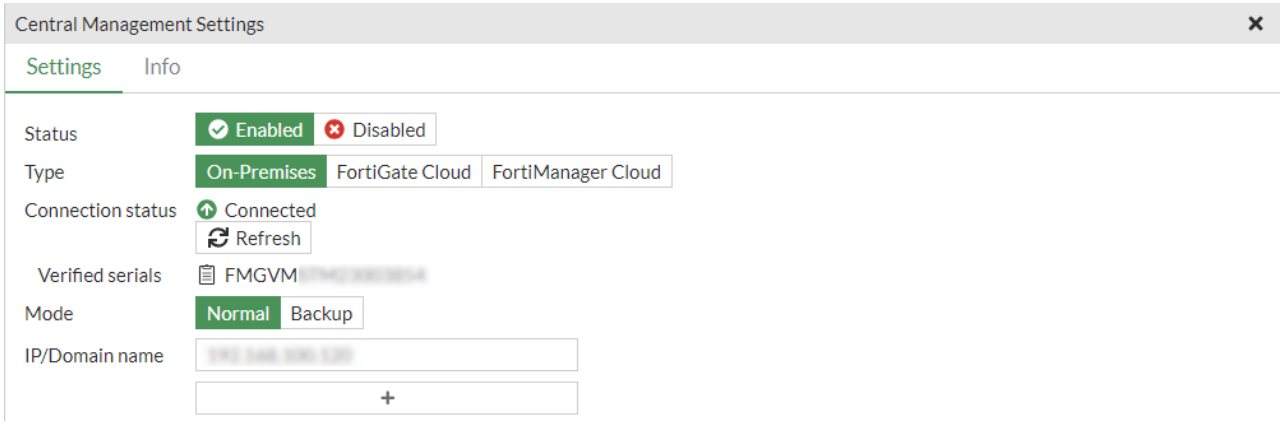
4. If VDOMs are enabled, enable *VDOM* and select the VDOM that has a route to the FortiManager.
5. Enter the *IP/Domain Name* of the FortiManager.
6. Click *OK*.
   The *Verify FortiManager Serial Number* pane appears.



7. Review the serial number, and click *Accept*.
   The *Request Sent & Received* pane appears, indicating the FortiGate must be authorized on FortiManager.



8. Go to FortiManager and authorize the FortiGate. See Authorizing the FortiGate in FortiManager on page 4353.
9. After the FortiGate is registered, log in to FortiGate again as either read-only or read/write.
10. Go to *Security Fabric > Fabric Connectors* and double-click the *Central Management* card. The *Connection Status* is updated to *Connected*.



To add a FortiManager to the Security Fabric in the CLI:

1. Enter the FortiManager connection information:

```
config system central-management
    set type fortimanager
    set fmg {<IP_address> | <Domain name>}
end

The Serial Number for FortiManager is not entered.
In order to verify identity of FortiManager serial number is needed.
If serial number is not set, connection will be set as unverified.
FortiGate can establish a connection to obtain the serial number now.Do you want to try to
connect now? (y/n)y
Obtained serial number from FortiManager 172.16.200.1 is: FMGVMSTM2300xxxx
Do you confirm that this is the correct serial number? (y/n)y

Successfully registered to FortiManager. This device may need to be authorized on
FortiManager.
Auto firmware upgrade in system.fortiguard has been paused since this FortiGate is now managed
by FortiManager. The upgrade will resume automatically when this FortiGate is released from
FortiManager. The upgrade status may be viewed using the following command diagnose test
application forticldd 13.
Any pending automatic patch-level firmware upgrade has been removed
```

2. Go to FortiManager and authorize the FortiGate. See Authorizing the FortiGate in FortiManager on page 4353.
3. If necessary on FortiGate, use the `diagnose fdsm central-mgmt-status` command to diagnose the connection.
   - If the connection is not yet successful because the FortiManager serial number is not verified, the following information is displayed:

```
# diagnose fdsm central-mgmt-status
Connection status: Handshake
Registration status: Unknown
Serial: FMGVMSTM2300xxxx
```

   - If the connection is up, but the FortiGate has not been authorized by FortiManager, the following information is displayed:

```
# diagnose fdsm central-mgmt-status
Connection status: Up
Registration status: Unregistered
Serial: FMGVMSTM2300xxxx
```

   - If the connection is up, and the FortiGate has been authorized, the following information is displayed:

```
# diagnose fdsm central-mgmt-status
Connection status: Up
Registration status: Registered
Serial: FMGVMSTM2300xxxx
```

Authorizing the FortiGate in FortiManager

After completing the GUI or CLI steps in FortiOS, go to FortiManager to authorize the FortiGate to complete the process.

To authorize the FortiGate in FortiManager:

1. On FortiManager, go to *Device Manager* and find the FortiGate in the *Unauthorized Devices* list.
   The unauthorized device list is located in the root ADOM, regardless of the firmware version of the root ADOM or FortiOS.
2. Select the FortiGate device or devices, and click *Authorize* in the toolbar.
3. In the *Authorize Device* pop-up, adjust the device names as needed, select the appropriate ADOM (if applicable), and click *OK*.

For more information about using FortiManager, see the FortiManager Administration Guide.

Configuring FortiManager Cloud

This cloud-based SaaS management service is available through FortiManager. This service is included in FortiCloud accounts with a FortiManager Cloud account level subscription (ALCI).

Configuring a per-device license

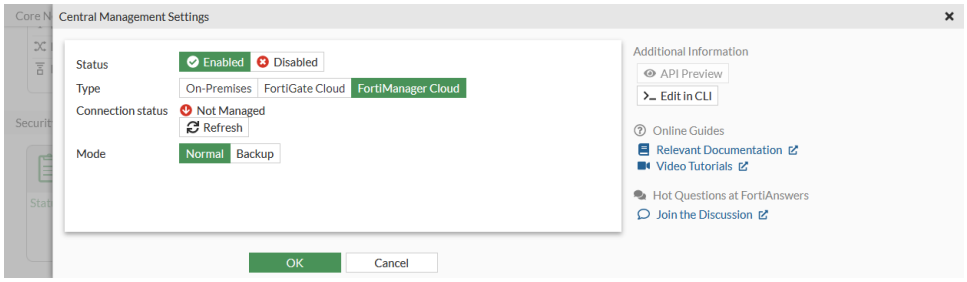
Once the FortiGate has acquired a contract named *FortiManager Cloud*, FortiCloud creates a cloud-based FortiManager instance under the user account. You can launch the portal for the cloud-based FortiManager from FortiCloud, and its URL starts with the User ID.

You can use a FortiGate with a contract for *FortiManager Cloud* to configure central management by using the FQDN of *fortimanager.forticloud.com*. A FortiGate-FortiManager tunnel is established between FortiGate and the FortiManager instance.

After the tunnel is established, you can execute FortiManager functions from the cloud-based FortiManager portal.

To configure FortiManager Cloud central management:

1. Enable FortiManager Cloud.
   a. Go to *Security Fabric* > *Fabric Connectors* and double-click the *Central Management* card.
   b. Set the *Status* to *Enabled*.
   c. Set the *Type* to *FortiManager Cloud*.



d. Click *OK*.

> The *FortiManager Cloud* button can only be selected if you have a FortiManager Cloud product entitlement.

2. In the FortiManager Cloud instance, go to *Device Manager* and authorize the FortiGate. See Authorizing devices for more information.

   When using the FortiGate to enable FortiManager Cloud, the FortiGate appears as an unauthorized device. After authorizing the FortiGate, it becomes a managed device.

   In FortiOS, the *Security Fabric > Fabric Connectors* page now displays green arrow in the *Central Management* card because FortiManager Cloud is registered.

Diagnostics

To verify the contract information:

```
# diagnose test update info contract
...
System contracts:
    ...
Account contracts:
    FMGC,Thu Dec  2 16:00:00 2022
...
```

To verify the FortiManager Cloud instance has launched and the FortiGate is registered:

```
# diagnose fdsm central-mgmt-status
Connection status: Up
Registration status: Registered
Serial: FMGVMSTM2300xxxx
```

Configuring sandboxing

The Security Fabric supports the following types of FortiSandbox deployments:

Type	Description	Requirements	Next steps
FortiGate Cloud Sandbox (FortiSandbox SaaS)	Files are sent to Fortinet's Cloud Sandbox cluster for post-processing.	• The FortiGate must be subscribed to the Advanced Malware Protection (AMP) license, which includes the AV license.	• Activate and configure your FortiGate Cloud Sandbox. • Use post-transfer scan with Antivirus.

Type	Description	Requirements	Next steps
FortiGuard Inline Malware Prevention System	Files are sent to Fortinet's Cloud Sandbox cluster for real-time processing.	• The FortiGate must either be subscribed to the Enterprise Protection bundle or have an a la carte Inline Malware Prevention license.	• Verify your license. • Configure your FortiGate. • Use Inline Scan with Antivirus. • Verify Sandbox detection.
FortiSandbox Cloud (FortiSandbox PaaS)	Files are sent to a dedicated FortiCloud hosted instance of FortiSandbox for processing.	• FortiCloud premium license. • FortiSandbox Cloud entitlement. • The FortiGate and FortiCloud licenses are registered to the same account.	• Provision your dedicated instance. • Configure your FortiGate for post-transfer scan or inline scan. • Verify Sandbox detection.
FortiSandbox Appliance	Files are sent to a physical or VM appliance, typically residing on premise, for processing.	• None	• Configure your FortiSandbox. • Configure your FortiGate for post-transfer scan or inline scan. • Verify Sandbox detection.

To apply sandboxing in a Security Fabric, connect one of the FortiSandbox deployments, then configure an antivirus profile to submit files for dynamic analysis. The submission results supplement the AV signatures on the FortiGate. FortiSandbox inspection can also be used in web filter profiles.

In a Security Fabric environment, sandbox settings are configured on the root FortiGate. Once configured, the root FortiGate pushes the settings to other FortiGates in the Security Fabric.

FortiGate Cloud Sandbox (FortiSandbox SaaS)

FortiGate Cloud Sandbox allows users to take advantage of FortiSandbox features without having to purchase, operate, and maintain a physical appliance. It also allows you to control the region where your traffic is sent to for analysis. This allows you to meet your country's compliance needs regarding data storage locations.

Users are not required to have a FortiCloud account to use FortiGate Sandbox Cloud.

The submission to the cloud with a valid FortiGuard Antivirus (AVDB) license is rate limited per FortiGate model. Refer to the Service Description for details. For those without an AVDB license, the submission is limited to only 100 per day.

To configure FortiGate Cloud Sandbox, you must first activate the connection. Note that FortiGate Cloud Sandbox is decoupled from FortiGate Cloud logging, so you do not need to have a FortiCloud account or have cloud logging enabled.

To ensure proper connectivity to FortiGate Cloud Sandbox, on the FortiGate in *Security Profiles > AntiVirus*, create a profile with *Send files to FortiSandbox for inspection* configured, and create a firewall policy with logging enabled that uses the Sandbox-enabled AV profile.

To enable FortiGate Cloud Sandbox in the GUI:

1. Go to *Security Fabric > Fabric Connectors* and double-click the *Sandbox* card.
2. Set the *Status* to *Enabled*.
3. Set the *Type* to *FortiGate Cloud*.

   If the *FortiSandbox Cloud* option is grayed out or not visible, enter the following in the CLI:

   ```
   config system global
       set gui-fortigate-cloud-sandbox enable
   end
   ```

4. Select a *Region* from the dropdown.
5. Click *OK*.

To activate the FortiGate Cloud Sandbox connection:

```
# execute forticloud-sandbox region
0  Europe
1  Global
2  Japan
3  US
Please select cloud sandbox region[0-3]:3
```

After a region is selected, the following configuration is added:

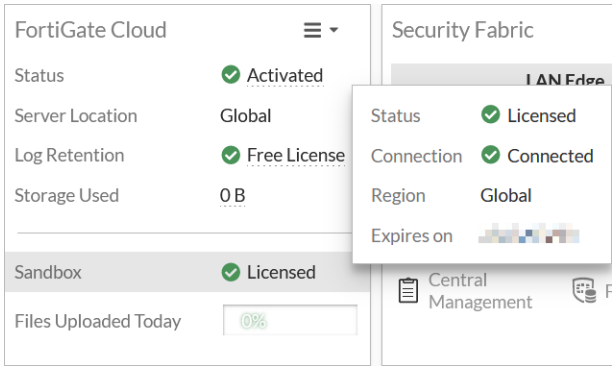
```
config system fortiguard
    set sandbox-region <region>
end
```

Alternatively, using the `execute forticloud-sandbox update` command also works.

To obtain or renew a FortiGuard antivirus license:

1. See the How to Purchase or Renew FortiGuard Services video for FortiGuard antivirus license purchase instructions.
2. Once a FortiGuard license is purchased and activated, users are provided with a paid FortiSandbox Cloud license.

a. Go to *Dashboard* > *Status* to view the FortiSandbox Cloud license indicator.



b. Alternatively, go to *System* > *FortiGuard* to view the FortiSandbox Cloud license indicator.

FortiSandbox Cloud (FortiSandbox PaaS)

FortiSandbox Cloud offers more features and better detection capability. Connecting to FortiSandbox Cloud will automatically use the cloud user ID of the FortiGate to connect to the dedicated FortiSandbox Cloud instance. The FortiGate automatically detects if there is a valid entitlement.

The following items are required to initialize FortiSandbox Cloud:

- A FortiCloud premium account.
- A valid FortiSandbox Cloud contract on the FortiGate. To view contract information in the CLI, enter `diagnose test update info`. The `User ID` at the end of the output shows FortiCloud which FortiSandbox Cloud account the FortiGate is connected to.
- A provisioned FortiSandbox Cloud. See Deploying FortiSandbox Cloud for information.

To configure FortiSandbox Cloud in the GUI:

1. Go to *Security Fabric* > *Fabric Connectors* and double-click the *Sandbox* card.
2. Set the *Status* to *Enabled*.
3. Set the *Type* to *FortiSandbox Cloud*.

   If the *FortiSandbox Cloud* option is grayed out or not visible, enter the following in the CLI:

   ```
   config system global
       set gui-fortigate-cloud-sandbox enable
   end
   ```

4. Click *OK*.

To configure FortiSandbox Cloud in the CLI:

```
config system fortisandbox
    set status enable
    set forticloud enable
```

```
    set server fortisandboxcloud.com
end
```

If the FortiGate does not detect the proper entitlement, a warning is displayed and the CLI configuration will not save.

If the FortiSandbox Cloud is running version 4.0.0 and later, the FortiGate will automatically connect to fortisandboxcloud.com, and then discover the specific region and server to connect to based on which region the customer selected to deploy their FortiSandbox Cloud instance. The FortiGate must have a FortiCloud premium account license and a FortiSandbox Cloud VM license for this functionality.

FortiSandbox appliance

FortiSandbox appliance is the on-premise option for a full featured FortiSandbox. Connecting to a FortiSandbox appliance requires that Cloud Sandbox is disabled.

FortiOS can authenticate FortiSandbox using Fortinet CA certificates. Third-party certificates are not supported.

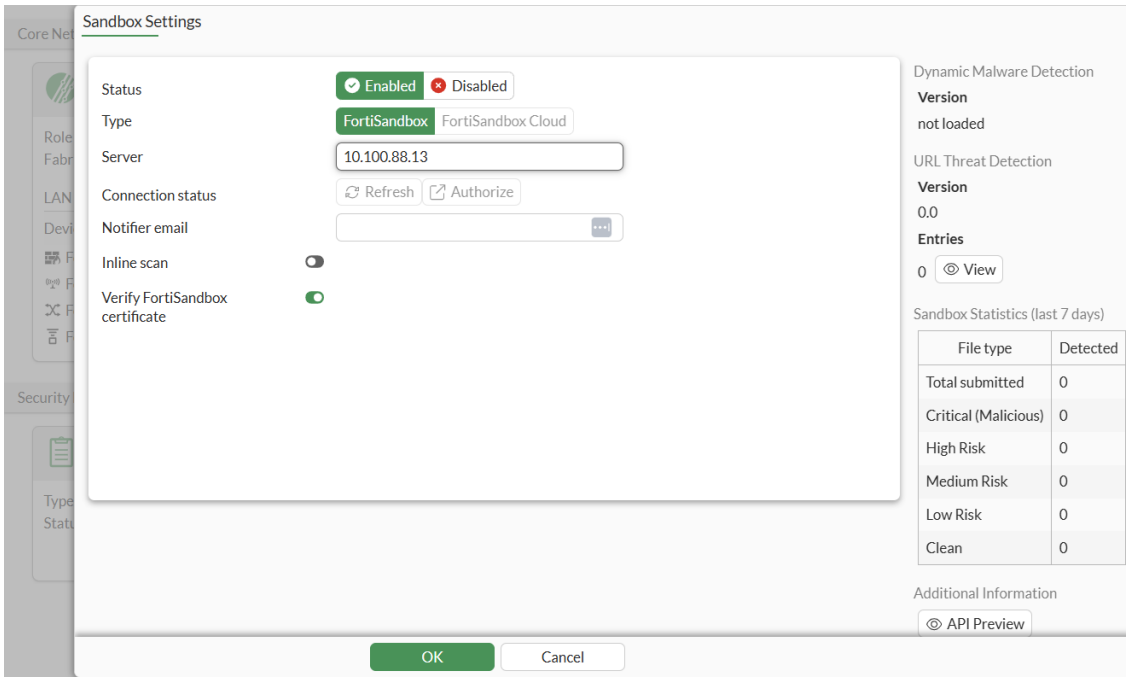
By default, FortiOS uses an automatic certificate verification process to authenticate FortiSandbox. In the fabric connector, enable FortiSandbox and enter its IP address. FortiOS connects to FortiSandbox to retrieve and display its serial number for verification. Once verified, FortiOS adds the serial number to the `cn` (Common Name) field.

You can also disable automatic certificate verification, and use the CLI to manually configure CA (Certificate Authority) and CN fields for FortiSandbox.

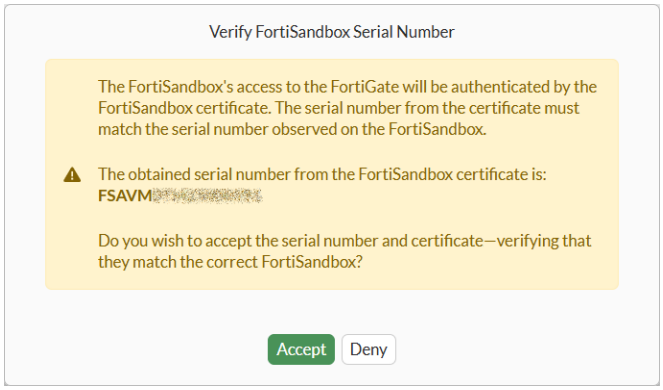
FortiOS uses the following methods with FortiSandbox: post transfer and inline. Certificate checks and verification have only been implemented for the post-transfer method.

To enable FortiSandbox appliance in the GUI:

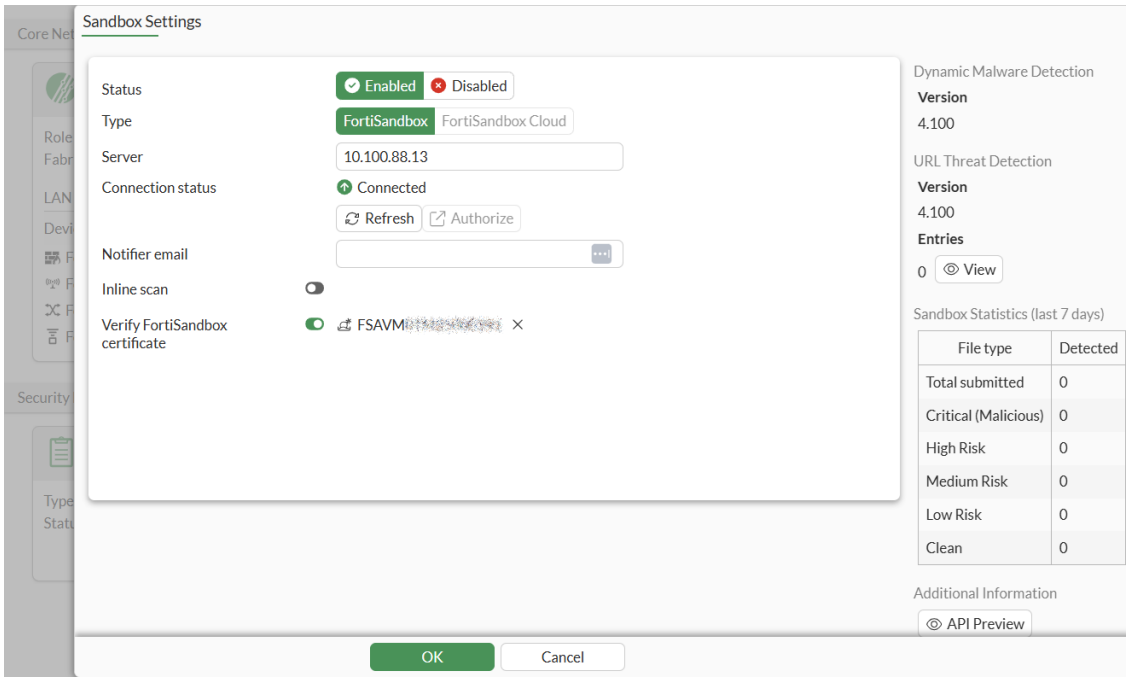
1. Go to *Security Fabric > Fabric Connectors* and double-click the *Sandbox* card.
2. Set the *Status* to *Enabled*.
3. In the *Server* field, enter the FortiSandbox device's IP address.



4. Optionally, enter a *Notifier email*.
5. Click *OK*. A verification message is displayed.



6. Click *Accept* to verify the correct serial number.
   The serial number is displayed beside *Verify FortiSandbox certificate*, and the *Connection status* is *Connected*.



To enable FortiSandbox appliance in the CLI:

1. Enable FortiSandbox and configure its IP address:

```
config system fortisandbox
    set status enable
    set forticloud disable
    set server "172.18.70.76"
end
```

2. After the message is displayed, type y to establish a connection and retrieve the FortiSandbox serial number:

```
In order to verify identity of FortiSandbox serial number is needed.
If serial number is not set, connection will be set as unverified and
access to local config and files will be accessible only with user name/password.
FortiGate can establish a connection to obtain the serial number now. Do you want to try to
connect now? (y/n)y
```

3. After FortiOS obtains the FortiSandbox serial number, type y to verify it:

```
Obtained serial number from X509 certificate of FortiSandbox is: FSA3KET321000049
Serial number from certificate MUST be the same as serial number observed in FortiSandbox.
If these two serial numbers don't match, connection will be dropped.
Please make sure the serial numbers are matching.
Do you confirm that this is the correct serial number? (y/n)y
```

   FortiOS adds the serial number to the cn field.

4. Show the FortiSandbox settings to verify the CN field:

```
show system fortisandbox
config system fortisandbox
    set status enable
    set server "172.18.70.76"
    set cn "FSA3KET321000049"
end
```

5. Check the FortiSandbox connection status:

```
# execute system fortisandbox test-connectivity
Reachable.
```

To configure CN and CA fields in the CLI:

1. Configure the fabric connector with CN and CA fields:

```
config system fortisandbox
    set status enable
    set server "172.18.70.76"
    set ca "Fortinet_CA_Backup"
    set cn "FSA3KET321000049"
    set certificate-verification disable
end
```

2. Check the connection status:

```
# execute system fortisandbox test-connectivity

Reachable
```

Authorizing the FortiGate from FortiSandbox Cloud and a FortiSandbox appliance

Once the FortiGate makes a connection to the FortiSandbox Cloud or appliance, the FortiGate must be authorized. See FortiGate devices in the FortiSandbox Administration guide for information.

Antivirus profiles

An antivirus profile must be configured to send files to the sandbox. Once submitted, sandbox inspection is performed on the file to detect malicious activities. The FortiGate can use the dynamic malware detection database from the sandbox to supplement the AV signature database. See Using FortiSandbox post-transfer scanning with antivirus on page 2267 for more information.

FortiSandbox inline scanning is supported on FortiSandbox appliances in proxy and flow mode. When inline scanning is enabled, the client's file is held while it is sent to FortiSandbox for inspection. Once a verdict is returned, the appropriate action is performed on the held file. If there is an error or timeout on the FortiSandbox, the FortiGate's configuration determines what to do with the held file. See Using FortiSandbox inline scanning with antivirus on page 2269 for more information.

Inline scanning requires FortiSandbox version 4.2 or later.

Web filter profiles

Sandbox inspection can be used in web filter profiles. The FortiGate uses URL threat detection database from the sandbox to block malicious URLs. See Block malicious URLs discovered by FortiSandbox on page 2335 for more information.

FortiSandbox Files FortiView monitor

In the *FortiSandbox Files* FortiView monitor, users can select a submitted file and drill down to view its static and dynamic file analysis. The full FortiSandbox report can be downloaded in PDF format. This feature works with FortiGate Cloud Sandbox, FortiSandbox Cloud, and FortiSandbox appliance. FortiSandbox must be running version 3.2.1 and later.

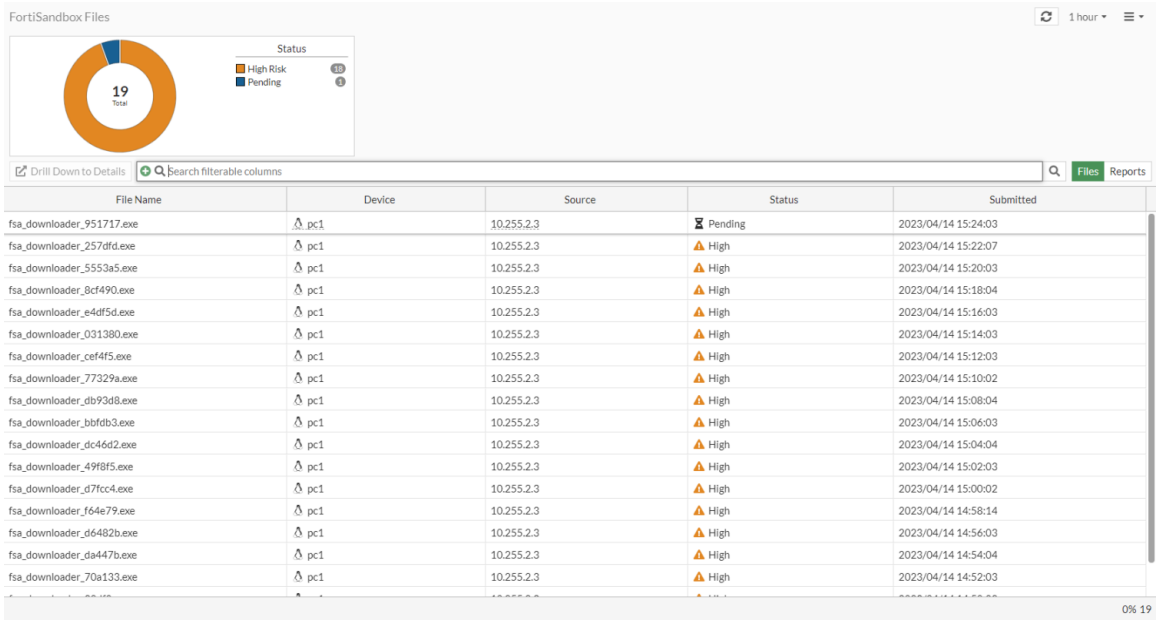
PDF reports are downloaded on-demand. By default, only 10 are kept in memory. PDFs are deleted from memory after 24 hours.

Prerequisites:

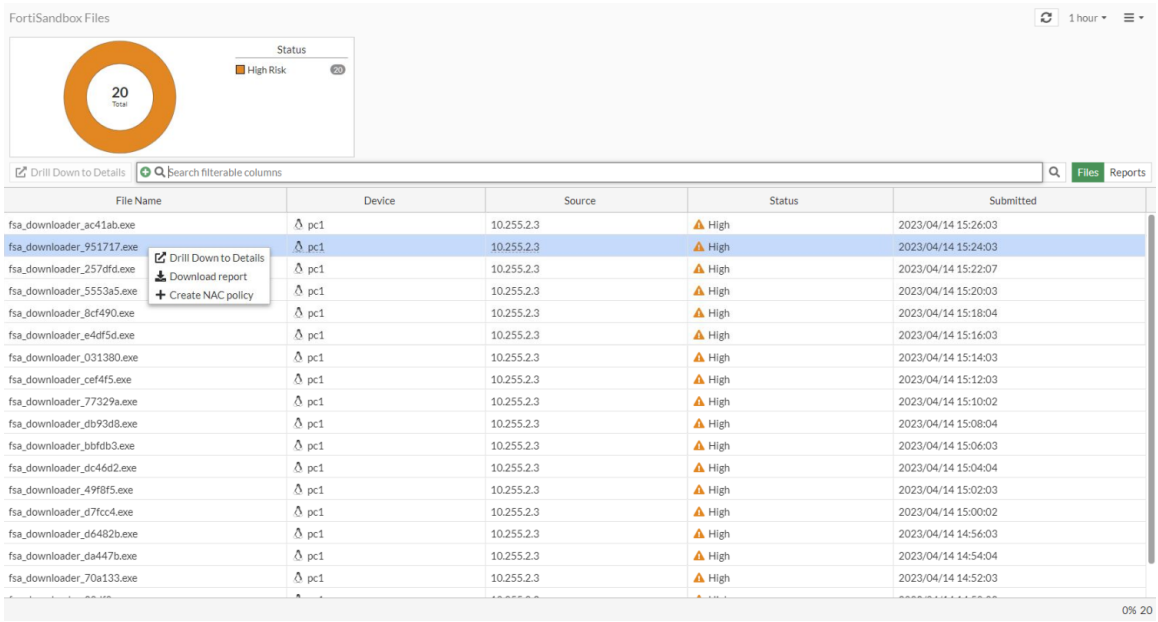
1. Add FortiSandbox to the Security Fabric.
2. Configure an AV profile with *Send files to FortiSandbox for inspection* enabled (see Using FortiSandbox post-transfer scanning with antivirus on page 2267).
3. Configure a firewall policy with the AV profile that allows traffic to the internet.
4. Add the *FortiSandbox Files* FortiView monitor (see Adding dashboards of widgets and monitors on page 192).
5. On a client PC, attempt to download a suspicious file.

To view the FortiSandbox analysis and download the PDF:

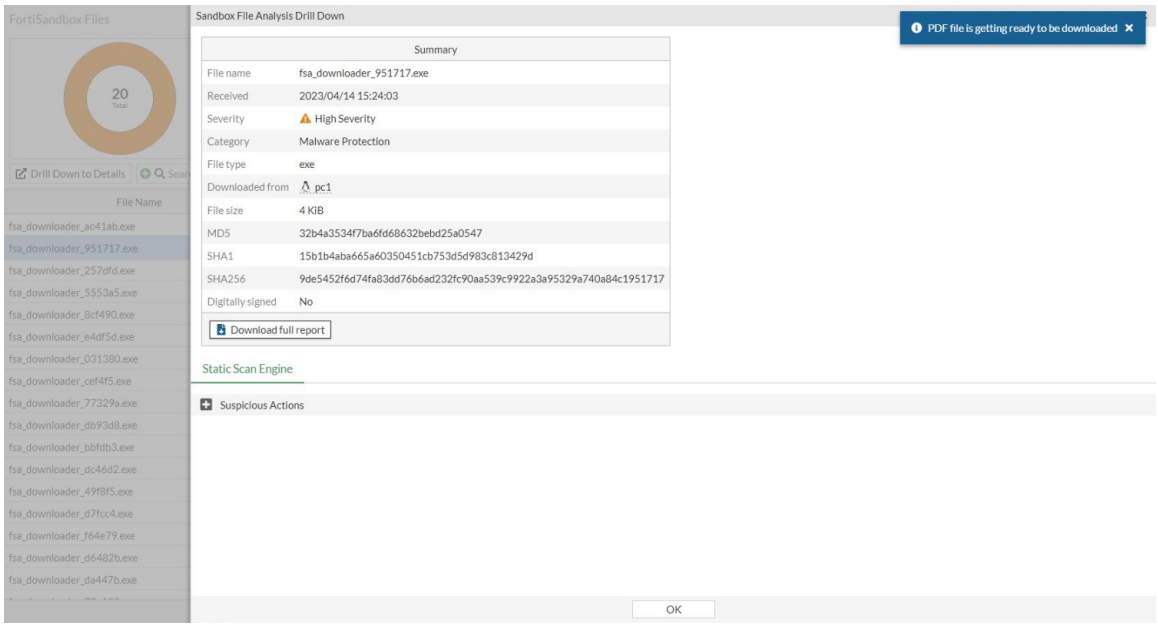
1. Go to *Dashboard* > *FortiSandbox Files*. The entry appears in the table, but the analysis is not available yet because the *Status* is *Pending*. The default view is *Files*.



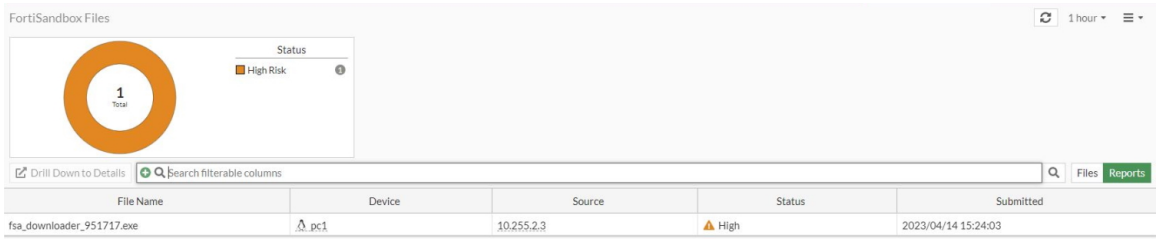
2. After about five to ten minutes, refresh the table. The analysis is now available.
3. Select the entry, then right-click and select *Drill Down to Details*.



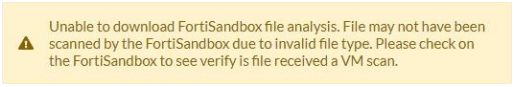
The *Sandbox File Analysis Drill Down* pane opens.



4. Click *Download full report* to download the detailed PDF report.
5. Change the view to *Reports* to verify that the file was downloaded successfully. The reports contains FortiSandbox job information and detailed file information.



When the file type is not supported, a warning message appears that the file was not scanned when the *Sandbox File Analysis Drill Down* pane opens.



To change the maximum number of PDFs kept in memory:

```
# diagnose test analytics-pdf-report max <integer>
```

The range is 1 - 10, and the default is 10. After the FortiGate is restarted, this value will revert to the default.

Connecting to FortiTelemetry agents

This topic applies only to a FortiGate acting as a FortiTelemetry Controller with *FortiTelemetry* enabled on the *System > Feature Visibility* page. For a summary of FortiTelemetry-related tasks that you can do in FortiOS, see FortiTelemetry on page 139. See also *FortiTelemetry Administration Guide*.

The FortiTelemetry Controller connects to FortiTelemetry agents through the *Telemetry* connector. The following agent connection methods are supported:

- FortiTelemetry Controller automatically discovers telemetry agents and displays them in the *Telemetry* connector for manual authorization. You must authorize each agent before the controller can use it.
- You can configure pre-authorized telemetry connectors, and FortiTelemetry Controller uses the connectors to automatically authorize discovered telemetry agents for use.

FortiTelemetry agents are either hardware-based or software-based. Deployed agents must be in the same subnet as the internal interface of the FortiTelemetry Controller.

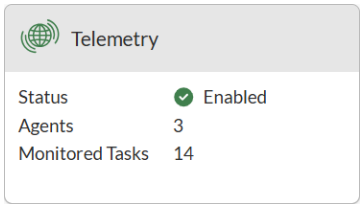
After FortiTelemetry agents are authorized, a telemetry address with the agent's name and *Telemetry* sub-type is automatically created.

Viewing the Telemetry connector and agents

The *Telemetry* connector displays the number of FortiTelemetry agents. Edit the *Telemetry* card to display the list of agents, and edit each agent to display and edit its details.

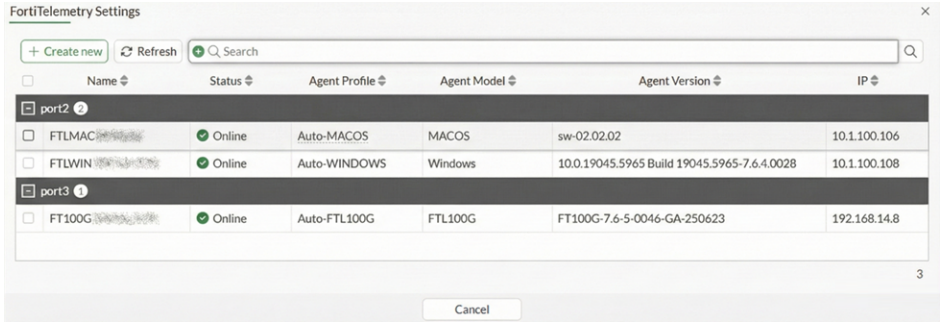
To view the Telemetry connector and FortiTelemetry agents:

1. Go to *Security Fabric* > *Fabric Connectors*. The *Telemetry* connector is displayed.



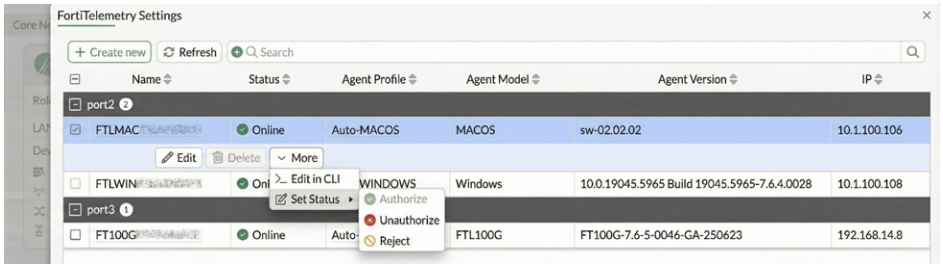
Status	Status of FortiTelemetry: *Enabled* or *Disabled*.
Agents	The number of online, authorized FortiTelemetry agents discovered by the FortiTelemetry Controller.
Monitored Tasks	Number of tasks being monitored by the FortiTelemetry agents based on the configured telemetry profile(s) selected in the firewall policy used by the FortiTelemetry Controller.

2. Click the *Telemetry* connector, and click *Edit*. The *FortiTelemetry Settings* pane opens.
   FortiTelemetry agents are grouped by interface.

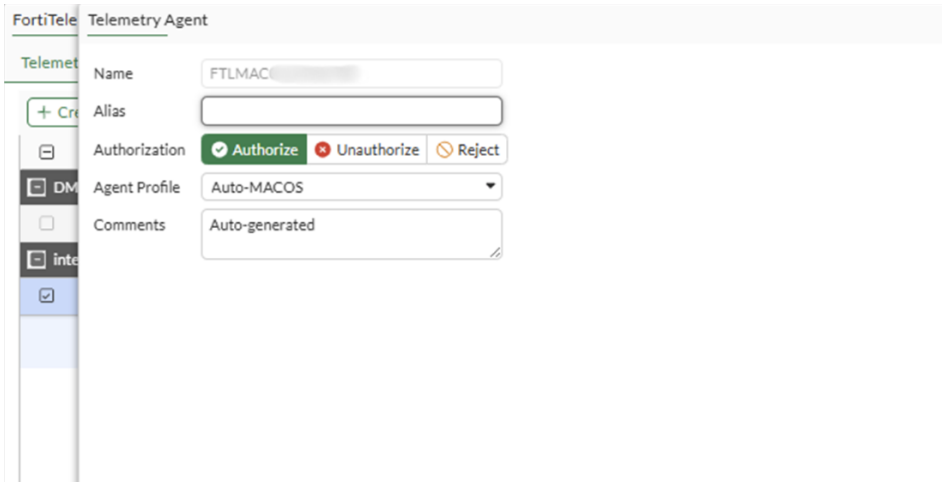


Create new	Click to create pre-authorized Telemetry connectors to automatically authorize FortiTelemetry agents. See Configuring pre-authorized telemetry connectors on page 4368.
Name	Name of the FortiTelemetry agent.
Status	Status of the FortiTelemetry agent. Change the status by hovering over the status to display the *Edit* icon. Click *Edit* to choose *Authorize*, *Unauthorize*, or *Reject* and click *Apply*.
Agent Profile	Profile assigned to the agent when FortiTelemetry Controller discovers the agent. FortiTelemetry Controller automatically creates and assigns the following profiles when no pre-configured profiles exist: • The *Auto-WINDOWS* agent profile is assigned to software agents. • The *Auto-FTL100G* agent profile is assigned to hardware agents. • The *Auto-MACOS* agent profile is assigned to software agents. Agent profile details can be viewed in the CLI using the `config telemetry-controller agent-profile` command.
Agent Model	Model of the agent: Windows for software agents and FTL100G for hardware agents.
Agent Version	Agent version.
IP	IP address of the FortiTelemetry agent.

3. Select an agent to access additional buttons, such as *Edit*, *Delete*, and *More*.



4. Select an agent and click *Edit*. The *Telemetry Agent* pane opens.



5. Click *OK* to close the *Telemetry Agent* pane.
6. Click *Cancel* to close the *FortiTelemetry Settings* pane.

Authorizing discovered FortiTelemetry agents

The FortiTelemetry Controller automatically discovers FortiTelemetry agents, displays them in the *Telemetry* connector, and assigns to each agent a profile.

You must manually authorize each discovered FortiTelemetry agent before the FortiTelemetry Controller can use it.

Some settings can be edited in the GUI or CLI, such as adding an alias or comment or changing the authorization status.

To authorize FortiTelemetry agents in the GUI:

1. Go to *Security Fabric* > *Fabric Connectors*.
2. Click the *Telemetry* connector, and click *Edit*. The *FortiTelemetry Settings* pane opens.
3. Select an agent, and click *More* > *Set Status* > *Authorize*.

To authorize FortiTelemetry agents in the CLI:

1. By default, automatically discovered telemetry agents are unauthorized, but you can authorize each agent after it connects to FortiGate.

```
config telemetry-controller agent
  edit "FT100GTK24000002"
    set authz authorize
  next
end
```

To edit FortiTelemetry agent settings in the GUI:

1. Go to *Security Fabric* > *Fabric Connectors*.
2. Click the *Telemetry* connector, and click *Edit*. The *FortiTelemetry Settings* pane opens.
3. Select an agent, and click *Edit* to view or edit the following options on the *Telemetry Agent* pane:

Name	Displays the name of the FortiTelemetry agent. You cannot change the name.
Alias	Enter an alias for the FortiTelemetry agent.
Authorization	Change the authorization status to *Authorize*, *Unauthorize*, or *Reject*.
Agent Profile	Displays the assigned agent profile. You cannot change the profile.
Comments	Enter optional comments to help identify the agent.

4. Click *OK* to save changes.

Configuring pre-authorized telemetry connectors

You can configure telemetry connectors to automatically authorize agents after they connect to the FortiTelemetry Controller. You must know the agent name to configure pre-authorized telemetry connectors. The agent name is used to match the discovered agent to the corresponding telemetry connector.

You can create and use a custom agent profile, or you can use a default agent profile (Auto-WINDOWS and Auto-MACOS for software agents, and Auto-FTL100G for hardware agents) if the FortiTelemetry Controller has created a default agent profile.

If you create an agent profile, ensure that the model in the agent profile matches the type of agent used.

To create agent profiles in the CLI:

1. Create an agent profile for the type of agent you are using.
   A profile for hardware agents should use the ftl-100g model, and a profile for software agents should use the windows or macos model.

```
config telemetry-controller agent-profile
    edit "WINDOWS-pre-auth"
        set comment dev win devices
        set model windows
    next
    edit "FTL100G-pre-auth"
        set comment hardware
        set model ftl-100g
    next
    edit "MACOS-pre-auth"
        set comment dev macos devices
        set model macos
    next
end
```

To create pre-authorized telemetry connectors in the GUI:

1. Go to *Security Fabric > Fabric Connectors*.
2. Click the *Telemetry* connector, and click *Edit*. The *FortiTelemetry Settings* pane opens.
3. Click *Create New*, and set the following options on the *Telemetry Agent* pane:

Name	Enter the agent name. The name starts with `FTLWIN` for Windows agents, `FTLMAC` for macOS agents and `FT100G` for hardware agents.
Alias	(Optional) Enter an alias for the FortiTelemetry agent.
Authorization	Select *Authorize*.
Agent Profile	Select an agent profile. Ensure the model configured in the profile matches the type of agent.
Comments	(Optional) Enter optional comments to help identify the agent.

4. Click *OK*. The telemetry connector is displayed in the uncategorized list until the FortiTelemetry Controller discovers the corresponding telemetry agent and uses the connector to automatically authorize the agent and assign a status of *Online*.

To create pre-authorized telemetry connectors in the CLI:

1. Create a pre-authorized telemetry connector for each agent to specify the agent name, authorization, and agent profile.

   The name starts with `FTLWIN` for Windows agents, `FTLMAC` for macOS agents and `FT100G` for hardware agents.

```
config telemetry-controller agent
    edit "FT100GTK24000007"
        set alias "FTL100G"
        set authz authorized
        set agent-profile "FTL100G-pre-auth"
    next
    edit "FTLWIN8662451639"
        set alias "WINDOWS-108"
        set authz authorized
        set agent-profile "WINDOWS-pre-auth"
    next
    edit "FTLMAC0123456789"
        set alias "macOS"
        set authz authorized
        set agent-profile "MACOS-pre-auth "
    next
end
```

Connecting to FortiData

Fabric connectors are used to integrate FortiData with FortiGate. You must configure both FortiData and FortiGate to establish the connection between them. Configure the products in the following order:

1. On FortiData, configure a fabric connector to FortiGate, and add FortiGate as an allowed device. See Connecting FortiData to FortiGate on page 134.
2. On FortiGate, enable FortiData. See Enabling FortiData on page 4369.
3. On FortiGate, authorize FortiData. See Authorizing FortiData on page 4370.

Information about Diagnostics on page 4371 is also included.

See FortiData labels on page 2662 for information about using FortiData labels.

Enabling FortiData

FortiData can be enabled only in the CLI.

To enable FortiData in the CLI:

1. In FortiOS, enable FortiData:

```
config system fortidata
    set status enable
    unset source-ip
    set interface-select-method auto
    set vrf-select 0
                              end
```

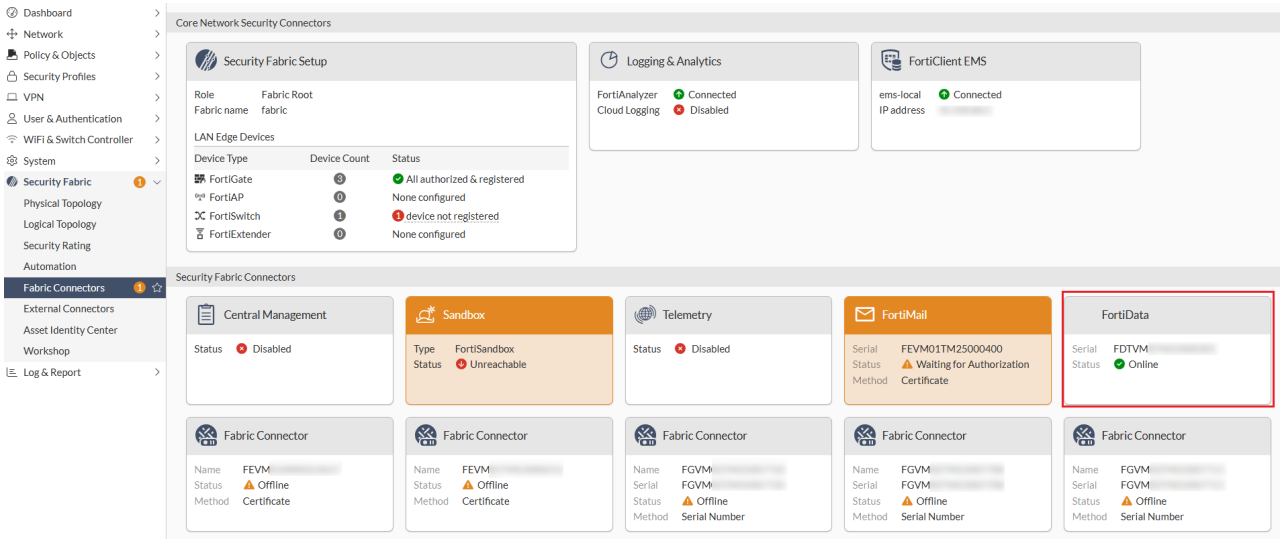
Command	Description
`status {enable | disable}`	Enable/disable FortiData.
`source-ip <string>`	Source IPv4 address used to communicate with FortiData.
`interface-select-method {auto | sdwan | specify}`	Specify how to select outgoing interface to reach server. • `auto`: Set outgoing interface automatically. • `sdwan`: Set outgoing interface by SD-WAN or policy routing rules. • `specify`: Set outgoing interface manually.
vrf-select	VRF ID used for connection to server (0 to 511, default = 0).

Authorizing FortiData

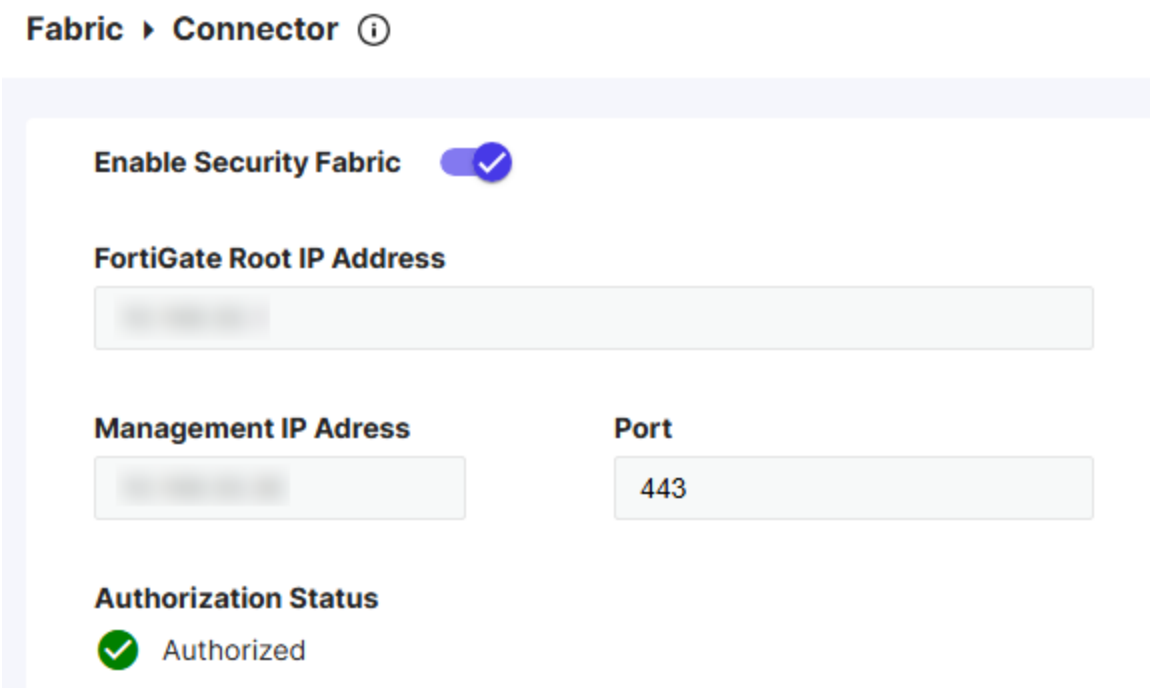
On FortiGate, authorize FortiData to establish a connection between the two products.

To authorize FortiData:

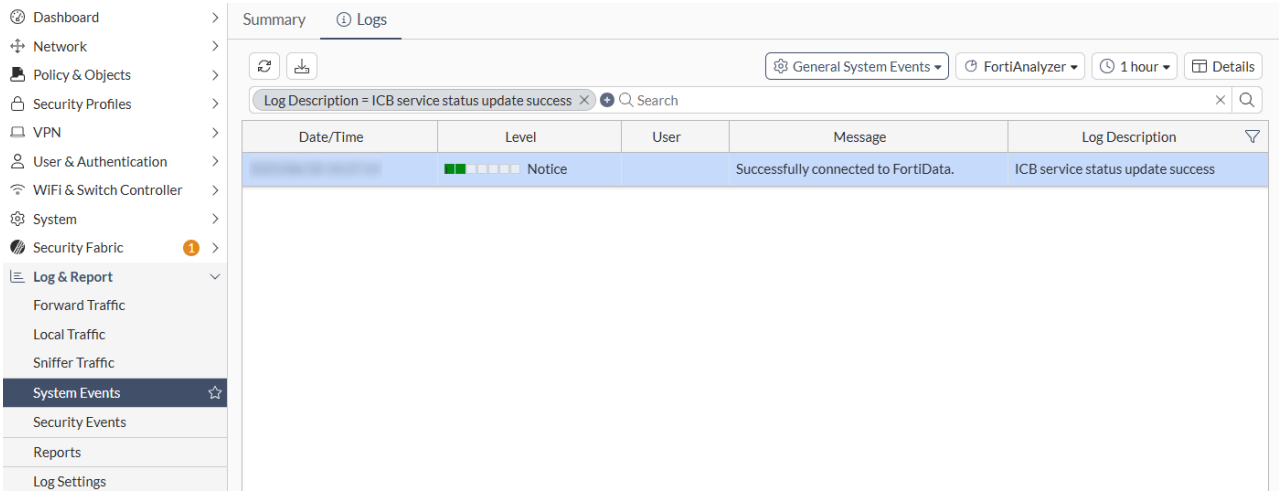
1. On the *Security Fabric > Fabric Connectors* page, click the *FortiData* card and select *Authorize*. Accept the certificate and verify that the status of the FortiData card changes to *Online*.



To further verify the authorization status in FortiData, go to the *Fabric > Connector* page and ensure that the authorization status is *Authorized*.



2. Go to the *Log & Report > System Events* page and verify the log stating that the FortiGate is successfully connected to the FortiData.



Diagnostics

Use the following diagnose commands to debug FortiData integration issues:

- Verify the integration is successful and a list of supported file types is returned:

```
# diagnose sys icb status
current time 2025-06-18 15:32:39 -07:00
icb.status_updated.count: 21
icb.status_updated.last.time: 1750282634
icb.status_updated.last.ago: 3325
```

```
icb.status_updated.last.local: 2025-06-18 14:37:14 -0700
======= fai =======
icb.state: stop
icb.file_type_list: not available
======= fsa =======
icb.state: stop
icb.file_type_list: not available
======= fortidata =======
icb.state: ready
icb.file_type_list:
ft.1: AVFT_JAVASCRIPT (7)
ft.2: AVFT_MSOFFICE (10)
ft.3: AVFT_PDF (42)
ft.4: AVFT_MSOFFICEX (52)
ft.5: AVFT_C_CPP (110)
ft.6: AVFT_JPEG (1006)
ft.7: AVFT_GIF (1007)
ft.8: AVFT_TIFF (1008)
ft.9: AVFT_PNG (1009)
ft.10: AVFT_BMP (1010)
icb.file_type_extensions:
ext.1: avif
ext.2: cer
ext.3: cfg
ext.4: conf
ext.5: crt
ext.6: csv
ext.7: env
ext.8: go
ext.9: ini
ext.10: java
ext.11: jpeg
ext.12: json
ext.13: key
ext.14: kt
ext.15: md
ext.16: odp
ext.17: ods
ext.18: odt
ext.19: p7b
ext.20: pem
ext.21: php
ext.22: py
ext.23: rtf
ext.24: service
ext.25: sh
ext.26: tiff
ext.27: toml
ext.28: txt
ext.29: webp
ext.30: yaml
```

```
ext.31: yml
```

- Show statistics:

```
# diagnose sys icb stats
```

- Restart the ICB manager daemon:

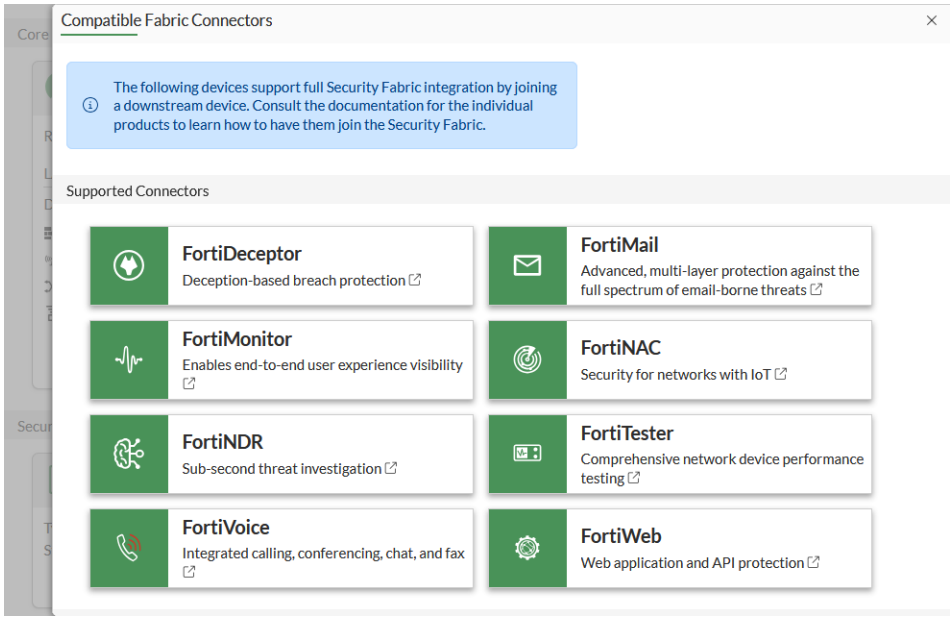
```
# diagnose sys icb restart
```

- Query FortiData-related information:

```
# diagnose sys fortidata {labels | local-labels | service-ip}
```

Configuring supported connectors

The *Supported Connectors* card displays the icons of different Fortinet devices that support full Security Fabric integration. Configuration of supported connectors requires some configuration on FortiOS and some configuration on the supported connector device.



Click a device name card to access documentation that explains how configure it in the Security Fabric.

Supported connectors overview

The following is an overview of how to add supported connectors to the Security Fabric:

1. In FortiOS, ensure Security Fabric is enabled, and prepare the root FortiGate for communication with supported Security Fabric devices. See Preparing FortiGate for supported Security Fabric devices on page 4374.

2. (Optional) Configure pre-authorization of the supported Security Fabric device. See Configuring pre-authorization of supported Security Fabric devices on page 4376.
3. On the supported Security Fabric device, configure the device to join the Security Fabric. See:
   - Configuring FortiDeceptor on page 4382
   - Configuring FortiMail on page 4383
   - Configuring FortiMonitor on page 4384
   - Configuring FortiNAC on page 4385
   - Configuring FortiNDR on page 4387
   - Configuring FortiTester on page 4387
   - Configuring FortiVoice on page 4389
   - Configuring FortiWeb on page 4390
4. Wait for the supported Security Fabric device to establish a connection with the root FortiGate in the Security Fabric.
5. In FortiOS, authorize the supported Security Fabric device. See Authorizing supported connectors on page 4381.

   If the supported device is pre-authorized, you can skip this step.
6. In FortiOS, go to *Security Fabric > Physical Topology* or *Security Fabric > Logical Topology* to view information about the authorized device.

Preparing FortiGate for supported Security Fabric devices

Before adding supported Security Fabric devices to FortiGate, ensure the following:

- On FortiGate, ensure that Security Fabric is enabled.
- On the root FortiGate of the Security Fabric, ensure that *Allow other Security Fabric devices to join* is enabled.
- On the root FortiGate, ensure that the appropriate interface is enabled to listen for supported Fabric devices.
- (As needed) On the root FortiGate, ensure that *Allow downstream device REST API access* is enabled, if the device requires REST API access to the root FortiGate, and select an administrator profile.

  The minimum permission required for the selected *Administrator profile* is *Read/Write* for *User & Device* (`set authgrp read-write`).

See Configuring the root FortiGate and downstream FortiGates on page 4302 for details.

Although optional, you can configure pre-authorization of the supported Fabric device on the root FortiGate. Pre-authorized devices can join the Security Fabric at any time, and do not require manual authorization in FortiOS. See Configuring pre-authorization of supported Security Fabric devices on page 4376.

The following table identifies commands used for adding supported devices to the Security Fabric.

Command	Description

`config system interface`
`edit <port name>`
`set allowaccess {protocols}`
	Specify management access to the port for the supported Security Fabric device.

Command	Description
next end	
config system csf set status enable	Enable the Security Fabric on FortiGate.
config system csf set group-name <string>	Specify a group name for the Security Fabric.
config system csf set downstream-access enable	On the root FortiGate of the Security Fabric, enable downstream access.
config system csf set downstream- accprofile <string>	Specify the administration profile used for REST API access.
config system csf config trusted-list	Configure pre-authorization for a device.

In this example FortiNDR is added to the Security Fabric using the CLI.

To add FortiNDR to the Security Fabric in the CLI:

1. Configure the interface to allow other Security Fabric devices to join:

```
config system interface
    edit "port1"
        set allowaccess ping https ssh http fgfm fabric
    next
end
```

2. Enable the Security Fabric:

```
config system csf
    set status enable
    set group-name "fabric-ai"
end
```

3. In FortiNDR, configure the device to join the Security Fabric:

```
config system csf
    set status enable
    set upstream-ip 10.6.30.14
    set managment-ip 10.6.30.251
end
```

4. Authorize the FortiNDR in FortiOS:

```
config system csf
    config trusted-list
        edit "FAIVMSTM21000000"
            set authorization-type certificate
            set certificate "********************"
        next
    end
end
```

Configuring pre-authorization of supported Security Fabric devices

When the serial number or certificate for a supported Security Fabric device is added to the trusted list on the root FortiGate, the device can join the Security Fabric as soon as it connects.

Pre-authorization is optional. When a supported Security Fabric device connects to the Security Fabric without pre-authorization configured, you can manually authorize the device in FortiOS. See Authorizing supported connectors on page 4381.

Before you can configure pre-authorization with a certificate, you must download the certificate for the device to your management computer.

To configure pre-authorization in the GUI:

1. On the root FortiGate, go to *Security Fabric* > *Fabric Connectors* and double-click the *Security Fabric Setup* card.
2. In the *Device authorization* field and click *Edit*. The *Device Authorization* pane opens.
3. Click *Create New* to add a new device for pre-authorization.
4. Enter the device name in the *Name* field.
5. Select the *Authorization type*, either *Serial Number* or *Certificate*.
6. If *Certificate* is selected, click *Browse* to upload the certificate from the management computer for the supported Security Fabric device.
7. Set the *Action* to *Accept*.
8. Click *OK* and add more devices as required.
9. Click *OK*.

To configure pre-authorization in the CLI:

This example shows how to configure pre-authorization of a FortiVoice with a certificate.

```
config system csf
    config trusted-list
        edit "<name>"
            set action accept
            set authorization-type certificate
            set certificate "-----BEGIN CERTIFICATE-----
...
```

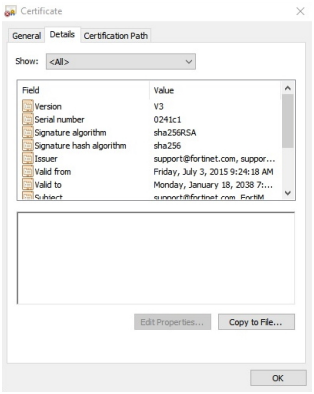
```
<encrypted_certificate_data>
...
-----END CERTIFICATE-----"
        next
    end
end
```

Pre-authorizing using the FortiMail certificate

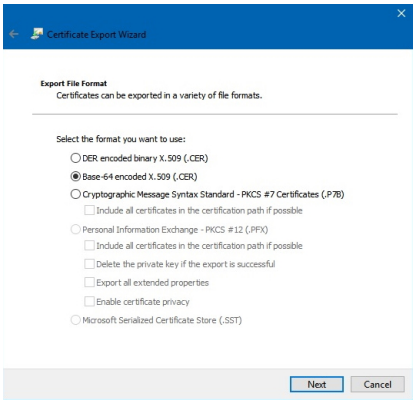
In this example, FortiMail is configured for pre-authorization using a certificate.

To pre-authorize FortiMail using a third-party or default certificate:

1. Log in to FortiMail.
2. Download the certificate. For example, in Chrome:
   a. In the left side of the address bar, click the icon to view the site information.
   b. Click *Certificate*.
   c. Click the *Details* tab, then click *Copy to File*.

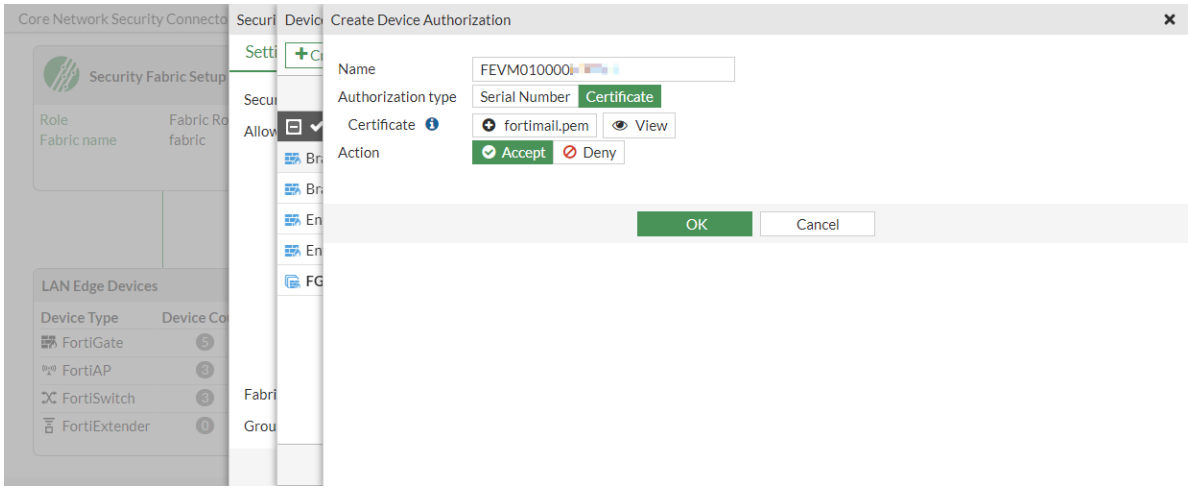


   d. The *Certificate Export Wizard* opens. Click *Next* to continue.
   e. For the file format, select *Base-64 encoded X.509 (.CER)*, then click *Next*.



   f. Browse to the folder location and enter a file name, then click *Next*.
   g. Click *Finish*, then click *OK* to close the dialog box.
3. In FortiOS, go to *Security Fabric > Fabric Connectors* and double-click the *Security Fabric Setup* card.

4. Beside *Device authorization*, click *Edit* > *Create New* and configure the following:
   a. Enter the FortiMail serial number.
   b. Set the *Authorization type* to *Certificate*.
   c. Click *Browse* to upload the .CER file you saved previously.



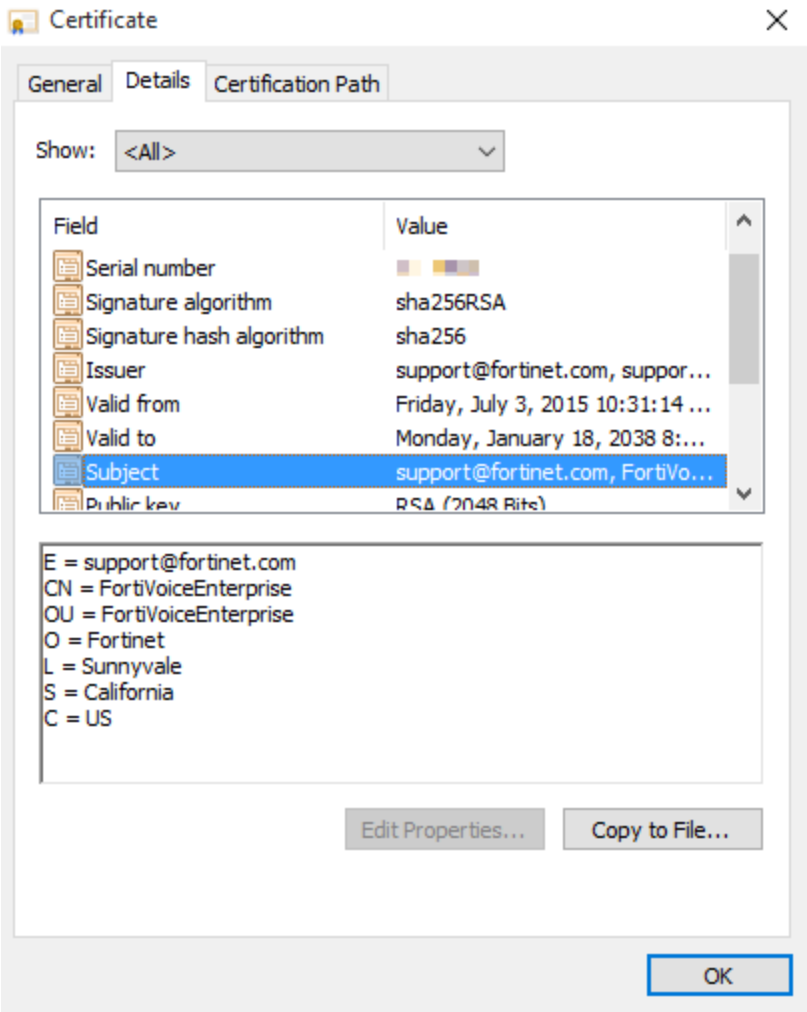
   d. Click *OK*.

Pre-authorizing using the FortiVoice certificate

In this example, FortiVoice is configured for pre-authorization using a certificate.

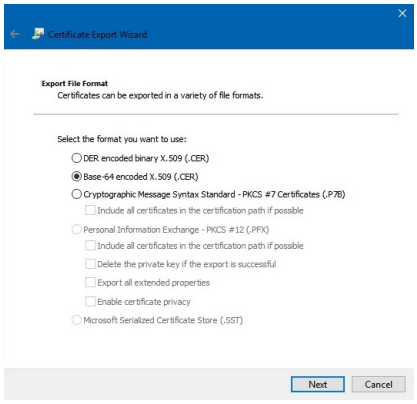
To pre-authorize a FortiVoice using a third-party or default certificate in the GUI:

1. Log in to the FortiVoice.
2. Download the certificate. For example, in Chrome:
   a. In the left side of the address bar, click the icon to view the site information.
   b. Click *Certificate*.
   c. In the *Certificate* window, click the *Details* tab, then click *Copy to File*.



d. The *Certificate Export Wizard* opens. Click *Next*.

e. Set the format to *Base-64 encoded X.509 (.CER)*, then click *Next*.



f. Browse to the folder location, enter a file name, then click *Next*.

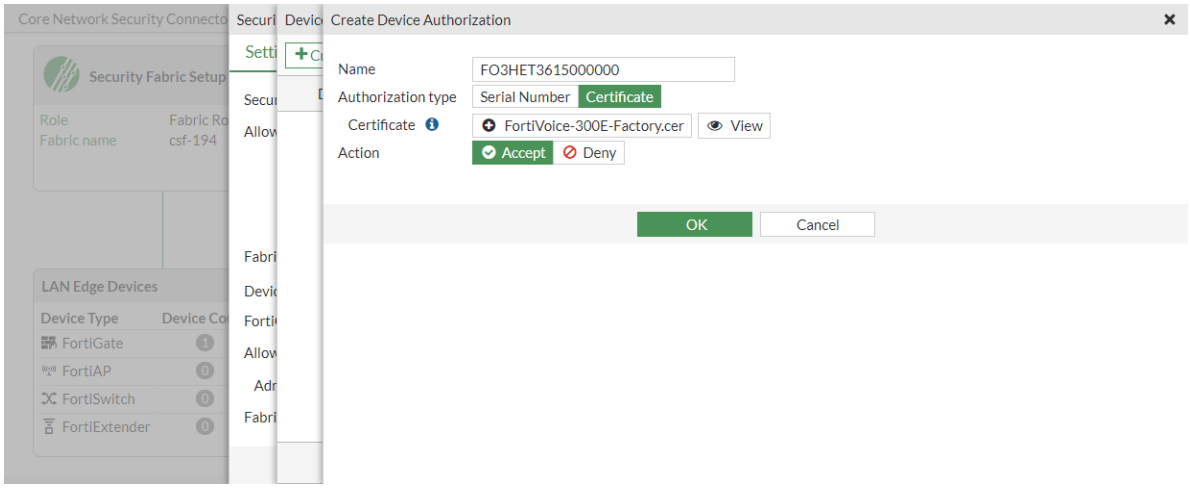
g. Click *Finish*, then click *OK* to close the wizard.

3. In FortiOS, go to *Security Fabric > Fabric Connectors* and double-click the *Security Fabric Setup* card.

4. Beside *Device authorization*, click *Edit*.

5. Click *Create New* and enter the following:

a. In the *Name* field, enter the FortiVoice serial number.
b. Set the *Authorization type* to *Certificate*.
c. Upload the .CER file.



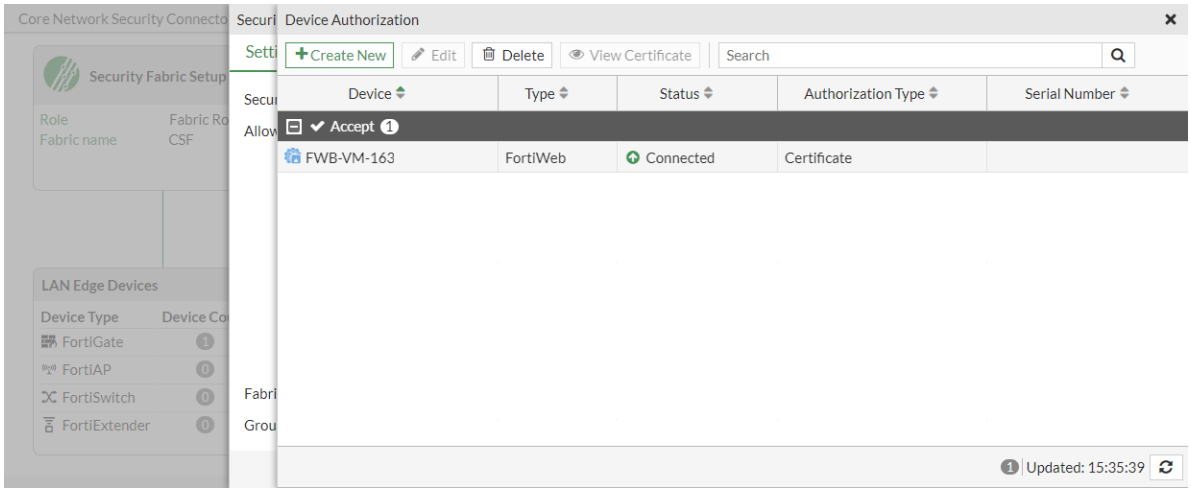
d. Click *OK*, then close the *Device authorization* pane.

Pre-authorizing using the FortiWeb certificate

In this example, FortiWeb is configured for pre-authorization using a certificate.

To authorize a FortiWeb to join the Security Fabric in FortiOS:

1. Go to *Security Fabric* > *Fabric Connectors* and double-click the *Security Fabric Setup* card.
2. Beside *Device authorization*, click *Edit*. The *Device authorization* pane opens.
3. Add the FortiWeb:
   a. Click *Create New* and enter a device name.
   b. For *Authorization type*, select *Certificate*.
   c. Click *Browse* to upload the certificate.
   d. For *Action*, select *Accept*.
   e. Click *OK*. The FortiWeb appears in the table.



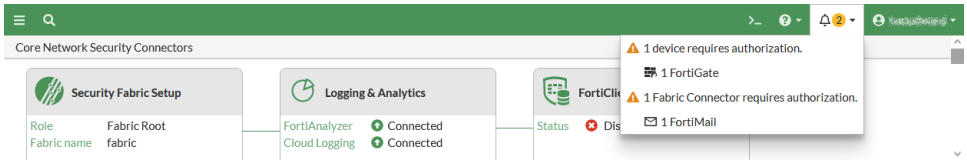
Authorizing supported connectors

Supported connectors can be authorized from the *Security Fabric > Fabric Connectors* page.

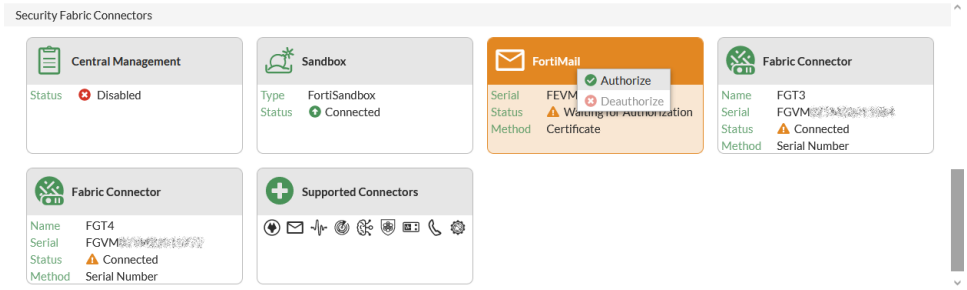
For information about authorizing other devices, see Authorizing devices on page 3773.

To authorize a supported connector in the GUI:

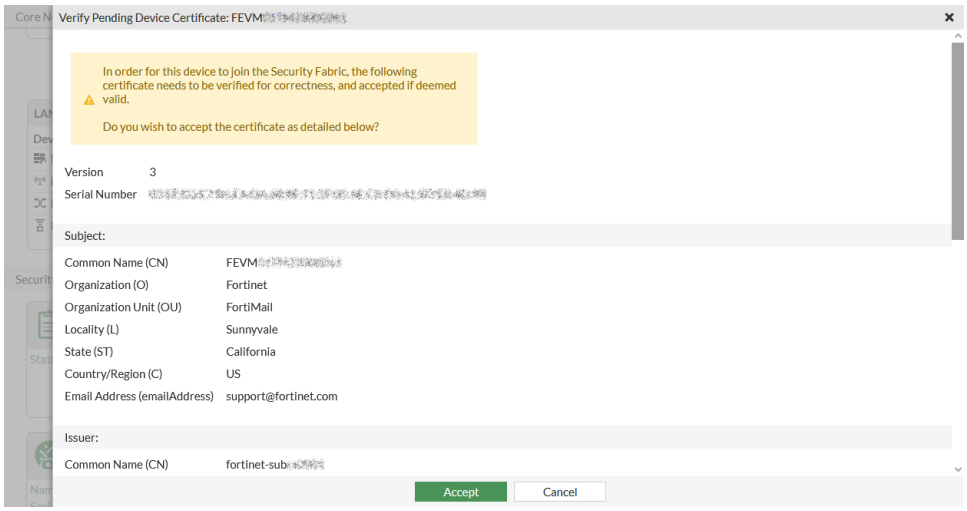
1. Go to *Security Fabric > Fabric Connectors* , either by selecting it in the tree menu, or clicking the link in the notification center dropdown.



2. In the *Security Fabric Connectors* section, click on the card of the unauthorized device, and select *Authorize*.



3. Verify the device certificate, then click *Accept*.



The device is authorized and should come online in a few moments. If the device does not come online in six to eight minutes, check its connection settings.

To authorize a supported connector in the CLI:

```
config system csf
    config trusted-list
        edit "<serial number>"
            set action accept
        next
    end
end
```

Configuring FortiDeceptor

FortiDeceptor can be added to the Security Fabric so it appears in the topology views and the dashboard widgets.

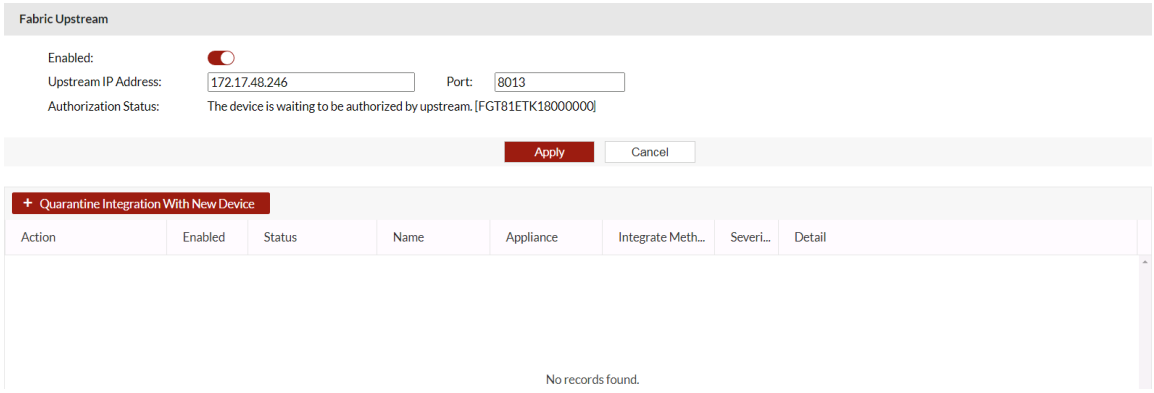
FortiDeceptor requires REST API access to FortiGate.

To add FortiDeceptor to the Security Fabric in the GUI:

1. In FortiOS, ensure that FortiGate is prepared to add FortiDeceptor to the Security Fabric. See Preparing FortiGate for supported Security Fabric devices on page 4374.
2. (Optional) In FortiOS, configure pre-authorization of FortiDeceptor to enable the device to join the Security Fabric as soon as it connects. See Configuring pre-authorization of supported Security Fabric devices on page 4376.
3. In FortiDeceptor, integrate the device:

   FortiDeceptor instructions are included for convenience. For the latest FortiDeceptor instructions, see the *FortiDeceptor Administration Guide*.

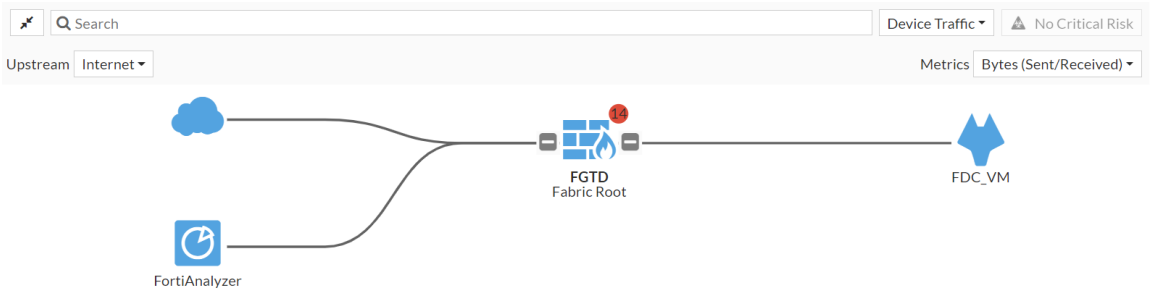
   a. Go to *Fabric* > *Quarantine Integration*.
   b. Click *Quarantine Integration With New Device*.
   c. Click the toggle to enable the device.
   d. For *Upstream IP Address*, enter the root FortiGate's management IP address.

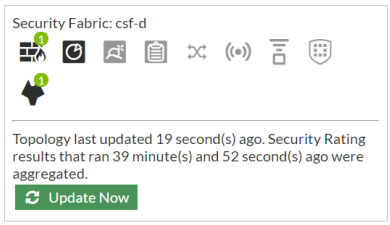


   e. Click *Apply*.
4. In FortiOS, authorize the FortiDeceptor. See Authorizing supported connectors on page 4381.

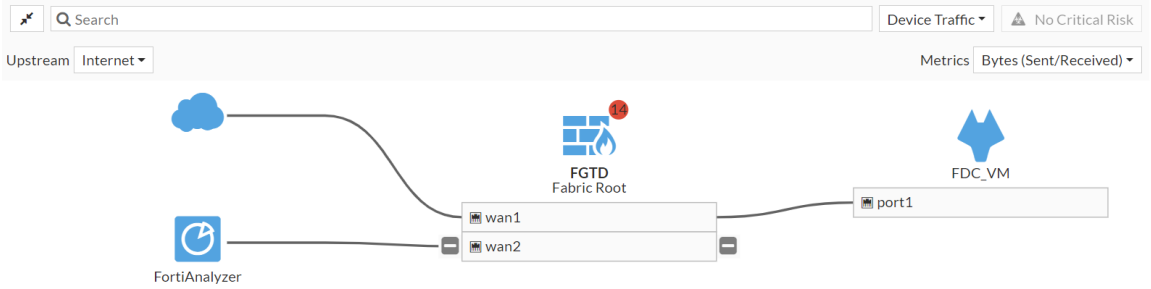
   If FortiDeceptor is pre-authorized, you can skip this step.
5. Go to *Security Fabric* > *Physical Topology* or *Security Fabric* > *Logical Topology* to view more information.

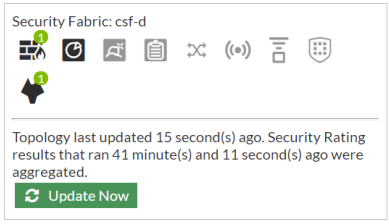
   Physical topology view:





Logical topology view:





Configuring FortiMail

FortiMail can be added to the Security Fabric so it appears in the topology views and the dashboard widgets.

To add FortiMail to the Security Fabric in the GUI:

1. In FortiOS, ensure that FortiGate is prepared to add FortiMail to the Security Fabric. See Preparing FortiGate for supported Security Fabric devices on page 4374.
2. (Optional) In FortiOS, configure pre-authorization of FortiMail to enable the device to join the Security Fabric as soon as it connects. See Configuring pre-authorization of supported Security Fabric devices on page 4376.
3. In FortiMail, integrate the device:

FortiMail instructions are included for convenience. For the latest FortiMail instructions, see the *FortiMail Administration Guide*.

a. Go to *System* > *Customization* and click the *Corporate Security Fabric* tab (or the *Corporate Security Fabric* tab in FortiMail 6.4.2 and earlier).
b. Click the toggle to enable the Fabric.
c. Enter the *Upstream IP Address* (root FortiGate) and the *Management IP* of the FortiMail.
d. Click *Apply*.

4. In FortiOS, authorize the device. See Authorizing supported connectors on page 4381.
   If FortiMail is pre-authorized, you can skip this step.
5. On FortiOS, go to *Security Fabric* > *Physical Topology* or *Security Fabric* > *Logical Topology* to view more information.

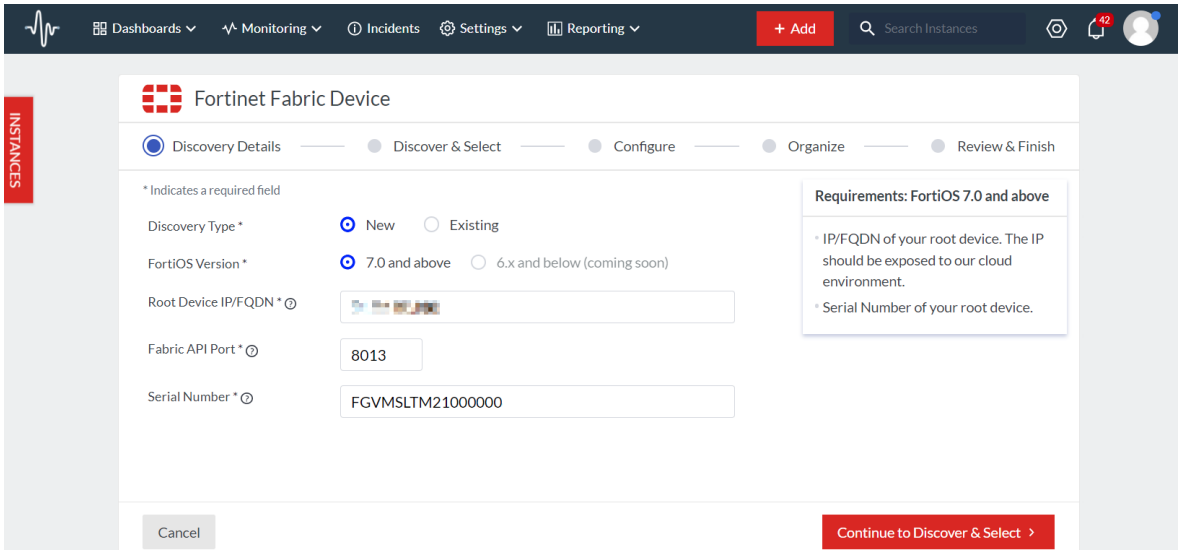
Configuring FortiMonitor

FortiMonitor can be added to the Security Fabric so it appears in the topology views and the dashboard widgets.

FortiMonitor requires REST API access to FortiGate.

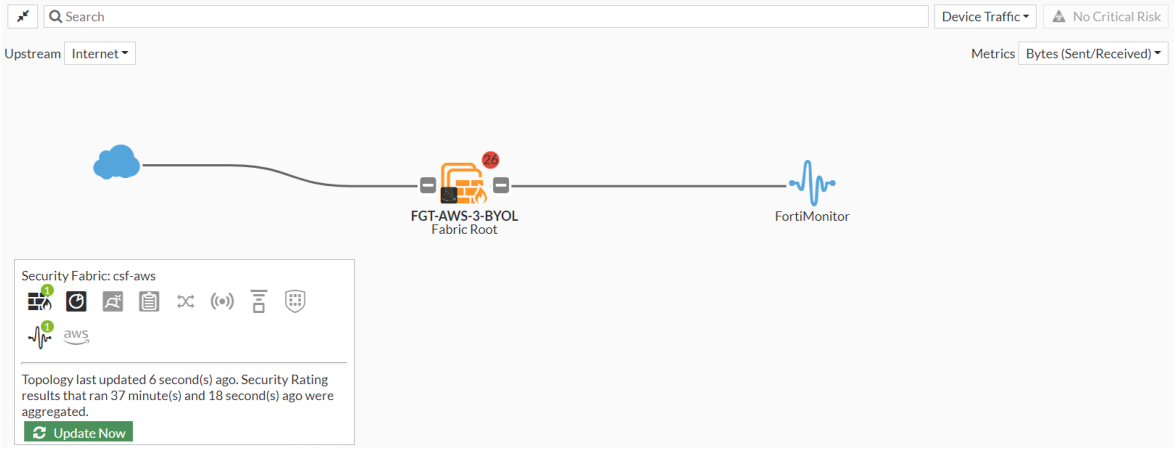
To add FortiMonitor to the Security Fabric:

1. In FortiOS, ensure that FortiGate is prepared to add FortiMonitor to the Security Fabric. See Preparing FortiGate for supported Security Fabric devices on page 4374.
2. (Optional) In FortiOS, configure pre-authorization of FortiMonitor to enable the device to join the Security Fabric as soon as it connects. See Configuring pre-authorization of supported Security Fabric devices on page 4376.
3. In FortiMonitor, start configuring the device to join the Security Fabric (see Enable Security Fabric monitoring for detailed instructions):
   a. Complete the *Discovery Details* page.

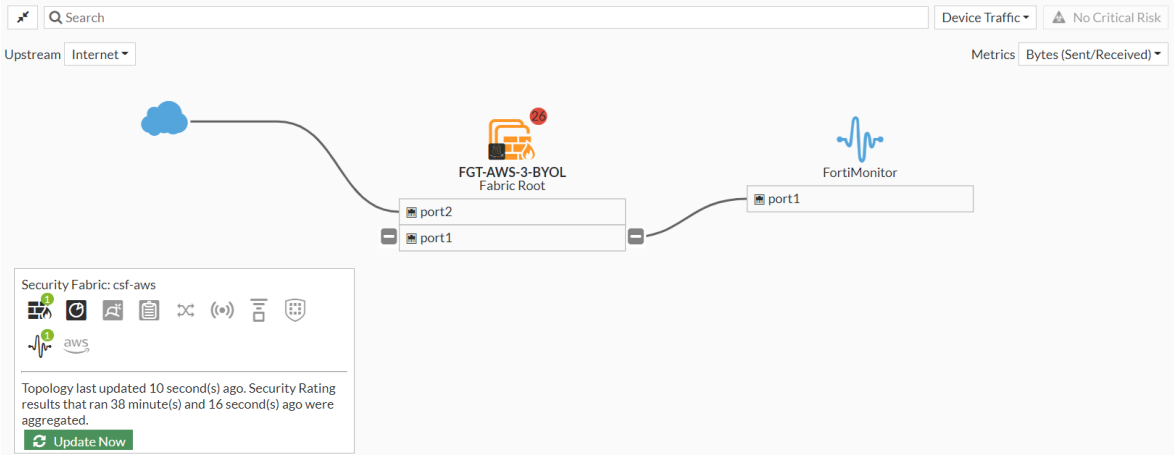


4. In FortiOS, authorize the FortiMonitor. See Authorizing supported connectors on page 4381.
   If FortiMonitor is pre-authorized, you can skip this step.

5. Go to *Security Fabric* > *Physical Topology* or *Security Fabric* > *Logical Topology* to view more information.
   Physical topology view:



   Logical topology view:



6. In FortiMonitor, complete the device configuration. (See Enable Security Fabric monitoring for detailed instructions.)

Configuring FortiNAC

A FortiNAC device can be added to the Security Fabric on the root FortiGate. After the device has been added and authorized, you can log in to the FortiNAC from the FortiGate topology views.

FortiNAC requires REST API access to FortiGate.

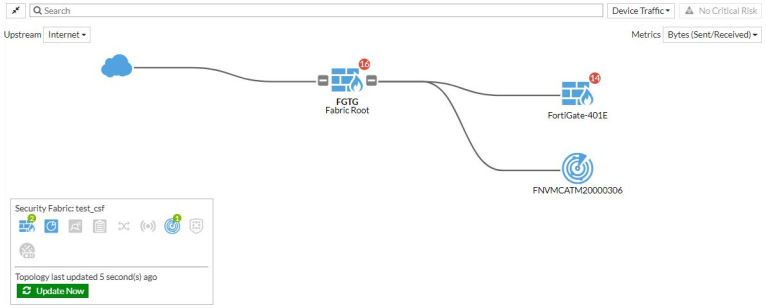
> Adding a FortiNAC to the Security Fabric requires a FortiNAC with a license issued in the year 2020 or later that includes an additional certificate. The device cannot be added if it has an older license. Use the `licensetool` in the FortiNAC CLI to determine if your license includes the additional certificate.

To add FortiNAC to the Security Fabric in the GUI:

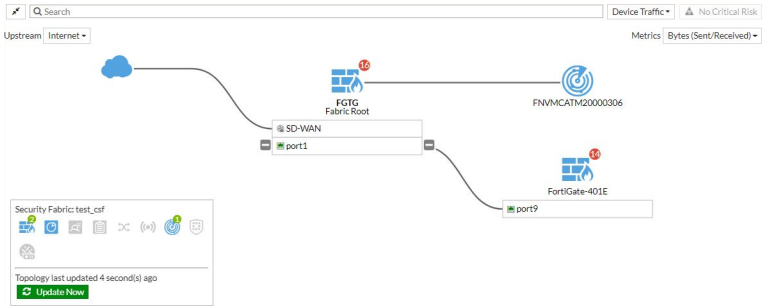
1. In FortiOS, ensure that FortiGate is prepared to add FortiNAC to the Security Fabric. See Preparing FortiGate for supported Security Fabric devices on page 4374. On the root FortiGate, *Allow downstream device REST API access* must be enabled.
2. (Optional) In FortiOS, configure pre-authorization of FortiNAC to enable the device to join the Security Fabric as soon as it connects. See Configuring pre-authorization of supported Security Fabric devices on page 4376.
3. On FortNAC, configure telemetry and input the IP address of the root FortiGate. See the FortiNAC Security Fabric SSO guide for more information.
4. In FortiOS on the root FortiGate, authorize the FortiNAC. See Authorizing supported connectors on page 4381.

   If FortiNAC is pre-authorized, you can skip this step.
5. Go to *Security Fabric > Physical Topology* or *Security Fabric > Logical Topology* to view more information.

   Physical topology view:



   Logical topology view:



6. Run the following command in the CLI to view information about the FortiNAC device's status:

```
# diagnose sys csf downstream-devices fortinac
{
  "path":"FG5H1E5818900126:FNVMCATM20000306",
  "mgmt_ip_str":"10.1.100.197",
  "mgmt_port":0,
  "admin_port":8443,
  "serial":"FNVMCATM20000306",
  "host_name":"adnac",
  "device_type":"fortinac",
  "upstream_intf":"port2",
  "upstream_serial":"FG5H1E5818900126",
  "is_discovered":true,
```

```
  "ip_str":"10.1.100.197",
  "downstream_intf":"eth0",
  "authorizer":"FG5H1E5818900126",
  "idx":1
}
```


Configuring FortiNDR

FortiNDR can be added to the Security Fabric so that it appears in the topology views and the dashboard widgets.

To add FortiNDR to the Security Fabric in the GUI:

1. In FortiOS, ensure that FortiGate is prepared to add FortiNDR to the Security Fabric. See Preparing FortiGate for supported Security Fabric devices on page 4374.
2. (Optional) In FortiOS, configure pre-authorization of FortiNDR to enable the device to join the Security Fabric as soon as it connects. See Configuring pre-authorization of supported Security Fabric devices on page 4376.
3. On FortiNDR:
   a. Verify that the pre-trained models (engines) are up to date. See FortiGuard in the FortiNDR Administration Guide for more information.
   b. Configure and authorize the FortiGate. See Security Fabric Connector in the FortiNDR Administration Guide for more information.
4. In FortiOS on the root FortiGate, authorize the FortiNDR. See Authorizing supported connectors on page 4381.
   If FortiNDR is pre-authorized, you can skip this step.
5. On FortiOS, go to *Security Fabric > Physical Topology* or *Security Fabric > Logical Topology* to view more information.

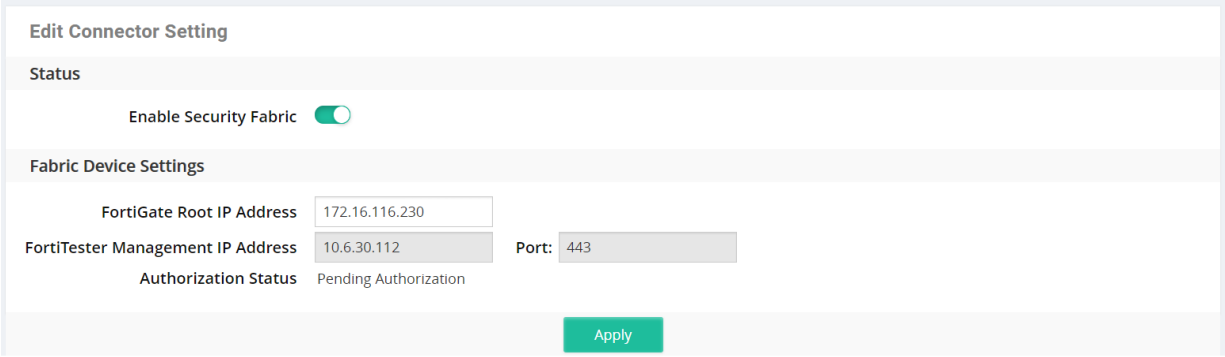
Configuring FortiTester

FortiTester can be added to the Security Fabric and authorized from the Security Fabric topology views. Once added, the FortiTester will appear in the *Security Fabric* widget on the dashboard.

To add FortiTester to the Security Fabric in the GUI:

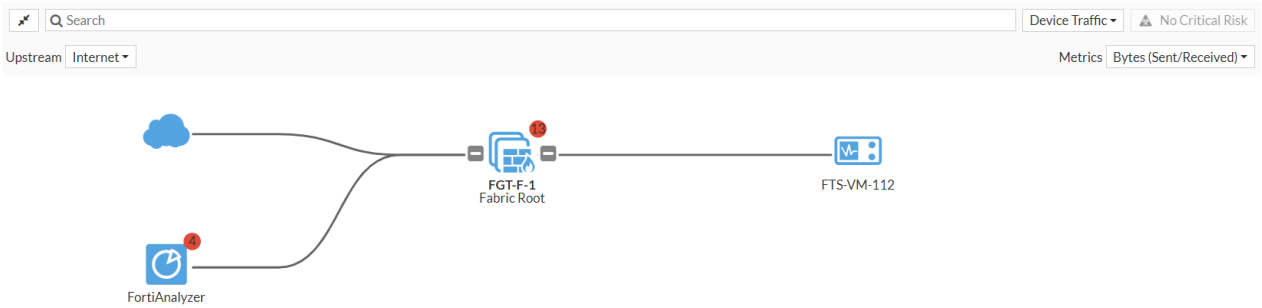
1. In FortiOS, ensure that FortiGate is prepared to add FortiTester to the Security Fabric. See Preparing FortiGate for supported Security Fabric devices on page 4374.
2. (Optional) In FortiOS, configure pre-authorization of FortiTester to enable the device to join the Security Fabric as soon as it connects. See Configuring pre-authorization of supported Security Fabric devices on page 4376.
3. In FortiTester, enable the Security Fabric:
   See also the *FortiTester Administration Guide*.
   a. Go to *System Settings > Security Fabric > Settings*.
   b. Click the toggle to enable the device (*Enable Security Fabric*).

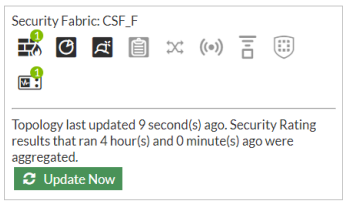
c. Enter the *FortiGate Root IP Address*.



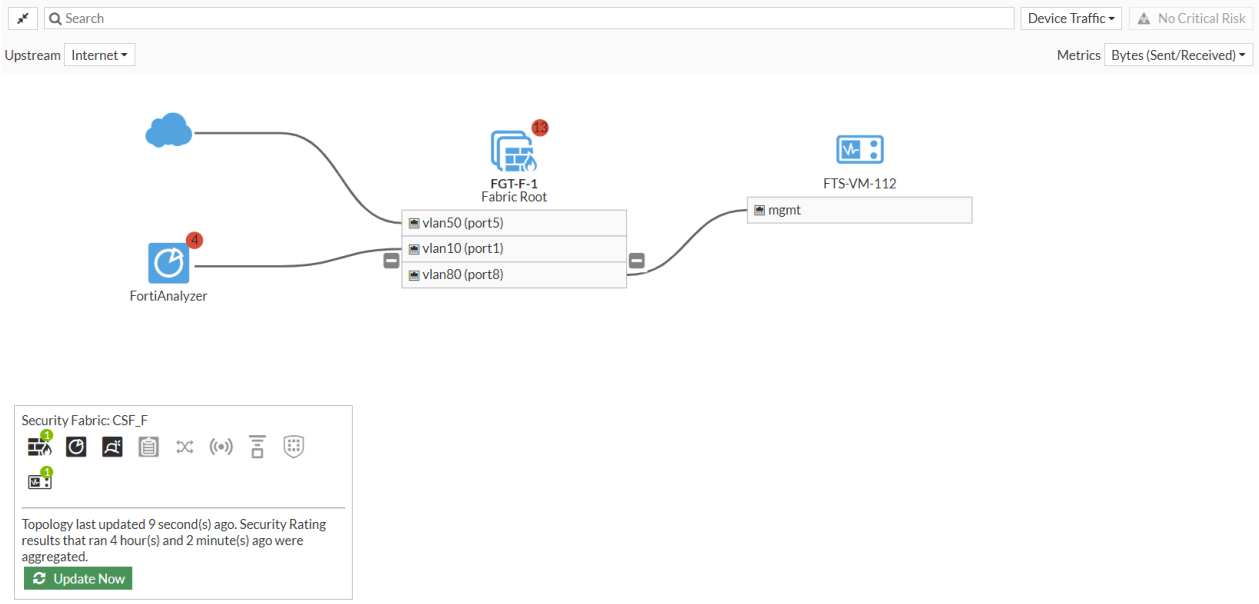
d. Click *Apply*.

4. In FortiOS, authorize the FortiTester. See Authorizing supported connectors on page 4381.
   If FortiTester is pre-authorized, you can skip this step.
5. Go to *Security Fabric > Physical Topology* or *Security Fabric > Logical Topology* to view more information.
   Physical topology view:





Logical topology view:

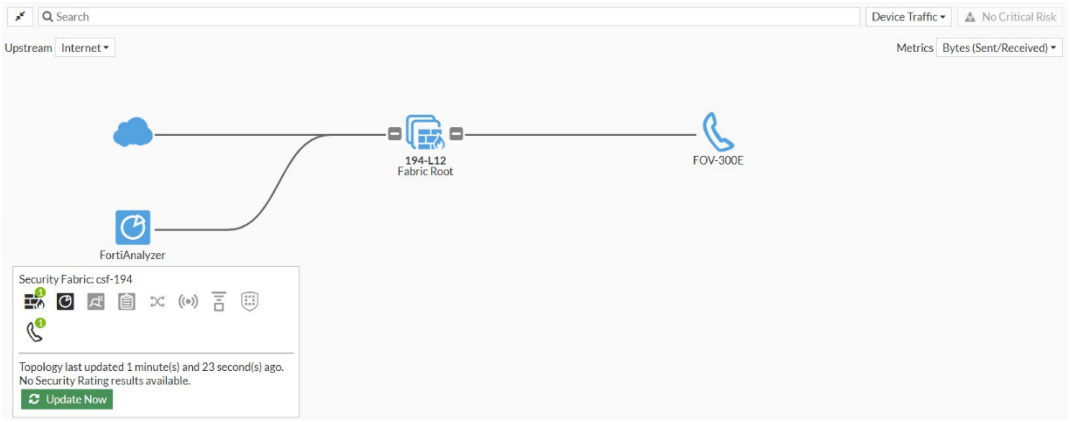


Configuring FortiVoice

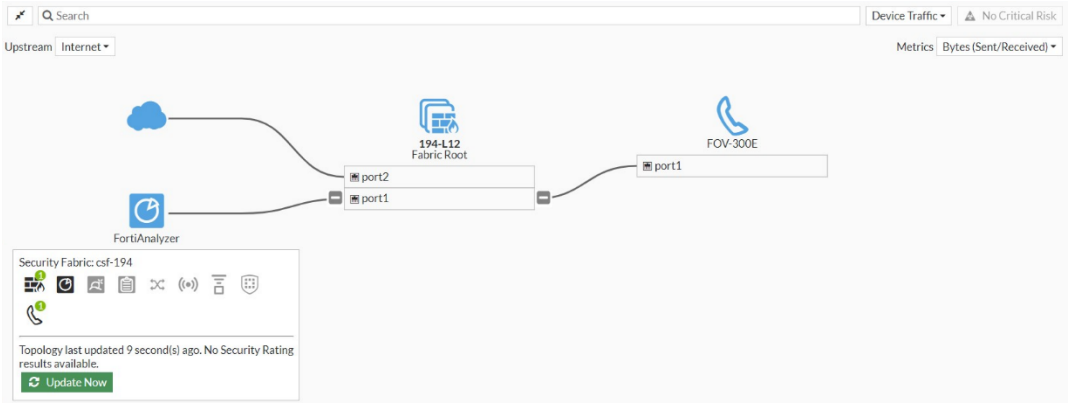
A FortiVoice can be added to the Security Fabric on the root FortiGate. Once the FortiVoice is added and authorized, you can log in to the device from the Security Fabric topology pages.

To add FortiVoice to the Security Fabric in the GUI:

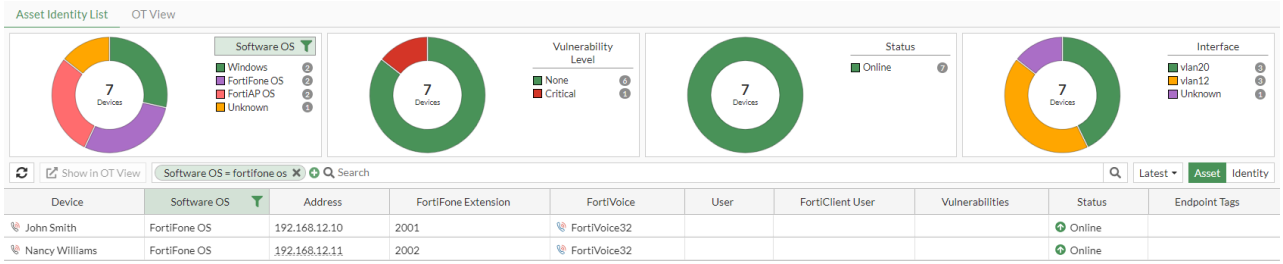
1. In FortiOS, ensure that FortiGate is prepared to add FortiVoice to the Security Fabric. See Preparing FortiGate for supported Security Fabric devices on page 4374.
2. (Optional) In FortiOS, configure pre-authorization of FortiVoice to enable the device to join the Security Fabric as soon as it connects. See Configuring pre-authorization of supported Security Fabric devices on page 4376.
3. On FortiVoice, enable the Security Fabric. See Enabling Security Fabric in the FortiVoice Phone System Administration Guide.
4. In FortiOS, authorize the FortiVoice. See Authorizing supported connectors on page 4381.
   If FortiVoice is pre-authorized, you can skip this step.
5. Go to *Security Fabric > Physical Topology* or *Security Fabric > Logical Topology* to view more information.
   Physical topology view:



Logical topology view:



6. For additional device (FortiFone) information, go to the *Security Fabric* > *Asset Identity Center* page.



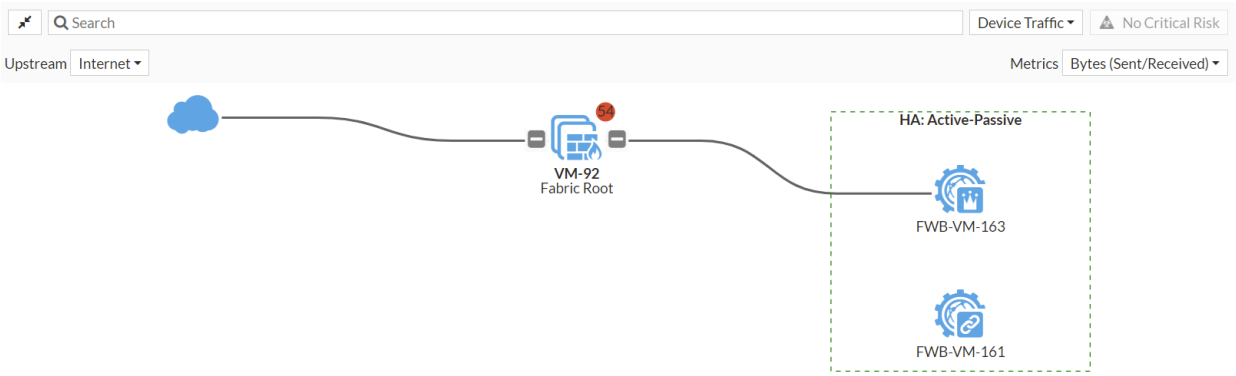
Configuring FortiWeb

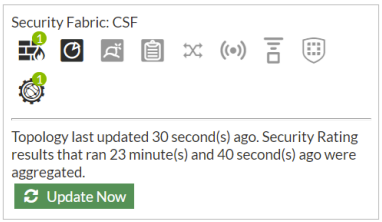
A FortiWeb can be added to the Security Fabric on the root FortiGate.

To add FortiWeb to the Security Fabric in the GUI:

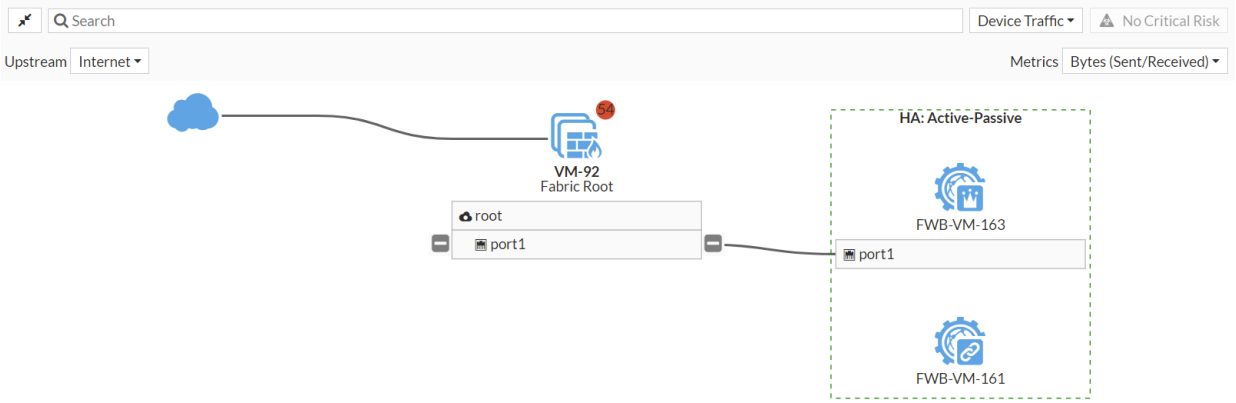
1. In FortiOS, ensure that FortiGate is prepared to add FortiWeb to the Security Fabric. See Preparing FortiGate for supported Security Fabric devices on page 4374.
2. (Optional) In FortiOS, configure pre-authorization of FortiWeb to enable the device to join the Security Fabric as soon as it connects. See Configuring pre-authorization of supported Security Fabric devices on page 4376.

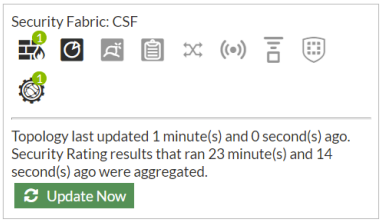
3. On FortiWeb, edit the FortiGate *Fabric Connector* settings. See Fabric Connector: Single Sign On with FortiGate. The *Connection Status* is currently *Authorize pending*.
4. In FortiOS, authorize the FortiWeb. See Authorizing supported connectors on page 4381.
   If FortiWeb is pre-authorized, you can skip this step.
5. Go to *Security Fabric* > *Physical Topology* or *Security Fabric* > *Logical Topology* to view more information.
   Physical topology view:





   Logical topology view:





Allowing FortiDLP Agent communication through the FortiGate

Every FortiDLP Agent requires a direct connection to the FortiDLP Cloud to report real-time data and receive configuration updates. This is outlined in Allowing communication between the FortiDLP Agent and FortiDLP Cloud.

As such, FortiDLP Agents operating behind the FortiGate firewall must be able to reach the FortiDLP Cloud servers. The servers must be trusted by the FortiGate and a corresponding firewall policy must allow traffic to these servers.

To view the FortiDLP Cloud server addresses on the FortiGate:

1. Go to *Policy & Objects* > *Internet Service Database*.
2. Search for *FortiDLP*.
3. Double-click the *Fortinet-FortiDLP.Cloud* entry to view it.
4. In the right hand panel, click *View/Edit Entries* to see the addresses.

To allow traffic to FortiDLP Cloud servers:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Click *Create new*.
3. Select the *Incoming interface* and the *Outgoing interface*.
4. Select the *Source* address.
5. For the *Destination* address
   a. Click in the field and, in the slide-out pane, select *Internet Service* from the drop-down menu.
   b. Search for *FortiDLP*.
   c. Select the *Fortinet-FortiDLP.Cloud* entry.
   d. Click *Close*.
6. Leave remaining settings as their default values and click *OK*.

Using the Security Fabric

The following topics provide information about using and deploying the Security Fabric:

- Dashboard widgets on page 4393
- Topology on page 4394
- Asset Identity Center page on page 4400
- OT asset visibility and network topology on page 4405
- KEV information in IoT/OT vulnerabilities on page 4412
- KEV information from FortiClient NEW on page 4416
- WebSocket for Security Fabric events on page 4422
- Deploying the Security Fabric on page 4423

- Deploying the Security Fabric in a multi-VDOM environment on page 4431
- Other Security Fabric topics on page 4437

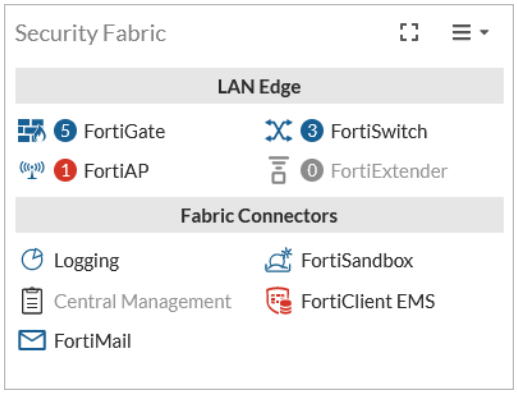
Dashboard widgets

Security Fabric widgets can be added to FortiGate dashboards, including:

- Security Fabric status on page 4393
- FortiGate Cloud on page 4393

Security Fabric status

The Security Fabric status widget shows a summary of the devices in the Security Fabric.



Hover the cursor over the top icons to view pop-ups showing the statuses of the devices in the fabric.

The device tree shows devices that are connected, or could be connected, to you Security Fabric, according to the following color scheme:

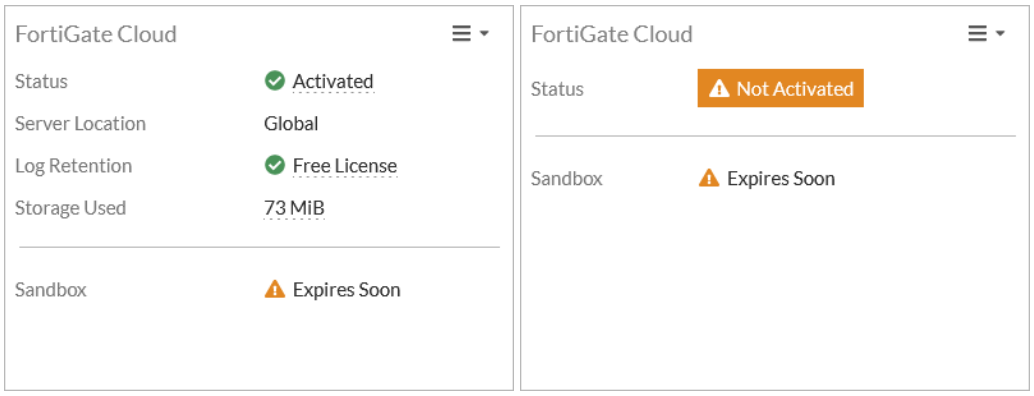
- Blue: connected to the network
- Gray: not configured or not detected
- Red: no longer connected or not authorized

Hover over a device in the tree to view details about the device, such as it's serial number, operation mode, IP address, CPU and memory usage, and others, depending on the device type.

Unauthorized FortiAP and FortiSwitch devices are highlighted in the list, and can be authorized by clicking on the device name.

FortiGate Cloud

The FortiGate Cloud widget shows the FortiGate Cloud status and information. If your account is not activated, you can activate it from the widget.



To activate your FortiGate Cloud account:

1. Click on the *Not Activated* button and select *Activate*. The *Activate FortiGate Cloud* pane opens.
2. If you already have an account:
   a. Fill in your email address, password, country or region, and reseller.
   b. Click *OK*.
3. If you are creating an account:
   a. In the *FortiCloud* field select *Create Account*.
   b. Fill in all of the required information.
   c. Click *OK*.

Topology

The full Security Fabric topology can be viewed on the root FortiGate. Downstream FortiGate devices' topology views do not include upstream devices.

The *Physical Topology* page shows the physical structure of your network, including all connected devices and the connections between them. The *Logical Topology* page shows information about the interfaces that connect devices to the Security Fabric.

In both topology pages, you can use filtering and sorting options to control the information that is shown. Hover the cursor over a device icon, port number, or endpoint to open a tooltip that shows information about that specific device, port, or endpoint. Right-click on a device to log into, configure, or deauthorize it. Right-click on an endpoint to perform various tasks, such as drilling down for more details in FortiView, quarantining the host, and banning the IP address.

The small number that might be shown in the top right corner of a device icon is the number of security ratings recommendations or warnings for that device. The circle color indicates the severity of the highest security rating check that failed. Clicking it opens the *Security Rating* page. See Security rating on page 4470 for more information.

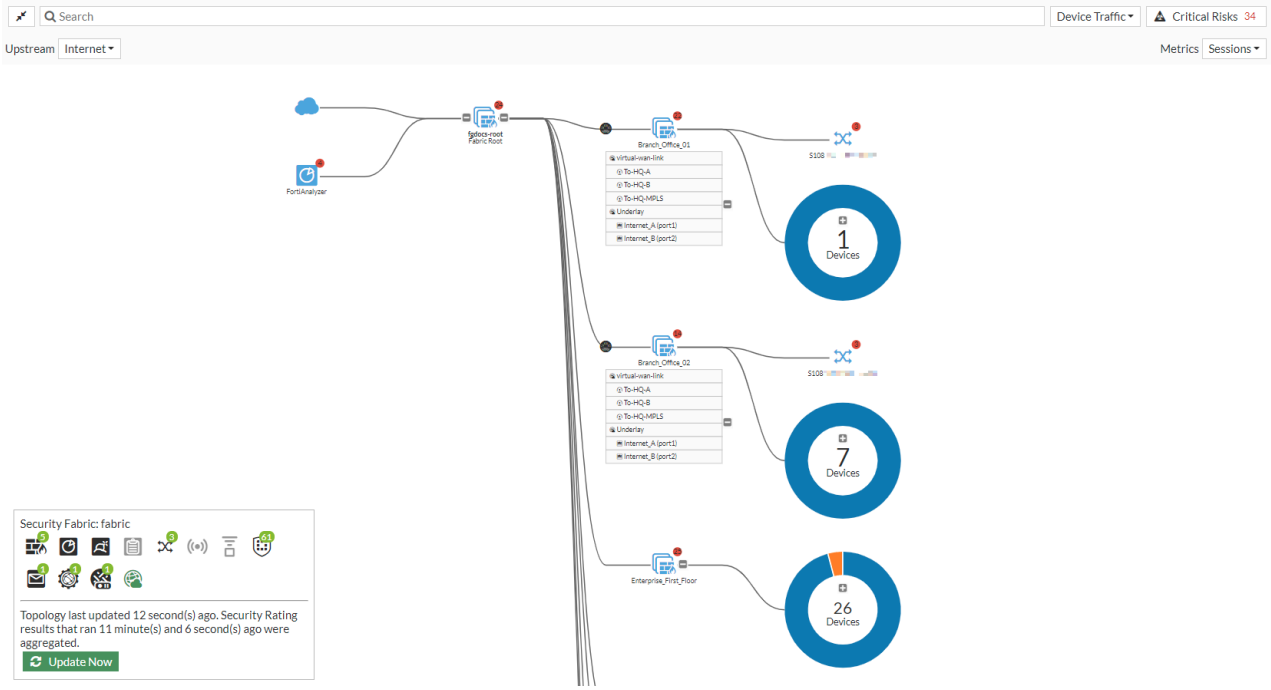
Views

From the dropdown list beside the search bar, select one of the following views:

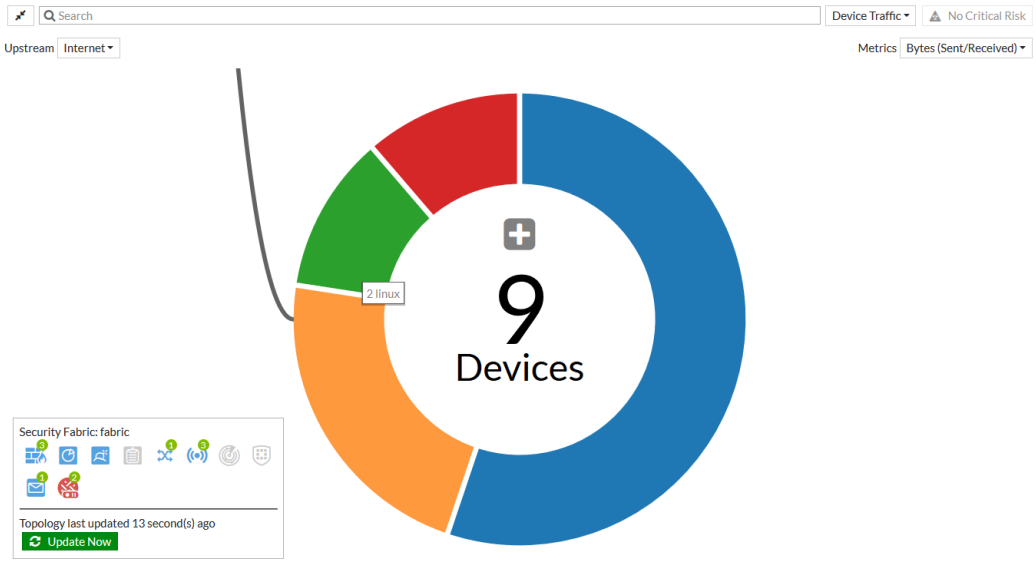
- *Device Traffic*: organize devices by traffic.
- *Device Count*: organize devices by the number of devices connected to it.
- *Device Operating System*: organize devices by operating system.
- *Device Hardware Vendor*: organize devices by hardware vendor.
- *Risk*: only include devices that have endpoints with medium, high, or critical risk values of the specified type: *All*, *Compromised Host*, *Vulnerability*, or *Threat Score*.
- *No Devices*: do not show endpoints.

Endpoint groups

The *Device Traffic* and *Device Count* views display endpoint groups as donut charts, with the total number of endpoints in the group in the center of the chart. Each sector of the donut chart represents a different endpoint operating system.



To zoom in on a donut chart, click any chart sector. Each sector represents a different endpoint OS. Hovering over each sector allows you to see the OS that the sector represents and the number of endpoints that have that OS installed.



In this example, the endpoint group contains a total of nine endpoints, with the following OSes installed:

Donut sector color	OS	Number of endpoints
Orange	Linux	2
Green	FortiMail	1
Red	FortiManager	1
Blue	Other	5

To view the endpoint group in a bubble pack display, click the + button in the center of the donut chart. You can view each individual endpoint in the bubble pack view.

FortiAP and FortiSwitch devices

Newly discovered FortiAP and FortiSwitch devices are initially shown in the topologies with gray icons to indicate that they have not been authorized. To authorize a device, click on the device icon or name and select *Authorize*. Once authorized, the device icon will turn blue.

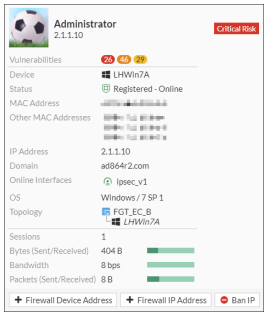
Right-click on an authorized FortiAP device to *Deauthorize* or *Restart* the device. Right-click on a FortiSwitch device to *Deauthorize*, *Restart*, or *Upgrade* the device, or to *Connect to the CLI*.

FortiAP and FortiSwitch links are enhanced to show link aggregation groups for the inter-switch link (ISL-LAG). To differentiate them from physical links, ISL-LAG links are shown with a thicker line. The endpoint circles can also be used as a reference to identify ISL-LAG groups that have more than two links.

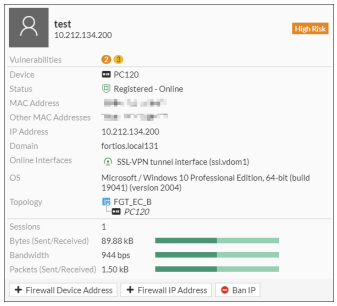
Managed clients connected over a VPN

When managed clients are connected over a VPN, EMS collects user information about these registered clients, such as the VPN connection information. The FortiGate can synchronize this user information from EMS and display it in the logical topology view to provide a detailed picture of clients and their associated VPN interfaces.

Client using an IPsec VPN interface:



Client using an Agentless VPN interface:



Critical risks

Click the *Critical Risks* button to see a list of endpoints that are deemed critical risks, organized by threat severity. These are the red endpoints in the current topology view.

For each endpoint, the user's photo, name, IP address, email address, and phone number are shown. The number of vulnerabilities of each severity is shown, and if the IoC verdict is that the endpoint is compromised.

If applicable, the endpoint's host can be quarantined (click *Quarantine Host*) or their IP address can be banned (click *Ban IP*).

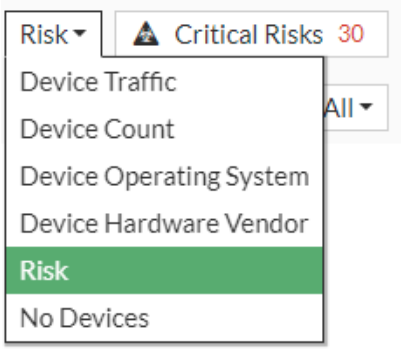
The dropdown menu also provides options to drill down to more information on compromised hosts or endpoint vulnerabilities.

Consolidated risk view

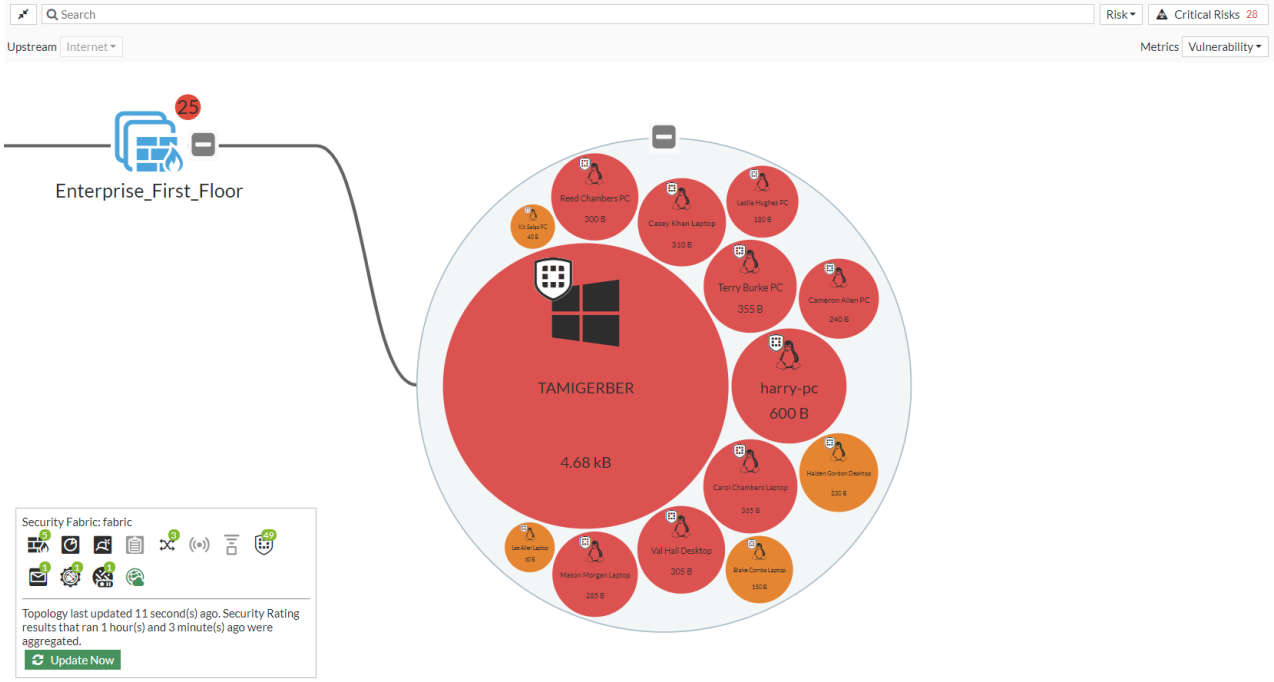
The consolidated *Risk* view mode displays different risks within the Security Fabric topology. You can use the *Risk* view mode to filter threats by *Compromised Hosts*, *Vulnerability*, and *Threat Score*.

To access the consolidated risk view mode:

1. On one of the topology pages, in the view option dropdown list beside the search bar, select *Risk*.



2. Select one of the following options from the *Risk Type* dropdown menu:
   a. *All*
   b. *Compromised Hosts*
   c. *Vulnerability*
   d. *Threat Score*
3. When devices fit into the risk metric, they will appear in the endpoint groups. Click the *+* in the endpoint group to display the devices in a bubble chart.



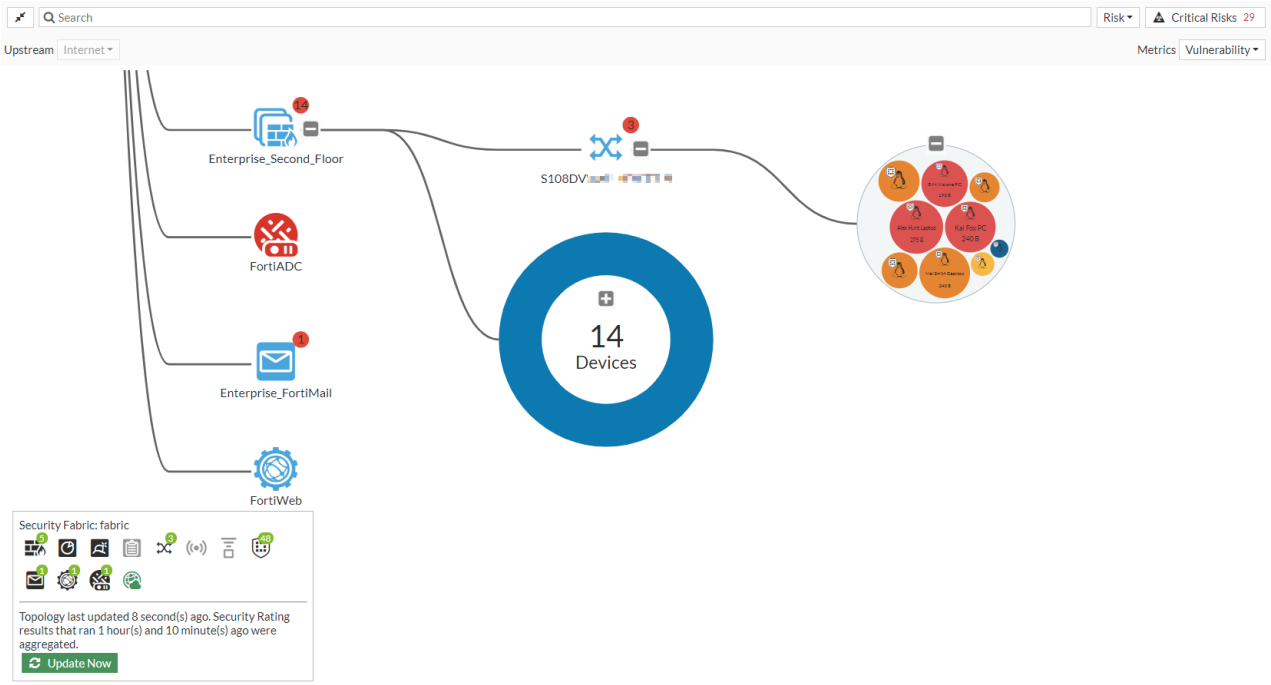
Viewing and controlling network risks in topology view

On the physical and logical topology pages, you can view and control compromised hosts. Compromised hosts behind a FortiSwitch or FortiAP can be quarantined.

To view a compromised endpoint host:

1. Test that FortiGate detects a compromised endpoint host by opening a browser on the endpoint host and entering a malicious website URL. The browser displays a *Web Page Blocked!* warning and does not allow access to the website.
2. On the root FortiGate, go to *Security Fabric > Physical Topology* or *Security Fabric > Logical Topology*. Expand the endpoint group connected to a FortiSwitch or FortiAP.

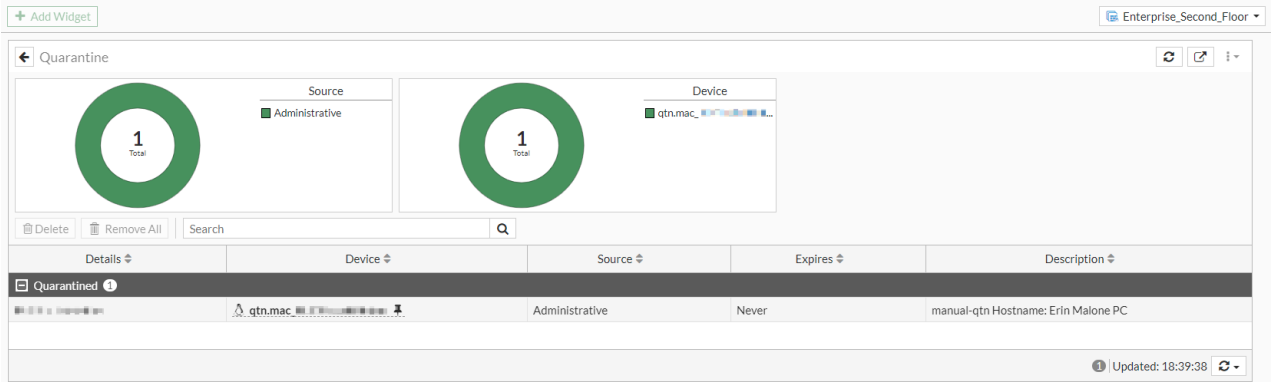
   The endpoint host connected to the switch is highlighted in red. Mouse over the endpoint host to view a tooltip that shows the IoC verdict. The endpoint host is compromised.



To quarantine a compromised endpoint host:

1. On the *Physical Topology* or *Logical Topology* page, right-click the endpoint host and select *Quarantine Host*.

   A dialog displays the FortiGate, host MAC address, and description of the host to be quarantined. Quarantine entries for each MAC address will be created on the FortiGate that the FortiSwitch or FortiAP is connected to.
2. Click *OK*.
3. Go to *Dashboard > Assets & Identities* and click the *Quarantine* widget to expand it.
4. In the top-right corner, use the dropdown to select the FortiGate in which this host was quarantined. In this example, it is the Enterprise_Second_Floor FortiGate.



5. On the endpoint host, open a browser and visit a website such as https://www.fortinet.com/. If the website cannot be accessed, this confirms that the endpoint host is quarantined.

To show the quarantined device from the CLI:

1. Log in to the downstream device where the host was quarantined (Enterprise_Second_Floor).
2. Enter the following show command:

```
Enterprise_Second_Floor # show user quarantine
config user quarantine
    set firewall-groups "QuarantinedDevices"
    config targets
        edit "Erin Malone PC"
            set description "Manually quarantined"
            config macs
                edit **:**:**:**:**:**
                    set description "manual-qtn Hostname: Erin Malone PC"
                next
            end
        next
    end
end
```

Asset Identity Center page

The *Asset Identity Center* page unifies information from detected addresses, devices, and users into a single page, while building a data structure to store the user and device information in the backend. The *Asset* view groups information by *Device*, while the *Identity* view groups information by *User*. Hover over a device or a user in the GUI to perform different actions relevant to the object, such as adding a firewall device address, adding an IP address, banning the IP, quarantining the host, and more.

When the FortiGate does not have a disk:

- Device charts are hidden on the *Security Fabric > Asset Identity Center* page.
- In *Dashboard > Assets & Identities*, the widget preview page displays a total count instead of the preview chart for asset-related widgets. Asset-related widgets include *Assets*, *Assets - FortiClient* and *Assets - Vulnerability*.

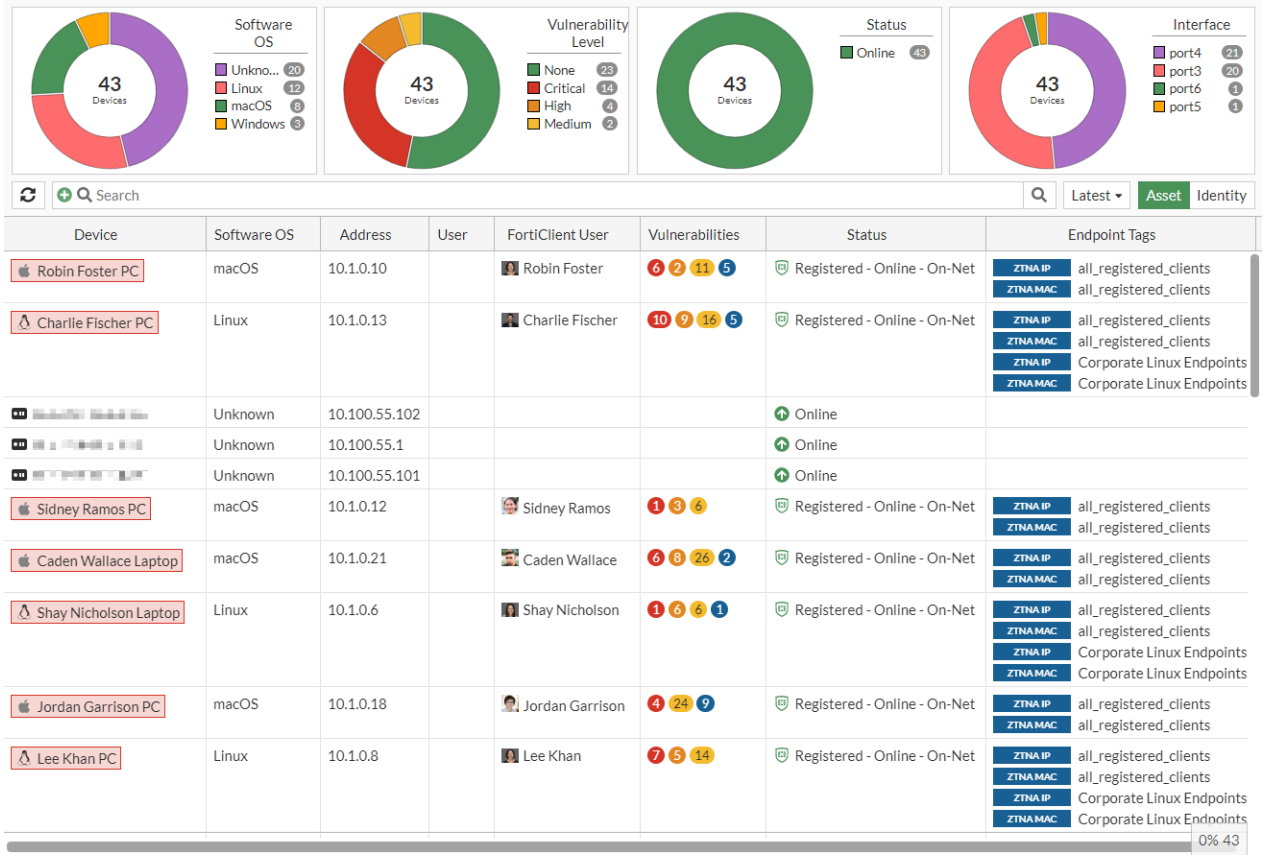
To view the Asset Identity Center page:

1. Go to *Security Fabric > Asset Identity Center*.
2. Click *Asset* to view information by device.

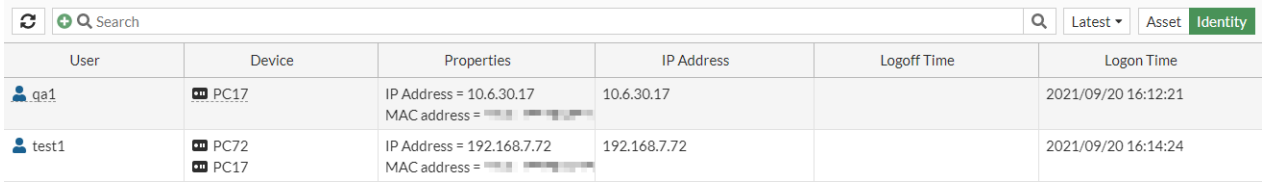
   There are four donut charts with device related information: *Software OS*, *Vulnerability Level*, *Status*, and *Interface*.

   The default table columns are *Device*, *Software OS*, *Address*, *User*, *FortiClient User*, *Vulnerabilities*, *Status*, and *Endpoint Tags*. The optional columns are *Device Family*, *Device Type*, *EMS Serial*, *EMS Tenant ID*, *Firewall Address*, *FortiSwitch*, *Hardware Vendor*, *Hardware Version*, *Hostname*, *Interface*, *IP Address*, *Last Seen*, *Port*, *Server*, *VLAN*, and *Vulnerability Level*.

   Devices with vulnerabilities are highlighted in red.

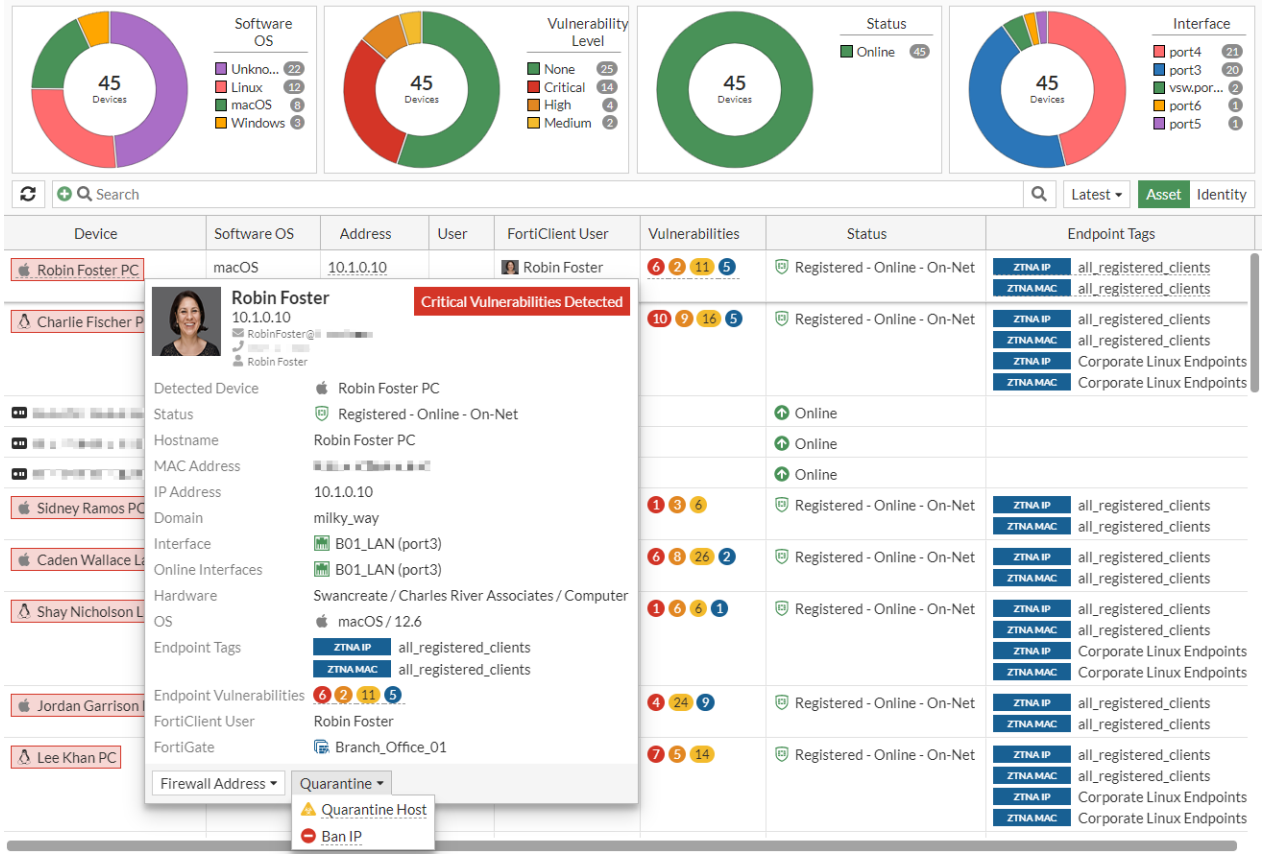


3. Click *Identity* to view information by user. The default table columns are *User*, *Device*, and *Properties*. The optional columns are *IP Address*, *Logoff Time*, and *Logon Time*.



Each view has a dropdown option to view the information within different time frames (*Latest*, *1 hour*, *24 hours*, and *7 days*). The page displays user and device relationships, such as which users are logged in to multiple devices or if multiple users are logged in to single devices.

4. Hover over a device in the list to view the tooltip and possible actions. The options under the *Firewall Address* dropdown are *Create Firewall Device Address* and *Create Firewall IP Address*. The options under the *Quarantine* dropdown are *Quarantine Host* and *Ban IP*.



Diagnostics

The following options are available for `diagnose user-device-store unified <option>`:

Option	Description
`device-memory-query`	Get device records and associated user records from memory.
`device-query`	Get device records and associated user records from memory and disk.

Option	Description
`user-memory-query`	Get user records and associated device records from memory.
`user-query`	Get user records and associated device records from memory and disk.
`re-query`	Retrieve query by `<query-id> <iteration-start> <iteration-count>` (takes 0-3 arguments).
`list`	List unified queries.
`clear`	Delete all unified queries.
`dump`	Dump unified query stats by `<query-id>` (takes 0-1 arguments).
`delete`	Delete unified query by `<query-id>` (takes 0-1 arguments).
`stats`	Get statistics for unified queries.
`debug`	Enable/disable debug logs for unified queries.

IoT vulnerabilities

Hovering over the data in the *Vulnerabilities* column displays a list of *FortiGuard IoT/OT Detected Vulnerabilities* and *FortiClient Detected Vulnerabilities*. Clicking the *View IoT/OT Vulnerabilities* button in the tooltip opens the *View IoT/OT Vulnerabilities* table that includes the *Vulnerability ID*, *Type*, *Severity*, *Reference*, *Description*, and *Patch Signature ID*. Each entry in the *Reference* column includes the CVE number and a link to the CVE details.

The following settings are required to display IoT devices:

1. The FortiGate must have a valid Attack Surface Security Rating service license.
2. Device detection must be configured on a LAN interface used by IoT devices.

   To configure device detection in the GUI:

   a. Go to *Network* > *Interfaces* and edit a LAN interface.
   b. Enable *Device detection*.
   c. Click *OK*.

   To configure device detection in the CLI:

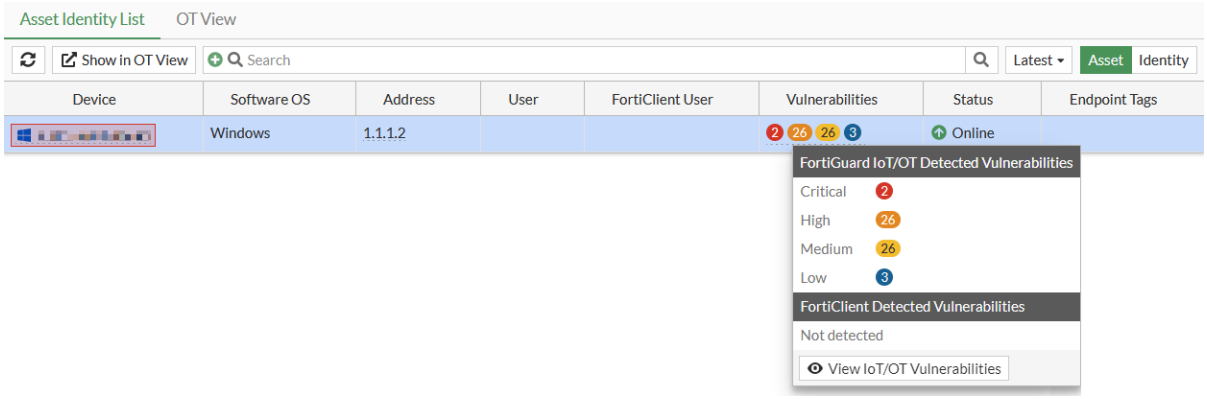
```
config system interface
    edit <name>
        set device-identification enable
    next
end
```

3. Configure a firewall policy with an application control sensor.

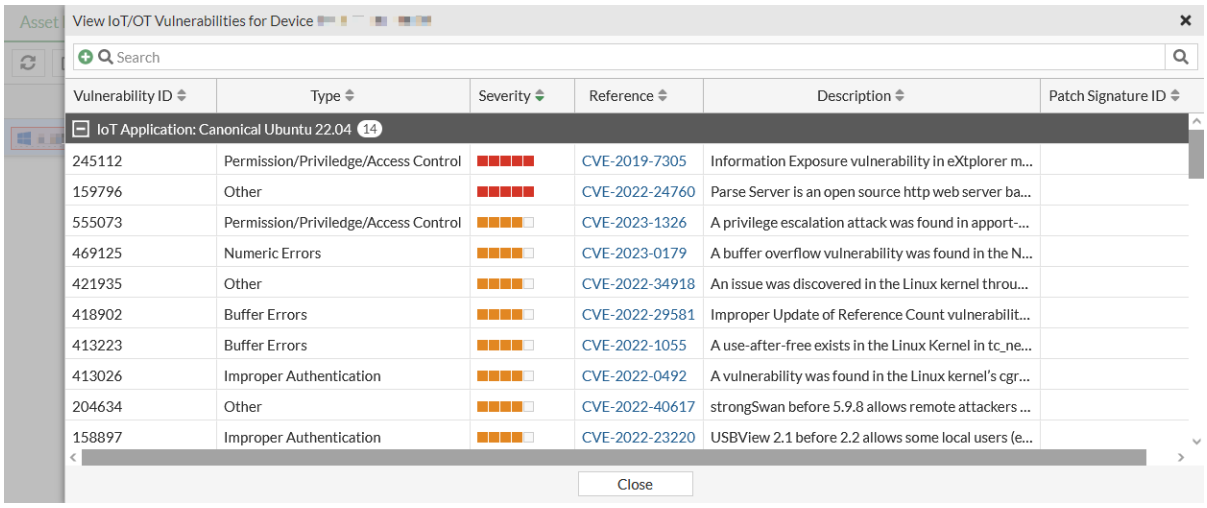
To view IoT asset vulnerabilities in the GUI:

1. Go to *Security Fabric* > *Asset Identity Center*. Ensure the *Asset* list view is selected.
2. Select a device with IoT vulnerabilities.

3. Hover over the *Vulnerabilities* count to view the tooltip and click *View IoT/OT Vulnerabilities*.



A table with the list of vulnerabilities and related information for the device is displayed, including the CVE references and descriptions.



4. Click a hyperlink in the *Reference* column to view more information about the CVE, or click *Close*.

To view IoT asset vulnerabilities in the CLI:

```
# diagnose user-device-store device memory list
...

        device_info
                'ipv4_address' = '1.1.1.2'
                'mac' = '**:**:**:**:**:**'
                'hardware_vendor' = 'Samsung'
                'hardware_type' = 'Home & Office'
                'hardware_family' = 'Computer'
                ...
                'purdue_level' = '3'
                'iot_vuln_count' = '57'
                'max_vuln_level' = 'Critical'
                'total_vuln_count' = '100'
        ...
        iot_info
```

```
                'vendor' = 'Mozilla'
                'product' = 'Firefox'
                'version-min' = '113.0'
                'validity' = 'true'
        iot_vulnerability
                'vulnerability_id' = '551873'
                'severity' = '2'
                'type' = 'Improper Authentication'
                'description' = 'The SSL protocol, as used in certain configurations in Microsoft
Windows and Microsoft Internet Explorer, Mozilla Firefox, Google Chrome, Opera, and other
products, encrypts data by using CBC mode with chained initialization vectors, which allows man-
in-the-middle attackers to obtain plaintext HTTP headers via a blockwise chosen-boundary attack
(BCBA) on an HTTPS session, in conjunction with JavaScript code that uses (1) the HTML5 WebSocket
API, (2) the Java URLConnection API, or (3) the Silverlight WebClient API, aka a "BEAST" attack.'
                'references' = 'CVE-2011-3389'
                'date_added' = '2023-04-19T12:12:32'
                'date_updated' = '2023-04-19T12:12:32'
        iot_vulnerability
                'vulnerability_id' = '534577'
                'severity' = '2'
                'type' = 'Other'
                'description' = 'The hb_buffer_ensure function in hb-buffer.c in HarfBuzz, as used
in Pango 1.28.3, Firefox, and other products, does not verify that memory reallocations succeed,
which allows remote attackers to cause a denial of service (NULL pointer dereference and
application crash) or possibly execute arbitrary code via crafted OpenType font data that triggers
use of an incorrect index.'
                'references' = 'CVE-2011-0064'
                'date_added' = '2023-04-19T11:59:20'
                'date_updated' = '2023-04-19T11:59:20'
        iot_vulnerability
                'vulnerability_id' = '525700'
                'severity' = '1'
                'type' = 'Other'
                'description' = 'The SPDY protocol 3 and earlier, as used in Mozilla Firefox,
Google Chrome, and other products, can perform TLS encryption of compressed data without properly
obfuscating the length of the unencrypted data, which allows man-in-the-middle attackers to obtain
plaintext HTTP headers by observing length differences during a series of guesses in which a
string in an HTTP request potentially matches an unknown string in an HTTP header, aka a "CRIME"
attack.'
                'references' = 'CVE-2012-4930'
                'date_added' = '2023-04-18T12:56:10'
                'date_updated' = '2023-04-18T12:56:10'
        ...
```

OT asset visibility and network topology

When the *Operational Technology (OT)* feature is enabled, tabs are added in the *Asset Identity Center* page to view the OT asset list and OT network topology using Purdue Levels. This feature is available regardless of whether a Security Fabric is enabled.

When the FortiGate does not have a disk:

- Device charts are hidden on the *Security Fabric > Asset Identity Center* page.
- In *Dashboard > Assets & Identities*, the widget preview page displays a total count instead of the preview chart for asset-related widgets. Asset-related widgets include *Assets*, *Assets - FortiClient* and *Assets - Vulnerability*.

To enable the OT features in the GUI:

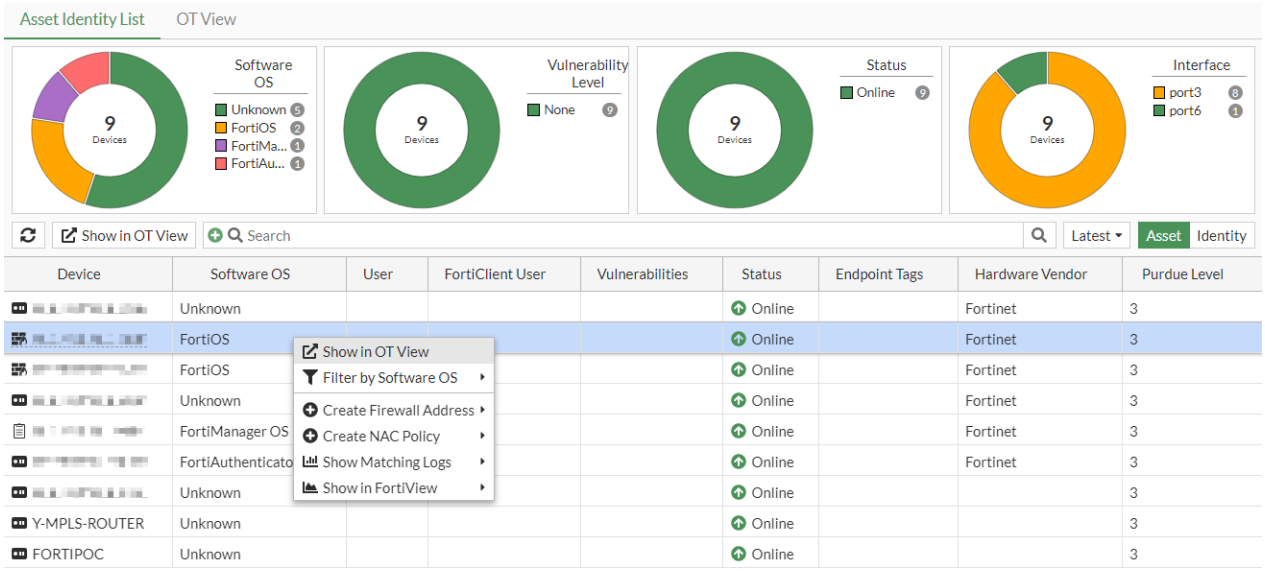
1. Go to *System > Feature Visibility*.
2. In the *Additional Features* section, enable *Purdue Levels*.
3. Click *Apply*.

To enable the OT features in the CLI:

```
config system settings
    set gui-ot enable
end
```

Once enabled, the *Security Fabric > Asset Identity Center* page displays an *Asset Identity List* tab and an *OT View* tab.

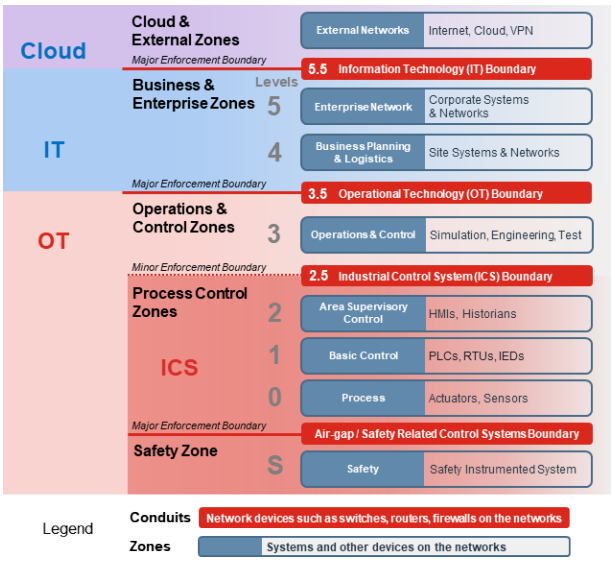
- The *Asset Identity List* tab includes a configurable *Purdue Level* column and a *Show in OT View* option for selected devices in the table.



- The *OT View* tab shows a topology of detected components and connections mapped to Purdue Levels. The default view is locked, but devices can be dragged and dropped to other Purdue Levels if the view is unlocked.

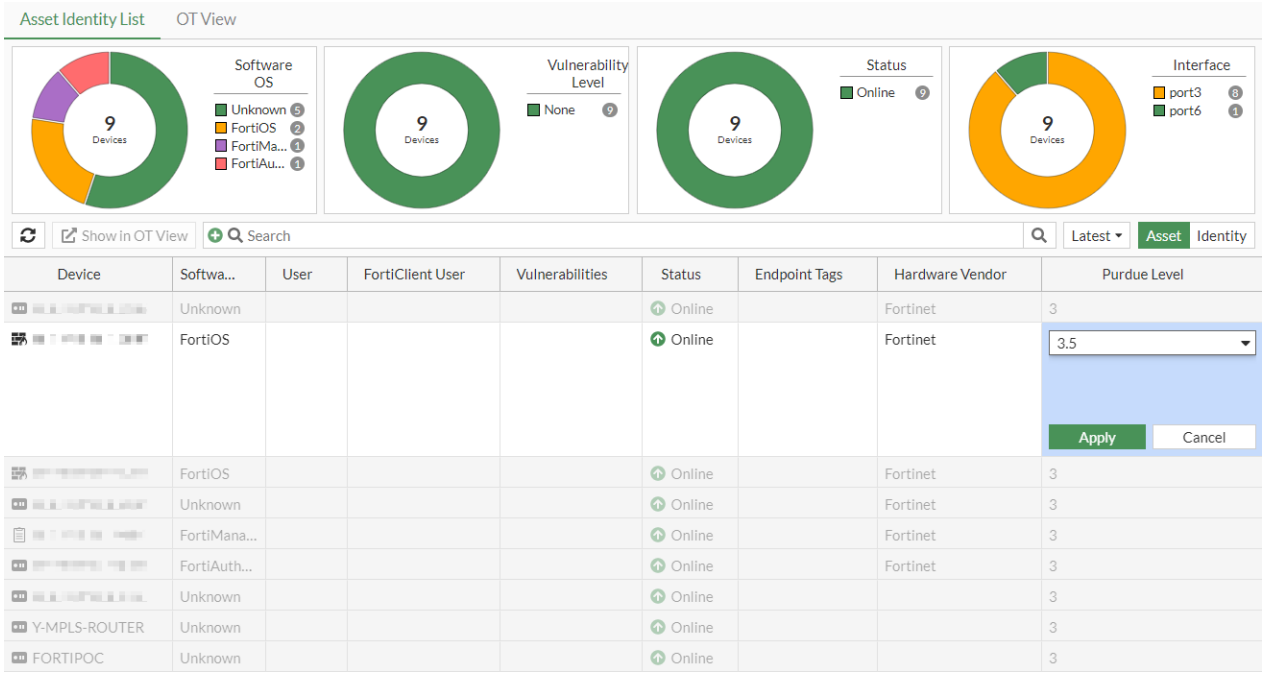
Devices are assigned Purdue Level 3 by default and can be changed (except to level S, 0, or external), including FortiGates, managed FortiSwitches, and FortiAPs.

The following diagram lists the Purdue Levels based on OT network topologies:



To change the Purdue Level in the Asset Identity List tab:

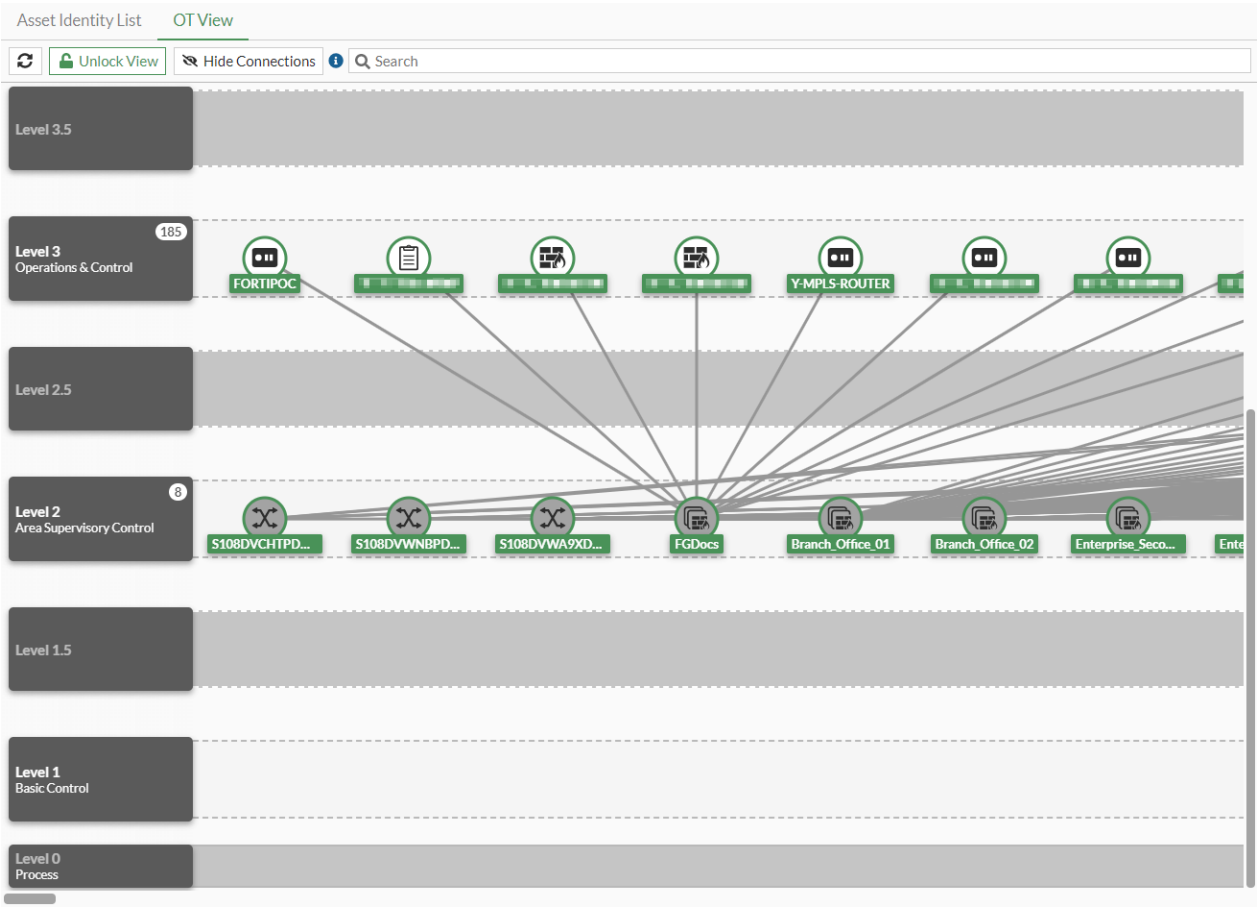
1. Go to *Security Fabric* > *Asset Identity Center* and select the *Asset Identity List* tab.
2. Add the *Purdue Level* column to the table:
   a. Hover over the table header and click the gear icon (*Configure Table*).
   b. Select *Purdue Level*.
   c. Click *Apply*.
3. Select a device and hover over the *Purdue Level* value.
4. Click the pencil icon to edit the level.
5. Select a value from the dropdown.



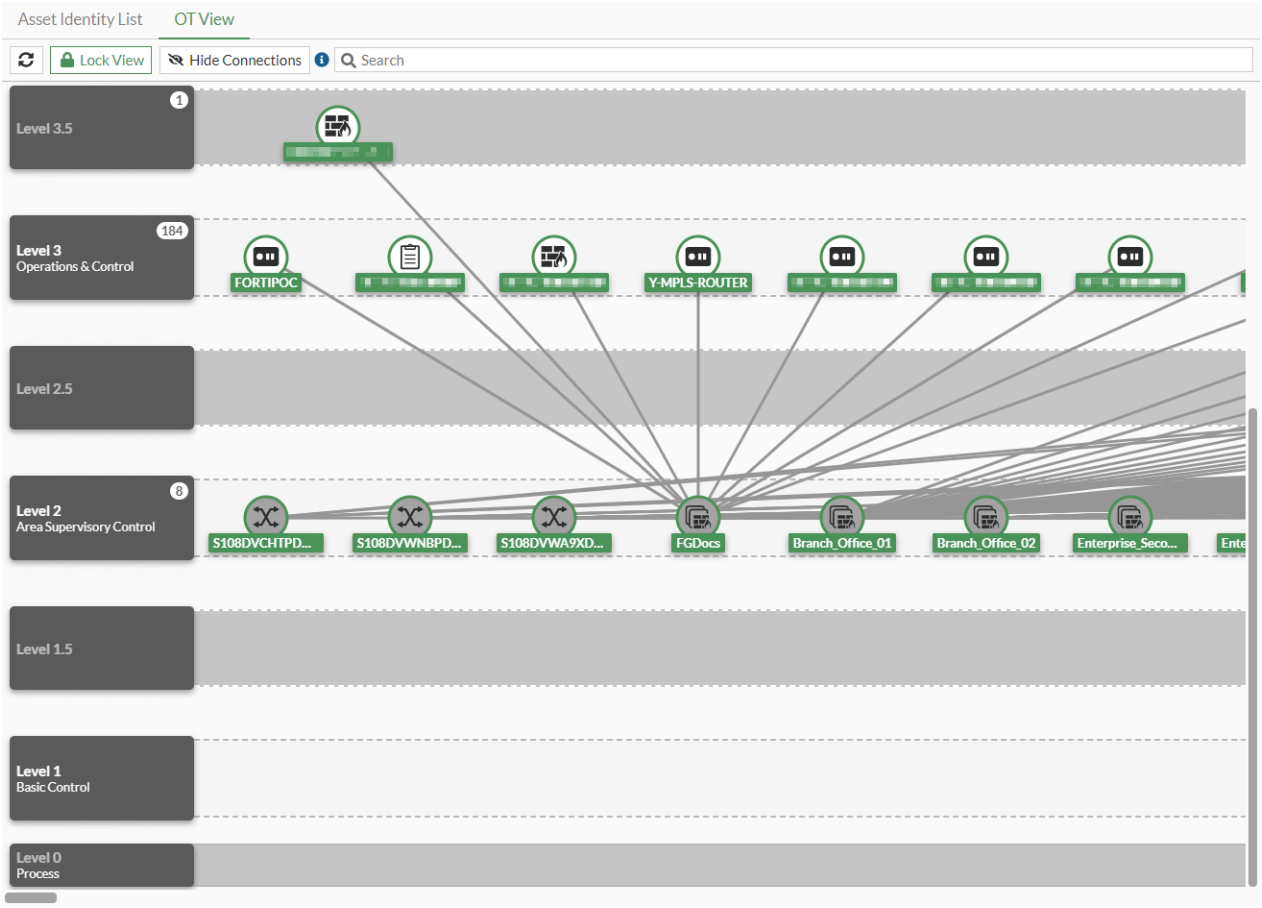
6. Click *Apply*.

To change the Purdue Level in the OT View tab:

1. Go to *Security Fabric* > *Asset Identity Center* and select the *OT View* tab.
2. Click *Unlock View*.



3. Select a device.
4. Drag the device icon to another level row.



5. Optionally, click *Lock View* to revert to the locked view.

To change the Purdue Level in the CLI:

```
# diagnose user-device-store device memory ot-purdue-set <mac> <ip> <level>
```

`mac`	Enter the MAC address of the device.
`ip`	Enter the IPv4 address of the device.
`level`	Enter the Purdue Level: 1, 1.5, 2, 2.5, 3, 3.5, 4, 5, 5.5.

To configure the FortiGate Purdue Level in the CLI:

```
config system global
    set purdue-level <level 1 - 5.5>
end
```

To configure the managed FortiSwitch Purdue Level in the CLI:

```
config switch-controller managed-switch
    edit "<managed FortiSwitch name>"
```

```
        set purdue-level <level 1 - 5.5>
    next
end
```

To configure the FortiAP Purdue Level in the CLI:

```
config wireless-controller wtp
    edit "<WTP ID>"
        set purdue-level <level 1 - 5.5>
    next
end
```

Configuring the Purdue Level for discovered assets based on detected interface

The default Purdue Level can be set or unset in the CLI (`default-purdue-level`) within the system interface configuration. The default Purdue Level can be applied to discovered assets based on the interface with which they were detected. This feature requires a FortiGuard Industrial Security Service (ISS) license on the FortiGate so the Industrial Database (ISDB) can be used. Device identification must be enabled on interfaces connected to OT devices.

```
config system interface
    edit <name>
        set device-identification enable
        set default-purdue-level {1 | 1.5| 2 | 2.5| 3 | 3.5 | 4 | 5 | 5.5}
    next
end
```

By default, the `default-purdue-level` value is `3`. If the asset's Purdue Level is manually overridden, then it takes precedence over this default value set in the interface.

For example, if the default Purdue Level on port1 is changed to 3.5, subsequently, the Purdue Level of a detected device on port1 is manually changed to 4 on the *Asset Identity Center* page. After the manual change on the device, the Purdue Level remains at 4.

To configure the default Purdue Level:

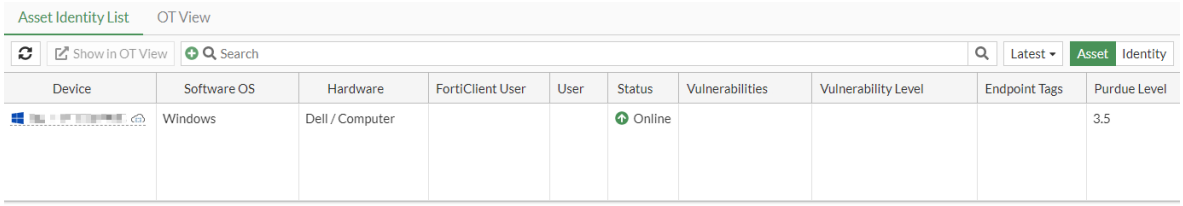
1. Configure the interface settings:

   ```
   config system interface
       edit "port1"
           set device-identification enable
           set default-purdue-level 3.5
       next
   end
   ```

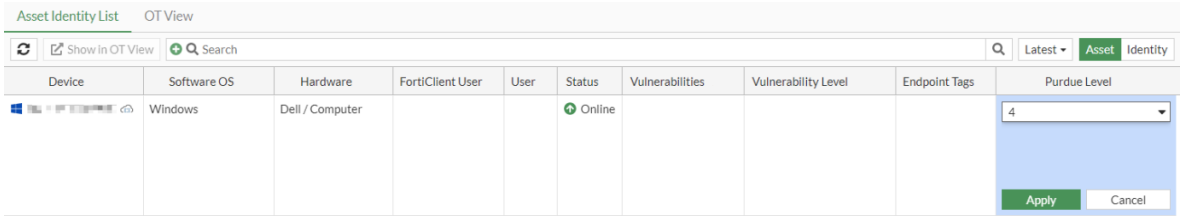
2. Verify that the Purdue Level as been updated in the user device store list:

```
# diagnose user-device-store device memory list

Record #1:

        device_info
                'ipv4_address' = '192.168.1.64'
                'mac' = '**:**:**:**:**:**'
                'hardware_vendor' = 'Dell'
                'hardware_type' = 'Home & Office'
                'hardware_family' = 'Computer'
                'vdom' = 'root'
                'os_name' = 'Windows'
                'os_version' = '10 / 2016'
                'last_seen' = '1680115135'
                'host_src' = 'mwbs'
                'unjoined_forticlient_endpoint' = 'false'
                'is_online' = 'true'
                'active_start_time' = '1680113976'
                'dhcp_lease_status' = 'leased'
                'dhcp_lease_expire' = '1680651757'
                'dhcp_lease_reserved' = 'false'
                'dhcp_server_id' = '2'
                'is_fortiguard_src' = 'true'
                'purdue_level' = '3.5'
                ...
```

3. Go to *Security Fabric > Asset Identity Center* and select the *Asset Identity List* tab. The device's *Purdue Level* is currently *3.5*.



4. Manually change the device's Purdue Level:
   a. Select the device and hover over the *Purdue Level* value.
   b. Click the pencil icon to edit the level.
   c. Select *4* and click *Apply*.



5. Verify that the Purdue Level as been updated in the user device store list:

```
# diagnose user-device-store device memory list

Record #1:
```

```
    device_info
            'ipv4_address' = '192.168.1.64'
            'mac' = '**:**:**:**:**:**'
            'hardware_vendor' = 'Dell'
            'hardware_type' = 'Home & Office'
            'hardware_family' = 'Computer'
            'vdom' = 'root'
            'os_name' = 'Windows'
            'os_version' = '10 / 2016'
            'last_seen' = '1680115467'
            'host_src' = 'mwbs'
            'unjoined_forticlient_endpoint' = 'false'
            'is_online' = 'true'
            'active_start_time' = '1680113976'
            'dhcp_lease_status' = 'leased'
            'dhcp_lease_expire' = '1680651757'
            'dhcp_lease_reserved' = 'false'
            'dhcp_server_id' = '2'
            'is_fortiguard_src' = 'true'
            'purdue_level' = '4'
            ...
```

KEV information in IoT/OT vulnerabilities

Known Exploited Vulnerabilities (KEVs) information is included in IoT/OT vulnerabilities in the user/device store. KEV counts and warnings are displayed in the *Assets* widget on the *Asset & Identities* dashboard, enhancing security visibility.

To enable device detection for IoT/OT devices, device identification must be enabled on the interface and utm-status must be enabled in firewall policy:

```
config system interface
    edit <interface>
        set device-identification enable
    next
end
config firewall policy
    edit <policy ID>
        set utm-status enable
    next
end
```

See also KEV information from FortiClient NEW on page 4416.

Example

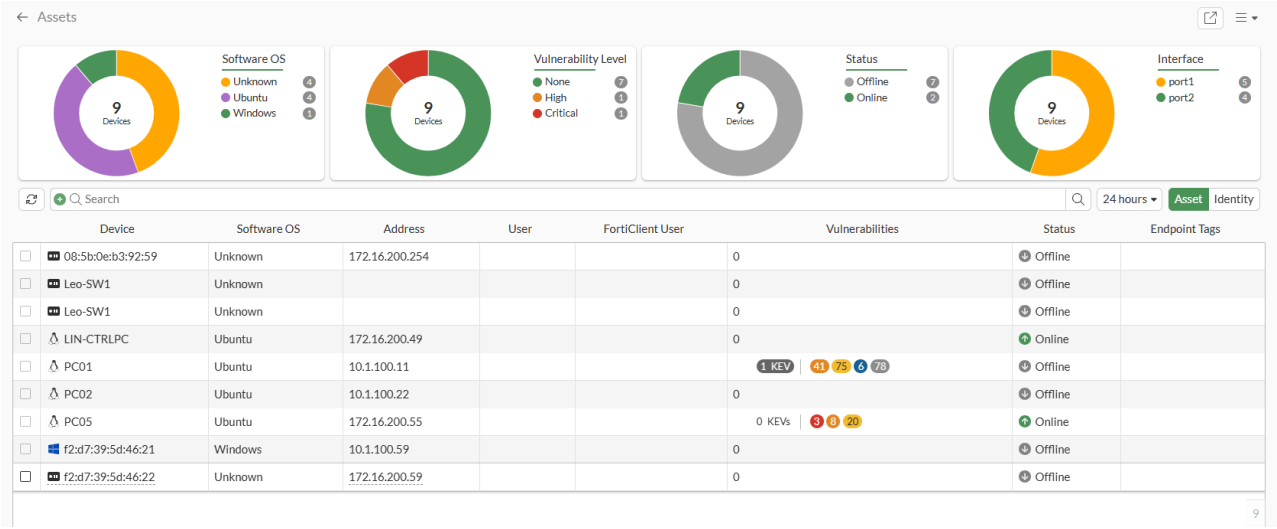
When Ubuntu is updated and upgraded from an older version, like 18.04, using the `sudo apt-get update && sudo apt-get -y upgrade` command, OT device information is logged in an appctrl log and sent to WAD:

```
1: date=2024-11-15 time=10:00:53 eventtime=1731621653528292983 tz="+1200" logid="1059028704"
type="utm" subtype="app-ctrl" eventtype="signature" level="information" vd="vd1" appid=10000501
srcip=10.1.100.11 srccountry="Reserved" dstip=91.189.91.81 dstcountry="United States"
srcport=37186 dstport=80 srcintf="port2" srcintfrole="undefined" dstintf="port1"
dstintfrole="undefined" proto=6 service="HTTP" direction="outgoing" policyid=1 poluuid="b8a98718-
dfc9-51ee-3aff-53c8c1b65d82" policytype="policy" sessionid=1469 applist="g-default" action="pass"
appcat="OT" app="Canonical.Ubuntu" hostname="ca.archive.ubuntu.com" incidentserialno=170919395
url="/ubuntu/dists/bionic/InRelease" agent="Debian APT-HTTP/1.3 (1.6.17)" httpmethod="GET"
msg="OT: Canonical.Ubuntu" clouddevice="Vendor=Canonical, Product=Ubuntu, Version=18.04"
apprisk="low"
```

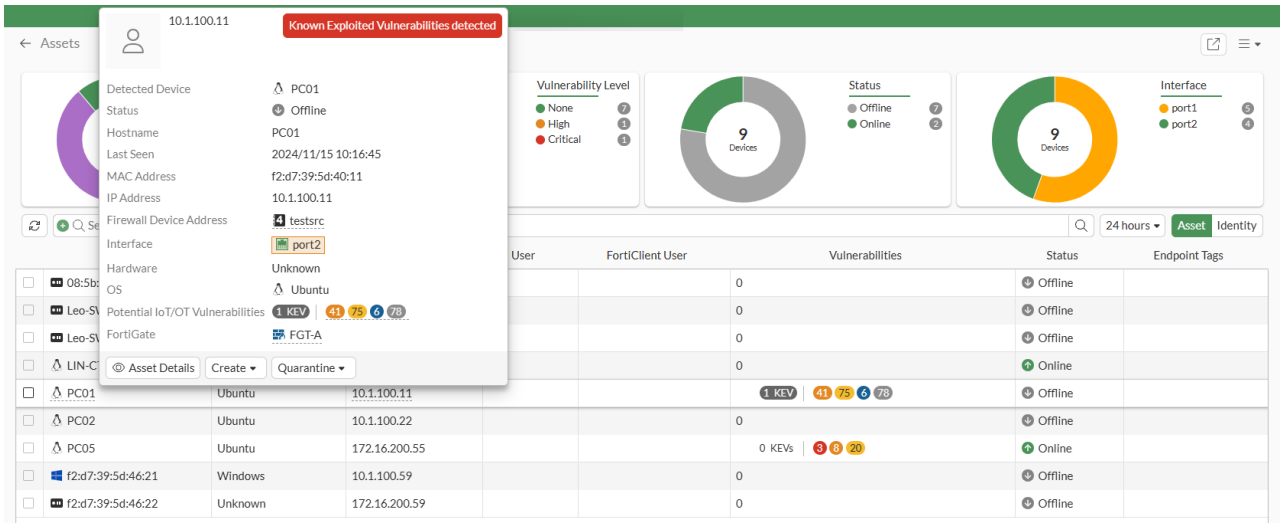
WAD then sends a device query to FortiGuard to retrieve the vulnerability list for this OT device. The KEV information is included in the vulnerability list and can be viewed in the GUI and CLI.

To view the KEV information in the GUI:

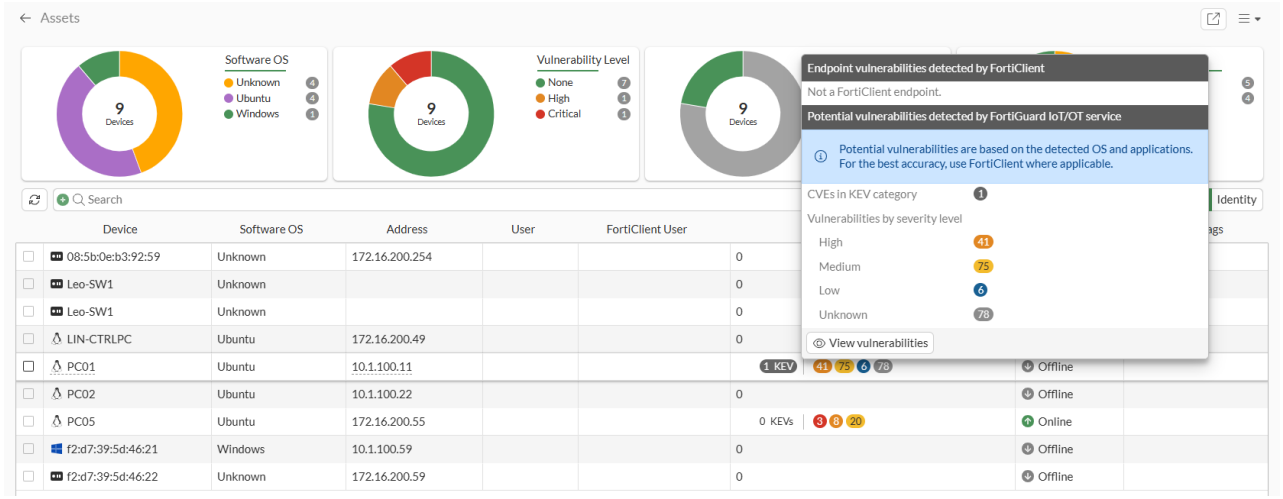
1. Go to *Dashboard* > *Assets & Identities*.
2. Expand the *Assets* widget. The KEV vulnerabilities are shown in the *Vulnerabilities* column:



3. Hover over the device name. In the tooltip, a *Known Exploited Vulnerabilities detected* label is added when KEVs are detected for IoT/OT:

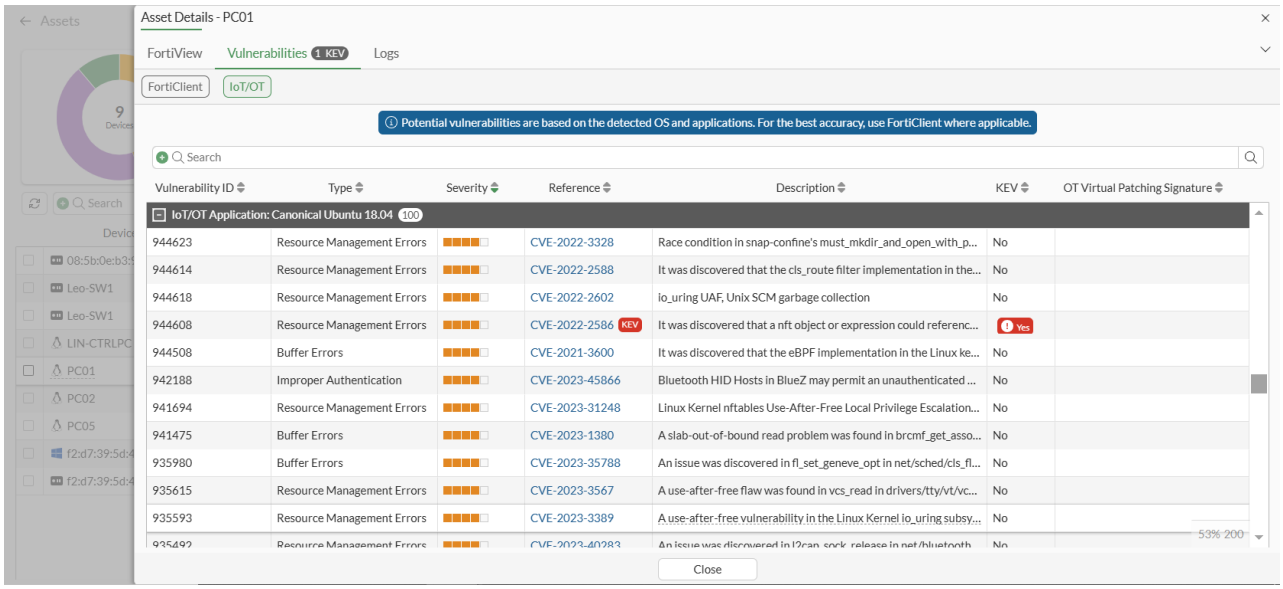


4. Hover over the vulnerabilities. In the tooltip, the *CVEs in KEV category* field shows the number of potential KEVs detected by the FortiGuard IoT/OT service:



5. In the tooltip, click *View vulnerabilities* to open the *Asset Details* pane on the *Vulnerabilities* tab and select *IoT/OT*.

   A *KEV* label is shown as a critical severity label next to the CVE-ID in the *Reference* column, and a *KEV* column can be added to show if a vulnerability has a KEV.



To view the KEV information in the CLI:

```
# diagnose user-device-store device memory list

Record #1:
        device_info
                'ipv4_address' = '10.1.100.11'
                'mac' = 'f2:d7:39:5d:40:11'
                'hardware_type' = 'Unknown'
                'vdom' = 'vd1'
                'os_name' = 'Ubuntu'
                'hostname' = 'PC01'
                'last_seen' = '1731621105'
                'host_src' = 'mwbs'
                'unjoined_forticlient_endpoint' = 'false'
                'is_online' = 'false'
                'active_start_time' = '1731612432'
                'is_fortiguard_src' = 'false'
                'purdue_level' = '3'
                'iot_vuln_count' = '200'
                'iot_kev_count' = '1'
                'max_vuln_level' = 'High'
                'total_vuln_count' = '200'
                'device_type' = 'OT'
                'generation' = '16'
        interface_info
                'ipv4_address' = '10.1.100.11'
                'mac' = 'f2:d7:39:5d:40:11'
                'master_mac' = 'f2:d7:39:5d:40:11'
                'detected_interface' = 'port2'
                'last_seen' = '1731621105'
                'is_master_device' = 'true'
                'is_detected_interface_role_wan' = 'false'
```

```
                'detected_interface_fortitelemetry' = 'false'
                'is_online' = 'false'
                'is_fortiguard_src' = 'false'
        iot_info
                'vendor' = 'Canonical'
                'product' = 'Ubuntu'
                'version-min' = '18.04'
                'validity' = 'true'
                'outdated' = 'false'
                'db_date_updated' = '2024-11-14T20:40:07'
                'kev_db_date_released' = '2024-11-14T18:00:41'
        iot_vulnerability
                'vulnerability_id' = '944608'
                'severity' = '3'
                'type' = 'Resource Management Errors'
                'description' = 'It was discovered that a nft object or expression could reference
a nft set on a different nft table, leading to a use-after-free once that table was deleted.'
                'references' = 'CVE-2022-2586'
                'kevs' = 'CVE-2022-2586'
                'date_added' = '2024-01-15T09:00:30'
                'date_updated' = '2024-01-15T09:00:30'
```

KEV information from FortiClient NEW

FortiGate detects and displays CISA Known Exploited Vulnerabilities (KEV) for FortiClient devices managed by FortiClient EMS in the *Security Fabric > Asset Identity Center* and the *Dashboard > Assets & Identities > Assets* widget in the GUI. Drilling down displays vulnerability details.

When FortiClient endpoints managed by FortiClient EMS run vulnerability scans and detect vulnerabilities, this information is synchronized to EMS and subsequently to FortiGate. When FortiGate receives detected vulnerabilities, it looks up the vulnerabilities against the KEV database and flags them accordingly. As a result, the FortiOS GUI can display the number of KEVs found on an endpoint and provide details about the vulnerabilities. The security rating check for FortiClient vulnerabilities also fails when a KEV vulnerability is detected.

See also KEV information in IoT/OT vulnerabilities on page 4412.

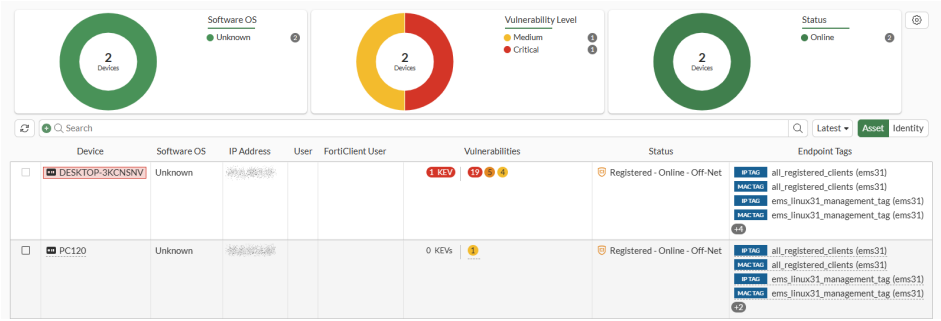
A new diagnose command is available:

```
# diagnose user-device-store device memory FortiClient-vulnerability-query <mac> <ip>
```

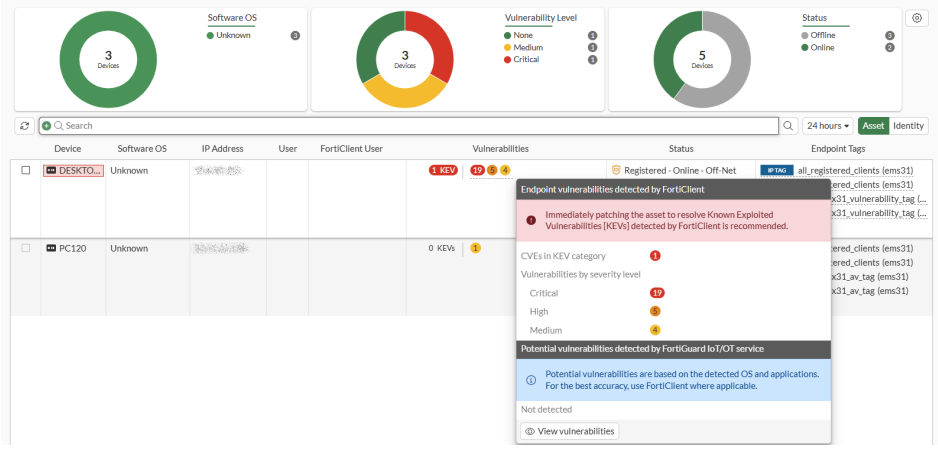
Option	Description
`diagnose user-device-store device memory FortiClient-vulnerability-query <mac> <ip>`	Query FortiClient vulnerability information by MAC and IP address. Takes 0-3 argument(s).

To view information in Security Fabric > Asset Identity Center:

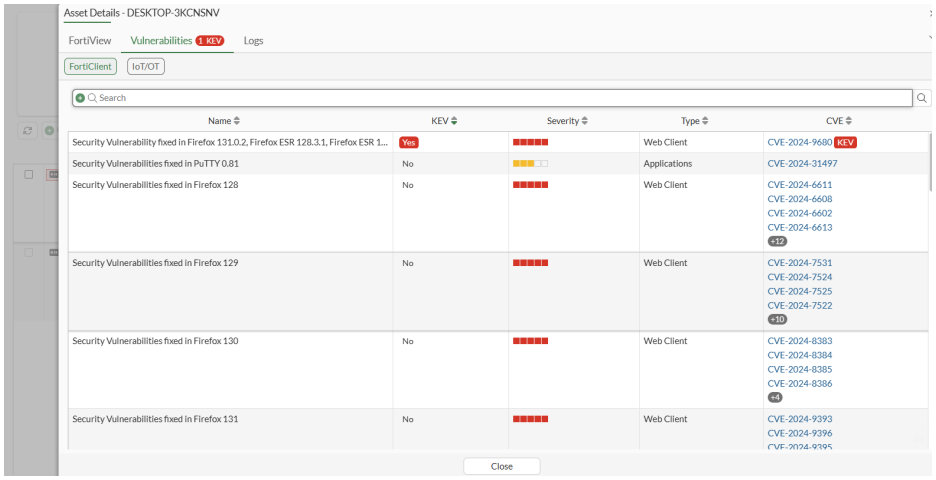
1. Go to *Security Fabric > Asset Identity Center* and view the *Vulnerabilities* column for KEV information detected by FortiClient.



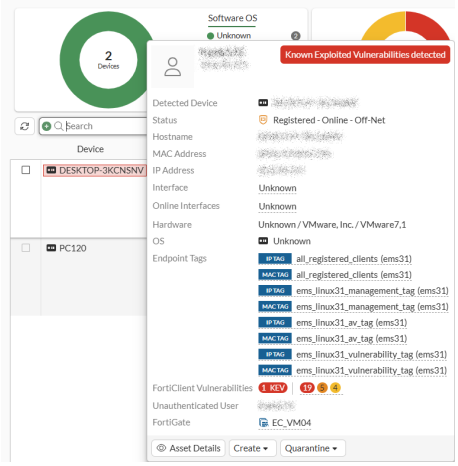
2. In the *Vulnerabilities* column, hover over a vulnerability to display a tooltip to view the KEV category and vulnerabilities by severity level.



3. Click *View vulnerabilities* to display additional information.

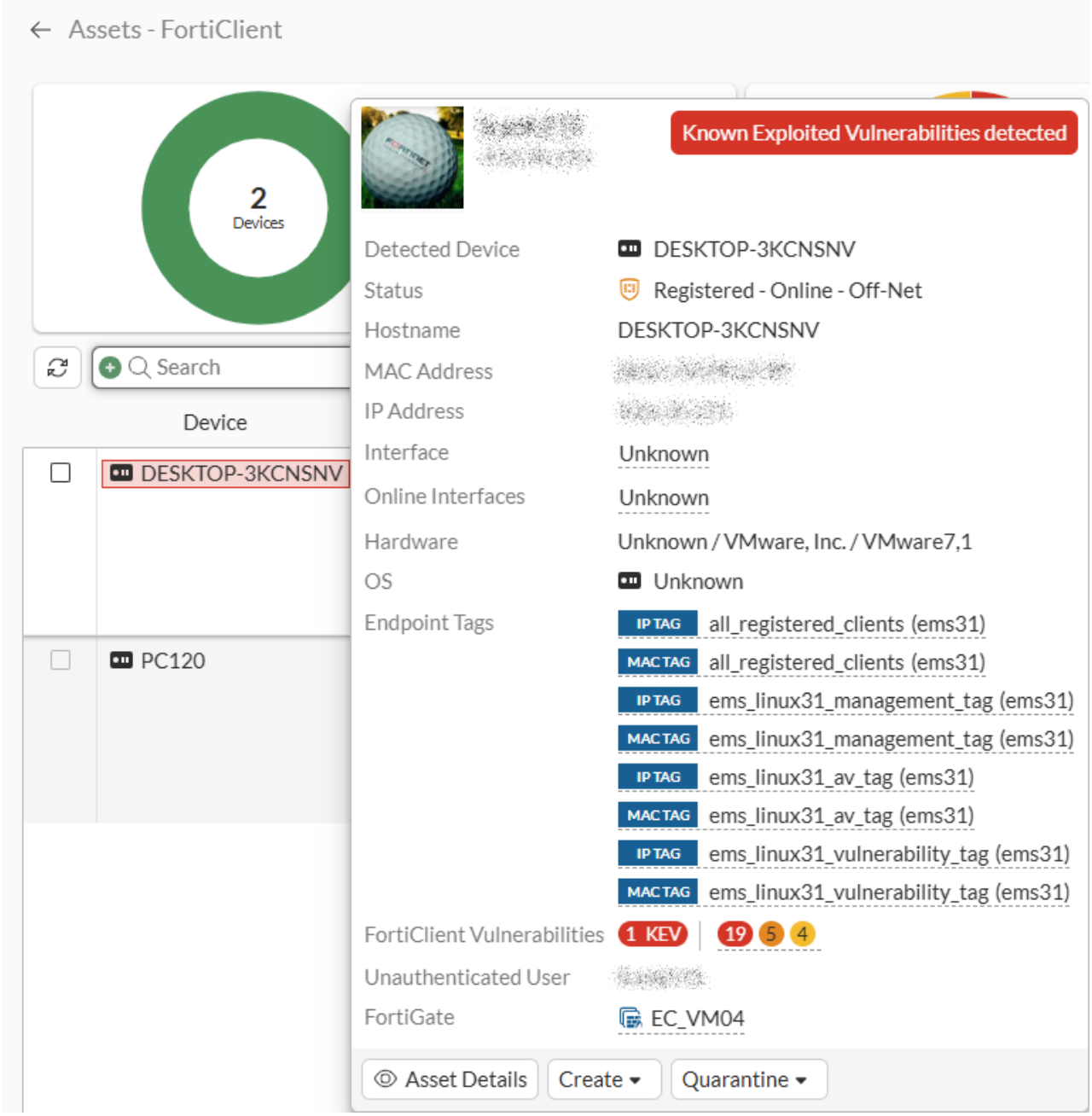


4. Click *Close*.
5. In the *Device* column, hover over a device name to display a tooltip of information about known, detected exploited vulnerabilities.

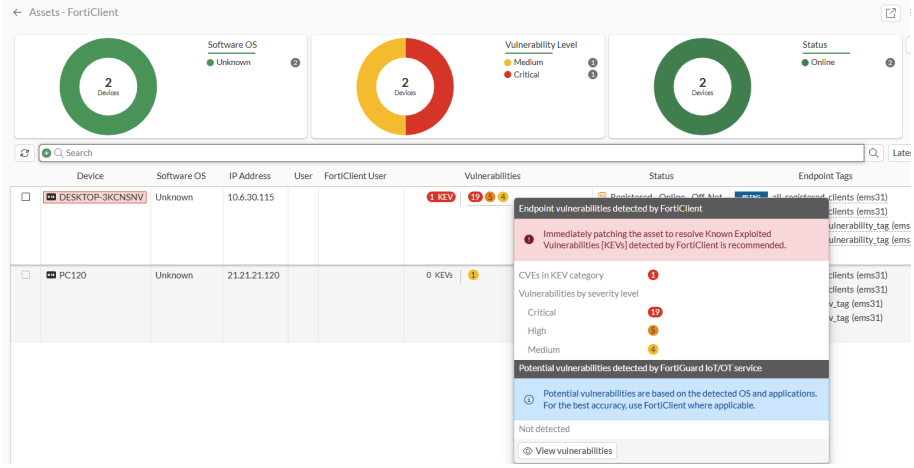


To view information in Dashboard > Assets & Identities:

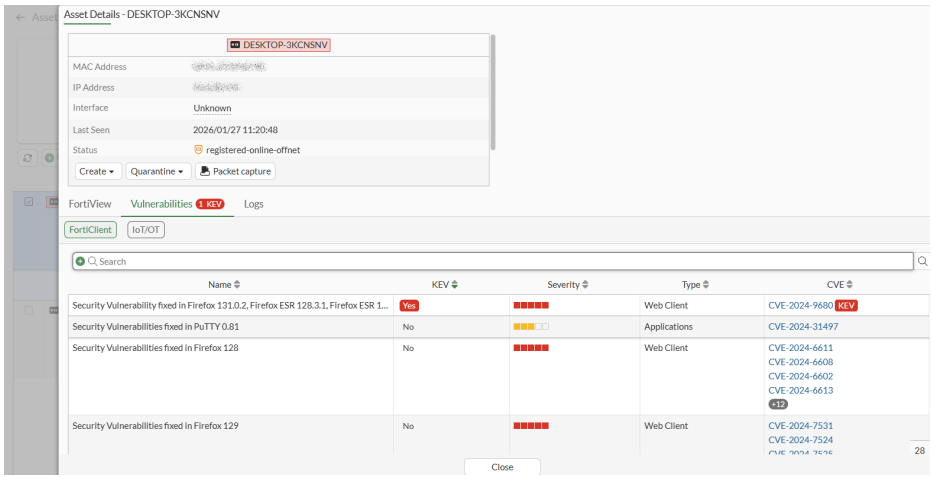
1. Go *to Dashboard > Assets & Identities > Assets - FortiClient* widget.
2. In the *Device* column, hover over a device name to display a tooltip of information about known, detected exploited vulnerabilities.



3. In the *Vulnerabilities* column, hover over a vulnerability to display a tooltip to view the KEV category and vulnerabilities by severity level.



4. Click *View vulnerabilities* to display additional information.



5. Click *Close*.

To query FortiClient vulnerability information by MAC and IP address:

```
# diagnose user-device-store device memory FortiClient-vulnerability-query <mac address> <IP
address>Record #1:

        fct_info
                'scan_time' = '1769477815'
        fct_vulnerability
                'vuln_id' = '5685'
                'severity' = '2'
                'category' = 'Applications'
                'vuln_name' = 'Security Vulnerabilities fixed in PuTTY 0.81'
                'cve_ids' = 'CVE-2024-31497'
        fct_vulnerability
                'vuln_id' = '80382'
                'severity' = '4'
                'category' = 'Web Client'
                'vuln_name' = 'Security Vulnerabilities fixed in Firefox 128'
                'cve_ids' = 'CVE-2024-6611'
```

```
                'cve_ids' = 'CVE-2024-6608'
                'cve_ids' = 'CVE-2024-6602'
                'cve_ids' = 'CVE-2024-6613'
                'cve_ids' = 'CVE-2024-6604'
                'cve_ids' = 'CVE-2024-6603'
                'cve_ids' = 'CVE-2024-6615'
                'cve_ids' = 'CVE-2024-6612'
                'cve_ids' = 'CVE-2024-6600'
                'cve_ids' = 'CVE-2024-6601'
                'cve_ids' = 'CVE-2024-6614'
                'cve_ids' = 'CVE-2024-6607'
                'cve_ids' = 'CVE-2024-6610'
                'cve_ids' = 'CVE-2024-6606'
                'cve_ids' = 'CVE-2024-6609'
                'cve_ids' = 'CVE-2024-6605'
        fct_vulnerability
                'vuln_id' = '80711'
                'severity' = '4'
                'category' = 'Web Client'
                'vuln_name' = 'Security Vulnerabilities fixed in Firefox 129'
                'cve_ids' = 'CVE-2024-7531'
                'cve_ids' = 'CVE-2024-7524'
                'cve_ids' = 'CVE-2024-7525'
                'cve_ids' = 'CVE-2024-7522'
                'cve_ids' = 'CVE-2024-7529'
                'cve_ids' = 'CVE-2024-7518'
                'cve_ids' = 'CVE-2024-7528'
                'cve_ids' = 'CVE-2024-7521'
                'cve_ids' = 'CVE-2024-7519'
                'cve_ids' = 'CVE-2024-7526'
                'cve_ids' = 'CVE-2024-7520'
                'cve_ids' = 'CVE-2024-7527'
                'cve_ids' = 'CVE-2024-7530'
                'cve_ids' = 'CVE-2024-7523'
        fct_vulnerability
                'vuln_id' = '81006'
                'severity' = '4'
                'category' = 'Web Client'
                'vuln_name' = 'Security Vulnerabilities fixed in Firefox 130'
                'cve_ids' = 'CVE-2024-8383'
                'cve_ids' = 'CVE-2024-8384'
                'cve_ids' = 'CVE-2024-8385'
                'cve_ids' = 'CVE-2024-8386'
                'cve_ids' = 'CVE-2024-8389'
                'cve_ids' = 'CVE-2024-8382'
                'cve_ids' = 'CVE-2024-8387'
                'cve_ids' = 'CVE-2024-8381'
        fct_vulnerability
                'vuln_id' = '81485'
                'severity' = '4'
                'category' = 'Web Client'
                'vuln_name' = 'Security Vulnerabilities fixed in Firefox 131'
```

```
                'cve_ids' = 'CVE-2024-9393'
                'cve_ids' = 'CVE-2024-9396'
                'cve_ids' = 'CVE-2024-9395'
                'cve_ids' = 'CVE-2024-9400'
                'cve_ids' = 'CVE-2024-9401'
                'cve_ids' = 'CVE-2024-9402'
                'cve_ids' = 'CVE-2024-9403'
                'cve_ids' = 'CVE-2024-9394'
                'cve_ids' = 'CVE-2024-9398'
                'cve_ids' = 'CVE-2024-9399'
                'cve_ids' = 'CVE-2024-9392'
                'cve_ids' = 'CVE-2024-9397'
                'cve_ids' = 'CVE-2024-9391'
        fct_vulnerability
                'vuln_id' = '81654'
                'severity' = '4'
                'category' = 'Web Client'
                'vuln_name' = 'Security Vulnerability fixed in Firefox 131.0.2, Firefox ESR
128.3.1, Firefox ESR 115.16.1'
                'cve_ids' = 'CVE-2024-9680'
                'kevs' = 'CVE-2024-9680'
        fct_vulnerability
                'vuln_id' = '81684'
                'severity' = '2'
                'category' = 'Web Client'
                'vuln_name' = 'Security Vulnerability fixed in Firefox 131.0.3'
                'cve_ids' = 'CVE-2024-9936'
```

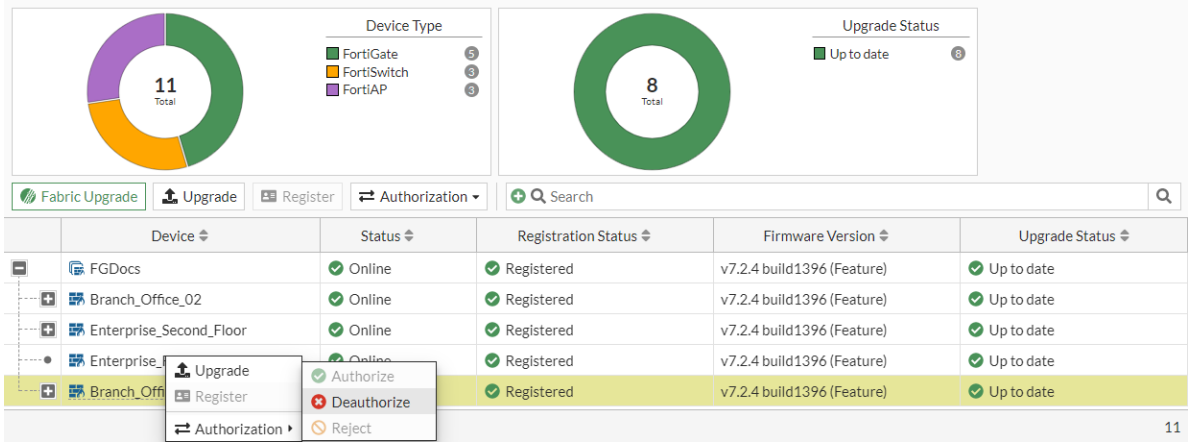
WebSocket for Security Fabric events

With the WebSocket for Security Fabric events, subscribers to the WebSocket (such as the *Firmware & Registration* page) are updated upon new Fabric events and alert users to reload the page.

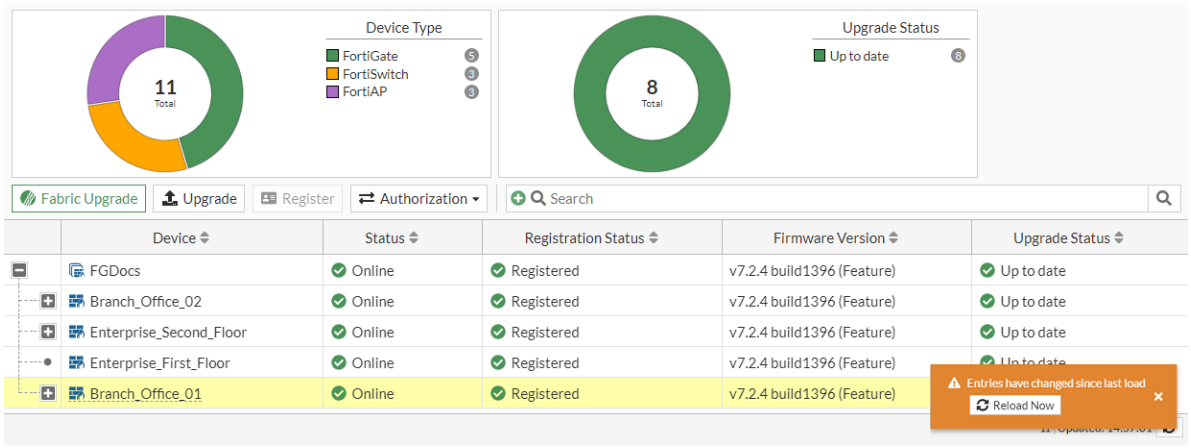
Example

To deauthorize a downstream FortiGate:

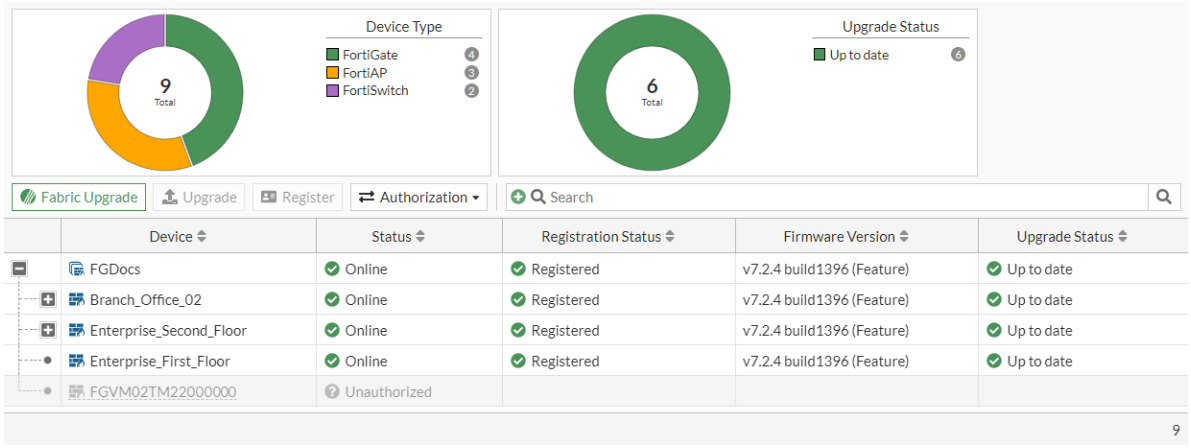
1. Go to *System* > *Firmware & Registration* and select a downstream FortiGate in the table.
2. Right-click on the device and select *Authorization* > *Deauthorize*.



3. An alert appears in the bottom-right corner of the page. Click *Reload Now* to refresh the page.



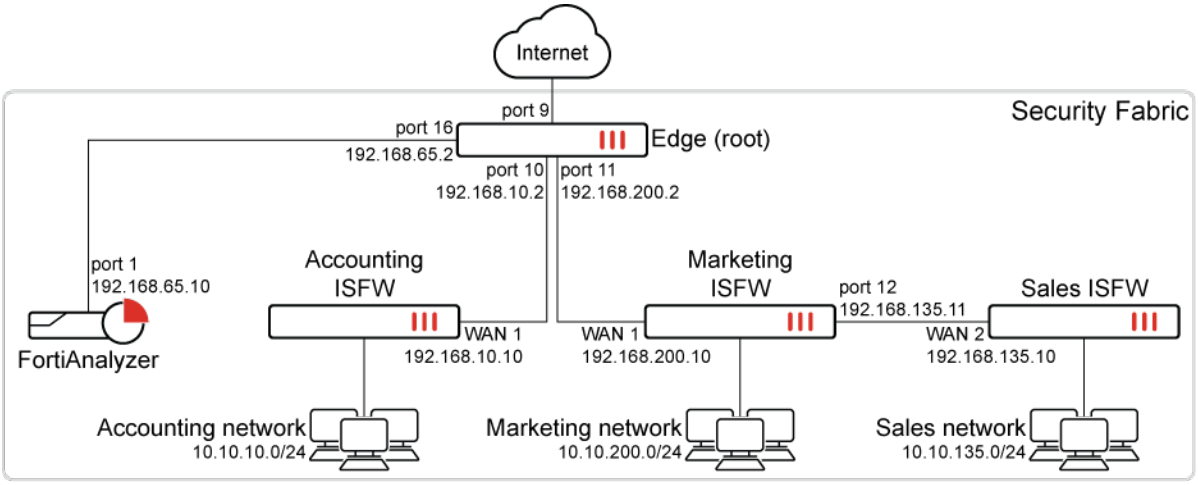
The deauthorized FortiGate's status is now listed as *Unauthorized*.



Deploying the Security Fabric

This topic provides an example of deploying Security Fabric with three downstream FortiGates connecting to one root FortiGate. To deploy a Security Fabric, you need a FortiAnalyzer running firmware version 6.2 or later.

The following shows a sample network topology with three downstream FortiGates (Accounting, Marketing, and Sales) connected to the root FortiGate (Edge).



To configure the root FortiGate (Edge):

1. Configure the interfaces:
   a. Go to *Network* > *Interfaces*.
   b. Edit *port16*:
      - Set *Role* to *DMZ*.
      - For the interface connected to FortiAnalyzer, set the *IP/Network Mask* to *192.168.65.2/255.255.255.0*
   c. Edit *port10*:
      - Set *Role* to *LAN*.
      - For the interface connected to the downstream FortiGate (Accounting), set the *IP/Network Mask* to *192.168.10.2/255.255.255.0*
   d. Edit *port11*:
      - Set *Role* to *LAN*.
      - For the interface connected to the downstream FortiGate (Marketing), set the *IP/Network Mask* to *192.168.200.2/255.255.255.0*
2. Configure the Security Fabric settings:
   a. Go to *Security Fabric* > *Fabric Connectors* and double-click the *Security Fabric Setup* card.
   b. Select the *Settings* tab, and set the *Security Fabric role* to *Serve as Fabric Root*.
   c. Enter a *Fabric name*, such as *Office-Security-Fabric*.
   d. Ensure *Allow other Security Fabric devices to join* is enabled and add *port10* and *port11*.
   e. Click *OK*.
3. Configure the FortiAnalyzer logging settings:
   a. Go to *Security Fabric* > *Fabric Connectors* and double-click the *Logging & Analytics* card.
   b. Select the *Settings* tab, select the *FortiAnalyzer* tab, and set the *Status* to *Enabled*.
   c. Enter the FortiAnalyzer IP in the *Server* field (*192.168.65.10*). The *Upload option* is automatically set to *Real Time*.
   d. Click *Refresh*.

A warning message indicates that the FortiGate is not authorized on the FortiAnalyzer. The authorization is configured in a later step on the FortiAnalyzer.

e. Click *OK*. The FortiAnalyzer serial number is verified.

4. Create the address objects to use in the firewall policies:
   a. Go to *Policy & Objects > Addresses*.
   b. Click *Create New*.
   - Set *Name* to *FAZ-addr*.
   - Set *Type* to *Subnet*.
   - Set *Subnet/IP Range* to *192.168.65.10/32*.
   - Set *Interface* to *any*.

   c. Click *OK*.
   d. Click *Create New*.
   - Set *Name* to *Accounting*.
   - Set *Type* to *Subnet*.
   - Set *Subnet/IP Range* to *192.168.10.10/32*.
   - Set *Interface* to *any*.

   e. Click *OK*.
5. Create a policy to allow the downstream FortiGate (Accounting) to access the FortiAnalyzer:
   a. Go to *Policy & Objects > Firewall Policy* and click *Create New*.
   - Set the *Name* to *Accounting-to-FAZ*.
   - Set *srcintf* to *port10*.
   - Set *dstintf* to *port16*.
   - Set *srcaddr* to *Accounting-addr*.
   - Set *dstaddr* to *FAZ-addr*.
   - Set *Action* to *Accept*.
   - Set *Schedule* to *Always*.
   - Set *Service* to *All*.
   - Enable *NAT*.
   - Set *IP Pool Configuration* to *Use Outgoing Interface Address*.

   b. Click *OK*.
6. Create a policy to allow the two downstream FortiGates (Marketing and Sales) to access the FortiAnalyzer:
   a. In the root FortiGate (Edge), go to *Policy & Objects > Addresses* and click *Create New*.
   - Set *Name* to *Marketing-addr*.
   - Set *Type* to *Subnet*.
   - Set *Subnet/IP Range* to *192.168.200.10/32*.
   - Set *Interface* to *any*.

   b. Click *OK*.
   c. In the root FortiGate (Edge), go to *Policy & Objects > Firewall Policy* and click *Create New*.
   - Set *Name* to *Marketing-to-FAZ*.
   - Set *srcintf* to *port11*.
   - Set *dstintf* to *port16*.
   - Set *srcaddr* to *Marketing-addr*.

- Set *dstaddr* to *FAZ-addr*.
- Set *Action* to *Accept*.
- Set *Schedule* to *Always*.
- Set *Service* to *All*.
- Enable *NAT*.
- Set *IP Pool Configuration* to *Use Outgoing Interface Address*.

d. Click *OK*.

To configure the downstream FortiGate (Accounting):

1. Configure the interface:
   a. Go to *Network > Interfaces*.
   b. Edit interface *wan1*:
   - Set *Role* to *WAN*.
   - For the interface connected to root, set the *IP/Network Mask* to *192.168.10.10/255.255.255.0*
2. Configure the default static route to connect to the root FortiGate (Edge):
   a. Go to *Network > Static Routes* and click *Create New* or *Create New > IPv4 Static Route*.
   - Set *Destination* to *0.0.0.0/0.0.0.0*.
   - Set *Interface* to *wan1*.
   - Set *Gateway Address* to *192.168.10.2*.

   b. Click *OK*.
3. Configure the Security Fabric:
   a. Go to *Security Fabric > Fabric Connectors* and double-click the *Security Fabric Setup* card.
   b. In the *Settings* tab, set the *Security Fabric role* to *Join Existing Fabric*.
   FortiAnalyzer automatically enables logging. Settings for the FortiAnalyzer are retrieved from the root FortiGate (Edge) when FortiGate (Accounting) connects to the root FortiGate (Edge).
   c. *Upstream FortiGate IP* is filled in automatically with the default static route *Gateway Address* of *192.168.10.2* set in the previous step.
   d. Disable *Allow other Security Fabric devices to join*, because there is no downstream FortiGate connecting to it.
   e. Click *OK*.

To configure the downstream FortiGate (Marketing):

1. Configure the interface:
   a. Go to *Network > Interfaces*.
   b. Edit *port12*:
   - Set *Role* to *LAN*.
   - For the interface connected to the downstream FortiGate (Sales), set the *IP/Network Mask* to *192.168.135.11/255.255.255.0*.

   c. Edit *wan1*:
   - Set *Role* to *WAN*.
   - For the interface connected to the root FortiGate (Edge), set the *IP/Network Mask* to *192.168.200.10/255.255.255.0*.
2. Configure the default static route to connect to the root FortiGate (Edge):

   a. Go to *Network > Static Routes* and click *Create New* or *Create New > IPv4 Static Route*.
      - Set *Destination* to *0.0.0.0/0.0.0.0*.
      - Set *Interface* to *wan1*.
      - Set *Gateway Address* to *192.168.200.2*.
   b. Click *OK*.
3. Configure the Security Fabric:
   a. Go to *Security Fabric > Fabric Connectors* and double-click the *Security Fabric Setup* card.
   b. In the *Settings* tab, set the *Security Fabric role* to *Join Existing Fabric*.
      FortiAnalyzer automatically enables logging. Settings for the FortiAnalyzer are retrieved from the root FortiGate (Edge) when FortiGate (Marketing) connects to the root FortiGate (Edge).
   c. *Upstream FortiGate IP* is filled in automatically with the default static route *Gateway Address* of *192.168.200.2* set in the previous step.
   d. Enable *Allow other Security Fabric devices to join* and add *port12*.
   e. Click *OK*.
4. Create the address objects to use in the firewall policies:
   a. Go to *Policy & Objects > Addresses* and click *Create New*.
      - Set *Name* to *FAZ-addr*.
      - Set *Type* to *Subnet*.
      - Set *Subnet/IP Range* to *192.168.65.10/32*.
      - Set *Interface* to *any*.
   b. Click *OK*.
   c. Click *Create New*.
      - Set *Name* to *Sales-addr*.
      - Set *Type* to *Subnet*.
      - Set *Subnet/IP Range* to *192.168.135.10/32*.
      - Set *Interface* to *any*.
   d. Click *OK*.
5. Create a policy to allow another downstream FortiGate (Sales) going through FortiGate (Marketing) to access the FortiAnalyzer:
   a. Go to *Policy & Objects > Firewall Policy* and click *Create New*.
      - Set *Name* to *Sales-to-FAZ*.
      - Set *srcintf* to *port12*.
      - Set *dstintf* to *wan1*.
      - Set *srcaddr* to *Sales-addr*.
      - Set *dstaddr* to *FAZ-addr*.
      - Set *Action* to *Accept*.
      - Set *Schedule* to *Always*.
      - Set *Service* to *All*.
      - Enable *NAT*.
      - Set *IP Pool Configuration* to *Use Outgoing Interface Address*.
   b. Click *OK*.

To configure the downstream FortiGate (Accounting):

1. Configure the interface:
   a. Go to *Network > Interfaces*.
   b. Edit interface *wan1*:
      - Set *Role* to *WAN*.
      - For the interface connected to root, set the *IP/Network Mask* to *192.168.10.10/255.255.255.0*
2. Configure the default static route to connect to the root FortiGate (Edge):
   a. Go to *Network > Static Routes* and click *Create New* or *Create New > IPv4 Static Route*.
      - Set *Destination* to *0.0.0.0/0.0.0.0*.
      - Set *Interface* to *wan1*.
      - Set *Gateway Address* to *192.168.10.2*.
   b. Click *OK*.
3. Configure the Security Fabric:
   a. Go to *Security Fabric > Fabric Connectors* and double-click the *Security Fabric Setup* card.
   b. In the *Settings* tab, set the *Security Fabric role* to *Join Existing Fabric*.
      FortiAnalyzer automatically enables logging. Settings for the FortiAnalyzer are retrieved from the root FortiGate (Edge) when FortiGate (Accounting) connects to the root FortiGate (Edge).
   c. *Upstream FortiGate IP* is filled in automatically with the default static route *Gateway Address* of *192.168.10.2* set in the previous step.
   d. Disable *Allow other Security Fabric devices to join*, because there is no downstream FortiGate connecting to it.
   e. Click *OK*.

To configure the downstream FortiGate (Sales):

1. Configure the interface:
   a. Go to *Network > Interfaces*.
   b. Edit *wan2*:
      - Set *Role* to *WAN*.
      - For the interface connected to the upstream FortiGate (Marketing), set the *IP/Network Mask* to *192.168.135.10/255.255.255.0*.
2. Configure the default static route to connect to the upstream FortiGate (Marketing):
   a. Go to *Network > Static Routes* and click *Create New* or *Create New > IPv4 Static Route*.
      - Set *Destination* to *0.0.0.0/0.0.0.0*.
      - Set *Interface* to *wan2*.
      - Set *Gateway Address* to *192.168.135.11*.
   b. Click *OK*.
3. Configure the Security Fabric:
   a. Go to *Security Fabric > Fabric Connectors* and double-click the *Security Fabric Setup* card.
   b. In the *Settings* tab, set the *Security Fabric role* to *Join Existing Fabric*.
      FortiAnalyzer automatically enables logging. Settings for the FortiAnalyzer are retrieved from the root FortiGate (Edge) when FortiGate (Sales) connects to the root FortiGate (Edge).

c. *Upstream FortiGate IP* is filled in automatically with the default static route *Gateway Address* of *192.168.135.11* set in the previous step.
d. Disable *Allow other Security Fabric devices to join*, because there is no downstream FortiGate connecting to it.
e. Click *OK*.

To authorize downstream FortiGates (Accounting, Marketing, and Sales) on the root FortiGate (Edge):

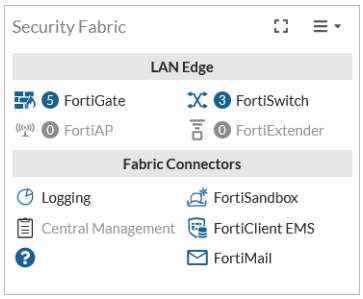
1. In the root FortiGate (Edge), go to *System > Firmware & Registration*.
   The table highlights two connected FortiGates with their serial numbers that are unauthorized.
2. Select the unauthorized device and click *Authorization > Authorize*.
   After they are authorized, the two downstream FortiGates (Accounting and Marketing) appear in the *Security Fabric* widget. This means that the two downstream FortiGates (Accounting and Marketing) have successfully joined the Security Fabric.
3. The table now highlights the Sales FortiGate with the serial number that is connected to the downstream Marketing FortiGate that is unauthorized.
4. Select the highlighted FortiGate and click *Authorization > Authorize*.
   After it is authorized, the downstream FortiGate (Sales) appears in the *Topology* tree in the *Security Fabric* widget. This means that the downstream FortiGates (Sales) has successfully joined the Security Fabric.

To use FortiAnalyzer to authorize all the Security Fabric FortiGates:

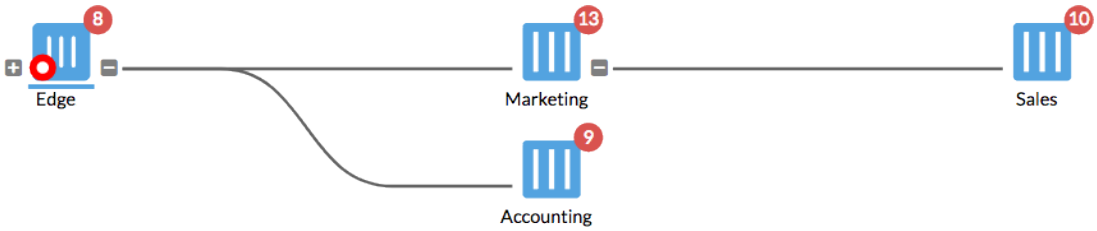
1. Authorize all the Security Fabric FortiGates on the FortiAnalyzer side:
   a. On the FortiAnalyzer, go to *System Settings > Network > All Interfaces*.
   b. Edit *port1* and set *IP Address/Netmask* to *192.168.65.10/255.255.255.0*.
   c. Go to *Device Manager > Unauthorized*. All of the FortiGates are listed as unauthorized.
      i. Select all the FortiGates and select *Authorize*. The FortiGates are now listed as authorized.
         After a moment, a warning icon appears beside the root FortiGate (Edge) because the FortiAnalyzer needs administrative access to the root FortiGate (Edge) in the Security Fabric.
      ii. Click the warning icon and enter the admin username and password of the root FortiGate (Edge).
2. Check FortiAnalyzer status on all the Security Fabric FortiGates:
   a. On each FortiGate, go to *Security Fabric > Fabric Connectors* and double-click the *FortiAnalyzer Logging* card.
   b. Check that *Storage usage* information is shown.

To check Security Fabric deployment result:

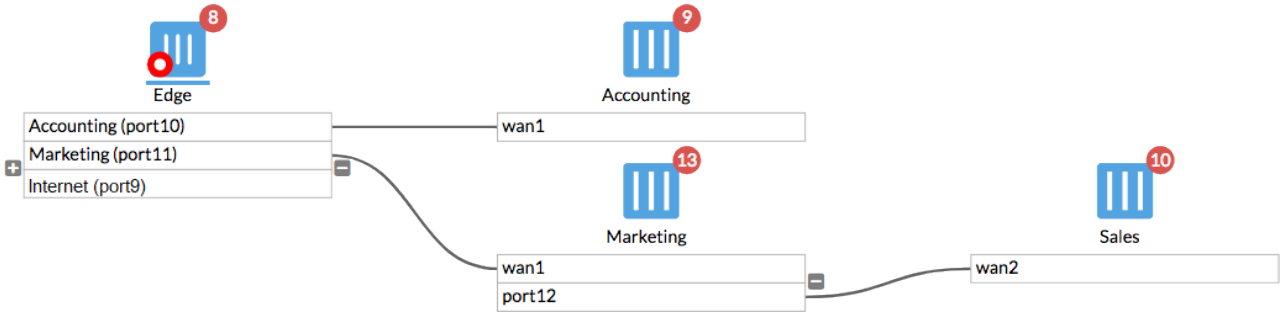
1. On FortiGate (Edge), go to *Dashboard* > *Status* and check the *Security Fabric* widget.



2. On FortiGate (Edge), go to *Security Fabric* > *Physical Topology*.
   This page shows a visualization of access layer devices in the Security Fabric.



3. On FortiGate (Edge), go to *Security Fabric* > *Physical Topology*.
   This dashboard shows information about the interfaces of each device in the Security Fabric.



To run diagnostics:

1. To view the downstream FortiGate pending authorization on the root FortiGate :

```
Edge # diagnose sys csf authorization pending-list
Serial                  IP Address      HA-Members    Path
--------------------------------------------------------------------------------
FG201ETK18902514        0.0.0.0                       FG3H1E5818900718:FG201ETK18902514
```

2. To view the downstream FortiGates after they join Security Fabric on the root or first level downstream FortiGate:

```
Edge # diagnose sys csf downstream
1:     FG201ETK18902514 (192.168.200.10) Management-IP: 0.0.0.0 Management-port:0 parent:
FG3H1E5818900718
        path:FG3H1E5818900718:FG201ETK18902514
```

```
        data received: Y downstream intf:wan1 upstream intf:port11 admin-port:443
        authorizer:FG3H1E5818900718
2:     FGT81ETK18002246 (192.168.10.10) Management-IP: 0.0.0.0 Management-port:0 parent:
FG3H1E5818900718
        path:FG3H1E5818900718:FGT81ETK18002246
        data received: Y downstream intf:wan1 upstream intf:port10 admin-port:443
        authorizer:FG3H1E5818900718
3:     FG101ETK18002187 (192.168.135.10) Management-IP: 0.0.0.0 Management-port:0 parent:
FG201ETK18902514
        path:FG3H1E5818900718:FG201ETK18902514:FG101ETK18002187
        data received: Y downstream intf:wan2 upstream intf:port12 admin-port:443
        authorizer:FG3H1E5818900718
```

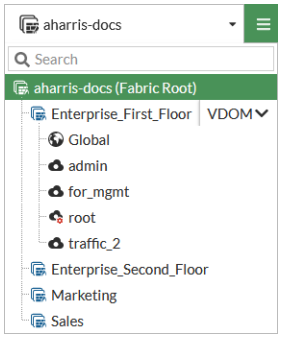
3. To view the upstream FortiGate after the downstream FortiGate joins Security Fabric:

```
Marketing # diagnose sys csf upstream
Upstream Information:
Serial Number:FG3H1E5818900718
IP:192.168.200.2
Connecting interface:wan1
Connection status:Authorized
```

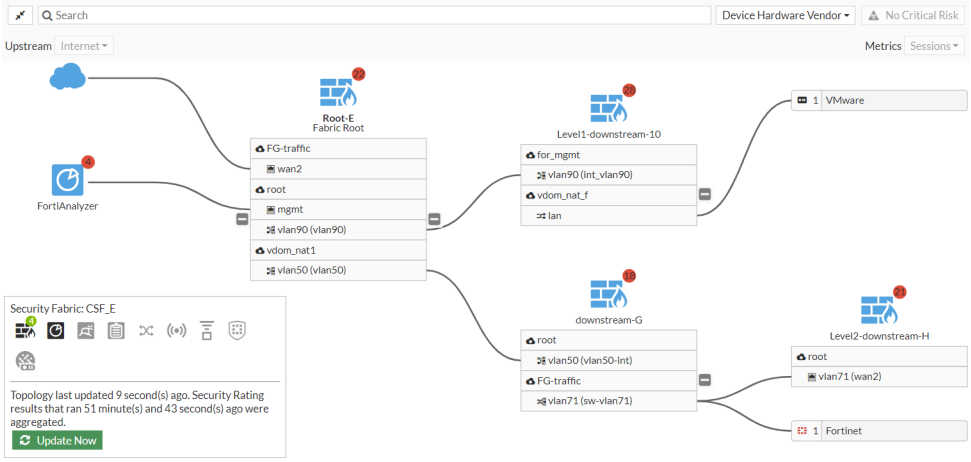
Deploying the Security Fabric in a multi-VDOM environment

A Security Fabric can be enabled in multi-VDOM environments. This allows access to all of the Security Fabric features, including automation, security rating, and topologies, across the VDOM deployment.

- Users can navigate to downstream FortiGate devices and VDOMs directly from the root FortiGate using the Fabric selection menu.



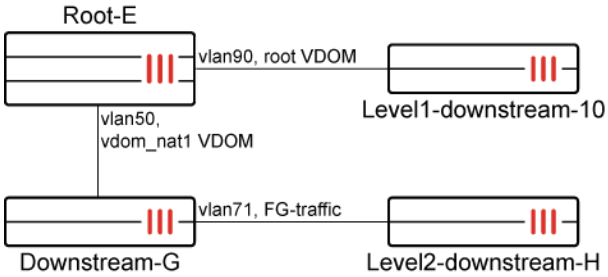
- The logical topology shows all of the configured VDOMs.



- Security rating reports include results for all of the configured VDOMs as well the entire Fabric.

Downstream FortiGate devices must connect to the upstream FortiGate from its management VDOM.

Topology



In this topology, there is a root FortiGate with three FortiGates connected through two different VDOMs. The root FortiGate is able to manage all devices running in multi-VDOM mode.

This example assumes multi-VDOM mode is already configured on each FortiGate, and that FortiAnalyzer logging is configured on the root FortiGate (see Configuring FortiAnalyzer on page 4311 and Configuring the root FortiGate and downstream FortiGates on page 4302 for more details).

To enable multi-VDOM mode:

```
config system global
    set vdom-mode multi-vdom
end
```

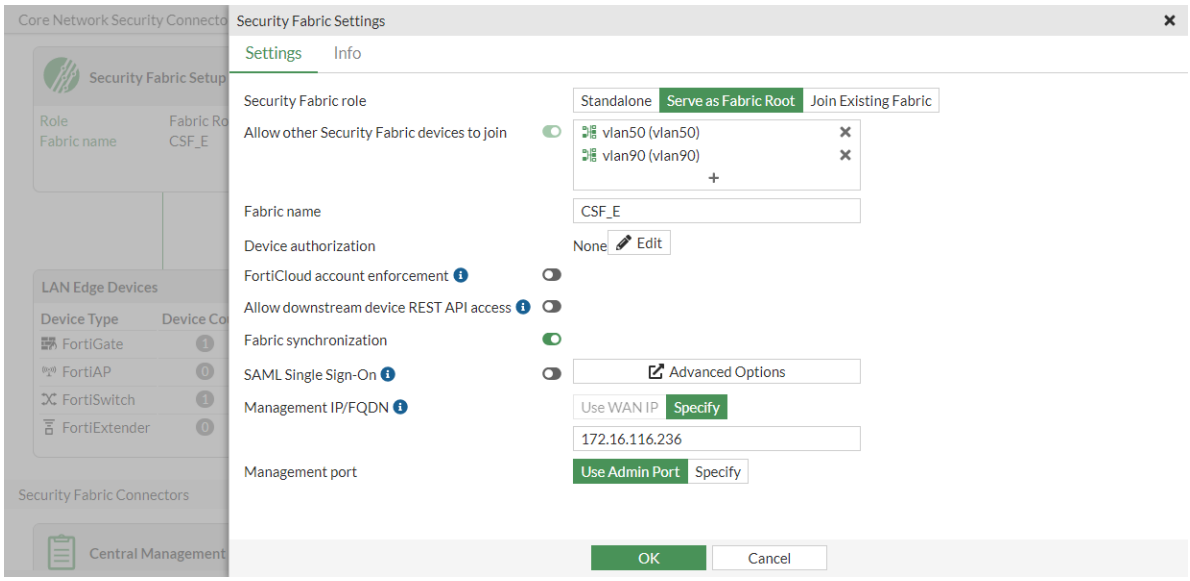
Device configurations

Root FortiGate (Root-E)

The Security Fabric is enabled, and configured so that downstream interfaces from all VDOMs can allow other Security Fabric devices to join.

To configure Root-E in the GUI:

1. Go to *Security Fabric* > *Fabric Connectors* and double-click the *Security Fabric Setup* card.
2. Set the *Security Fabric role* to *Serve as Fabric Root*.
3. Enable *Allow other Security Fabric devices to join* and click the *+* to add the interfaces (vlan50 and vlan90) from the vdom_nat1 and root VDOMs.



4. Configure the other settings as needed.
5. Click *OK*.

To configure Root-E in the CLI:

1. Enable the Security Fabric:

```
config system csf
    set status enable
    set group-name "CSF_E"
end
```

2. Configure the interfaces:

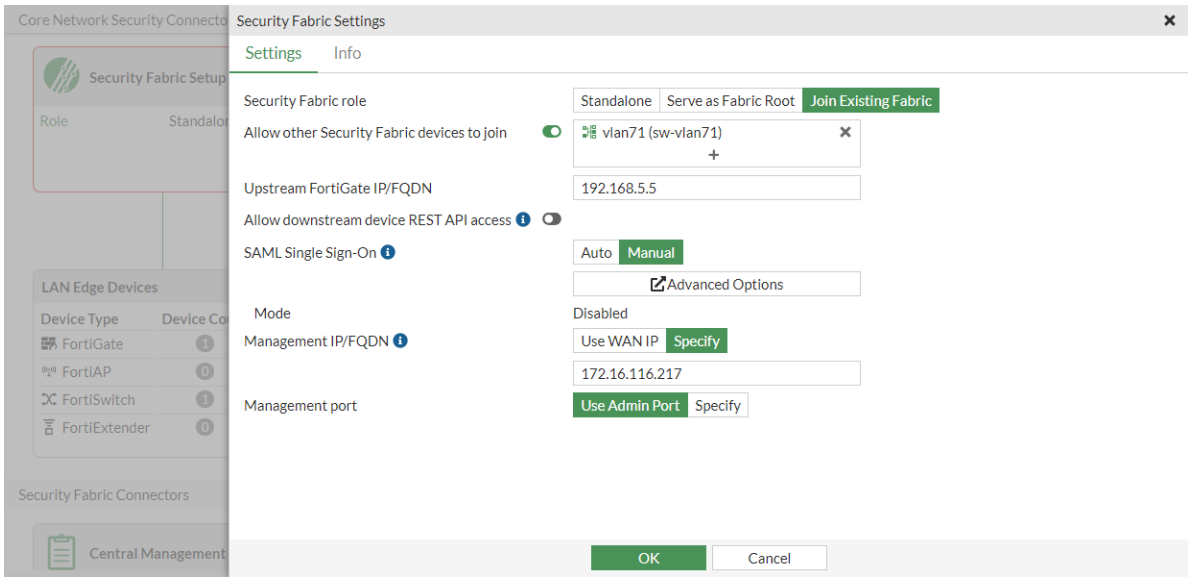
```
config system interface
    edit "vlan50"
        set vdom "vdom_nat1"
        ...
        set allowaccess ping https ssh http fgfm fabric
```

```
        ...
    next
    edit "vlan90"
        set vdom "root"
        ...
        set allowaccess ping https ssh http fgfm fabric
        ...
    next
end
```

Downstream FortiGate 1 (Downstream-G)

To configure Downstream-G in the GUI:

1. Go to *Security Fabric* > *Fabric Connectors* and double-click the *Security Fabric Setup* card.
2. In the *Settings* tab, set the *Security Fabric role* to *Join Existing Fabric*.
3. Enter the *Upstream FortiGate IP*, which is the IP of the root FortiGate vdom_nat1 interface (192.168.5.5). Downstream-G must use the interface from the management VDOM to connect to the upstream FortiGate IP.
4. Enable *Allow other Security Fabric devices to join* and click the *+* to add the downstream interface (sw-vlan71) from the FG-traffic VDOM.



5. Configure the other settings as needed.
6. Click *OK*.

To configure Downstream-G in the CLI:

1. Enable the Security Fabric:

```
config system csf
    set status enable
```

```
    set upstream-ip 192.168.5.5
end
```

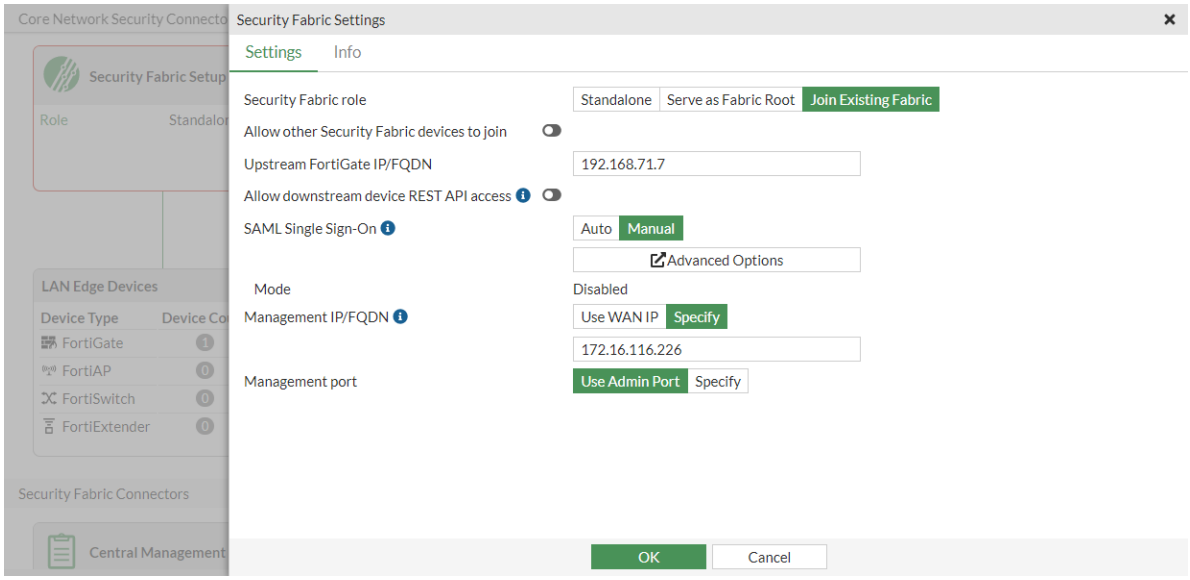
2. Configure the interfaces:

```
config system interface
    edit "sw-vlan71"
        set vdom "FG-traffic"
        ...
        set allowaccess ping https ssh http fgfm fabric
        ...
    next
end
```

Downstream FortiGate 2 (Level2-downstream-H)

To configure Level2-downstream-H in the GUI:

1. Go to *Security Fabric > Fabric Connectors* and double-click the *Security Fabric Setup* card.
2. In the *Settings* tab, set the *Security Fabric role* to *Join Existing Fabric*.
3. Enter the *Upstream FortiGate IP*, which is the IP of the root VDOM on Downstream-G (192.168.71.7).



4. Configure the other settings as needed.
5. Click *OK*.

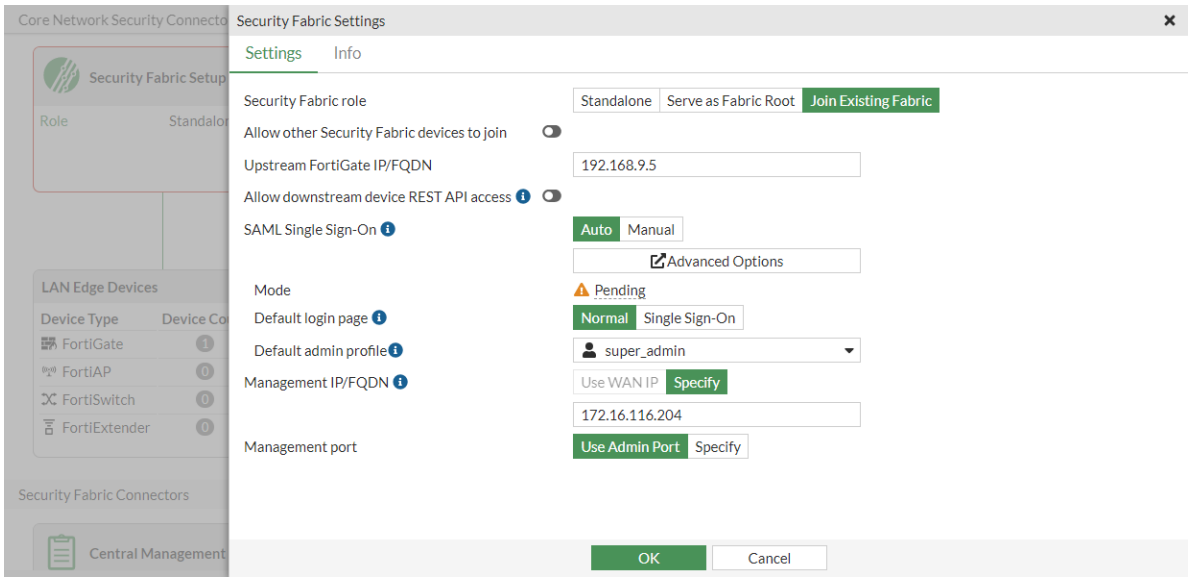
To configure Level2-downstream-H in the CLI:

```
config system csf
    set status enable
    set upstream-ip 192.168.71.7
end
```

Downstream FortiGate 3 (Level1-downstream-10)

To configure Level1-downstream-10 in the GUI:

1. Go to *Security Fabric* > *Fabric Connectors* and double-click the *Security Fabric Setup* card.
2. In the *Settings* tab, set the *Security Fabric role* to *Join Existing Fabric*.
3. Enter the *Upstream FortiGate IP*, which is the IP of the root VDOM on Root-E (192.168.9.5).



4. Configure the other settings as needed.
5. Click *OK*.

To configure Level1-downstream-10 in the CLI:

```
config system csf
    set status enable
    set upstream-ip 192.168.9.5
end
```

Device authorization and verification

To authorize the downstream devices on the root FortiGate:

1. On Root-E, go to *System* > *Firmware & Registration*.
2. Select the unauthorized device and click *Authorization* > *Authorize* for each downstream FortiGate.

   Once all the devices are authorized, the physical topology page shows the root and downstream FortiGates. The logical topology page shows the root and downstream FortiGates connected to interfaces in their corresponding VDOMs.

Other Security Fabric topics

The following topics provide instructions on configuring other Security Fabric use cases:

- Synchronizing objects across the Security Fabric on page 4437
- Group address objects synchronized from FortiManager on page 4444
- Security Fabric over IPsec VPN on page 4445
- Leveraging LLDP to simplify Security Fabric negotiation on page 4451
- Integrate user information from EMS and Exchange connectors in the user store on page 4454

Synchronizing objects across the Security Fabric

When the Security Fabric is enabled, various objects such as addresses, services, and schedules are synced from the upstream FortiGate to all downstream devices by default. FortiOS has the following settings for object synchronization across the Security Fabric:

- Set object synchronization (`fabric-object-unification`) to `default` or `local` on the root FortiGate.
- Set a per object option to toggle whether the specific Fabric object will be synchronized or not. After upgrading from 6.4.3, this option is disabled for supported Fabric objects. The synchronized Fabric objects are kept as locally created objects on downstream FortiGates.
- Define the number of task workers to handle synchronizations.

The firewall object synchronization wizard helps identify objects that are not synchronized and resolves any conflicts. A warning message appears in the topology tree if there is a conflict.

Summary of CLI commands

To configure object synchronization:

```
config system csf
    set fabric-object-unification {default | local}
    set configuration-sync {default | local}
    set fabric-workers <integer>
end
```

Parameter	Description
`fabric-object-unification`	*default*: Global CMDB objects will be synchronized in the Security Fabric. *local*: Global CMDB objects will not be synchronized to and from this device. This command is available on the root FortiGate. If set to local, the device does not synchronize objects from the root, but will send the synchronized objects downstream.
`configuration-sync`	*default*: Synchronize configuration for FortiAnalyzer, FortiSandbox, and Central Management to root node. *local*: Do not synchronize configuration with root node.

Parameter	Description
	If downstream FortiGates are set to local, the synchronized objects from the root to downstream are not applied locally. However, the downstream FortiGate will send the configuration to lower FortiGates.
`fabric-workers`	Define how many task worker process are created to handle synchronizations (1- 4, default = 2). The worker processes dies if there is no task to perform after 60 seconds.

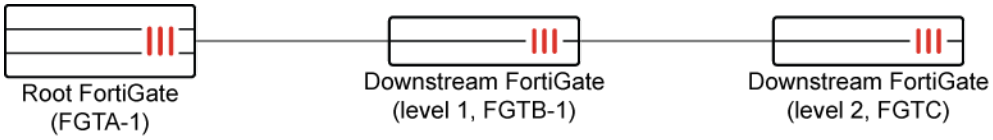
The per object setting can be configured on the root FortiGate as follows:

```
config firewall <object>
    edit <name>
        set fabric-object {enable | disable}
    next
end
```

Where:

- `<object>` is one of the following: `address`, `address6`, `addrgrp`, `addrgrp6`, `service category`, `service custom`, `service group`, `schedule group`, `schedule onetime`, or `schedule recurring`.
- Enabling `fabric-object` sets the object as a Security Fabric-wide global object that is synchronized to downstream FortiGates.
- Disabling `fabric-object` sets the object as local to this Security Fabric member.
- If a device in the Fabric is in multi-VDOM mode, the GUI will not display the Fabric synchronization option. Even if this is enabled in the CLI, the object will not be synchronized to any downstream devices.

Sample topology



In this Security Fabric, the root FortiGate (FGTA-1) has `fabric-object-unification` set to `default` so the Fabric objects can be synchronized to the downstream FortiGate. The level 1 downstream FortiGate (FGTB-1) has `configuration-sync` set to `local`, so it will not apply the synchronized objects locally. The level 2 downstream FortiGate (FGTC) has `configuration-sync` set to `default`, so it will apply the synchronized objects locally.

In this example, firewall addresses and address groups are used. Other supported Fabric objects have the same behaviors. The following use cases illustrate common synchronization scenarios:

- If no conflicts exist, firewall addresses and address groups can be synchronized to downstream FortiGates (see example below).
- If a conflict exists between the root and downstream FortiGates, it can be resolved with the conflict resolution wizard. After the conflict is resolved, the firewall addresses and address groups can be synchronized to downstream FortiGates (see example below).
- If `set fabric-object` (*Fabric synchronization* option in the GUI) is disabled for firewall addresses and address groups on the root FortiGate, they will not be synchronized to downstream FortiGates (see example below).

To configure the FortiGates used in this example:

```
FGTA-1 # config system csf
    set status enable
    set group-name "fabric"
    set fabric-object-unification default
    ...
end
```

```
FGTB-1 # config system csf
    set status enable
    set upstream-ip 10.2.200.1
    set configuration-sync local
    ...
end
```

```
FGTC # config system csf
    set status enable
    set upstream-ip 192.168.7.2
    set configuration-sync default
    ...
end
```

To synchronize a firewall address and address group in the Security Fabric:

1. Configure the firewall address on the root FortiGate:

```
FGTA-1 # config firewall address
    edit "add_subnet_1"
        set fabric-object enable
        set subnet 22.22.22.0 255.255.255.0
    next
end
```

2. Configure the address group on the root FortiGate:

```
FGTA-1 # config firewall addrgrp
    edit "group_subnet_1"
        set member "add_subnet_1"
        set fabric-object enable
    next
end
```

3. Check the firewall address and address group on the downstream FortiGates:

```
FGTB-1 # show firewall address add_subnet_1
entry is not found in table
```

```
FGTB-1 # show firewall addrgrp group_subnet_1
entry is not found in table
```

The synchronized objects are not applied locally on this FortiGate because `configuration-sync` is set to `local`.

```
FGTC # show firewall address add_subnet_1
config firewall address
    edit "add_subnet_1"
        set uuid 378a8094-34cb-51eb-ce40-097f298fcfdc
        set fabric-object enable
        set subnet 22.22.22.0 255.255.255.0
    next
end
```

```
FGTC # show firewall addrgrp group_subnet_1
config firewall addrgrp
    edit "group_subnet_1"
        set uuid 4d7a8a52-34cb-51eb-fce7-d93f76915319
        set member "add_subnet_1"
        set color 19
        set fabric-object enable
    next
end
```

The objects are synchronized on this FortiGate because `configuration-sync` is set to `default`.

To resolve a firewall address and address group conflict in the Security Fabric:

1. On FGTC, create a firewall address:
   a. Go to *Policy & Objects* > *Addresses* and select *Address*.
   b. Click *Create new*.
   c. Configure the following:

Name	sync_add_1
IP/Netmask	33.33.33.0 255.255.255.0

   d. Click *OK*.
2. On FGTA-1 (Fabric root), create the firewall address with same name but a different subnet:
   a. Go to *Policy & Objects* > *Addresses* and select *Address*.
   b. Click *Create new*.
   c. Configure the following:

Name	sync_add_1
IP/Netmask	11.11.11.0 255.255.255.0
Fabric synchronization	Enable

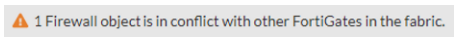
   d. Click *OK*.
3. Add the address to a different address group than what is configured on FGTC:
   a. Go to *Policy & Objects* > *Addresses* and select *Address Group*.
   b. Click *Create new*.

c. Configure the following:

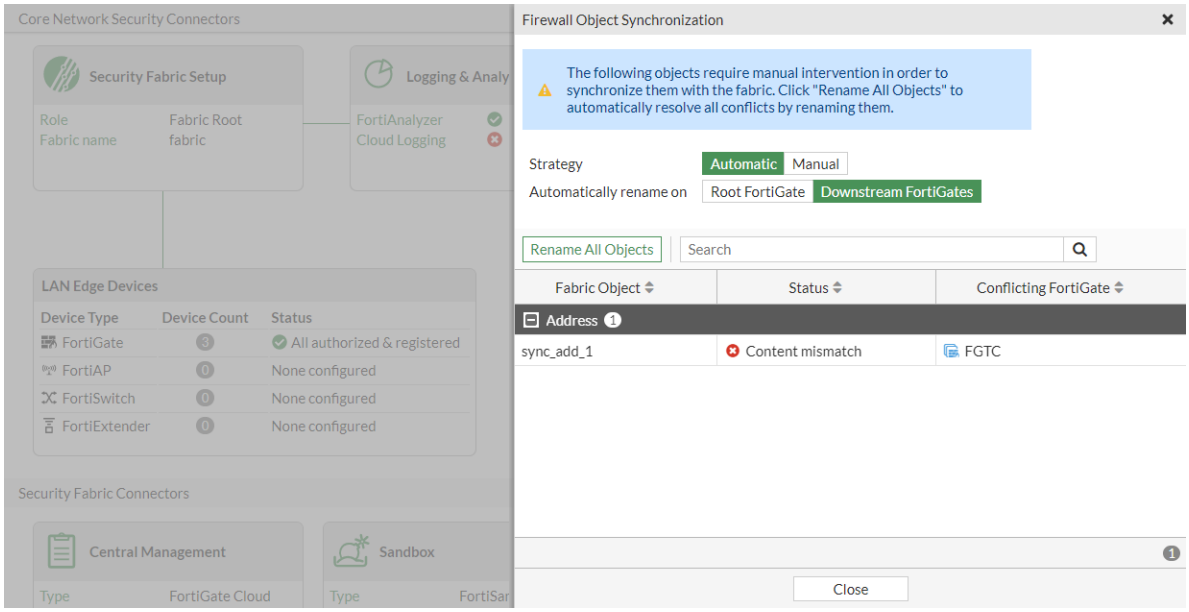
Name	sync_group4
Members	sync_add_1
Fabric synchronization	Enable

d. Click *OK*.

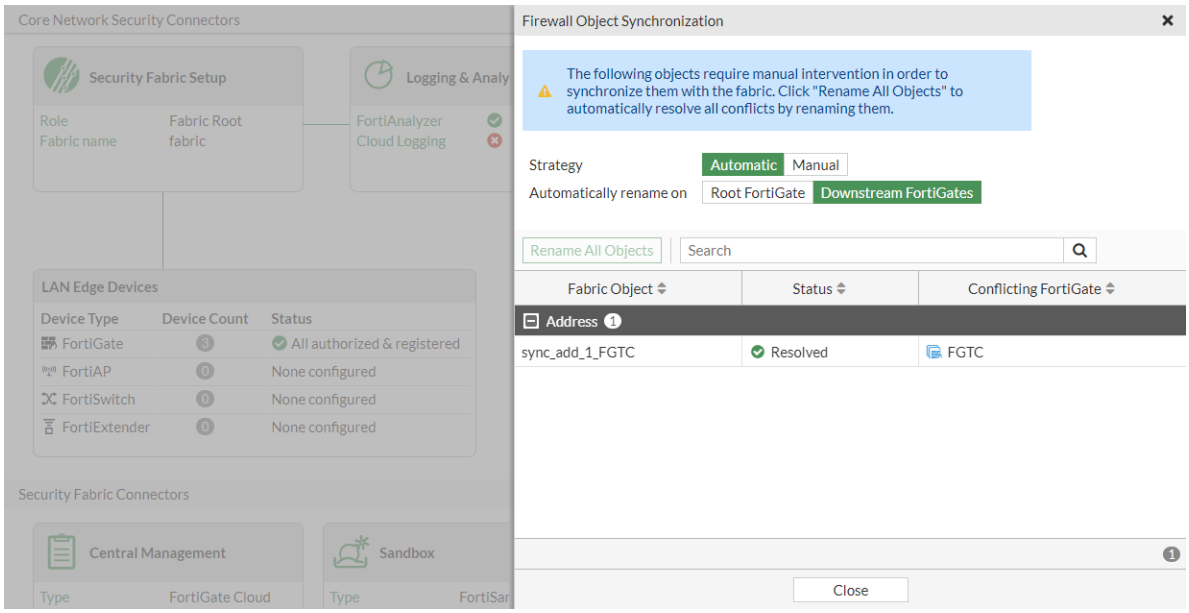
4. Open the notification center drop dropdown. There is a message that *1 Firewall object is conflict with other FortiGates in the fabric*.



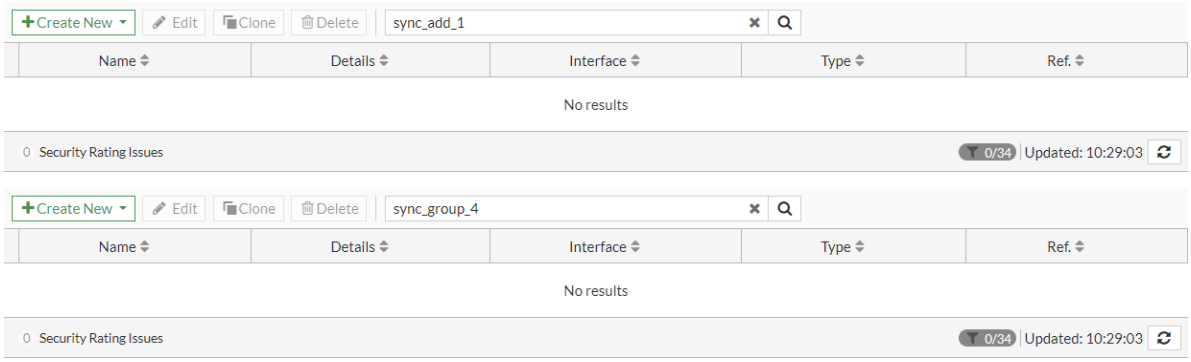
5. Resolve the conflict:
   a. Click the message in the notification center drop dropdown. The *Firewall Object Synchronization* pane opens.
   b. Click *Rename All Objects*. The conflicted object will be renamed on the downstream FortiGate.



   c. The conflict is resolved. Click *Close* to exit the *Firewall Object Synchronization* pane.



6. Verify the results on the downstream FortiGates:
   a. On FGTB-1, go to *Policy & Objects* > *Addresses*.
   b. Search for *sync_add_1* and *sync_group4* in the *Address* and *Group Address* pages, respectively. No results are found. The synchronized objects are not applied locally on this FortiGate because `configuration-sync` is set to `local`.



   c. On FGTC, go to *Policy & Objects* > *Addresses*.
   d. Search for *sync_add_1* in the *Address* page. The original firewall address *sync_add_1* was renamed to *sync_add_1_FGTC* by resolving the conflict on FGTA-1. The address *sync_add_1* and address group *sync_group4* are synchronized from FGTA-1.

To disable Fabric synchronization on the root FortiGate in the GUI:

1. On FGTA-1, create a firewall address:
   a. Go to *Policy & Objects* > *Addresses* and select *Address*.
   b. Click *Create new*.
   c. Configure the following:

Name	add_subnet_3

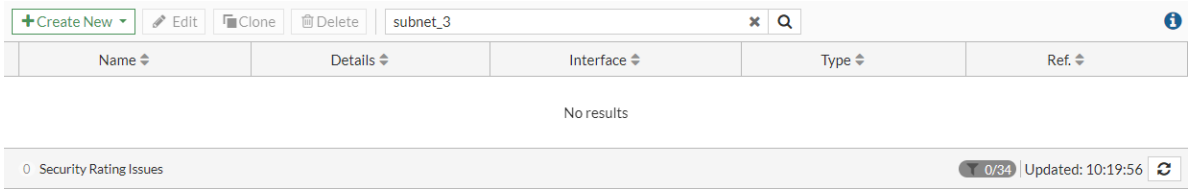
IP/Netmask	33.33.33.0 255.255.255.0
Fabric synchronization	Disable

d. Click *OK*.

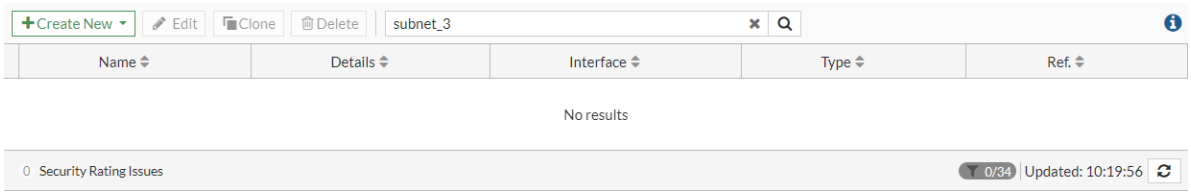
2. Create the firewall address group and add the address:
   a. Go to *Policy & Objects* > *Addresses* and select *Address Group*.
   b. Click *Create new*.
   c. Configure the following:

Name	group_subnet_3
Members	add_subnet_3
Fabric synchronization	Disable

   d. Click *OK*.
3. On FGTB-1, go to *Policy & Objects* > *Addresses* and search for *subnet_3*. No results are found because Fabric synchronization is disabled on the root FortiGate (FGTA-1).



4. On FGTC, go to *Policy & Objects* > *Addresses* and search for *subnet_3*. No results are found because Fabric synchronization is disabled on the root FortiGate (FGTA-1).



To disable Fabric synchronization on the root FortiGate in the CLI:

1. Configure the firewall address on the root FortiGate:

```
FGTA-1 # config firewall address
    edit "add_subnet_3"
        set subnet 33.33.33.0 255.255.255.0
        set fabric-object disable
    next
end
```

2. Configure the address group on the root FortiGate:

```
FGTA-1 # config firewall addrgrp
    edit "group_subnet_3"
        set member "add_subnet_3"
        set fabric-object disable
```

```
    next
end
```

3. Check the firewall address and address group on the downstream FortiGates:

```
FGTB-1 # show firewall address add_subnet_3
entry is not found in table
```

```
FGTB-1 # show firewall addrgrp group_subnet_3
entry is not found in table
```

```
FGTC # show firewall address add_subnet_3
entry is not found in table
```

```
FGTC # show firewall addrgrp group_subnet_3
entry is not found in table
```

The objects are not synchronized from the root FortiGate (FGTA-1) because the `fabric-object` setting is disabled.

Group address objects synchronized from FortiManager

Address objects from external connectors that are learned by FortiManager are synchronized to FortiGate. These objects can be grouped together with the FortiGate CLI to simplify selecting connector objects in the FortiGate GUI. Multiple groups can be created.

This option is only available for objects that are synchronized from FortiManager.

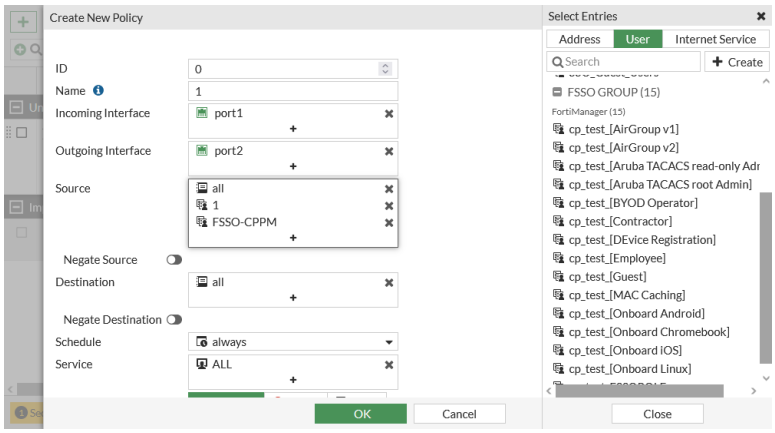
To add an object to a connector group:

```
config user adgrp
    edit <object_name>
        set server-name "FortiManager"
        set connector-source <group_name>
    next
end
```

Example

In this example, objects learned by the FortiManager from an Aruba ClearPass device are synchronized to the FortiGate. Some of the objects are then added to a group called *ClearPass* to make them easier to find in the object list when creating a firewall policy.

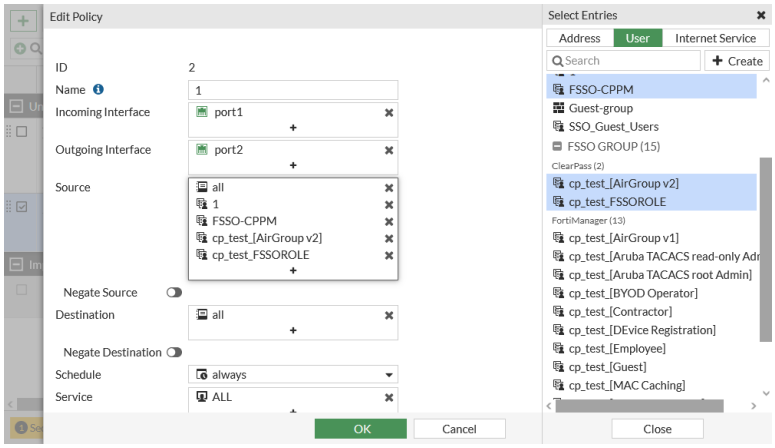
Prior to being grouped, the synchronized objects are listed under the FortiManager heading in the object lists.



To add some of the objects to a group:

```
config user adgrp
    edit "cp_test_FSSOROLE"
        set server-name "FortiManager"
        set connector-source "ClearPass"
    next
    edit "cp_test_[AirGroup v2]"
        set server-name "FortiManager"
        set connector-source "ClearPass"
    next
end
```

The objects are now listed under the *ClearPass* heading.

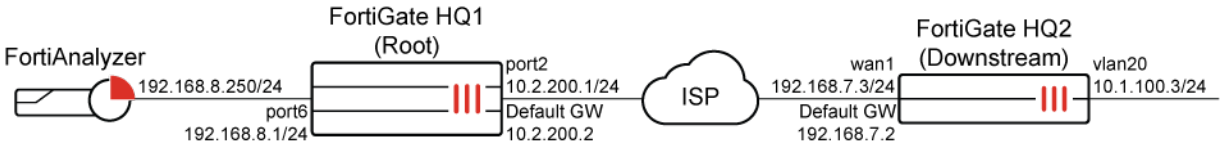


Security Fabric over IPsec VPN

This is an example of configuring Security Fabric over IPsec VPN.

Sample topology

This sample topology shows a downstream FortiGate (HQ2) connected to the root FortiGate (HQ1) over IPsec VPN to join Security Fabric.



Sample configuration

To configure the root FortiGate (HQ1):

1. Configure the interface:
   a. Go to *Network* > *Interfaces*.
   b. Edit *port2*:
      - Set *Role* to *WAN*.
      - For the interface connected to the internet, set the *IP/Network Mask* to *10.2.200.1/255.255.255.0*
   c. Edit *port6*:
      - Set *Role* to *DMZ*.
      - For the interface connected to FortiAnalyzer, set the *IP/Network Mask* to *192.168.8.250/255.255.255.0*
2. Configure the static route to connect to the internet:
   a. Go to *Network* > *Static Routes* and click *Create New* or *Create New* > *IPv4 Static Route*.
      - Set *Destination* to *0.0.0.0/0.0.0.0*.
      - Set *Interface* to *port2*.
      - Set *Gateway Address* to *10.2.200.2*.
   b. Click *OK*.
3. Configure the IPsec VPN:
   a. Go to *VPN* > *IPsec Wizard*.
      - Set *Name* to *To-HQ2*.
      - Set *Template Type* to *Custom*.
      - Click *Next*.
      - Set *Authentication* to *Method*.
      - Set *Pre-shared Key* to *123456*.
   b. Leave all other fields in their default values and click *OK*.
4. Configure the IPsec VPN interface IP address which will be used to form Security Fabric:
   a. Go to *Network* > *Interfaces*.
   b. Edit *To-HQ2*:
      - Set *Role* to *LAN*.
      - Set the *IP/Network Mask* to *10.10.10.1/255.255.255.255*.
      - Set *Remote IP/Network Mask* to *10.10.10.3/255.255.255.0*.
5. Configure the IPsec VPN local and remote subnets:

a. Go to *Policy & Objects > Addresses*.
b. Click *Create New*
- Set *Name* to *To-HQ2_remote_subnet_2*.
- Set *Type* to *Subnet*.
- Set *IP/Network Mask* to *10.10.10.3/32*.

c. Click *OK*.
d. Click *Create New*
- Set *Name* to *To-HQ2_local_subnet_1*.
- Set *Type* to *Subnet*.
- Set *IP/Network Mask* to *192.168.8.0/24*.

e. Click *OK*.
f. Click *Create New*
- Set *Name* to *To-HQ2_remote_subnet_1*.
- Set *Type* to *Subnet*.
- Set *IP/Network Mask* to *10.1.100.0/24*.

g. Click *OK*.

6. Configure the IPsec VPN static routes:
a. Go to *Network > Static Routes*.
b. Click *Create New* or *Create New > IPv4 Static Route*.
- For *Named Address*, select *Type* and select *To-HQ2_remote_subnet_1*.
- Set *Interface* to *To-HQ2*.

Click *OK*.
c. Click *Create New* or *Create New > IPv4 Static Route*.
- For *Named Address*, select *Type* and select *To-HQ2_remote_subnet_1*.
- Set *Interface* to *Blackhole*.
- Set *Administrative Distance* to *254*.

d. Click *OK*.

7. Configure the IPsec VPN policies:
a. Go to *Policy & Objects > Firewall Policy*
b. Click *Create New*.
- Set *Name* to *vpn_To-HQ2_local*.
- Set *Incoming Interface* to *port6*.
- Set *Outgoing Interface* to *To-HQ2*.
- Set *Source* to *To-HQ2_local_subnet_1*.
- Set *Destination* to *To-HQ2_remote_subnet_1*.
- Set *Schedule* to *Always*.
- Set *Service* to *All*.
- Disable *NAT*.

c. Click *OK*.
d. Click *Create New*.
- Set *Name* to *vpn_To-HQ2_remote*.
- Set *Incoming Interface* to *To-HQ2*.

- Set *Outgoing Interface* to *port6*.
- Set *Source* to *To-HQ2_remote_subnet_1, To-HQ2_remote_subnet_2*.
- Set *Destination* to *To-HQ2_local_subnet_1*.
- Set *Schedule* to *Always*.
- Set *Service* to *All*.
- Enable *NAT*.
- Set *IP Pool Configuration* to *Use Outgoing Interface Address*.

e. Click *OK*.

8. Configure the Security Fabric:
   a. Go to *Security Fabric > Fabric Connectors* and double-click the *Security Fabric Setup* card.
   b. Select the *Settings* tab, and set the *Security Fabric role* to *Serve as Fabric Root*.
   c. Enter a *Fabric name*, such as *Office-Security-Fabric*.
   d. Ensure *Allow other Security Fabric devices to join* is enabled and add VPN interface *To-HQ2*.
   e. Click *OK*.
9. Configure the FortiAnalyzer logging settings:
   a. Go to *Security Fabric > Fabric Connectors* and double-click the *Logging & Analytics* card.
   b. Select the *Settings* tab, select the *FortiAnalyzer* tab, and set the *Status* to *Enabled*.
   c. Enter the FortiAnalyzer IP in the *Server* field (*192.168.8.250*). The *Upload option* is automatically set to *Real Time*.
   d. Click *Refresh*. The FortiAnalyzer serial number is verified.
   e. Click *OK*.

To configure the downstream FortiGate (HQ2):

1. Configure the interface:
   a. Go to *Network > Interfaces*.
   b. Edit interface *wan1*:
   - Set *Role* to *WAN*.
   - For the interface connected to the internet, set the *IP/Network Mask* to *192.168.7.3/255.255.255.0*.

   c. Edit interface *vlan20*:
   - Set *Role* to *LAN*.
   - For the interface connected to local endpoint clients, set the *IP/Network Mask* to *10.1.100.3/255.255.255.0*.

2. Configure the static route to connect to the internet:
   a. Go to *Network > Static Routes* and click *Create New* or *Create New > IPv4 Static Route*.
   - Set *Destination* to *0.0.0.0/0.0.0.0*.
   - Set *Interface* to *wan1*.
   - Set *Gateway Address* to *192.168.7.2*.

   b. Click *OK*.
3. Configure the IPsec VPN:

a. Go to *VPN > IPsec Wizard*.
- Set *VPN Name* to *To-HQ1*.
- Set *Template Type* to *Custom*.
- Click *Next*.
- In the *Network IP Address*, enter *10.2.200.1*.
- Set *Interface* to *wan1*.
- Set *Authentication* to *Method*.
- Set *Pre-shared Key* to *123456*.

b. Leave all other fields in their default values and click *OK*.

4. Configure the IPsec VPN interface IP address which will be used to form Security Fabric:
a. Go to *Network > Interfaces*.
b. Edit *To-HQ1*:
- Set *Role* to *WAN*.
- Set the *IP/Network Mask* to *10.10.10.3/255.255.255.255*.
- Set *Remote IP/Network Mask* to *10.10.10.1/255.255.255.0.0*.

5. Configure the IPsec VPN local and remote subnets:
a. Go to *Policy & Objects > Addresses*.
b. Click *Create New*
- Set *Name* to *To-HQ1_local_subnet_1*.
- Set *Type* to *Subnet*.
- Set *IP/Network Mask* to *10.1.100.0/24*.

c. Click *OK*.
d. Click *Create New*
- Set *Name* to *To-HQ1_remote_subnet_1*.
- Set *Type* to *Subnet*.
- Set *IP/Network Mask* to *192.168.8.0/24*.

e. Click *OK*.

6. Configure the IPsec VPN static routes:
a. Go to *Network > Static Routes* and click *Create New* or *Create New > IPv4 Static Route*.
- For *Named Address*, select *Type* and select *To-HQ1_remote_subnet_1*.
- Set *Interface* to *To-HQ1*.

b. Click *OK*.
c. Click *Create New* or *Create New > IPv4 Static Route*.
- For *Named Address*, select *Type* and select *To-HQ1_remote_subnet_1*.
- Set *Interface* to *Blackhole*.
- Set *Administrative Distance* to *254*.

d. Click *OK*.

7. Configure the IPsec VPN policies:
a. Go to *Policy & Objects > Firewall Policy* and click *Create New*.
- Set *Name* to *vpn_To-HQ1_local*.
- Set *Incoming Interface* to *vlan20*.
- Set *Outgoing Interface* to *To-HQ1*.

- Set *Source* to *To-HQ1_local_subnet_1*.
- Set *Destination* to *To-HQ1_remote_subnet_1*.
- Set *Schedule* to *Always*.
- Set *Service* to *All*.
- Disable *NAT*.

b. Click *OK*.

c. Click *Create New*.

- Set *Name* to *vpn_To-HQ1_remote*.
- Set *Incoming Interface* to *To-HQ1*.
- Set *Outgoing Interface* to *vlan20*.
- Set *Source* to *To-HQ1_remote_subnet_1*.
- Set *Destination* to *-HQ1_local_subnet_1*.
- Set *Schedule* to *Always*.
- Set *Service* to *All*.
- Disable *NAT*.

d. Click *OK*.

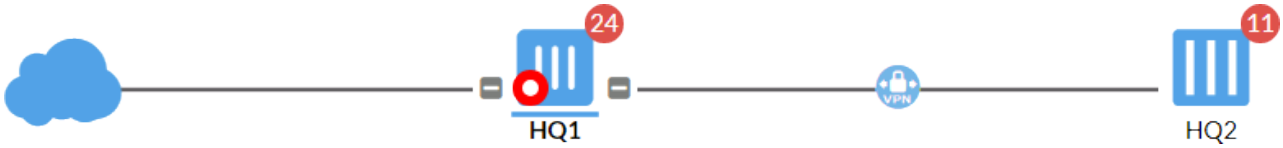
8. Configure the Security Fabric:
   a. Go to *Security Fabric > Fabric Connectors* and double-click the *Security Fabric Setup* card.
   b. In the *Settings* tab, set the *Security Fabric role* to *Join Existing Fabric*.
   FortiAnalyzer automatically enables logging. FortiAnalyzer settings will be retrieved when the downstream FortiGate connects to the root FortiGate.
   c. Set the *Upstream FortiGate IP* to *10.10.10.1*.
   d. Click *OK*.

To authorize the downstream FortiGate (HQ2) on the root FortiGate (HQ1):

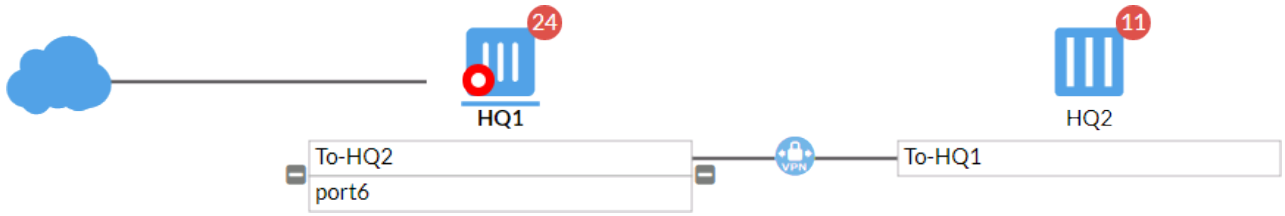
1. In the root FortiGate (HQ1), go to *System > Firmware & Registration*.
   The table highlights the connected FortiGate with its serial numbers that is unauthorized.
2. Select the unauthorized device and click *Authorization > Authorize*.
   After authorization, the downstream FortiGate (HQ2) appears in the *Security Fabric* widget. This means the downstream FortiGate (HQ2) has successfully joined the Security Fabric.

To check the Security Fabric over IPsec VPN:

1. On the root FortiGate (HQ1), go to *Security Fabric > Physical Topology*.
   The root FortiGate (HQ1) is connected by the downstream FortiGate (HQ2) with VPN icon in the middle.



2. On the root FortiGate (HQ1), go to *Security Fabric > Logical Topology*.
   The root FortiGate (HQ1) VPN interface *To-HQ2* is connected by downstream FortiGate (HQ2) VPN interface *To-HQ1* with VPN icon in the middle.



To run diagnostics:

1. To view the downstream FortiGate pending authorization on root FortiGate (HQ1):

```
HQ1 # diagnose sys csf authorization pending-list
Serial                  IP Address      HA-Members                                      Path
--------------------------------------------------------------------------------
FG101ETK18002187        0.0.0.0
FG3H1E5818900718:FG101ETK18002187
```

2. To view the downstream FortiGate (HQ2) after it joins the Security Fabric:

```
HQ1 # diagnose sys csf downstream
 1:     FG101ETK18002187 (10.10.10.3) Management-IP: 0.0.0.0 Management-port:0 parent:
FG3H1E5818900718
        path:FG3H1E5818900718:FG101ETK18002187
        data received: Y downstream intf:To-HQ1 upstream intf:To-HQ2 admin-port:443
        authorizer:FG3H1E5818900718
```

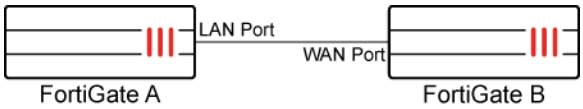
3. To view the root FortiGate (HQ1) on the downstream FortiGate (HQ2) after joining the Security Fabric:

```
HQ2 # diagnose sys csf upstream
Upstream Information:
Serial Number:FG3H1E5818900718
IP:10.10.10.1
Connecting interface:To-HQ1
Connection status:Authorized
```

Leveraging LLDP to simplify Security Fabric negotiation

LLDP reception is enabled on WAN interfaces, which prompts FortiGates that are joining the Security Fabric if the upstream FortiGate asks.

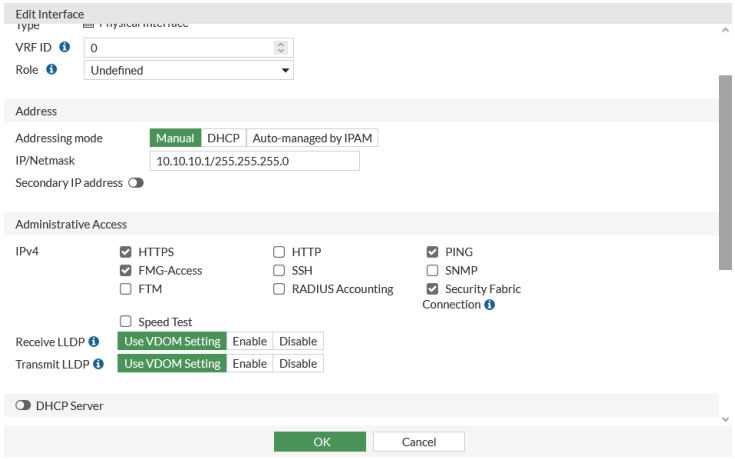
- If the interface role is undefined, LLDP reception and transmission inherit settings from the VDOM.
- If the interface role is WAN, LLDP reception is enabled.
- If the interface role is LAN, LLDP transmission is enabled.



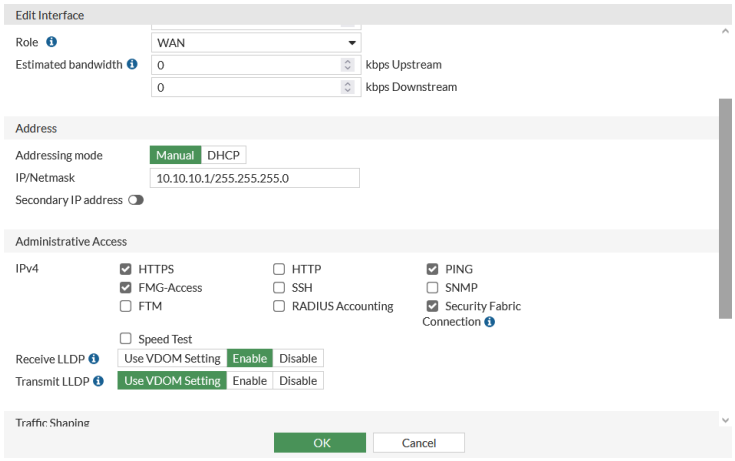
When a FortiGate B's WAN interface detects that FortiGate A's LAN interface is immediately upstream (through the default gateway), and FortiGate A has Security Fabric enabled, FortiGate B will show a notification on the GUI asking to join the Security Fabric.

To configure LLDP reception and join a Security Fabric in the GUI:

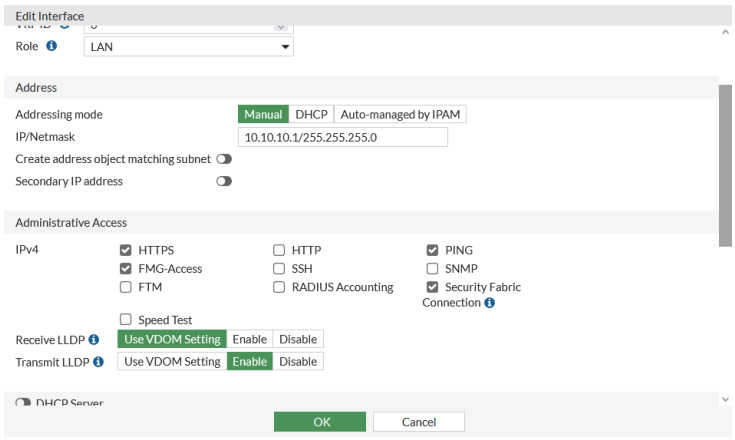
1. On FortiGate A, go to *Network* > *Interfaces*.
2. Configure an interface:
   - If the interface's role is undefined, under *Administrative Access*, set *Receive LLDP* and *Transmit LLDP* to *Use VDOM Setting*.



   - If the interface's role is WAN, under *Administrative Access*, set *Receive LLDP* to *Enable* and *Transmit LLDP* to *Use VDOM Setting*.



   - If the interface's role is LAN, under *Administrative Access*, set *Receive LLDP* to *Use VDOM Setting* and *Transmit LLDP* to *Enable*.



3. Click *OK*. A notification is shown on FortiGate B, *You can connect to a Security Fabric via an upstream FortiGate at 10.10.10.1*.
4. Click the notification. The *Security Fabric Settings* page opens. All the required settings automatically configured.
5. Click *OK* to apply the settings.

To configure LLDP reception and join a Security Fabric in the CLI:

1. Configure the interface on FortiGate A:
   - Undefined role

```
config system interface
    edit "port3"
        set lldp-reception vdom
        set lldp-transmission vdom
        set role undefined
        ...
    next
end
```

   - WAN role

```
config system interface
    edit "wan1"
        set lldp-reception enable
        set lldp-transmission vdom
        set role wan
        ...
    next
end
```

   - LAN role

```
config system interface
    edit "port2"
        set lldp-reception vdom
        set lldp-transmission enable
        set role lan
```

```
        ...
    next
end
```

2. Edit the Security Fabric settings on FortiGate B:

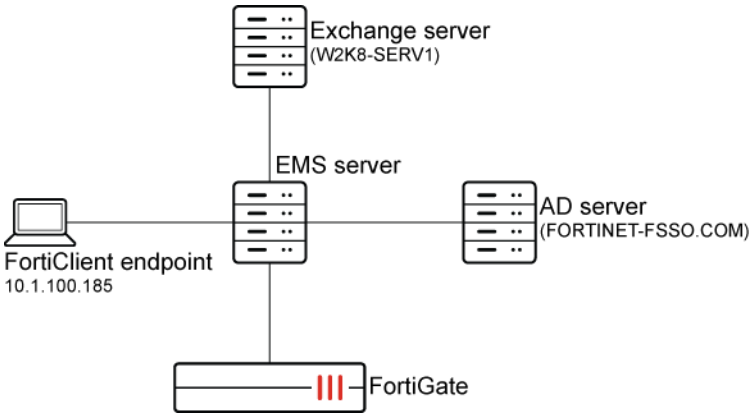
```
config system csf
    set status enable
    set upstream-ip 10.2.200.1
end
```

Integrate user information from EMS and Exchange connectors in the user store

When a FortiClient endpoint is managed by EMS, logged in user and domain information is shared with FortiOS through the EMS connector. This information can be joined with the Exchange connector to produce more complete user information in the user store.

The `diagnose user-device-store device memory list` command displays detailed device information.

Example



In this example, the FortiClient PC user (test1) logs on to the AD domain (FORTINET-FSSO.COM), which is also the same domain as the Exchange server. The user information is pushed to the EMS server that the user is registered to. The FortiGate synchronizes the information from EMS, and at the same time looks up the user on the Exchange server under the Exchange connector. If the user exists on the Exchange server, additional information is fetched. These details are combined in the user store, which is visible in the *FortiClient* widget in the *Status* dashboard.

To configure the Exchange server:

```
config user exchange
    edit "exchange-140"
        set server-name "W2K8-SERV1"
        set domain-name "FORTINET-FSSO.COM"
        set username "Administrator"
```

```
        set password ********
    next
end
```

To configure the EMS server:

```
config endpoint-control fctems
    edit 1
        set status enable
        set name "ems133"
        set server "172.18.62.12"
        set certificate-fingerprint "4F:A6:76:E2:00:4F:A6:76:E2:00:4F:A6:76:E2:00:E0"
    next
end
```

To view the user information in the GUI:

1. Go to *Dashboard > Status*.
2. In the *FortiClient* widget, hover over a device or user name to view the information.

To view the user information in the CLI:

```
# diagnose user-device-store device memory list
...
Record #13:
        device_info
                'ipv4_address' = '10.1.100.185'
                'mac' = '00:0c:29:11:5b:6b'
                'hardware_vendor' = 'VMware'
                'vdom' = 'root'
                'os_name' = 'Microsoft'
                'os_version' = 'Windows 7 Professional Edition, 32-bit Service Pack 1  (build
7601)'
                'hostname' = 'win7-5'
                'unauth_user' = 'Administrator'
                'last_seen' = '1611356490'
                'host_src' = 'forticlient'
                'user_info_src' = 'forticlient'
                'is_forticlient_endpoint' = 'true'
                'unjoined_forticlient_endpoint' = 'false'
                'is_forticlient_unauth_user' = 'true'
                'avatar_source' = 'OS'
                'domain' = 'Fortinet-FSSO.COM'
                'forticlient_id' = '********************************'
                'forticlient_username' = 'Administrator'
                'forticlient_version' = '6.4.2'
                'on_net' = 'true'
                'quarantined_on_forticlient' = 'false'
                'vuln_count' = '0'
                'vuln_count_critical' = '0'
```

```
                'vuln_count_high' = '0'
                'vuln_count_info' = '0'
                'vuln_count_low' = '0'
                'vuln_count_medium' = '0'
                'is_online' = 'true'
        interface_info
                'ipv4_address' = '10.1.100.185'
                'mac' = '00:0c:29:11:5b:6b'
                'master_mac' = '00:0c:29:11:5b:6b'
                'detected_interface' = 'port10'
                'last_seen' = '1611356490'
                'is_master_device' = 'true'
                'is_detected_interface_role_wan' = 'false'
                'detected_interface_fortitelemetry' = 'true'
                'forticlient_gateway_interface' = 'port10'
                'on_net' = 'true'
                'is_online' = 'true'
```

Configuring the Security Fabric with SAML

Security Assertion Markup Language (SAML) is an open standard for exchanging authentication and authorization data between one Identity Provider (IdP) and one or more Service Providers (SP). Both parties exchange messages using the XML protocol as transport. FortiGate firewall devices can be configured as IdPs or SPs.

When the Security Fabric is enabled, you can configure the root FortiGate as the IdP. You can also configure downstream FortiGates to be automatically configured as SPs, with all links required for SAML communication, when added to the Security Fabric. Administrators must still be authorized on each device. Credentials are verified by the root FortiGate, and login credentials are shared between devices. Once authorized, an administrator can move between Fabric devices without logging in again.

Optionally, the downstream FortiGate can also be manually configured as an SP, and then linked to the root FortiGate.

The authentication service is provided by the root FortiGate using local system admin accounts for authentication. Any of the administrator account types can be used for SAML log in. After successful authentication, the administrator logs in to the first downstream FortiGate SP, and can then connect to other downstream FortiGates that have the SSO account properly configured, without needing to provide credentials again, as long as admins use the same browser session. In summary, the root FortiGate IdP performs SAML SSO authentication, and individual device administrators define authorization on FortiGate SPs by using security profiles.

Configuring single-sign-on in the Security Fabric

SAML SSO enables a single FortiGate device to act as the identify provider (IdP), while other FortiGate devices act as service providers (SP) and redirect logins to the IdP.

Only the root FortiGate can be the identity provider (IdP). The downstream FortiGates can be configured as service providers (SP).

The process is as follows:

1. Configuring the root FortiGate as the IdP on page 4457
2. Configuring a downstream FortiGate as an SP on page 4458
3. Configuring certificates for SAML SSO on page 4460
4. Verifying the single-sign-on configuration on page 4461

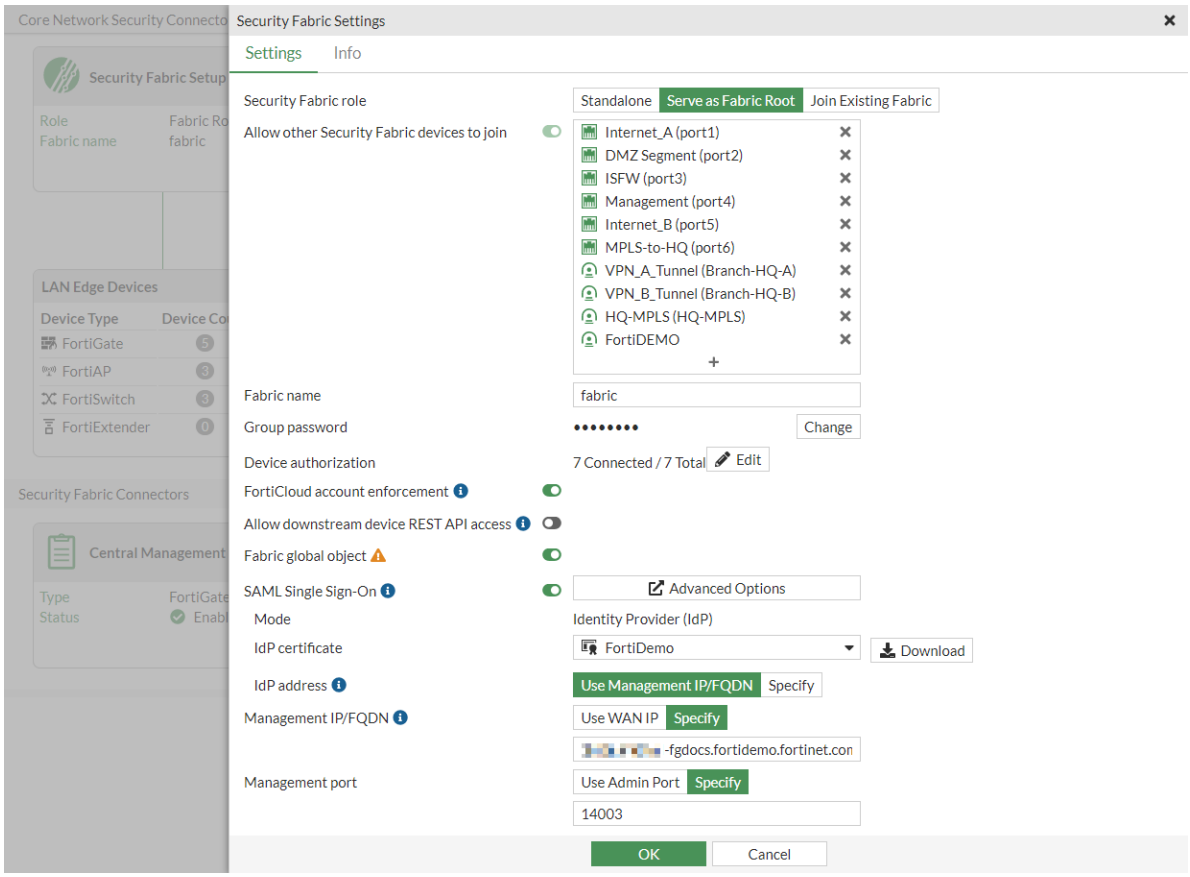
You can also use the CLI. See CLI commands for SAML SSO on page 4462.

Configuring the root FortiGate as the IdP

To configure the root FortiGate as the IdP:

1. Log in to the root FortiGate.
2. Go to *Security Fabric > Fabric Connectors* and double-click the *Security Fabric Setup* card.
3. Enable *SAML Single Sign-On*. The *Mode* field is automatically populated as *Identity Provider (IdP)*.
4. Enter an IP address in the *Management IP/FQDN* field.
5. Enter a management port in the *Management port* field.
   The *Management IP/FQDN* will be used by the SPs to redirect the login request. The *Management IP/FQDN* and *Management port* must be reachable from the user's device.

6. Select the *IdP certificate*.



7. Click *OK*.

Configuring a downstream FortiGate as an SP

There are two ways to configure the downstream FortiGate:

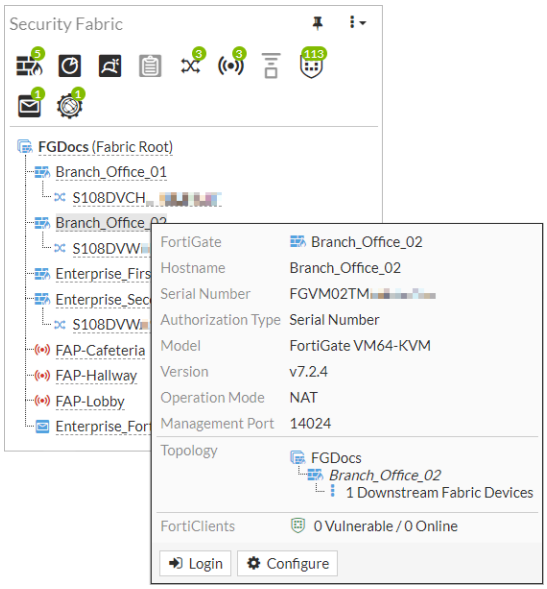
- From the root FortiGate
- From within the downstream device

> An SP must be a member of the Security Fabric before you configure it.

To configure the downstream FortiGate from the root FortiGate:

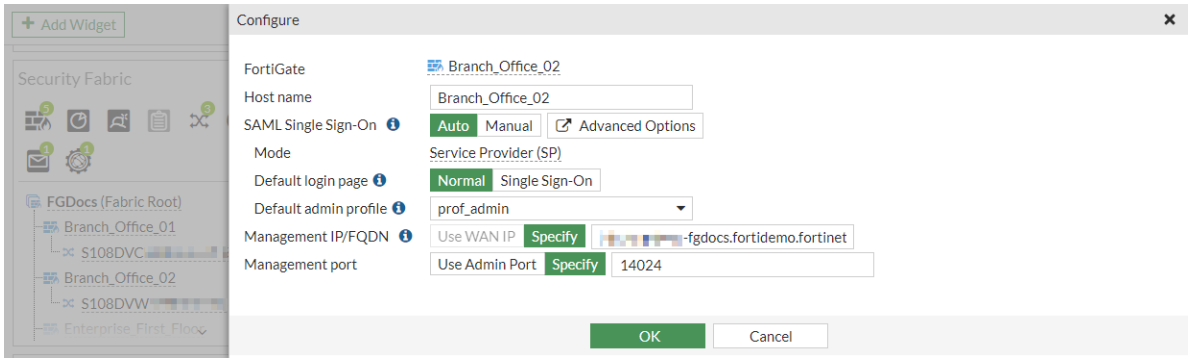
1. Log in to the root FortiGate.
2. Go to *Dashboard* > *Status* and locate the *Security Fabric* widget.

3. In the topology tree, hover over a FortiGate and click *Configure*.



The *Configure* pane opens.

4. Select a *SAML Single Sign-On* option. *Auto* sets the device to SP mode. *Manual* allows you to configure the SSO settings by clicking *Advanced Options*.
5. Select a *Default login page* option.
6. Select one of the following *Default admin profile* types: *prof_admin*, *super_admin*, or *super_admin_readonly*.
7. Enter an IP address in the *Management IP/FQDN* field.
8. Enter a management port in the *Management port* field.
   The *Management IP/FQDN* will be used by the IdP and so other SPs can redirect to each other. The *Management port* must be reachable from the user's device.
9. Click *OK*.



To configure the downstream FortiGate within the device:

1. Log in to the downstream FortiGate.
2. Go to *Security Fabric* > *Fabric Connectors* and double-click the *Security Fabric Setup* card.
3. In the *Settings* tab, select a *SAML Single Sign-On* option. *Auto* sets the device to SP mode. *Manual* allows you to configure the SSO settings by clicking *Advanced Options*.
4. Select a *Default login page* option.

5. Select one of the following *Default admin profile* types: *prof_admin*, *admin_no_access*, *super_admin*, or *super_admin_readonly*.
6. Enter an IP address in the *Management IP/FQDN* field.
7. Enter a management port in the *Management port* field.
   The *Management IP/FQDN* will be used by the IdP and so other SPs can redirect to each other. The *Management port* must be reachable from the user's device.
8. Click *OK*.

Configuring certificates for SAML SSO

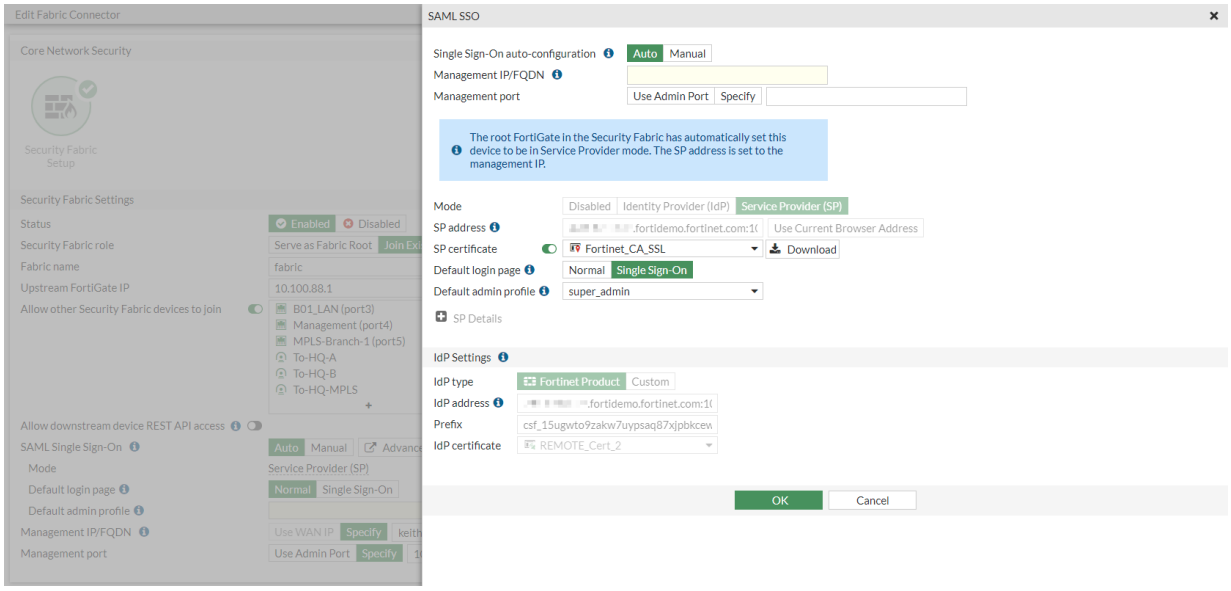
Because communication between the root FortiGate IdP and FortiGate SPs is secured, you must select a local server certificate in the *IdP certificate* option on the root FortiGate. When downstream SPs join the IdP (root FortiGate), the SP automatically obtains the certificate.

In the following SP example, the *IdP certificate* displays *REMOTE_Cert_2*, which is the root server certificate for the IdP:

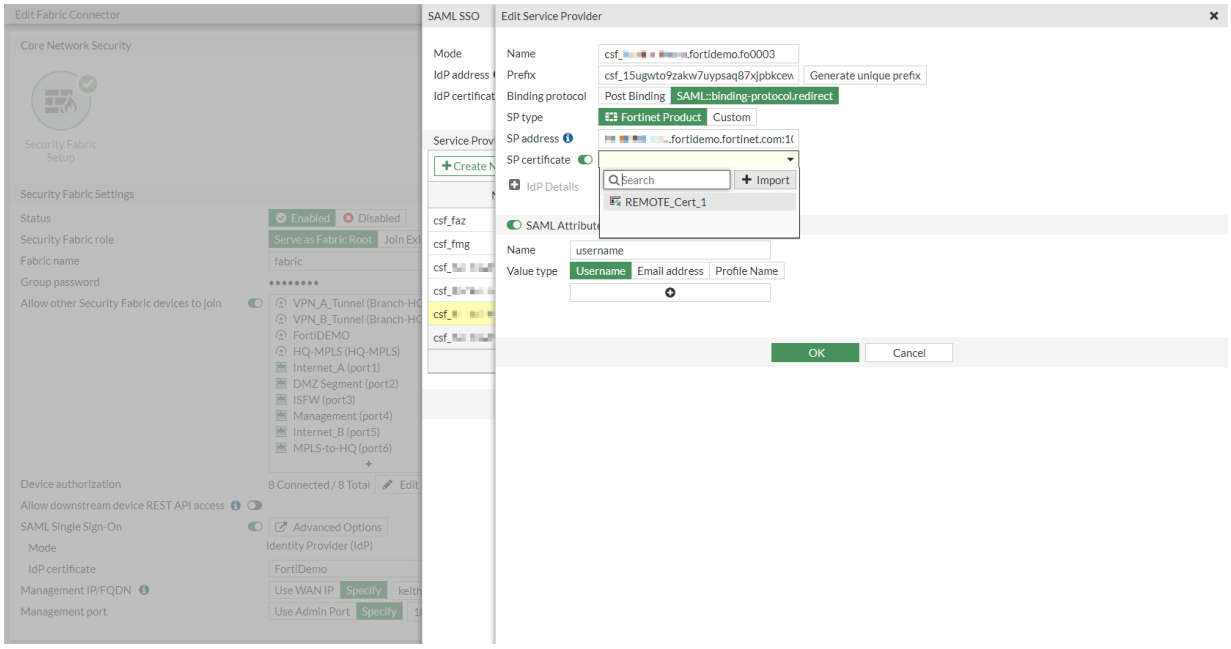
It is possible to manually import a certificate from an SP to the IdP so it can be used for authentication.

To manually import an SP certificate to an IdP:

1. Add the certificate:
   a. On the SP, go to *Security Fabric > Fabric Connectors* and double-click the *Security Fabric Setup* card.
   b. Click *Advanced Options*. The *SAML SSO* pane opens.
   c. Enable *SP certificate* and select a certificate from the dropdown box.
   d. Click *Download*. The certificate is downloaded on the local file system.
   e. Click *OK* to close the *SAML SSO* pane.
   f. Click *OK* to close the *Security Fabric Setup* card.



2. Import the certificate:
   a. On the IdP, go to *Security Fabric > Fabric Connectors* and double-click the *Security Fabric Setup* card.
   b. Click *Advanced Options*. The *SAML SSO* pane opens.
   c. In the *Service Providers* table, select the SP from step 1 and click *Edit*.
   d. Enable *SP certificate* and in the dropdown box, click *Import*.



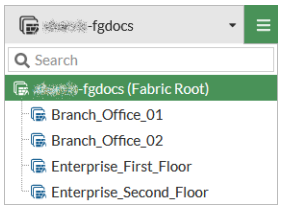
   The *Upload Remote Certificate* window opens.
   e. Click *Upload* and select the certificate downloaded in step 1.
   f. Select *REMOTE_Cert_2*.
   g. Click *OK*. The certificate is imported.
   h. In the *IdP certificate* list, select the certificate that you imported.
   i. Click *OK* to close the *SAML SSO* pane.
   j. Click *OK* to close the *Security Fabric Setup* card.

Verifying the single-sign-on configuration

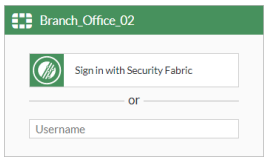
After you have logged in to a Security Fabric member using SSO, you can navigate between any Security Fabric member with SSO configured.

To navigate between Security Fabric members:

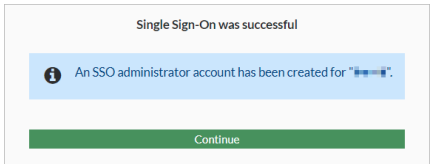
1. Log in to a Security Fabric member that is using SSO.
2. In the top banner, click the name of the device you are logged in to. A list of Security Fabric members displays.



3. Click a Security Fabric member. The login page appears. Click *Sign in with Security Fabric*.

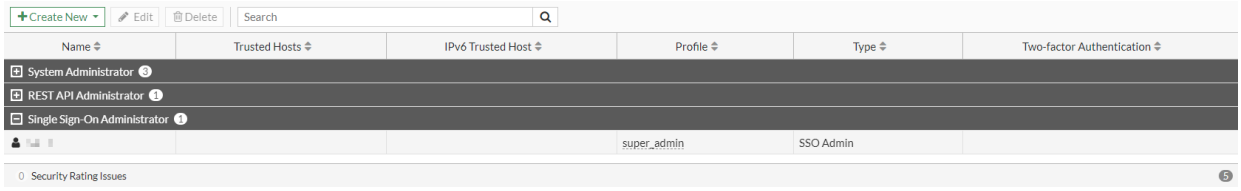


4. A prompt appears that an SSO administrator account has been created. Click *Continue*.



You are now logged in to the Security Fabric member with SSO. The letters "SSO" also display beside the user name in the top banner.

5. Go to *System* > *Administrators* > *Single-Sign-On Administrator* to view the list of SSO admins created.



CLI commands for SAML SSO

To enter a question mark (?) or a tab, Ctrl + V must be entered first. Question marks and tabs cannot be typed or copied into the CLI Console or some SSH clients.

To configure the IdP:

```
config system saml
    set status enable
    set role identity-provider
    set cert "Fortinet_Factory"
    set server-address "172.16.106.74"
    config service-providers
        edit "csf_172.16.106.74:12443"
            set prefix "csf_ngczjwqxujfsbhgr9ivhehwu37fml20"
            set sp-entity-id "http://172.16.106.74/metadata/"
            set sp-single-sign-on-url "https://172.16.106.74/saml/?acs"
            set sp-single-logout-url "https://172.16.106.74/saml/?sls"
            set sp-portal-url "https://172.16.106.74/saml/login/"
            config assertion-attributes
```

```
                edit "username"
                next
                edit "tdoc@fortinet.com"
                    set type email
                next
            end
        next
    end
end
```

To configure an SP:

```
config system saml
    set status enable
    set cert "Fortinet_Factory"
    set idp-entity-id "http://172.16.106.74/saml-idp/csf_
ngczjwqxujfsbhgr9ivhehwu37fml20/metadata/"
    set idp-single-sign-on-url "https://172.16.106.74/csf_
ngczjwqxujfsbhgr9ivhehwu37fml20/login/"
    set idp-single-logout-url "https://172.16.106.74/saml-idp/csf_
ngczjwqxujfsbhgr9ivhehwu37fml20/logout/"
    set idp-cert "REMOTE_Cert_1"
    set server-address "172.16.106.74:12443"
end
```

To configure an SSO administrator:

```
config system sso-admin
    edit "SSO-admin-name"
        set accprofile <SSO admin user access profile>
        set vdom <Virtual domain(s) that the administrator can access>
    next
end
```

SAML SSO with pre-authorized FortiGates

You can set up SAML SSO authentication in a Security Fabric environment by starting with a root FortiGate that has one or more pre-authorized FortiGates.

After the initial configuration, you can add more downstream FortiGates to the Security Fabric, and they are automatically configured with default values for a service provider.

To set up basic SAML SSO for the Security Fabric:

1. Log in to the root FortiGate of the Security Fabric.
2. Go to *Security Fabric > Fabric Connectors* and double-click the *Security Fabric Setup* card.
3. Join two pre-authorized FortiGates to the root FortiGate (see Configuring the root FortiGate and downstream FortiGates on page 4302).

4. Configure the IdP (see Configuring the root FortiGate as the IdP on page 4457).
5. Configure the SPs (see Configuring a downstream FortiGate as an SP on page 4458).

Navigating between Security Fabric members with SSO

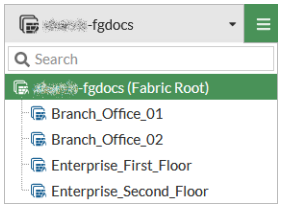
After you have logged in to a Security Fabric member by using SSO, you can navigate between any Security Fabric member with SSO configured. This can be done using the Security Fabric members dropdown menu or by logging in to a FortiGate SP from the root FortiGate IdP.

Security Fabric members dropdown

The Security Fabric members dropdown menu allows you to easily switch between all FortiGate devices that are connected to the Security Fabric. You can also use this menu to customize a FortiGate in the Security Fabric.

To navigate between Security Fabric members:

1. Log in to a Security Fabric member by using SSO.
2. In the top banner, click the name of the device you are logged into with SSO.
   A list of Security Fabric members is displayed.



3. Click the Security Fabric member.
   You are logged in to the Security Fabric member without further authentication.

To customize a FortiGate in the Security Fabric:

1. In the Security Fabric members dropdown menu, hover the cursor over a FortiGate so the tooltip is shown.
2. Click *Configure*. The *Configure* pane opens.
3. Edit the settings as required.
4. Click *OK*.

Logging in to an SP from the root IdP

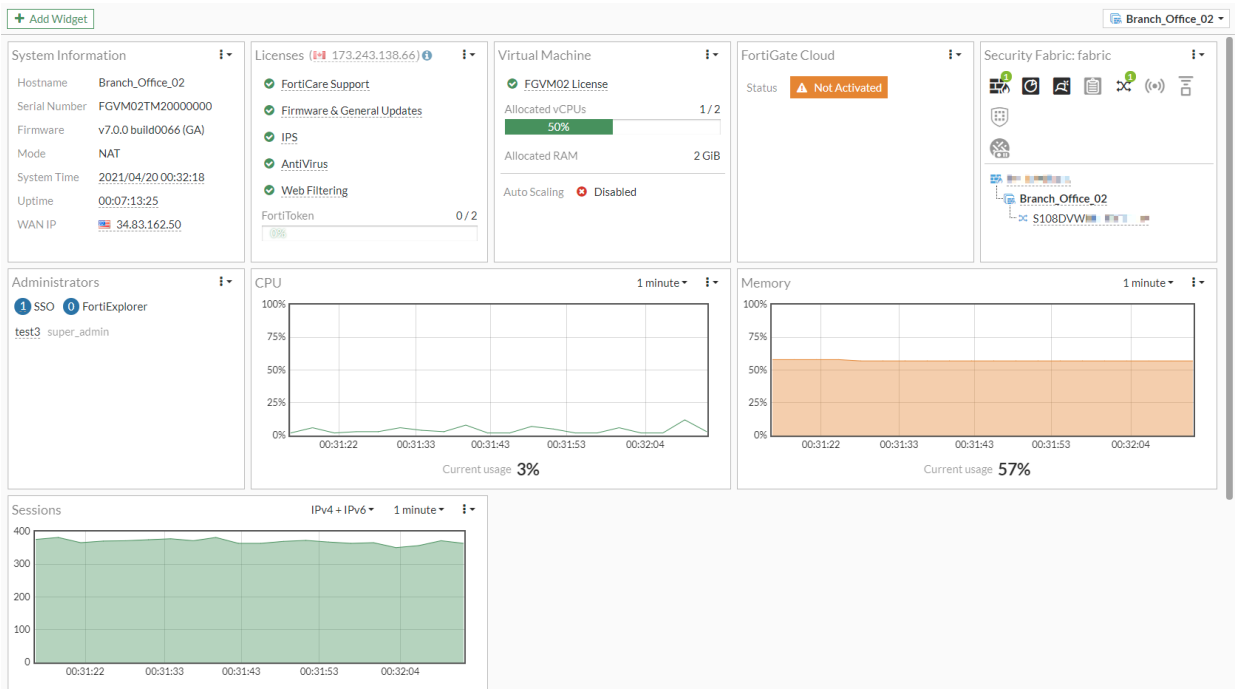
The following example describes how to log in to a root FortiGate IdP, and navigate to other FortiGate SPs in the Security Fabric without further authentication. The local administrator account is named *test3*. The local administrator account must also be available as an SSO administrator account on all downstream FortiGate SPs. Different tabs of the same browser are used to log in to the various FortiGates.

To log in to a FortiGate SP from a root FortiGate IdP:

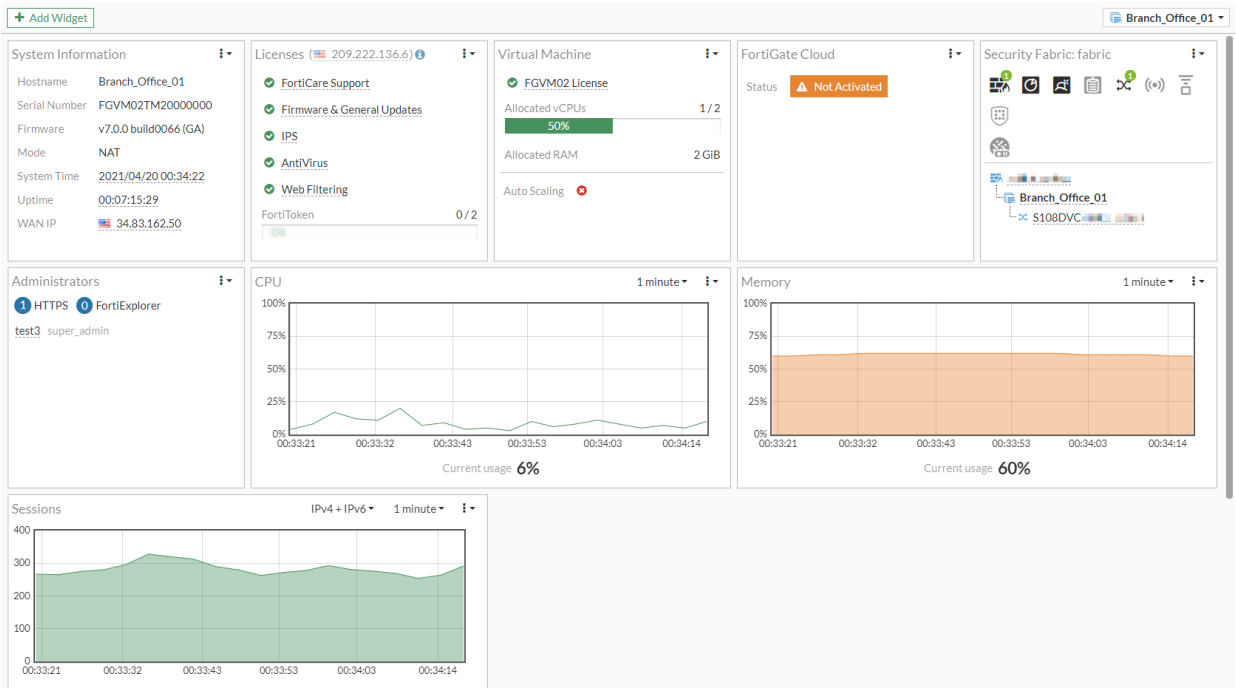
1. Log in to the root FortiGate IdP by using the local administrator account.
   In this example, the local administrator account is named *test3*.

2. Go to *Dashboard* > *Status* and locate the *Security Fabric* widget.
3. In the topology tree, click one of the downstream FortiGate SPs, and select *Login to <name of FortiGate>*. The login screen is displayed.
4. In the login screen, select *Single Sign-On*.

   By using cookies in your local browser for the already-authenticated SSO administrator, FortiGate logs you in to the downstream FortiGate SP as the SSO administrator. In this example, the SSO administrator name is *test3*.



5. While still logged into the root FortiGate IdP in your browser, go to the browser tab for the root FortiGate IdP, and log in to another FortiGate SP that is displayed on the *Security Fabric* widget in the GUI.



SAML SSO login uses *SAML_IDP* session cookies of already authenticated admin users in your local browser cache to send to the root FortiGate IdP for authentication. If your browser cache is manually cleared, or you close your browser, you must authenticate again.

It is possible to log in to one downstream FortiGate SP in a Security Fabric, and then open another tab in your browser to connect to another FortiGate SP that is not a member of the Security Fabric.

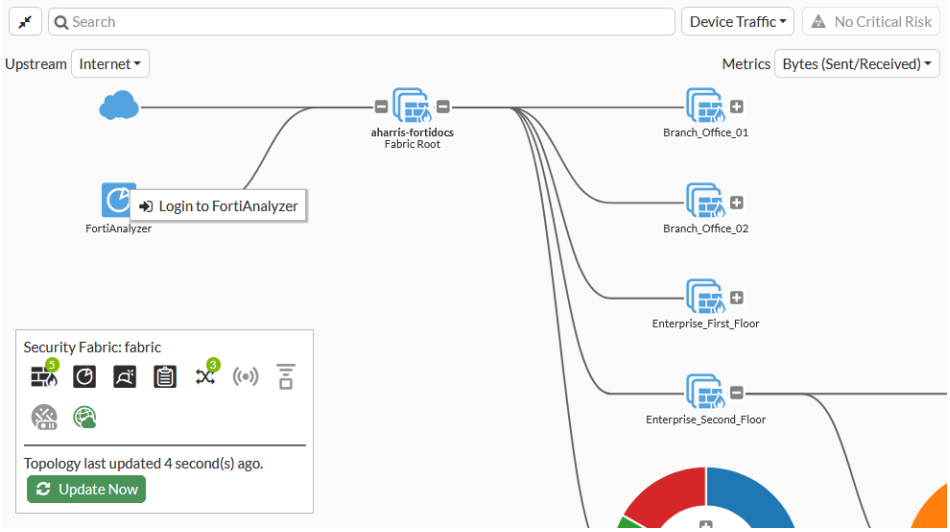
This is useful in cases where the SSO administrator and the local system administrator on the FortiGate SP both have the same login name, but are two different entities.

Integrating FortiAnalyzer management using SAML SSO

When a FortiGate acting as a Security Fabric root is configured as a SAML SSO identity provider (IdP), the FortiAnalyzer of the Security Fabric can register itself as a service provider (SP). This simplifies the configuration by enabling the setting in FortiAnalyzer to facilitate Fabric SSO access to the FortiAnalyzer once authenticated to the root FortiGate. When signed in using SSO, the FortiAnalyzer includes a Security Fabric navigation dropdown, which allows easy navigation to FortiGates in the Fabric.

To enable FortiAnalyzer as a Fabric SP in the GUI:

1. On the root FortiGate, go to *Security Fabric* > *Physical Topology* or *Logical Topology*.
2. In the topology, click the *FortiAnalyzer* icon and select *Login to FortiAnalyzer*.



3. Enter the credentials to log in. A Security Fabric must be configured with the Fabric devices listed under the Fabric name.
4. See Enabling SAML authentication in a Security Fabric in the FortiAnalyzer Administration Guide for more details.

To enable FortiAnalyzer as a Fabric SP in the CLI:

1. In FortiAnalyzer, enable the device as a Fabric SP:

```
config system saml
    set status enable
    set role FAB-SP
    set server-address "192.168.1.99"
    set user-auto-create enable
end
```

   FortiAnalyzer will register itself on the FortiGate as an appliance.
2. Verify the configuration in FortiOS:

```
show system saml
config system saml
    set status enable
    set role identity-provider
    set cert "fortigate.domain.tld"
    set server-address "192.168.1.99"
    config service-providers
        edit "appliance_192.168.1.103"
            set prefix "csf_76sh0bm4e7hf1ty54w42yrrv88tk8uj"
            set sp-entity-id "http://192.168.1.103/metadata/"
            set sp-single-sign-on-url "https://192.168.1.103/saml/?acs"
            set sp-single-logout-url "https://192.168.1.103/saml/?sls"
            set sp-portal-url "https://192.168.1.103/saml/login/"
            config assertion-attributes
                edit "username"
                next
```

```
                edit "profilename"
                    set type profile-name
                next
            end
        next
    end
end
```

To navigate between devices using SAML SSO in FortiOS:

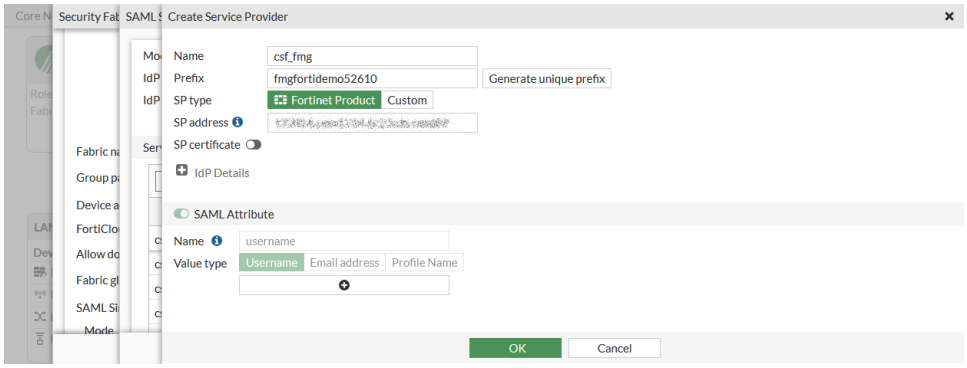
1. Log in to the root FortiGate.
2. Go to *Security Fabric* > *Physical Topology* or *Logical Topology*.
3. In the topology, click the *FortiAnalyzer* icon and select *Login to FortiAnalyzer*.

Integrating FortiManager management using SAML SSO

When a FortiGate is configured as the SAML SSO IdP, FortiManager can be added as an SP.

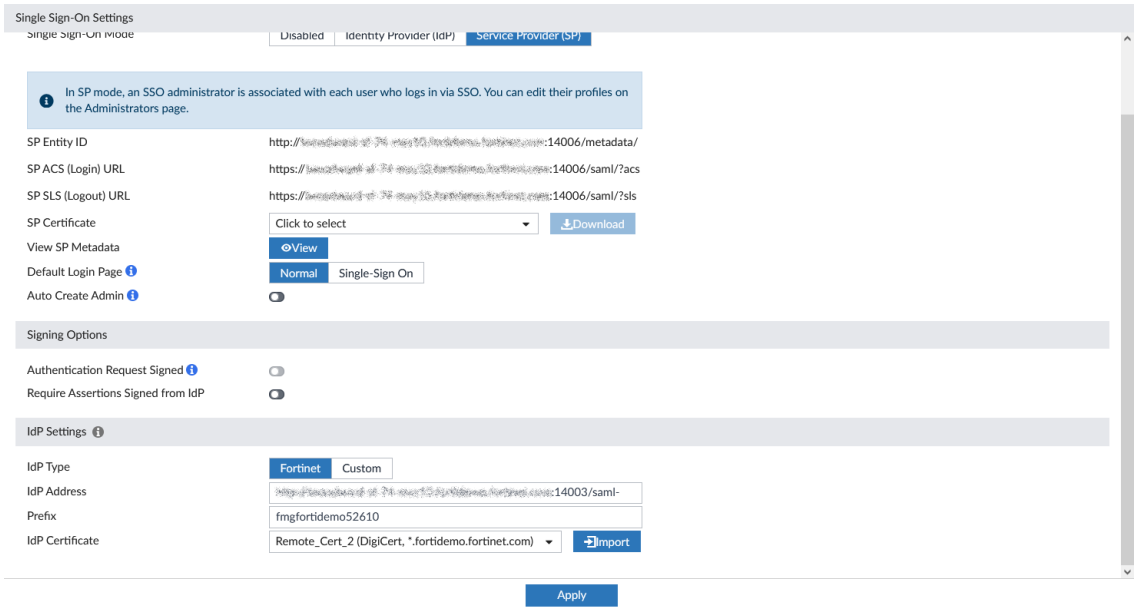
To configure FortiManager as a Fabric SP:

1. On the root FortiGate, go to *Security Fabric* > *Fabric Connectors*, and edit the *Security Fabric Setup* connector.
2. In the *SAML Single Sign-On* section, click *Advanced Options*.
3. In the *Service Providers* section, click *Create New*.
4. Enter a name and a prefix for the SP. FortiOS generates a unique prefix, but you can enter your own.
5. In *SP address*, enter the FortiManager address including the port number.

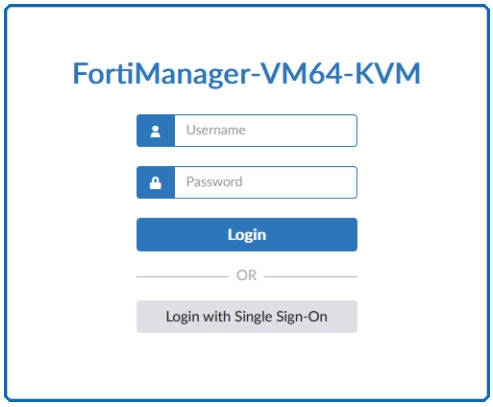


6. Click *OK*.
7. In FortiManager, go to *System Settings* > *SAML SSO* and in the *Single Sign-On Mode* section, click *Service Provider (SP)*.
8. Configure the *IdP Settings*:
   a. For *IdP Type*, click *Fortinet*.
   b. For *IdP Address*, enter the root FortiGate address including the port number.
   c. Enter the *Prefix* of the SP.
   d. For *IdP Certificate*, import the same certificate used on the root FortiGate.

e. Click *Apply*.



9. To verify that the configuration works, log out of FortiManager and log in using the *Login via Single-Sign-On* link.

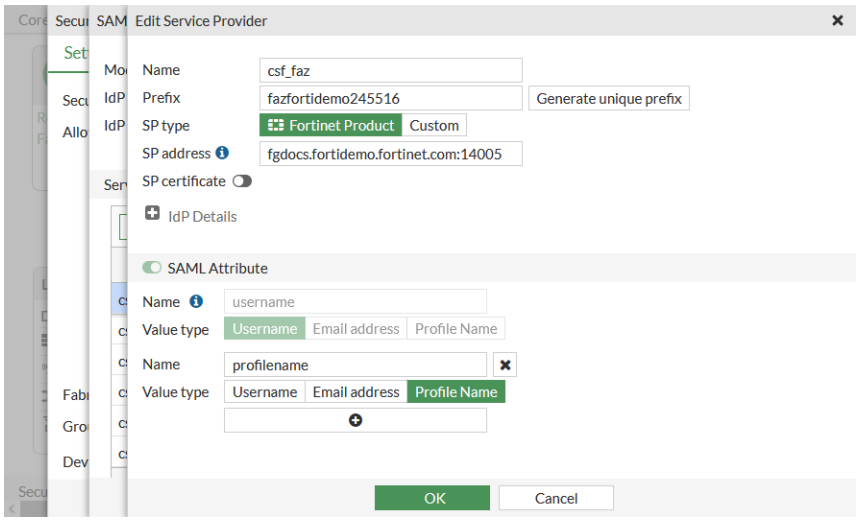


Advanced option - FortiGate SP changes

From a root FortiGate IdP, you can edit each of the FortiGate SPs. For example, you can edit a FortiGate SP to generate a new prefix, or you can add or modify SAML attributes. When you generate a new prefix value, it is propagated to the respective downstream FortiGates.

To edit an SP from the root FortiGate (IdP):

1. Go to *Security Fabric* > *Fabric Connectors* and double-click the *Security Fabric Setup* card.
2. Click *Advanced Options*. The *SAML SSO* pane opens.
3. In the *Service Providers* table, select a device and click *Edit*. The *Edit Service Provider* pane opens.



4. Edit the settings as needed.
5. Click *OK*.

Security rating

The security rating uses real-time monitoring to analyze your Security Fabric deployment, identify potential vulnerabilities, highlight best practices that can be used to improve the security and performance of your network, and calculate Security Fabric scores.

The *Security Fabric* > *Security Rating* page contains the following tabs:

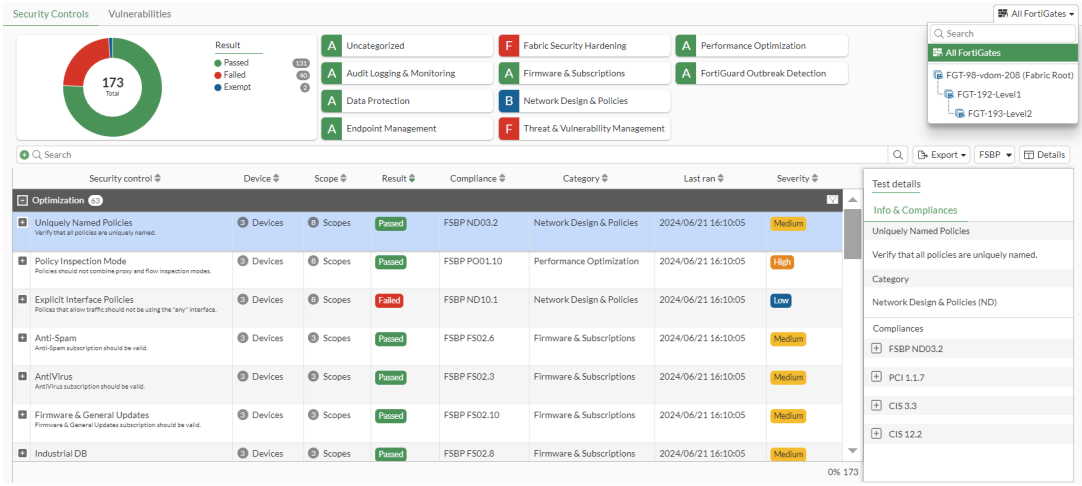
- Security Controls on page 4470
- Vulnerabilities on page 4475

This section also includes:

- Security Fabric score on page 4477
- Security rating insights on page 4478
- Logging the security rating on page 4480
- Security rating visibility on page 4480
- Security rating tooltip controls on page 4484

Security Controls

On the *Security Fabric* > *Security Rating* page, the *Security Controls* tab displays the results for all security rating checks:

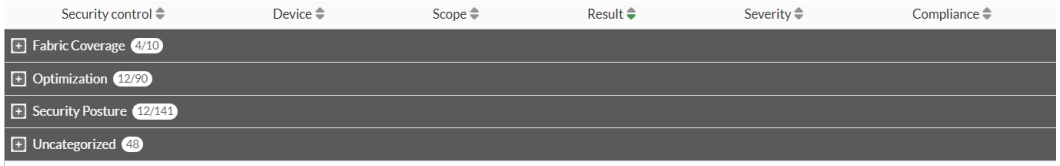


- A summary chart identifies of how many items passed, failed, or are exempt from security rating checks.
- Categories against which the security rating tests were run are identified (such as *Uncategorized*, *Audit Logging & Monitoring*, *Data Protection*, and so on) and the assigned grade for each category is displayed (such as *A*, *B*, *F*, and so on).

  The letter grade is calculated based on the percent of tests in a category that passed:

  - A = 90% and above
  - B = 80% to <90%
  - C = 70% to <80%
  - D = 60% to <70%
  - E = 50% to <60%
  - F = Less than 50%

  For example, if eight out of ten tests in a category passed, then 80% of the tests passed, and the category would be given a B grade.
- A report of details that is organized in the following security rating categories: *Fabric Coverage* (available for devices in a Security Fabric), *Optimization*, *Security Posture*, and *Uncategorized*. Expand each category to display rows of test details. Click each row to display the *Test Details* pane.



The following licensing options are available for security rating checks:

- A base set of free checks
- A licensed set that requires a FortiGuard Security Rating Service subscription

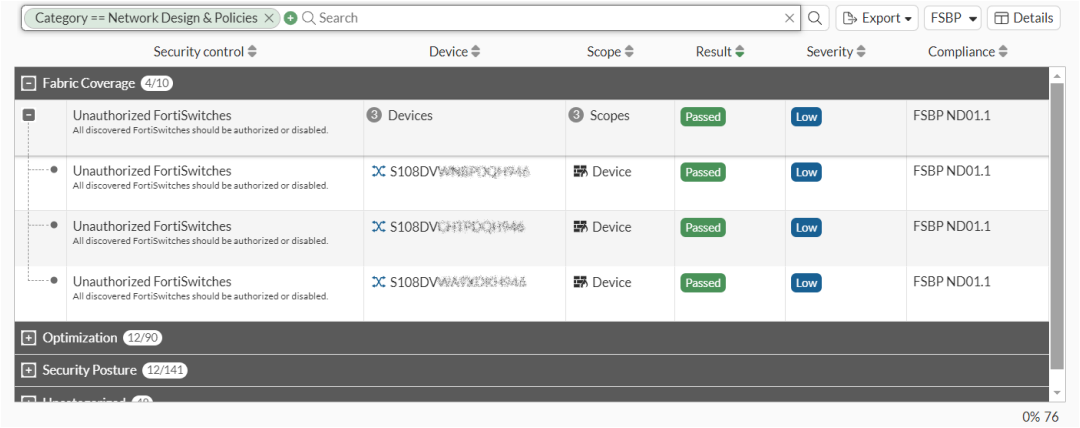
The base set of free checks can be run locally on FortiGate. Full details about security rating including the free and licensed checks available can be found in the Security Rating Reference Guide.

To view security controls:

1. On the root FortiGate, go to *Security Fabric > Security Rating*. The *Security Controls* pane opens.
2. For the graded test categories, hover the cursor over a test category to view the calculation breakdown.
3. For the summary chart, click the *Passed*, *Failed*, or *Exempt* words or associated colors in the chart to filter the report results.

   For example, click *Failed* to display only failed tests in the report.

   Click *Result* to remove the filter, or click the *X* beside the filter in the *Search* bar
4. Expand each security rating category in the report to view its details.

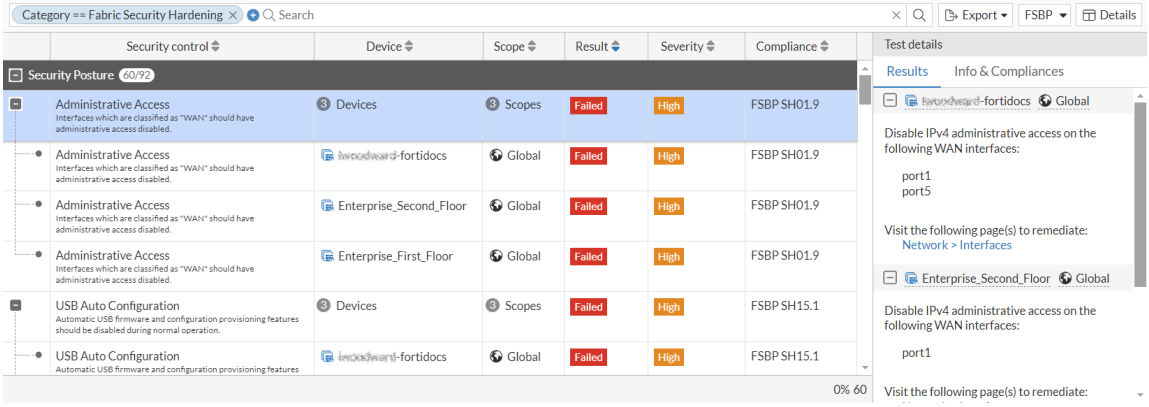


5. In the report, click each row to view its *Test details* pane, which includes two tabs: *Results* and *Info & Compliance*.

   Click *Details* to hide and display the *Test details* pane for a selected row in the report.

   If a test category failed, the *Results* section includes a link to the GUI page where you can resolve the problem.

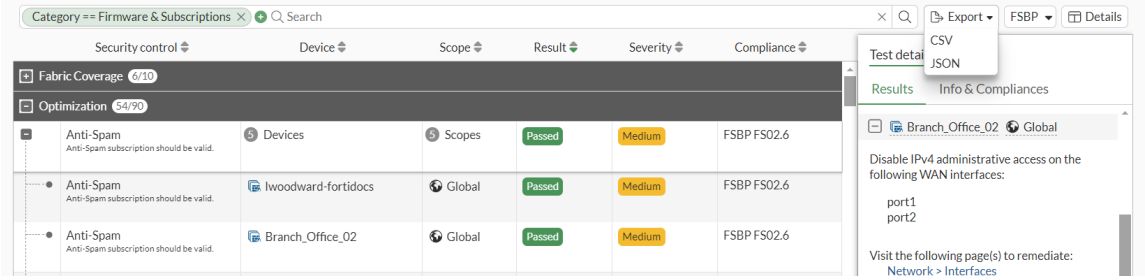
   The *Info & Compliance* tab includes the security controls used for the test and links to specific FSBP, PCI, or CIS compliance policies.



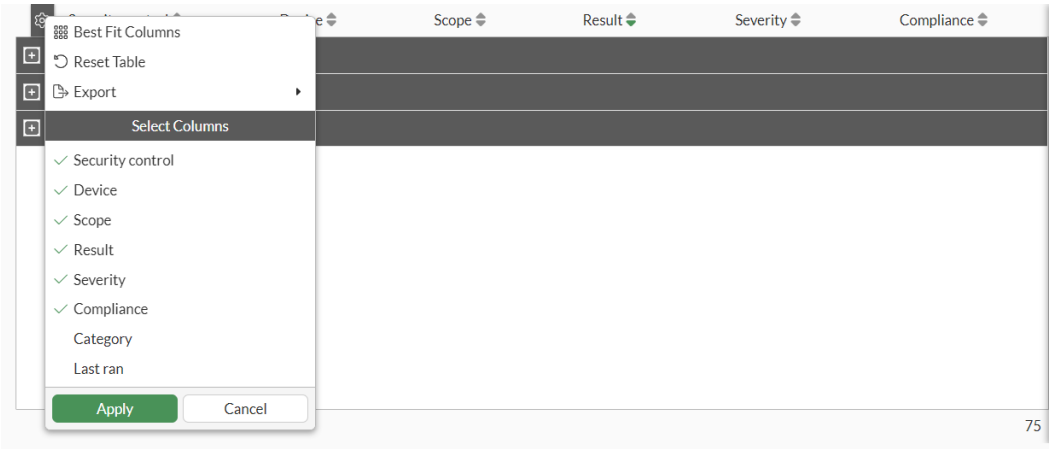
6. Select *FSBP*, *PCI*, or *CIS* to filter the report for the selected compliance policy.

   The FortiGate must have a valid Attack Surface Security Rating license to view security ratings grouped by CIS.

7. Click *Export* to export the report to a CSV or JSON file.



8. Click the gear icon to customize the report table by adding more columns.



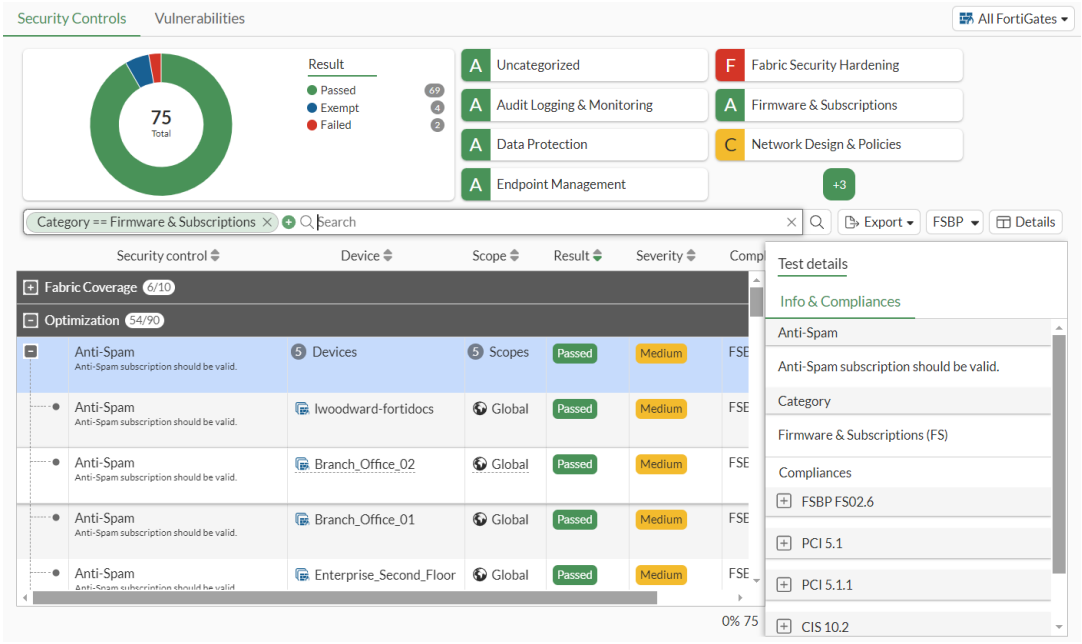
To exit the current view, click the x beside the search item to return to the summary view.

Multi-VDOM mode

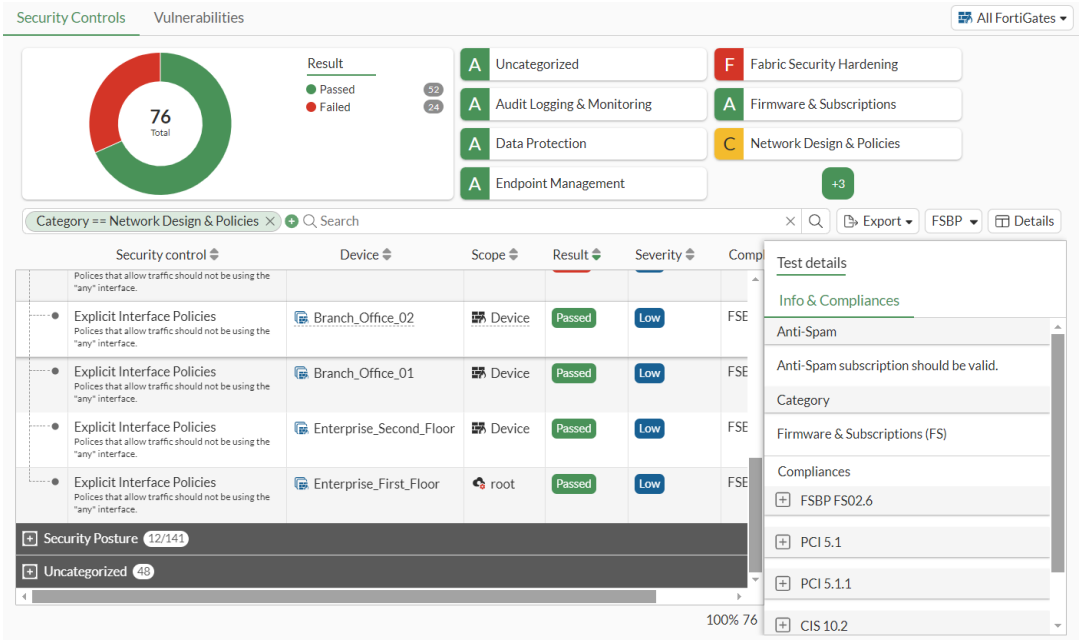
In multi-VDOM mode, security rating reports can be generated in the Global VDOM for all of the VDOMs on the device. Administrators with read/write access can run the security rating report in the Global VDOM. Administrators with read-only access can only view the report.

On the report scorecards, the *Scope* column shows the VDOMs that the check was run on.

Global scope:



VDOM scope:



The security rating event log is available on the root VDOM.

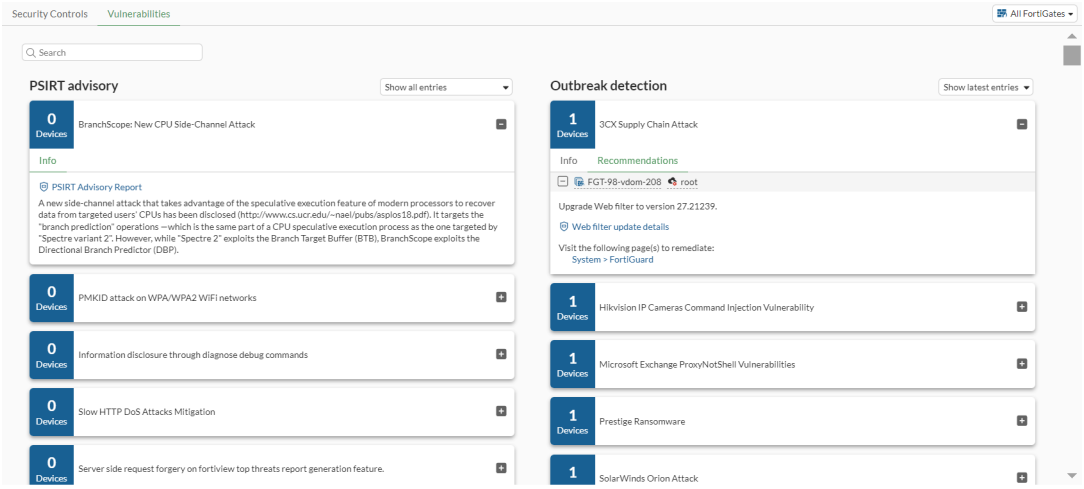
Security rating check scheduling

By default, security rating checks are scheduled to run automatically every eight hours.

The security rating also runs automatically for only the specific check when the relevant configuration changes are made.

Vulnerabilities

On the *Security Fabric > Security Rating* page, the *Vulnerabilities* tab displays *PSIRT advisory* and *Outbreak detection* entries that are included in the downloaded Security Rating package:

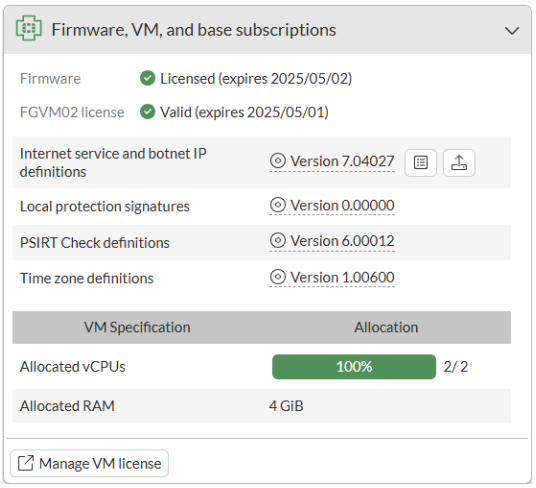


PSIRT-related notifications

On a FortiGate with a valid Firmware license, the separate Security Rating package downloaded from FortiGuard supports PSIRT vulnerabilities, which are highlighted in security rating results. PSIRT Package Definitions are part of the Firmware entitlement.

To verify the FortiGuard license entitlement in the GUI:

1. Go to *System > FortiGuard* and expand the *License Information* table.
2. Expand the *Firmware & General Updates* section.
3. Check that *PSIRT Check definitions* appears in the list and the license is valid.

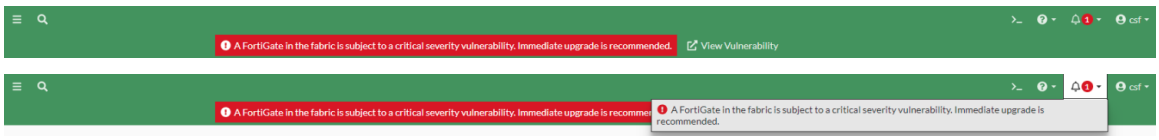


To verify the FortiGuard license entitlement in the CLI:

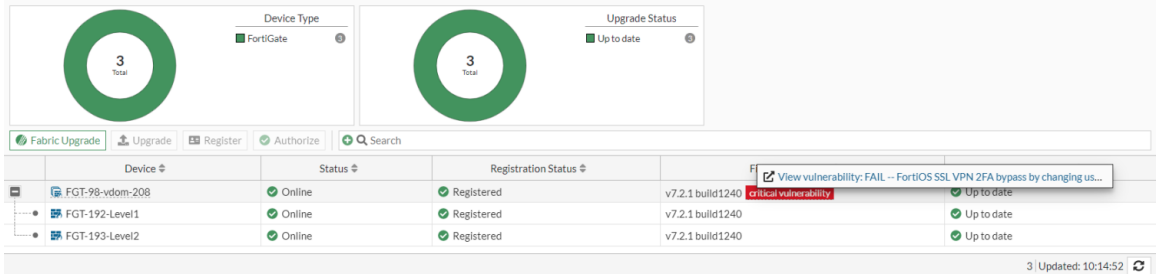
```
# diagnose autoupdate versions
...
Security Rating Data Package
---------
Version: 5.00022
Contract Expiry Date: Fri Nov 24 2023
Last Updated using scheduled update on Mon Sep 11 09:44:21 2023
Last Update Attempt: Tue Sep 12 16:29:10 2023
Result: No Updates
```

The following notifications are visible in the GUI.

- Warning message: if the security rating result indicates a vulnerability with a critical severity, then the FortiOS GUI displays a warning message in the header and a new notification under the bell icon. The *View Vulnerability* link appears in the header for global administrators.



  Clicking the warning message redirects to the *System > Firmware & Registration* page, where administrators are encouraged to update any affected Fortinet Fabric devices to the latest firmware releases to resolve the critical vulnerabilities.
- Tooltip: a tooltip for the *critical vulnerability* label on the *System > Firmware & Registration* page lists the vulnerability, and it links to the *Security Fabric > Security Rating* page where more details about the vulnerability are displayed.



To view vulnerability results after performing security rating scan:

```
# diagnose report-runner vuln-read
Index: 0
Name: FG-IR-23-001: FortiOS / FortiManager / FortiAnalyzer / FortiWeb / FortiProxy /
FortiSwitchManager - Heap buffer underflow in administrative interface
FortiGate Serial: FGVM02TM23000000
```

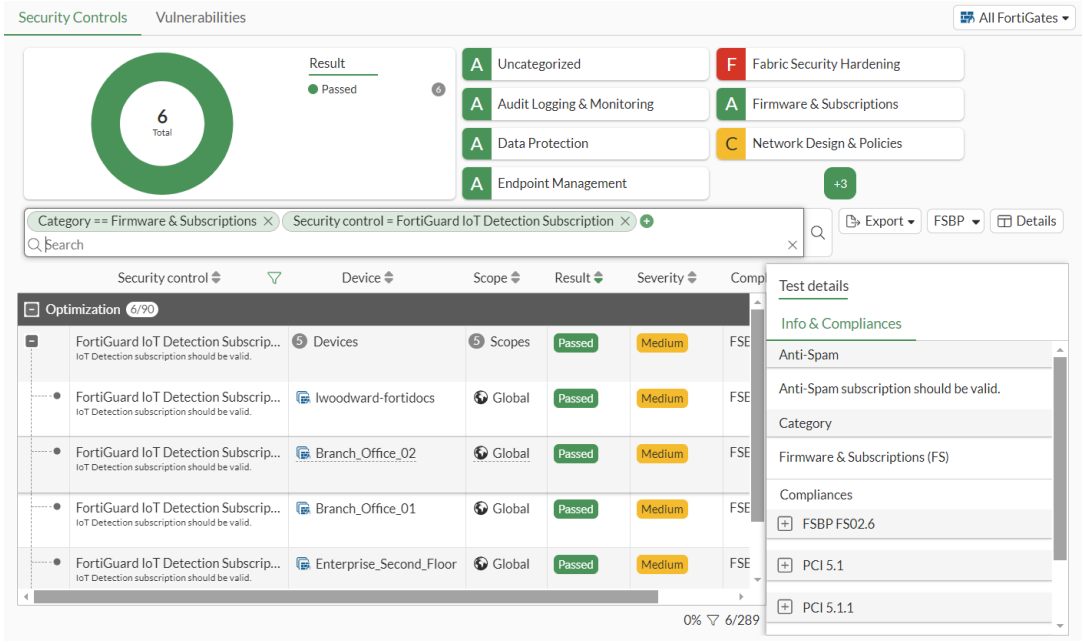
To clear the vulnerability result:

```
# diagnose report-runner vuln-clean
Deleted temporary critical vulnerability file
```

FortiGuard IoT vulnerability-related checks

There are two rating checks in the *Security Posture* report related to IoT vulnerabilities:

- The *FortiGuard IoT Detection Subscription* rating check will pass if the *System > FortiGuard* page shows that the *IoT Detection Definitions* (under the *Attack Surface Security Rating* entitlement) is licensed. In this example, the result is marked as *Passed* because the license is valid.



- The *FortiGuard IoT Vulnerability* rating check will fail if any IoT vulnerabilities are found.
  Hover over the device name to display the tooltip, which includes an option to *View IoT Vulnerabilities*.

To detect IoT vulnerabilities, the FortiGate must have a valid IoT Definitions license, device detection must be configured on a LAN interface used by IoT devices, and a firewall policy with an application control sensor must be configured.

Security Fabric score

The Security Fabric score is calculated when a security rating check is run, based on the severity level of the checks that are passed or failed. A higher scores represents a more secure network. Points are added for passed checks and removed for failed checks.

Severity level	Weight (points)
Critical	50
High	25
Medium	10
Low	5

To calculate the number of points awarded to a device for a passed check, the following equation is used:

$$\text{score} = \frac{\text{<severity level weight>}}{\text{<\# of FortiGates>}} \times \text{<secure FortiGate multiplier>}$$

The secure FortiGate multiplier is determined using logarithms and the number of FortiGate devices in the Security Fabric.

For example, if there are four FortiGate devices in the Security Fabric that all pass the compatible firmware check, the score for each FortiGate device is calculated with the following equation:

$$\frac{50}{4} \times 1.292 = 16.15 \text{ points}$$

All of the FortiGate devices in the Security Fabric must pass the check in order to receive the points. If any one of the FortiGate devices fails a check, the devices that passed are not awarded any points. For the device that failed the check, the following equation is used to calculated the number of points that are lost:

$$\text{score} = \text{<severity level weight>} \times \text{<secure FortiGate multiplier>}$$

For example, if the check finds two critical FortiClient vulnerabilities, the score is calculated with the following equation:

$$-50 \times 2 = -100 \text{ points}$$

Scores are not affected by checks that do not apply to your network. For example, if there are no FortiAP devices in the Security Fabric, no points will be added or subtracted for the FortiAP firmware version check.

Security rating insights

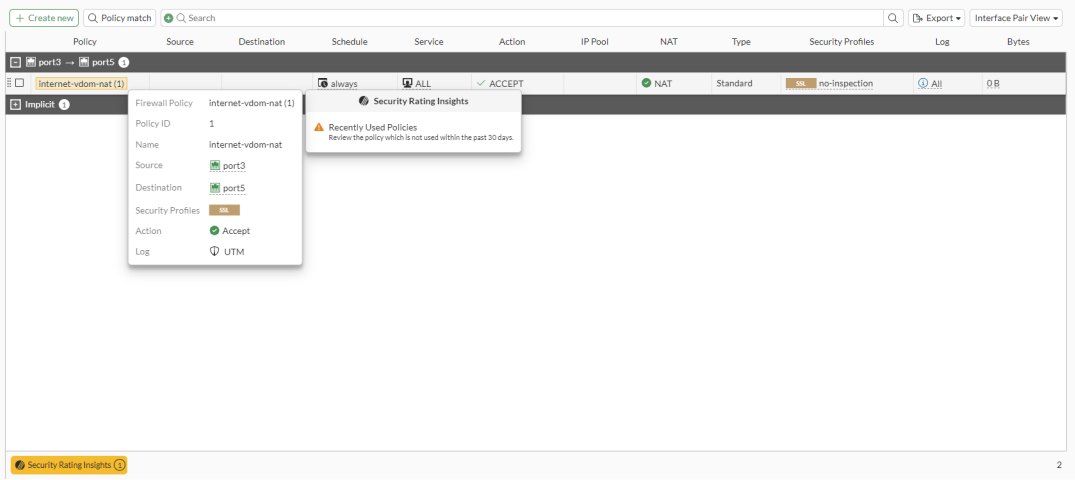
Objects, such as firewall policies or interfaces, with security rating recommendations are highlighted in the GUI. Hover over highlighted objects to display a *Security Rating Insights* tooltip. A *Security Rating Insights* button also displays at the bottom of the table.

Dismissed issues are unique for each administrator. Hashes for dismissed notifications are saved in local storage. If a user clears the local storage, all issues will show up again as not dismissed.

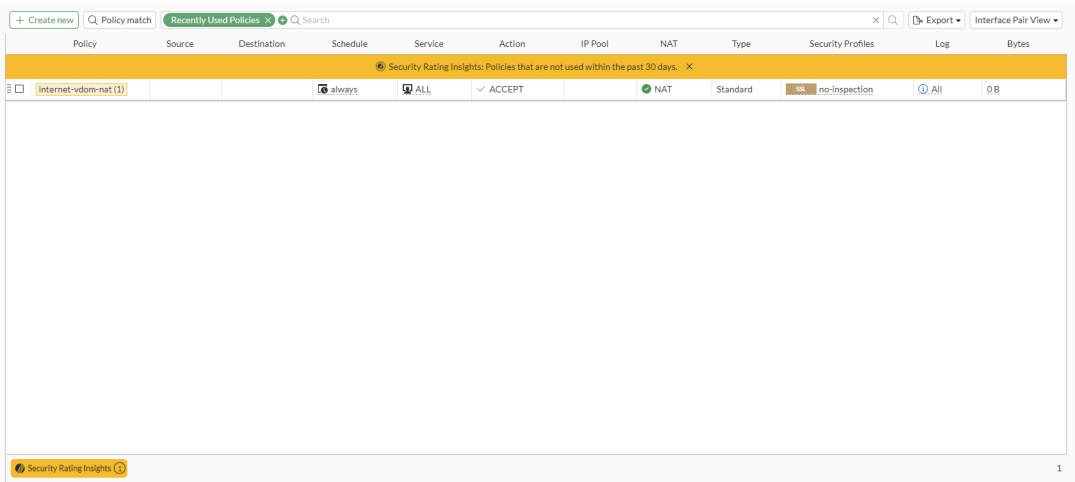
Once an issue is resolved, the notification disappears after the next security rating report runs.

To view security rating insights for firewall policies:

1. Go to *Policy & Objects* > *Firewall Policy*.
   Objects with security rating recommendations are highlighted.
2. Hover over any highlighted object to display a tooltip that shows *Security Rating Insights*:



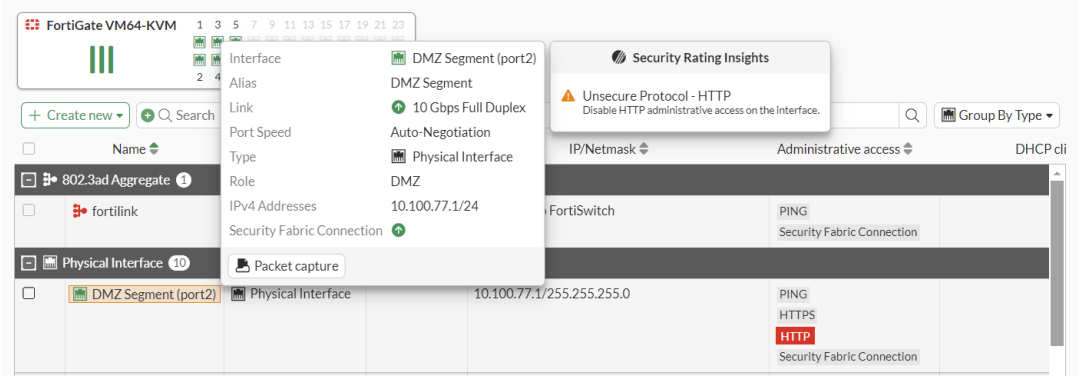
3. Below the table, click *Security Rating Insights* to display relevant issues.
4. Select an issue, such as *Unused Policies* to display a banner and filter that you can use to filter down to the applicable entries.



5. Click the *X* in the banner to close the issue.

To view security rating insights for network interfaces:

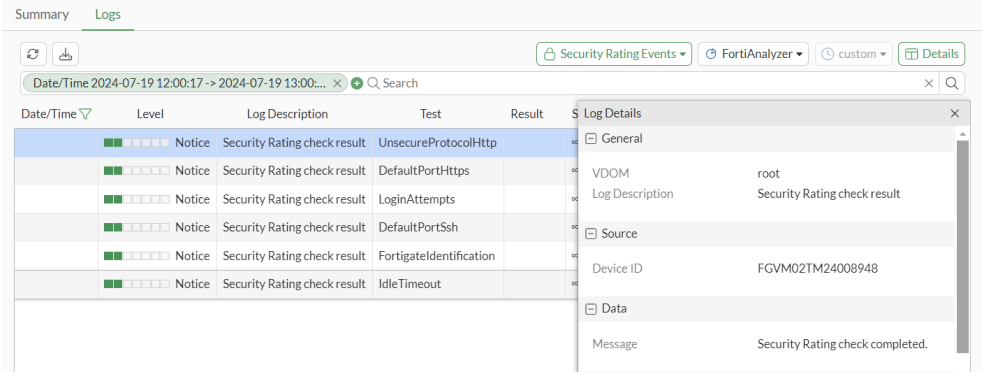
1. Go to *Network* > *Interfaces*.
   Objects with security rating recommendations are highlighted.
2. Hover over any highlighted object to display a tooltip that shows *Security Rating Insights*:



3. Below the table, click *Security Rating Insights* to display relevant issues.
4. Select an issue to display a banner and filter that you can use to filter down to the applicable entries.
5. Click the *X* in the banner to close the issue.

Logging the security rating

The results of past security checks are available on the *Log & Report > System Events* page. Click the *Security Rating Events* card to see the detailed log.



An event filter subtype can be created for the Security Fabric rating so event logs are created on the root FortiGate that summarize the results and show detailed information for the individual tests.

To configure security rating logging using the CLI:

```
config log eventfilter
    set security-rating enable
end
```

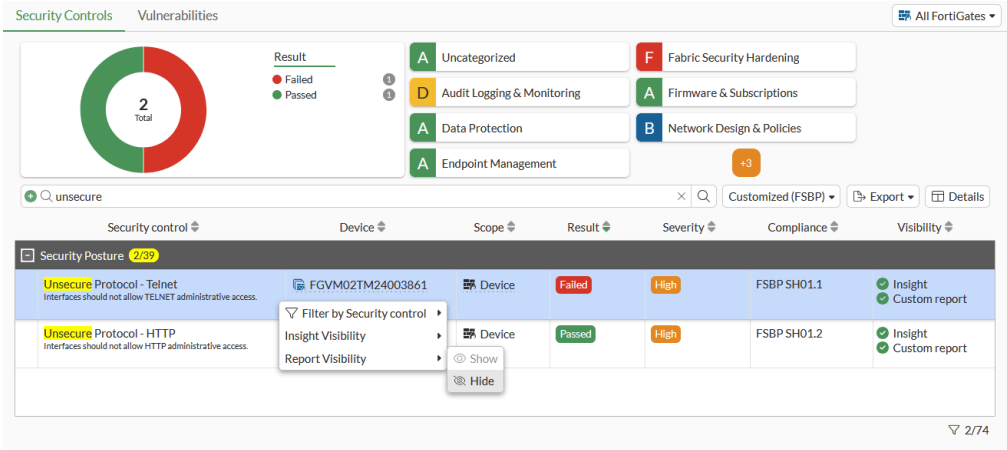
Security rating visibility

Security ratings tests that are not relevant can be hidden, streamlining the user experience by displaying only pertinent information.

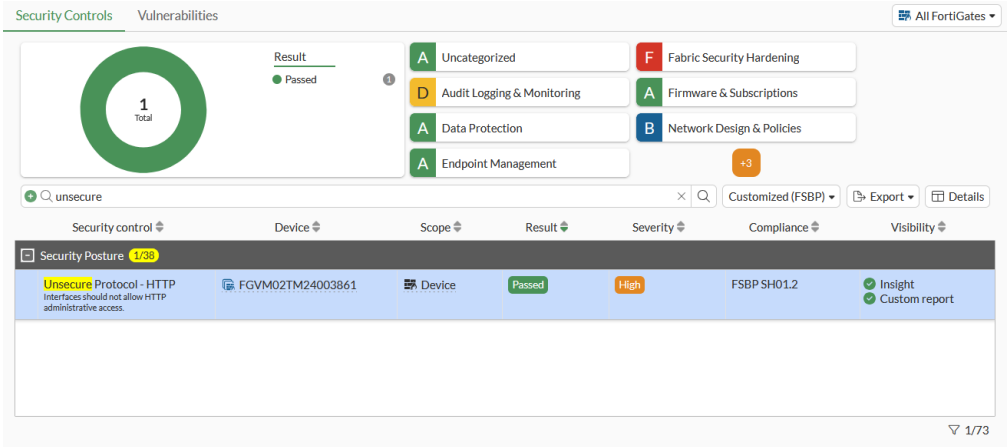
A Security Fabric is not required for this feature. If multiple FortiGates are in a Security Fabric, hidden security ratings can be synchronized from the root FortiGate device to downstream FortiGate devices, or overridden locally on the downstream devices.

To disable showing a security rating control in the default report:

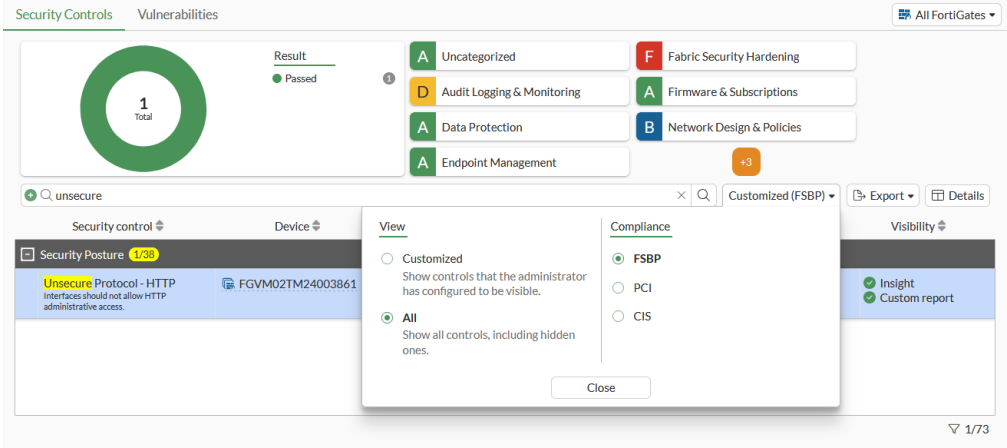
1. Go to *Security Fabric > Security Rating*.
2. Find *Unsecure Protocol - Telnet* in the table, right-click on it, and select *Report Visibility > Hide*.



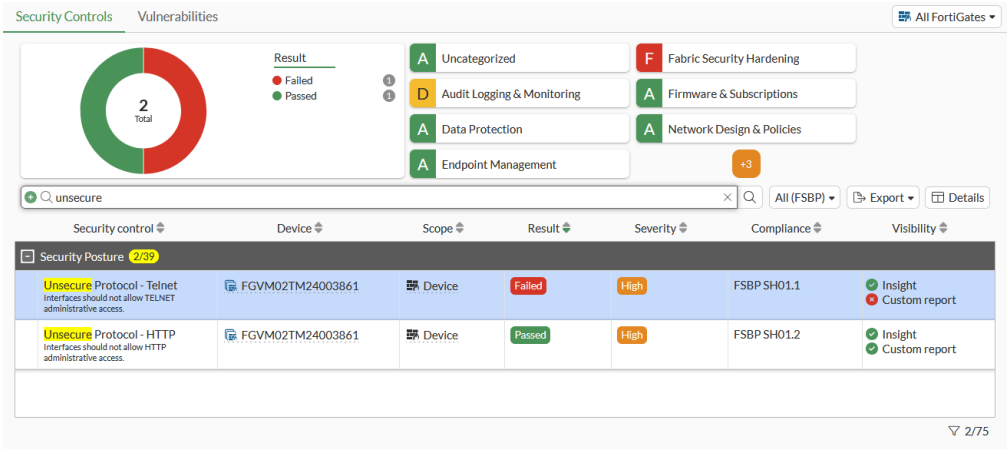
The *Unsecure Protocol - Telnet* control is no longer shown in the list.



3. Change the *View* to *All* to show the *Unsecure Protocol - Telnet* control in the table when *Report Visibility* is set to *Hide*.



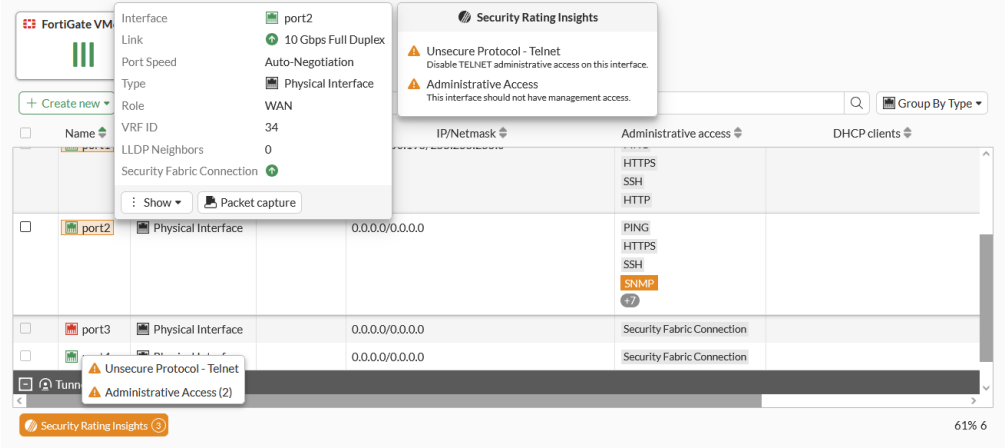
All hidden controls are then shown in the table.



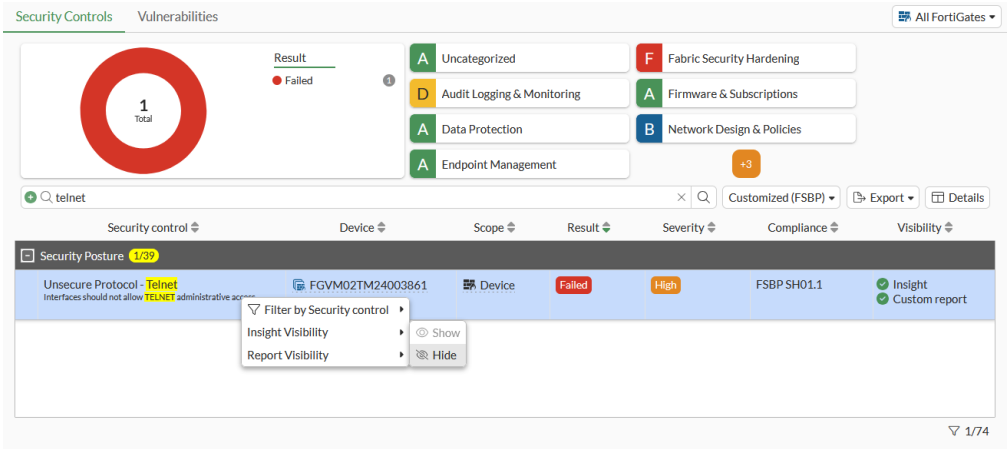
The *Visibility* column is not shown by default. Edit the column settings to show it in the table. See Column settings on page 58 for details.

To disable showing a security rating control as an insight across the GUI:

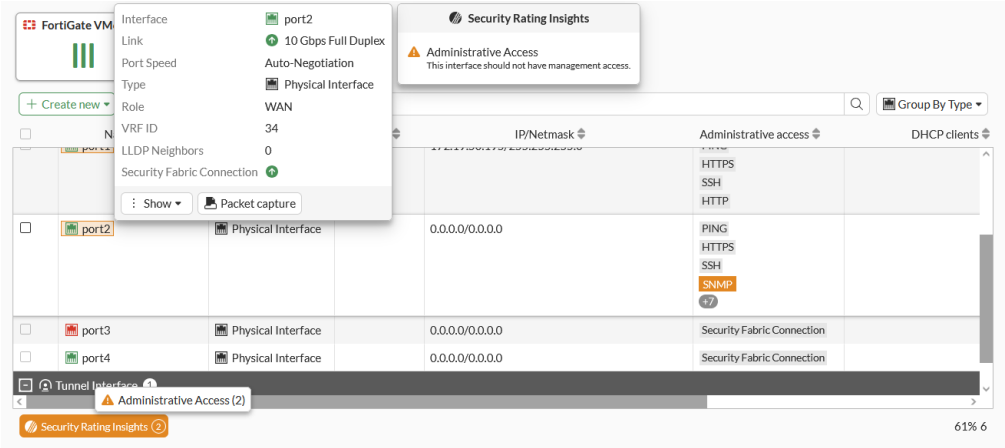
1. Go to *Network* > *Interfaces*.
2. Edit an interface to enable *TELNET* administrative access.
3. Click on the *Security Rating Insights*, or hover over the interface name to see the *Unsecure Protocol - Telnet* warning.



4. Go to *Security Fabric* > *Security Rating*.
5. Find *Unsecure Protocol - Telnet* in the table, right-click on it, and select *Insight Visibility* > *Hide*.



6. Go to *Network* > *Interfaces*.
7. Click on the *Security Rating Insights*, or hover over the interface name to see that the *Unsecure Protocol - Telnet* warning is no longer shown.



To configure security rating control in the CLI:

```
config system security-rating controls
   edit <control name>
      set display-report {enable | disable}
      set display-insight {enable | disable}
    next
end
```

Variable	Description
`display-report {enable | disable}`	Enable/disable displaying the Security Rating control in the default report (default = enable).
`display-insight {enable | disable}`	Enable/disable displaying the Security Rating control as an insight across the GUI (default = enable).

Security rating control names are hidden in the CLI until they are configured.

To configure synchronizing security rating visibility settings on downstream devices in a Security Fabric:

```
config system csf
    set configuration-sync {default | local}
end
```

```
config system security-rating settings
    set override-sync {enable | disable}
end
```

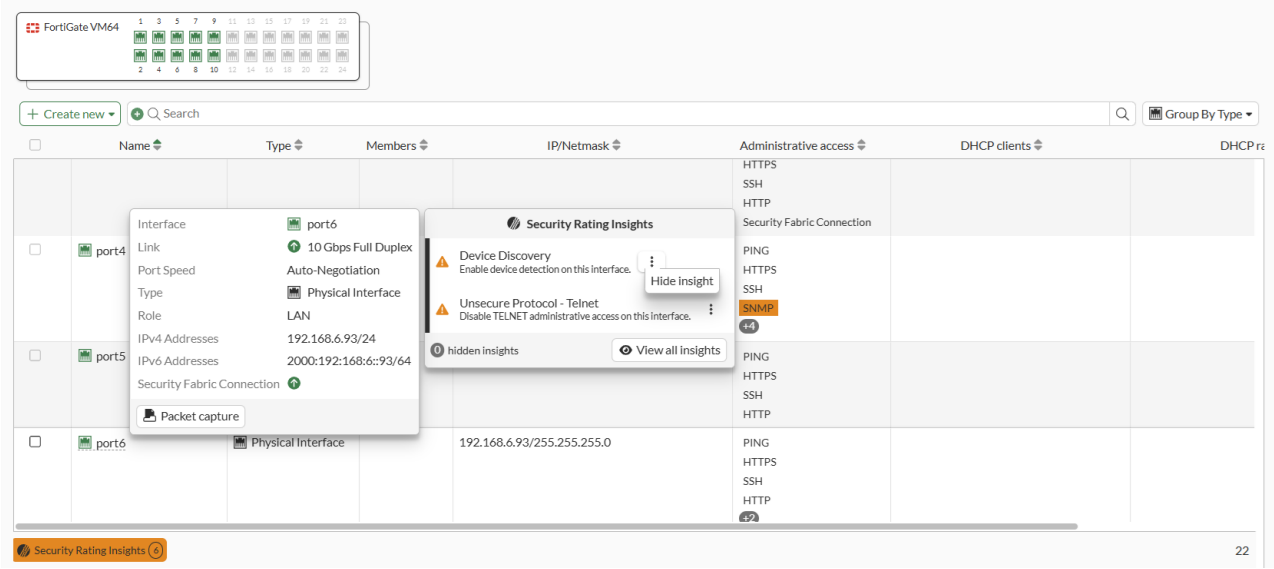
Variable	Description
`configuration-sync {default | local}`	Configuration sync mode. • `default`: Synchronize configuration for IPAM, FortiAnalyzer, FortiSandbox, and Central Management to root node (default). • `local`: Do not synchronize configuration with root node.
`override-sync {enable | disable}`	Enable/disable overriding Security Rating control settings that are synchronized from the Security Fabric's root FortiGate (default = disable).

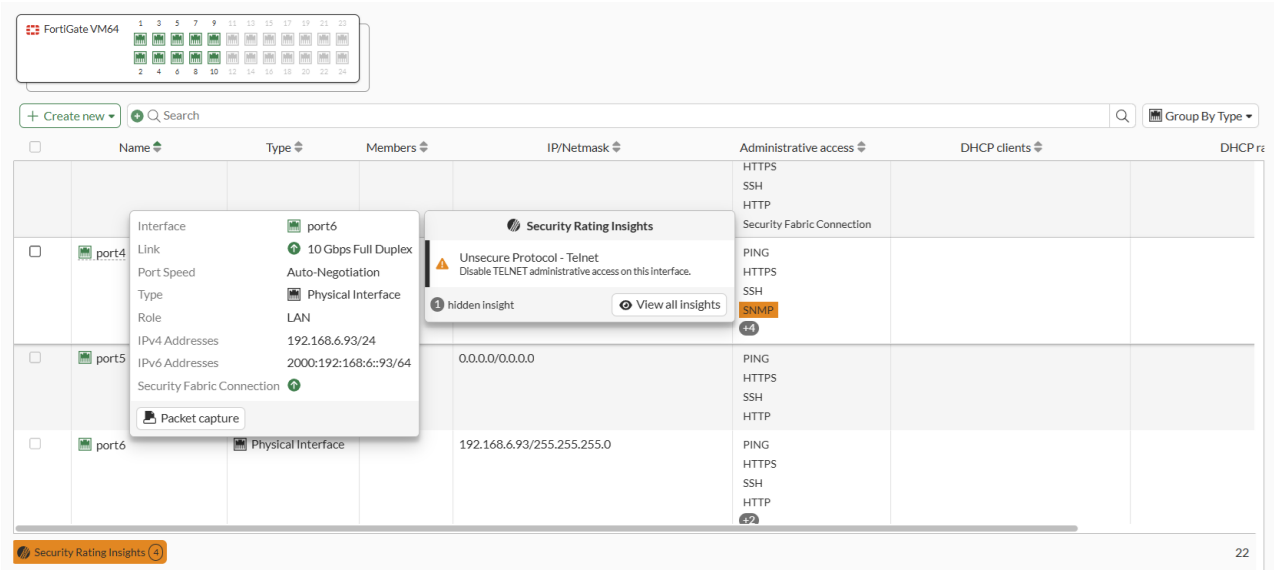
When `configuration-sync` is set to `local`, the `system security-rating settings` command is not available.

Security rating tooltip controls

Security Rating tooltips include a footer button to view all insights for a configuration object, plus individual controls to hide specific insights directly from the tooltip. Hidden insights are still indicated, improving visibility and user control.

In the Security Rating Insight, tooltips to allow actions for the individual insights so they can be hidden directly from the tooltip.

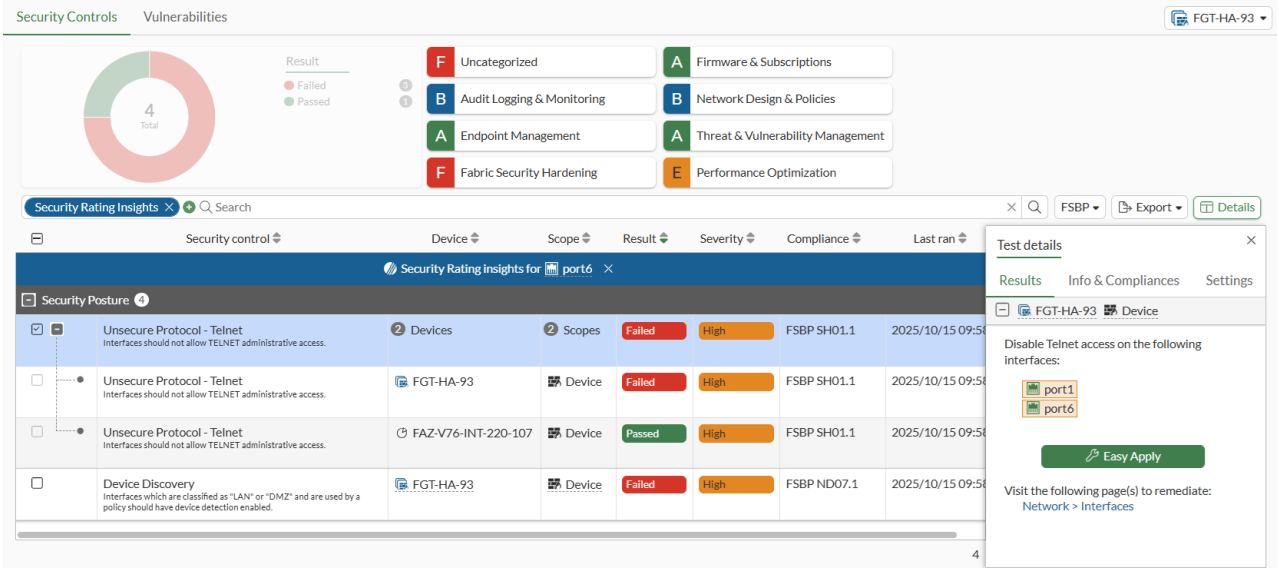




The insights can also be hidden or unhidden using the CLI:

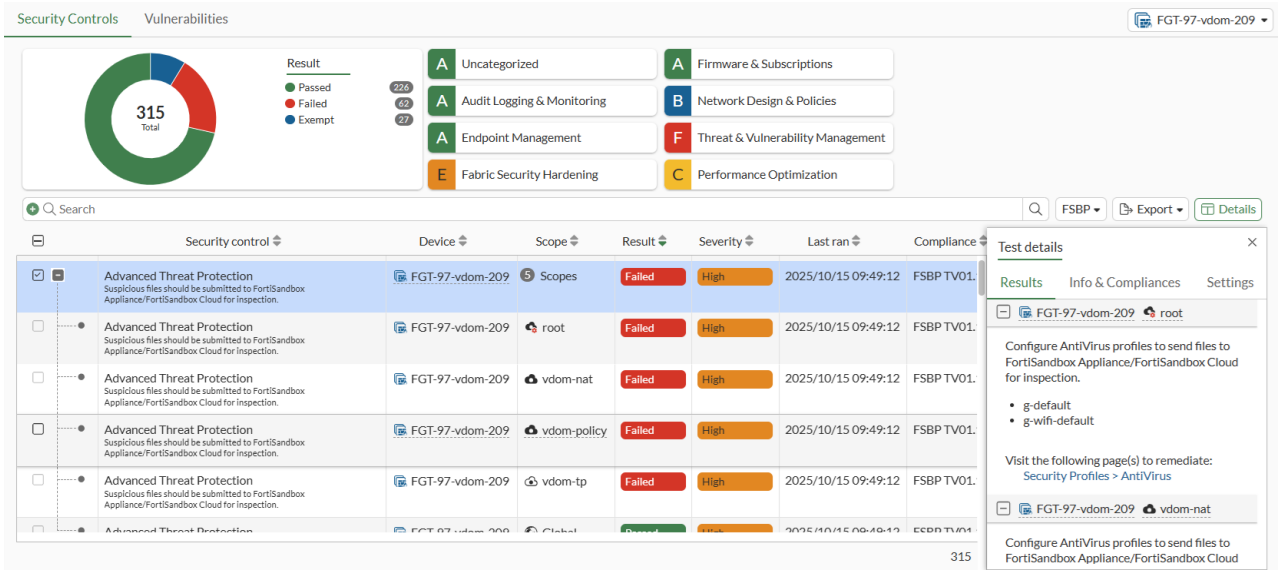
```
config system security-rating controls
    edit "DeviceDiscovery"
        set display-insight {enable | disable}
    next
end
```

In the tooltip, the *View all insights* button will open the Security Rating page, where a special filter is pre-applied that shows all the insights (hidden or not) for the specific entry.

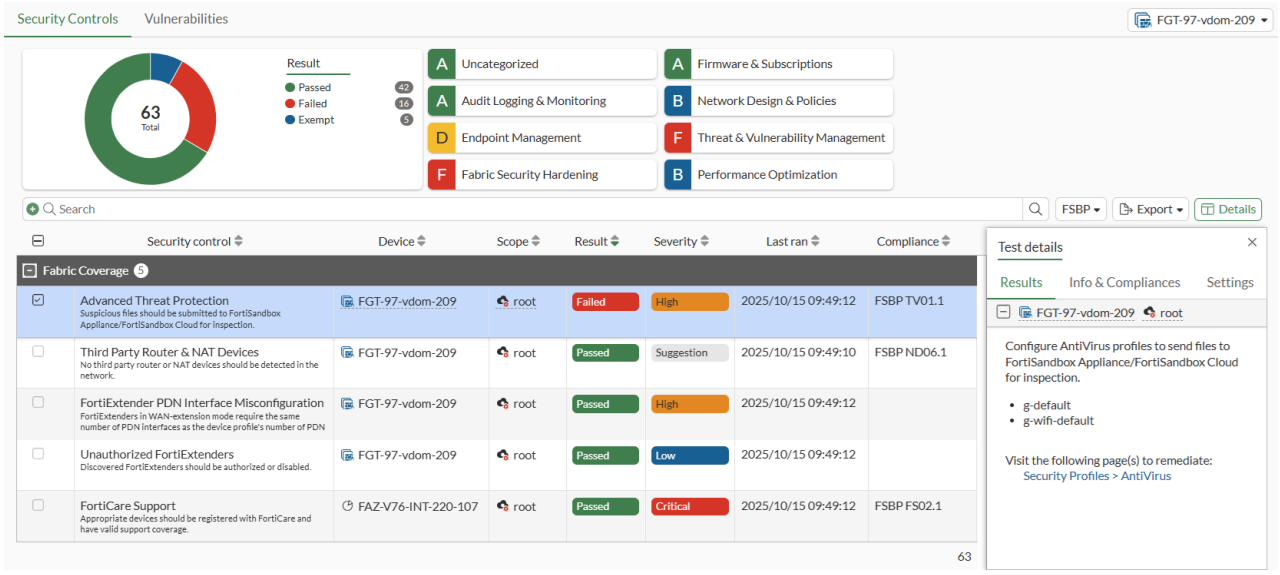


Security Rating is visible in the VDOM scope for global administrators, and is displayed in both the Global and VDOM scopes.

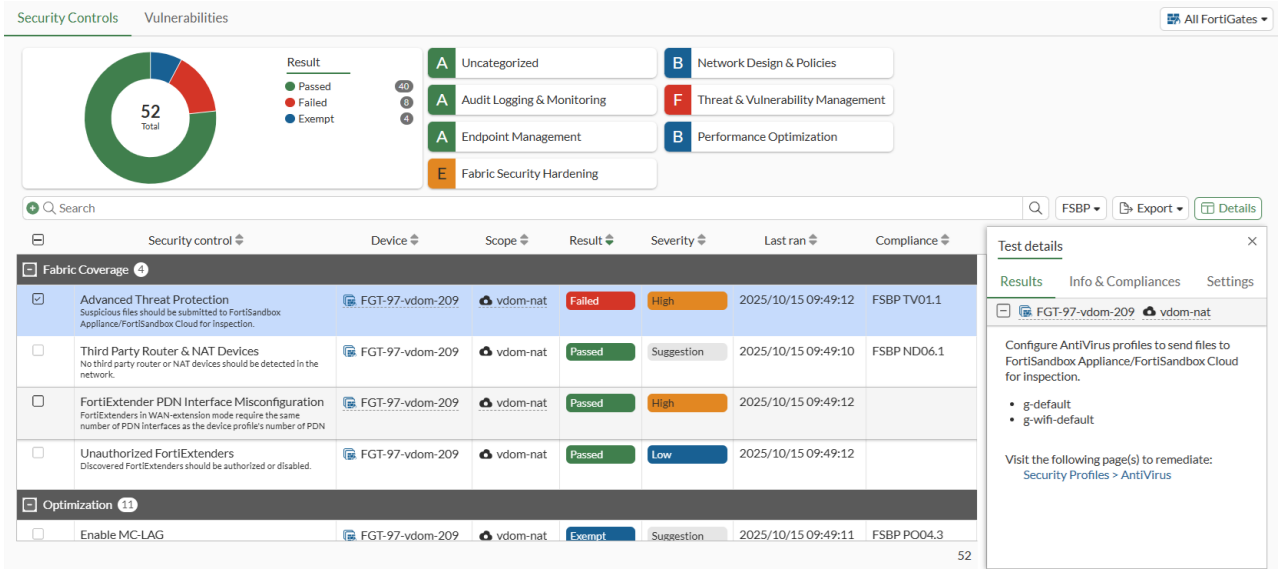
- Global



- Root VDOM



- VDOM



Automation stitches

Automation stitches automate the activities between the different components in the Security Fabric, which decreases the response times to security events. Events from any source in the Security Fabric can be monitored, and action responses can be set up to any destination.

The automation settings can be synchronized within the Security Fabric, or can only apply to an individual FortiGate in the Security Fabric. Automation stitches can only be created on the root FortiGate in a Security Fabric.

To configure automation setting synchronization in a Security Fabric:

```
config automation setting
    set fabric-sync {enable | disable}
end
```

Automation stitches can also be used on FortiGates that are not part of a Security Fabric.

An automation stitch consists of two parts: the trigger and the actions. The trigger is the condition or event on the FortiGate that activates the action, for example, a specific log, or a failed log in attempt. The action is what the FortiGate does in response to the trigger.

Automation stitches that use cloud-based actions (AWS Lambda, Azure Function, Google Cloud Function, and AliCloud Function) have the option to delay an action after the previous action is completed.

Diagnose commands are available in the CLI to test, log, and display the stitch history and settings.

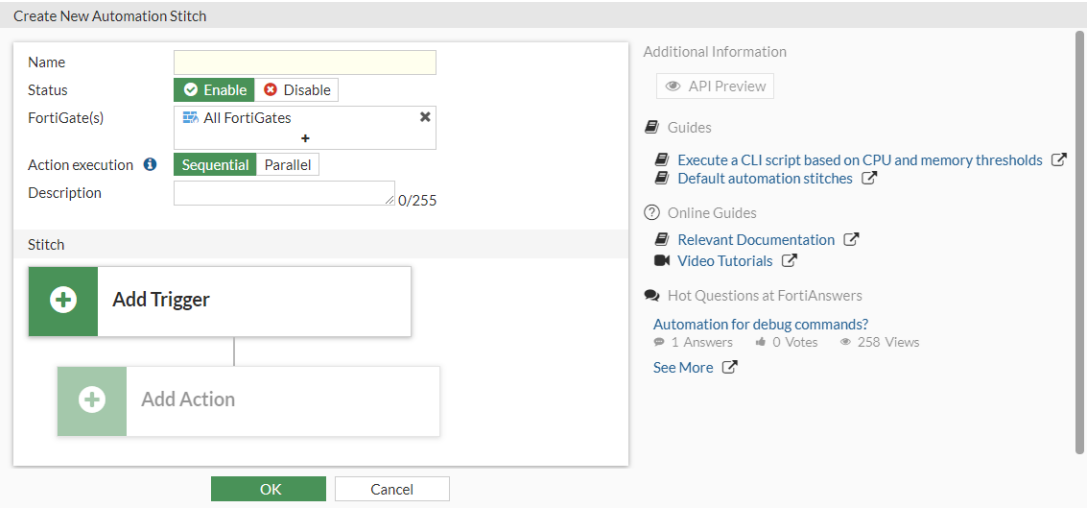
Creating automation stitches

To create an automation stitch, a trigger event and a response action or actions are selected. Automation stitches can be tested after they are created.

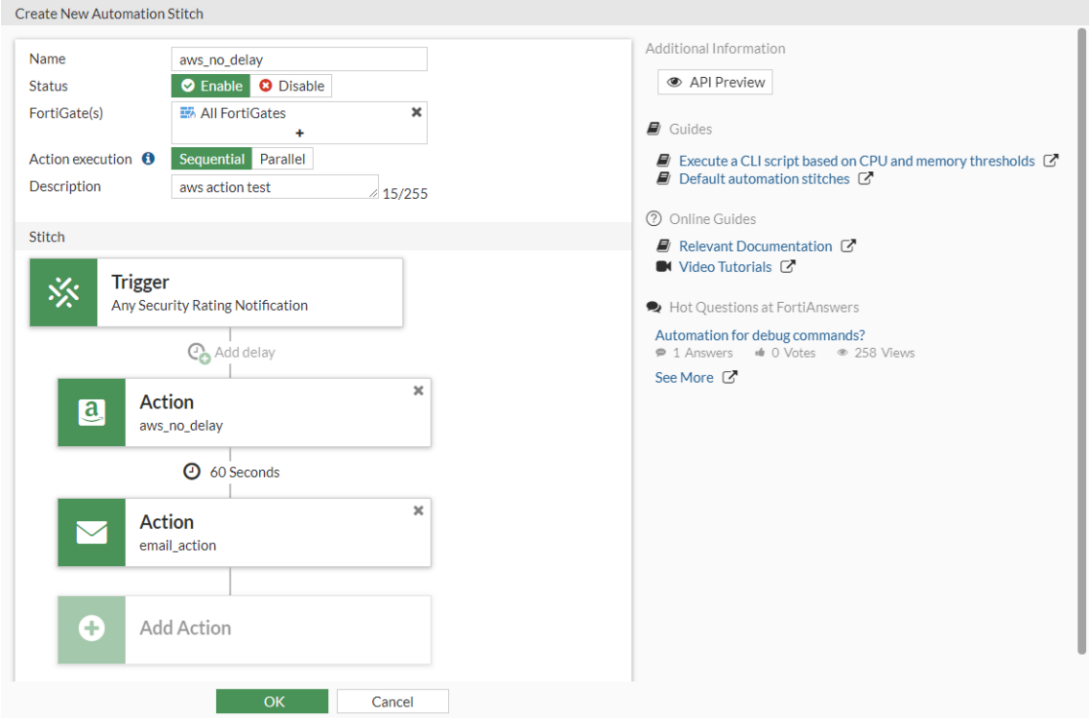
In the GUI, go to *Security Fabric* > *Automation* and click *Create New*. Automation stitches, actions, and triggers are configured in separate dialogs.

The stitch *Action execution* can be set to either *Sequential* or *Parallel*. In sequential execution, actions will execute one after another with a delay (if specified). If one action fails, then the action chain stops. This is the default setting. In parallel execution, all actions will execute immediately when the stitch is triggered.

When creating a stitch, clicking *Add Trigger* and *Add Action* displays a list of available triggers and actions, and the option to create a new one.



Once the stitch is configured, a process diagram of the trigger, actions, and delays is displayed. A delay can be added before an action if *Sequential* action execution is used. Executing the next action can be delayed by up to 3600 seconds (one hour).



Triggers and actions can be configured separately, and then added to an automation stitch.

The maximum number of automation stitches that are allowed to run concurrently can be configured in the CLI (32 - 256, default = 128).

To configure the maximum number of concurrent automation stitches:

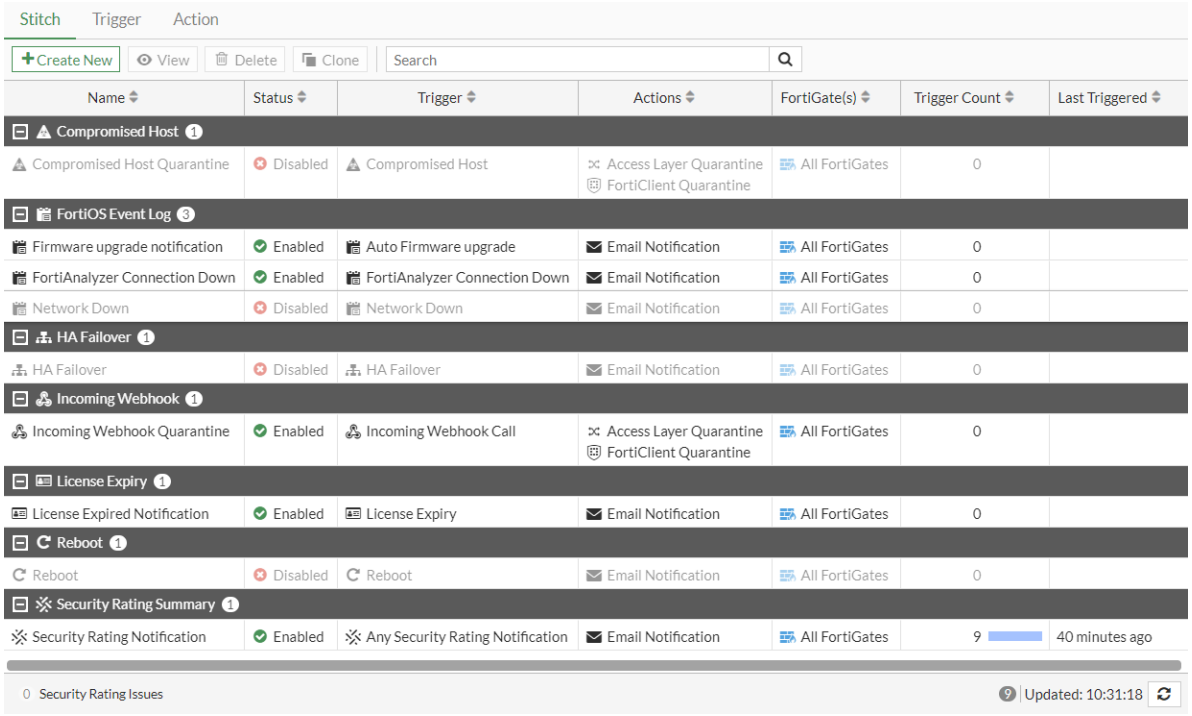
```
config automation setting
    set max-concurrent-stitches <integer>
end
```

Starting in FortiOS 7.6.4, administrators can use the CLI to control memory pool usage by the automation daemon. The reserved memory scales with the number of concurrent automation stitches to reduce the overall memory overhead across all FortiGate models.

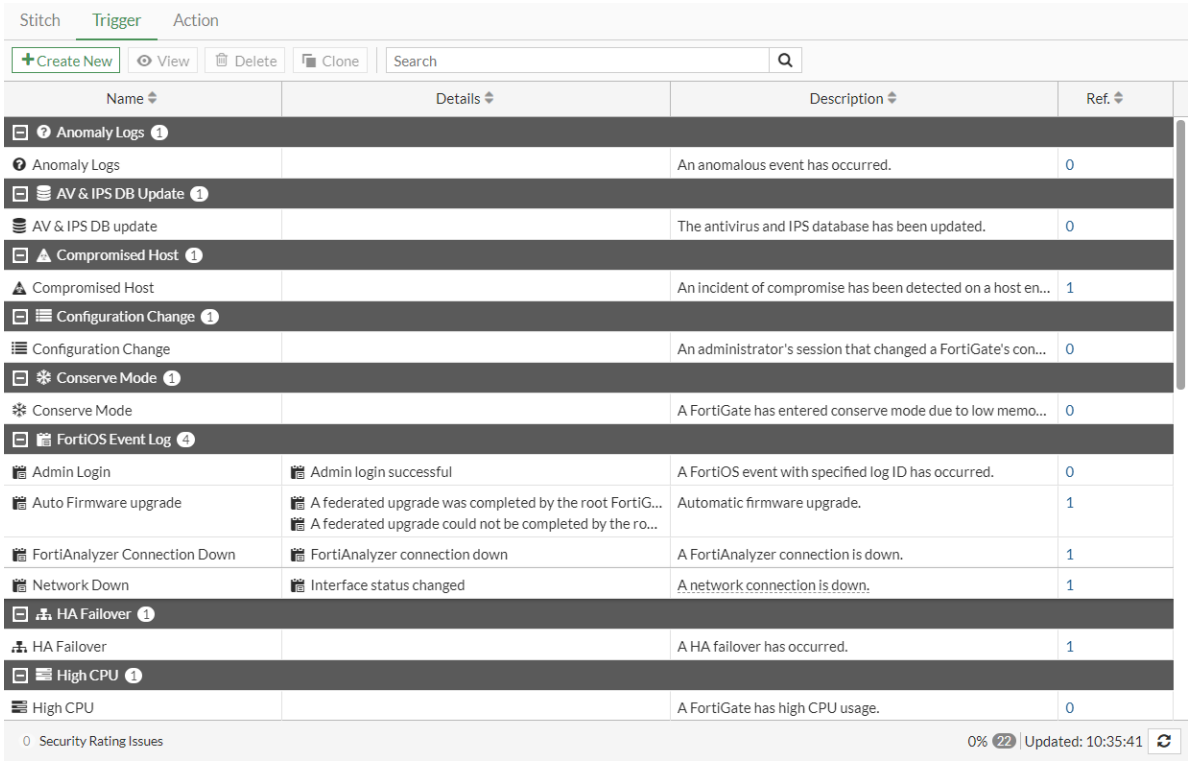
```
config automation setting
    set secure-mode {enable | disable}
end
```

Tabs on the Automation page

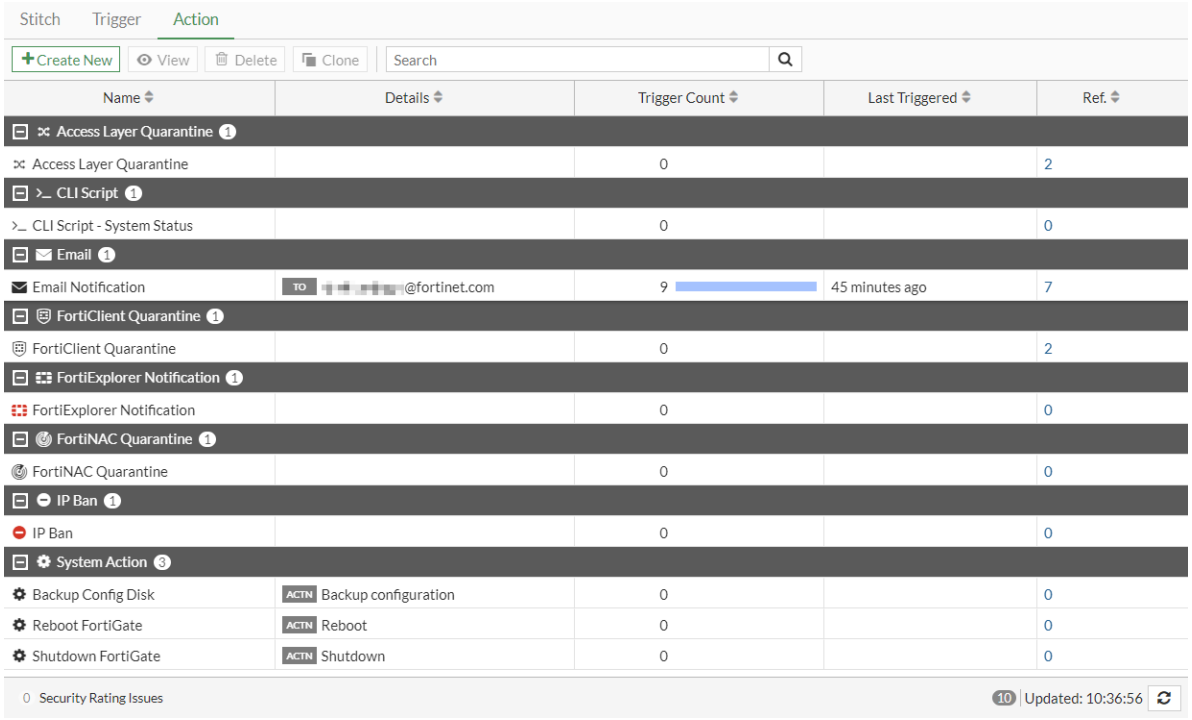
On the *Security Fabric > Automation* page, there are tabs for *Stitch*, *Trigger*, and *Action*. The *Stitch* tab is the default view that lists the trigger and actions used in each stitch. Individual triggers and actions can be created or edited in the corresponding tabs.



Click *Trigger* to view the list of triggers.



Click *Action* to view the list of actions.



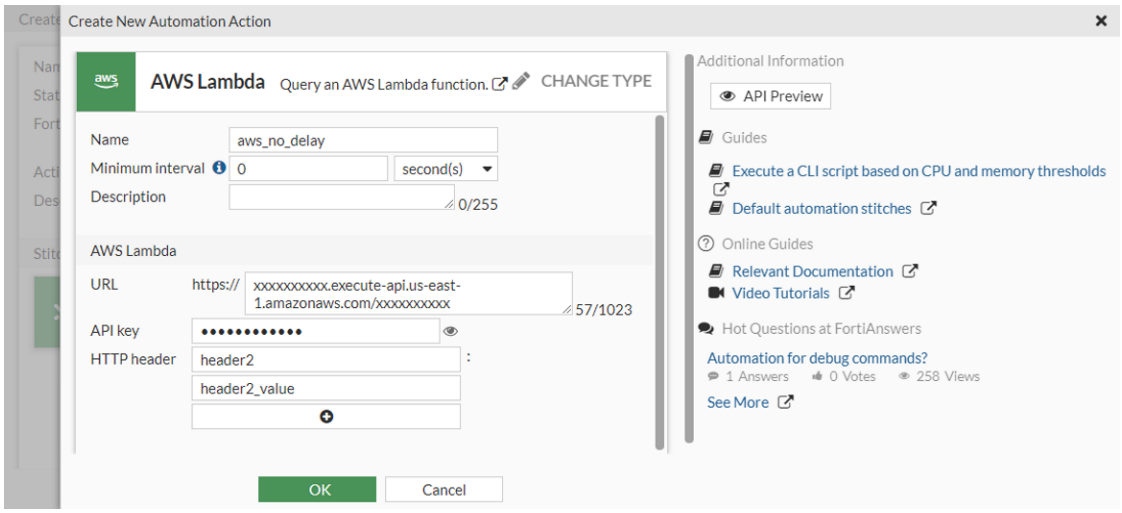
Sample configuration

The following example shows how to configure an automation stitch with an Any Security Rating Notification trigger with AWS Lambda and Email actions. There is a 60-second delay before the Email action.

To configure the automation stitch in the GUI:

1. Go to *Security Fabric* > *Automation* and click *Create New*.
2. Enter the stitch name and description.
3. Configure the trigger:
   a. Click *Add Trigger*.
   b. Select *Any Security Rating Notification*.
   c. Click *Apply*.
4. Configure the AWS Lambda function action:
   a. Click *Add Action*.
   b. Click *Create* and select *AWS Lambda*.
   c. Enter the following:

Name	*aws_no_delay*
URL	Enter the request API URI
API key	Enter the API key
HTTP header	*header2 : header2_value*



d. Click *OK*.

e. Select the action in the list and click *Apply*.

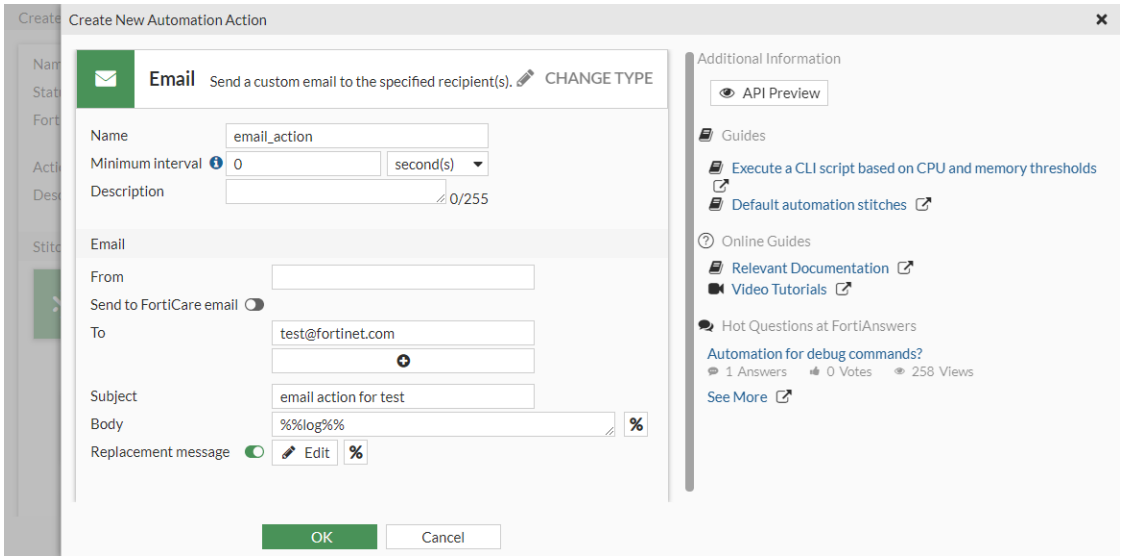
5. Configure the Email notification action:

a. Click *Add Action*.

b. Click *Create* and select *Email*.

c. Enter the following:

Name	*email_action*
To	Enter an email address
Subject	*email action for test*
Replacement message	Enable



d. Click *OK*.

e. Select the action in the list and click *Apply*.

6. Click the *Add delay* located between both actions. Enter *60* and click *OK*.

7. Click *OK*.

To configure the automation stitch in the CLI:

1. Configure the trigger:

```
config system automation-trigger
    edit "Any Security Rating Notification"
        set event-type security-rating-summary
        set report-type any
    next
end
```

2. Configure the actions:

```
config system automation-action
    edit "aws_no_delay"
        set action-type aws-lambda
        set aws-api-key xxxxxxxxxxxx
        set uri "xxxxxxxxxx.execute-api.us-east-1.amazonaws.com/xxxxxxxxxx"
        config http-headers
            edit 1
                set key "header2"
                set value "header2_value"
            next
        end
    next
    edit "email_action"
        set action-type email
        set email-to "test@fortinet.com"
        set email-subject "email action for test"
        set replacement-message enable
    next
end
```

3. Configure the stitch:

```
config system automation-stitch
    edit "aws_no_delay"
        set description "aws action test"
        set trigger "Any Security Rating Notification"
        config actions
            edit 1
                set action "aws_no_delay"
                set required enable
            next
            edit 2
                set action "email_action"
                set delay 60
                set required enable
            next
        end
    next
end
```

Testing automation stitches

In the GUI, go to *Security Fabric* > *Automation*, right-click on the automation stitch and select *Test Automation Stitch*.

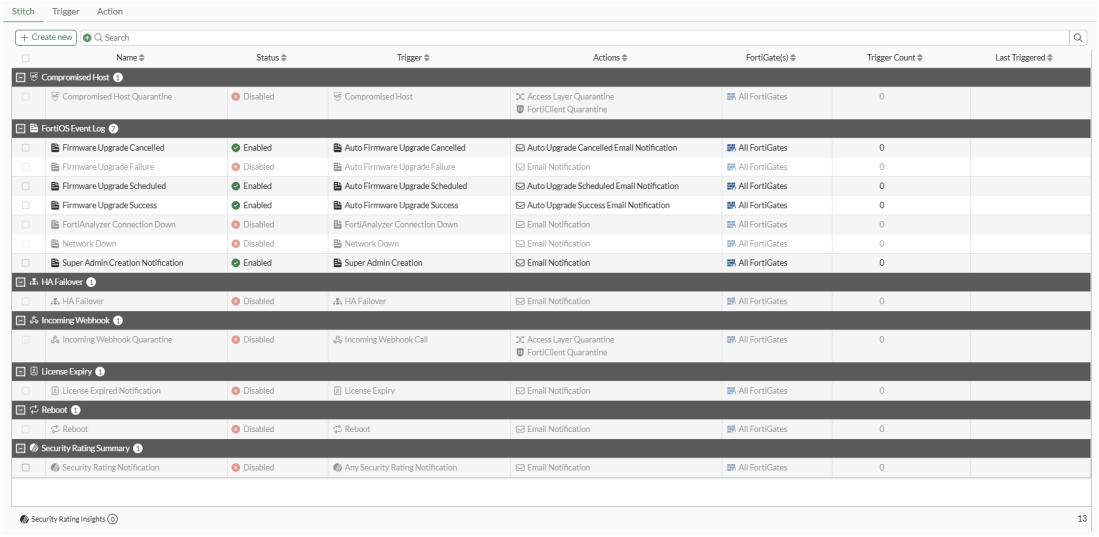
In the CLI, enter `diagnose automation test <automation-stitch name>`.

Default automation stitches

The following default automation stitches are included in FortiOS:

Category	Name
Compromised Host	• Compromised Host Quarantine
FortiOS Event Log	• Firmware Upgrade Cancelled • Firmware Upgrade Failure • Firmware Upgrade Scheduled • Firmware Upgrade Success • FortiAnalyzer Connection Down • Network Down • Super Admin Creation Notification
HA Failover	• HA Failover
Incoming Webhook	• Incoming Webhook Quarantine
License Expiry	• License Expired Notification
Reboot	• Reboot
Security Rating Summary	• Security Rating Notification

To view and edit the automation stitches in the GUI, go to *Security Fabric* > *Automation*.



CLI configurations

Compromised Host Quarantine

```
config system automation-action
    edit "Access Layer Quarantine"
        set description "Quarantine the MAC address on access layer devices (FortiSwitch and
FortiAP)."
        set action-type quarantine
    next
    edit "FortiClient Quarantine"
        set description "Use FortiClient EMS to quarantine the endpoint device."
        set action-type quarantine-forticlient
    next
end
```

```
config system automation-trigger
    edit "Compromised Host"
        set description "An incident of compromise has been detected on a host endpoint."
    next
end
```

```
config system automation-stitch
    edit "Compromised Host Quarantine"
        set description "Quarantine a compromised host on FortiAPs, FortiSwitches, and FortiClient
EMS."
        set status disable
        set trigger "Compromised Host"
        config actions
            edit 1
                set action "Access Layer Quarantine"
            next
            edit 2
                set action "FortiClient Quarantine"
            next
        end
    next
end
```

Firmware Upgrade Cancelled

```
config system automation-action
    edit "Auto Upgrade Cancelled Email Notification"
        set description "Send auto upgrade cancelled email notification to the FortiCare email
address registered on this device."
        set action-type email
        set forticare-email enable
        set email-subject "Firmware Update Cancelled Notification : %%log.devname%%"
        set message "Date: %%log.date%% Time: %%log.time%%
An automated firmware update has been cancelled on device: %%log.devname%%, serial number
%%log.devid%%.
```

```
This is an informational notification and was generated by automation stitch configuration on
%%log.devname%%."
    next
end
```

```
config system automation-trigger
    edit "Auto Firmware Upgrade Cancelled"
        set description "Automatic firmware upgrade cancelled."
        set event-type event-log
        set logid 32325
    next
end
```

```
config system automation-stitch
    edit "Firmware Upgrade Cancelled"
        set description "Automatic firmware upgrade cancelled notification."
        set trigger "Auto Firmware Upgrade Cancelled"
        config actions
            edit 1
                set action "Auto Upgrade Cancelled Email Notification"
            next
        end
    next
end
```

Firmware Upgrade Failure

```
config system automation-action
    edit "Email Notification"
        set description "Send a custom email notification to the FortiCare email address
registered on this device."
        set action-type email
        set forticare-email enable
        set email-subject "%%log.logdesc%%"
    next
end
```

```
config system automation-trigger
    edit "Auto Firmware Upgrade Failure"
        set description "Automatic firmware upgrade failure."
        set event-type event-log
        set logid 22095 22097
    next
end
```

```
config system automation-stitch
    edit "Firmware Upgrade Failure"
        set description "Automatic firmware upgrade failure notification."
        set status disable
        set trigger "Auto Firmware Upgrade Failure"
```

```
        config actions
            edit 1
                set action "Email Notification"
            next
        end
    next
end
```

Firmware Upgrade Scheduled

```
config system automation-action
    edit "Auto Upgrade Scheduled Email Notification"
        set description "Send scheduled auto upgrade email notification to the FortiCare email
address registered on this device."
        set action-type email
        set forticare-email enable
        set email-subject "Firmware Update Scheduled Notification : %%log.devname%%"
        set message "Date: %%log.date%% Time: %%log.time%%
An automated firmware update has been scheduled on device: %%log.devname%%, serial number
%%log.devid%%.
%%log.msg%%

This is an informational notification and was generated by automation stitch configuration on
%%log.devname%%."
    next
end
```

```
config system automation-trigger
    edit "Auto Firmware Upgrade Scheduled"
        set description "Automatic firmware upgrade scheduled."
        set event-type event-log
        set logid 32263
    next
end
```

```
config system automation-stitch
    edit "Firmware Upgrade Scheduled"
        set description "Automatic firmware upgrade scheduled notification."
        set trigger "Auto Firmware Upgrade Scheduled"
        config actions
            edit 1
                set action "Auto Upgrade Scheduled Email Notification"
            next
        end
    next
end
```

Firmware Upgrade Success

```
config system automation-action
    edit "Auto Upgrade Success Email Notification"
        set description "Send auto upgrade success email notification to the FortiCare email
address registered on this device."
        set action-type email
        set forticare-email enable
        set email-subject "Firmware Update Notification : %%log.devname%%"
        set message "An automated firmware update has completed successfully. The device
%%log.devname%%, serial number %%log.devid%%, was updated to firmware version %%log.version%% on
%%log.date%% %%log.time%%.

This is an informational notification and was generated by automation stitch configuration on
%%log.devname%%."
    next
end
```

```
config system automation-trigger
    edit "Auto Firmware Upgrade Success"
        set description "Automatic firmware upgrade complete successfully."
        set event-type event-log
        set logid 22096 22094
    next
end
```

```
config system automation-stitch
    edit "Firmware Upgrade Success"
        set description "Automatic firmware upgrade success notification."
        set trigger "Auto Firmware Upgrade Success"
        config actions
            edit 1
                set action "Auto Upgrade Success Email Notification"
            next
        end
    next
end
```

FortiAnalyzer Connection Down

```
config system automation-action
    edit "Email Notification"
        set description "Send a custom email to the specified recipient(s)."
        set action-type email
        set forticare-email enable
        set email-subject "%%log.logdesc%%"
        set message "%%log%%"
    next
end
```

```
config system automation-trigger
    edit "FortiAnalyzer Connection Down"
        set description "A FortiAnalyzer connection is down."
        set event-type event-log
        set logid 22902
    next
end
```

```
config system automation-stitch
    edit "FortiAnalyzer Connection Down"
        set description "Send an email notification when the connection to FortiAnalyzer is lost."
        set status disable
        set trigger "FortiAnalyzer Connection Down"
        config actions
            edit 1
                set action "Email Notification"
            next
        end
    next
end
```

Network Down

```
config system automation-action
    edit "Email Notification"
        set description "Send a custom email to the specified recipient(s)."
        set action-type email
        set forticare-email enable
        set email-subject "%%log.logdesc%%"
        set message "%%log%%"
    next
end
```

```
config system automation-trigger
    edit "Network Down"
        set description "A network connection is down."
        set event-type event-log
        set logid 20099
        config fields
            edit 1
                set name "status"
                set value "DOWN"
            next
        end
    next
end
```

```
config system automation-stitch
    edit "Network Down"
        set description "Send an email when a network goes down."
        set status disable
```

```
        set trigger "Network Down"
        config actions
            edit 1
                set action "Email Notification"
            next
        end
    next
end
```

HA Failover

```
config system automation-action
    edit "Email Notification"
        set description "Send a custom email to the specified recipient(s)."
        set action-type email
        set forticare-email enable
        set email-subject "%%log.logdesc%%"
        set message "%%log%%"
    next
end
```

```
config system automation-trigger
    edit "HA Failover"
        set description "A HA failover has occurred."
        set event-type ha-failover
    next
end
```

```
config system automation-stitch
    edit "HA Failover"
        set description "Send an email when a HA failover is detected."
        set status disable
        set trigger "HA Failover"
        config actions
            edit 1
                set action "Email Notification"
            next
        end
    next
end
```

Incoming Webhook Quarantine

```
config system automation-action
    edit "Access Layer Quarantine"
        set description "Quarantine the MAC address on access layer devices (FortiSwitch and
FortiAP)."
        set action-type quarantine
    next
    edit "FortiClient Quarantine"
```

```
        set description "Use FortiClient EMS to quarantine the endpoint device."
        set action-type quarantine-forticlient
    next
end
```

```
config system automation-trigger
    edit "Incoming Webhook Call"
        set description "An incoming webhook call is received"
        set event-type incoming-webhook
    next
end
```

```
config system automation-stitch
    edit "Incoming Webhook Quarantine"
        set description "Quarantine a provided MAC address on FortiAPs, FortiSwitches, and
FortiClient EMS using an Incoming Webhook."
        set status disable
        set trigger "Incoming Webhook Call"
        config actions
            edit 1
                set action "Access Layer Quarantine"
            next
            edit 2
                set action "FortiClient Quarantine"
            next
        end
    next
end
```

License Expired Notification

```
config system automation-action
    edit "Email Notification"
        set description "Send a custom email to the specified recipient(s)."
        set action-type email
        set forticare-email enable
        set email-subject "%%log.logdesc%%"
        set message "%%log%%"
    next
end
```

```
config system automation-trigger
    edit "License Expiry"
        set description "A FortiGate license is near expiration."
        set event-type license-near-expiry
        set license-type any
    next
end
```

```
config system automation-stitch
    edit "License Expired Notification"
```

```
        set description "Send an email notification when a license is near expiration."
        set status disable
        set trigger "License Expiry"
        config actions
            edit 1
                set action "Email Notification"
            next
        end
    next
```

Reboot

```
config system automation-action
    edit "Email Notification"
        set description "Send a custom email to the specified recipient(s)."
        set action-type email
        set forticare-email enable
        set email-subject "%%log.logdesc%%"
        set message "%%log%%"
    next
end
```

```
config system automation-trigger
    edit "Reboot"
        set description "A FortiGate is rebooted."
        set event-type reboot
    next
end
```

```
config system automation-stitch
    edit "Reboot"
        set description "Send an email when a FortiGate is rebooted."
        set status disable
        set trigger "Reboot"
        config actions
            edit 1
                set action "Email Notification"
            next
        end
    next
end
```

Security Rating Notification

```
config system automation-action
    edit "Email Notification"
        set description "Send a custom email to the specified recipient(s)."
        set action-type email
        set forticare-email enable
        set email-subject "%%log.logdesc%%"
```

```
        set message "%%log%%"
    next
end
```

```
config system automation-trigger
    edit "Any Security Rating Notification"
        set description "A security rating summary report has been generated."
        set event-type security-rating-summary
        set report-type any
    next
end
```

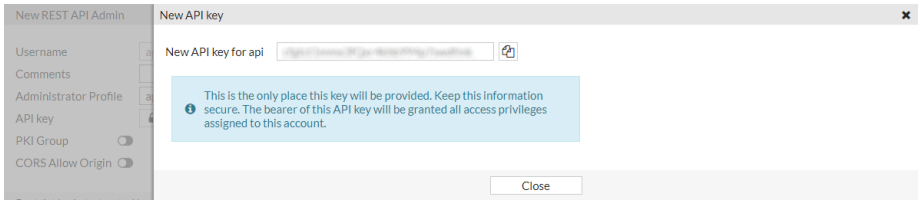
```
config system automation-stitch
    edit "Security Rating Notification"
        set description "Send an email notification when a new Security Rating report is
available."
        set status disable
        set trigger "Any Security Rating Notification"
        config actions
            edit 1
                set action "Email Notification"
            next
        end
    next
end
```

Incoming Webhook Quarantine stitch

The Incoming Webhook Quarantine stitch for API calls to the FortiGate accepts multiple parameters (MAC address and FortiClient UUID) from an Incoming Webhook trigger, which enacts either the Access Layer Quarantine action (MAC address) or the FortiClient Quarantine action (FortiClient UUID). This is a default automation stitch included in FortiOS.

To trigger the Incoming Webhook Quarantine stitch in the GUI:

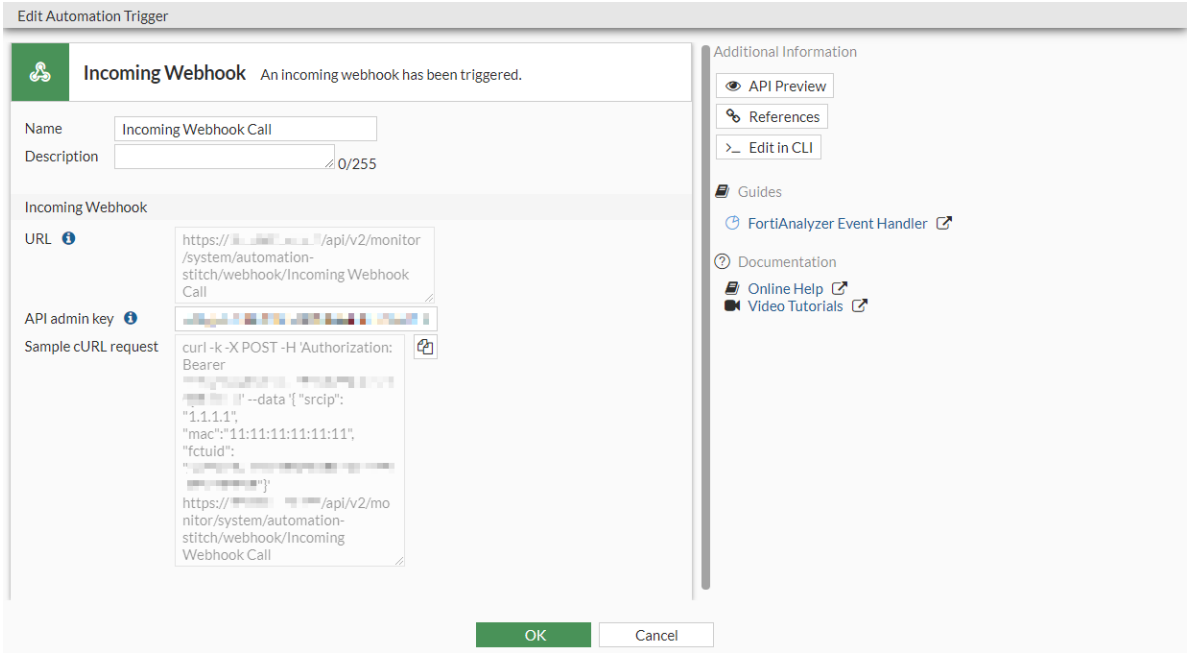
1. Create a new API user:
   a. Go to *System* > *Administrators*.
   b. Click *Create New* > *REST API Admin*.
   c. Configure the *New REST API Admin* settings, and copy the API key to the clipboard.



2. Enable the stitch:

a. Go to *Security Fabric* > *Automation*.
b. Under *Incoming Webhook*, right-click *Incoming Webhook Quarantine*, and select *Select Status* > *Enable*.

3. Get the sample cURL request:
   a. Click the *Trigger* trigger tab.
   b. Under *Incoming Webhook*, right-click *Incoming Webhook Call*, and select *Edit*.
   c. In the *API admin key* field, enter the API key you recorded previously. The *Sample cURL request* field updates.



   d. Copy the *Sample cURL request* to the clipboard.
   e. Click *OK*.

4. Execute the request:
   a. Edit the sample cURL request you just copied.
   b. Add parameters to the `data` field (`"mac"` and `"fctuid"`), and then execute the request.

```
root@pc:~# curl -k -X POST -H 'Authorization: Bearer cfgtct1mmx3fQxr4khb994p7swdfmk' --data '{
"mac":"0c:0a:00:0c:ce:b0", "fctuid": "0000BB0B0ABD0D00B0D0A0B0E0F0B00B"}'
https://172.16.116.226/api/v2/monitor/system/automation-
stitch/webhook/Incoming%20Webhook%20Quarantine
```

```
{
    "http_method":"POST",
    "status":"success",
    "http_status":200,
    "serial":"FGT00E0Q00000000",
    "version":"v6.4.0",
    "build":1545
```

Encode spaces in the automation stitch name with %20. For example, Incoming%20Webhook%20Quarantine

Once the automation stitch is triggered, the MAC address is quarantined by the FortiGate, and an event log is created. The FortiClient UUID is quarantined on the EMS server side.

To trigger the Incoming Webhook Quarantine stitch in the CLI:

1. Create a new API user and note the API key:

```
config system api-user
```

2. Enable the automation stitch:

```
config system api-user
    edit "api"
        set api-key ************
        set accprofile "api_profile"
        set vdom "root"
        config trusthost
            edit 1
                set ipv4-trusthost 10.6.30.0 200.200.200.0
            next
        end
    next
end
```

3. Edit the cURL request to include parameters in the data field ("mac" and "fctuid"), then execute the request:

```
root@pc56:~# curl -k -X POST -H 'Authorization: Bearer cfgtct1mmx0fQxr4khb000p70wdfmk' --data
'{ "mac":"0c:0a:00:0c:ce:b0", "fctuid": "3000BB0B0ABD0D00B0D0A0B0E0F0B00B"}'
https://100.10.100.200/api/v2/monitor/system/automation-
stitch/webhook/Incoming%20Webhook%20Quarantine
```

```
{
    "http_method":"POST",
    "status":"success",
    "http_status":200,
    "serial":"FGT00E0Q00000000",
    "version":"v6.4.0",
    "build":1545
```

Encode spaces in the automation stitch name with %20. For example, Incoming%20Webhook%20Quarantine.

Once the automation stitch is triggered, the MAC address is quarantined by the FortiGate, and an event log is created. The FortiClient UUID is quarantined on the EMS server side.

Sample log

```
date=2020-02-14 time=15:37:48 logid="0100046600" type="event" subtype="system" level="notice"
vd="root" eventtime=1581723468644200712 tz="-0800" logdesc="Automation stitch triggered"
stitch="Incoming Webhook Quarantine" trigger="Incoming Webhook Quarantine"
stitchaction="Compromised Host Quarantine_quarantine,Compromised Host Quarantine_quarantine-
forticlient" from="log" msg="stitch:Incoming Webhook Quarantine is triggered."
```

Triggers

There are two types of automation triggers that can be configured in automation stitches: static and dynamic.

Static automation triggers are included in FortiOS by default. They require only a name, description, and one setting. Static automation triggers can be edited, but they cannot be deleted.

Dynamic automation triggers require multiple settings to be configured. Dynamic automation triggers can be created by clicking the *Create New* button on the *Trigger* tab, or clicking *Create* within the *Create Automation Stitch* page.

The following table outlines the available static triggers.

Trigger	Description
Anomaly Logs	An anomalous event has occurred. See Event log category triggers on page 4521 for an example.
Any Security Rating Notification	A summary is available for a recently run Security Rating report. The default report type is Any. Other available options include: • Security Posture • Fabric Coverage • Optimization
AV & IPS DB Update	The antivirus and IPS database is updating.
Compromised Host	An indicator of compromise (IoC) is detected on a host endpoint. Additional actions are available only for *Compromised Host* triggers: • Access Layer Quarantine • FortiClient Quarantine • VMware NSX Security Tag • IP Ban
Configuration Change	A FortiGate configuration change has occurred.
Conserve Mode	A FortiGate entered conserve mode due to low memory. See Execute a CLI script based on memory and CPU thresholds on page 4571 for an example.
FortiGate Cloud-Based IOC	IOC detection from the FortiGate Cloud IOC service. This option requires an IOC license, a web filter license, and FortiCloud logging must be enabled.
HA Failover	An HA failover is occurring.

Trigger	Description
High CPU	A FortiGate has high CPU usage. See Execute a CLI script based on memory and CPU thresholds on page 4571 for an example.
IPS Logs	An IPS event has occurred.
License Expiry	A FortiGuard license is expiring. The default license type is Any. Other available options include: The license type must be selected. Options include: • FortiCare Support • FortiGuard Web Filter • FortiGuard AntiSpam • FortiGuard AntiVirus • FortiGuard IPS • FortiGuard Management Service • FortiGate Cloud
Local Certificate Expiry	A local certificate is about to expire. See Certificate expiration trigger on page 4525 for an example.
Reboot	A FortiGate is rebooting.
SSH Logs	An SSH event has occurred.
Traffic Violation	A traffic policy has been violated.
Virus Logs	A virus event has occurred.
Web Filter Violation	A web filter policy has been violated.

The following table outlines the available dynamic triggers.

Category	Trigger	Description
Security Fabric		
	Fabric Connector Event	An event has occurred on a specific Fabric connector. See Fabric connector event trigger on page 4513 for details.
	FortiAnalyzer Event Handler	The specified FortiAnalyzer event handler has occurred. See FortiAnalyzer event handler trigger on page 4508 for details.
	FortiGate Cloud Event Handler	The specified FortiGate Cloud event handler has occurred. This option requires a FortiGate Cloud log retention license.
Miscellaneous		
	FortiOS Event Log	The specified FortiOS log has occurred. Multiple event log IDs can be selected, and log field filters can be applied. See FortiOS event log trigger on page 4518 for an example.
	Incoming Webhook	An incoming webhook is triggered.

Category	Trigger	Description
	Schedule	A scheduled monthly, weekly, daily, hourly, or one-time trigger. Set to occur on a specific minute of an specific hour on a specific day. When using the one-time trigger, set to occur on specific date and time in the future. See Schedule trigger on page 4527 for an example.

FortiAnalyzer event handler trigger

You can trigger automation stitches based on FortiAnalyzer event handlers. This allows you to define rules based on complex correlations across devices, log types, frequencies, and other criteria.

To set up a FortiAnalyzer event handler trigger:

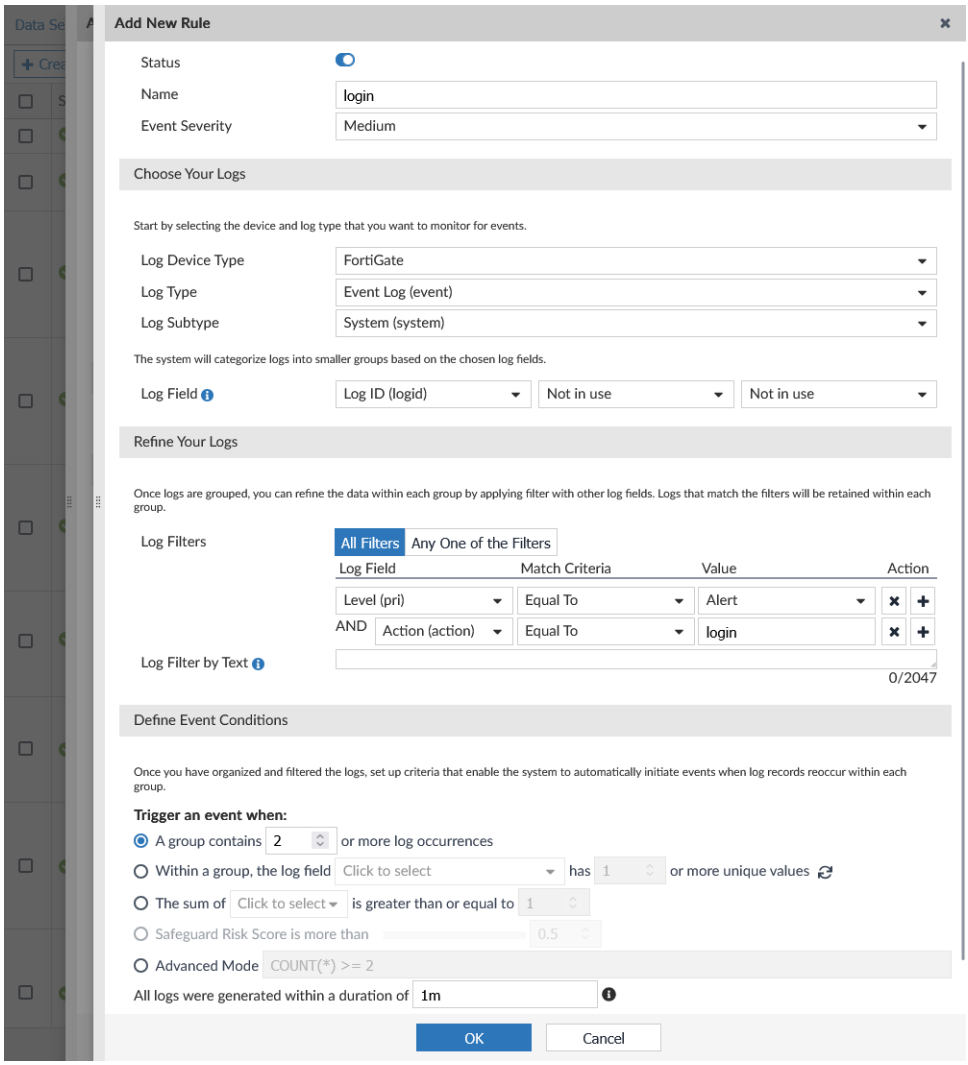
1. Configure a FortiGate event handler on the FortiAnalyzer
2. Configure FortiAnalyzer logging on the FortiGate on page 4510
3. Configure an automation stitch that is triggered by a FortiAnalyzer event handler on page 4510

Configure a FortiGate event handler on the FortiAnalyzer

On the FortiAnalyzer, configure an event handler for the automation stitch. In this example, the event handler is triggered by two or more invalid log in attempts to the FortiGate in less than one minute. See Creating a custom event handler in the FortiAnalyzer Administration Guide for more information.

To configure an event handler on the FortiAnalyzer:

1. Go to *Incidents & Events* > *Event Handlers* and click *Create New*.
2. Set *Name* to *log-handler*.
3. Set *Event Handler Type* to *Basic*.
4. Enable *Automation Stitch*.
5. Click *Add New Rule* and configure a rule with two conditions:



Field	Value
Name	login
Event Severity	Medium
Log Device Type	FortiGate
Log Type	Event Log (event)
Log Subtype	System (system)
Log Field	Log ID (logid)
Log Filters	All Filters
Log Field	Level (pri)
Match Criteria	Equal To
Value	Alert

Field	Value
Log Field	Action (action)
Match Criteria	Equal To
Value	login
Trigger an event when	A group contains 2 or more log occurrences All logs were generated within a duration of 1m
Tags	(Optional) InvalidLogin

6. Configure the other settings as needed then click *OK*.
7. Click *OK*.

Configure FortiAnalyzer logging on the FortiGate

See Configuring FortiAnalyzer on page 4311 for more information.

To configure FortiAnalyzer logging:

1. On the FortiGate, go to *Security Fabric* > *Fabric Connectors*, double-click the *Logging & Analytics* card, and go to the *FortiAnalyzer* tab.
2. Check that the *Status* is *Enabled*.
3. Set *Server* to the IP address of the FortiAnalyzer.
4. Set *Upload option* as needed.
5. Click *OK* and accept the FortiAnalyzer certificate.
6. On the FortiAnalyzer, authorize the FortiGate. See Unauthorized devices in the FortiAnalyzer Administration Guide

Configure an automation stitch that is triggered by a FortiAnalyzer event handler

When a FortiAnalyzer event handler is triggered, it sends a notification to the FortiGate automation framework, which generates a log and triggers the automation stitch.

To configure an automation stitch that is triggered by a FortiAnalyzer event handler in the GUI:

1. Go to *Security Fabric* > *Automation* and click *Create New*.
2. Enter the stitch name, *auto-faz-1*.
3. Configure the trigger:
   a. Click *Add Trigger*.
   b. Click *Create* and select *FortiAnalyzer Event Handler*.
   c. Enter the following:

Name	auto-faz-1
Event handler name	log-handler
Event severity	Medium

Event tag	Optional. Enter only if you have specified a custom tag for the event handler. In this case, set to *InvalidLogin*.

   d. Click *OK*.
   e. Select the trigger in the list and click *Apply*.
4. Configure the Email notification action:
   a. Click *Add Action*.
   b. Click *Create* and select *Email*.
   c. Enter the following:

Name	auto-faz-1_email
To	Enter an email address
Subject	CSF stitch alert
Body	User login FortiGate failed attempts %%log%%

   d. Click *OK*.
   e. Select the action in the list and click *Apply*.
5. Click *OK*.

To configure an automation stitch that is triggered by a FortiAnalyzer event handler in the CLI:

1. Create an automation trigger:

```
config system automation-trigger
    edit "auto-faz-1"
        set event-type faz-event
        set faz-event-name "log-handler"
        set faz-event-severity "medium"
        set faz-event-tags "InvalidLogin"
    next
end
```

2. Create an automation action:

```
config system automation-action
    edit "auto-faz-1_email"
        set action-type email
        set email-to "tsmith@ztnademo.com"
        set email-subject "CSF stitch alert"
        set message "User login FortiGate failed attempts %%log%%"
    next
end
```

3. Create the automation stitch:

```
config system automation-stitch
    edit "auto-faz-1"
        set trigger "auto-faz-1"
```

```
        config actions
            edit 1
                set action "auto-faz-1_email"
                set required enable
            next
        end
    next
end
```

View the trigger event log

To view the trigger event log:

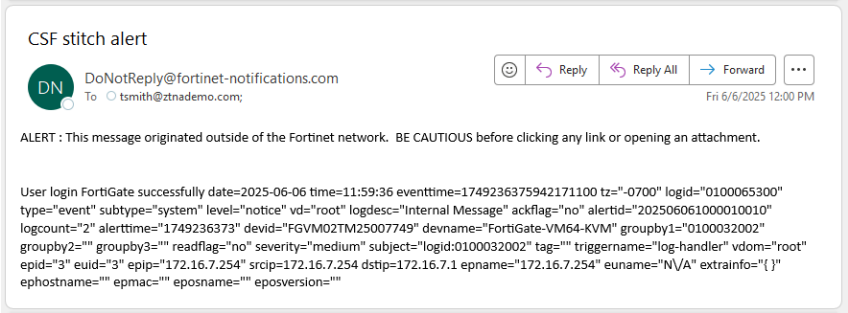
1. Attempt to log in to the FortiGate with the wrong password twice in a row.
2. Check that two system logs with severity *Alert* are generated on the FortiGate:

```
# execute log filter field subtype system
# execute log filter category event
# execute log display
    ...
11: date=2025-06-06 time=11:59:14 eventtime=1749236354989858133 tz="-0700" logid="0100032002"
type="event" subtype="system" level="alert" vd="root" logdesc="Admin login failed" sn="0"
user="admin" ui="https(172.16.7.254)" method="https" srcip=172.16.7.254 dstip=172.16.7.1
action="login" status="failed" reason="passwd_invalid" msg="Administrator admin login failed
from https(172.16.7.254) because of invalid password"
12: date=2025-06-06 time=11:59:12 eventtime=1749236352450280347 tz="-0700" logid="0100032002"
type="event" subtype="system" level="alert" vd="root" logdesc="Admin login failed" sn="0"
user="admin" ui="https(172.16.7.254)" method="https" srcip=172.16.7.254 dstip=172.16.7.1
action="login" status="failed" reason="passwd_invalid" msg="Administrator admin login failed
from https(172.16.7.254) because of invalid password"
```

3. This triggers an event on the FortiAnalyzer that sends a notification to the FortiGate automation framework:

```
# diagnose debug application autod -1
# diagnose debug enable
__handle_req_body()-109: subscriber update: name subscr-node msg_type 0 msg_len 309:
{"format":0,"category_filter":2,"free_style":[{"category":"event","filter":"(logid 0100022040
0100065304 0100022041 0100065305 0100022042 0100065306 0100022043 0100022044 0100022045
0100022901 0100022902 0100022903 0100022808 0100022809 0100053400 0100053401 0100026004
0100065311)","filter_type":"include"}]}

pid:2086-__handle_msg()-414: Subscriber:4 received package. pubid:0 pkgid:97 pkg_index:0
pid:2086-__handle_pkg_logs()-356: Subscriber:4 processing package size:1512 logs:1 pickup:1
pid:2086-__subscr_close_cur_pkg()-140: close package size:1512 logs:1
__action_email_hdl()-190: email action (auto-faz-1_email) is called.
from:
to:tsmith@ztnademo.com;
subject:CSF stitch alert
```

An alert email is sent out and an automation stitch log is generated. The email sent by the action will look similar to the following:



4. To view the log in the GUI, go to *Log & Report* > *System Events*, select *General System Events*, and from the log location dropdown select *FortiAnalyzer*.
5. To view the trigger event log in the CLI:

```
# execute log display
    ...
9: date=2025-06-06 time=11:59:37 eventtime=1749236376747950301 tz="-0700" logid="0100046600"
type="event" subtype="system" level="notice" vd="root" logdesc="Automation stitch triggered"
stitch="auto-faz-1" trigger="auto-faz-1" stitchaction="auto-faz-1_email" from="log"
msg="stitch:auto-faz-1 is triggered.
```

Fabric connector event trigger

With the *Fabric Connector Event* trigger, any supported Fabric connector is able to trigger an automation stitch on the FortiGate based on a specific event defined on the Fabric connector. Currently, only FortiDeceptor 4.1 and later supports this trigger for the *Insider Threat*, *Notify Ban*, and *Notify Unban* events.

In the following example, an authorized FortiDeceptor in the Security Fabric deploys a decoy called ubuntu16 configured with SSH, SAMBA, HTTP, and HTTPS services.

This example assumes the Security Fabric is already configured. Refer to Configuring the root FortiGate and downstream FortiGates and FortiDeceptor for detailed configuration steps. On the root FortiGate, the *Allow downstream device REST API access* option must be enabled (`set downstream-access enable`). The minimum permission required for the selected *Administrator profile* is *Read/Write* for *User & Device* (`set authgrp read-write`).

Three stitches are configured, one for each FortiDeceptor trigger type:

Stitch name	Fabric connector event trigger	Actions
fortideceptor_threat	Insider threat	Email and IP ban
fortideceptor_ban	Notify ban	Email and IP ban
fortideceptor_unban	Notify unban	Email and CLI script

To configure stitches with the Fabric connector event trigger in the GUI:

1. Configure the triggers:
   a. Go to *Security Fabric* > *Automation*, select the *Trigger* tab, and click *Create New*.
   b. In the *Security Fabric* section, click *Fabric Connector Event* and enter the following:

Name	*fdc_Insider_Threat*
Description	*Insider_Threat*
Connector	Select the FortiDeceptor connector
Event Name	*Insider Threat*

c. Click *OK*.

d. Repeat these steps to create two more triggers with the following settings:

Name	*fdc_Notify_Ban*
Description	*Notify_Ban*
Connector	Select the FortiDeceptor connector
Event Name	*Notify Ban*

Name	*fdc_Notify_Unban*
Description	*Notify_Unban*
Connector	Select the FortiDeceptor connector
Event Name	*Notify Unban*

2. Configure the actions (the default *IP Ban* action will be added later to the stitch):
   a. Go to *Security Fabric > Automation*, select the *Action* tab, and click *Create New*.
   b. In the *Notifications* section, click *Email* and enter the following:

Name	*email_log*
To	Enter an email address
Subject	*CSF stitch alert*

   c. Click *OK*.
   d. Repeat these steps to create a *CLI Script* (in the *General* section) action with the following settings:

Name	*fdc_unban*
Script	*diagnose user banned-ip delete src4 %%log.srcip%%*
Administrator profile	*super_admin*

3. Configure the *fortideceptor_threat* stitch:
   a. Go to *Security Fabric > Automation*, select the *Stitch* tab, and click *Create New*.
   b. Enter the name, *fortideceptor_threat*.
   c. Click *Add Trigger*. Select *fdc_Insider_Threat* and click *Apply*.
   d. Click *Add Action*. Select *email_log* and click *Apply*.
   e. Click *Add Action*. Select *IP Ban* and click *Apply*.
   f. Click the *Add delay* located between both actions. Enter *5* and click *OK*.
   g. Click *OK*.
4. Configure the *fortideceptor_ban* stitch:

   a. Go to *Security Fabric > Automation*, select the *Stitch* tab, and click *Create New*.
   b. Enter the name, *fortideceptor_ban*.
   c. Click *Add Trigger*. Select *fdc_Notify_Ban* and click *Apply*.
   d. Click *Add Action*. Select *email_log* and click *Apply*.
   e. Click *Add Action*. Select *IP Ban* and click *Apply*.
   f. Click the *Add delay* located between both actions. Enter *5* and click *OK*.
   g. Click *OK*.
5. Configure the *fortideceptor_unban* stitch:
   a. Go to *Security Fabric > Automation*, select the *Stitch* tab, and click *Create New*.
   b. Enter the name, *fortideceptor_unban*.
   c. Click *Add Trigger*. Select *fdc_Notify_Unban* and click *Apply*.
   d. Click *Add Action*. Select *email_log* and click *Apply*.
   e. Click *Add Action*. Select *fdc_unban* and click *Apply*.
   f. Click the *Add delay* located between both actions. Enter *5* and click *OK*.
   g. Click *OK*.

To configure stitches with the Fabric connector event trigger in the CLI:

1. Configure the triggers:

```
config system automation-trigger
    edit "fdc_Insider_Threat"
        set description "Insider_Threat"
        set event-type fabric-event
        set serial "FDC-VMTM210000**"
        set fabric-event-name "insider_threat"
    next
    edit "fdc_Notify_Ban"
        set description "Notify_Ban"
        set event-type fabric-event
        set serial "FDC-VMTM210000**"
        set fabric-event-name "notify_ban"
    next
    edit "fdc_Notify_Unban"
        set description "Notify_Unban"
        set event-type fabric-event
        set serial "FDC-VMTM210000**"
        set fabric-event-name "notify_unban"
    next
end
```

2. Configure the actions:

```
config system automation-action
    edit "IP Ban"
        set action-type ban-ip
    next
    edit "fdc_unban"
        set action-type cli-script
```

```
        set script "diagnose user banned-ip delete src4 %%log.srcip%%"
        set output-size 10
        set timeout 0
        set accprofile "super_admin"
    next
    edit "email_log"
        set action-type email
        set email-to "*******@fortinet.com"
        set email-subject "CSF stitch alert"
    next
end
```

3. Configure the stitches:

```
config system automation-stitch
    edit "fortideceptor_threat"
        set trigger "fdc_Insider_Threat"
        config actions
            edit 1
                set action "email_log"
                set required enable
            next
            edit 2
                set action "IP Ban"
                set delay 5
                set required enable
            next
        end
    next
    edit "fortideceptor_ban"
        set trigger "fdc_Notify_Ban"
        config actions
            edit 1
                set action "email_log"
                set required enable
            next
            edit 2
                set action "IP Ban"
                set delay 5
                set required enable
            next
        end
    next
    edit "fortideceptor_unban"
        set trigger "fdc_Notify_Unban"
        config actions
            edit 1
                set action "email_log"
                set required enable
            next
            edit 2
                set action "fdc_unban"
```

```
                set delay 5
                set required enable
            next
        end
    next
end
```

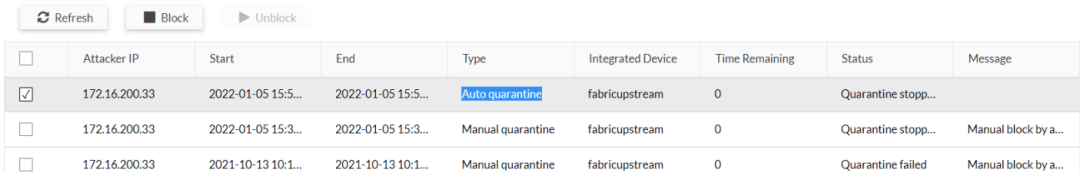
Verification

A device with IP 172.16.200.33 uses SSH to access the decoy (ubuntu16) deployed in the FortiDeceptor. The FortiDeceptor will detect the attacker IP 172.16.200.33, automatically quarantine it, and send the insider threat notification to the FortiGate. This notification will trigger the *fortideceptor_threat* stitch due to the insider threat event trigger, so an email alert is sent and the attacker IP (172.16.200.33) is banned.

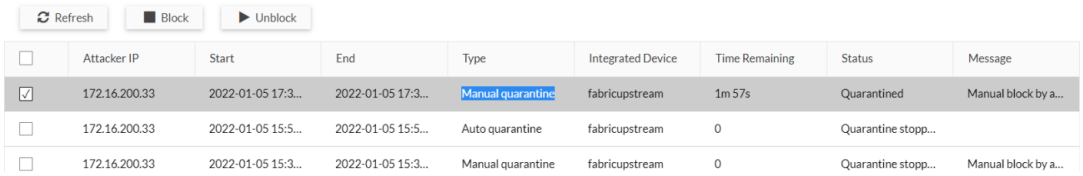
In FortiDeceptor, if the attacker IP (172.16.200.33) is manually blocked or unblocked, the FortiDeceptor will send out the internal block or unblock notification to FortiGate (see Quarantine Status for more details). This notification will trigger the *fortideceptor_ban* or *fortideceptor_unban* stitch due the notify ban or unban event trigger. An email alert is sent, and based on the event, the IP is banned or the CLI script runs to unban the IP.

To view the quarantine details in FortiDeceptor:

1. Go to *Fabric* > *Quarantine Status*.
   a. Automatic quarantine:

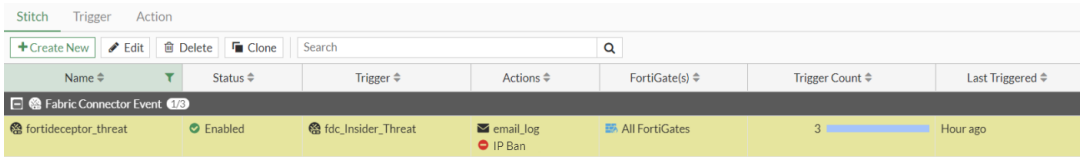


   b. Manual block or unblock:

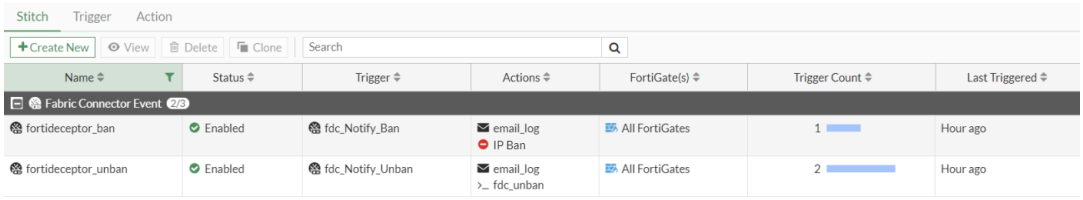


To confirm that the stitch was triggered in the FortiOS GUI:

1. Go to *Security Fabric* > *Automation* and select the *Stitch* tab.
   a. Triggered insider threat:



b. Triggered notify ban or unban:



To view the quarantined IP details in the FortiOS CLI:

```
# diagnose user banned-ip list
src-ip-addr       created                  expires                  cause
172.16.200.33     Wed Jan  5 15:57:41 2022 indefinite               Administrative
```

If the IP is unbanned by the stitch, the list will be empty:

```
# diagnose user banned-ip list
src-ip-addr       created                  expires                  cause
```

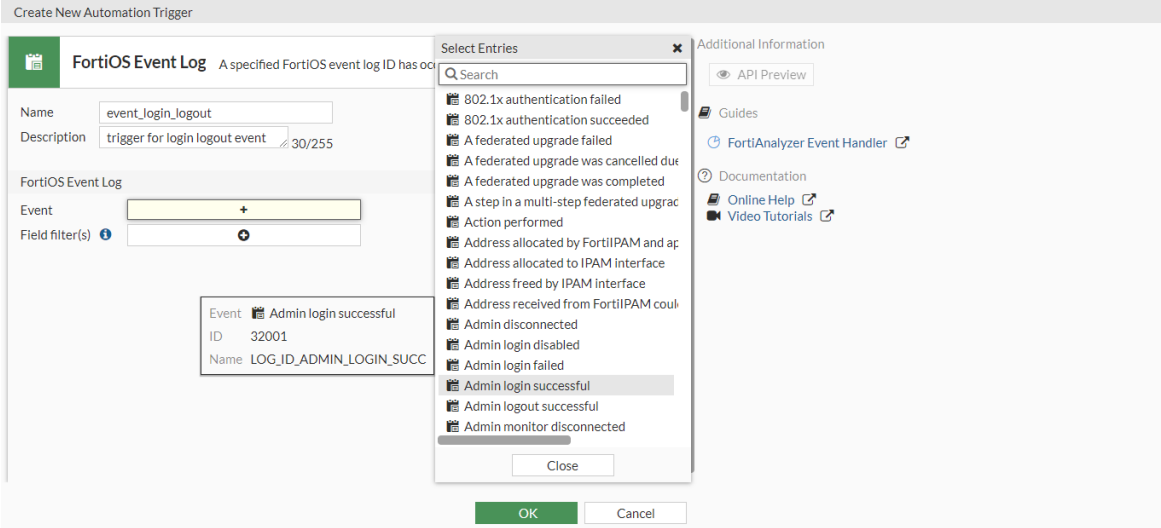
FortiOS event log trigger

You can configure a FortiOS event log trigger for when a specific event log ID occurs. You can select multiple event log IDs, and apply log field filters. FortiOS event log triggers can be configured from the *Security Fabric > Automation > Trigger* page, or by using the shortcut on the *Log & Report > System Events > Logs* page.

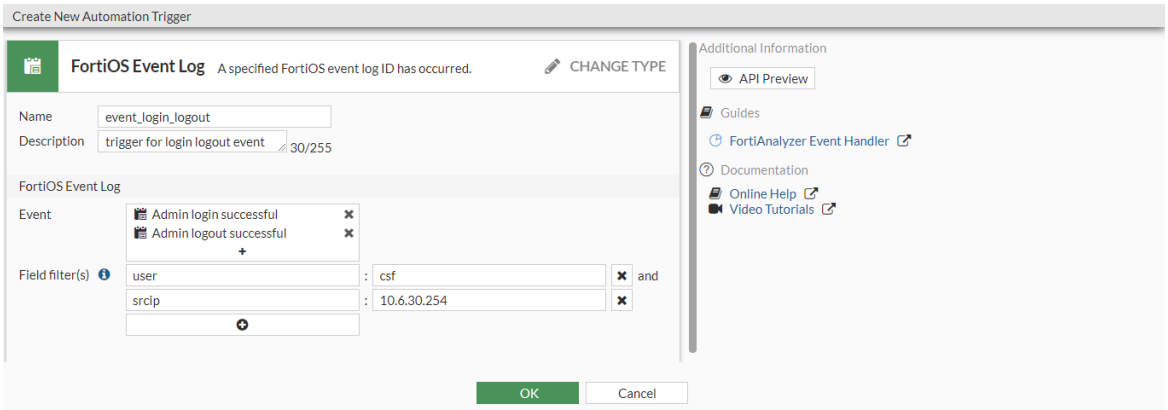
A maximum of 16 log IDs can be set as triggers for the event log.

To configure a FortiOS event log trigger in the GUI:

1. Go to *Security Fabric > Automation*, select the *Trigger* tab, and click *Create New*.
2. In the *Miscellaneous* section, click *FortiOS Event Log*.
3. Enter a name and description.
4. In the *Event* field, click the *+* to select multiple event log IDs.
   The *Event* options correspond to the *Message Meaning* listed in the FortiOS Log Message Reference. Hover over an entry to view the tooltip that includes the event ID and log name. In this example, the *Admin login successful* event in the GUI corresponds to log ID *32001*, which is *LOG_ID_ADMIN_LOGIN_SUCC*.



5. In the *Field filter(s)* field, click the + to add multiple field filters. The configured filters must match in order for the stitch to be triggered.
   a. To view the list of available fields for a log, refer to the FortiOS Log Message Reference by appending the log ID to the document URL (https://docs.fortinet.com/document/fortigate/8.0.0/fortios-log-message-reference/<log_ID>).



6. Click *OK*.

To configure a FortiOS event log trigger in the CLI:

```
config system automation-trigger
    edit "event_login_logout"
        set description "trigger for login logout event"
        set event-type event-log
        set logid 32001 32003
        config fields
            edit 1
                set name "user"
                set value "csf"
            next
            edit 2
                set name "srcip"
```

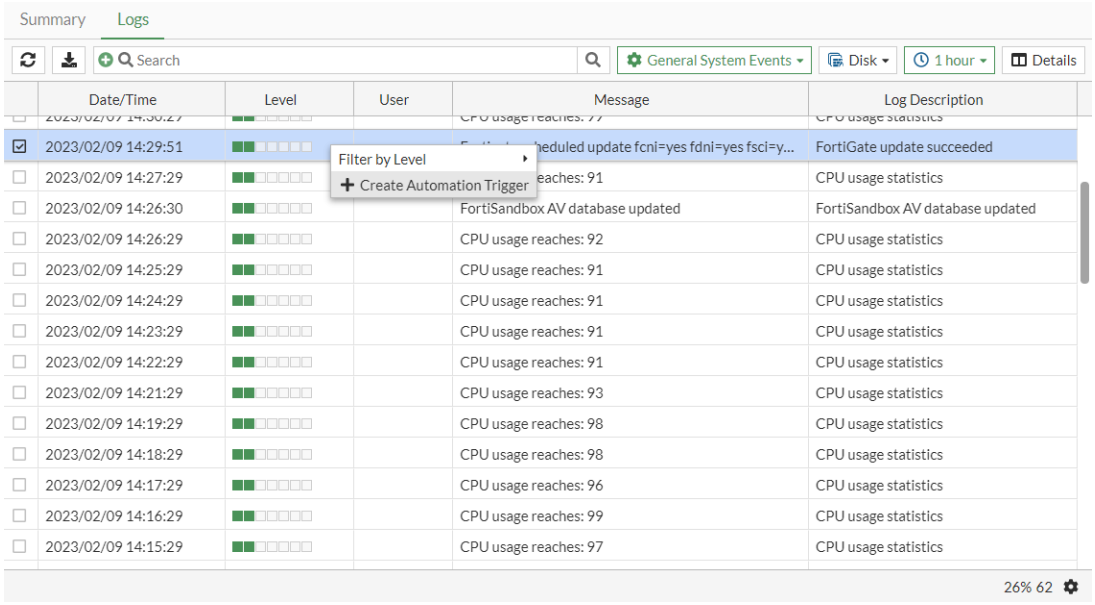
```
                set value "10.6.30.254"
            next
        end
    next
end
```

System Events page shortcut

A FortiOS Event Log trigger can be created using the shortcut on the *System Events > Logs* page. In this example, a trigger is created for a FortiGate update succeeded event log.

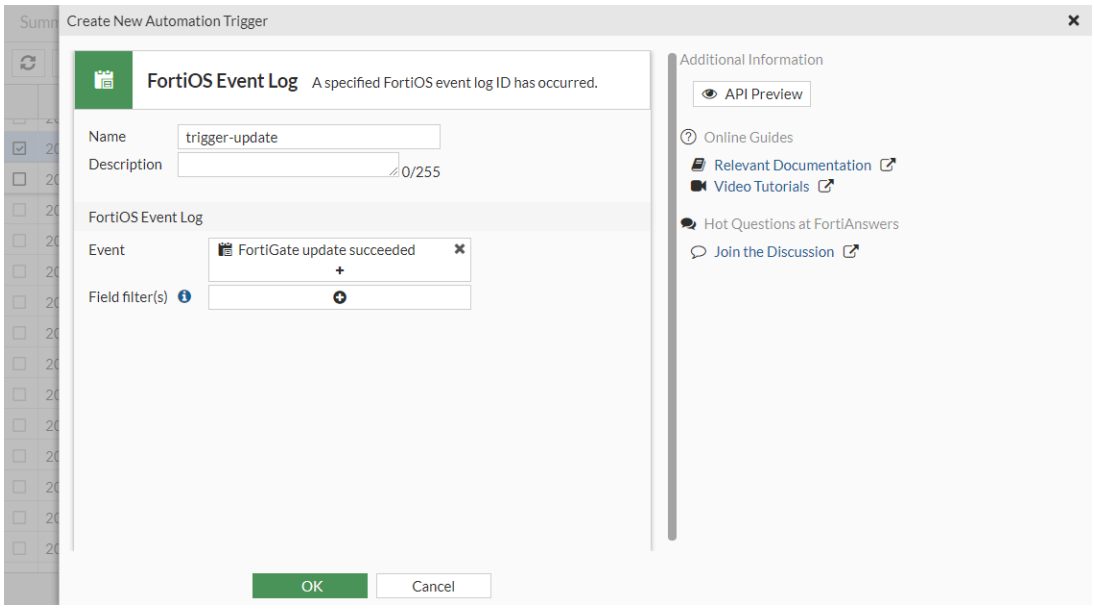
To configure a FortiOS Event Log trigger from the System Events page:

1. Go to *Log & Report > System Events* and select the *Logs* tab.
2. Select a log for a successful FortiGate update, then right-click and select *Create Automation Trigger*.

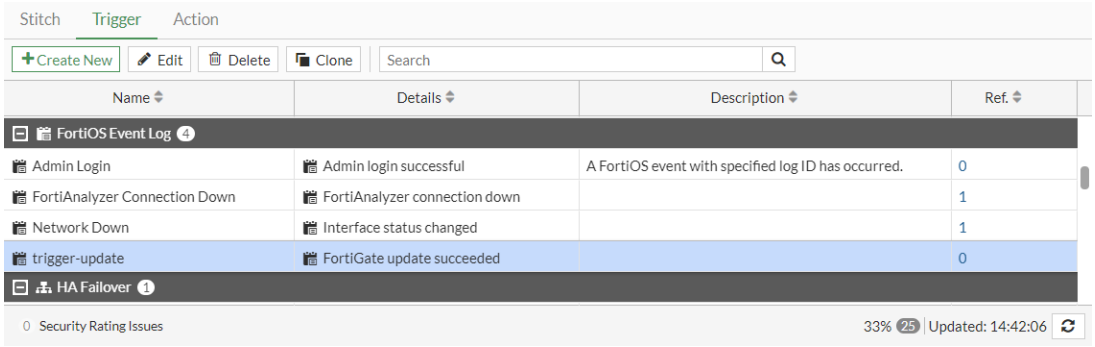


The *Create New Automation Trigger* pane opens to configure the FortiOS Event Log settings.

3. Enter a name (such as *trigger-update*). The *Event* field is already populated with *FortiGate update succeeded*.



4. Optionally in the *Field filter(s)* field, click the + to add multiple field filters. The configured filters must match in order for the stitch to be triggered.
5. Click *OK*. The trigger is now listed on the *Security Fabric > Automation > Trigger* page.



Event log category triggers

There are six default automation triggers based on event log categories:

- Anomaly logs
- IPS logs
- SSH logs
- Traffic violations
- Virus logs
- Web filter violations

When multi-VDOM mode is enabled, individual VDOMs can be specified so that the trigger is only applied to those VDOMs.

```
config system automation-trigger
    edit "Anomaly Logs"
```

```
        set trigger-type event-based
        set event-type anomaly-logs
        set vdom <name>
    next
    edit "IPS Logs"
        set trigger-type event-based
        set event-type ips-logs
        set vdom <name>
    next
    edit "SSH Logs"
        set trigger-type event-based
        set event-type ssh-logs
        set vdom <name>
    next
    edit "Traffic Violation"
        set trigger-type event-based
        set event-type traffic-violation
        set vdom <name>
    next
    edit "Virus Logs"
        set trigger-type event-based
        set event-type virus-logs
        set vdom <name>
    next
    edit "Webfilter Violation"
        set trigger-type event-based
        set event-type webfilter-violation
        set vdom <name>
    next
end
```

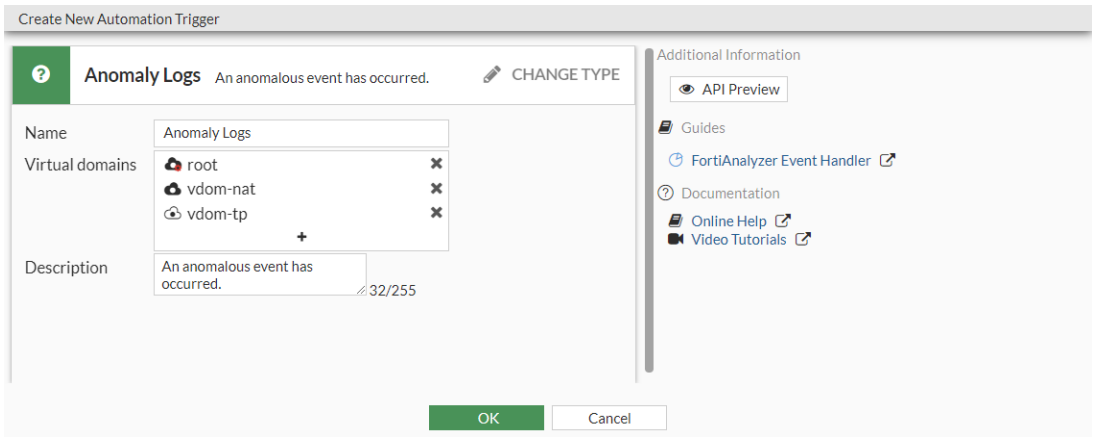
A maximum of 16 log IDs can be set as triggers for the event log.

Example

In this example, an automation stitch is created that uses the anomaly logs trigger and an email notification action. The trigger specifies which VDOMs should be used. There is a three-second delay between the trigger and action.

To configure an automation stitch with the anomaly logs trigger in the GUI:

1. Configure the trigger:
   a. Go to *Security Fabric > Automation*, select the *Trigger* tab.
   b. Edit the *Anomaly Logs* trigger.
   c. Add the required VDOMs (*root*, *vdom-nat*, *vdom-tp*).



d. Click *OK*.

2. Configure the action:
   a. Go to *Security Fabric > Automation*, select the *Action* tab, and click *Create New*.
   b. In the *Notifications* section, click *Email* and enter the following:

Name	*email_default_rep_message*
To	Enter an email address
Subject	*CSF stitch alert*
Replacement message	Enable

   c. Click *OK*.
3. Configure the stitch:
   a. Go to *Security Fabric > Automation*, select the *Stitch* tab, and click *Create New*.
   b. Enter the name, *anomaly-logs-stitch*.
   c. Click *Add Trigger*. Select *Anomaly Logs* and click *Apply*.
   d. Click *Add Action*. Select *email_default_rep_message* and click *Apply*.
   e. Click *Add delay* (between the trigger and action). Enter *3* and click *OK*.
   f. Click *OK*.

To configure an automation stitch with the anomaly logs trigger in the CLI:

1. Configure the trigger:

```
config system automation-trigger
    edit "Anomaly Logs"
        set event-type anomaly-logs
        set vdom "root" "vdom-nat" "vdom-tp"
    next
end
```

2. Configure the action:

```
config system automation-action
    edit "email_default_rep_message"
```

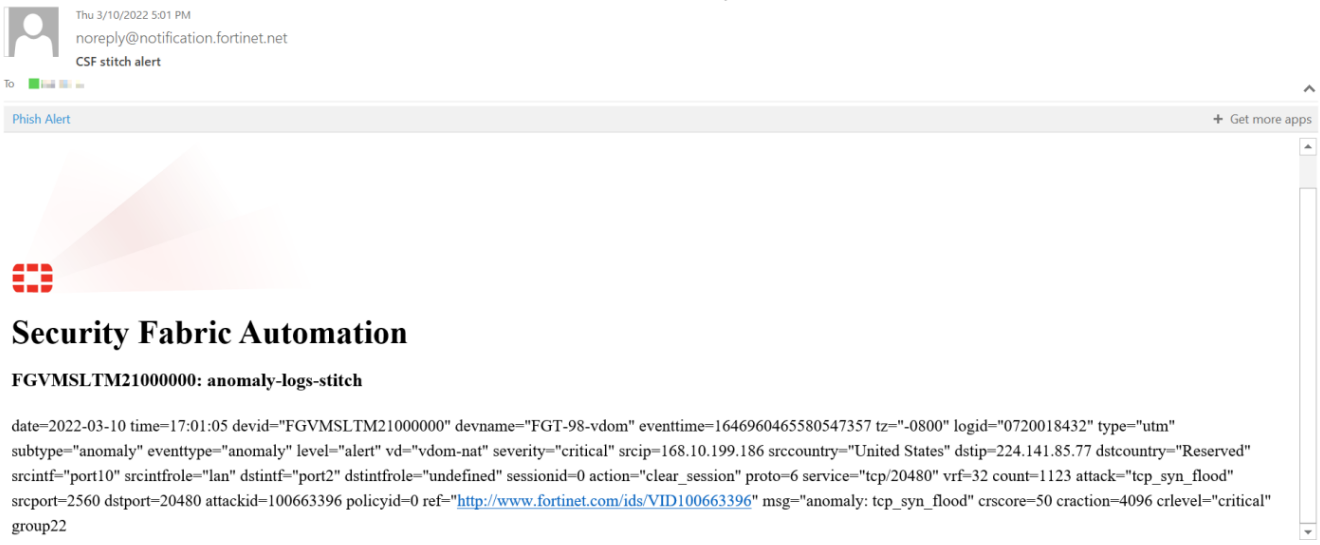
```
        set action-type email
        set email-to "admin@fortinet.com"
        set email-subject "CSF stitch alert"
        set replacement-message enable
    next
end
```

3. Configure the stitch:

```
config system automation-stitch
    edit "anomaly-logs-stitch"
        set description "anomaly-logs"
        set trigger "Anomaly Logs"
        config actions
            edit 1
                set action "email_default_rep_message"
                set delay 3
                set required enable
            next
        end
    next
end
```

Verification

Once the anomaly log is generated, the automation stitch is triggered end the email notification is sent.



To confirm that the stitch was triggered in the GUI:

1. Go to *Security Fabric* > *Automation* and select the *Stitch* tab.
2. Verify the *Last Triggered* column.

To confirm that the stitch was triggered in the CLI:

```
# diagnose test application autod 2
...
stitch: anomaly-logs-stitch
        destinations: all
        trigger: Anomaly Logs
                type:anomaly logs

                field ids:
                        (id:6)vd=root,vdom-nat,vdom-tp

        local hit: 1 relayed to: 0 relayed from: 0
        actions:
                email_default_rep_message type:email interval:0
                        delay:3 required:yes
                        subject: CSF stitch alert
                        body: %%log%%
                        sender:
                        mailto:admin@fortinet.com;
```

Certificate expiration trigger

The local certificate expiry trigger (`local-certificate-near-expiry`) can be used in an automation stitch if a user-supplied local certificate used for Agentless VPN, deep inspection, or other purpose is about to expire. This trigger relies on a VPN certificate setting in the CLI configuration setting for the certificate log expiring warning threshold:

```
config vpn certificate setting
    set cert-expire-warning <integer>
end
```

`cert-expire-warning <integer>`	Set the certificate log expiring warning threshold, in days (0 - 100, default = 14).

Example

In this example, the default local certificate expiry trigger is used with an email notification action to remind an administrator to re-sign or load a new local certificate to avoid any service interruptions. The local certificate, fw-cert-30-days, will expire in 30 days. The certificate log expiring warning threshold is set to 31 days.

To configure the certificate log expiring warning threshold:

```
config vpn certificate setting
    set cert-expire-warning 31
end
```

To configure an automation stitch with the local certificate expiry trigger in the GUI:

1. Configure the action:
   a. Go to *Security Fabric > Automation*, select the *Action* tab, and click *Create New*.
   b. In the *Notifications* section, click *Email*, and enter the following:

Name	*email_no_rep_message*
To	Enter an email address.
Subject	*CSF stitch alert*

   c. Click *OK*.
2. Configure the stitch:
   a. Go to *Security Fabric > Automation*, select the *Stitch* tab, and click *Create New*.
   b. Enter the name, *cert-expiry*.
   c. Click *Add Trigger*. Select *Local Certificate Expiry* and click *Apply*.
   d. Click *Add Action*. Select *email_no_rep_message* and click *Apply*.
   e. Click *OK*.

To configure an automation stitch with the local certificate expiry trigger in the CLI:

1. Configure the trigger:

```
config system automation-trigger
    edit "Local Certificate Expiry"
        set description "A local certificate is near expiration."
        set event-type local-cert-near-expiry
    next
end
```

2. Configure the action:

```
config system automation-action
    edit "email_no_rep_message"
        set action-type email
        set email-to "*******@fortinet.com"
        set email-subject "CSF stitch alert"
    next
end
```

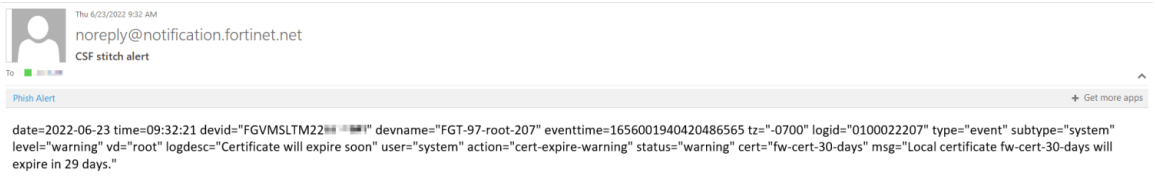
3. Configure the stitch:

```
config system automation-stitch
    edit "cert-expiry"
        set trigger "Local Certificate Expiry"
        config actions
            edit 1
                set action "email_no_rep_message"
                set required enable
            next
        end
```

```
    next
end
```

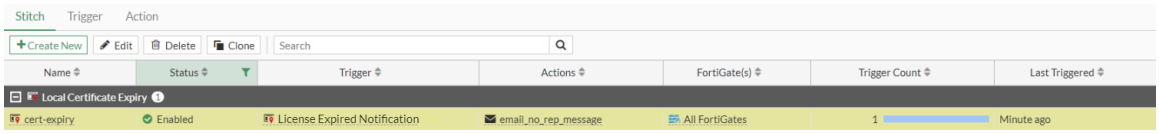
Verification

Once the event log is generated for the local certificate expiry, the automation stitch is triggered end the email notification is sent.



To confirm that the stitch was triggered in the GUI:

1. Go to *Security Fabric* > *Automation* and select the *Stitch* tab.
2. Verify the *Last Triggered* column.



To confirm that the stitch was triggered in the CLI:

```
# diagnose test application autod 3
alert mail log count: 0

stitch: cert-expiry

        local hit: 1 relayed to: 0 relayed from: 0
        last trigger:Thu Jun 23 09:32:21 2022
        last relay:
        actions:
                email_no_rep_message:
                        done: 1 relayed to: 0 relayed from: 0
                        last trigger:Thu Jun 23 09:32:21 2022
                        last relay:

logid to stitch mapping:
id:22207  local hit: 1 relayed hits: 0
        cert-expiry
```

Schedule trigger

The schedule automation trigger can be used for monthly, weekly, daily, hourly, or one-time triggers, including a one-time configuration backup to a disk, a reboot, or a shutdown.

```
config system automation-trigger
    edit <name>
        set trigger-type scheduled
        set trigger-frequency {hourly | daily | weekly | monthly | once}
        set trigger-hour <integer>
        set trigger-minute <integer>
        set trigger-weekday {sunday | monday | tuesday | wednesday | thursday | friday | saturday}
        set trigger-day <integer>
        set trigger-datetime <YYYY-MM-DD HH:MM:SS>
    next
end
```

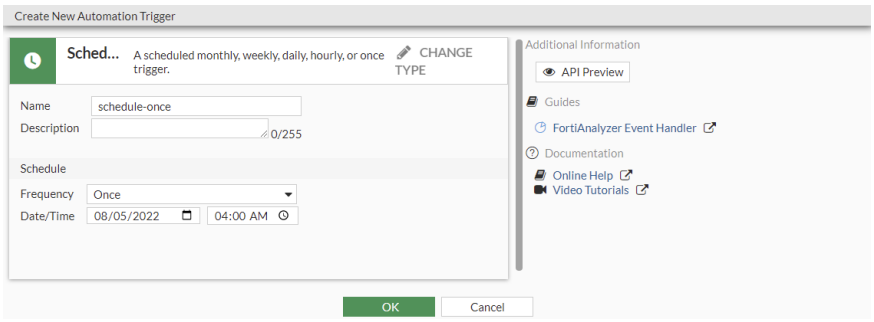
`trigger-frequency {hourly | daily | weekly | monthly | once}`	Set the scheduled trigger frequency (default = daily).
`trigger-hour <integer>`	Set the hour of the day on which to trigger (0 - 23, default = 1), available for daily, weekly, and monthly schedule.
`trigger-minute <integer>`	Set the minute of the hour on which to trigger (0 - 59, default = 0), available for hourly, daily, weekly, and monthly schedule.
`trigger-weekday {sunday | monday | tuesday | wednesday | thursday | friday | saturday}`	Set the day of the week to trigger, available for weekly schedule.
`trigger-day <integer>`	Set the day within a month to trigger, available for monthly schedule.
`trigger-datetime <YYYY-MM-DD HH:MM:SS>`	Set the trigger time in YYYY-MM-DD HH:MM:SS format for one-time triggers.

Example

In this example, an automation stitch is created to trigger a one-time configuration backup to a disk. The backup will occur August 5, 2022 at 4:00 AM.

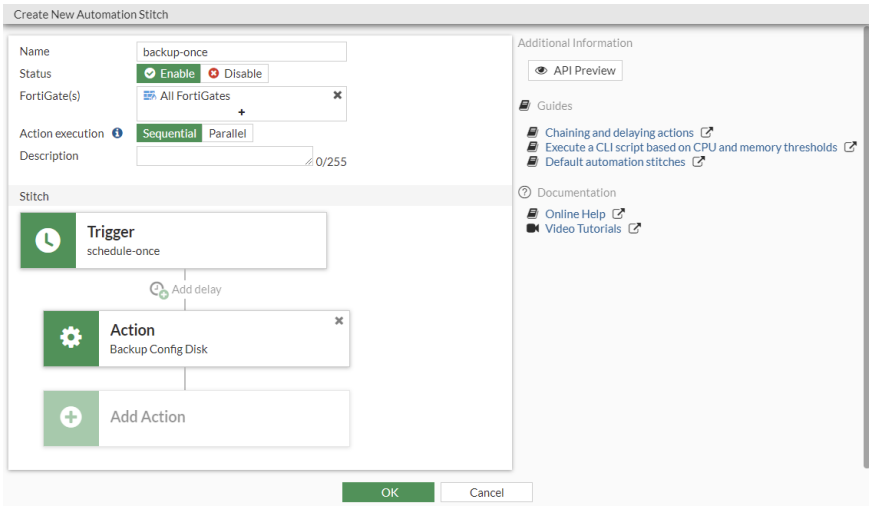
To schedule a one-time automation stitch in the GUI:

1. Configure the trigger:
   a. Go to *Security Fabric > Automation*, select the *Trigger* tab, and click *Create New*.
   b. In the *Miscellaneous* section, click *Schedule*.
   c. Enter a name (*schedule-once*) in the *Name* field.
   d. In the *Frequency* dropdown list, select *Once*.
   e. Select when the trigger will occur in the *Date/Time* fields.



f. Click *OK*.

2. Configure the stitch:
   a. Go to *Security Fabric > Automation*, select the *Stitch* tab, and click *Create New*.
   b. Enter the name, *backup-once*.
   c. Click *Add Trigger*. Select *schedule-once* and click *Apply*.
   d. Click *Add Action*. Select *Backup Config Disk* and click *Apply*.



e. Click *OK*. The backup configuration will occur once at the date and time you set.

To schedule a one-time automation stitch in the CLI:

1. Configure the trigger:

```
config system automation-trigger
    edit "schedule-once"
        set trigger-type scheduled
        set trigger-frequency once
        set trigger-datetime 2022-08-05 04:00:00
    next
end
```

2. Configure the action:

```
config system automation-action
    edit "Backup Config Disk"
```

```
        set description "Backup the configuration on disk."
        set action-type system-actions
        set system-action backup-config
    next
end
```

3. Configure the stitch:

```
config system automation-stitch
    edit "backup-once"
        set trigger "schedule-once"
        config actions
            edit 1
                set action "Backup Config Disk"
                set required enable
            next
        end
    next
end
```

4. To view automation stitch information:

```
# diagnose test application autod 3
stitch: backup-once (scheduled)

        local hit: 0 relayed to: 0 relayed from: 0
        last trigger:   last relay:
        next scheduled trigger:Fri Aug  5 05:00:00 2022
        actions:
                Backup Config Dis:
                        done: 0 relayed to: 0 relayed from: 0
                        last trigger:                   last relay:

logid to stitch mapping:
id:0 (scheduled stitches) local hit: 0 relayed hits: 0
        backup-once
```

Actions

There are two types of automation actions that can be configured in automation stitches: static and dynamic.

Static automation actions are included in FortiOS by default. They require only a name, description, and one setting. Static automation actions can be edited, but they cannot be deleted.

Dynamic automation actions require multiple settings to be configured. Dynamic automation actions can be created by clicking the *Create New* button on the *Action* tab, or clicking *Create* within the *Create Automation Stitch* page.

Multiple actions can be added to an automation stitch. Actions can be reorganized in the *Edit Automation Stitch* page by dragging and dropping the actions in the diagram.

The following table outlines the available static actions.

Action	Description
Access Layer Quarantine	This option is only available for Compromised Host triggers. Quarantine the MAC address on access layer devices (FortiSwitch and FortiAP).
FortiClient Quarantine	This option is only available for Compromised Host triggers. Use FortiClient EMS to block all traffic from the source addresses that are flagged as compromised hosts. Quarantined devices are flagged on the Security Fabric topology views. Go to the *Dashboard* > *Assets & Identities* > *Quarantine* widget to view and manage quarantined IP addresses.
FortiNAC Quarantine	This option is only available for Compromised Host and Incoming Webhook triggers. Use FortiNAC to quarantine a client PC and disable its MAC address. See FortiNAC Quarantine action on page 4534 for details.
IP Ban	This option is only available for Compromised Host triggers. Block all traffic from the source addresses flagged by the IoC. Go to the *Dashboard* > *Assets & Identities* > *Quarantine* widget to view and manage quarantined IP addresses.
System Action > Backup Config Disk	Back up the FortiGate's configuration. The default minimum interval is 0 seconds. See System actions on page 4593 for an example.
System Action > Reboot FortiGate	Reboot the FortiGate. The default minimum interval is 5 minutes (300 seconds in the CLI). See System actions on page 4593 for an example.
System Action > Shutdown FortiGate	Shut down the FortiGate. The default minimum interval is 0 seconds.

The following table outlines the available dynamic actions.

Category	Action	Description
Security Response		
	VMware NSX Security Tag	This option is only available for Compromised Host triggers. If an endpoint instance in a VMware NSX environment is compromised, the configured security tag is assigned to the compromised endpoint. See VMware NSX security tag action on page 4537 and VMware NSX-T security tag action on page 4542 for details.
Notifications		
	Email	Send a custom email message to the selected recipients. At least one recipient and an email subject must be specified.

Category	Action	Description
		Enable *Send to FortiCare email* to send the message to the email address associated with the FortiCare Support entitlement. This is the FortiCloud email address visible on the *System* > *FortiGuard* page under the *FortiCare Support* license information. The email body can use parameters from logs or previous action results. Wrapping the parameter with %% will replace the expression with the JSON value for the parameter, for example: *%%results.source%%* is the source property from the previous action. Replacement messages can be enabled in the email body to create branded email alerts. See Replacement messages for email alerts on page 4546 for details.
	FortiExplorer Notification	Send push notifications to FortiExplorer Go for iOS. The FortiGate must be registered to FortiCare on the mobile app that will receive the notification.
	Slack Notification	Send a notification to a Slack channel. See Slack Notification action on page 4550 for details.
	Microsoft Teams Notification	Send a notification to channels in Microsoft Teams. See Microsoft Teams Notification action on page 4555 for details.
Cloud Compute		
	AWS Lambda	Send log data to an integrated AWS service. See AWS Lambda action on page 4560 for details.
	Azure Function	Send log data to an Azure function. See Azure Function action on page 4562 for details.
	Google Cloud Function	Send log data to a Google Cloud function. See Google Cloud Function action on page 4564 for details.
	AliCloud Function	Send log data to an AliCloud function. See AliCloud Function action on page 4565 for details.
General		
	CLI Script	Run one or more CLI scripts. See CLI script action on page 4567 for details. See Execute a CLI script based on memory and CPU thresholds on page 4571 for an example.
	Webhook	Send an HTTP request using a REST callback. See Webhook action on page 4576 for details, and Slack integration webhook on page 4589 and Microsoft Teams integration webhook on page 4591 for examples.
	Alert	Generate a FortiOS dashboard alert. This option is only available in the CLI.

Category	Action	Description
	Disable SSID	Disable the SSID interface. This option is only available in the CLI.

Variables in actions

A variable can be used to extract information from a trigger event, or the results of a previous action. When the variable is used within an action, the information can be inserted into the content of the new action.

For example, in an automation stitch where a trigger event is an event log and the action is an email notification, the email action can use the `%%log%%` variable to add the log into the email body. In an automation stitch which executes a webhook action followed by an email action, the email action can use the `%%results%%` variable to place the webhook results into the email body.

Both the `%%log%%` and `%%results%%` variables can use sub-variables to drill down on a portion of the source.

For instance, using the following trigger event log:

```
date=2025-02-06 time=18:32:21 devid="FGVM04TM2400xxxx" devname="FGDocs"
eventtime=1738895541338331122 tz="-0800" logid="0100032002" type="event" subtype="system"
level="alert" vd="root" logdesc="Admin login failed" sn="0" user="abc" ui="https(192.168.2.204)"
method="https" srcip=192.168.2.204 dstip=192.168.2.87 action="login" status="failed"
reason="passwd_invalid" msg="Administrator abc login failed from https(192.168.2.204) because of
invalid password"
```

- `%%log.user%%` returns `abc`
- `%%log.status%%` returns `failed`
- `%%log.devname%%` returns `FGDocs`
- Sub-variables are determined by the log format and output.

Conversely, `%%results%%` requires an input from a previous action that returns a valid JSON payload. For example, this could be a JSON payload from a webhook action:

```
{
    "x": "example content x",
    "y": "example content y",
    "results": {
        "port1": {
            "ip": "10.10.10.15",
            "allowaccess": "https"
        }
    },
    "speeds": [
        100,
        1000,
        10000
    ]
}
```

- `%%results.result.port1.ip%%` returns `10.10.10.15`
- `%%results.result.port1.allowaccess%%` returns `https`

- `%%results.x%%` returns `example content x`
- `%%results[set_ip].y%%` returns the value of property `y` from a previous action called `set_ip`
- `%%results.speeds.1%%` returns the item in the array index 1 from the array `speeds`

FortiNAC Quarantine action

Users can configure an automation stitch with the FortiNAC Quarantine action with a Compromised Host or Incoming Webhook trigger. When the automation is triggered, the client PC will be quarantined and its MAC address is disabled in the configured FortiNAC.

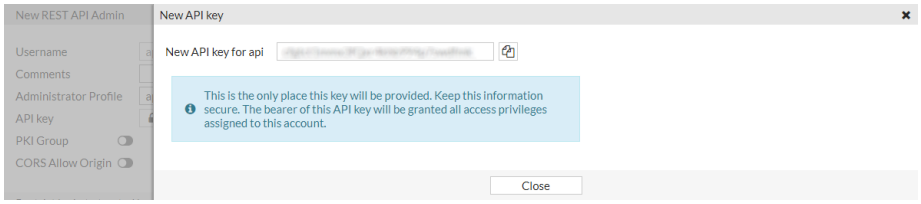
In this example, the FortiNAC has been configured to join an enabled Security Fabric. See Configuring FortiNAC on page 4385 for more information.

The FortiNAC must also be configured to isolate disabled hosts:

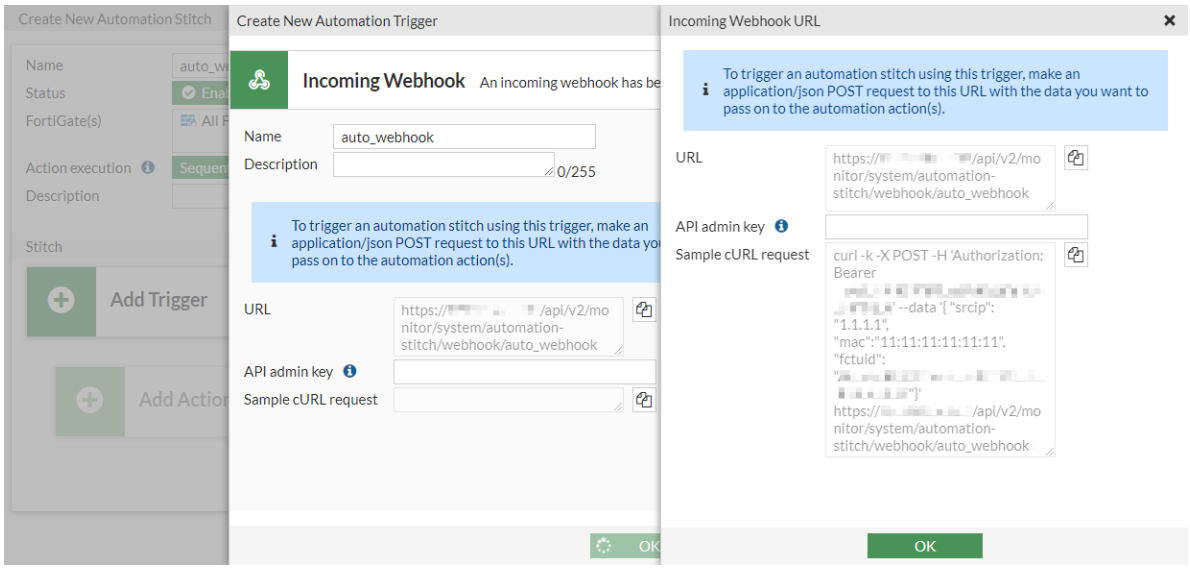
- Endpoints connecting to FortiWiFi or wired ports on FortiGate:
  - See the requisite *Configure FortiNAC* section in the FortiGate Endpoint Management Integration Guide.
- Endpoints connecting to FortiAP:
  - Set the *Dead End* VLAN. See Model configuration.
- Endpoints connecting to FortiSwitch:
  - Set the *Dead End* VLAN. See Model configuration.
  - Add the switch to the physical address filtering group. See Systems groups and Modify a group.

To configure an automation stitch with a FortiNAC quarantine action in the GUI:

1. Create a new API user and generate the API key:
   a. Go to *System* > *Administrators* and click *Create New* > *REST API Admin*.
   b. Configure the settings as needed.
   c. Click *OK*. The *New API key* window opens.
   d. Copy the key to the clipboard and click *Close*.



   e. Click *OK*.
2. Configure the automation stitch trigger:
   a. Go to *Security Fabric* > *Automation* and click *Create New*.
   b. Enter the stitch name (*auto_webhook*).
   c. Click *Add Trigger*.
   d. Click *Create* and select *Incoming Webhook*.
   e. Enter a name (*auto_webhook*).
   f. Click *OK*.
   g. Paste the key in the *API admin key* field.



h. Click *OK*.

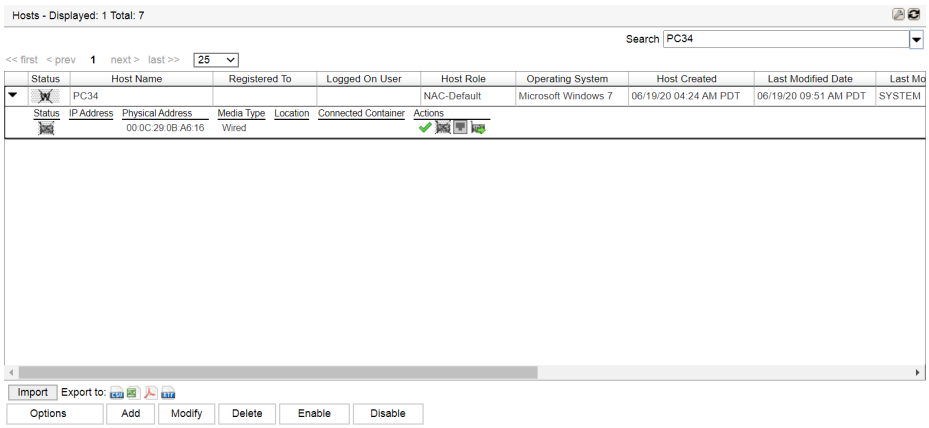
i. Select the trigger in the list and click *Apply*.

3. Configure the automation stitch action:
   a. Click *Add Action*.
   b. Select *FortiNAC Quarantine*.
   c. Click *Apply*.
   d. Click *OK*.
4. On a Linux PC accessible by the FortiGate, create a cURL request to trigger the automation stitch:

```
root@pc56:~# curl -k -X POST -H 'Authorization: Bearer ckx7d9xdzzx14Nztd1Ncr701dpwwy9' --data
'{ "srcip": "1.1.1.1", "mac":"00:0C:29:0B:A6:16", "fctuid":
"A8BA0B12DA694E47BA4ADF24F8358E2F"}'
https://172.17.48.225:4431/api/v2/monitor/system/automation-stitch/webhook/auto_webhook
```

5. In FortiOS, verify the automation stitch is triggered and the action is executed:
   a. Go to *Log & Report > System Events* to confirm that the stitch was activated.
   b. Go to *Security Fabric > Automation* to see the last time that the stitch was triggered.

   In FortiNAC, the *Host View* shows the status of the client PC. It is quarantined and its MAC address is disabled.



To configure an automation stitch with a FortiNAC quarantine action in the CLI:

1. Create a new API user and generate the API key:

```
config system api-user
    edit "g-api-rw-user"
        set api-key ************
        set accprofile "super_admin"
        set vdom "root"
        config trusthost
            edit 1
                set ipv4-trusthost 10.6.30.0 255.255.255.0
            next
        end
    next
end
```

2. Configure the automation trigger:

```
config system automation-trigger
    edit "auto_webhook"
        set event-type incoming-webhook
    next
end
```

3. Configure the automation action:

```
config system automation-action
    edit "FortiNAC Quarantine"
        set action-type quarantine-fortinac
    next
end
```

4. Configure the automation stitch:

```
config system automation-stitch
    edit "auto_webhook"
        set trigger "auto_webhook"
```

```
        config actions
            edit 1
                set action "FortiNAC Quarantine"
                set required enable
            next
        end
    next
end
```

5. On a Linux PC accessible by the FortiGate, create a cURL request to trigger the automation stitch:

```
root@pc56:~# curl -k -X POST -H 'Authorization: Bearer ckx7d9xdzzx14Nztd1Ncr701dpwwy9' --data
'{ "srcip": "1.1.1.1", "mac":"00:0C:29:0B:A6:16", "fctuid":
"A8BA0B12DA694E47BA4ADF24F8358E2F"}'
https://172.17.48.225:4431/api/v2/monitor/system/automation-stitch/webhook/auto_webhook
```

6. In FortiOS, verify that the automation stitch is triggered and the action is executed:

```
# diagnose test application autod 2
csf: enabled    root:yes
version:1592949233 sync time:Tue Jun 23 15:03:15 2020

total stitches activated: 1

stitch: auto_webhook
        destinations: all
        trigger: auto_webhook

                (id:15)service=auto_webhook

        local hit: 1 relayed to: 0 relayed from: 0
        actions:
                FortiNAC Quarantine type:quarantine-fortinac interval:0

date=2020-06-23 time=15:25:44 logdesc="Internal Message" path="system" name="automation-
stitch" action="webhook" mkey="auto_webhook" srcip="1.1.1.1" mac="00:0C:29:0B:A6:16"
fctuid="A8BA0B12DA694E47BA4ADF24F8358E2F" vdom="root" service="auto_webhook"

date=2020-06-23 time=15:25:44 logid="0100046600" type="event" subtype="system" level="notice"
vd="root" eventtime=1592951144401490054 tz="-0700" logdesc="Automation stitch triggered"
stitch="auto_webhook" trigger="auto_webhook" stitchaction="FortiNAC Quarantine" from="log"
msg="stitch:auto_webhook is triggered."
```

VMware NSX security tag action

If an endpoint instance in a VMware NSX environment is compromised, this action will assign the configured security tag to the compromised endpoint.

This action is only available when the automation trigger is set to compromised host.

To set up the NSX quarantine action, you need to:

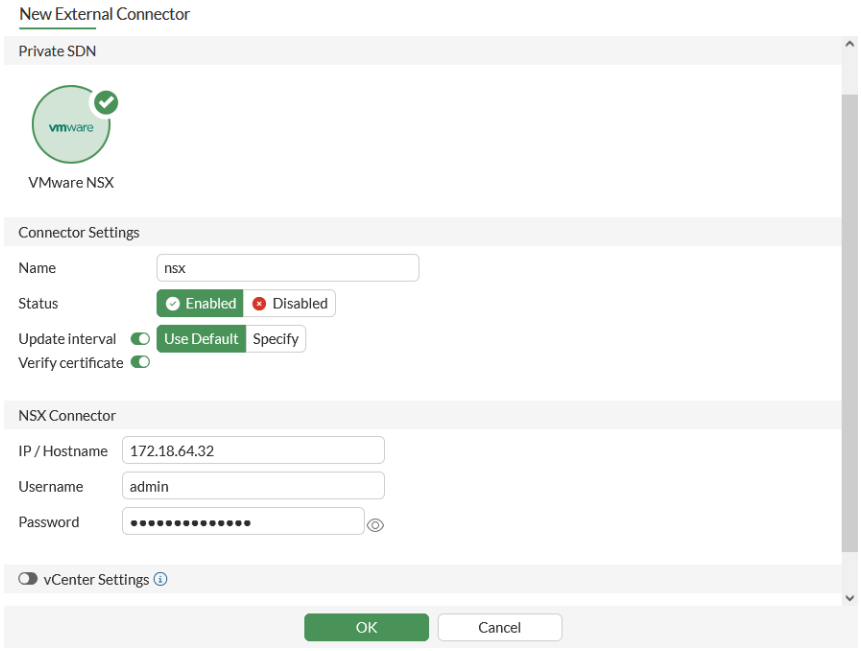
1. Configure a VMware NSX SDN connector
2. Configure an NSX security tag automation stitch
3. Configure FortiAnalyzer logging on the FortiGate

Configure a VMware NSX SDN connector

The FortiGate retrieves security tags from the VMware NSX server through the connector.

To configure a VMware NSX SDN connector in the GUI:

1. Go to *Security Fabric* > *External Connectors*.
2. Click *Create New*.
3. Select *VMware NSX*.
4. Configure the settings as needed.



5. Click *OK*.

To configure a VMware NSX SDN connector in the CLI:

```
config system sdn-connector
    edit "nsx"
        set type nsx
        set server "172.18.64.32"
        set username "admin"
        set password xxxxxxxxxxxx
    next
end
```

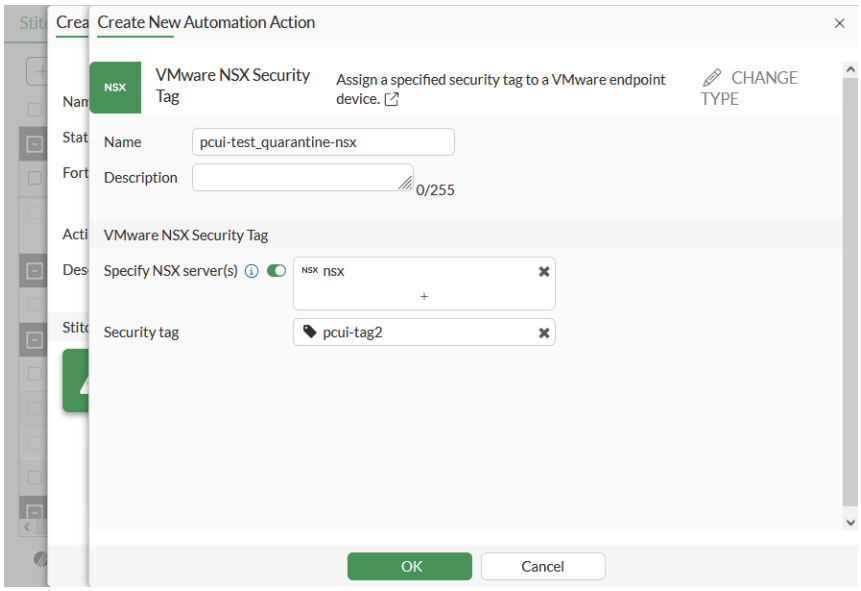
Configure an NSX security tag automation stitch

Security tags are retrieved from the VMware NSX server through the NSX SDN connector.

To configure an automation stitch with an NSX security tag in the GUI:

1. Go to *Security Fabric > Automation* and click *Create New*.
2. Enter the stitch name (*pcui-test*).
3. Configure the trigger:
   a. Click *Add Trigger*.
   b. Select *Compromised Host*.
   c. Click *Apply*.
4. Configure the VMware NSX Security Tag action:
   a. Click *Add Action*.
   b. Click *Create* and select *VMware NSX Security Tag*.
   c. Enter the following:

Name	pcui-test_quarantine-nsx
Specify NSX server(s)	Enable and select the SDN connector
Security tag	Select an existing tag, or create a new one



   d. Click *OK*.
   e. Select the action in the list and click *Apply*.
5. Click *OK*.

To configure an automation stitch with an NSX security tag in the CLI:

1. Create the automation trigger:

```
config system automation-trigger
    edit "Compromised Host"
    next
end
```

2. Create the automation action:

```
config system automation-action
    edit "pcui-test_quarantine-nsx"
        set action-type quarantine-nsx
        set security-tag "pcui-tag2"
        set sdn-connector "nsx"
    next
end
```

3. Create the automation stitch:

```
config system automation-stitch
    edit "pcui-test"
        set trigger "Compromised Host"
        config actions
            edit 1
                set action "pcui-test_quarantine-nsx"
                set required enable
            next
        end
    next
end
```

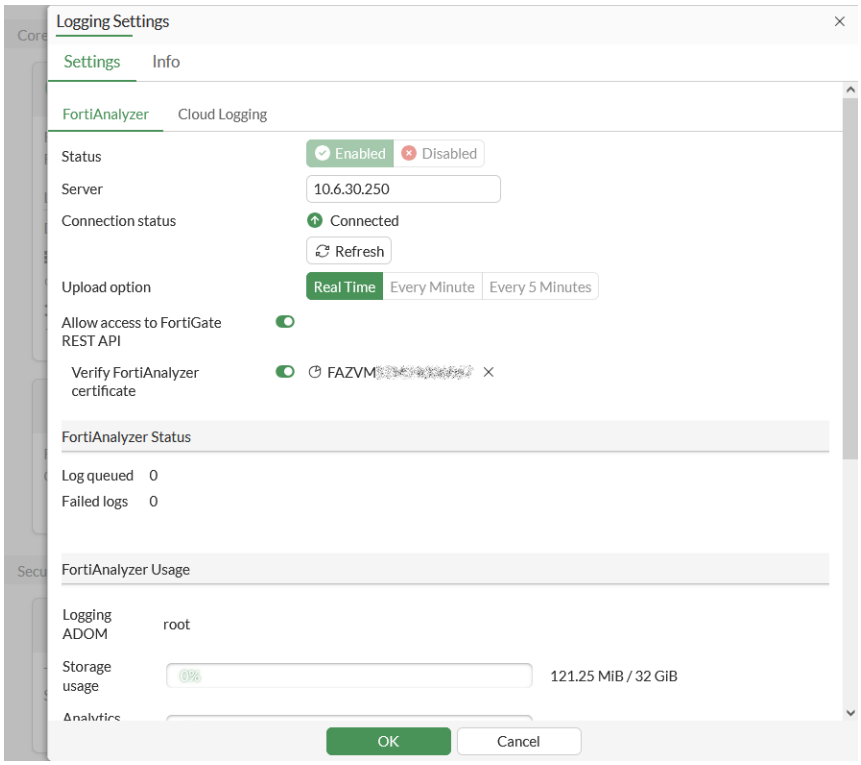
Configure FortiAnalyzer logging on the FortiGate

The FortiAnalyzer is used to send endpoint compromise notification to the FortiGate.

See Configuring FortiAnalyzer on page 4311 for more information.

To configure FortiAnalyzer logging in the GUI:

1. Go to *Security Fabric* > *Fabric Connectors* and double-click the *Logging & Analytics* card.
2. In the *FortiAnalyzer* tab, ensure the *Status* is *Enabled*, and configure the settings as needed.



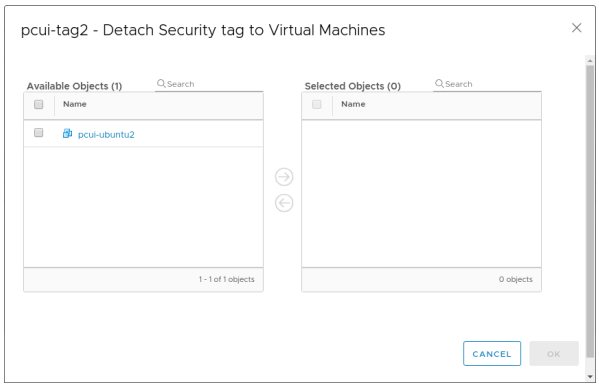
3. Click *OK*.

To configure FortiAnalyzer logging in the CLI:

```
config log fortianalyzer setting
    set status enable
    set server "172.18.64.234"
    set serial "FL-8HFT000000000"
    set upload-option realtime
    set reliable enable
end
```

When an endpoint instance is compromised

When an endpoint instance, such as *pcui-ubuntu2*, in the VMware NSX environment is compromised, the automation stitch is triggered. The FortiGate then assigns the configured security tag, *pcui-tag2* in this example, to the compromised NSX endpoint instance.



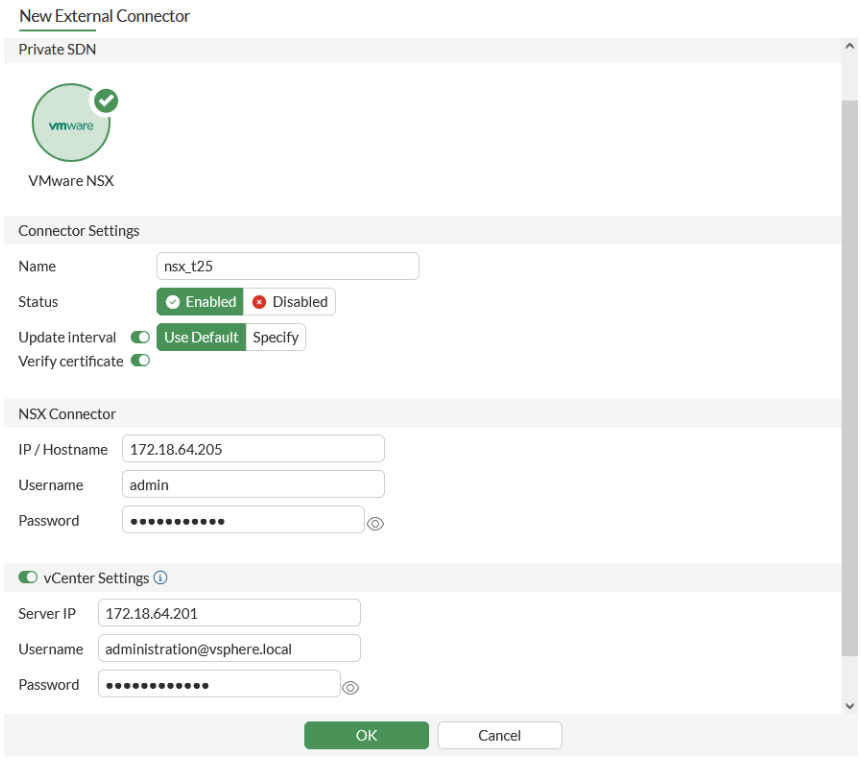
VMware NSX-T security tag action

VMware NSX SDN connectors' vCenter server and credentials can be configured so the FortiGate resolves NSX-T VMs. The FortiGate uses the VMWare NSX Security Tag automation action to assign a tag to the VM through an automation stitch.

The FortiGate is notified of a compromised host on the NSX-T network by an incoming webhook or other means, such as FortiGuard IOC. An automation stitch can be configured to process this trigger and action it by assigning a VMware NSX security tag on the VM instance.

To configure an automation stitch to assign a security tag to NSX-T VMs in the GUI:

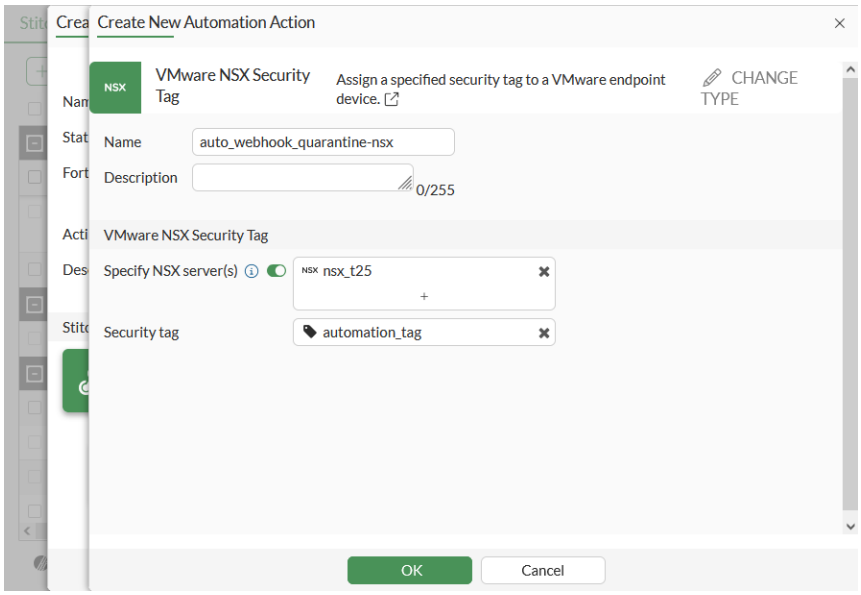
1. Configure the NSX SDN connector:
   a. Go to *Security Fabric* > *External Connectors* and click *Create New*.
   b. Select *VMware NSX*.
   c. Configure the connector settings.
   d. Enable *vCenter Settings* and configure as needed.



e. Click *OK*.

2. Configure the automation stitch trigger:
   a. Go to *Security Fabric > Automation* and click *Create New*.
   b. Enter the stitch name (*auto_webhook*).
   c. Click *Add Trigger*.
   d. Click *Create* and select *Incoming Webhook*.
   e. Enter a name (*auto_webhook*).
   f. Click *OK* to close the *Incoming Webhook URL* prompt.
   g. Select the trigger in the list and click *Apply*.
3. Configure the automation stitch action:
   a. Click *Add Action*.
   b. Click *Create* and select *VMware NSX Security Tag*.
   c. Enter the following:

Name	auto_webhook_quarantine-nsx
Specify NSX server(s)	Enable and select the SDN connector
Security tag	Select an existing tag, or create a new one



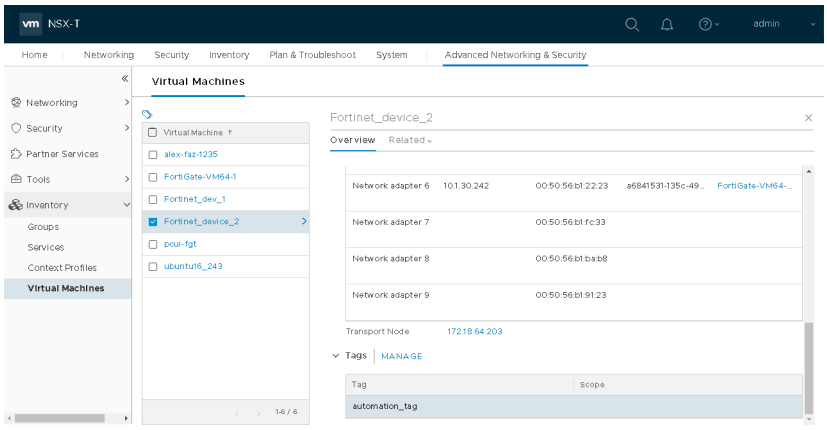
d. Click *OK*.

e. Select the action in the list and click *Apply*.

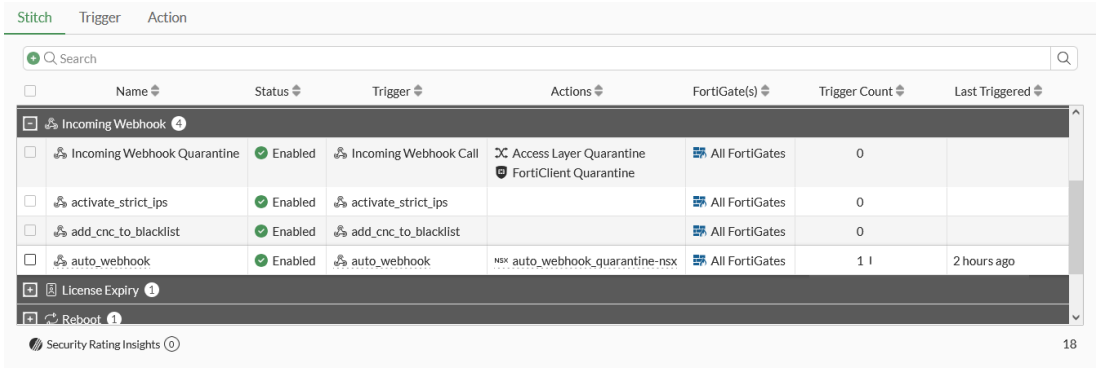
4. Click *OK*.
5. In NSX-T, create a cURL request to trigger the automation stitch on the FortiGate:

```
root@pc56:/home# curl -k -X POST -H 'Authorization: Bearer 3fdxNG08mgNg0fh4NQ51g1NQ1QHcxx' --
data '{ "srcip": "10.1.30.242"}' https://172.16.116.230/api/v2/monitor/system/automation-
stitch/webhook/auto_webhook
{
  "http_method":"POST",
  "status":"success",
  "http_status":200,
  "serial":"FGVM08TM20000000",
  "version":"v6.4.0",
  "build":1608
}
```

The automation stitch is triggered and the configured tag is added to the NSX-T VM.



In FortiOS, the *Security Fabric > Automation* page shows the last trigger time.



To configure an automation stitch to assign a security tag to NSX-T VMs in the CLI:

1. Configure the NSX SDN connector:

```
config system sdn-connector
    edit "nsx_t25"
        set type nsx
        set server "172.18.64.205"
        set username "admin"
        set password xxxxxxxxxxxx
        set vcenter-server "172.18.64.201"
        set vcenter-username "administrator@vsphere.local"
        set vcenter-password xxxxxxxxxxxx
    next
end
```

2. Configure the automation stitch:

```
config system automation-trigger
    edit "auto_webhook"
        set trigger-type event-based
        set event-type incoming-webhook
    next
end
```

```
config system automation-action
    edit "auto_webhook_quarantine-nsx"
        set action-type quarantine-nsx
        set security-tag "automation_tag"
        set sdn-connector "nsx_t25"
    next
end
```

```
config system automation-stitch
    edit "auto_webhook"
        set trigger "auto_webhook"
        config actions
            edit 1
                set action "auto_webhook_quarantine-nsx"
```

```
                set required enable
            next
        end
    next
end
```

3. In NSX-T, create a cURL request to trigger the automation stitch on the FortiGate:

```
root@pc56:/home# curl -k -X POST -H 'Authorization: Bearer 3fdxNG08mgNg0fh4NQ51g1NQ1QHcxx' --
data '{ "srcip": "10.1.30.242"}' https://172.16.116.230/api/v2/monitor/system/automation-
stitch/webhook/auto_webhook
{
  "http_method":"POST",
  "status":"success",
  "http_status":200,
  "serial":"FGVM08TM20000000",
  "version":"v6.4.0",
  "build":1608
}
```

To verify the automation stitch is triggered and the action is executed:

```
# diagnose test application autod 2

csf: enabled root:yes
version:1586883541 sync time:Tue Apr 14 11:04:05 2020

total stitches activated: 1

stitch: auto_webhook
destinations: all
trigger: auto_webhook

(id:15)service=auto_webhook

local hit: 1 relayed to: 0 relayed from: 0
actions:
auto_webhook_quarantine-nsx type:quarantine-nsx interval:0
security tag:automation_tag
sdn connector:
nsx_t25;
```

Replacement messages for email alerts

Automation stitches with an Email action can leverage the formatting options provided by replacement messages to create branded email alerts.

You can enable a replacement message and edit the message body or select a customized replacement message group when you configure the automation action. When the automation stitch is triggered, the FortiGate will send the email with the defined replacement message.

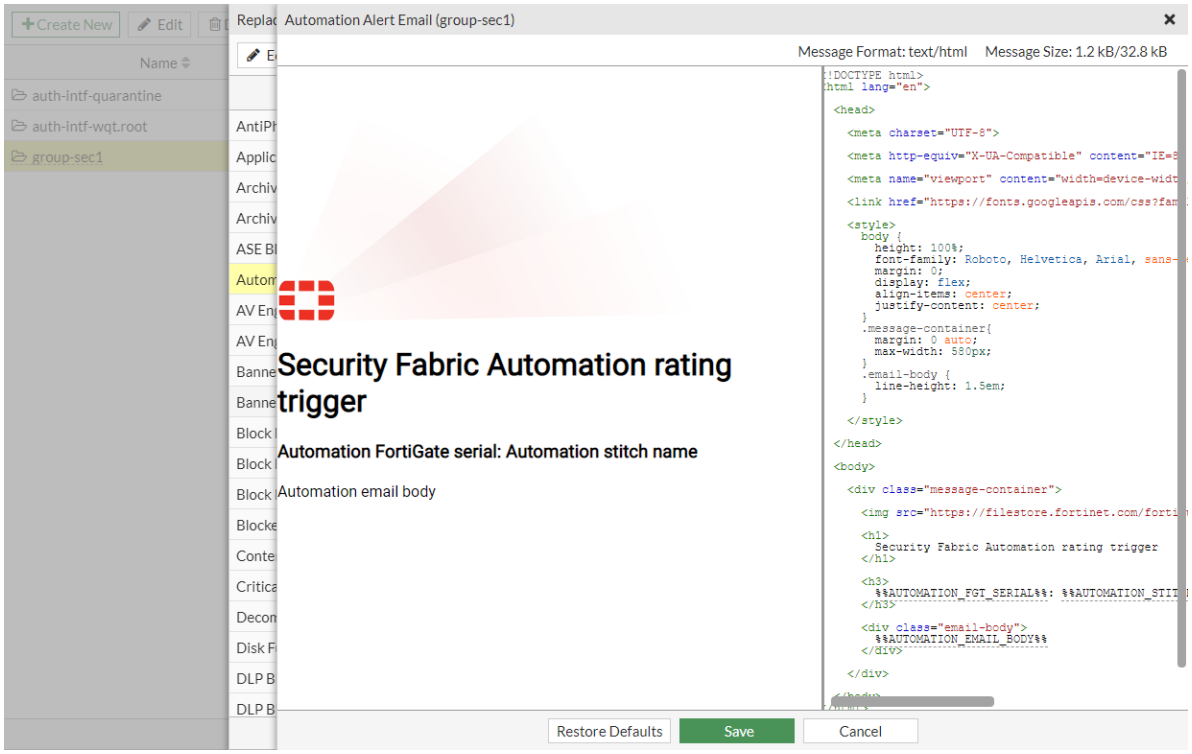
In this example, a Security Rating report triggers an Email notification action. The email uses a customized replacement message group.

To configure the replacement message group in the GUI:

1. Go to *System* > *Replacement Message Groups* and click *Create New*.
2. Enter the following:

Name	group-sec1
Group Type	Security

3. Click *OK*.
4. Select the group in the list and click *Edit*.
5. Select *Automation Alert Email* and click *Edit*.



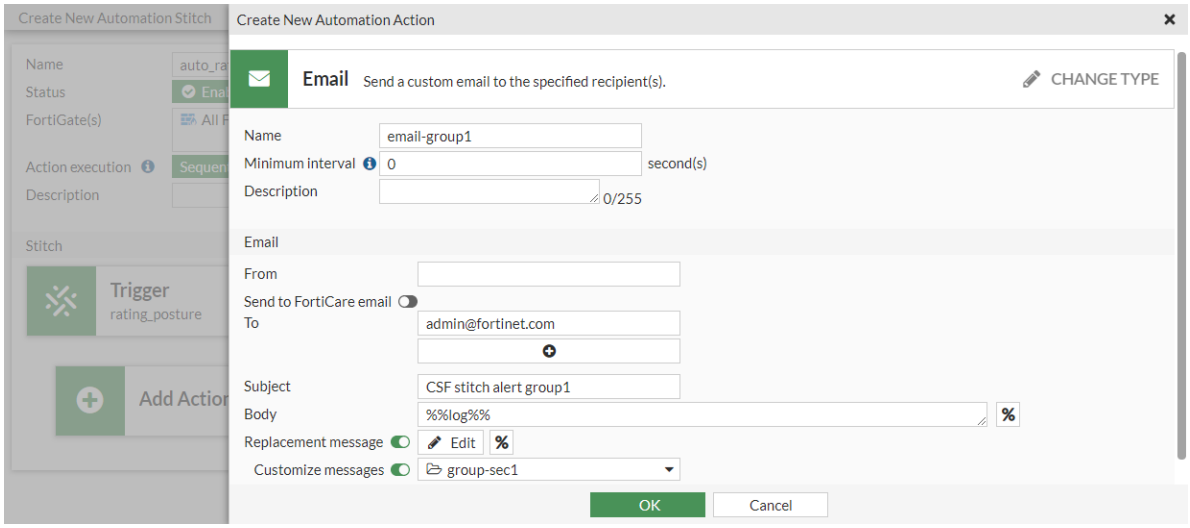
6. Edit the HTML code as needed, then click *Save*.

To configure the email action in the GUI:

1. Go to *Security Fabric* > *Automation* and click *Create New*.
2. Enter the stitch name.
3. Configure the trigger:
   a. Click *Add Trigger*.
   b. Select *Any Security Rating Notification*.
   c. Click *Apply*.
4. Configure the Email notification action:

a. Click *Add Action*.
b. Click *Create* and select *Email*.
c. Enter the following:

Name	email-group1
To	Enter an email address
Subject	CSF stitch alert group1
Replacement message	Enable
Customize messages	Enable and select group-sec1 from the dropdown



d. Click *OK*.
e. Select the action in the list and click *Apply*.

5. Click *OK*.
6. Right-click the automation stitch, and click *Test Automation Stitch*.
   After the Security Rating report is finished, the automation is triggered, and the email is delivered with the customized replacement message in the email body.

To configure the replacement message group in the CLI:

```
config system replacemsg-group
    edit "group-sec1"
        set comment ""
        set group-type utm
        config automation
            edit "automation-email"
                set buffer "...<h1> Security Fabric Automation rating trigger </h1>..."
                ...
            next
        end
    next
end
```

To configure the email action in the CLI:

1. Configure the automation trigger:

```
config system automation-trigger
    edit "Any Security Rating Notification"
        set event-type security-rating-summary
        set report-type any
    next
end
```

2. Configure the automation action:

```
config system automation-action
    edit "email-group1"
        set action-type email
        set email-to "admin@fortinet.com"
        set email-subject "CSF stitch alert group1"
        set replacement-message enable
        set replacemsg-group "group-sec1"
    next
end
```

3. Configure the automation stitch:

```
config system automation-stitch
    edit "auto_rating"
        set trigger "Any Security Rating Notification"
        config actions
            edit 1
                set action "email-group1"
                set required enable
            next
        end
    next
end
```

4. To view the automation stitch information after it is triggered:

```
# diagnose test application autod 3
stitch: auto_rating
        local hit: 1 relayed to: 0 relayed from: 0
        last trigger:Tue Mar 16 15:11:29 2021
        last relay:
        actions:
                email-group1:
                        done: 1 relayed to: 0 relayed from: 0
                        last trigger:Tue Mar 16 15:11:29 2021
                        last relay:

logid2stitch mapping:
id:52000  local hit: 1 relayed hits: 0
        auto_rating
```

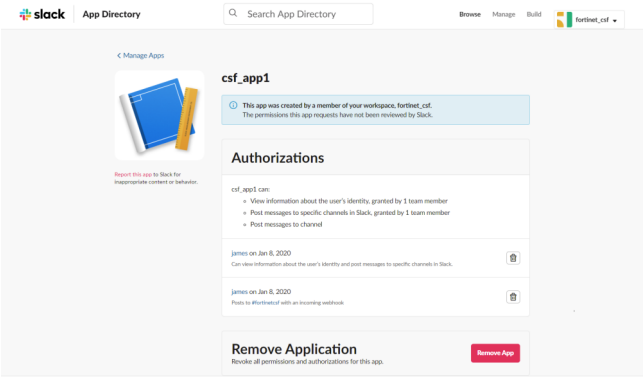
Slack Notification action

To configure an automation stitch with a Slack Notification action, you first need to configure an incoming webhook in Slack. Then you can enter the webhook URL when you configure the Slack Notification action.

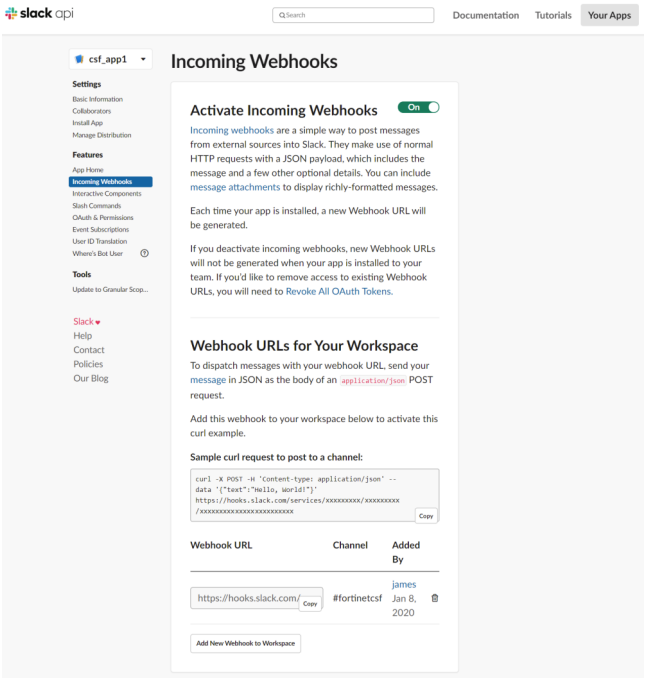
This example uses the default Any Security Rating Notification trigger in the automation stitch with two Slack Notification actions with different notification messages. One message is a custom message, and the other is for the Security Rating Summary log with a 90 second delay.

To create an Incoming Webhook in Slack:

1. Go to the Slack website, and create a workspace.
2. Create a Slack application for the workspace.



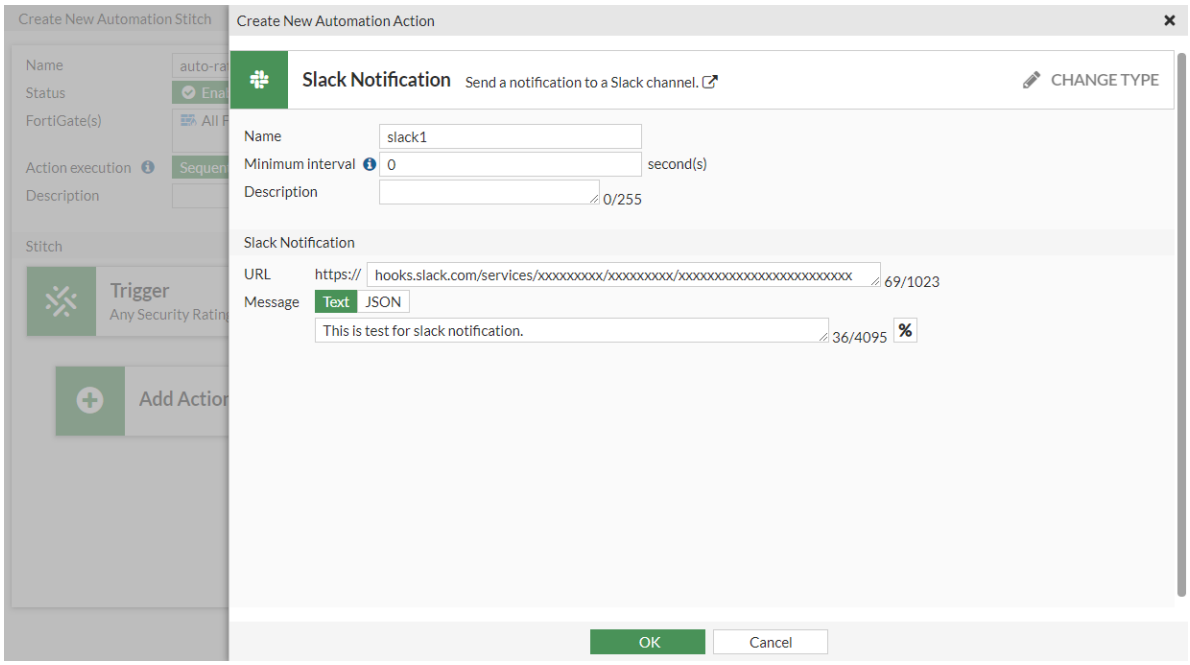
3. Add an Incoming Webhook to a channel in the workspace (see Sending messages using Incoming Webhooks for more details).
4. Activate the Incoming Webhook, and copy the *Webhook URL* to the clipboard.



To configure an automation stitch with Slack Notification actions in the GUI:

1. Go to *Security Fabric > Automation* and click *Create New*.
2. Enter the stitch name.
3. Configure the trigger:
   a. Click *Add Trigger*.
   b. Select *Any Security Rating Notification*.
   c. Click *Apply*.
4. Configure the first Slack Notification action:
   a. Click *Add Action*.
   b. Click *Create* and select *Slack Notification*.
   c. Enter the following:

Name	slack1
URL	Paste the webhook URL from the clipboard
Message	Text
Message text	This is test for slack notification.



d. Click *OK*.

e. Select the action in the list and click *Apply*.

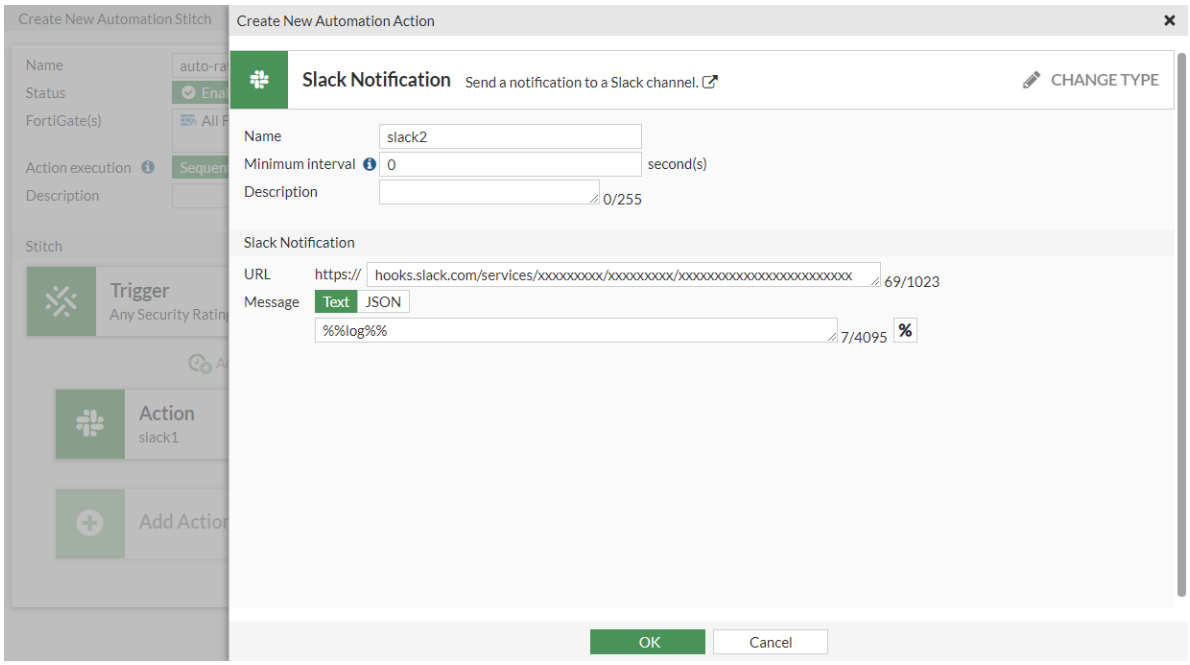
5. Configure the second Slack Notification action:

a. Click *Add Action*.

b. Click *Create* and select *Slack Notification*.

c. Enter the following:

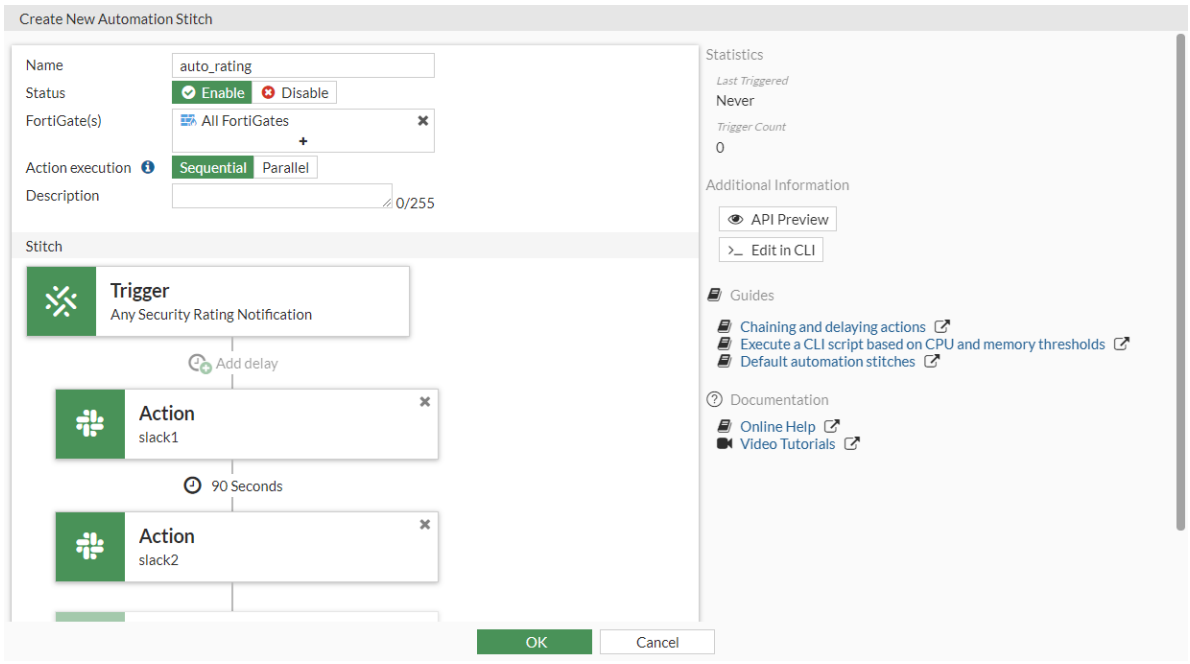
Name	slack2
URL	Paste the webhook URL from the clipboard
Message	Text
Message text	%%log%%



d. Click *OK*.

e. Select the action in the list and click *Apply*.

f. Click the *Add delay* located between both actions. Enter *90* and click *OK*.

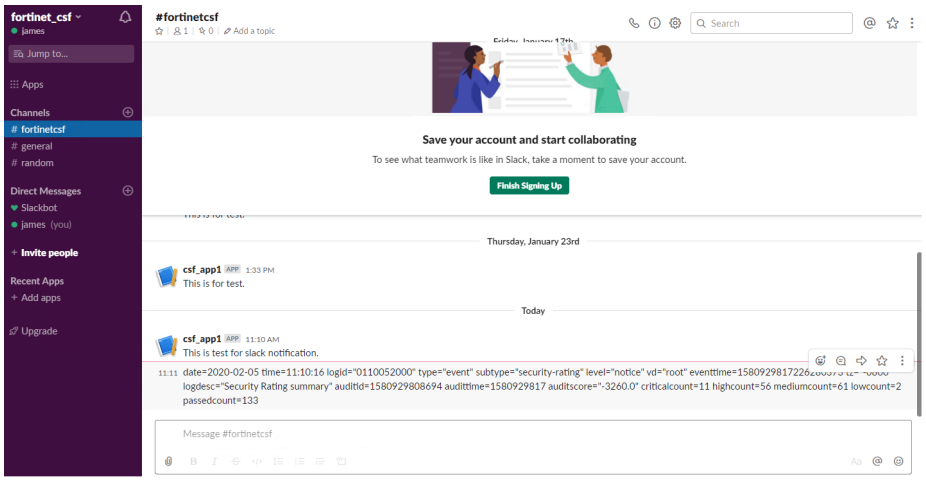


6. Click *OK*.

7. Trigger the automation stitch:

a. Right-click the automation stitch and select *Test Automation Stitch*.

After the Security Rating report is finished, the automation is triggered and an event log is created by the FortiGate. The two notifications are sent to the Slack channel.



To configure an automation stitch with Slack Notification actions in the CLI:

1. Configure the automation trigger:

```
config system automation-trigger
    edit "Any Security Rating Notification"
        set event-type security-rating-summary
        set report-type any
    next
end
```

2. Configure the automation actions:

```
config system automation-action
    edit "slack1"
        set action-type slack-notification
        set message "This is test for slack notification."
        set uri "hooks.slack.com/services/xxxxxxxxx/xxxxxxxxx/xxxxxxxxxxxxxxxxxxxxxxxx"
    next
    edit "slack2"
        set action-type slack-notification
        set uri "hooks.slack.com/services/xxxxxxxxx/xxxxxxxxx/xxxxxxxxxxxxxxxxxxxxxxxx"
    next
end
```

3. Configure the automation stitch:

```
config system automation-stitch
    edit "auto_rating"
        set trigger "Any Security Rating Notification"
        config actions
            edit 1
                set action "slack1"
                set required enable
            next
            edit 2
                set action "slack2"
```

```
                set delay 90
                set required enable
            next
        end
    next
end
```

4. Verify that the automation action was triggered:

```
# diagnose test application autod 3
stitch: auto-rating
    local hit: 1 relayed to: 0 relayed from: 0
    last trigger:Wed Feb 05 11:10:23 2020
    last relay:
    actions:
        slack1:
            done: 1 relayed to: 0 relayed from: 0
            last trigger:Wed Feb 11:10:23 2020
            last relay:
        slack2:
            done: 1 relayed to: 0 relayed from: 0
            last trigger:Wed Feb 05 11:10:23 2020
            last relay:
```

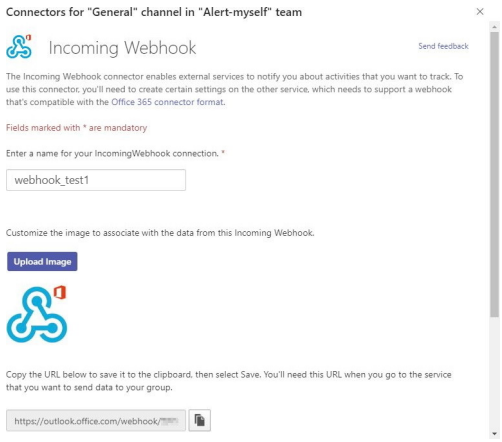
Microsoft Teams Notification action

Microsoft Teams Notification actions can be configured to send notifications to channels in Microsoft Teams. To trigger the notifications, you need to add an Incoming Webhook connector to a channel in Microsoft Teams, then you can configure the automation stitch with the webhook URL.

In the following example, you will configure an automation stitch with the default Any Security Rating Notification trigger and two Microsoft Teams Notification actions with different notification messages. One message is for the Security Rating Summary log, and the other is a custom message with a ten second delay.

To add the Incoming Webhook connector in a Microsoft Teams channel:

1. In Microsoft Teams, click the ... (*More options*) beside the channel name, and select *Connectors*.
2. Search for *Incoming Webhook* and click *Configure*.
3. Enter a name for the webhook, upload an image for the webhook, and click *Create*.
4. Copy the webhook to the clipboard and save it.

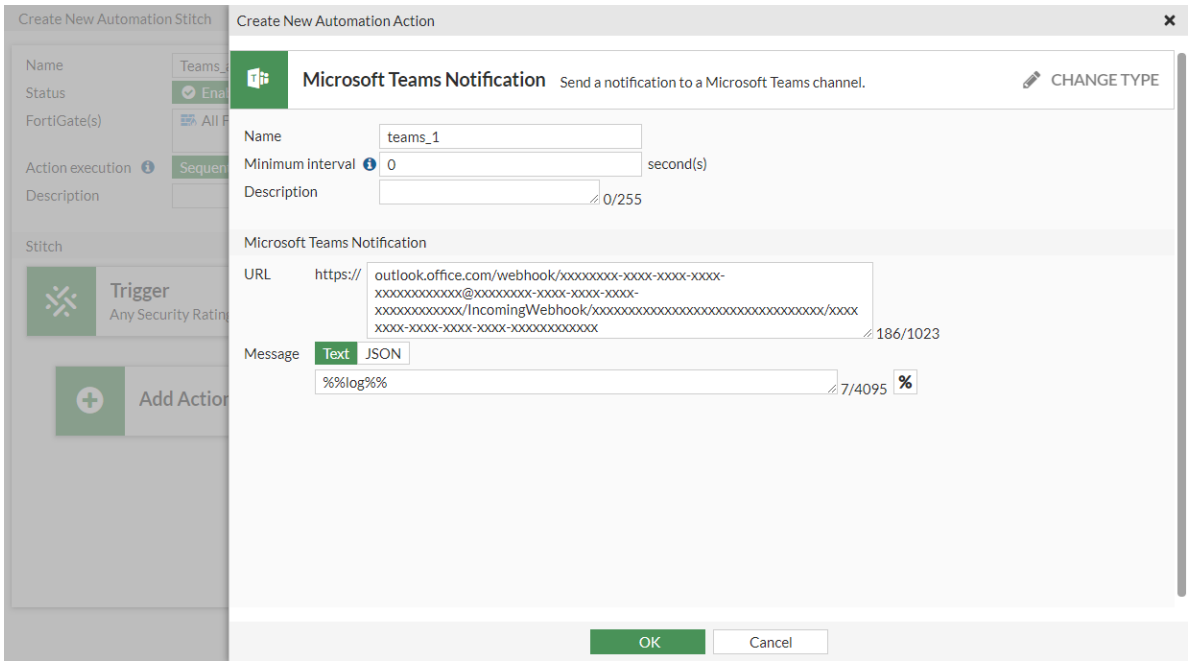


5. Click *Done*.

To configure an automation stitch with Microsoft Teams Notification actions in the GUI:

1. Go to *Security Fabric > Automation* and click *Create New*.
2. Enter the stitch name.
3. Configure the trigger:
   a. Click *Add Trigger*.
   b. Select *Any Security Rating Notification*.
   c. Click *Apply*.
4. Configure the first Microsoft Teams Notification action:
   a. Click *Add Action*.
   b. Click *Create* and select *Microsoft Teams Notification*.
   c. Enter the following:

Name	teams_1
URL	Paste the webhook URI from the clipboard
Message	Text
Message text	%%log%%



d. Click *OK*.

e. Select the action in the list and click *Apply*.

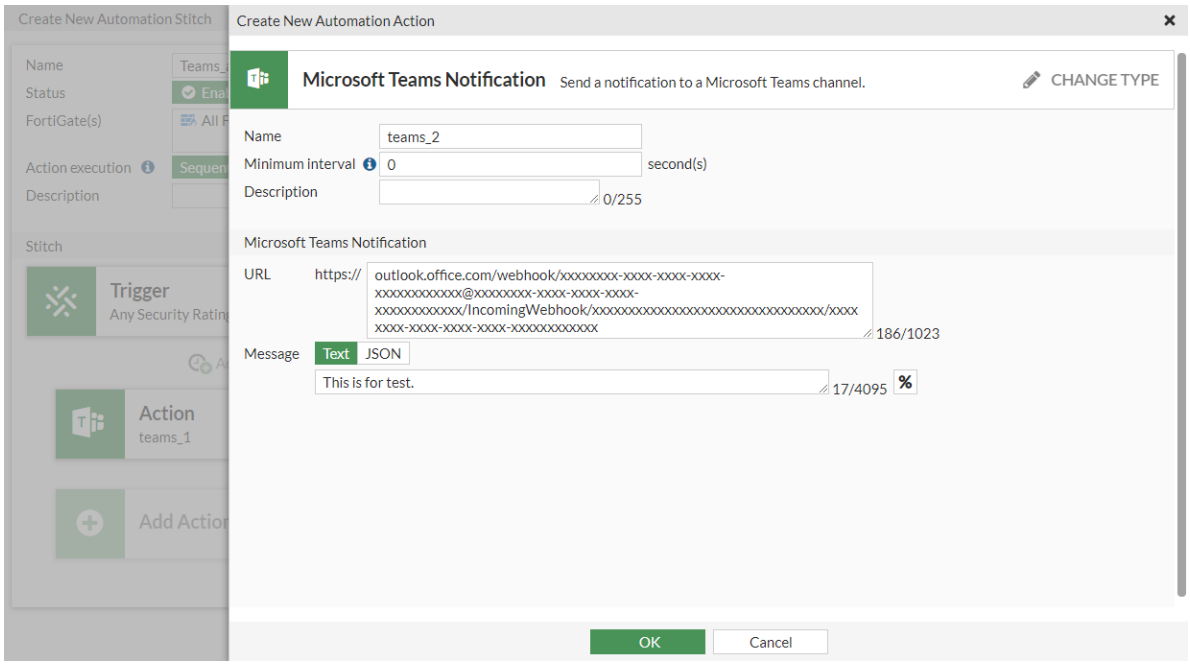
5. Configure the second Microsoft Teams Notification action:

a. Click *Add Action*.

b. Click *Create* and select *Microsoft Teams Notification*.

c. Enter the following:

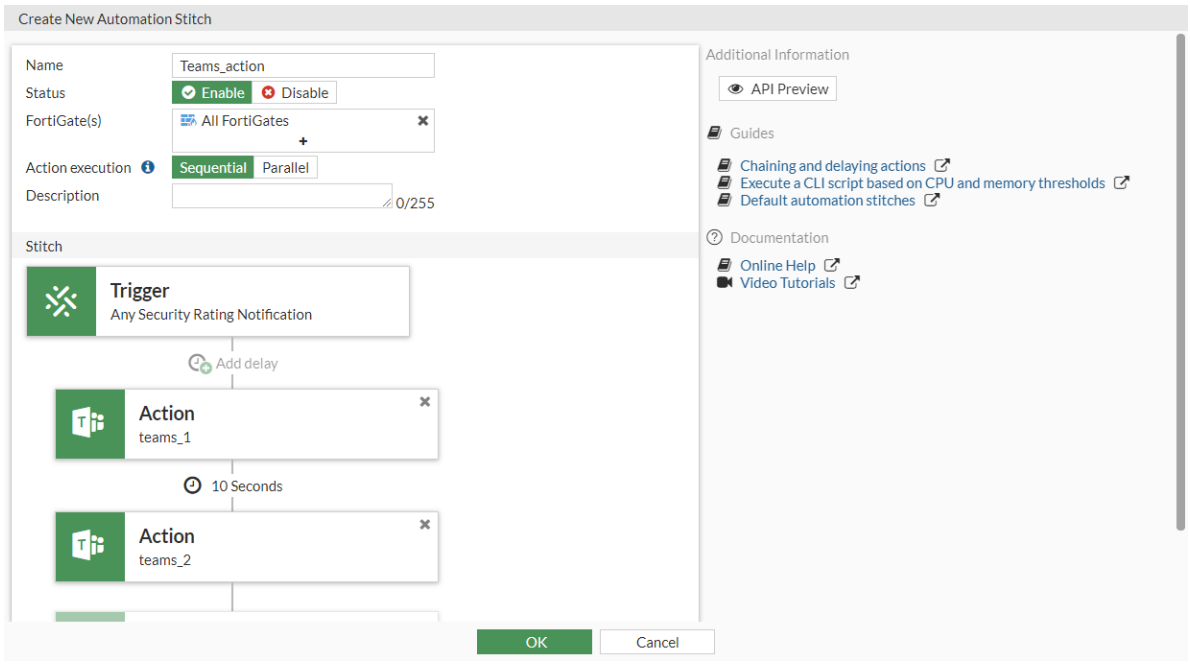
Name	teams_2
URL	Paste the webhook URI from the clipboard
Message	Text
Message text	This is for test.



d. Click *OK*.

e. Select the action in the list and click *Apply*.

f. Click the *Add delay* located between both actions. Enter *10* and click *OK*.

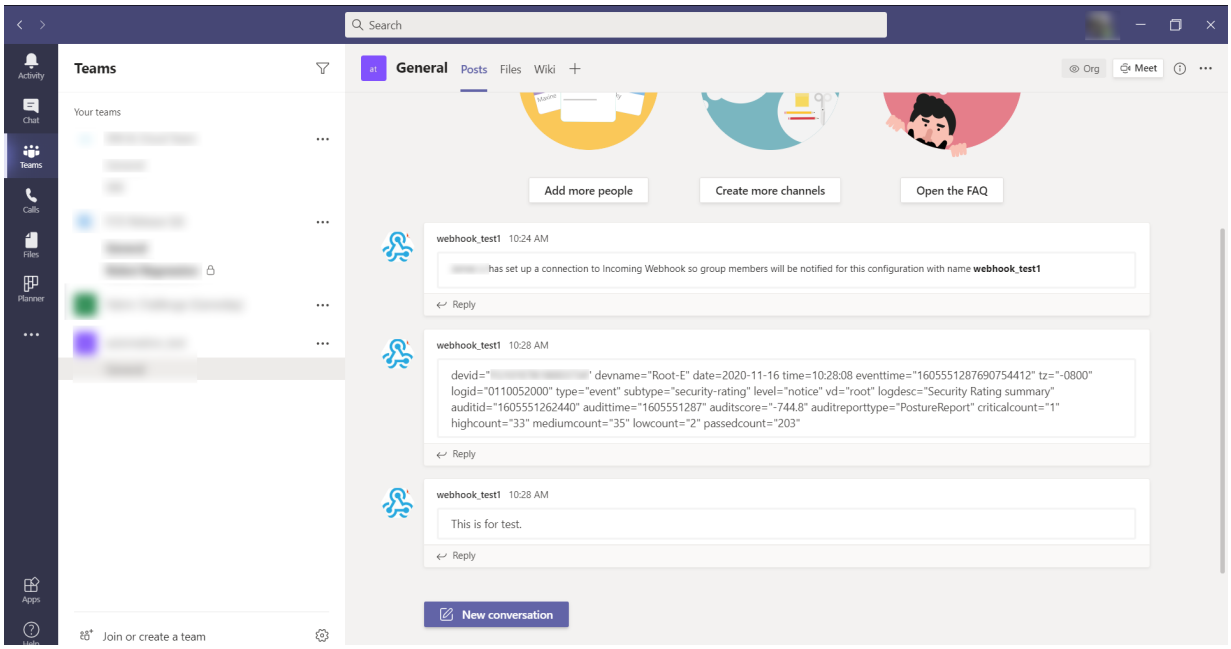


6. Click *OK*.

7. Trigger the automation stitch:

a. Right-click the automation stitch and select *Test Automation Stitch*.

After the Security Rating report is finished, the automation is triggered and an event log is created by the FortiGate. The two notifications are sent to the Microsoft Teams channel.



To configure an automation stitch with Microsoft Teams Notification actions in the CLI:

1. Configure the automation trigger:

```
config system automation-trigger
    edit "Any Security Rating Notification"
        set event-type security-rating-summary
        set report-type any
    next
end
```

2. Configure the automation actions:

```
config system automation-action
    edit "teams_1"
        set action-type microsoft-teams-notification
        set uri "outlook.office.com/webhook/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx@xxxxxxxx-
xxxx-xxxx-xxxx-xxxxxxxxxxxx/IncomingWebhook/xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/xxxxxxxx-xxxx-
xxxx-xxxx-xxxxxxxxxxxx"
    next
    edit "teams_2"
        set action-type microsoft-teams-notification
        set message "This is for test."
        set uri "outlook.office.com/webhook/xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx@xxxxxxxx-
xxxx-xxxx-xxxx-xxxxxxxxxxxx/IncomingWebhook/xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx/xxxxxxxx-xxxx-
xxxx-xxxx-xxxxxxxxxxxx"
    next
end
```

3. Configure the automation stitch:

```
config system automation-stitch
    edit "Teams_action"
        set trigger "Any Security Rating Notification"
        config actions
            edit 1
                set action "teams_1"
                set required enable
            next
            edit 2
                set action "teams_2"
                set delay 10
                set required enable
            next
        end
    next
end
```

4. Verify that the automation action was triggered:

```
# diagnose test application autod 3
stitch: Teams_action
    local hit: 2 relayed to: 0 relayed from: 0
    last trigger:Mon Nov 16 10:28:08 2020
    last relay:
    actions:
        teams_1:
        done: 2 relayed to: 0 relayed from: 0
            last trigger:Mon Nov 16 10:28:08 2020
            last relay:
        teams_2:
        done: 2 relayed to: 0 relayed from: 0
            last trigger:Mon Nov 16 10:28:08 2020
            last relay:
logid2stitch mapping:
id:52000  local hit: 22 relayed hits: 0
Teams_action
```

AWS Lambda action

AWS Lambda functions can be called when an automation stitch is triggered. This example uses the default Any Security Rating Notification trigger in the automation stitch.

To configure an AWS Lambda function automation stitch in the GUI:

1. Go to *Security Fabric* > *Automation* and click *Create New*.
2. Enter the stitch name.
3. Configure the trigger:
   a. Click *Add Trigger*.
   b. Select *Any Security Rating Notification*.
   c. Click *Apply*.

4. Configure the AWS Lambda function action:
   a. Click *Add Action*.
   b. Click *Create* and select *AWS Lambda*.
   c. Enter the following:

Name	aws-action-1
URL	Enter the request API URI
API key	Enter the API key
HTTP header	header2 : header2_value

   d. Click *OK*.
   e. Select the action in the list and click *Apply*.
5. Click *OK*.

To configure an AWS Lambda function automation stitch in the CLI:

1. Create the automation trigger:

```
config system automation-trigger
    edit "Any Security Rating Notification"
        set event-type security-rating-summary
        set report-type any
    next
end
```

2. Create the automation action:

```
config system automation-action
    edit "aws-action-1"
        set action-type aws-lambda
        set aws-api-key *************
        set uri "0100000000.execute-api.us-east-2.amazonaws.com/default/xxxxx-autobatoon-XXX-
lambdaXXX"
        config http-headers
            edit 1
                set key "header2"
                set value "header2_value"
            next
        end
    next
end
```

3. Create the automation stitch:

```
config system automation-stitch
    edit "auto-aws"
        set trigger "Any Security Rating Notification"
        config actions
            edit 1
                set action "aws-action-1"
```

```
                set required enable
            next
        end
    next
end
```

When the automation stitch is triggered, the *Security Fabric > Automation* page shows the stitch trigger time. In AWS, the log shows that the function was called, executed, and finished.

Azure Function action

Azure functions can be called when an automation stitch is triggered. This example uses the default Any Security Rating Notification trigger in the automation stitch.

To configure an Azure function automation stitch in the GUI:

1. Go to *Security Fabric > Automation* and click *Create New*.
2. Enter the stitch name.
3. Configure the trigger:
   a. Click *Add Trigger*.
   b. Select *Any Security Rating Notification*.
   c. Click *Apply*.
4. Configure the Azure Function action:
   a. Click *Add Action*.
   b. Click *Create* and select *Azure Function*.
   c. Enter the following:

Name	azure_function
URL	Enter the request API URI
Authorization	Function
API key	Enter the API key
HTTP header	header1 : value1

   d. Click *OK*.
   e. Select the action in the list and click *Apply*.
5. Click *OK*.

To configure an Azure function automation stitch in the CLI:

1. Create an automation trigger:

```
config system automation-trigger
    edit "Any Security Rating Notification"
        set event-type security-rating-summary
        set report-type any
```

```
    next
end
```

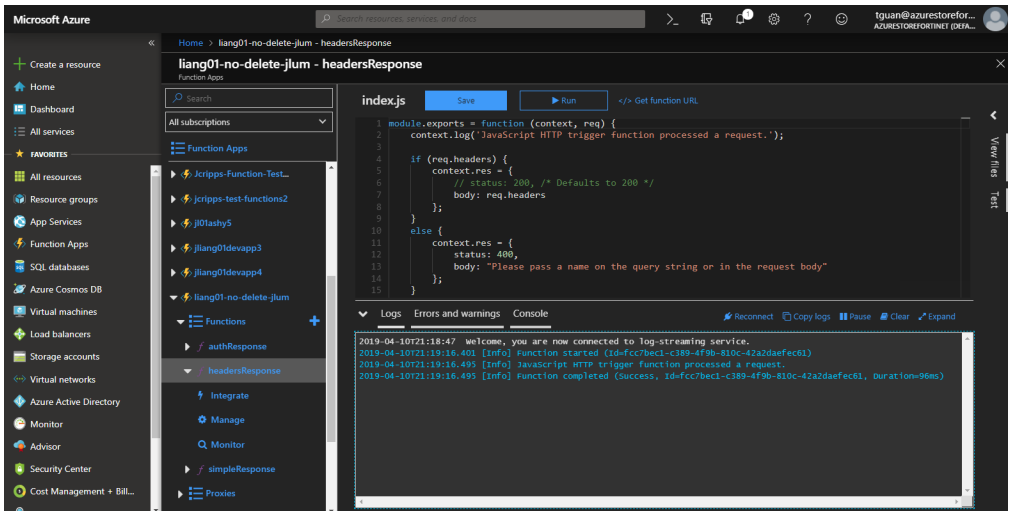
2. Create an automation action:

```
config system automation-action
    edit "azure_function"
        set action-type azure-function
        set azure-function-authorization function
        set azure-api-key **********
        set uri "xxxxx00-no-delete-xxxx.azurewebsites.net/api/headersResponse"
        config http-headers
            edit 1
                set key "header1"
                set value "value1"
            next
        end
    next
end
```

3. Create the automation stitch:

```
config system automation-stitch
    edit "auto-azure"
        set trigger "Any Security Rating Notification"
        config actions
            edit 1
                set action "azure_function"
                set required enable
            next
        end
    next
end
```

When the automation stitch is triggered, the *Security Fabric > Automation* page shows the stitch trigger time. In Azure, the function log shows that the function was called, executed, and finished:



Google Cloud Function action

Google Cloud functions can be called when an automation stitch is triggered. This example uses the default Any Security Rating Notification trigger in the automation stitch.

To configure a Google Cloud function automation stitch in the GUI:

1. Go to *Security Fabric > Automation* and click *Create New*.
2. Enter the stitch name.
3. Configure the trigger:
   a. Click *Add Trigger*.
   b. Select *Any Security Rating Notification*.
   c. Click *Apply*.
4. Configure the Google Cloud Function action:
   a. Click *Add Action*.
   b. Click *Create* and select *Google Cloud Function*.
   c. Enter the following:

Name	google-echo
URL	Enter the request API URI
HTTP header	echo-header : echo-value

   d. Click *OK*.
   e. Select the action in the list and click *Apply*.
5. Click *OK*.

To configure a Google Cloud function automation stitch in the CLI:

1. Create an automation trigger:

```
config system automation-trigger
    edit "Any Security Rating Notification"
        set event-type security-rating-summary
        set report-type any
    next
end
```

2. Create an automation action:

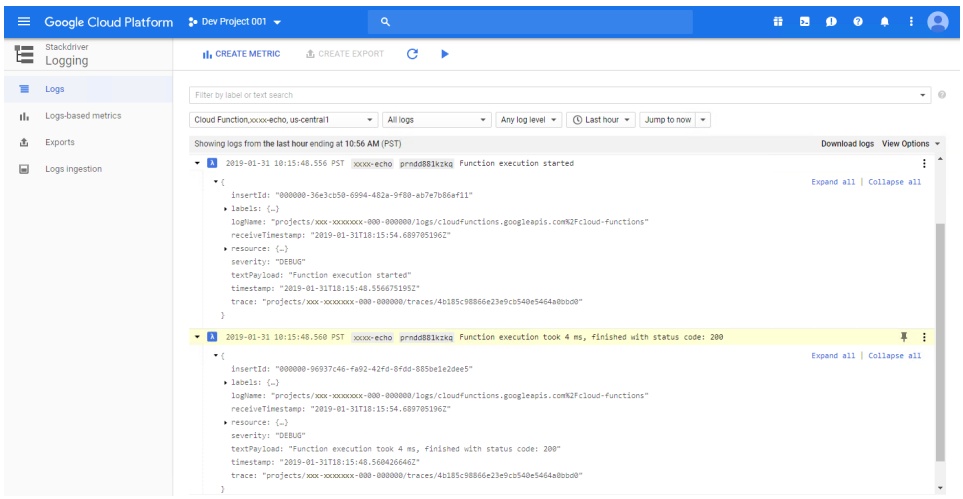
```
config system automation-action
    edit "google-echo"
        set action-type google-cloud-function
        set uri "us-central1-xxx-xxxxxxx-000-000000.cloudfunctions.net/xxxx-echo"
        config http-headers
            edit 1
                set key "echo-header"
                set value "echo-value"
            next
```

```
        end
    next
end
```

3. Create the automation stitch:

```
config system automation-stitch
    edit "auto-google1"
        set trigger "Any Security Rating Notification"
        config actions
            edit 1
                set action "google-echo"
                set required enable
            next
        end
    next
end
```

When the automation stitch is triggered, the *Security Fabric > Automation* page shows the stitch trigger time. In Google Cloud, go to *Logs* to see the function log showing that the configured function was called, executed, and finished:



AliCloud Function action

AliCloud functions can be called when an automation stitch is triggered. This example uses the default Any Security Rating Notification trigger in the automation stitch.

To configure an AliCloud function automation stitch in the GUI:

1. Go to *Security Fabric > Automation* and click *Create New*.
2. Enter the stitch name.
3. Configure the trigger:
   a. Click *Add Trigger*.
   b. Select *Any Security Rating Notification*.

c. Click *Apply*.

4. Configure the AliCloud Function action:
   a. Click *Add Action*.
   b. Click *Create* and select *AliCloud Function*.
   c. Enter the following:

Name	Ali-Action-1
URL	Enter the request API URI
Authorization	Function
AccessKey ID	Enter the access key ID
AccessKey Secret	Enter the access key secret

   d. Click *OK*.
   e. Select the action in the list and click *Apply*.
5. Click *OK*.

To configure an AliCloud function automation stitch in the CLI:

1. Create an automation trigger:

```
config system automation-trigger
    edit "Any Security Rating Notification"
        set event-type security-rating-summary
        set report-type any
    next
end
```

2. Create an automation action:

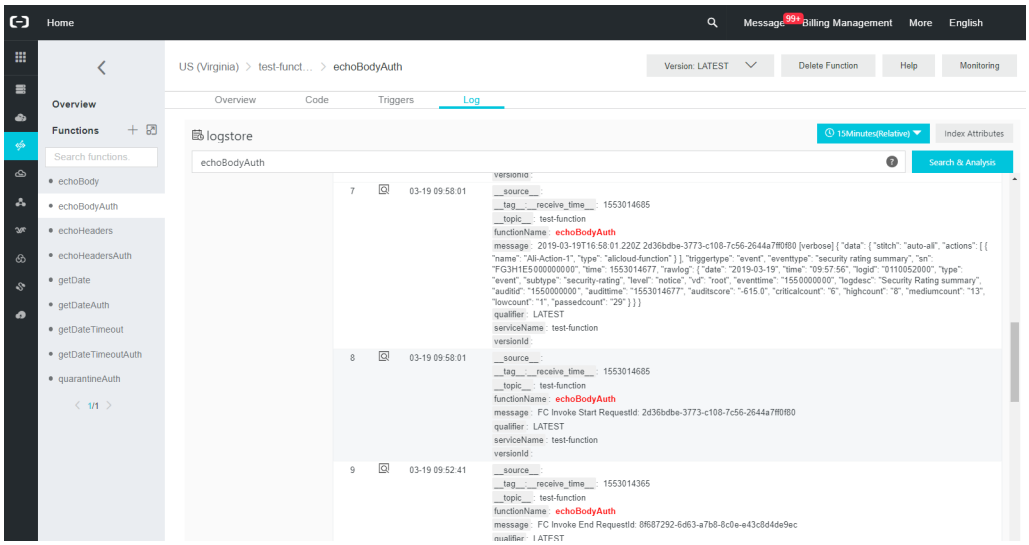
```
config system automation-action
    edit "Ali-Action-1"
        set action-type alicloud-function
        set alicloud-function-authorization function
        set alicloud-access-key-id "XXXXXxXXXXxxxxxx"
        set alicloud-access-key-secret xxxxxx
        set uri "0000000000000000.us-east-1.fc.aliyuncs.com/2099-99-99/proxy/test-
function/echoBodyAuth/"
    next
end
```

3. Create the automation stitch:

```
config system automation-stitch
    edit "auto-ali"
        set trigger "Any Security Rating Notification"
        config actions
            edit 1
                set action "Ali-Action-1"
                set required enable
```

```
            next
        end
    next
end
```

When the automation stitch is triggered, the *Security Fabric > Automation* page shows the stitch trigger time. In AliCloud, the function log shows that the function was called, executed, and finished:



CLI script action

The CLI script action can run when an automation stitch is triggered. It executes a series of commands in the CLI, as defined by the administrator. The scripts commands can be entered manually, uploaded as a file, or recorded in the CLI console. The output of the script can be fed as a variable (`%%results%%`) into the next action in the stitch. This could then be sent as an email using the email action, for example.

The CLI script action utilizes the *auto-script* feature to perform the execution of the script commands. The output size of the auto-script feature controls the size of the output for the script execution (10MB by default). This output is read into a buffer for use by the automation stitch action, and that buffer is limited to 192K characters for the email action. This means that the total allowable limit for CLI script output that is used in an email action is 192K characters. The buffer sizes for other actions may vary, for example the buffer to use the results in a webhook action could be a different size.

```
config system automation-action
    edit <name>
        set action-type cli-script
        set minimum-interval <integer>
        set script <string>
        set output-size <integer>
        set timeout <integer>
        set execute-security-fabric {enable | disable}
        set accprofile <profile>
```

```
    next
end
```

`minimum-interval <integer>`	Limit execution to no more than once in this interval, in seconds (0 - 2592000, 0 = no minimum, default = 0).
`script <string>`	The commands to be run by this CLI script action.
`output-size <integer>`	Set the size to limit the script output, in megabytes (1 - 1024, default = 10).
`timeout <integer>`	Set the maximum running time for this script, in seconds (0 - 300, 0 = no timeout, default = 0).
`execute-security-fabric {enable | disable}`	Enable/disable execution of CLI script on all or only one FortiGate unit in the Security Fabric. • `enable`: the action will run on all fabric devices. • `disable` (default): the action will run only on the device where the action was triggered.
`accprofile <profile>`	Access profile for CLI script action to access FortiGate features.

Certain diagnostic commands may not function as expected with CLI scripts and result in no output. For example, when used in a CLI script, the diagnostic command `dia test application dnsproxy 6` fails to produce any output because the cli-script feature does not support `daemon message()` prints.

Execute on Security Fabric (execute-security-fabric enable) can be used in a way where an event triggered on a device causes all devices to take the same action.

Some actions, such as `execute backup config ftp`, when taken on multiple devices could cause an overwrite on the FTP server when multiple devices run the same action at the same time.

Example

In this example, the script sets the idle timeout value to 479 minutes, and sends an email with the script output.

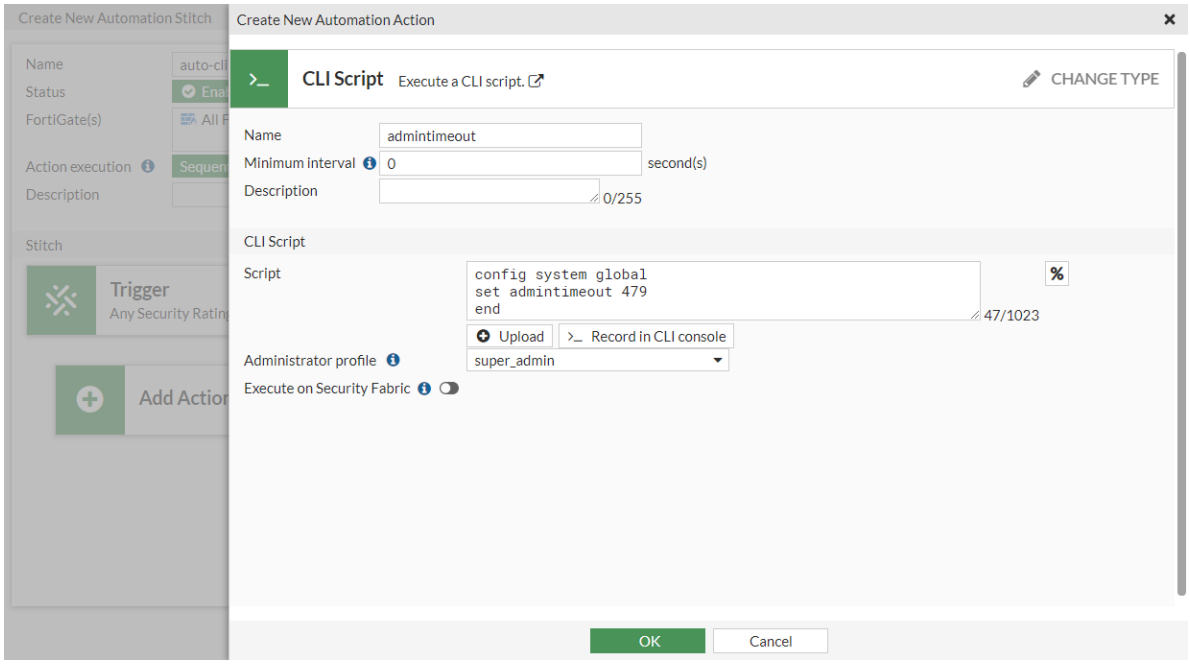
To configure a stitch with a CLI script action in the GUI:

1. Go to *Security Fabric* > *Automation* and click *Create New*.
2. Enter the stitch name (*auto-cli-1*).
3. Configure the trigger:
   a. Click *Add Trigger*.
   b. Select *Any Security Rating Notification*.
   c. Click *Apply*.
4. Configure the CLI Script action:
   a. Click *Add Action*.
   b. Click *Create* and select *CLI Script*.
   c. Enter the following:

Name	admintimeout

Script	`config system global`
	`    set admintimeout 479`
	`end`
	
	Alternatively, click *Upload* to upload a file, or click *>_Record in CLI console* and enter the CLI commands.

Administrator profile	Select a profile



d. Click *OK*.

e. Select the action in the list and click *Apply*.

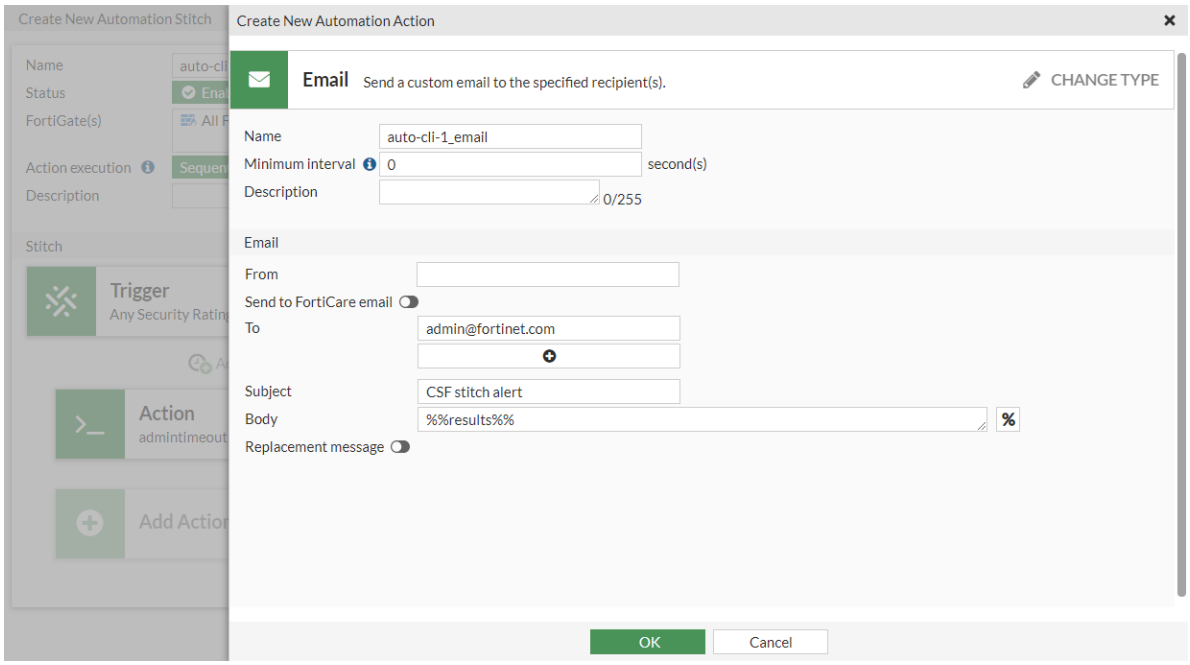
5. Configure the Email notification action:

a. Click *Add Action*.

b. Click *Create* and select *Email*.

c. Enter the following:

Name	auto-cli-1_email
To	Enter an email address
Subject	CSF stitch alert
Body	%%results%%



d. Click *OK*.

e. Select the action in the list and click *Apply*.

6. Click *OK*.

To configure a stitch with a CLI script action in the CLI:

1. Create the automation trigger:

```
config system automation-trigger
    edit "Any Security Rating Notification"
        set event-type security-rating-summary
        set report-type any
    next
end
```

2. Create the automation actions:

```
config system automation-action
    edit "admintimeout"
        set action-type cli-script
        set script "config system global
            set admintimeout 479
            end"
        set output-size 10
        set timeout 0
        set accprofile "super_admin"
    next
    edit "auto-cli-1_email"
        set action-type email
        set email-to "admin@fortinet.com"
        set email-subject "CSF stitch alert"
```

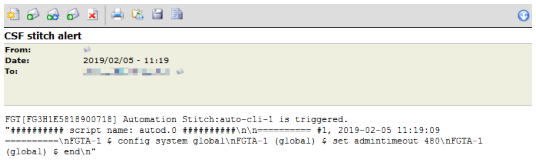
```
        set message "%%results%%"
    next
end
```

3. Create the automation stitch:

```
config system automation-stitch
    edit "auto-cli-1"
        set trigger "Any Security Rating Notification"
        config actions
            edit 1
                set action "admintimeout"
                set required enable
            next
            edit 2
                set action "auto-cli-1_email"
                set required enable
            next
        end
    next
end
```

Sample email

The email sent by the action will look similar to the following:



Execute a CLI script based on memory and CPU thresholds

Automation stitches can be created to run a CLI script and send an email message when memory or CPU usage exceeds specified thresholds.

CLI script action output that exceeds 64KB is sent as an attachment in the next email action.

To define memory and CPU usage thresholds:

```
config system global
    set cpu-use-threshold <percent>
    set memory-use-threshold-extreme <percent>
    set memory-use-threshold-green <percent>
    set memory-use-threshold-red <percent>
end
```

Where:

`cpu-use-threshold`	Threshold at which CPU usage is reported, in percent of total possible CPU utilization (default = 90).
`memory-use-threshold-extreme`	Threshold at which memory usage is considered extreme, and new sessions are dropped, in percent of total RAM (default = 95).
`memory-use-threshold-green`	Threshold at which memory usage forces the FortiGate to exit conserve mode, in percent of total RAM (default = 82).
`memory-use-threshold-red`	Threshold at which memory usage forces the FortiGate to enter conserve mode, in percent of total RAM (default = 88).

Configuring a high memory usage stitch

In this example, an automation stitch is created that runs a CLI script to collect debug information, and then email the results of the script to a specified email address when the memory usage causes the FortiGate to enter conserve mode.

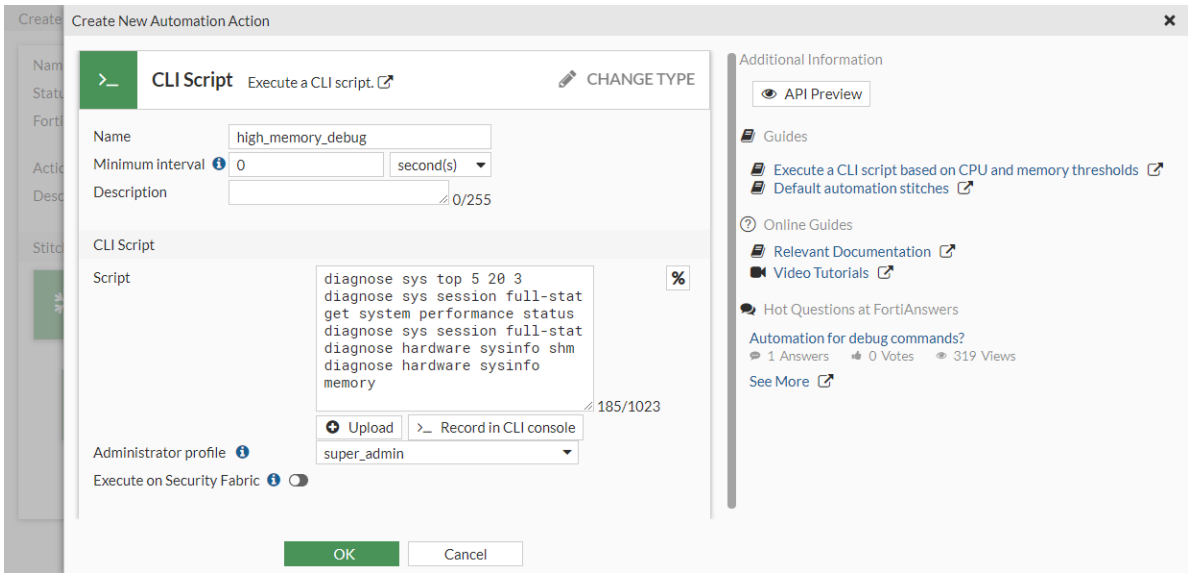
To create an automation stitch for high memory usage in the GUI:

1. Go to *Security Fabric > Automation* and click *Create New*.
2. Enter the stitch name (*auto_high_memory*).
3. Configure the trigger:
   a. Click *Add Trigger*.
   b. Select *Conserve Mode*.
   c. Click *Apply*.
4. Configure the CLI Script action:
   a. Click *Add Action*.
   b. Click *Create* and select *CLI Script*.
   c. Enter the following:

Name	high_memory_debug

Script	`diagnose sys top 5 20 3`
	`diagnose sys session full-stat`
	`get system performance status`
	`diagnose sys session full-stat`
	`diagnose hardware sysinfo shm`
	`diagnose hardware sysinfo memory`

Administrator profile	Select a profile



d. Click *OK*.

e. Select the action in the list and click *Apply*.

5. Configure the Email notification action:

a. Click *Add Action*.

b. Click *Create* and select *Email*.

c. Enter the following:

Name	auto_high_memory_email
To	Enter an email address
Subject	CSF stitch alert: high_memory
Body	%%results%%

d. Click *OK*.

e. Select the action in the list and click *Apply*.

6. Click *OK*.

To create an automation stitch for high memory usage in the CLI:

1. Create the automation trigger:

```
config system automation-trigger
    edit "Conserve Mode"
        set event-type low-memory
    next
end
```

2. Create the automation actions:

```
config system automation-action
    edit "high_memory_debug"
        set action-type cli-script
```

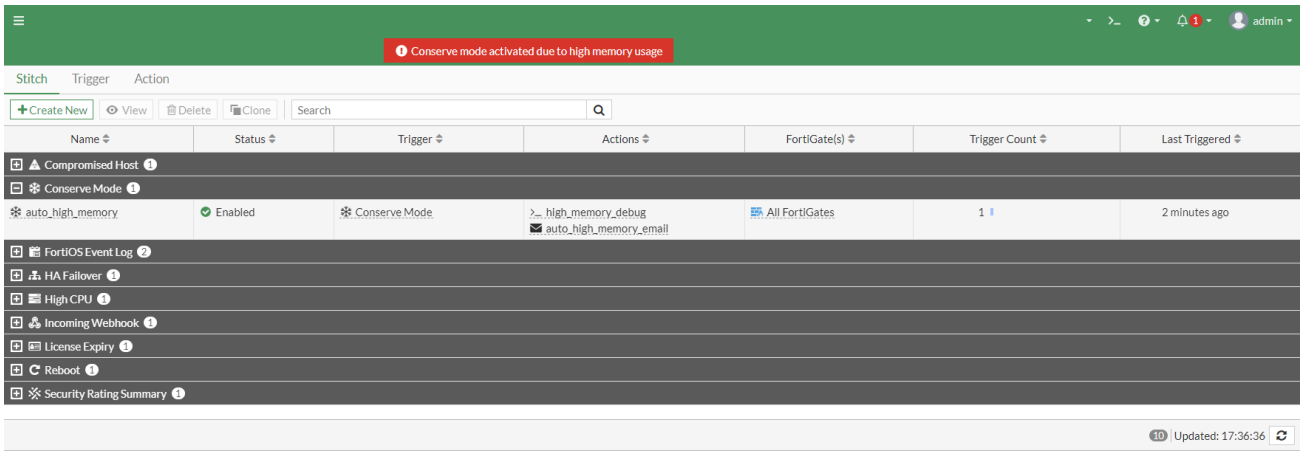
```
        set script "diagnose sys top 5 20 3
diagnose sys session full-stat
get system performance status
diagnose sys session full-stat
diagnose hardware sysinfo shm
diagnose hardware sysinfo memory"
        set output-size 10
        set timeout 0
        set accprofile "super_admin"
    next
    edit "auto_high_memory_email"
        set action-type email
        set email-to "person@fortinet.com"
        set email-subject "CSF stitch alert: high_memory"
        set message "%%results%%"
    next
end
```

3. Create the automation stitch:

```
config system automation-stitch
    edit "auto_high_memory"
        set trigger "Conserve Mode"
        config actions
            edit 1
                set action "high_memory_debug"
                set required enable
            next
            edit 2
                set action "auto_high_memory_email"
                set required enable
            next
        end
    next
end
```

Results

When the FortiGate enters conserve mode due to the `memory-use-threshold-red` being exceeded, the GUI displays a notice, and the *auto_high_memory* automation stitch is triggered. This causes the CLI script to run and the script results are emailed to the specified address.



Here is sample text from the email message:

```
CSF stitch alert: high_memory
DoNotReply@fortinet-notifications.com
Tue 05/16/2023 5:34 PM
########## script name: autod.0 ##########
========== #1, 2023-05-16 17:34:17 ==========
Client_Fgt $  diagnose sys top 5 20 3
Run Time:  0 days, 0 hours and 0 minutes 61U, 0N, 6S, 33I, 0WA, 0HI, 0SI, 0ST; 1356T, 129F
       ipshelper     2601      S <    61.6     8.0    0
       ipsengine     2745      S <     4.9     8.5    0
         cmdbsvr     2528      S N     0.0     7.9    0
         cmdbsvr     2529      S       0.0     5.0    0
       scanunitd     2610      S <     0.0     3.8    0
         miglogd     2603      S       0.0     3.6    0
          cw_acd     2634      S       0.0     3.4    0
            node     2574      S       0.0     3.3    0
       forticron     2584      S       0.0     2.9    0
         miglogd     2693      S       0.0     2.8    0
         reportd     2604      S       0.0     2.5    0
          httpsd     2573      S       0.0     2.4    0

...
```

Configuring a high CPU usage stitch

Similar to the previous example, an automation stitch can be created that runs a CLI script to collect debug information, and then email the results of the script to a specified email address when CPU usage threshold is exceeded (High CPU trigger).

The following commands are recommended for collecting debug information, but they are not the only options. Other commands can be used.

```
diagnose sys cmdb info
diagnose sys vd list | grep fib
diagnose sys top 5 20 2
diagnose sys session full-stat
diagnose sys session list | grep "\<dirty\>" -c
```

```
get system performance status
diagnose sys session full-stat
diagnose hardware sysinfo memory
diagnose sys cmdb info
diagnose sys vd list | grep fib
```

Webhook action

The webhook automation stitch action makes HTTP and HTTPS requests to a specified server, with custom headers, bodies, ports, and methods. It can be used to leverage the ubiquity of HTML requests and APIs to integrate with other tools.

The URI and HTTP body can use parameters from logs or previous action results. Wrapping the parameter with %% will replace the expression with the JSON value for the parameter, for example: *%%results.source%%* is the source property from the previous action.

In this example, a specific log message (failed administrator log in attempt) triggers the FortiGate to send the contents of the log to a server. The server responds with a generic reply. This example assumes that the server is already configured and able to communicate with the FortiGate.

To configure the webhook automation stitch in the GUI:

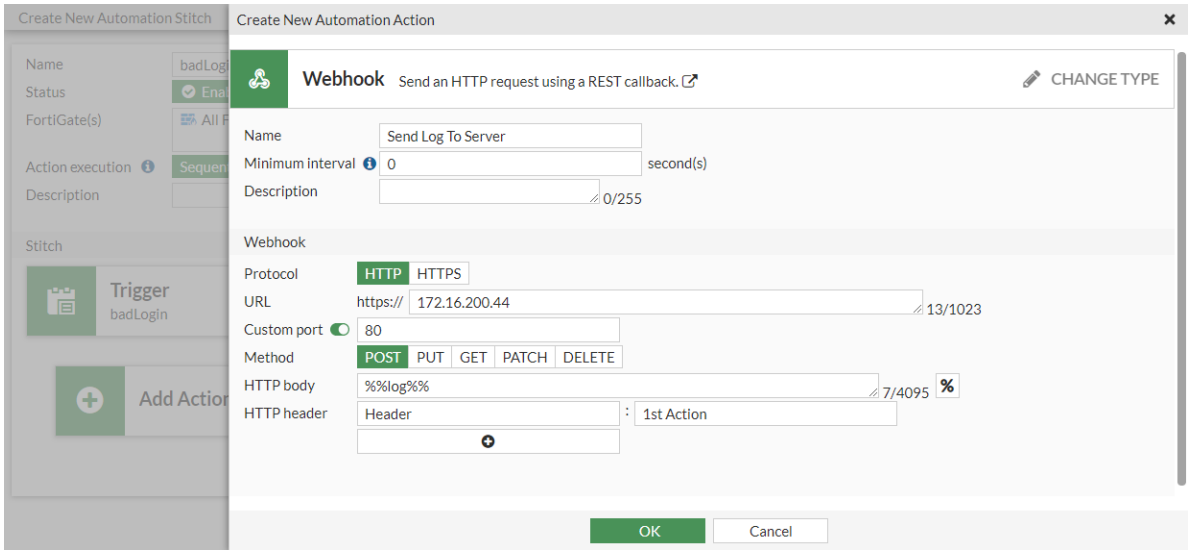
1. Go to *Security Fabric > Automation* and click *Create New*.
2. Enter the stitch name (*badLogin*).
3. Configure the trigger:
   a. Click *Add Trigger*.
   b. Click *Create* and select *FortiOS Event Log*.
   c. Enter the following:

Name	badLogin
Event	Admin login failed

   d. Click *OK*.
   e. Select the trigger in the list and click *Apply*.
4. Configure the automation stitch action:
   a. Click *Add Action*.
   b. Click *Create* and select *Webhook*.
   c. Enter the following:

Name	Send Log To Server
Protocol	HTTP
URL	172.16.200.44
Custom port	Enable and enter 80
Method	POST

HTTP body	%%log%%
HTTP header	Header : 1st Action



d. Click *OK*.

e. Select the action in the list and click *Apply*.

5. Click *OK*.

To configure the webhook automation stitch in the CLI:

1. Create an automation trigger:

```
config system automation-trigger
    edit "badLogin"
        set event-type event-log
        set logid 32002
    next
end
```

2. Create the automation action:

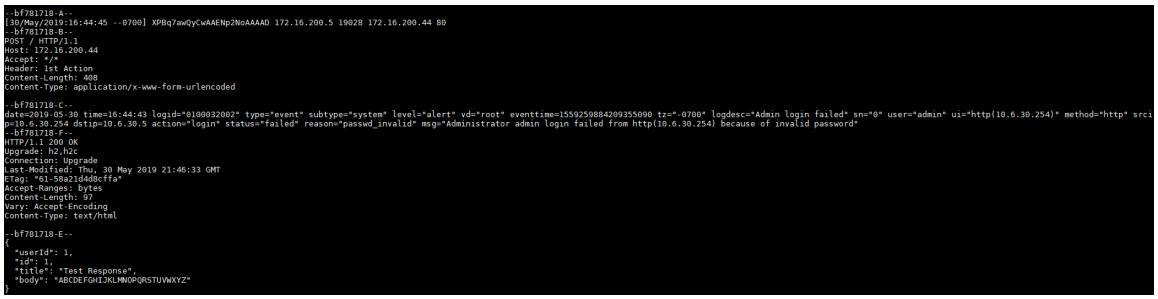
```
config system automation-action
    edit "Send Log To Server"
        set action-type webhook
        set uri "172.16.200.44"
        set http-body "%%log%%"
        set port 80
        config http-headers
            edit 1
                set key "Header"
                set value "1st Action"
            next
        end
    next
end
```

3. Create the automation stitch:

```
config system automation-stitch
    edit "badLogin"
        set trigger "badLogin"
        config actions
            edit 1
                set action "Send Log To Server"
                set required enable
            next
        end
    next
end
```

To test the automation stitch:

1. Attempt to log in to the FortiGate with an incorrect username or password.
2. On the server, check the log to see that its contents were sent by the FortiGate.



   The body content is replaced with the log from the trigger.
3. On the FortiGate, go to *Log & Report* > *System Events* to confirm that the stitch was activated.
4. Go to *Security Fabric* > *Automation* to see the last time that the stitch was triggered.

Diagnose commands

To enable log dumping:

```
# diagnose test application autod 1
autod dumped total:1 logs, num of logids:1
autod log dumping is enabled

vdom:root(0) logid:32002 len:408 log:
date=2019-05-30 time=17:41:03 logid="0100032002" type="event" subtype="system" level="alert"
vd="root" eventtime=1559263263858888451 tz="-0700" logdesc="Admin login failed" sn="0"
user="admin" ui="http(10.6.30.254)" method="http" srcip=10.6.30.254 dstip=10.6.30.5 action="login"
status="failed" reason="passwd_invalid" msg="Administrator admin login failed from http
(10.6.30.254) because of invalid password"
autod log dumping is disabled

autod logs dumping summary:
        logid:32002 count:1
```

```
autod dumped total:1 logs, num of logids:1
```

To show the automation settings:

```
# diagnose test application autod 2
csf: enabled   root:yes
total stitches activated: 2

stitch: badLogin
       destinations: all
       trigger: badLogin

       local hit: 6 relayed to: 6 relayed from: 6
       actions:
               Send Log To Server type:webhook interval:0
                       delay:0 required:no
                       proto:0 method:0 port:80
                       uri: 172.16.200.44
                       http body: %%log%%
                       headers:
                       0. Header:1st Action
```

To show the automation statistics:

```
# diagnose test application autod 3

stitch: badLogin

       local hit: 1 relayed to: 1 relayed from: 1
       last trigger:Wed Jul 10 12:14:14 2019
       last relay:Wed Jul 10 12:14:14 2019

       actions:
               Send Log To Server:
                       done: 1 relayed to: 1 relayed from: 1
                       last trigger:Wed Jul 10 12:14:14 2019
                       last relay:Wed Jul 10 12:14:14 2019

logid2stitch mapping:
id:32002  local hit: 3 relayed to: 3 relayed from: 3
       badLogin

action run cfg&stats:
total:55 cur:0 done:55 drop:0
       email:
               flags:10
               stats: total:4 cur:0 done:4 drop:0
       fortiexplorer-notification:
               flags:1
               stats: total:0 cur:0 done:0 drop:0
```

```
        alert:
                flags:0
                stats: total:0 cur:0 done:0 drop:0
        disable-ssid:
                flags:7
                stats: total:0 cur:0 done:0 drop:0
        quarantine:
                flags:7
                stats: total:0 cur:0 done:0 drop:0
        quarantine-forticlient:
                flags:4
                stats: total:0 cur:0 done:0 drop:0
        quarantine-nsx:
                flags:4
                stats: total:0 cur:0 done:0 drop:0
        ban-ip:
                flags:7
                stats: total:0 cur:0 done:0 drop:0
        aws-lambda:
                flags:11
                stats: total:21 cur:0 done:21 drop:0
        webhook:
                flags:11
                stats: total:6 cur:0 done:6 drop:0
        cli-script:
                flags:10
                stats: total:4 cur:0 done:4 drop:0
        azure-function:
                flags:11
                stats: total:0 cur:0 done:0 drop:0
        google-cloud-function:
                flags:11
                stats: total:0 cur:0 done:0 drop:0
        alicloud-function:
                flags:11
                stats: total:20 cur:0 done:20 drop:0
```

To enable debug output and turn on automation debug messages for about 30 minutes:

```
# diagnose debug enable
# diagnose debug application autod -1
__auto_generate_generic_curl_request()-358: Generating generic automation CURL request for action
(Send Log To Server).
__auto_generate_generic_curl_request()-406: Generic automation CURL request POST data for action
(Send Log To Server):
date=2019-05-30 time=16:44:43 logid="0100032002" type="event" subtype="system" level="alert"
vd="root" eventtime=1559259884209355090 tz="-0700" logdesc="Admin login failed" sn="0"
user="admin" ui="http(10.6.30.254)" method="http" srcip=10.6.30.254 dstip=10.6.30.5 action="login"
status="failed" reason="passwd_invalid" msg="Administrator admin login failed from http
(10.6.30.254) because of invalid password"

__auto_generic_curl_request_close()-512: Generic CURL request response body from
```

```
http://172.16.200.44:
{
  "userId": 1,
  "id": 1,
  "title": "Test Response",
  "body": "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
}
```

Webhook action with Twilio for SMS text messages

The FortiGate automation stitch framework can be used to interact with third-party services with webhooks to perform various tasks. A webhook action to Twilio can be used to automate tasks that send alerts and information to administrators by SMS text messages.

This topic includes two examples where the FortiGate uses webhooks to trigger Twilio to send SMS text messages to an administrator.

- Example 1: using an SD-WAN health check to trigger a Twilio webhook action
- Example 2: using an incoming webhook to trigger a Twilio webhook action

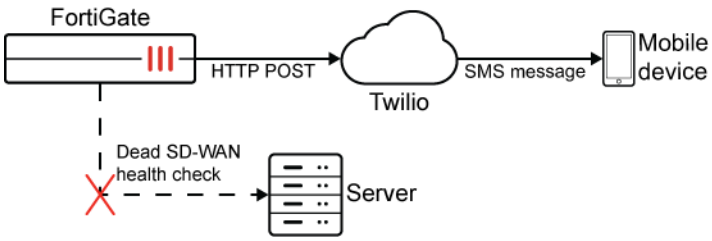
Prerequisites

- An active Twilio account with a virtual phone number that is able to send SMS text messages to the receiver in the desired region
- A valid Twilio Account SID and Auth token for sending SMS text messages

> These examples assume that the Twilio account is already configured. For more information, see Creating or Updating Resources with the POST Method in the Twilio documentation.

Example 1: using an SD-WAN health check to trigger a Twilio webhook action

In this example, an administrator wants to monitor the FortiGate's SD-WAN health, particularly when a dead health check is reported. An automation stitch will trigger the alert based on an SD-WAN log event (log ID 0113022931, SD-WAN SLA information warning) and perform a webhook action to inform Twilio to send an SMS message.



Sample SD-WAN log event

```
date=2023-01-13 time=10:39:26 eventtime=1673635167489361827 tz="-0800" logid="0113022931"
type="event" subtype="sdwan" level="warning" vd="root" logdesc="SDWAN SLA information warning"
```

```
eventtype="Health Check" healthcheck="Google" interface="VLAN1101" probeproto="ping"
oldvalue="alive" newvalue="dead" msg="SD-WAN health-check member changed state."
```

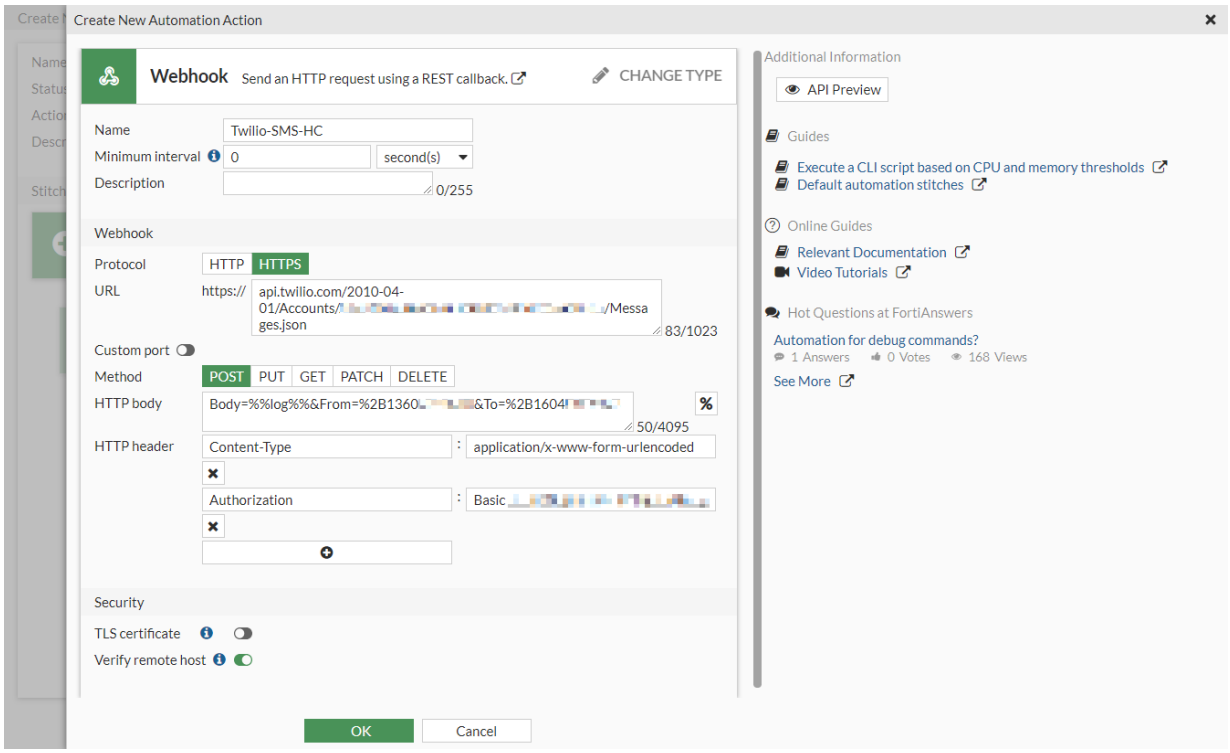
To configure the automation stitch in the GUI:

1. Go to *Security Fabric > Automation* and click *Create New*.
2. Enter the stitch name (*SD-WAN-HC-Down-SMS*).
3. Configure the FortiOS event log trigger:
   a. Click *Add Trigger*.
   b. Click *Create* and select *FortiOS Event Log*.
   c. Enter the following:

Name	*SD-WAN-HC-Down*
Event	*SDWAN SLA information warning*
Field filter(s)	Set *Field name* to *newvalue*. Set *Value* to *dead*.

   d. Click *OK*.
   e. Select the trigger in the list and click *Apply*.
4. Configure the webhook action:
   a. Click *Add Action*.
   b. Click *Create* and select *Webhook*.
   c. Enter the following:

Name	*Twilio-SMS-HC*
Protocol	*HTTPS*
URL	Enter the URL provided by Twilio for sending SMS messages using the POST method. The URL can be found in the cURL code sample in the following format: *https://api.twilio.com/2010-04-01/Accounts/<Twilio _Account_SID>/Messages.json*.
Method	*POST*
HTTP body	*Body=%%log%%&From=%2B1360x*******&To=%2B1604******* This string for the body text includes the SD-WAN log message, and the Twilio from and to phone numbers.
HTTP header	*Content-Type : application/x-www-form-urlencoded* *Authorization : Basic <Base64_encoded_authentication_code>*



d. Click *OK*.

e. Select the action in the list and click *Apply*.

5. Click *OK* to save the stitch.

To configure the automation stitch in the CLI:

1. Configure the automation trigger:

```
config system automation-trigger
    edit "SD-WAN-HC-Down"
        set event-type event-log
        set logid 22931
        config fields
            edit 1
                set name "newvalue"
                set value "dead"
            next
        end
    next
end
```

2. Configure the automation action:

```
config system automation-action
    edit "Twilio-SMS-HC"
        set action-type webhook
        set protocol https
        set uri "api.twilio.com/2010-04-
```

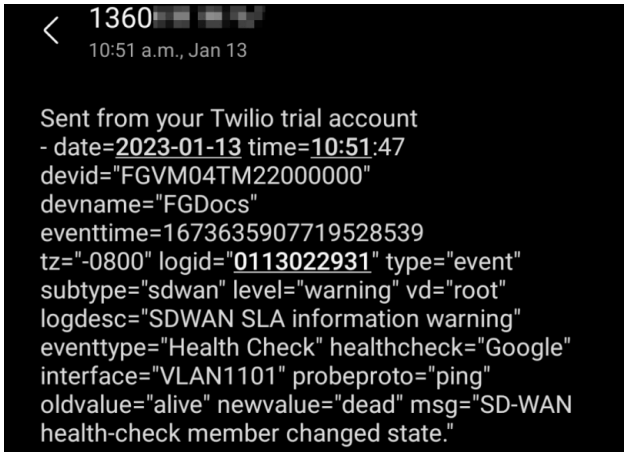
```
01/Accounts/********************************/Messages.json"
        set http-body "Body=%%log%%&From=%2B1360*******&To=%2B1604*******"
        set port 443
        config http-headers
            edit 1
                set key "Content-Type"
                set value "application/x-www-form-urlencoded"
            next
            edit 2
                set key "Authorization"
                set value "Basic ************"
            next
        end
    next
end
```

3. Configure the automation stitch:

```
config system automation-stitch
    edit "SD-WAN-HC-Down-SMS"
        set trigger "SD-WAN-HC-Down"
        config actions
            edit 1
                set action "Twilio-SMS-HC"
                set required enable
            next
        end
    next
end
```

Verification:

1. Simulate an SD-WAN health check failure to trigger the automation stitch.
2. Twilio sends an SMS message to the administrator.



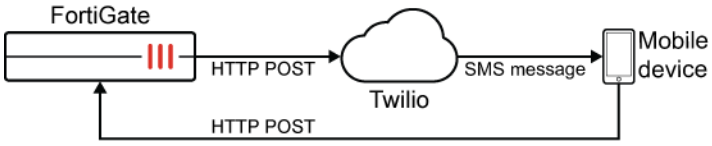
3. Go to *Security Fabric > Automation* and locate the *SD-WAN-HC-Down-SMS* stitch. The *Trigger Count* value has increased by one.

System log after the stitch was triggered:

```
date=2023-01-13 time=10:51:47 eventtime=1673635907720476287 tz="-0800" logid="0100046600"
type="event" subtype="system" level="notice" vd="root" logdesc="Automation stitch triggered"
stitch="SD-WAN-HC-Down-SMS" trigger="SD-WAN-HC-Down" stitchaction="Twilio-SMS-HC" from="log"
msg="stitch:SD-WAN-HC-Down-SMS is triggered."
```

Example 2: using an incoming webhook to trigger a Twilio webhook action

In this example, an administrator wants to trigger an automation stitch remotely to retrieve the device uptime with an SMS text message. An incoming webhook will be used to send an HTTP POST request to trigger the event to occur. Subsequently, the FortiGate performs a CLI action followed by a webhook action to inform Twilio to send an SMS text message.



A REST API administrator with write privileges must be configured to apply an API key to this incoming webhook. See REST API administrator on page 3712 for more information.

To configure the automation stitch in the GUI:

1. Go to *Security Fabric > Automation* and click *Create New*.
2. Enter the stitch name (*Check-Uptime*).
3. Configure the incoming webhook trigger:
   a. Click *Add Trigger*.
   b. Click *Create* and select *Incoming Webhook*.
   c. Enter the trigger name (*Check-Uptime-Webhook*).
   d. Click *OK*.
   e. Select the trigger in the list and click *Apply*.
4. Configure the CLI script action:
   a. Click *Add Action*.
   b. Click *Create* and select *CLI Script*.
   c. Enter the following:

Name	*Uptime*
Script	`get system performance status | grep Uptime`
Administrator profile	Select a profile with REST API write privileges (*prof_admin*)

   d. Click *OK*.
   e. Select the action in the list and click *Apply*.
5. Configure the webhook action:

a. Click *Add Action*.
b. Click *Create* and select *Webhook*.
c. Enter the following:

Name	*Twilio-SMS-Uptime*
Protocol	*HTTPS*
URL	Enter the URL provided by Twilio for sending SMS messages using the POST method. The URL can be found in the cURL code sample in the following format: *https://api.twilio.com/2010-04-01/Accounts/<Twilio _ Account_SID>/Messages.json*.
Method	*POST*
HTTP body	*Body=%%results%%&From=%2B1360x*******&To=%2B1604******* This string for the body text includes the results from the preceding CLI script action.
HTTP header	*Content-Type : application/x-www-form-urlencoded* *Authorization : Basic <Base64_encoded_authentication_code>*

d. Click *OK*.
e. Select the action in the list and click *Apply*.

6. Click *OK* to save the stitch.

To configure the automation stitch in the CLI:

1. Configure the automation trigger:

```
config system automation-trigger
    edit "Check-Uptime-Webhook"
        set event-type incoming-webhook
    next
end
```

2. Configure the automation actions:

```
config system automation-action
    edit "Uptime"
        set action-type cli-script
        set script "get system performance status | grep Uptime"
        set accprofile "prof_admin"
    next
    edit "Twilio-SMS-Uptime"
        set action-type webhook
        set protocol https
        set uri "api.twilio.com/2010-04-
01/Accounts/********************************/Messages.json"
        set http-body "Body=%%results%%&From=%2B1360*******&To=%2B1604*******"
        set port 443
        config http-headers
            edit 1
```

```
                set key "Content-Type"
                set value "application/x-www-form-urlencoded"
            next
            edit 2
                set key "Authorization"
                set value "Basic ************"
            next
        end
    next
end
```

3. Configure the automation stitch:

```
config system automation-stitch
    edit "Check-Uptime"
        set trigger "Check-Uptime-Webhook"
        config actions
            edit 1
                set action "Uptime"
                set required enable
            next
            edit 2
                set action "Twilio-SMS-Uptime"
                set required enable
            next
        end
    next
end
```

Verification:

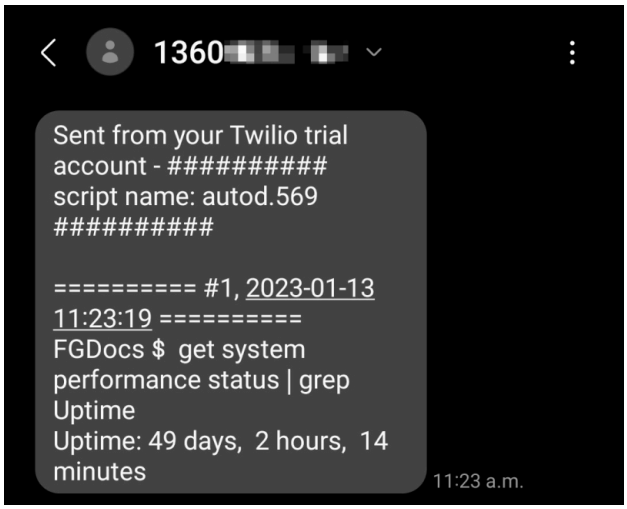
1. From a command prompt, issue the sample cURL command as recommended by the Check-Uptime-Webhook incoming webhook:

```
>curl -k -X POST -H "Authorization: Bearer <API_token>" --data "{ 'srcip': '1.1.1.1',
'mac':'11:11:11:11:11:11', 'fctuid': '********************************'}"
https://x.x.x.x/api/v2/monitor/system/automation-stitch/webhook/Check-Uptime-Webhook
```

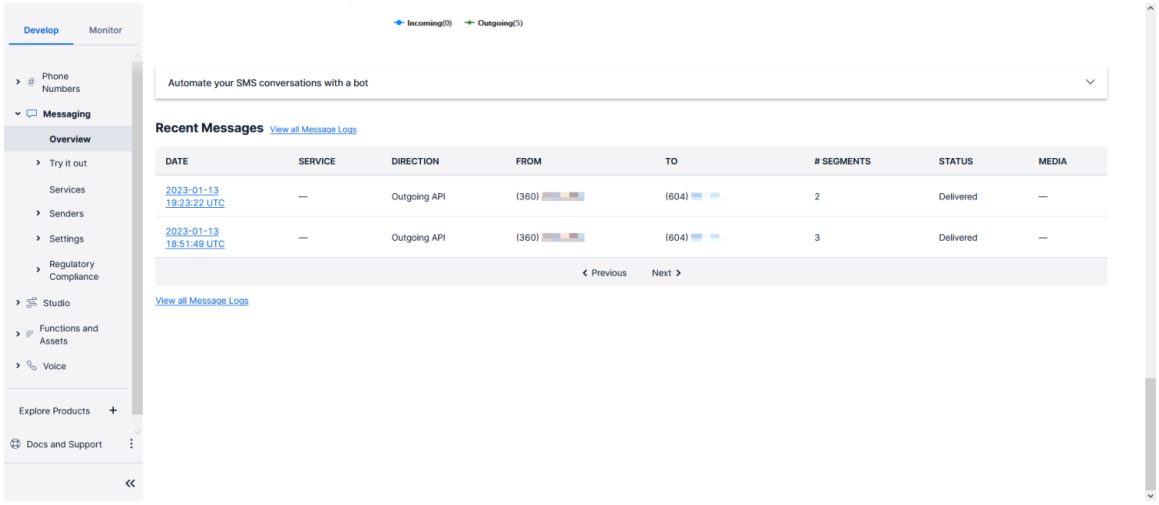
   Response:

```
{
  "http_method":"POST",
  "status":"success",
  "http_status":200,
  "serial":"FGVM04TM20000000",
  "version":"v7.2.4",
  "build":1368
}
```

2. This triggers the automation stitch on the FortiGate to send a HTTP POST request to Twilio. In response, Twilio sends an SMS text message to the recipient.



3. From the Twilio dashboard, go to *Messaging* > *Overview* and verify the *Recent Messages* section to confirm that the SMS text message was delivered.



To view automation stitch diagnostics:

```
# diagnose debug enable
# diagnose debug application autod -1
```

```
2023-01-13 11:23:19 __action_cli_script_open()-180: cli script action:Uptime is called. svc
ctx:0x10b4d5c0
accprof:prof_admin script:
get system performance status | grep Uptime

2023-01-13 11:23:19 0: get system performance status | grep Uptime
2023-01-13 11:23:20 __cli_script_close()-115: cli script:
autod.569
 output:
########## script name: autod.569 ##########

========== #1, 2023-01-13 11:23:19 ==========
```

```
FGDocs $  get system performance status | grep Uptime
Uptime: 49 days,  2 hours,  14 minutes

2023-01-13 11:23:20 0: config system auto-script
2023-01-13 11:23:20 0: delete "autod.569"
2023-01-13 11:23:20 0: end
2023-01-13 11:23:20 __action_cli_script_close()-207: cli script action is done. script:
get system performance status | grep Uptime
output:
########## script name: autod.569 ##########

========== #1, 2023-01-13 11:23:19 ==========
FGDocs $  get system performance status | grep Uptime
Uptime: 49 days,  2 hours,  14 minutes

2023-01-13 11:23:20 auto_generate_generic_curl_request()-443: Generating generic automation CURL
request for action (Twilio-SMS-Uptime).
2023-01-13 11:23:20 auto_generate_generic_curl_request()-462: Formatting HTTP body with action
parameters.
2023-01-13 11:23:20 auto_generate_generic_curl_request()-495: Generic automation CURL request POST
data for action (Twilio-SMS-Uptime):
Body=########## script name: autod.569 ##########

========== #1, 2023-01-13 11:23:19 ==========
FGDocs $  get system performance status | grep Uptime
Uptime: 49 days,  2 hours,  14 minutes
&From=%2B1360*******&To=%2B1604*******

2023-01-13 11:23:20 auto_generate_generic_curl_request()-550: Generic automation CURL request Host
header: api.twilio.com
2023-01-13 11:23:20 auto_generate_generic_curl_request()-553: Adding 2 user defined headers
2023-01-13 11:23:23 auto_curl_perform()-114: Failed to send curl request. http status code: 201
```

Although the final line in this debug output shows `Failed to send curl request. http status code: 201`, the HTTP status code 201 indicates that the request was successful and a response code was created.

Slack integration webhook

A webhook can be created to post messages and notifications to Slack.

In this example, a configuration change triggers the FortiGate to post a message to Slack.

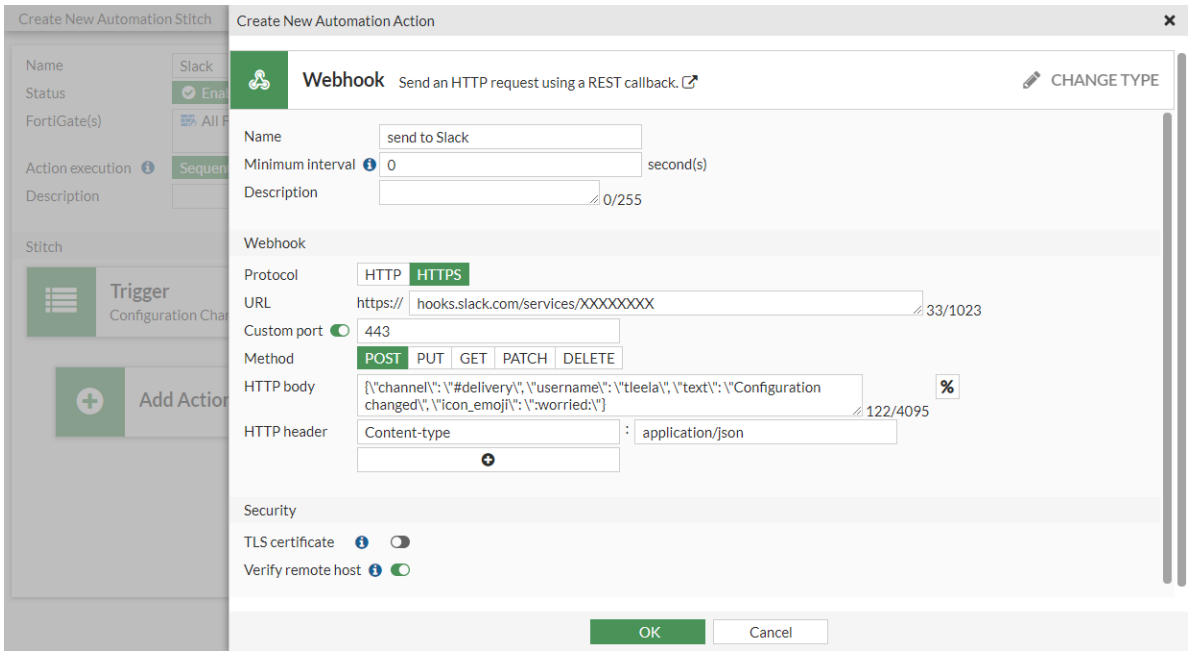
To create a webhook automation stitch for Slack integration in the GUI:

1. Create an incoming webhook in Slack. See Sending messages using Incoming Webhooks for more information.
2. Go to *Security Fabric > Automation* and click *Create New*.
3. Enter the stitch name.
4. Configure the trigger:

a. Click *Add Trigger*.
b. Select *Configuration Change*.
c. Click *Apply*.

5. Configure the action:
   a. Click *Add Action*.
   b. Click *Create* and select *Webhook*.
   c. Enter the following:

Name	send to Slack
Protocol	HTTPS
URL	Enter the incoming webhook URL created in Slack
Custom port	Enable and enter 443
Method	POST
HTTP body	{\"channel\": \"#delivery\", \"username\": \"tleela\", \"text\": \"Configuration changed\", \"icon_emoji\": \":worried:\"}
HTTP header	Content-type : application/json



   d. Click *OK*.
   e. Select the action in the list and click *Apply*.
6. Click *OK*.

To create a webhook automation stitch for Slack integration in the CLI:

1. Create an incoming webhook in Slack. See Sending messages using Incoming Webhooks for more information.
2. Create the automation trigger:

```
config system automation-trigger
    edit "Configuration Change"
        set event-type config-change
    next
end
```

3. Create the automation action:

```
config system automation-action
    edit "send to Slack"
        set action-type webhook
        set protocol https
        set uri "hooks.slack.com/services/XXXXXXXX"
        set http-body "{\"channel\": \"#delivery\", \"username\": \"tleela\", \"text\":
\"Configuration changed\", \"icon_emoji\": \":worried:\"}"
        set port 443
        config http-headers
            edit 1
                set key "Content-type"
                set value "application/json"
            next
        end
    next
end
```

4. Create the automation stitch:

```
config system automation-stitch
    edit "Slack"
        set trigger "Configuration Change"
        config actions
            edit 1
                set action "send to Slack"
                set required enable
            next
        end
    next
end
```

Microsoft Teams integration webhook

A webhook can be created to post messages and notifications to Microsoft Teams.

In this example, a configuration change triggers the FortiGate to post a message to Teams.

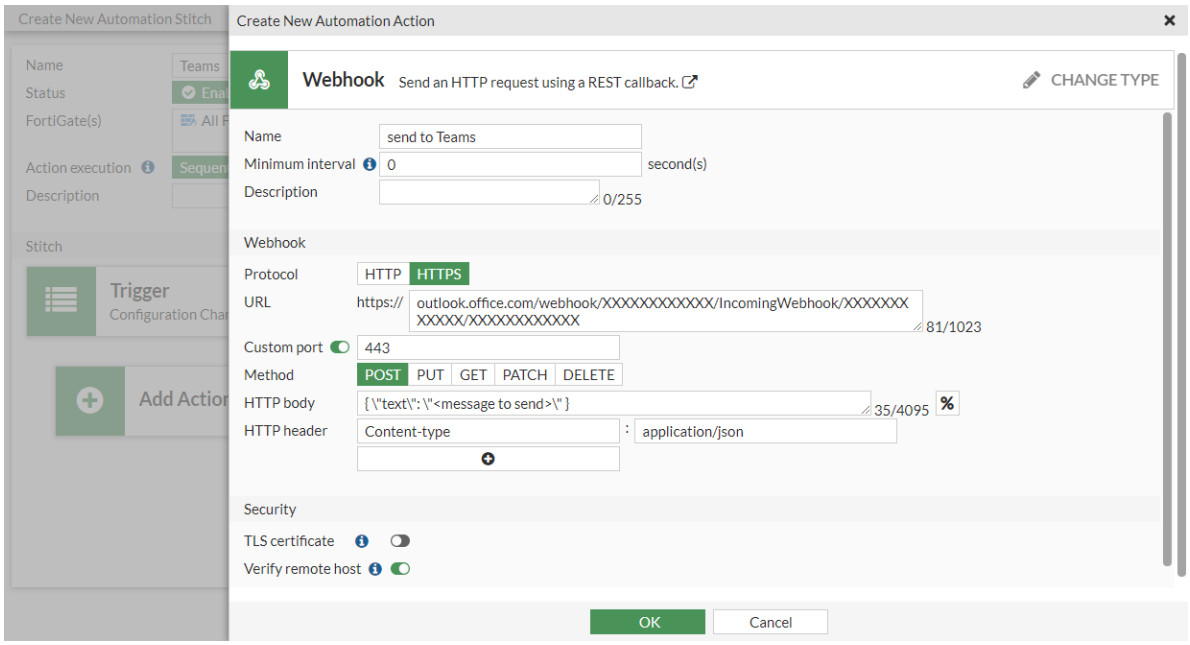
To create a webhook automation stitch for Teams integration in the GUI:

1. Create an incoming webhook in Teams. See Create an incoming webhook for information.
2. Go to *Security Fabric > Automation* and click *Create New*.
3. Enter the stitch name.
4. Configure the trigger:

a. Click *Add Trigger*.
b. Select *Configuration Change*.
c. Click *Apply*.

5. Configure the action:
   a. Click *Add Action*.
   b. Click *Create* and select *Webhook*.
   c. Enter the following:

Name	send to Teams
Protocol	HTTPS
URL	Enter the incoming webhook URL created in Teams
Custom port	Enable and enter 443
Method	POST
HTTP body	{ \"text\": \"<message to send>\" }
HTTP header	Content-type : application/json



   d. Click *OK*.
   e. Select the action in the list and click *Apply*.
6. Click *OK*.

To create a webhook automation stitch for Teams integration in the CLI:

1. Create an incoming webhook in Teams. See Create an incoming webhook for information.
2. Create the automation trigger:

```
config system automation-trigger
    edit "Configuration Change"
        set event-type config-change
    next
end
```

3. Create the automation action:

```
config system automation-action
    edit "send to Teams"
        set action-type webhook
        set protocol https
        set uri
"outlook.office.com/webhook/XXXXXXXXXXX/IncomingWebhook/XXXXXXXXXXX/XXXXXXXXXXX"
        set http-body "{ \"text\": \"<message to send>\" }"
        set port 443
        config http-headers
            edit 1
                set key "Content-type"
                set value "application/json"
            next
        end
    next
end
```

4. Create the automation stitch:

```
config system automation-stitch
    edit "Teams"
        set trigger "Configuration Change"
        config actions
            edit 1
                set action "send to Teams"
                set required enable
            next
        end
    next
end
```

For information about more advanced messages that can be configured and sent to the Teams incoming webhook, see Sending messages to connectors and webhooks.

System actions

The system actions can be used to back up the configuration of the FortiGate, reboot the FortiGate, or shut down the FortiGate.

These actions can occur even if the FortiGate is in conserve mode, and allows the automation stitch to bypass the CLI user confirmation prompts, which the CLI script action does not support.

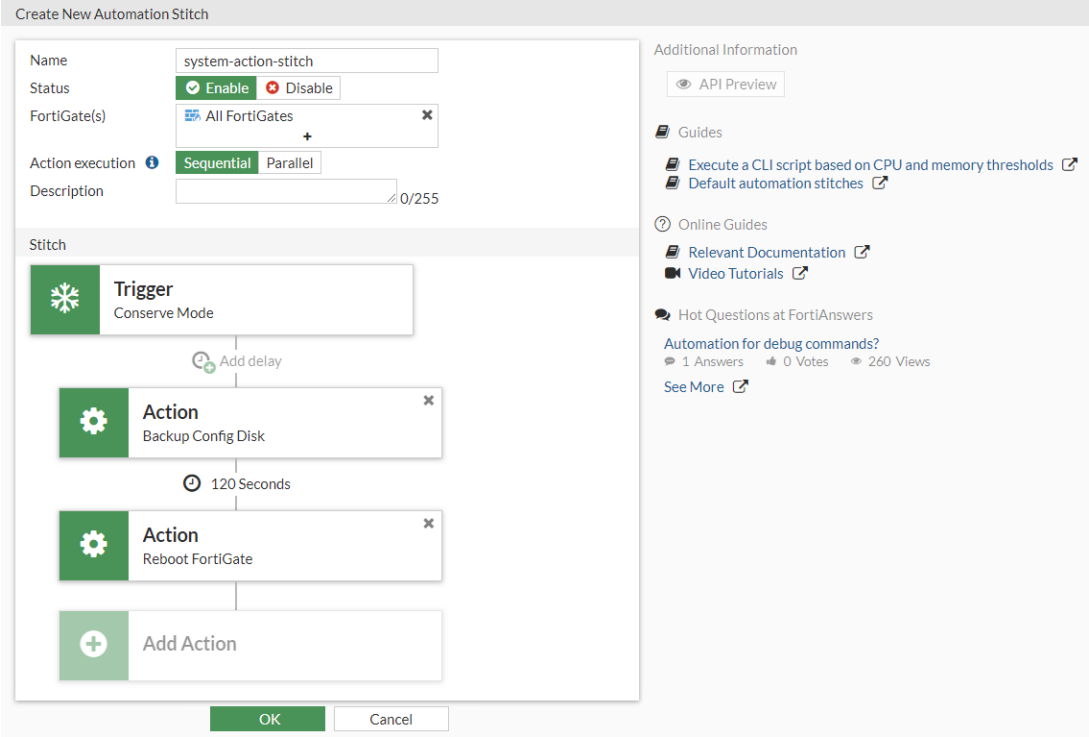
```
config system automation-action
    edit "Backup Config Disk"
        set action-type system-actions
        set system-action backup-config
    next
    edit "Reboot FortiGate"
        set action-type system-actions
        set system-action reboot
    next
    edit "Shutdown FortiGate"
        set action-type system-actions
        set system-action shutdown
    next
end
```

Example

In this example, an automation stitch is created that uses the Conserve Mode trigger, a Backup Config Disk action to back up the configuration to the FortiGate's disk (see Configuration backups and reset on page 4053 for more details), and then a Reboot FortiGate action. There is a 120-second delay between the two actions.

To configure an automation stitch with system actions in the GUI:

1. Go to *Security Fabric > Automation*, select the *Stitch* tab, and click *Create New*.
2. Enter the name, *system-action-stitch*.
3. Click *Add Trigger*. Select *Conserve Mode* and click *Apply*.
4. Click *Add Action*. Select *Backup Config Disk* and click *Apply*.
5. Click *Add Action*. Select *Reboot FortiGate* and click *Apply*.
6. Click *Add delay* (between the actions). Enter *120* and click *OK*.



7. Click *OK*.

To configure an automation stitch with system actions in the CLI:

1. Configure the trigger:

```
config system automation-trigger
    edit "Conserve Mode"
        set event-type low-memory
    next
end
```

2. Configure the back up and reboot actions:

```
config system automation-action
    edit "Backup Config Disk"
        set description "Backup the configuration on disk."
        set action-type system-actions
        set system-action backup-config
    next
    edit "Reboot FortiGate"
        set description "Reboot this FortiGate unit."
        set action-type system-actions
        set system-action reboot
        set minimum-interval 300
    next
end
```

3. Configure the stitch:

```
config system automation-stitch
    edit "system-action-stitch"
        set trigger "Conserve Mode"
        config actions
            edit 1
                set action "Backup Config Disk"
                set required enable
            next
            edit 2
                set action "Reboot FortiGate"
                set delay 120
                set required enable
            next
        end
    next
end
```

Verification

When the FortiGate enters conserve mode due to low memory, the automation stitch will be triggered and it will back up the configuration to the FortiGate disk, then reboot the FortiGate.

To confirm that the stitch was triggered in the GUI:

1. Go to *Security Fabric > Automation* and select the *Stitch* tab.
2. Verify the *Last Triggered* column.

To confirm that the stitch was triggered in the CLI:

```
# diagnose test application autod 3
alert mail log count: 0

stitch: system-action-stitch

    local hit: 1 relayed to: 0 relayed from: 0
    last trigger:Thu Jun 23 11:31:25 2022
    last relay:
    actions:
        Backup Config Disk:
            done: 1 relayed to: 0 relayed from: 0
            last trigger:Thu Jun 23 11:31:25 2022
            last relay:
        Reboot FortiGate:
            done: 0 relayed to: 0 relayed from: 0
            last trigger:Thu Jun 23 11:31:25 2022
            last relay:

logid to stitch mapping:
id:22011  local hit: 1 relayed hits: 0
    system-action-stitch
```

```
log category to stitch mapping:
```

To locate the backed up configuration in the GUI:

1. Click on the user name in the upper right-hand corner of the screen and select *Configuration* > *Revisions*.
2. Click the + in the table to expand and view more details.

To locate the backed up configuration in the CLI:

```
# execute revision list config
Last Firmware Version: V0.0.0-build000-REL0
1   2022-04-01 09:27:26    daemon_admin      V7.2.0-build1157-REL0      Automatic backup (upgrade)
2   2022-06-20 13:41:02    daemon_admin      V7.2.1-build1254-REL0      Automatic backup (upgrade)
3   2022-06-23 11:31:25    daemon_admin      V7.2.1-build1254-REL0      Autod backup config by
stitch: system-action-stitch
```

Public and private SDN connectors

Cloud SDN connectors provide integration and orchestration of Fortinet products with public and private cloud solutions. In a typical cloud environment, resources are dynamic and often provisioned and scaled on-demand. By using an SDN connector, you can ensure that changes to cloud environment attributes are automatically updated in the Security Fabric.

To protect the East-West or North-South traffic in these environments, the FortiGate uses the SDN connector to sync the dynamic addresses that these volatile environments use. You can then configure the dynamic address objects as sources or destinations for firewall policies. When you make changes to cloud environment resources, such as moving them to a new location or assigning different IP addresses to them, you do not need to modify the policy in FortiOS, as the SDN connector syncs changes to the cloud address objects.

These configurations consist of three primary steps:

1. Configure the cloud SDN connector to connect your FortiGate and public or private cloud account.
2. Create dynamic address objects to use the SDN connector. Use filters to sync only cloud address objects that you require.
3. Apply the dynamic address objects to your firewall policy to protect your traffic.

This chapter explores the steps in detail and describes how to connect to each currently supported cloud platform. This chapter does not discuss cloud account role-based or permission requirements. The respective cloud documents contain this information.

The following external connectors are available in the Security Fabric:

Category	Connector	Example configuration
Public SDN		

Category	Connector	Example configuration
	Amazon Web Services (AWS)	AWS SDN connector using access keys on page 4606
	Microsoft Azure	Azure SDN connector using service principal on page 4611
	Google Cloud Platform (GCP)	GCP SDN connector using service account on page 4625
	Oracle Cloud Infrastructure (OCI)	OCI SDN connector using certificates on page 4656
	IBM Cloud	IBM Cloud SDN connector using API keys on page 4627
	AliCloud	AliCloud SDN connector using access key on page 4603
Private SDN		
	Kubernetes	Kubernetes (K8s) SDN connectors on page 4630
	VMware ESXi	VMware ESXi SDN connector using server credentials on page 4665
	VMware NSX	VMware NSX-T Manager SDN connector using NSX-T Manager credentials on page 4667
	OpenStack (Horizon)	OpenStack SDN connector using node credentials on page 4659
	Application Centric Infrastructure (ACI)	Cisco ACI SDN connector using a standalone connector on page 4614
	Nuage Virtualized Services Platform	Nuage SDN connector using server credentials on page 4651
	Nutanix	Nutanix SDN connector using server credentials on page 4654
	SAP	SAP SDN connector on page 4662
Endpoint/Identity		
	FSSO Agent on Windows AD	Fortinet single sign-on agent on page 4679
	Symantec Endpoint Protection	Symantec endpoint connector on page 4681
	Poll Active Directory Server	Poll Active Directory server on page 4680
	RADIUS Single Sign-On Agent	RADIUS single sign-on agent on page 4688
	Exchange Server	Exchange Server connector on page 4691
External Feeds		
	FortiGuard Category	External feeds on page 4695

Category	Connector	Example configuration
	IP Address	IP address external feed on page 4712
	Domain Name	Domain name external feed on page 4716
	MAC Address	MAC address external feed on page 4718
	Malware Hash	Malware hash external feed on page 4720
Generic Object Importer		
	Generic Address	Generic connector for importing addresses on page 4736

If VDOMs are enabled, SDN and External Feeds connectors are in the global settings, and Endpoint/Identity connectors are per VDOM.

Getting started with public and private SDN connectors

You can use SDN connectors to connect your FortiGate to public and private cloud solutions. By using an SDN connector, you can ensure that changes to cloud environment attributes are automatically updated in the Security Fabric. You can use SDN connector address objects to create policies that provide dynamic access control based on cloud environment attribute changes. There is no need to manually reconfigure addresses and policies whenever changes to the cloud environment occur.

There are four steps to creating and using an SDN connector:

1. Gather the required information. The required information depends on which public or private cloud solution SDN connector you are configuring.
2. Creating the SDN connector on page 4599
3. Creating an SDN connector address on page 4600
4. Adding the address to a firewall policy on page 4602

The following provides general instructions for creating an SDN connector and using the dynamic address object in a firewall policy. For instructions for specific public and private cloud solutions, see the relevant topic in this guide. For advanced scenarios regarding SDN connectors, see the appropriate FortiOS 8.0 cloud platform guide.

Creating the SDN connector

To create an SDN connector in the GUI:

1. Go to *Security Fabric* > *External Connectors*.
2. Click *Create New*.
3. Click the desired public or private cloud.
4. Enter the *Name*, *Status*, and *Update Interval* for the connector.
5. Enter previously collected information for the connector as needed.
6. Click *OK*.

To create an SDN connector in the CLI:

```
config system sdn-connector
    edit <name>
        set status {enable | disable}
        set type {connector type}
        ...
        set update-interval <integer>
    next
end
```

The available CLI commands vary depending on the selected SDN connector type.

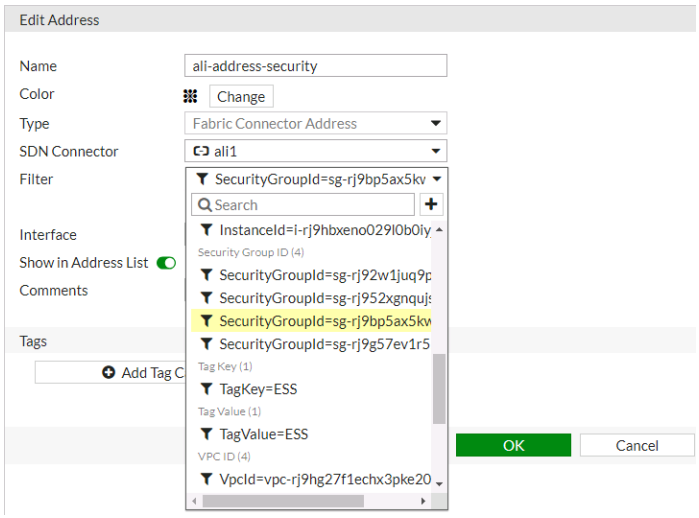
Creating an SDN connector address

You can use an SDN connector address in the following ways:

- As the source or destination address for firewall policies.
- To automatically update changes to addresses in the public or private cloud environment, based on specified filters.
- To automatically apply changes to firewall policies that use the address, based on specified filters.

To create an SDN connector address in the GUI:

1. Go to *Policy & Objects* > *Addresses* and select *Address*.
2. Click *Create new*.
3. Configure the address:
   a. Set the *Type* to *Dynamic*.
   b. From the *Sub Type* dropdown list, select *Fabric Connector Address*.
   c. From the *SDN Connector* dropdown list, select the desired SDN connector.
   d. From the *Filter* dropdown list, configure the desired filter. The filters available depend on the selected SDN connector type. The SDN connector automatically populates and updates IP addresses only for instances that satisfy the filter requirements. In this example, the address will automatically populate and update IP addresses only for AliCloud instances that belong to the specified security group:



You can set filtering conditions using multiple entries with AND ("&") or OR ("|"). When both AND and OR are specified, AND is interpreted first, then OR.

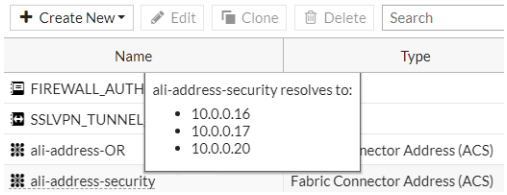
e. Configure other settings as desired.

f. Click *OK*.

4. Ensure that the SDN connector resolves dynamic firewall IP addresses as configured:

a. Go to *Policy & Objects* > *Addresses* and select *Address*.

b. Hover over the address that you created to see a list of IP addresses for instances that satisfy the filter that you configured. In this case, the IP addresses of instances that belong to the specified security group display:



To create an SDN connector address in the CLI:

1. Create the address:

```
config firewall address
    edit <name>
        set type dynamic
        set sdn <sdn_connector>
        set visibility enable
        set associated-interface <interface_name>
        set color <integer>
        ...
        set comment <comment>
        config tagging
            edit <name>
                set category <string>
                set tags <strings>
            next
```

```
        end
    next
end
```

2. Ensure that the SDN connector resolves dynamic firewall IP addresses as configured by running `show`. The following shows example output:

```
config firewall address
    edit "ali-address-security"
        set type dynamic
        config list
            edit "10.0.0.16"
            next
            edit "10.0.0.17"
            next
            edit "10.0.20.20"
            next
        end
        ...
    next
end
```

The available CLI commands vary depending on the selected SDN connector type.

Adding the address to a firewall policy

You can use an SDN connector address as the source or destination address in a policy.

To add the address to a firewall policy in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Click *Create New*.
3. Use the SDN connector address as the source or destination address.
4. Configure the remaining settings as needed.
5. Click *OK*.

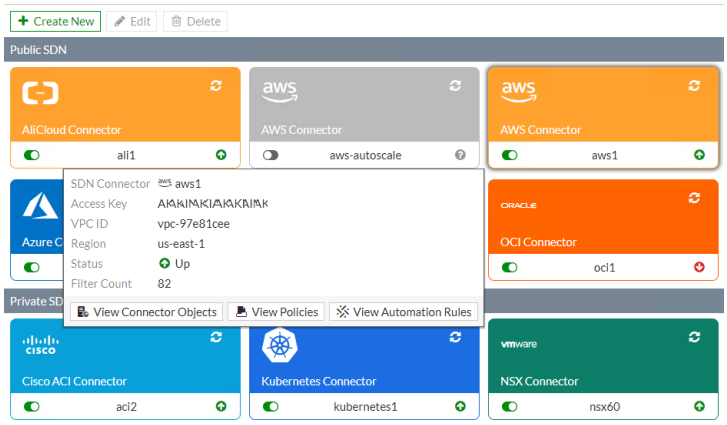
To add the address to a firewall policy in the CLI:

```
config firewall policy
    edit 0
        set name <name>
        set srcintf <port_name>
        set dstintf <port_name>
        set srcaddr <firewall_address>
        set dstaddr <firewall_address>
        set action accept
        set schedule <schedule>
        set service <service>
```

```
    next
end
```

Connector tooltips

In *Security Fabric* > *External Connectors*, hover over an SDN connector to view a tooltip that shows basic configuration information.



Three buttons provide additional information:

Button	Information
View Connector Objects	Connector's dynamic objects, such as filters and instances.
View Policies	List of policies that use the dynamic addresses from the connector.
View Automation Rules	List of automation actions that use the connector.

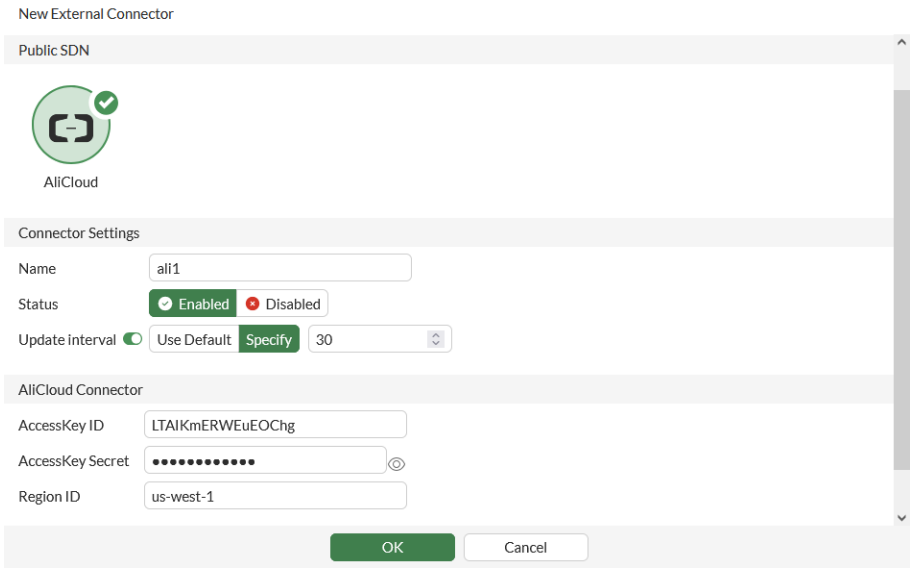
AliCloud SDN connector using access key

FortiOS automatically updates dynamic addresses for AliCloud using an AliCloud SDN connector, including mapping the following attributes from AliCloud instances to dynamic address groups in FortiOS:

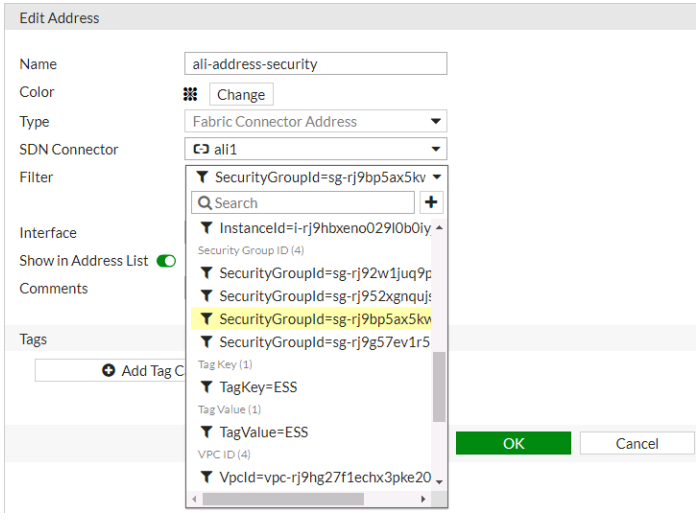
- ImageId
- InstanceId
- SecurityGroupId
- VpcId
- VSwitchId
- TagKey
- TagValue

To configure AliCloud SDN connector using the GUI:

1. Configure the AliCloud SDN connector:
   a. Go to *Security Fabric* > *External Connectors*.
   b. Click *Create New*, and select *AliCloud*.
   c. Configure as shown, substituting the access key, secret, and region ID for your deployment. The update interval is in seconds.



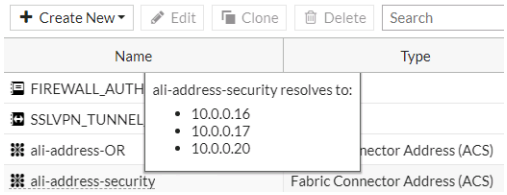
2. Create a dynamic firewall address for the configured AliCloud SDN connector:
   a. Go to *Policy & Objects* > *Addresses* and select *Address*.
   b. Click *Create new*.
   c. Configure the address as shown, selecting the desired filter in the *Filter* dropdown list. In this example, the address will automatically populate and update IP addresses only for AliCloud instances that belong to the specified security group:



3. Ensure that the AliCloud SDN connector resolves dynamic firewall IP addresses:

a. Go to *Policy & Objects* > *Addresses* and select *Address*.

b. Hover over the address created in step 2 to see a list of IP addresses for instances that belong to the security group configured in step 2:



To configure AliCloud SDN connector using CLI commands:

1. Configure the AliCloud SDN connector:

```
config system sdn-connector
    edit "ali1"
        set type acs
        set access-key "LTAIKmERWEuEOChg"
        set secret-key xxxxx
        set region "us-west-1"
        set update-interval 30
    next
end
```

2. Create a dynamic firewall address for the configured AliCloud SDN connector with the supported AliCloud filter. In this example, the AliCloud SDN Connector will automatically populate and update IP addresses only for instances that belong to the specified security group:

```
config firewall address
    edit "ali-address-security"
        set type dynamic
        set sdn "ali1"
        set filter "SecurityGroupId=sg-rj9bp5ax5kwy3gqdizqb"
    next
end
```

3. Confirm that the AliCloud SDN connector resolves dynamic firewall IP addresses using the configured filter:

```
config firewall address
    edit "ali-address-security"
        set type dynamic
        set sdn "ali1"
        set filter "SecurityGroupId=sg-rj9bp5ax5kwy3gqdizqb"
        config list
            edit "10.0.0.16"
            next
            edit "10.0.0.17"
            next
            edit "10.0.0.20"
            next
        end
```

```
    next
end
```

AWS SDN connector using access keys

FortiOS automatically updates dynamic addresses for AWS using an AWS SDN connector, including mapping attributes from AWS instances to dynamic address groups in FortiOS.

Configuring the SDN connector using the GUI, then checking the configuration using the CLI is recommended.

To configure an AWS SDN connector using the GUI:

1. Configure the AWS SDN connector:
   a. Go to *Security Fabric > External Connectors*.
   b. Click *Create New*, and select *Amazon Web Services (AWS)*.
   c. In the *Access key ID* field, enter the key created in the AWS management portal.
   d. In the *Secret access key* field, enter the secret access key accompanying the above access key.
   e. In the *Region name* field, enter the region name. Refer to AWS service endpoints for the desired region name.
   f. If desired, enable *VPC ID*. In the *VPC ID* field, enter the VPC ID within the specified region you desire to cover with the SDN connector.
   g. Click *OK*.
2. Check the configuration using the CLI:

```
config system sdn-connector
    edit "&lt;connector-name&gt;"
        show
```

The output resembles the following:

```
config system sdn-connector
    edit "&lt;connector-name&gt;"
        set access-key "&lt;example-access-key&gt;"
        set secret-key ENC &lt;example-secret-key&gt;
        set region "us-west-2"
        set vpc-id "vpc-e1e4b587"
        set update-interval 1
    next
end
```

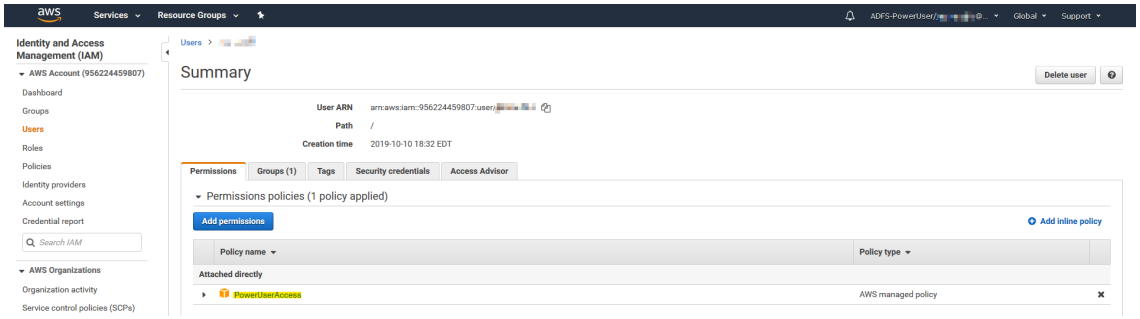
If you see that the SDN connector is disabled in *Security Fabric > External Connectors* in the GUI, run the following commands to enable the SDN connector:

```
diagnose debug application awsd -1
diagnose debug enable
```

The output may display an error like the following:

```
FGT # awsd sdn connector AWS_SDN prepare to update
awsd sdn connector AWS_SDN start updating
aws curl response err, 403
&lt;?xml version="1.0" encoding="UTF-8"?&gt;

&lt;Response&gt;&lt;Errors&gt;&lt;Error&gt;&lt;Code&gt;UnauthorizedOperation&lt;/Code&gt;&lt;M
essage&gt;You are not authorized to perform this
operation.&lt;/Message&gt;&lt;/Error&gt;&lt;/Errors&gt;&lt;RequestID&gt;8403cc11-b185-41da-
ad6d-23bb4db7d00a&lt;/RequestID&gt;&lt;/Response&gt;
awsd curl failed 403
awsd sdn connector AWS_SDN failed to get instance list
aws curl response err, 403
{"Message":"User: arn:aws:iam::956224459807:user/jcarcavallo is not authorized to perform:
eks:ListClusters on resource: arn:aws:eks:us-east-1:956224459807:cluster/*"}
awsd sdn connector AWS_SDN get EKS cluster list failed
awsd sdn connector AWS_SDN list EKS cluster failed
awsd sdn connector AWS_SDN start updating IP addresses
awsd sdn connector AWS_SDN finish updating IP addresses
awsd reap child pid: 569
```

In this case, you must configure power user access for the current administrator in the AWS management console:



After configuring power user access, run the following commands:

```
diagnose debug application awsd -1
diagnose debug enable
```

The output should display without error, as follows:

```
FGT # AWSD: update sdn connector AWS_SDN status to enabled
awsd sdn connector AWS_SDN prepare to update
awsd sdn connector AWS_SDN start updating
awsd get ec2 instance info successfully
awsd sdn connector AWS_SDN start updating IP addresses
awsd sdn connector AWS_SDN finish updating IP addresses
awsd reap child pid: 893
```

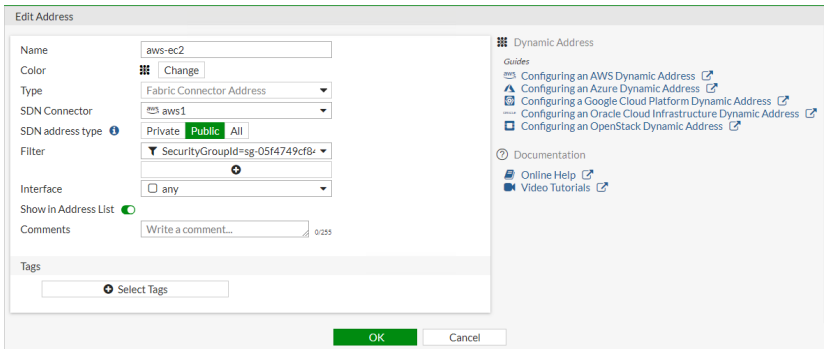
The AWS connector is now enabled.

3. Create a dynamic firewall address for the configured AWS SDN connector:
   a. Go to *Policy & Objects* > *Addresses*.
   b. Click *Create New*, then select *Address*.

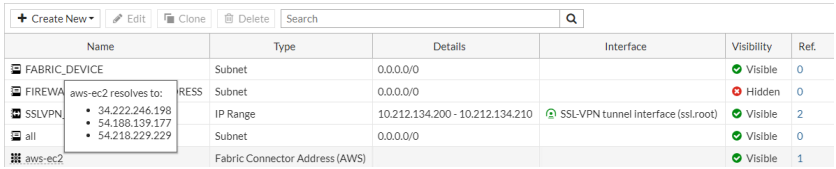
c. From the *Type* dropdown list, select *Dynamic*.
d. From the *Sub Type* dropdown list, select *Fabric Connector Address*.
e. From the *SDN Connector* dropdown list, select the AWS connector that you created.
f. In the *Filter* field, add the desired filters. The following filters are supported:

Description	Key	Example value
Architecture	architecture	x86
Autoscaling group	AutoScaleGroup	10703c-4f731e90-fortigate-payg-auto-scaling-group
AZ	placement.availabilityzone	us-east-1a
Endpoint DNS name	EndpointDNSName	vpce-06795... This filter spports the following ENI IP address types: • API gateway private endpoint • VPC endpoint for data API for Aurora • AWS PrivateLink for S3 • VPC endpoints for Lamdba
Group name	placement.groupname	
Image ID	imageId	ami-123456
Instance ID	instanceId	i-12345678
Instance type	instanceType	t2.micro
Key name	keyName	
Kubernetes (K8s) cluster	k8s_cluster	
K8s label and its name	k8s_label.Name	
K8s namespace	k8s_namespace	
K8s node name	k8s_nodename	
K8s pod name	k8s_podname	
K8s region	k8s_region	
K8s service name	k8s_servicename	
K8s zone	k8s_zone	
Private DNS name	privateDnsName	ip-172-31-10-211.us-west-2.compute.internal

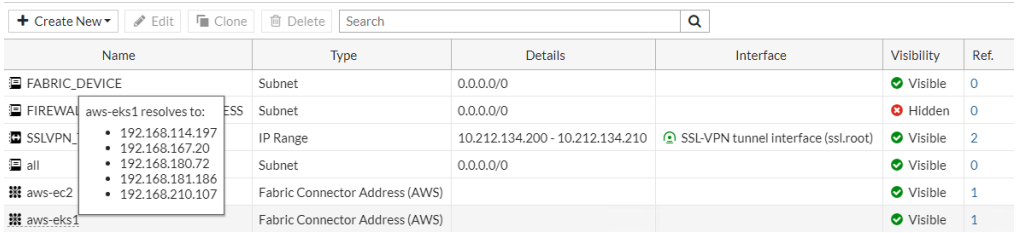
Description	Key	Example value
Public DNS name	publicDnsName	ec2-54-202-168-254.us-west-2.compute.amazonaws.com
Security group ID	SecurityGroupId	
Subnet ID	subnetId	sub-123456
Tag and its name. This key supports a maximum of eight tags.	tag.Name	
Tenancy placement	placement.tenancy	
VPC ID	VpcId	



4. Ensure that the AWS SDN connector resolves dynamic firewall IP addresses:
   a. Go to *Policy & Objects* > *Addresses*.
   b. Hover over the address that you created to see a list of IP addresses for instances that belong to the configured security group. The following is an example for a public SDN address type:



   The following is an example for a private SDN address type:



To configure AWS SDN connector using CLI commands:

1. Configure the AWS connector:

```
config system sdn-connector
    edit "&lt;connector-name&gt;"
```

```
        set access-key "&lt;example-access-key&gt;"
        set secret-key ENC &lt;example-secret-key&gt;
        set region "us-west-2"
        set vpc-id "vpc-e1e4b587"
        set update-interval 1
    next
end
```

2. Create a dynamic firewall address for the configured AWS SDN connector with the supported filter:

```
config firewall address
    edit "aws-ec2"
        set type dynamic
        set sdn "&lt;connector-name&gt;"
        set filter "SecurityGroupId=sg-05f4749cf84267548"
        set sdn-addr-type public
    next
    edit "aws-eks1"
        set type dynamic
        set sdn "&lt;connector-name&gt;"
        set filter "K8S_Region=us-west-2"
    next
end
```

3. Confirm that the AWS SDN connector resolves dynamic firewall IP addresses using the configured filter:

```
config firewall address
    edit "aws-ec2"
        set type dynamic
        set sdn "&lt;connector-name&gt;"
        set filter "SecurityGroupId=sg-05f4749cf84267548"
        set sdn-addr-type public
        config list
            edit "34.222.246.198"
            next
            edit "54.188.139.177"
            next
            edit "54.218.229.229"
            next
        end
    next
    edit "aws-eks1"
        set type dynamic
        set sdn "&lt;connector-name&gt;"
        set filter "K8S_Region=us-west-2"
        config list
            edit "192.168.114.197"
            next
            edit "192.168.167.20"
            next
            edit "192.168.180.72"
            next
```

```
            edit "192.168.181.186"
            next
            edit "192.168.210.107"
            next
        end
    next
end
```

To add an EC2 instance to test automatic address population:

1. Assume that you want to boot up another instance with an IP address of 34.222.246.178, which is currently stopped. This instance belongs to the security group that the aws-ec2 address is filtering for. In the AWS management portal, start the instance.
2. Verify that the instance is running.
3. At this point, running `show` again shows the SDN connector has automatically populated and added the 34.222.246.178 instance:

```
config firewall address
    edit "aws-ec2"
        set type dynamic
        set sdn "&lt;connector-name&gt;"
        set filter "SecurityGroupId=sg-05f4749cf84267548"
        set sdn-addr-type public
        config list
            edit "34.222.246.198"
            next
            edit "54.188.139.177"
            next
            edit "54.218.229.229"
            next
            edit "34.222.246.178"
            next
        end
    next
end
```

Therefore, administrators do not need to add this instance to the address manually. When you apply a firewall policy to this address, the policy automatically covers 34.222.246.178.

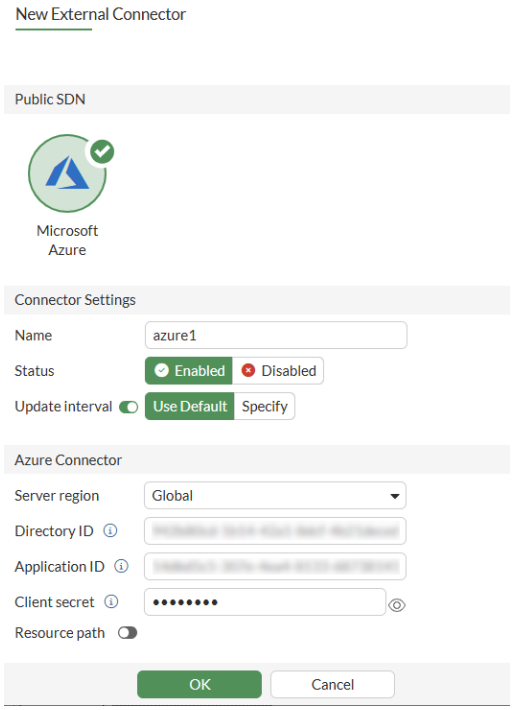
Azure SDN connector using service principal

FortiOS automatically updates dynamic addresses for Azure using Azure SDN connector, including mapping attributes from Azure instances to dynamic address groups in FortiOS.

To configure the Azure SDN connector using service principal:

1. Create an Azure SDN connector:
   a. Go to *Security Fabric* > *External Connectors* and click *Create New*.
   b. Select *Microsoft Azure*.

c. Configure the connector. See Azure SDN connector service principal configuration requirements:



d. Click *OK*.

2. Create a dynamic firewall address for the Azure connector.
   a. Go to *Policy & Objects* > *Addresses* and select Address.
   b. Click *Create new*.
   c. From the *Type* dropdown list, select *Dynamic*.
   d. From the *Sub Type* dropdown list, select *Fabric Connector Address*.
   e. From the *SDN Connector* dropdown list, select the Azure SDN connector.
   f. In the *Filter* field, add filters as desired. The Azure SDN connector supports the following filters:
      - `vm=<VM name>`
      - `securitygroup=<nsg id>`
      - `vnet=<VNet id>`
      - `subnet=<subnet id>`
      - `vmss=<VM scale set>`
      - `tag.<key>=<value>`
      - `servicetag=<value>`
      - `tag.<key>=<value>`
   g. Click *OK*.
   h. Hover the cursor over the address name to see the dynamic IP addresses that the connector resolves.

Configuring SDN connector proxy via FortiManager

FortiOS Azure SDN connector API calls can be relayed through a FortiManager proxy. FortiManager 8.0 supports this feature. This is recommended for large-scale deployments.

To configure Azure SDN connector relay through FortiManager support:

1. Configure the FortiManager:
   a. Provision an FMG_VM64_AZURE 8.0 instance in Azure. See Creating a FortiManager-VM.
   b. License the FortiManager instance. See Connecting to FortiManager.
   c. In FortiManager, go to *System Settings* > *Administrators*.
   d. Create a new administrator or edit an existing one.
   e. For *JSON API Access*, select *Read-Write*.
   f. Configure other fields as desired, then click *OK*.
2. Provision a FGT_VM64_AZURE pay as you go instance in Azure.
3. Configure the FortiManager proxy in the CLI:

```
config system sdn-proxy
    edit "FMG_proxy"
        set type fortimanager
        set server "fmg.labs.ca"
        set server-port 443
        set username "admin"
        set password "-=redacted=-"
    next
end
```

4. Configure two SDN connectors:

```
config system sdn-connector
    edit "FMG_proxy"
        set type azure
        set proxy "FMG_proxy"
        set use-metadata-iam disable
        set tenant-id "<tenant ID>"
        set client-id "<client ID>"
        set client-secret "-=redacted=-"
        set subscription-id "<subscription ID>"
        set resource-group "<resource group >"
    next
end
config firewall address
    edit "FMG_proxy"
        set type dynamic
        set sdn "FMG_proxy"
        set filter "Vnet=VNET0"
        set sdn-addr-type all
    next
end
config system sdn-connector
    edit "AZURE"
        set type azure
        set use-metadata-iam disable
        set tenant-id "<tenant ID>"
        set client-id "<client ID>"
        set client-secret "-=redacted=-"
```

```
        set subscription-id "<subscription ID>"
        set resource-group "<resource group >"
    next
end
config firewall address
    edit "AZURE"
        set type dynamic
        set sdn "AZURE"
        set filter "Vnet=VNET0"
        set sdn-addr-type all
    next
end
```

5. Go to Security Fabric > External Connectors and confirm that the connectors were created.
6. Compare the resolved IP address list between the FMG_proxy and AZURE connectors and verify that the list is complete.

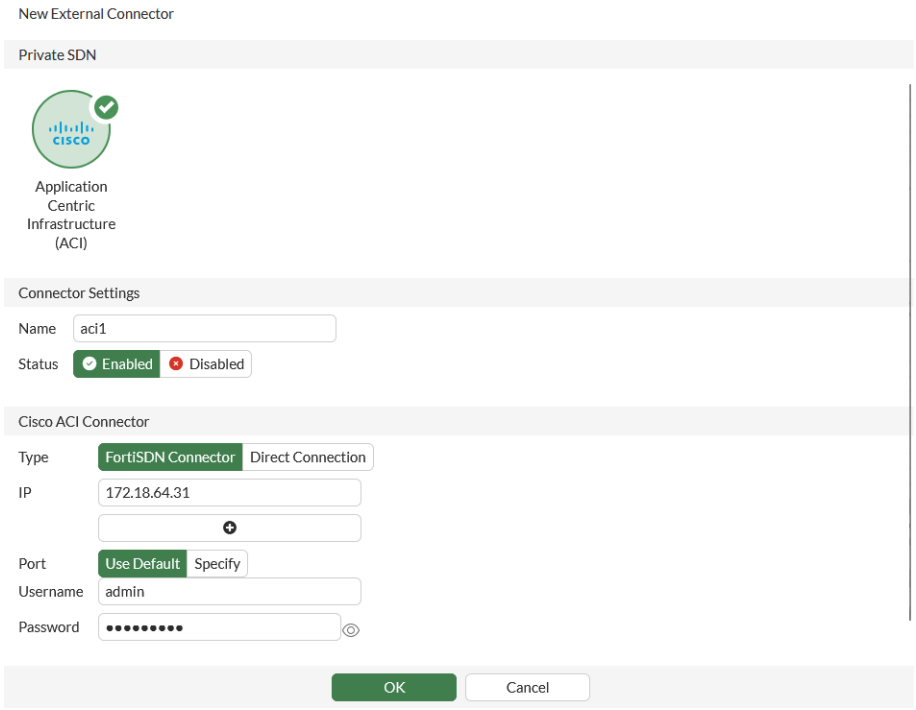
Cisco ACI SDN connector using a standalone connector

You can use Cisco ACI (Application Centric Infrastructure) SDN connectors in dynamic firewall addresses.

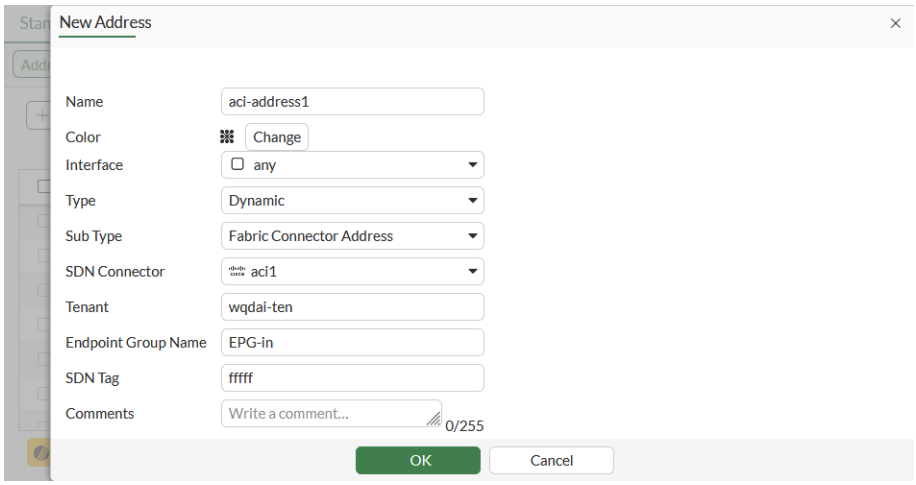
The Fortinet SDN Connector for Cisco ACI and Nuage Networks is a standalone connector that connects to SDN controllers within Cisco ACI and Nuage Networks. You must configure a connection to the Fortinet SDN connector in FortiOS to query the dynamic addresses.

To configure a Cisco ACI connector in the GUI:

1. Create the Cisco ACI SDN connector:
   a. Go to *Security Fabric > External Connectors* and click *Create New*.
   b. In the *Private SDN* section, click *Application Centric Infrastructure (ACI)*.
   c. In the *Cisco ACI Connector* section, for *Type*, select *FortiSDN Connector* and configure the remaining settings as needed.
   d. Click *OK*.



2. Create the dynamic firewall address for the connector:
   a. Go to *Policy & Objects* > *Addresses* and select *Address*.
   b. Click *Create new*.
   c. Configure the following settings:
      i. For *Type*, select *Dynamic*.
      ii. For *Sub Type*, select *Fabric Connector Address*.
      iii. For *SDN Connector*, select the first ACI connector.
      iv. Configure the remaining settings as needed.
   d. Click *OK*.



To verify the dynamic firewall IPs are resolved by the SDN connector in the GUI:

1. Go to *Policy & Objects* > *Addresses* and select *Address*.
2. In the address table, hover over the address to view which IPs it resolves to.

To configure a Cisco ACI connector in the CLI:

1. Create the SDN connector:

```
config system sdn-connector
    edit "aci1"
        set type aci
        set server "172.18.64.31"
        set username "admin"
        set password xxxxxxx
    next
end
```

2. Create the dynamic firewall address for the connector:

```
config firewall address
    edit "aci-address1"
        set type dynamic
        set sdn "aci1"
        set color 17
        set tenant "wqdai-ten"
        set epg-name "EPG-in"
        set sdn-tag "fffff"
    next
end
```

To verify the dynamic firewall IPs are resolved by the SDN connector in the CLI:

```
# diagnose firewall dynamic list

List all dynamic addresses:
aci1.aci.wqdai-ten.EPG-in.fffff: ID(171)
        ADDR(192.168.100.20)
```

Import IPv6 addresses from an APIC controller

IPv6 addresses can be imported from an APIC controller. The ACI direct connection SDN connectors support the `Tenant`, `Application`, `Epg`, `Tag`, and `Esg` filters.

In the following example, an APIC controller with IPv6 addresses is configured. The IPv6 addresses are then imported to the FortiGate through an ACI direct connection SDN connector.

To import IPv6 addresses from an APIC controller in the CLI:

1. Configure the external APIC controller with IPv6 addresses configured in the subnets.
2. In the FortiGate, configure the ACI direct connection SDN connector to connector to the APIC controller:

```
config system sdn-connector
    edit "aci_direct_van"
        set type aci-direct
        set verify-certificate disable
        set server-list "10.10.8.35"
        set username "admin"
        set password xxxxxx
    next
end
```

3. Confirm that the SDN connector status is up:

```
# diagnose sys sdn status aci_direct_van
SDN Connector                       Type        Status
------------------------------------------------------------
aci_direct_van                      aci-direct  Up
```

4. Create an IPv6 dynamic firewall address associated with the SDN connector and define specific filters to filter out needed addresses:

```
config firewall address6
    edit "aci_direct_v6_epg_1"
        set type dynamic
        set sdn "aci_direct_van"
        set filter "Epg=*lzou*"
    next
    edit "aci_direct_v6_tenant_1"
        set type dynamic
        set sdn "aci_direct_van"
        set filter "Tenant=lzou-v6"
    next
    edit "aci_direct_v6_app_1"
        set type dynamic
        set sdn "aci_direct_van"
        set filter "Application=*ap*"
    next
    edit "aci_direct_v6_tag_1"
        set type dynamic
        set sdn "aci_direct_van"
        set filter "tag.lzou-app-key=lzou-app-val"
    next
    edit "aci_direct_v6_esg_1"
        set type dynamic
        set sdn "aci_direct_van"
        set filter "Esg=lzou-esg*"
    next
end
```

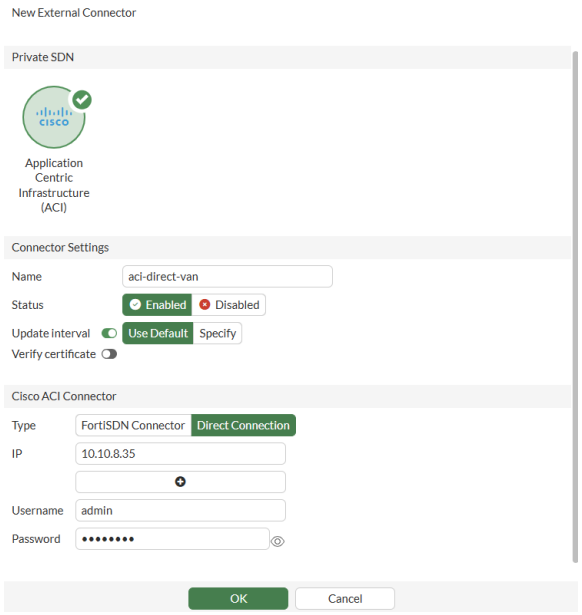
5. Review the resolved IPv6 addresses that were saved in the IPv6 firewall address configuration:

```
# show firewall address6
config firewall address6
......
    edit "aci_direct_v6_epg_1"
        set type dynamic
        set sdn "aci_direct_van"
        set filter "Epg=*lzou*"
        config list
            edit "fd00:0:0:1::111"
            next
            edit "fd00:0:0:2::222"
            next
        end
    next
    edit "aci_direct_v6_tenant_1"
        set type dynamic
        set sdn "aci_direct_van"
        set filter "Tenant=lzou-v6"
        config list
            edit "fd00:0:0:1::111"
            next
            edit "fd00:0:0:2::222"
            next
        end
    next
    edit "aci_direct_v6_app_1"
        set type dynamic
        set sdn "aci_direct_van"
        set filter "Application=*ap*"
        config list
            edit "2001:db8:3333:4444:cccc:dddd:eeee:ffff"
            next
            edit "2001:db8:85a3::8a2e:370:1122"
            next
        end
    next
    edit "aci_direct_v6_tag_1"
        set type dynamic
        set sdn "aci_direct_van"
        set filter "tag.lzou-app-key=lzou-app-val"
        config list
            edit "fd00:0:0:1::111"
            next
        end
    next
    edit "aci_direct_v6_esg_1"
        set type dynamic
        set sdn "aci_direct_van"
        set filter "Esg=lzou-esg*"
        config list
```

```
            edit "fd00:0:0:1::111"
            next
        end
    next
end
```

To import IPv6 addresses from an APIC controller in the GUI:

1. Configure the external APIC controller with IPv6 addresses configured in the subnets.
2. In the FortiGate, configure the ACI direct connection SDN connector to connector to the APIC controller:
   a. Go to *Security Fabric* > *External Connectors*.
   b. Click *Create New*.
   c. Select the SDN type.
   d. Enter a *Name*.
   e. Set the *Type* to *Direct Connection*.
   f. Enter the IP address.
   g. Configure other fields, as needed.
   h. Click *OK*.

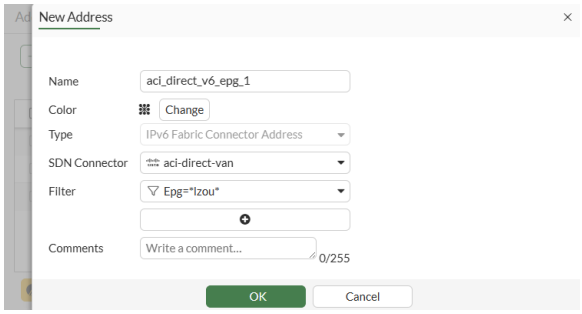


3. Confirm that the SDN connector status is up in the CLI:

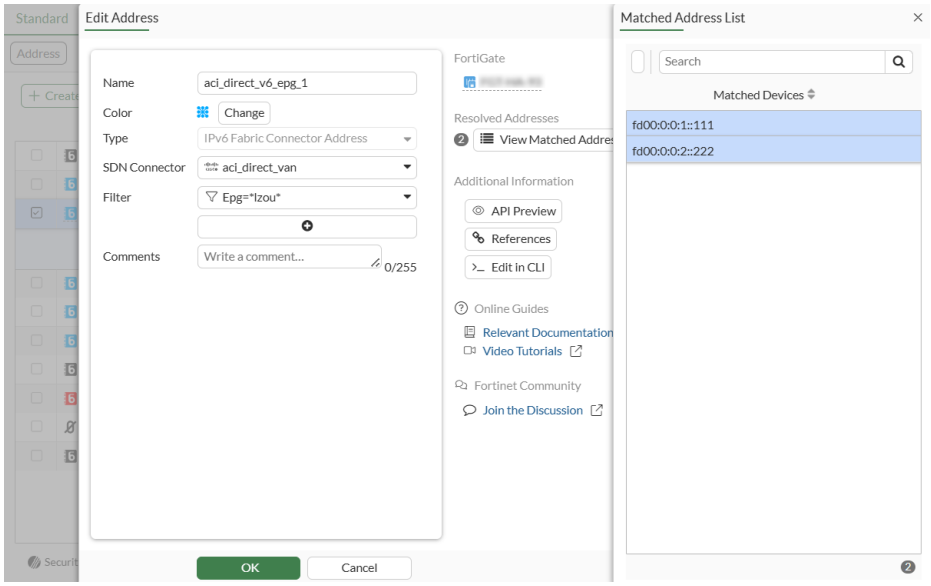
```
# diagnose sys sdn status aci_direct_van
SDN Connector                       Type        Status
------------------------------------------------------------
aci-direct-van                      aci-direct  Up
```

4. Create an IPv6 dynamic firewall address associated with the SDN connector:
   a. Go to *Policy & Objects* > *Addresses*.
   b. Select the *IPv6 Address* tab.
   c. Click *Create New*.

d. Enter a *Name*.
e. Set the *Type* to *IPv6 Fabric Connector Address*.
f. Select the *SDN Connector* previously created.
g. Set the *Filter* selection to filter out needed addresses.
h. Click *OK*.



5. Review the resolved IPv6 addresses that were saved in the IPv6 firewall address configuration:
   a. Go to *Policy & Objects* > *Addresses*.
   b. Select the *IPv6 Address* tab.
   c. Edit the new address.
   d. Click *View Matched Addresses*. A list is displayed of resolved IPv6 addresses.

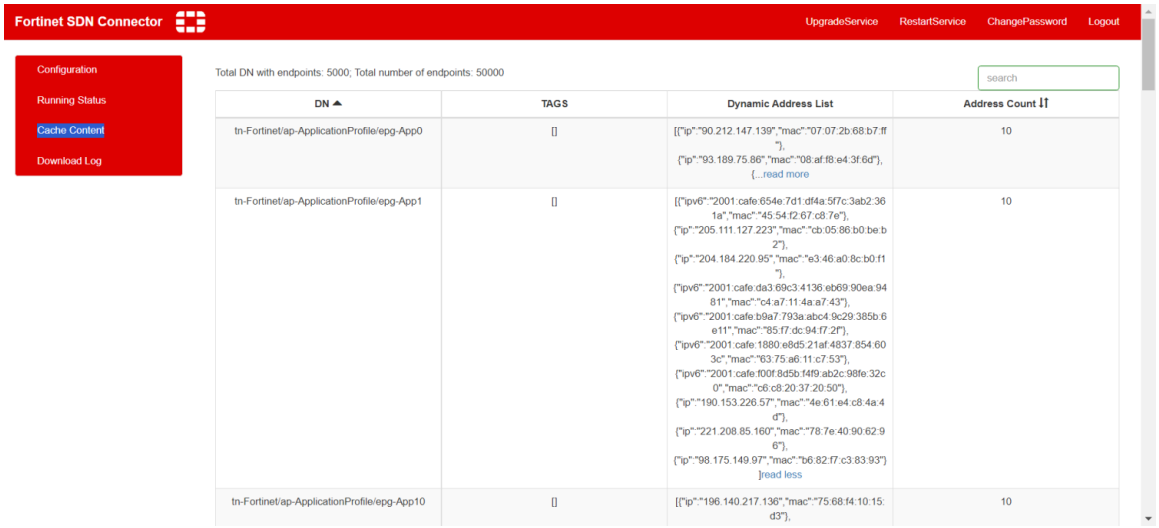


Retrieve IPv6 dynamic addresses from Cisco ACI SDN connector

IPv6 dynamic addresses can be retrieved from Cisco ACI SDN connectors. IPv6 addresses imported from Cisco ACI to the Fortinet SDN Connector VM can be imported into the FortiGate as IPv6 dynamic addresses. The Fortinet SDN Connector VM must be running version 1.1.10 or later.

```
config firewall address6
    edit <name>
        set type dynamic
        set sdn <ACI_connector>
    next
end
```

The following example assumes the Fortinet SDN Connector VM has already connected to Cisco ACI and learned the IPv6 addresses. See Configuring the SDN Connector in the Cisco ACI Administration Guide for more information. The *Dynamic Address List* values for the DN with the filter *tn-Fortinet/ap-ApplicationProfile/epg-App1* is used in this example.



To configure the Cisco ACI connector and dynamic address:

1. Configure the Cisco ACI SDN connector:

```
config system sdn-connector
    edit "aci_64.115_115"
        set type aci
        set server-list "10.6.30.115"
        set server-port 5671
        set username "admin"
        set password xxxxxxx
    next
end
```

2. Verify that the SDN connector status is up:

```
# diagnose sys sdn status "aci_64.115_115"
SDN Connector                       Type        Status
------------------------------------------------------------
aci_64.115_115                      aci         Up
```

3. Configure the IPv6 dynamic firewall address (filters for tenant and endpoint group are used in this example):

```
config firewall address6
    edit "aci-add6-App1"
        set type dynamic
        set sdn "aci_64.115_115"
        set color 17
        set tenant "Fortinet"
        set epg-name "App1"
    next
end
```

4. Verify the list of resolved IPv6 addresses:

```
# diagnose firewall dynamic6 list "aci-add6-App1"
aci_64.115_115.aci.Fortinet.App1.*: ID(220)
        ADDR(2001:cafe:654e:7d1:df4a:5f7c:3ab2:361a)
        ADDR(2001:cafe:da3:69c3:4136:eb69:90ea:9481)
        ADDR(2001:cafe:b9a7:793a:abc4:9c29:385b:6e11)
        ADDR(2001:cafe:1880:e8d5:21af:4837:854:603c)
        ADDR(2001:cafe:f00f:8d5b:f4f9:ab2c:98fe:32c0)
```

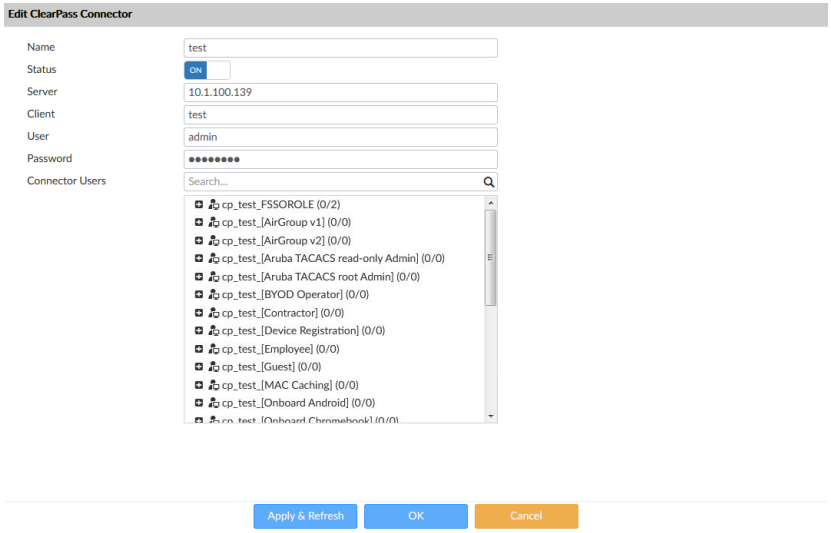
ClearPass endpoint connector via FortiManager

ClearPass Policy Manager (CPPM) is a network access system that can send information about authenticated users to third party systems, such as a FortiGate or FortiManager.

In this example, communications are established between CPPM and FortiManager, and then the FortiManager forwards information to a managed FortiGate. On the FortiGate, the user information can be used in firewall policies and added to FSSO dynamic addresses.

Configure the FortiManager

Establish communications between FortiManager and CPPM so that FortiManager can synchronize CPPM user groups. See Creating a ClearPass connector in the FortiManager Administration Guide.



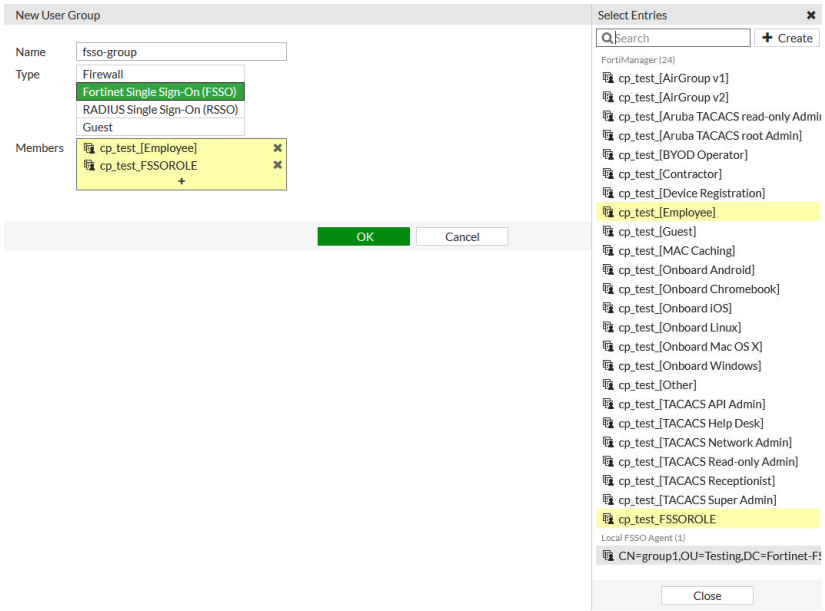
FortiManager forwards the group information to managed FortiGates.

Adding CPPM FSSO user groups to a local user group

To add CPPM user groups to a local user group in the GUI:

1. On the FortiGate, go to *User & Authentication* > *User Groups*.
2. Click *Create New*.
3. Enter a name for the group and set *Type* to *Fortinet Single Sign-On (FSSO)*.
4. Click the *Members* field, and add one or more FSSO groups.

   FSSO groups can come from multiple sources; CPPM FSSO groups are prefixed with *cp_* and are listed under the *FortiManager* heading.



5. Click *OK*.

To add CPPM user groups to a local user group in the CLI:

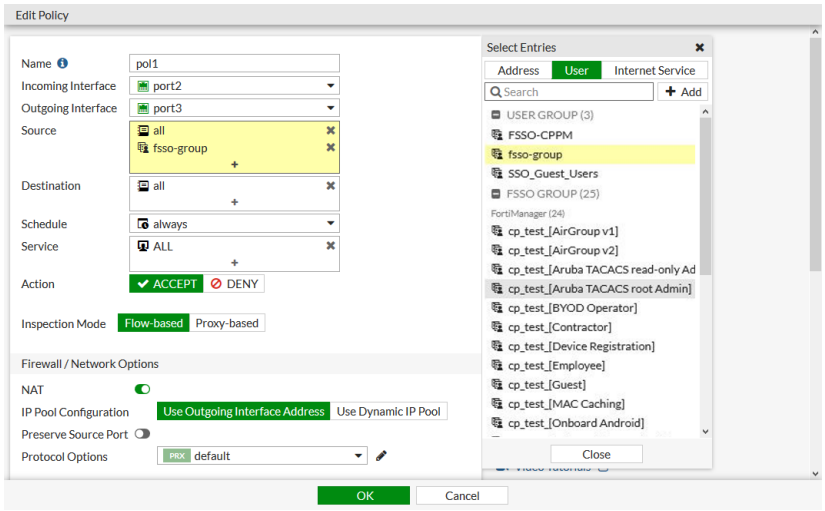
```
config user group
    edit fsso-group
        set group-type fsso-service
        set member "cp_test_[Employee]" "cp_test_FSSOROLE"
    next
end
```

Using the local FSSO user group in a firewall policy

To add the local FSSO user group to a firewall policy in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Create a new policy, or edit an existing one.

3. Click in the *Source* field and add the *fsso-group* user group.



CPPM user groups can also be added directly to the policy.

4. Click *OK*.

To add the local FSSO user group to a firewall policy in the CLI:

```
config firewall policy
    edit 1
        set name "pol1"
        set srcintf "port2"
        set dstintf "port3"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set groups "fsso-group"
        set nat enable
    next
end
```

Verification

To verify that a user was added to the FSSO list on the FortiGate:

1. Log on to the client and authenticate with CPPM.
   After successful authentication, the user is added to the FSSO list on the FortiGate.
2. On the FortiGate, go to *Dashboard* > *Assets & Identities* and look at the *Firewall Users* widget to verify that the user was added.

The user group *cp_test_FSSOROLE* is listed separately because the user is a member of that group on the CPPM.

To verify that traffic can pass the firewall:

1. Log on to the client and browse to an external website.
2. On the FortiGate, go to *Dashboard* > *FortiView Sources*.
3. Double-click on the user and select the *Destinations* tab to verify that traffic is being passed by the firewall.

To verify the user address groups:

```
show user adgrp
    config user adgrp
        edit "cp_test_FSSOROLE"
            set server-name "FortiManager"
        next
        edit "cp_test_[AirGroup v1]"
            set server-name "FortiManager"
        next
        edit "cp_test_[AirGroup v2]"
            set server-name "FortiManager"
        next
        edit "cp_test_[Aruba TACACS read-only Admin]"
            set server-name "FortiManager"
        next
        edit "cp_test_[Aruba TACACS root Admin]"
            set server-name "FortiManager"
        next
        edit "cp_test_[BYOD Operator]"
            set server-name "FortiManager"
        next
        edit "cp_test_[Contractor]"
            set server-name "FortiManager"
        next
        edit "cp_test_[Device Registration]"
            set server-name "FortiManager"
        next
        ...
        edit "CN=group1,OU=Testing,DC=Fortinet-FSSO,DC=COM"
            set server-name "Local FSSO Agent"   <----- !!!
        next
    end
```

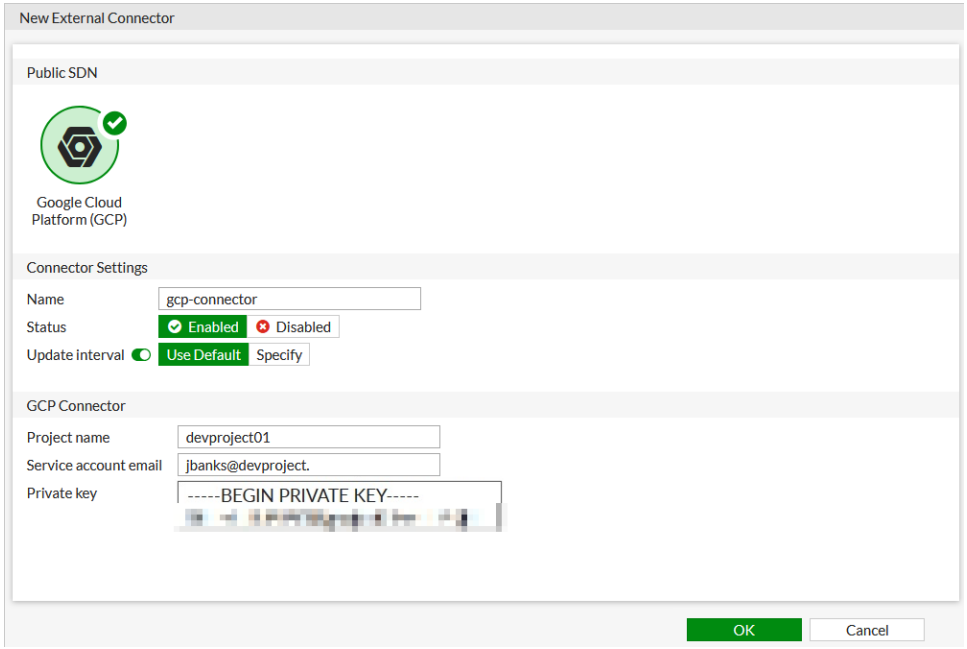
GCP SDN connector using service account

FortiOS automatically updates dynamic addresses for GCP using a GCP SDN connector, including mapping attributes from GCP instances to dynamic address groups in FortiOS.

To configure GCP connector using the GUI:

1. In FortiOS, go to *Security Fabric* > *External Connectors*.
2. Click *Create New*, and select *Google Cloud Platform (GCP)*.

   Note you can create only one SDN Connector per connector type. For example, you can create one entry for GCP.
3. Configure the connector as follows:
   a. *Project name*: Enter the name of the GCP project. The VMs whose IP addresses you want to populate should be running within this project.
   b. *Service account email*: Enter the email address associated with the service account that will call APIs to the GCP project specified above.
   c. *Private key*: Enter the private key statement as shown in the text box. For details, see Creating a GCP service account.



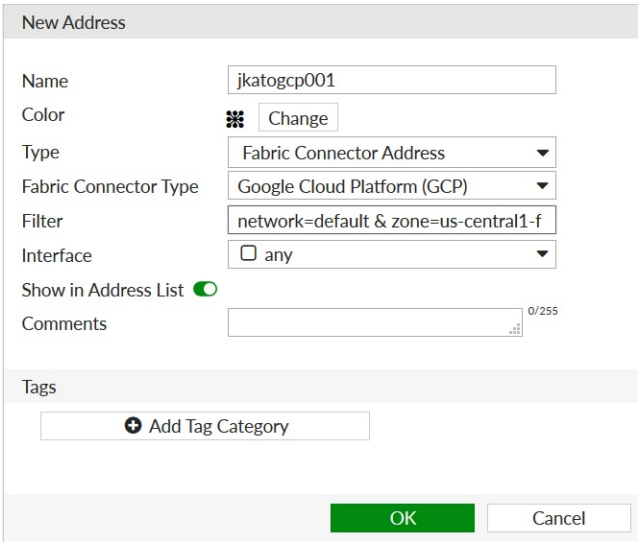
   Once the connector is successfully configured, a green indicator appears at the bottom right corner. If the indicator is red, the connector is not working. See Troubleshooting GCP SDN Connector.
4. Create a dynamic firewall address for the configured GCP SDN connector:
   a. Go to *Policy & Objects* > *Addresses* and select *Address*.
   b. Click *Create new*.
   c. Configure the address:
      i. *Name*: Enter the desired name.
      ii. *Type*: Select *Dynamic*.
      iii. *Fabric Connector Type*: Select *Google Cloud Platform (GCP)*.
      iv. *Filter*: The SDN connector automatically populates and updates only instances that match this filtering condition. Currently GCP supports the following filters:
         - `id=<instance id>` : This matches an VM instance ID.
         - `name=<instance name>` : This matches a VM instance name.
         - `zone=<gcp zones>` : This matches a zone name.

- `network=<gcp network name>` : This matches a network name.
- `subnet=<gcp subnet name>` : This matches a subnet name.
- `tag=<gcp network tags>` : This matches a network tag.
- `label.<gcp label key>=<gcp label value>` : This matches a free form GCP label key and its value.

In the example, the filter is set as `'network=default & zone=us-central-1f'`. This configuration populates all IP addresses that belong to the default network in the zone us-central-1f.

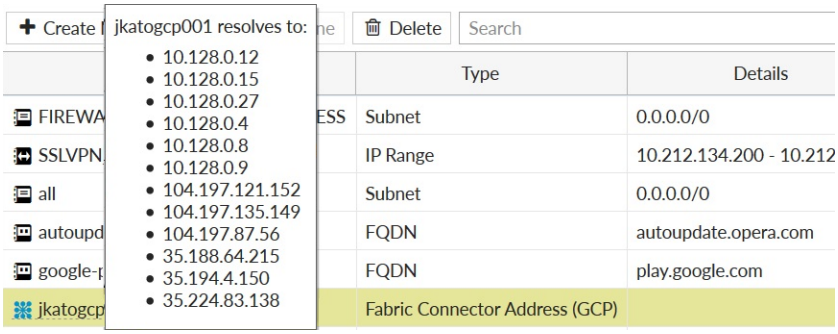
You can set filtering conditions using multiple entries with AND ("&") or OR ("|"). When both AND and OR are specified, AND is interpreted first, then OR.

Note that wildcards (such as the asterisk) are not allowed in filter values.



v. Click *OK*.

The address has been created. Wait for a few minutes before the setting takes effect. You will know that the address is in effect when the exclamation mark disappears from the address entry. When you hover over the address, you can see the list of populated IP addresses.



If the exclamation mark does not disappear, check the address settings.

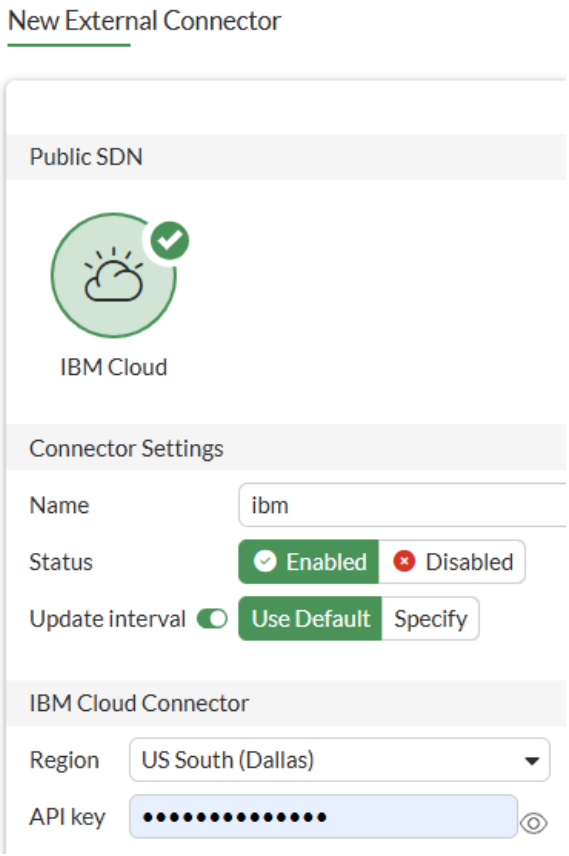
IBM Cloud SDN connector using API keys

FortiOS can automatically update dynamic addresses for IBM Cloud using an SDN connector. For information on creating and managing the API key, see the IBM Cloud documentation.

In 8.0.0, you do not need to specify the generation of IBM Compute Generation.

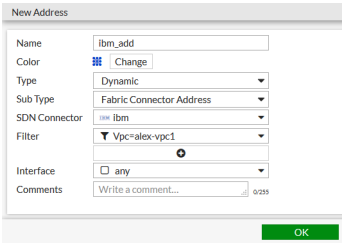
To configure an IBM Cloud SDN connector using the GUI:

1. Create an SDN connector:
   a. Go to *Security Fabric* > *External Connectors*.
   b. Click *Create New*, then select *IBM Cloud*.
   c. In the *Name* field, enter the desired connector name.
   d. For *Status*, select *Enabled*.
   e. For *Update interval*, configure the desired update interval for dynamic object updates.
   f. From the *Region* dropdown list, select the IBM Cloud region.
   g. In the *API key* field, enter the IBM Cloud API key.



   h. Click *OK*.
2. Create dynamic firewall addresses for the configured connector:
   a. Go to *Policy & Objects* > *Addresses* and select *Address*.
   b. Click *Create new*.
   c. From the *Type* dropdown list, select *Dynamic*.
   d. From the *Sub Type* dropdown list, select *Fabric Connector Address*.
   e. From the *SDN Connector* dropdown list, select the IBM SDN connector.
   f. In the *Filter* field, add the desired filters. The following filters are supported:

- <InstanceId>
- <InstanceName>
- <ImageId>
- <ImageName>
- <Architecture>
- <Profile>
- <Vpc>
- <Zone>
- <Subnet>
- <ResourceGroup>

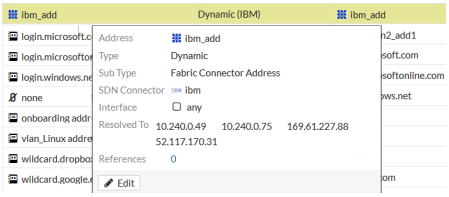


g. Click *OK*.

3. Ensure that the connector resolves dynamic firewall IP addresses:

a. Go to *Policy & Objects* > *Addresses*.

b. Hover over the addresses created in step 2 to see a list of IP addresses that the connector has resolved:



To configure an IBM Cloud SDN connector using the CLI:

1. Create SDN connector:

```
config system sdn-connector
    edit "ibm"
        set status enable
        set type ibm
        set api-key xxxxxx
        set ibm-region us-south
        set update-interval 60
    next
end
```

2. Create dynamic firewall addresses for the configured connector:

```
config firewall address
    edit "ibm_add"
        set type dynamic
```

```
        set sdn "ibm"
        set color 19
        set filter "Vpc=alex-vpc1"
    next
end
```

3. Ensure that the connector resolves dynamic firewall IP addresses:

```
# show firewall address ibm_add
config firewall address
    edit "ibm_gen1_add"
        set uuid 586841c4-7f46-51ea-dc66-dbf840af03d3
        set type dynamic
        set sdn "ibm"
        set color 19
        set filter "Vpc=alex-vpc1"
        config list
            edit "10.240.0.49"
            next
            edit "10.240.0.75"
            next
            edit "169.61.227.88"
            next
            edit "52.117.170.31"
            next
        end
    next
end
```

Kubernetes (K8s) SDN connectors

The following topics provide information about configuring Kubernetes SDN connectors:

- AliCloud Kubernetes SDN connector using access key on page 4630
- EKS SDN connector using access key on page 4633
- Azure Kubernetes (AKS) SDN connector using client secret on page 4635
- GCP Kubernetes (GKE) SDN connector using service account on page 4638
- Oracle Kubernetes (OKE) SDN connector using certificates on page 4641
- Private cloud K8s SDN connector using secret token on page 4644
- Multus CNI for Kubernetes connectors on page 4647

AliCloud Kubernetes SDN connector using access key

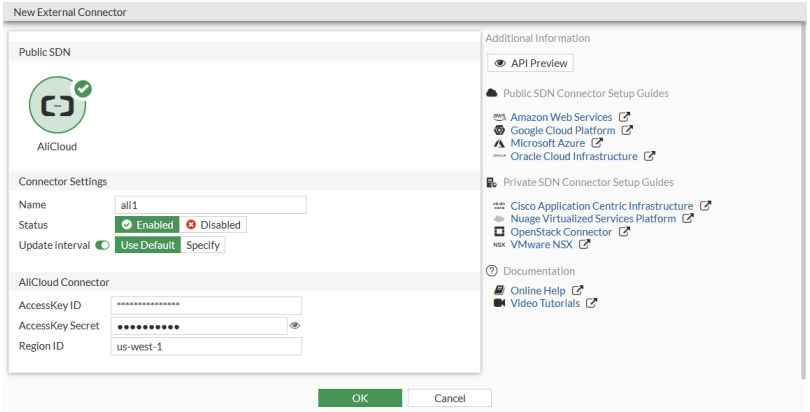
When an AliCloud SDN connector is configured, dynamic address objects can support Kubernetes filters based on cluster, service, node, pod, and more.

The following address filters can be applied:

- K8S_Cluster
- K8S_Namespace
- K8S_ServiceName
- K8S_NodeName
- K8S_PodName
- K8S_Region
- K8S_Zone
- K8S_Label

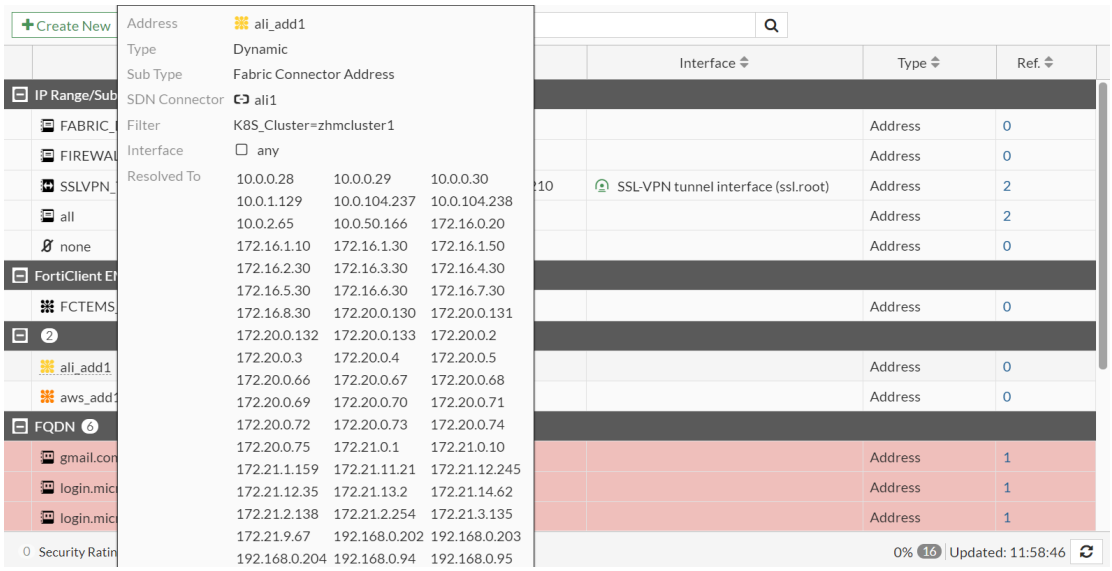
To configure an AliCloud SDN connector with a Kubernetes filter in the GUI:

1. Configure the AliCloud SDN connector:
   a. Go to *Security Fabric* > *External Connectors*.
   b. Click *Create New*, and select *AliCloud*.
   c. Configure the settings as needed and click *OK*.



2. Create a dynamic firewall address with the supported Kubernetes filter:
   a. Go to *Policy & Objects* > *Addresses* and select *Address*.
   b. Click *Create new* and enter a name.
   c. Configure the following settings:
      i. For *Type*, select *Dynamic*.
      ii. For *Sub Type*, select *Fabric Connector Address*.
      iii. For *SDN Connector*, select the connector created in step 1.
      iv. For *SDN address type*, select *Private*.
      v. For *Filter*, select *K8S_Cluster=zhmcluster*.
   d. Click *OK*.

      The corresponding IP addresses are dynamically updated and resolved after applying the Kubernetes filter.
3. Confirm that the AliCloud SDN connector resolves dynamic firewall IP addresses using the configured filter:
   a. Go to *Policy & Objects* > *Addresses* and select *Address*.
   b. In the address table, hover over the address created in step 2 to view which IPs it resolves to:



To configure an AliCloud SDN connector with a Kubernetes filter in the CLI:

1. Configure the AliCloud SDN connector:

```
config system sdn-connector
    edit "ali1"
        set type alicloud
        set access-key "*****************"
        set secret-key xxxxxxxx
        set region "us-west-1"
    next
end
```

2. Create a dynamic firewall address with the supported Kubernetes filter:

```
config firewall address
    edit "ali_add1"
        set type dynamic
        set sdn "ali1"
        set color 10
        set filter "K8S_Cluster=zhmcluster1"
    next
end
```

3. Confirm that the AliCloud SDN connector resolves dynamic firewall IP addresses using the configured filter:

```
config firewall address
    edit "ali_add1"
        show
            config firewall address
                edit "ali_add1"
                    set uuid c48e4f00-5435-51eb-0547-aced5cf80f1f
                    set type dynamic
                    set sdn "ali1"
```

```
                    set color 10
                    set filter "K8S_Cluster=zhmcluster1"
                    config list
                        edit "10.0.0.28"
                        next
                        edit "10.0.0.29"
                        next
                        edit "10.0.0.30"
                        next
                        ...
                    end
                next
            end
    next
end
```

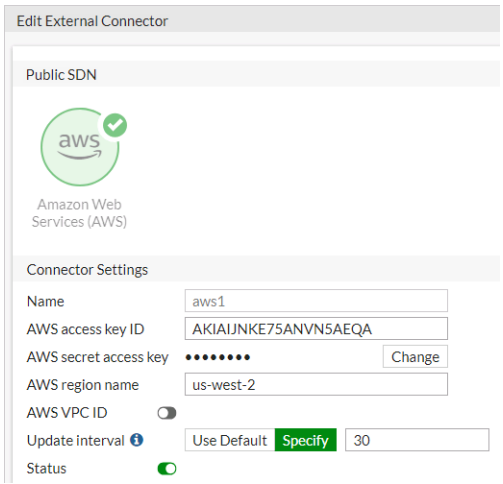
EKS SDN connector using access key

AWS SDN connectors support dynamic address groups based on AWS Kubernetes (EKS) filters.

If there is an authorization issue with the dynamic address resolution of Kubernetes (K8s) IP addresses and/or the K8s dynamic addresses fail to display, confirm that you have set the correct Identity & Access Management permissions or role assignments. For more information, see How do I provide access to other IAM users and roles after cluster creation in Amazon EKS?.

To enable an AWS SDN connector to fetch IP addresses from EKS:

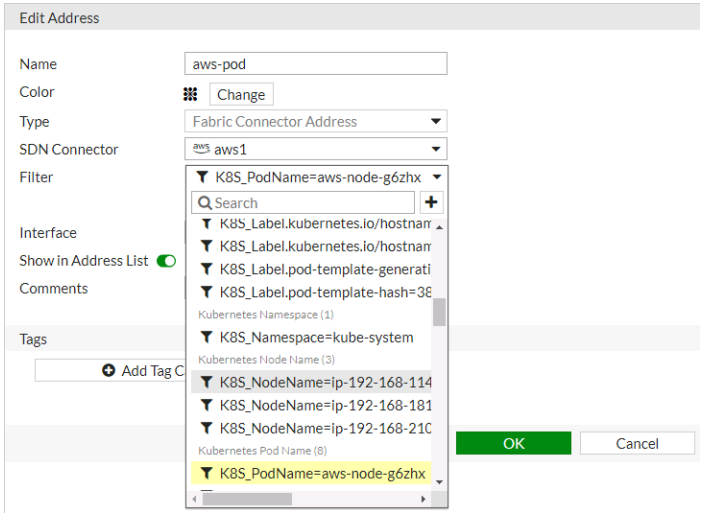
1. Go to *Security Fabric > External Connectors*. Click *Create New*, then select *Amazon Web Services (AWS)*. Configure the SDN connector as desired. See AWS SDN connector using access keys on page 4606.



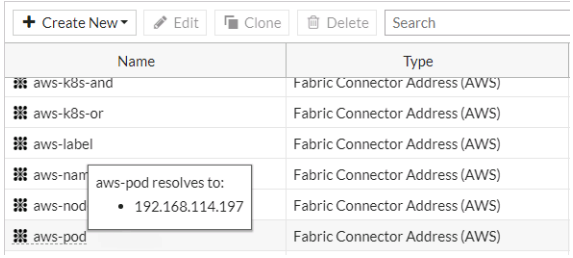
2. Go to *Policies & Objects > Addresses* and select *Address*.
3. Click *Create new* to create a dynamic firewall address for the configured SDN connector using the supported K8s filter.
4. From the *Type* dropdown list, select *Dynamic*.

5. From the *Sub Type* dropdown list, select *Fabric Connector Address*.
6. From the *SDN Connector* dropdown list, select the desired SDN connector.
7. In the *Filter* field, add the desired filters. The following filters are supported:

Filter	Description
k8s_cluster	Name of K8s cluster.
k8s_namespace	Namespace of a K8s service or pod.
k8s_svcname	Name of a K8s service.
k8s_nodename	Name of a K8s node.
k8s_zone	Zone of a K8s node.
k8s_region	Region of a Kubernetes node.
k8s_podname	Name of a Kubernetes pod.
k8s_label.xxx	Name of label of a Kubernetes resource (cluster/service/node/pod).



8. Configure the rest of the settings, then click *OK*.
9. Ensure that the SDN connector resolves the dynamic firewall address IP addresses by going to *Policy & Objects* > *Addresses* and hovering over the newly created address.



To configure an EKS SDN connector through the CLI:

1. Configure the SDN connector:

```
config system sdn-connector
    edit "aws1"
        set type aws
        set access-key "AKIAIJNKE75ANVN5AEQA"
        set secret-key xxxxx
        set region "us-west-2"
        set update-interval 30
    next
end
```

2. Create a dynamic firewall address for the SDN connector with a supported Kubernetes filter:

```
config firewall address
    edit "aws-pod"
        set type dynamic
        set sdn "aws1"
        set filter "K8S_PodName=aws-node-g6zhx"
    next
end
```

The SDN connector resolves the dynamic firewall address IP address:

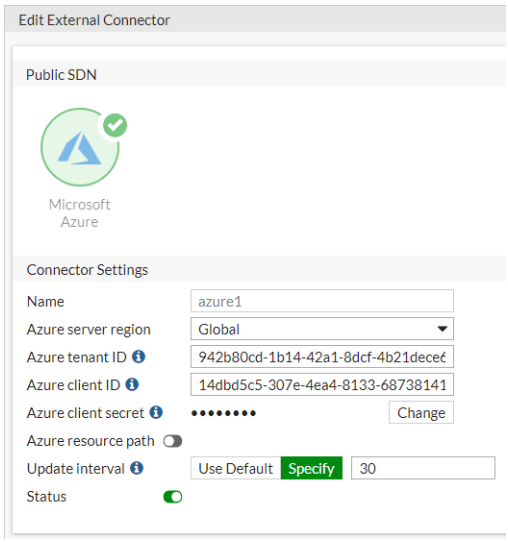
```
config firewall address
    edit "aws-pod"
        set type dynamic
        set sdn "aws1"
        set filter "K8S_PodName=aws-node-g6zhx"
        config list
            edit "192.168.114.197"
            next
        end
    next
end
```

Azure Kubernetes (AKS) SDN connector using client secret

Azure SDN connectors support dynamic address groups based on Azure Kubernetes (AKS) filters.

To enable an Azure SDN connector to fetch IP addresses from Azure Kubernetes:

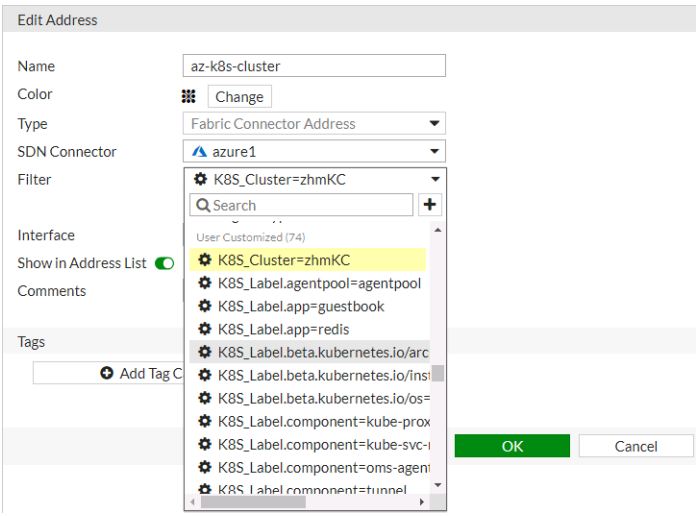
1. Configure the Azure SDN connector:
   a. Go to *Security Fabric* > *External Connectors*.
   b. Click *Create New*, and select *Azure*.
   c. Configure as shown substituting the region, tenant and client IDs, and client secret for your deployment. See Azure SDN connector service principal configuration requirements.



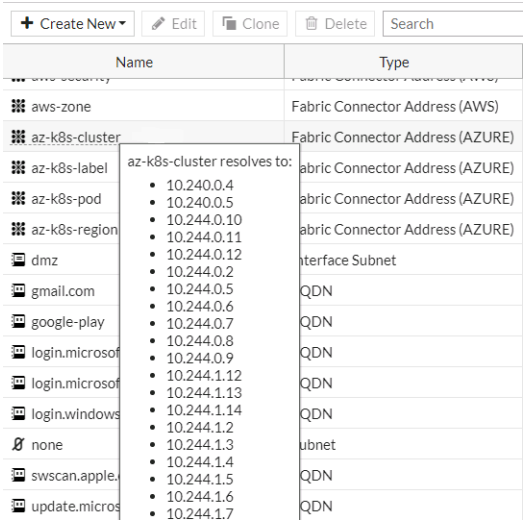
2. Create a dynamic firewall address for the configured K8s SDN connector:
   a. Go to *Policy & Objects* > *Addresses* and select *Address*.
   b. Click *Create new*.
   c. From the *Type* dropdown list, select *Dynamic*.
   d. From the *Sub Type* dropdown list, select *Fabric Connector Address*.
   e. From the *SDN Connector* dropdown list, select the desired SDN connector.
   f. In the *Filter* field, add the desired filter. The following filters are supported:

Filter	Description
k8s_cluster	Name of Kubernetes cluster.
k8s_namespace	Namespace of a Kubernetes service or pod.
k8s_svcname	Name of a Kubernetes service.
k8s_nodename	Name of a Kubernetes node.
k8s_zone	Zone of a Kubernetes node.
k8s_region	Region of a Kubernetes node.
k8s_podname	Name of a Kubernetes pod.
k8s_label.xxx	Name of label of a Kubernetes resource (cluster/service/node/pod).

In this example, the address is configured to automatically populate and update IP addresses only for instances that belong to the zhmKC cluster:



3. Ensure that the K8s SDN connector resolves dynamic firewall IP addresses:
   a. Go to *Policy & Objects* > *Addresses*.
   b. Hover over the address created in step 2 to see a list of IP addresses for instances that belong to the zhmKC cluster as configured in step 2:



To configure an Azure Kubernetes SDN connector through the CLI:

1. Configure the SDN connector:

```
config system sdn-connector
    edit "azure1"
        set type azure
        set tenant-id "942b80cd-1b14-42a1-8dcf-4b21dece61ba"
        set client-id "14dbd5c5-307e-4ea4-8133-68738141feb1"
        set client-secret xxxxx
        set update-interval 30
    next
end
```

2. Create a dynamic firewall address for the SDN connector with a supported Kubernetes filter. In this example, the address will automatically populate and update IP addresses only for instances that belong to the zhmKC cluster:

```
config firewall address
    edit "az-k8s-cluster"
        set type dynamic
        set sdn "azure1"
        set filter "K8S_Cluster=zhmKC"
    next
end
```

3. Confirm that the Azure SDN connector resolves dynamic firewall IP addresses using the configured filter:

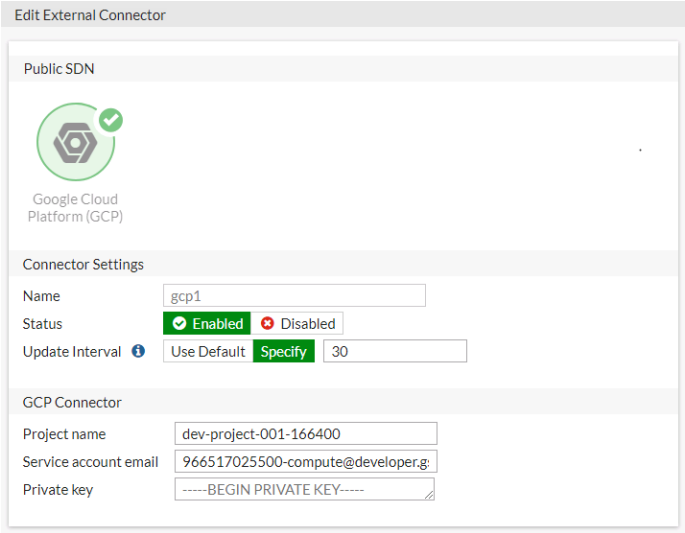
```
config firewall address
    edit "az-k8s-cluster"
        set type dynamic
        set sdn "azure1"
        set filter "K8S_Cluster=zhmKC"
        config list
            edit "10.240.0.4"
            next
            edit "10.240.0.5"
            next
            edit "10.244.0.10"
            next
        end
    next
end
```

GCP Kubernetes (GKE) SDN connector using service account

Google Cloud Platform (GCP) SDN connectors support dynamic address groups based on GCP Kubernetes Engine (GKE) filters.

To enable a GCP SDN connector to fetch IP addresses from GKE:

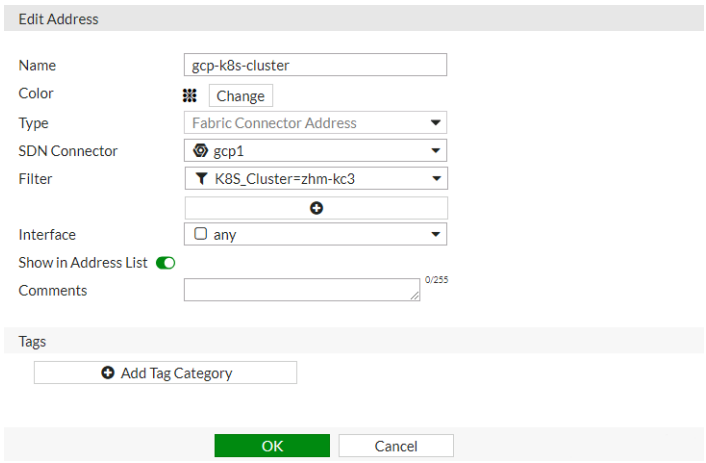
1. Go to *Security Fabric* > *External Connectors*, and configure an SDN connector for GCP.



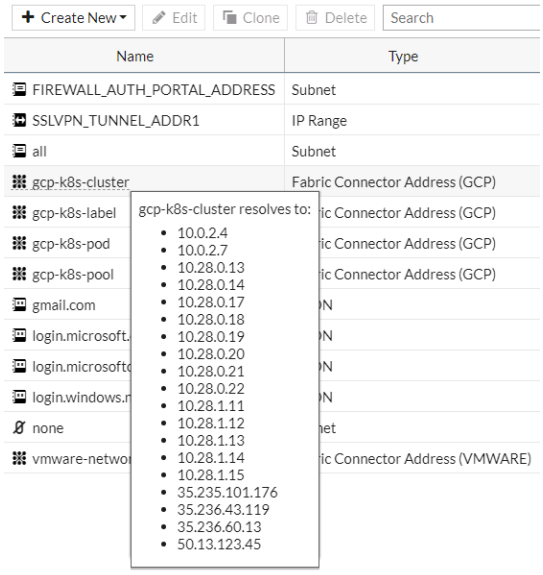
2. Go to *Policies & Objects* > *Addresses* and create a dynamic firewall address for the configured SDN connector using the supported Kubernetes filter.
3. To filter out the Kubernetes IP addresses, select the address filter or filters. The following filters are supported:

Filter	Description
k8s_cluster	Name of Kubernetes cluster.
k8s_namespace	Namespace of a Kubernetes service or pod.
k8s_svcname	Name of a Kubernetes service.
k8s_nodename	Name of a Kubernetes node.
k8s_zone	Zone of a Kubernetes node.
k8s_region	Region of a Kubernetes node.
k8s_podname	Name of a Kubernetes pod.
k8s_label.xxx	Name of label of a Kubernetes resource (cluster/service/node/Pod).

In this example, the GCP SDN connector will automatically populate and update IP addresses only for instances that belong to the zhm-kc3 cluster:



4. Configure the rest of the settings, then click *OK*.
   The dynamic firewall address IP is resolved by the SDN connector.



To configure a GCP Kubernetes SDN connector through the CLI:

1. Configure an SDN connector for Kubernetes:

```
config system sdn-connector
    edit "gcp1"
        set type gcp
        set gcp-project "dev-project-001-166400"
        set service-account "966517025500-compute@developer.gserviceaccount.com"
        set update-interval 30
    next
end
```

2. Create a dynamic firewall address for the SDN connector with a supported Kubernetes filter:

```
config firewall address
    edit "gcp-k8s-cluster"
        set type dynamic
        set sdn "gcp1"
        set filter "K8S_Cluster=zhm-kc3"
    next
end
```

The dynamic firewall address IP is resolved by the SDN connector:

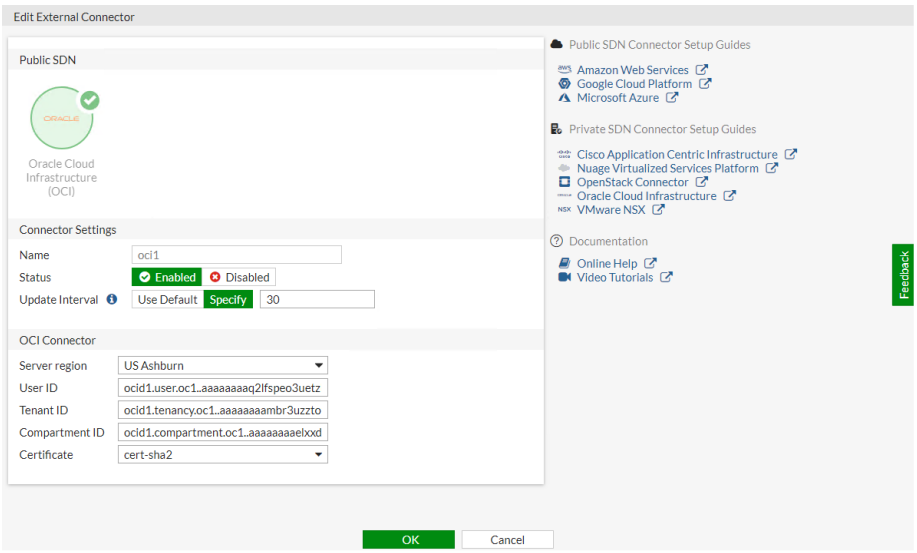
```
config firewall address
    edit "gcp-k8s-cluster"
        set type dynamic
        set sdn "gcp1"
        set filter "K8S_Cluster=zhm-kc3"
        config list
            edit "10.0.2.4"
            next
            edit "10.0.2.7"
            next
            edit "10.28.0.13"
            next
        end
    next
end
```

Oracle Kubernetes (OKE) SDN connector using certificates

OCI SDN connectors support dynamic address groups based on Oracle Kubernetes (OKE) filters.

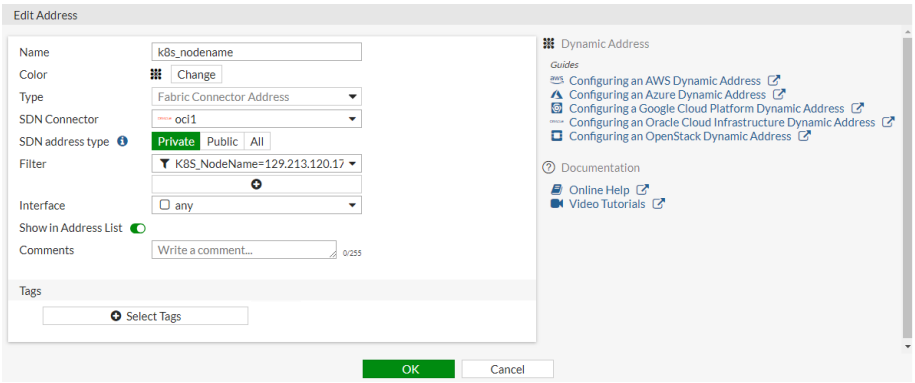
To enable an OCI SDN connector to fetch IP addresses from Oracle Kubernetes:

1. Configure the OCI SDN connector:
   a. Go to *Security Fabric > External Connectors*.
   b. Click *Create New*, and select *Oracle Cloud Infrastructure (OCI)*.
   c. Configure as shown substituting the region, tenant and client IDs, and client secret for your deployment. The update interval is in seconds.



2. Create dynamic firewall addresses for the configured SDN connector with supported Kubernetes filter:
   a. Go to *Policy & Objects* > *Addresses* and select *Address*.
   b. Click *Create new*.
   c. In the *Filter* field, select the desired filters. The following filters are supported:

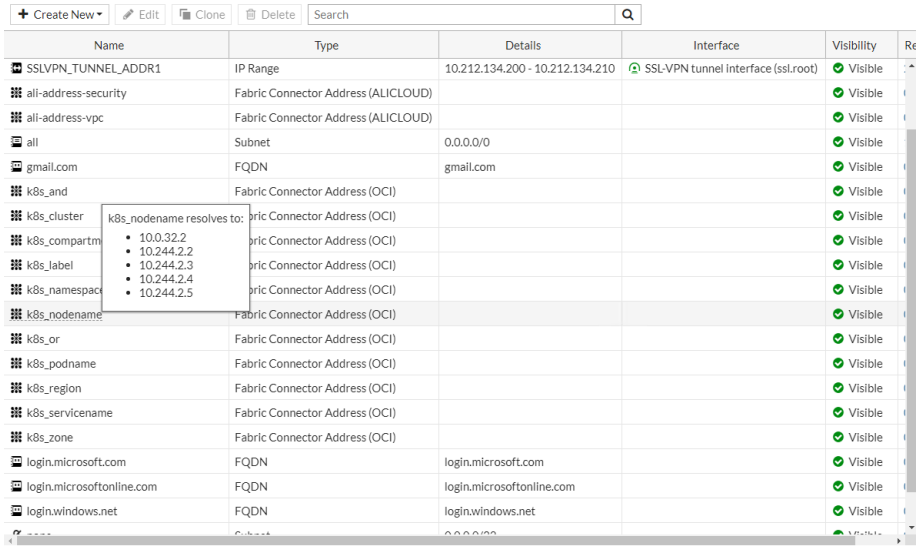
Filter	Description
k8s_cluster	Name of Kubernetes cluster.
k8s_namespace	Namespace of a Kubernetes service or pod.
k8s_svcname	Name of a Kubernetes service.
k8s_nodename	Name of a Kubernetes node.
k8s_zone	Zone of a Kubernetes node.
k8s_region	Region of a Kubernetes node.
k8s_podname	Name of a Kubernetes pod.
k8s_label.xxx	Name of label of a Kubernetes resource (cluster/service/node/Pod).



3. Confirm that the SDN connector resolves dynamic firewall IP addresses:

a. Go to *Policy & Objects* > *Addresses*.

b. Hover over the address created in step 2 to see a list of IP addresses for instances:



To configure an SDN connector through the CLI:

1. Configure the OCI SDN connector:

```
config system sdn-connector
    edit "oci1"
        set type oci
        set tenant-id
"ocid1.tenancy.oc1..aaaaaaaambr3uzztoyhweohbzqqdo775h7d3t54zpmzkp4b2cf35vs55cxxx"
        set user-id
"ocid1.user.oc1..aaaaaaaaq2lfspeo3uetzbzpiv2pqvzzevozccnys347stwssvizqlatfxxx"
        set compartment-id
"ocid1.compartment.oc1..aaaaaaaaelxxdjazqo7nzczpgypyiqcgkmytjry6nfq5345vw7eavpwnmxxx"
        set oci-region ashburn
        set oci-cert "cert-sha2"
        set update-interval 30
    next
end
```

2. Create dynamic firewall addresses for the configured SDN connector with supported Kubernetes filter:

```
config firewall address
    edit "k8s_nodename"
        set type dynamic
        set sdn "oci1"
        set filter "K8S_NodeName=129.213.120.172"
    next
end
```

3. Confirm that the SDN connector resolves dynamic firewall IP addresses:

```
config firewall address
    edit "k8s_nodename"
        set type dynamic
        set sdn "oci1"
        set filter "K8S_NodeName=129.213.120.172"
        config list
            edit "10.0.32.2"
            next
            edit "10.244.2.2"
            next
            edit "10.244.2.3"
            next
            edit "10.244.2.4"
            next
            edit "10.244.2.5"
            next
        end
    next
end
```

Private cloud K8s SDN connector using secret token

FortiOS automatically updates dynamic and cluster IP addresses for Kubernetes (K8s) by using a K8s SDN connector, enabling FortiOS to manage K8s pods as global address objects, as with other connectors. This includes mapping the following attributes from K8s instances to dynamic address groups in FortiOS:

Filter	Description
Namespace	Filter service IP addresses in a given namespace.
ServiceName	Filter service IP addresses by the given service name.
NodeName	Filter node IP addresses by the given node name.
PodName	Filter IP addresses by the pod name.
Label.XXX	Filter service or node IP addresses with the given label XXX. For example: `K8S_Label.app=nginx`.

FortiOS 6.2.3 and later collect cluster IP addresses in addition to external IP addresses for exposed K8s services.

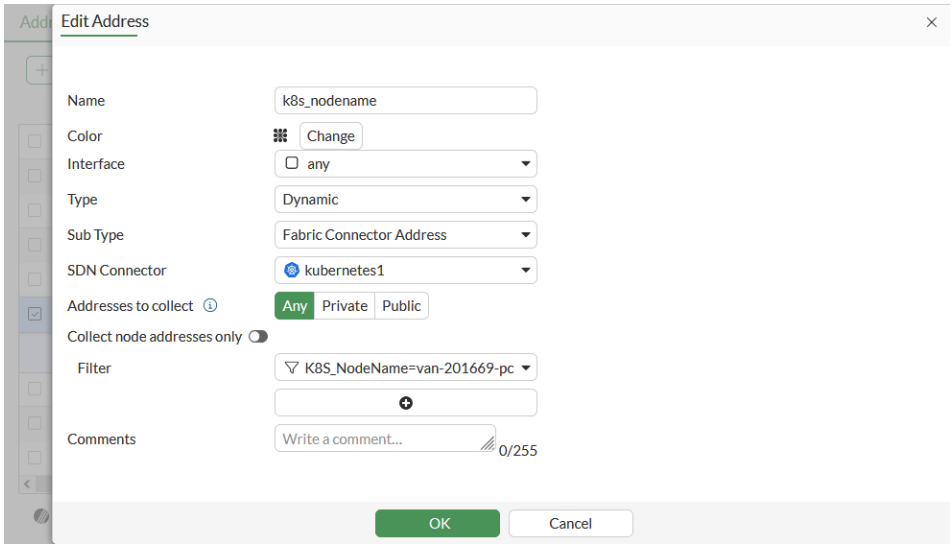
There is no maximum limit for the number of IP addresses populated with the filters.

To configure K8s SDN connector using the GUI:

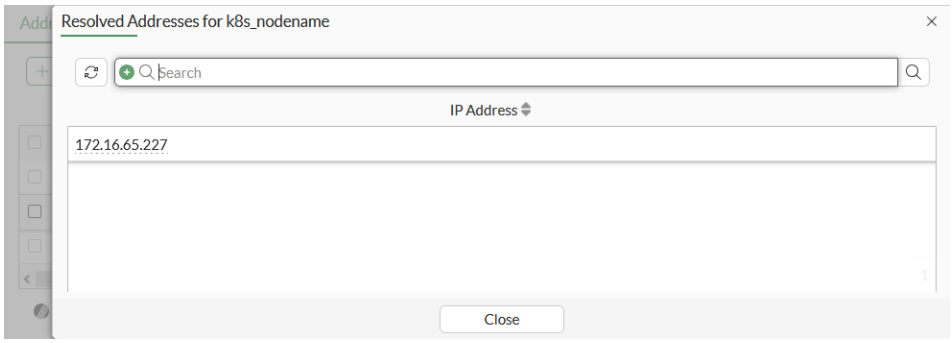
1. Configure the K8s SDN connector:
   a. Go to *Security Fabric* > *External Connectors* and click *Create New*.
   b. Select *Kubernetes*.
   c. Enter a name for the connector.

d. In the *IP* field, enter the IP address that you obtained in Obtaining the IP address, port, and secret token in Kubernetes.
e. In the *Port* field, select *Specify*, then enter the port that you obtained in Obtaining the IP address, port, and secret token in Kubernetes.
f. In the *Secret token* field, enter the token that you obtained in Obtaining the IP address, port, and secret token in Kubernetes.
g. Configure the remaining fields as needed.

2. Create a dynamic firewall address for the configured K8s SDN connector:
   a. Go to *Policy & Objects* > *Addresses* and, on the *Address* tab, click *Create new*.
   b. Set the address name, and configure the interface.
   c. Set *Type* to *Dynamic* and *Sub Type* to *Fabric Connector Address*.
   d. Select the required filter from the *Filter* dropdown list. In this example, the K8s SDN connector will automatically populate and update IP addresses only for node instances that match the specified node name:



3. Ensure that the K8s SDN connector resolves dynamic firewall IP addresses:
   a. Go to *Policy & Objects* > *Addresses*.
   b. Hover over the just created address and click *View Matched Addresses* in the pop up to see a list of IP addresses for node instances that match the node name:



To configure K8s SDN connector using CLI commands:

1. Configure the K8s SDN connector:
   See Obtaining the IP address, port, and secret token in Kubernetes.

```
config system sdn-connector
    edit "kubernetes1"
        set type kubernetes
        set server <IP address>
        set server-port <port>
        set secret-token <secret token>
        set update-interval 30
    next
end
```

2. Create a dynamic firewall address for the configured K8s SDN connector with the supported K8s filter. In this example, the K8s SDN connector will automatically populate and update IP addresses only for node instances that match the specified node name:

```
config firewall address
    edit "k8s_nodename"
        set type dynamic
        set sdn "kubernetes1"
        set filter "K8S_NodeName=van-201669-pc1"
    next
end
```

3. Confirm that the K8s SDN connector resolves dynamic firewall IP addresses using the configured filter:

```
config firewall address
    edit "k8s_nodename"
        set type dynamic
        set sdn "kubernetes1"
        set filter "K8S_NodeName=van-201669-pc1"
        config list
            edit "172.16.65.227"
            next
        end
    next
end
```

To troubleshoot the connection:

1. In FortiOS, run the following commands:

```
diagnose deb application kubed -1
diagnose debug enable
```

2. Reset the connection on the web UI to generate logs and troubleshoot the issue. The following shows the output in the case of a failure:

```
fortigate # diagnose deb application kubed -1
Debug messages will be on for 30 minutes.

fortigate # diagnose debug enable

fortigate # k8s: update sdn connector kubernetes1 status to enabled
k8s: update sdn connector kubernetes2 status to disabled
kubed sdn connector kubernetes1 prepare to update
getting token
kubed sdn connector kubernetes1 start updating
kube url: https://172.17.215.10:6443/api/v1/services
kube host: 172.17.215.10:6443:172.17.215.10
{"kind":"Status","apiVersion":"v1","metadata":{},"status":"Failure","message":"s
ervices is forbidden: User \"system:serviceaccount:default:fortigateconnector\"
cannot list resource \"services\" in API group \"\" at the cluster scope","reaso
n":"Forbidden","details":{"kind":"services"},"code":403}

kubed failed to list kubernetes services.
kubed failed to get IPs from kubedrnetes services.
kubed failed to get ip addr list
kubed reap child pid: 1226
```

The following shows the output in the case of a success:

```
 kube-system
k8s pod ip: 10.180.1.2, podname: metrics-server-v0.3.6-64655c969-djt8s, namespac
e: kube-system
k8s pod ip: 10.138.0.6, podname: netd-4qvvn, namespace: kube-system
k8s pod ip: 10.138.0.5, podname: netd-756ch, namespace: kube-system
k8s pod ip: 10.138.0.4, podname: netd-hr75d, namespace: kube-system
k8s pod ip: 10.138.0.6, podname: prometheus-to-sd-59trp, namespace: kube-system
k8s pod ip: 10.138.0.4, podname: prometheus-to-sd-q6qv5, namespace: kube-system
k8s pod ip: 10.138.0.5, podname: prometheus-to-sd-rq2zm, namespace: kube-system
k8s pod ip: 10.180.1.3, podname: stackdriver-metadata-agent-cluster-level-6c4f64
f8cc-zgnp5, namespace: kube-system
k8s pod ip: 10.180.0.3, podname: nginx-deployment-c68885cbb-sf6f5, namespace: ng
inx
k8s pod ip: 10.180.1.4, podname: nginx-deployment-c68885cbb-w5w2b, namespace: ng
inx
kubed get IP address list from Kubernetes:
kubed sdn connector kubernetes2 start updating IP addresses
kubed checking firewall address object gcp-address, vd 0
 address num change 0/3, new ip list:
  10.180.0.3
  10.180.1.4
  10.184.0.1
kubed sdn connector kubernetes2 finish updating IP addresses
kubed reap child pid: 1252
```

Multus CNI for Kubernetes connectors

FortiGate now supports Multus CNI (container network interface) for Kubernetes connectors, ensuring that all IP addresses, including those configured at runtime, are accurately retrieved and added to Firewall address dynamic objects, enhancing network security integration.

Prerequisites

To enable FortiGate to integrate with a Kubernetes cluster using Multus CNI, the following prerequisites must be met:

Kubernetes cluster setup:

- A Kubernetes cluster must be deployed and operational.
- The cluster must have Multus CNI configured to support multiple network interfaces per pod.

Pod deployment with required annotations:

A pod must be deployed with the appropriate Multus CNI annotations to expose both IPv4 and IPv6 addresses. Ensure the following:

1. k8s.v1.cni.cncf.io/network-status
   This annotation must include a default network interface:

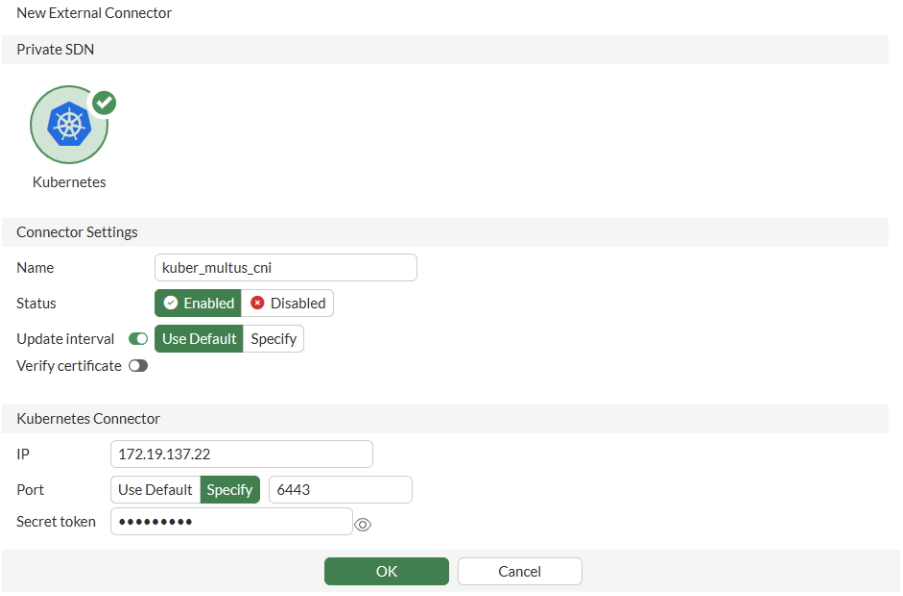
- IPv4 address: **10.244.0.11**
- Interface name: such as **eth0**

2. k8s.v1.cni.cncf.io/network-extra-info
   This annotation must define two additional networks:
   - Network 1:
     - default/bridge-net-static
     - IPv4 address: 10.244.1.111
     - IPv6 address: 2001:0db8:85a3:0000:0000:8a2e:0370:0033
   - Network 2:
     - default/bridge-net-static-2
     - IPv4 address: 10.244.2.122
     - IPv6 address: 2001:db8:85a3::8a2e:370:7334

FortiGate configuration

To configure a Kubernetes SDN connector in the GUI:

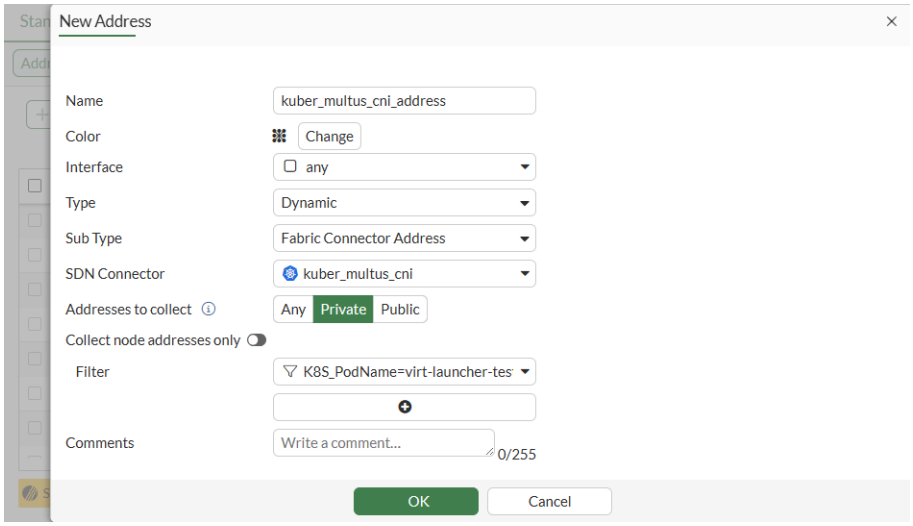
1. Go to *Security Fabric* > *External Connectors* and click *Create New*.
2. Select *Kubernetes*.
3. Set *IP* to *172.19.137.22*.
4. Set *Port* to *Specify*, then enter the port number *6443*.
5. In the *Secret token* field, enter the token.



   For more information, see Private cloud K8s SDN connector using secret token on page 4644 and Obtaining the IP address, port, and secret token in Kubernetes.
6. Configure the other fields as needed, then click *OK*.
7. Hover over the connector and ensure that the *Status* is *Up*.

To create dynamic firewall addresses associated with the SDN connector in the GUI:

1. Go to *Policy & Objects* > *Addresses*, and on the *Standard* tab, select *Address* then click *Create New*.
2. Configure the address:

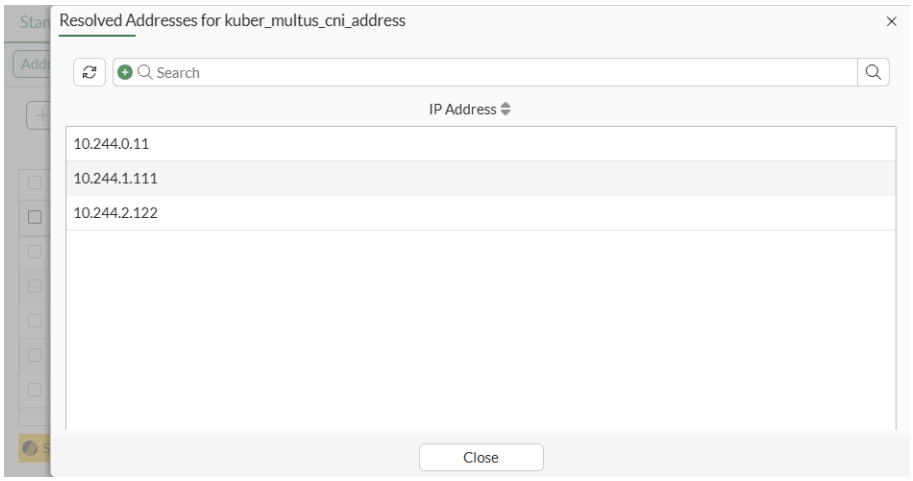


Field	Value
Name	kuber_multus_cni_address
Interface	any
Type	Dynamic
Sub Type	Fabric Connector Address
SDN Connector	kuber_multi_cni
Addresses to collect	Private
Filter	K8S_PodName=virt-launcher-testvm3-7q5gl

3. Click *OK*.
   IPv6 dynamic addresses can only be configure in the CLI.

To ensure that the connector resolves dynamic firewall IP addresses in the GUI:

1. Go to *Policy & Objects* > *Addresses*.
2. Hover over the *kuber_multus_cni_address* address and click *View Matched Addresses* in the pop-up.
   The resolved addresses are shown. They are the same as the IP addresses configured in the Kubernetes cluster.



You can also view the matched addresses by edit the addresses and clicking *View Matched Addresses* in the right side bar.

To configure a Kubernetes SDN connector in the CLI:

```
config system sdn-connector
    edit "kuber_multus_cni"
        set type kubernetes
        set verify-certificate disable
        set server "172.19.137.22"
        set server-port 6443
        set secret-token xxxxxx
    next
end
```

Confirm that the connector is up:

```
# diagnose sys sdn status "kuber_multus_cni"
SDN Connector                       Type        Status
------------------------------------------------------------
kuber_multus_cni                    kubernetes  Up
```

To create dynamic firewall addresses associated with the SDN connector in the CLI:

```
config firewall address
    edit "kuber_multus_cni_address"
        set type dynamic
        set sdn "kuber_multus_cni"
        set filter "K8S_PodName=virt-launcher-testvm3-7q5gl"
    next
end
```

```
config firewall address6
    edit "kuber_multus_cni_address6"
        set type dynamic
        set sdn "kuber_multus_cni"
```

```
        set filter "K8S_PodName=virt-launcher-testvm3-7q5gl"
    next
end
```

To ensure that the connector resolves dynamic firewall IP addresses:

```
# show firewall address "kuber_multus_cni_address"
config firewall address
    edit "kuber_multus_cni_address"
        set uuid 78cff8c2-53a7-51f0-ced2-eec27f839f7b
        set type dynamic
        set sdn "kuber_multus_cni"
        set color 19
        set filter "K8S_PodName=virt-launcher-testvm3-7q5gl"
        config list
            edit "10.244.0.11"
            next
            edit "10.244.1.111"
            next
            edit "10.244.2.122"
            next
        end
    next
end
```

```
# show firewall address6 "kuber_multus_cni_address6"
config firewall address6
    edit "kuber_multus_cni_address6"
        set uuid 40f105f0-53ab-51f0-d04d-21cbd699b4d9
        set type dynamic
        set sdn "kuber_multus_cni"
        set filter "K8S_PodName=virt-launcher-testvm3-7q5gl"
        config list
            edit "2001:db8:85a3::8a2e:370:33"
            next
            edit "2001:db8:85a3::8a2e:370:7334"
            next
        end
    next
end
```

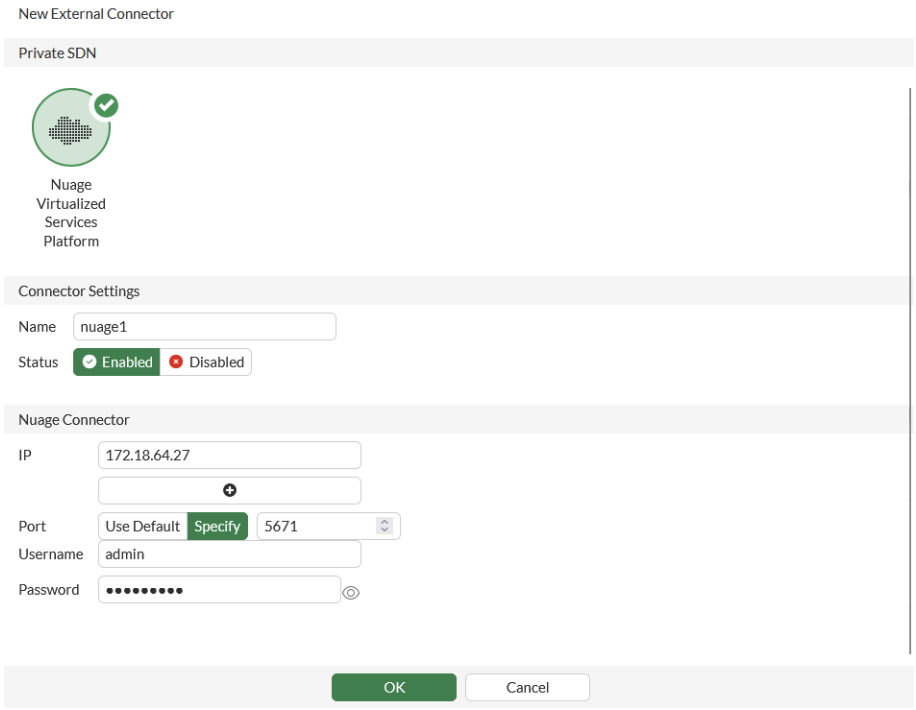
Nuage SDN connector using server credentials

You can use Nuage SDN connectors in dynamic firewall addresses.

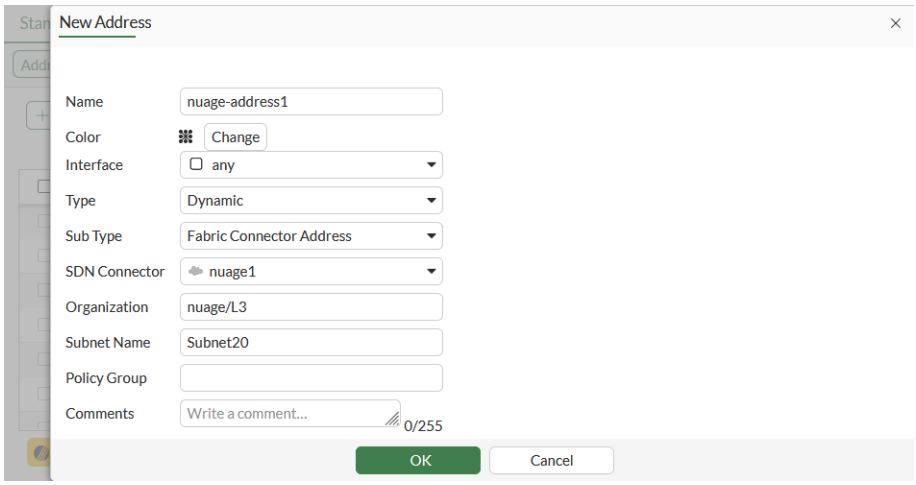
The Fortinet SDN Connector for Cisco ACI and Nuage Networks is a standalone connector that connects to SDN controllers within Cisco ACI and Nuage Networks. You must configure a connection to the Fortinet SDN connector in FortiOS to query the dynamic addresses.

To configure a Nuage connector in the GUI:

1. Create the Nuage SDN connector:
   a. Go to *Security Fabric* > *External Connectors* and click *Create New*.
   b. In the *Private SDN* section, click *Nuage Virtualized Services Platform*.
   c. Configure the settings as needed.
   d. Click *OK*.



2. Create the dynamic firewall address for the connector:
   a. Go to *Policy & Objects* > *Addresses* and select *Address*.
   b. Click *Create new*.
   c. Configure the following settings:
      i. For *Type*, select *Dynamic*.
      ii. For *Sub Type*, select *Fabric Connector Address*.
      iii. For *SDN Connector*, select the Nuage connector.
      iv. Configure the remaining settings as needed.
   d. Click *OK*.



To verify the SDN connector resolves the dynamic firewall IP addresses in the GUI:

1. Go to *Policy & Objects* > *Addresses*.
2. In the address table, hover over an address to view which IP addresses it resolves to.

To configure a Nuage connector in the CLI:

1. Create the SDN connector:

```
config system sdn-connector
    edit "nuage1"
        set type nuage
        set server "172.18.64.27"
        set server-port 5671
        set username "admin"
        set password xxxxxxx
    next
end
```

2. Create the dynamic firewall address for the connector:

```
config firewall address
    edit "nuage-address1"
        set type dynamic
        set sdn "nuage1"
        set color 19
        set organization "nuage/L3"
        set subnet-name "Subnet20"
    next
end
```

To verify the SDN connector resolves the dynamic firewall IP addresses in the CLI:

```
# diagnose firewall dynamic list
```

```
List all dynamic addresses:
nuage1.nuage.nuage/L3.Subnet20.*: ID(196)
        ADDR(192.168.20.92)
        ADDR(192.168.20.240)
```

Nutanix SDN connector using server credentials

FortiOS automatically updates dynamic addresses for Nutanix using an Nutanix SDN connector, including mapping the following attributes from Nutanix instances to dynamic address groups in FortiOS:

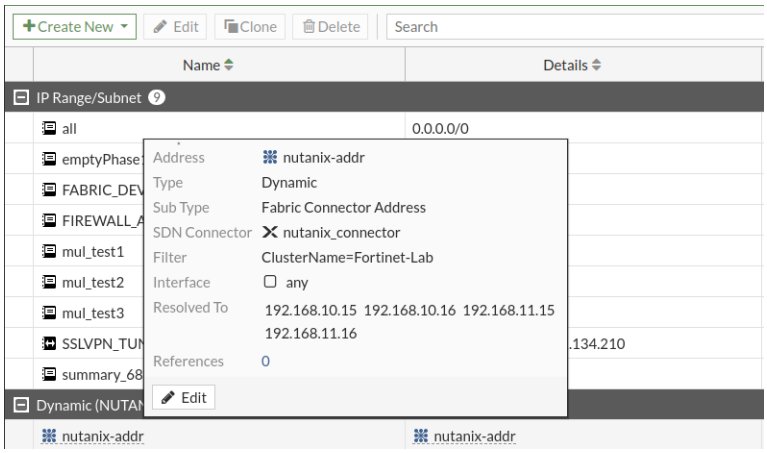
- Cluster name
- Cluster UUID
- Description
- Host name
- Host UUID
- Hypervisor type
- Image name
- Image UUID
- Subnet name
- Subnet UUID
- VM name
- VM UUID

To configure a Nutanix connector using the GUI:

1. Configure the Nutanix SDN connector:
   a. Go to *Security Fabric > External Connectors*.
   b. Select *Nutanix*.
   c. In the *IP address* field, enter the IP address for your Nutanix environment.
   d. In the *Port* field, enter the desired port.
   e. In the *Username* and *Password* fields, enter the credentials for your Nutanix environment.
   f. Click *OK*.
2. Create a dynamic firewall address for the configured Nutanix SDN connector:
   a. Go to *Policy & Objects > Addresses* and select *Address*.
   b. Click *Create new*.
   c. From the *Type* dropdown list, select *Dynamic*.
   d. From the *Sub Type* dropdown list, select *Fabric Connector Address*.
   e. From the *SDN Connector* dropdown list, select the Nutanix connector.
   f. From the *Filter* dropdown list, select the desired filters.
   g. Click *OK*.
3. Ensure that the Nutanix SDN connector resolves dynamic firewall IP addresses:

a. Go to *Policy & Objects* > *Addresses*.

b. Hover over the address created in step 2 to see a list of IP addresses for instances that satisfy the filter requirements configured in step 2. In this example, the configured filter is "ClusterName=Fortinet-Lab":



To configure a Nutanix connector using the CLI:

1. Configure the Nutanix SDN connector:

```
config system sdn-connector
    edit "nutanix_connector"
        set status disable
        set type nutanix
        set server "172.18.33.59"
        set server-port 9440
        set username "admin"
        set password **********
        set update-interval 60
    next
end
```

2. Create a dynamic firewall address for the configured Nutanix SDN connector:

```
config firewall address
    edit "nutanix-addr"
        set uuid 382ceafe-8e72-51eb-7300-0807ee907946
        set type dynamic
        set sdn "nutanix_connector"
        set color 2
        set filter "ClusterName=Fortinet-Lab"
    next
end
```

3. Ensure that the Nutanix SDN connector resolves dynamic firewall IP addresses:

```
config firewall address
    edit "nutanix-addr"
        set uuid 382ceafe-8e72-51eb-7300-0807ee907946
```

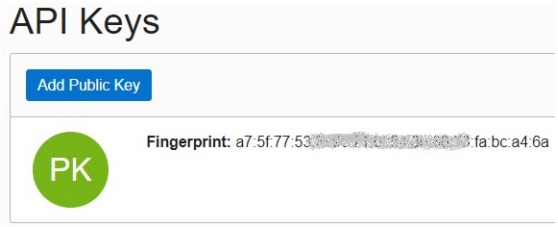
```
        set type dynamic
        set sdn "nutanix_connector"
        set color 2
        set filter "ClusterName=Fortinet-Lab"
        config list
            edit "192.168.10.15"
            next
            edit "192.168.10.16"
            next
            edit "192.168.11.15"
            next
            edit "192.168.11.16"
            next
        end
    next
end
```

OCI SDN connector using certificates

When configuring SDN connector integration with Oracle Cloud Infrastructure (OCI), administrators have the option to use certificate-based authentication instead of using account information to authenticate. This example shows how to use FortiGate's default Fortinet_Factory certificate to authenticate against OCI. Subsequently, after the connection is made, administrators can use addresses dynamically retrieved from OCI on the FortiGate.

To configure an OCI SDN connector in the GUI:

1. Go to *Security Fabric* > *External Connectors* and click *Create New*.
2. In the *Public SDN* section, select *Oracle Cloud Infrastructure (OCI)*.
3. Configure the connector:
   a. *User ID*: Enter the OCID of the OCI user who belongs to the administrator group. See Certificate-based SDN connector requirements.
   b. For the *OCI Certificate* field, you must select a certificate that satisfies OCI key size limits. The minimum size is 2048 bits. Do one of the following:
      i. Select the built-in default certificate called Fortinet_Factory.
      ii. Follow steps 1-2 in Using custom certificates to configure a custom certificate.
4. Click *OK*.
5. At this stage, you must register the certificate's fingerprint to the specified OCI user.
   a. Go to the OCI user, then *API Keys* > *Add Public Key*.
   b. If you previously selected the Fortinet_Factory certificate, do the following:
      i. In FortiOS, go to *System* > *Certificate*. Select Fortinet_Factory, then click *Download*.
      ii. You now have the Fortinet_Factory.cer file. Create a public key file in PEM format from it, using a freely available tool of your choice such as OpenSSL.
   c. Copy and paste the content of the certificate PEM key file in the *Add Public Key* window in OCI. Click *Add*.
   d. You now see the fingerprint.



You can configure the following for the fingerprint:

i. *Update Interval*: The default value is 60 seconds. You can change the value to between 1 and 3600 seconds.

ii. *Status*: Green means that the connector is enabled. You can disable it at any time by toggling the switch.

e. Click *OK*.

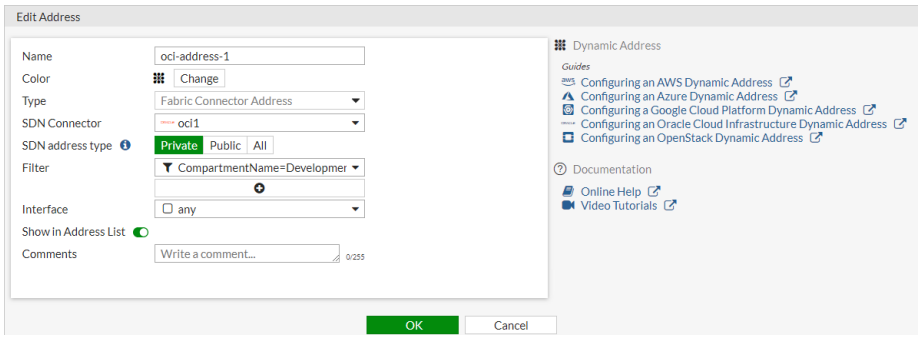
6. Go to *Policy & Objects* > *Addresses* and select *Address*.
7. Click *Create new*.
8. Configure the address, selecting the OCI connector in the *SDN Connector* field. The following filters are supported:

   `'vm_name=<vm name>'`: matches VM instance name.

   `'instance_id=<instance id>'`: matches instance OCID.

   `'tag.<key>=<value>'`: matches freeform tag key and its value.

   `'definedtag.<namespace>.<key>=<value>'`: matches a tag namespace, tag key, and its value.



9. Click *OK*.

To configure an OCI SDN connector in the CLI:

1. Configure an SDN connector:

```
config system sdn-connector
    edit "oci1"
        set status enable
        set type oci
        set tenant-id
"ocid1.tenancy.oc1..aaaaaaaaaaa3aaaaaaaaaaaaaaaaa77xxxxxx54bbbbbb4xxxx35xx55xxxx"
        set user-id
"ocid1.user.oc1..aaaaaaaaa2laaaaa3aaaaaaaaaabbbbbbbbbbcccc3ccccccccccxxxxxxxx"
        set compartment-id
"ocid1.compartment.oc1..aaaaaaaaaaaaaaaaaa7bbbbbbbbbbccccccccc6xxx53xxxx7xxxxxxxxxx"
        set oci-region "us-ashburn-1"
```

```
        set oci-region-type commercial
        set oci-cert "cert-sha2"
        set update-interval 30
    next
end
```

2. Create a dynamic firewall address for the SDN connector with a supported filter:

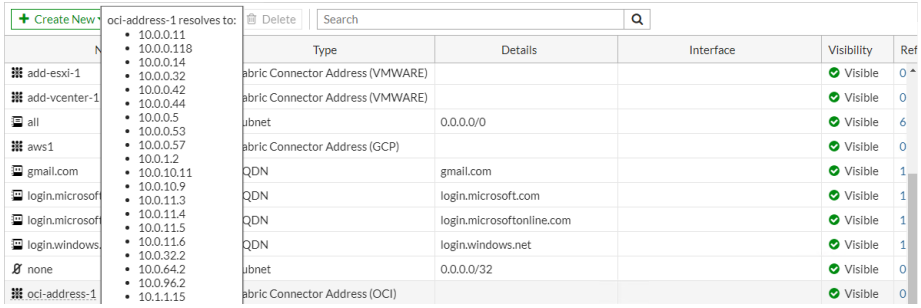
```
config firewall address
    edit "oci-address-1"
        set type dynamic
        set sdn "oci1"
        set filter "CompartmentName=DevelopmentEngineering"
    next
end
```

To confirm that dynamic firewall addresses are resolved by the SDN connector:

1. In the CLI, check that the addresses are listed:

```
config firewall address
    edit "oci-address-1"
        set type dynamic
        set sdn "oci1"
        set filter "CompartmentName=DevelopmentEngineering"
        config list
            edit "10.0.0.11"
            next
            edit "10.0.0.118"
            next
            ...
            next
        end
    next
end
```

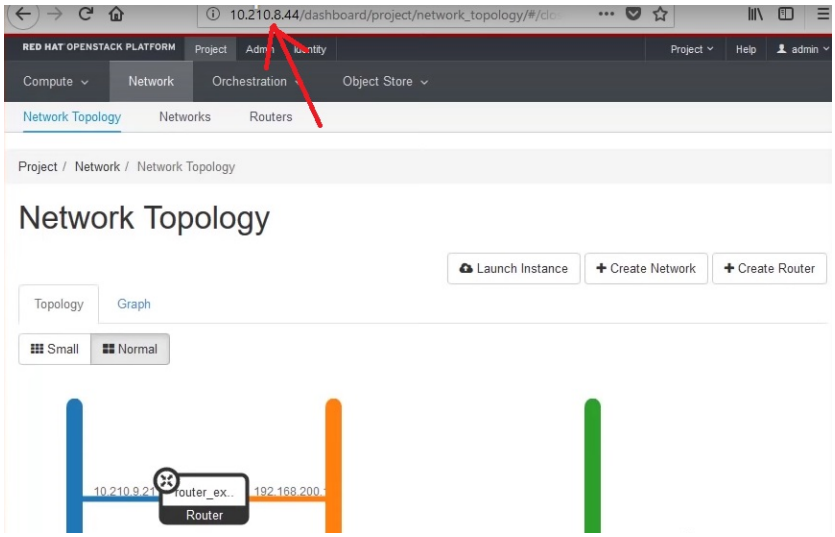
2. In the GUI, go to *Policy & Objects* > *Addresses* and hover the cursor over the address name.



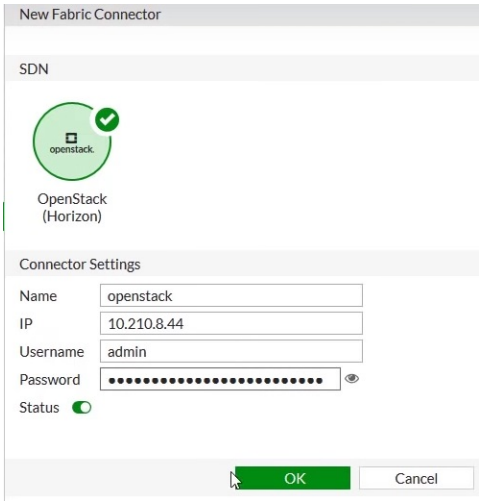
OpenStack SDN connector using node credentials

To configure OpenStack SDN connector using node credentials:

1. Go to *Security Fabric* > *External Connectors*.
2. Click *Create New*, then select *OpenStack (Horizon)*.
3. Configure the fields as follows:
   a. *Name*: Name the connector as desired.
   b. *IP*: Enter the OpenStack management component's IP address. Generally you can find it in the OpenStack identity.



   c. *User name*: Enter the specified node's administrator name.
   d. *Password*: Enter the administrator password.



4. Click *OK*. The SDN connector is now configured.

To configure a dynamic firewall address:

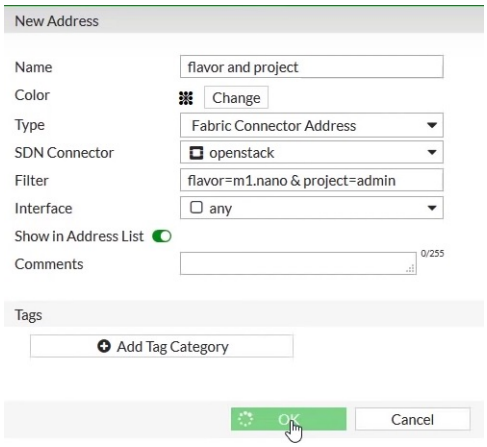
The next step is to create an address that will be used as an address group or single address that acts as the source/destination for firewall policies. The address is based on IP addresses and contains VM instances' IP addresses.

No matter what changes occur to the instances, the SDN connector populates and updates the changes automatically based on the specified filtering condition so that administrators do not need to reconfigure the address content manually. Appropriate firewall policies using the address are applied to instances that are members of the address.

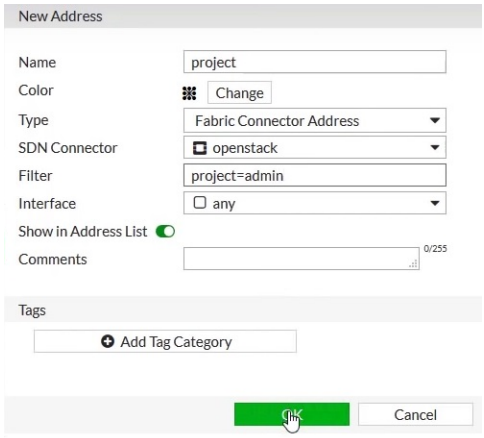
1. Go to *Policy & Objects* > *Address*. Click *Create New*, then select *Address*.
2. Configure the address as follows:
   a. *Name*: Name the address as desired.
   b. *Type*: Select *Dynamic*.
   c. *Sub Type*: Select *Fabric Connector Address*.
   d. *SDN Connector*: Select *openstack*.
   e. *Filter*: The SDN connector automatically populates and updates only IP addresses belonging to the specified filter that matches the condition. OpenStack Horizon connectors support the following filters:
      i. `id=<instance id>`: This matches a VM instance ID.
      ii. name=<instance name>: This matches a VM instance name.
      iii. `flavor=<instance flavor name>`: This matches an instance flavor name.
      iv. `keypair=<key pair name>`: This matches a key pair name.
      v. `network=<net name>`: This matches a network name.
      vi. `project=<project name>`: This matches a project name.
      vii. `availabilityzone=<zone name>`: This matches an availability zone name.
      viii. `servergroup=<group name>`: This matches a server group name.
      ix. `securitygroup=<security group name>`: This matches a security group name.
      x. `metadata.<key>=<value>`: This matches metadata with its key and value pair.

   You can set filtering conditions using multiple entries with AND ("&") or OR ("|"). When both AND and OR are specified, AND is interpreted first, then OR.

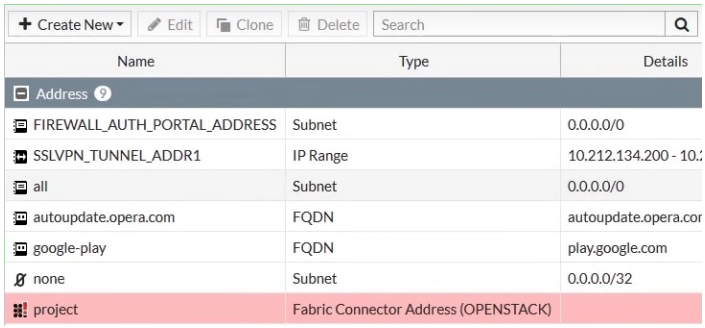
   For example, you could enter `flavor=m1.nano&project=admin`. In this case, IP addresses of instances that match both the flavor name and project name are populated. Wildcards (asterisks) are not allowed in values.



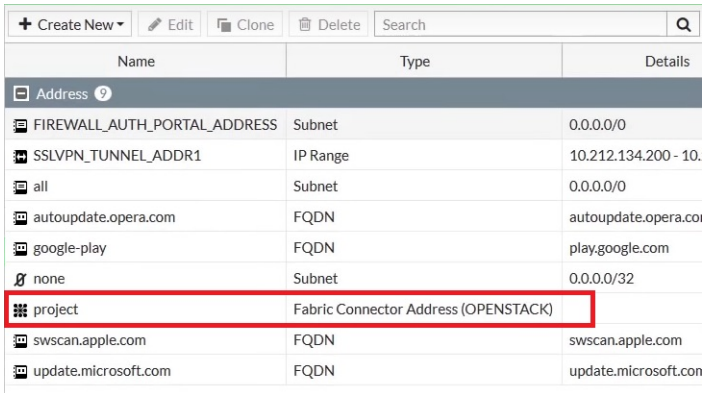
In this example, let's use project=admin, assuming the project name is admin.

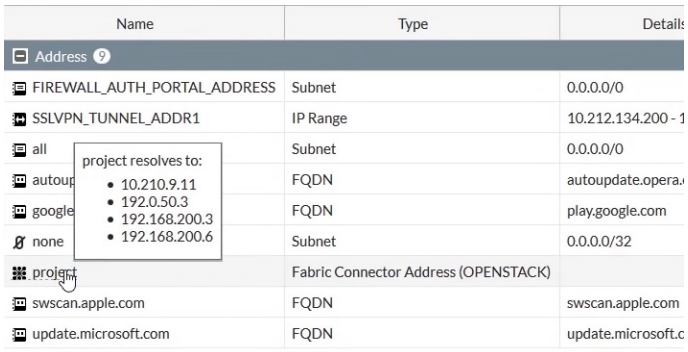


3. Click *OK* after completing all required fields.
4. Ensure that the address was created.



5. After a few minutes, the new address takes effect. Hover your cursor on the address to see a list of IP addresses and instances with the project name "admin".



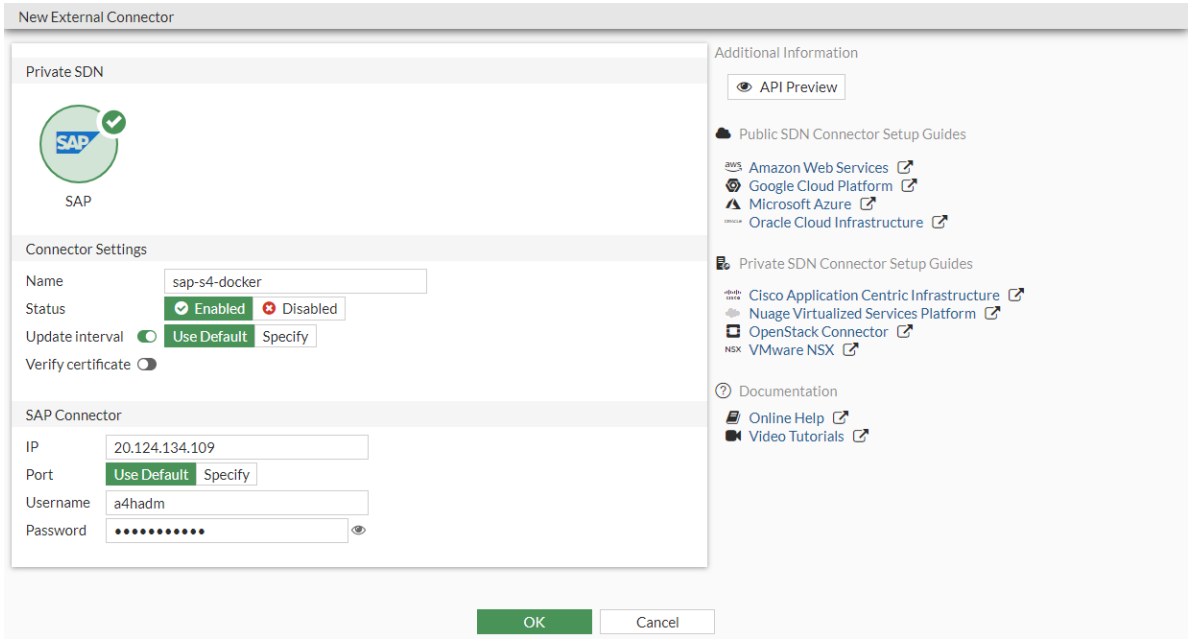


SAP SDN connector

The SAP external Fabric connector allows the FortiGate to connect to an SAP instance to synchronize dynamic address objects and ports for SAP workloads. These address objects can be used in firewall policies to grant access control to dynamic SAP workloads.

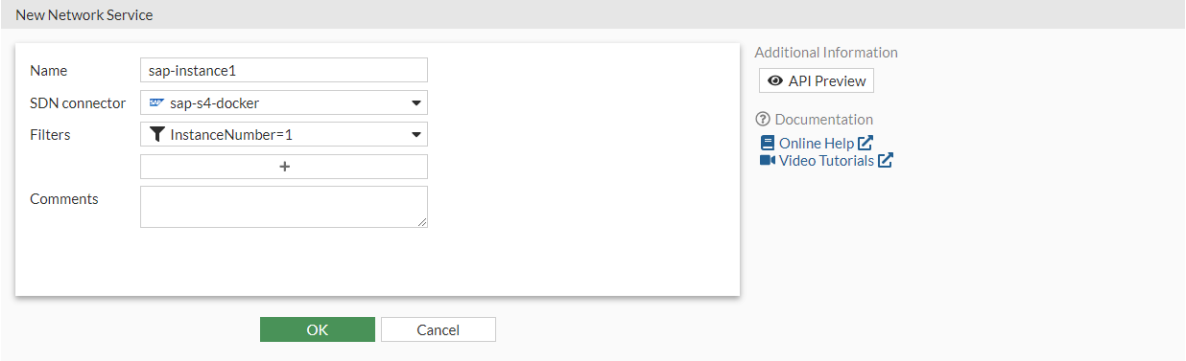
To configure an SAP connector in the GUI:

1. Configure the SAP SDN connector:
   a. Go to *Security Fabric* > *External Connectors* and click *Create New*.
   b. In the *Private SDN* section, select *SAP*.
   c. Enter a *Name* (*sap-s4-docker*).
   d. Enter the *IP* for the SAP instance.
   e. Enter the *Username* and *Password*.



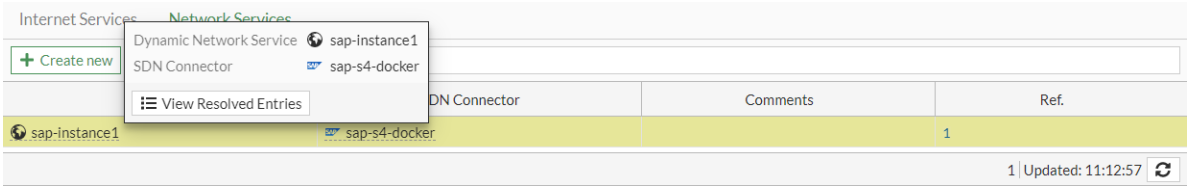
   f. Click *OK*.
2. Configure a network service associated with the configured SAP SDN connector:

a. Go to *Policy & Objects* > *Internet Service Database*, select the *Network Services* tab, and click *Create New*.
b. Enter a *Name* (*sap-instance1*).
c. Set *SDN connector* to *sap-s4-docker*.
d. Select a filter, such as *InstanceNumber=1*. The available filters are for HostName, InstanceNumber, and ServiceName.

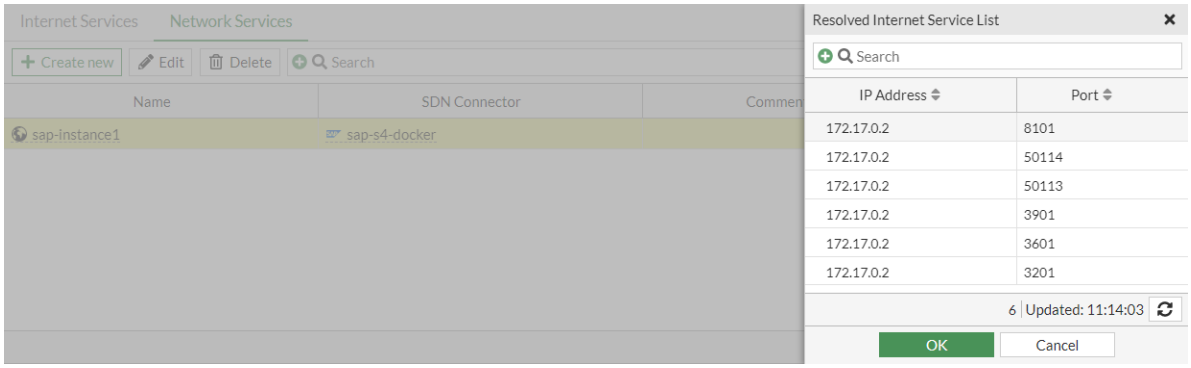


e. Click *OK*.

3. Ensure that the SAP SDN connector resolves dynamic network services:
a. Go to *Policy & Objects* > *Internet Service Database*, select the *Network Services* tab.
b. Hover over the *sap-instance1* and click *View Resolved Entries*.



A list of resolved internet services is displayed.



Click *OK* to close the list.

4. Configure a firewall policy with the resolved dynamic network service as the destination:
a. Go to *Policy & Objects* >*Firewall Policy* and click *Create New*.
b. Set the *Destination* to the *sap-instance1* network service.
c. Configure the other settings as needed.
d. Click *OK*.

To configure an SAP connector in the CLI:

1. Configure the SAP SDN connector:

```
config system sdn-connector
    edit "sap-s4-docker"
        set type sap
        set verify-certificate disable
        set server "20.124.134.109"
        set server-port 50014
        set username "a4hadm"
        set password ************
    next
end
```

2. Configure a network service associated with the configured SAP SDN connector (available filters are HostName, InstanceNumber, and ServiceName):

```
config firewall network-service-dynamic
    edit "sap-instance1"
        set sdn "sap-s4-docker"
        set filter "InstanceNumber=1"
    next
end
```

3. Ensure that the SAP SDN connector resolves dynamic network services:

```
# diagnose firewall network-service-dynamic list "sap-instance1"
List internet service in kernel(custom):
name=sap-instance1 id=4294770689 reputation=0 (null) singularity=0 flags=0x0 protocol=6
port=8101-8101
addr ip range(1): 172.17.0.2-172.17.0.2
name=sap-instance1 id=4294770689 reputation=0 (null) singularity=0 flags=0x0 protocol=6
port=50114-50114
addr ip range(1): 172.17.0.2-172.17.0.2
name=sap-instance1 id=4294770689 reputation=0 (null) singularity=0 flags=0x0 protocol=6
port=50113-50113
addr ip range(1): 172.17.0.2-172.17.0.2
name=sap-instance1 id=4294770689 reputation=0 (null) singularity=0 flags=0x0 protocol=6
port=3901-3901
addr ip range(1): 172.17.0.2-172.17.0.2
name=sap-instance1 id=4294770689 reputation=0 (null) singularity=0 flags=0x0 protocol=6
port=3601-3601
addr ip range(1): 172.17.0.2-172.17.0.2
name=sap-instance1 id=4294770689 reputation=0 (null) singularity=0 flags=0x0 protocol=6
port=3201-3201
addr ip range(1): 172.17.0.2-172.17.0.2
```

4. Configure a firewall policy with the resolved dynamic network service as the destination:

```
config firewall policy
    edit 2
        set name "FGT97-service-dynamic"
```

```
        set srcintf "port3"
        set dstintf "port10"
        set action accept
        set srcaddr "all"
        set internet-service enable
        set network-service-dynamic "sap-instance1"
        set schedule "always"
        set nat enable
    next
end
```

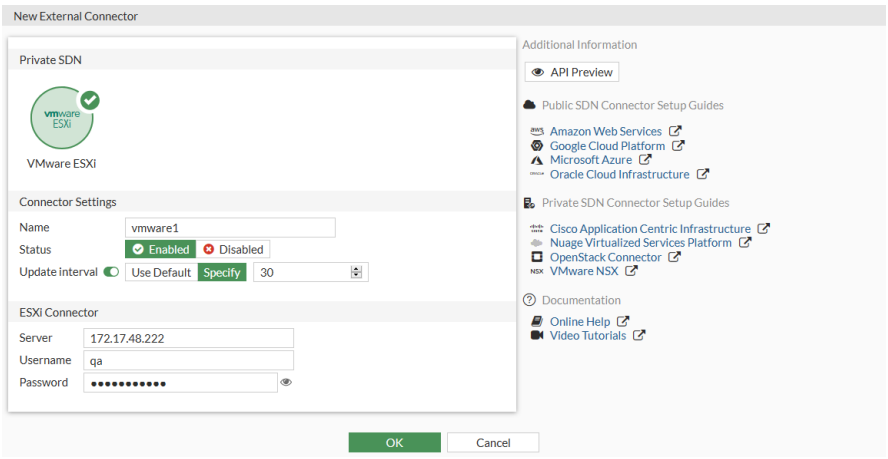
VMware ESXi SDN connector using server credentials

Dynamic addresses for VMware ESXi and vCenter servers can be automatically updated by using a VMware ESXi SDN connector, including mapping the following attributes from VMware ESXi and vCenter objects to dynamic address groups in FortiOS:

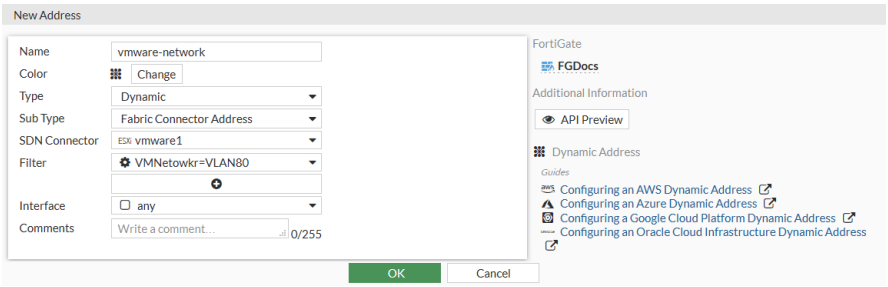
- vmid
- host
- name
- uuid
- vmuuid
- vmnetwork
- guestid
- guestname
- annotation
- datacenter
- tag

To configure VMware ESXi SDN connector using the GUI:

1. Configure the VMware ESXi SDN connector:
   a. Go to *Security Fabric* > *External Connectors*.
   b. Click *Create New*, and select *VMware ESXi*.
   c. Configure as shown, substituting the server IP address, username, and password for your deployment. The update interval is in seconds. The password cannot contain single or double quotes.



2. Create a dynamic firewall address for the configured VMware ESXi SDN connector:
   a. Go to *Policy & Objects* > *Addresses* and select *Address*.
   b. Click *Create new*.
   c. Configure the address as shown, selecting the desired filter in the *Filter* dropdown list. In this example, the VMware ESXi fabric connector will automatically populate and update IP addresses only for instances that belong to VLAN80:



3. Ensure that the VMware ESXi SDN connector resolves dynamic firewall IP addresses:
   a. Go to *Policy & Objects* > *Addresses*.
   b. Hover over the address created in step 2 to see a list of IP addresses for instances that belong to VLAN80 as configured in step 2.

To configure VMware ESXi SDN connector using CLI commands:

1. Configure the VMware ESXi SDN connector:

```
config system sdn-connector
    edit "vmware1"
        set type vmware
        set server "172.17.48.222"
        set username "example_username"
        set password xxxxx
        set update-interval 30
    next
end
```

2. Create a dynamic firewall address for the configured VMware ESXi SDN connector with the supported VMware ESXi filter. In this example, the VMware ESXi SDN connector will automatically populate and update

IP addresses only for instances that belong to the specified VLAN:

```
config firewall address
    edit "vmware-network"
        set type dynamic
        set sdn "vmware1"
        set filter "vmnetwork=VLAN80"
    next
end
```

3. Confirm that the VMware ESXi SDN connector resolves dynamic firewall IP addresses using the configured filter:

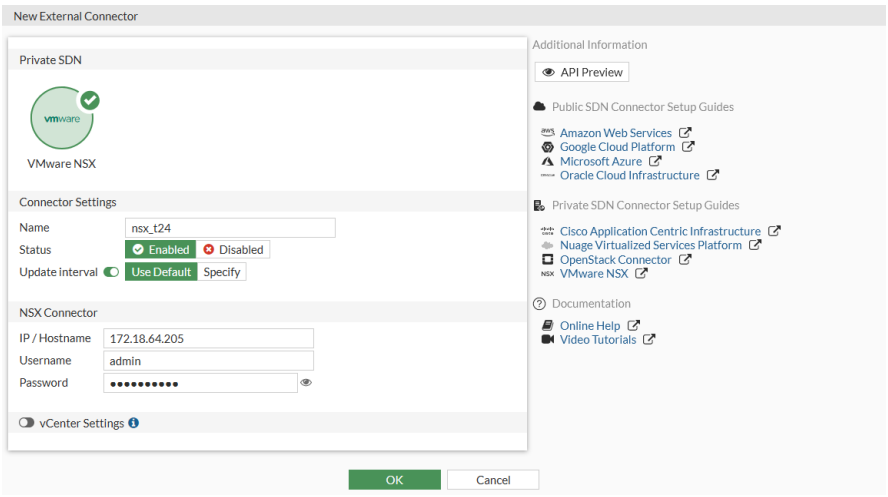
```
config firewall address
    edit "vmware-network"
        set type dynamic
        set sdn "vmware1"
        set filter "vmnetwork=VLAN80"
        config list
            edit "192.168.8.240"
            next
        end
    next
end
```

VMware NSX-T Manager SDN connector using NSX-T Manager credentials

This feature provides SDN connector configuration for VMware NSX-T manager. You can import specific groups, or all groups from the NSX-T Manager.

To configure SDN connector for NSX-T Manager in the GUI:

1. Go to *Security Fabric* > *External Connectors* and click *Create New*.
2. In the *Private SDN* section, click *VMware NSX*.



3. Configure the settings and click *OK*.

To configure SDN connector for NSX-T Manager in the CLI:

```
config system sdn-connector
    edit "nsx_t24"
        set type nsx
        set server "172.18.64.205"
        set username "admin"
        set password xxxxxx
    next
end
```

To import a specific group from the NSX-T Manager:

```
# execute nsx group import nsx_t24 root csf_ns_group
[1] 336914ba-0660-4840-b0f1-9320f5c5ca5e csf_ns_group:
  Name:csf_ns_group
  Address:1.1.1.0
  Address:1.1.1.1
  Address:172.16.10.104
  Address:172.16.20.104
  Address:172.16.30.104
  Address:2.2.2.0
  Address:2.2.2.2
  Address:4.4.4.0
  Address:5.5.5.0
  Address:6.6.6.6
  Address:7.7.7.7
```

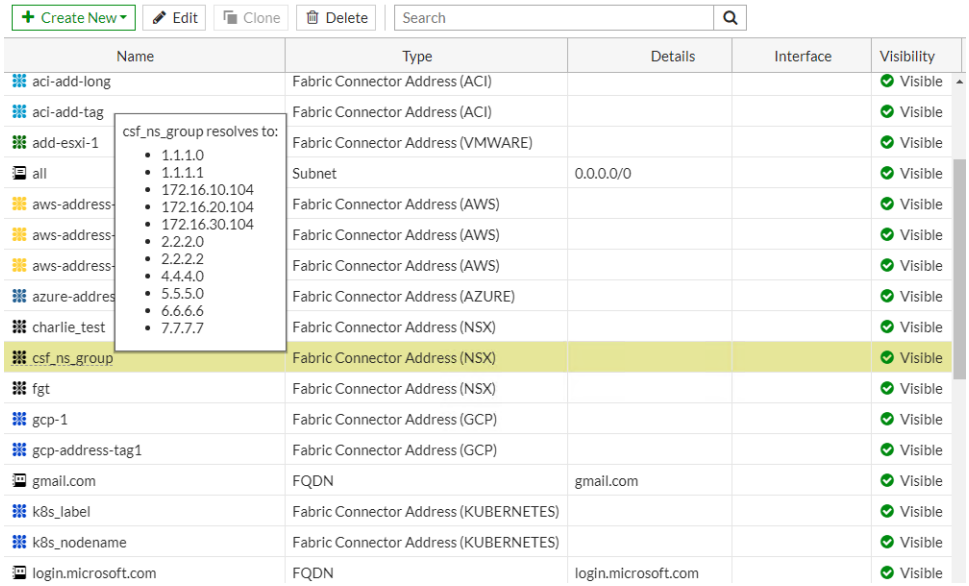
To import all groups from NSX-T Manager:

```
# execute nsx group import nsx_t24 root
[1] 663a7686-b9a3-4659-b06f-b45c908349a0 ServiceInsertion_NSGroup:
  Name:ServiceInsertion_NSGroup
```

```
  Address:10.0.0.2
[2] 336914ba-0660-4840-b0f1-9320f5c5ca5e csf_ns_group:
  Name:csf_ns_group
  Address:1.1.1.0
  Address:1.1.1.1
  Address:172.16.10.104
  Address:172.16.20.104
  Address:172.16.30.104
  Address:2.2.2.0
  Address:2.2.2.2
  Address:4.4.4.0
  Address:5.5.5.0
  Address:6.6.6.6
  Address:7.7.7.7
[3] c462ec4d-d526-4ceb-aeb5-3f168cecd89d charlie_test:
  Name:charlie_test
  Address:1.1.1.1
  Address:2.2.2.2
  Address:6.6.6.6
  Address:7.7.7.7
[4] ff4dcb08-53cf-46bd-bef4-f7aeda9c0ad9 fgt:
  Name:fgt
  Address:172.16.10.101
  Address:172.16.10.102
  Address:172.16.20.102
  Address:172.16.30.103
[5] 3dd7df0d-2baa-44e0-b88f-bd21a92eb2e5 yongyu_test:
  Name:yongyu_test
  Address:1.1.1.0
  Address:2.2.2.0
  Address:4.4.4.0
  Address:5.5.5.0
```

To view the dynamic firewall IP addresses that are resolved by the SDN connector in the GUI:

1. Go to *Policy & Objects* > *Addresses* to view the IP addresses resolved by an SDN connector.



To view the dynamic firewall IP addresses that are resolved by the SDN connector in the CLI:

```
# show firewall address csf_ns_group
config firewall address
    edit "csf_ns_group"
        set uuid ee4a2696-bacd-51e9-f828-59457565b880
        set type dynamic
        set sdn "nsx_t24"
        set obj-id "336914ba-0660-4840-b0f1-9320f5c5ca5e"
        config list
            edit "1.1.1.0"
            next
            edit "1.1.1.1"
            next
            edit "172.16.10.104"
            next
            edit "172.16.20.104"
            next
            edit "172.16.30.104"
            next
            edit "2.2.2.0"
            next
            edit "2.2.2.2"
            next
            edit "4.4.4.0"
            next
            edit "5.5.5.0"
            next
            edit "6.6.6.6"
```

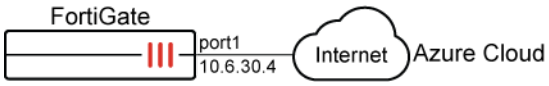
```
            next
            edit "7.7.7.7"
            next
        end
    next
end
```

Multiple concurrent SDN connectors

This guide shows how to configure SDN connectors and resolve dynamic firewall addresses through the configured SDN connector in FortiOS.

FortiOS supports multiple SDN connectors including public connectors (AWS, Azure, GCP, OCI, AliCloud) and private connectors (Kubernetes, VMware ESXi, VMware NSX, OpenStack, Cisco ACI, Nuage). FortiOS also supports multiple instances for each type of SDN connector.

This guide uses an Azure SDN connector as an example. The configuration procedure for all supported SDN connectors is the same. In the following topology, the FortiGate accesses the Azure public cloud through the Internet:



This process consists of the following:

1. Configure the interface.
2. Configure a static route to connect to the Internet.
3. Configure two Azure SDN connectors with different client IDs.
4. Check the configured SDN connectors.
5. Create two firewall addresses.
6. Check the resolved firewall addresses after the update interval.
7. Run diagnose commands.

To configure the interface:

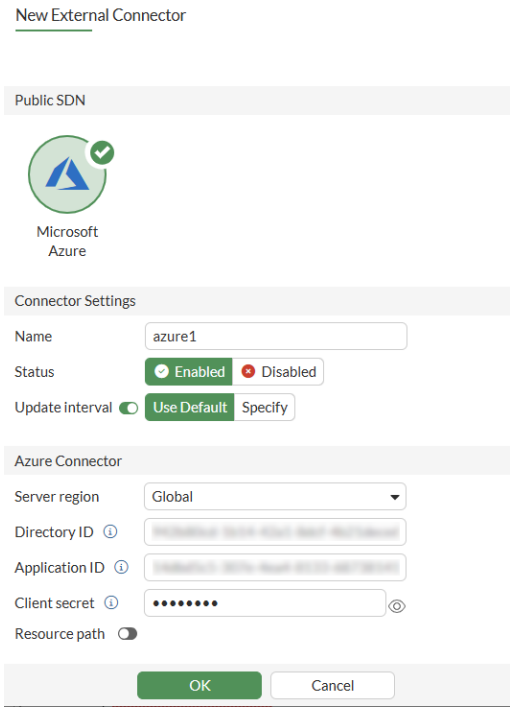
1. In FortiOS, go to *Network* > *Interfaces*.
2. Edit port1:
   a. From the *Role* dropdown list, select *WAN*.
   b. In the *IP/Network Mask* field, enter 10.6.30.4/255.255.255.0 for the interface connected to the Internet.

To configure a static route to connect to the Internet:

1. Go to *Network* > *Static Routes*. Click *Create New*.
2. In the *Destination* field, enter 0.0.0.0/0.0.0.0.
3. From the *Interface* dropdown list, select *port1*.
4. In the *Gateway Address* field, enter 10.60.30.254.

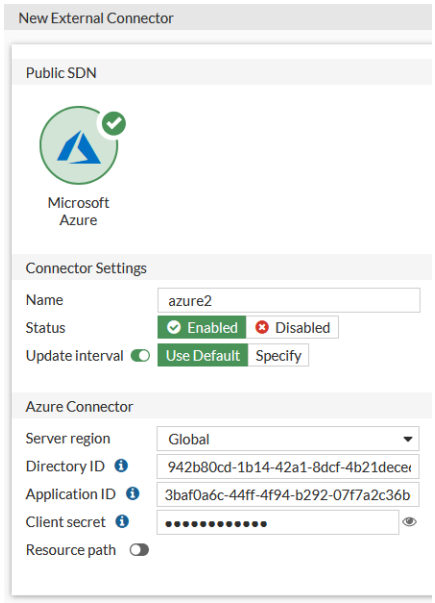
To configure two Azure SDN connectors with different client IDs:

1. Go to *Security Fabric* > *External Connectors*.
2. Click *Create New*. Configure the first SDN connector:
   a. Select *Microsoft Azure*.
   b. In the *Name* field, enter azure1.
   c. In the *Status* field, select *Enabled*.
   d. From the *Server* region dropdown list, select *Global*.
   e. In the *Directory ID* field, enter the directory ID. In this example, it is 942b80cd-1b14-42a1-8dcf-4b21dece61ba.
   f. In the *Application ID* field, enter the application ID. In this example, it is 14dbd5c5-307e-4ea4-8133-68738141feb1.
   g. In the *Client secret* field, enter the client secret.
   h. Leave the *Resource path* disabled.
   i. Click *OK*.



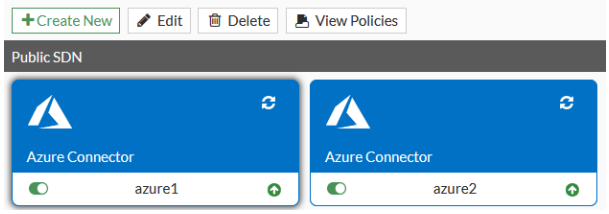
3. Click *Create New*. Configure the second SDN connector:
   a. Select *Microsoft Azure*.
   b. In the *Name* field, enter azure2.
   c. In the *Status* field, select *Enabled*.
   d. From the *Server* region dropdown list, select *Global*.
   e. In the *Directory ID* field, enter the directory ID. In this example, it is 942b80cd-1b14-42a1-8dcf-4b21dece61ba.
   f. In the *Application ID* field, enter the application ID. In this example, it is 3baf0a6c-44ff-4f94-b292-07f7a2c36be6.
   g. In the *Client secret* field, enter the client secret.

h. Leave the *Resource path* disabled.
i. Click *OK*.



To check the configured SDN connectors:

1. Go to *Security Fabric* > *External Connectors*.
2. Click the *Refresh* icon in the upper right corner of each configured SDN connector. A green up arrow appears in the lower right corner, meaning that both SDN connectors are connected to the Azure cloud using different client IDs.

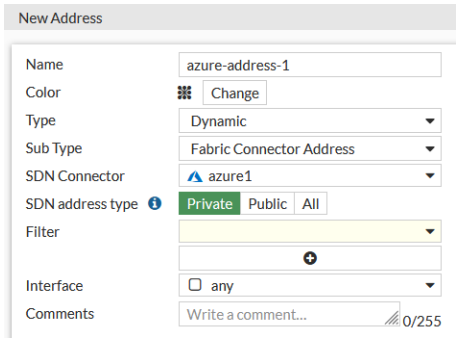


To create two firewall addresses:

This process creates two SDN connector firewall addresses to associate with the configured SDN connectors.

1. Go to *Policy & Objects* > *Addresses* and select *Address*.
2. Click *Create new*. Configure the first SDN connector firewall address:
   a. In the *Name* field, enter azure-address-1.
   b. From the *Type* dropdown list, select *Dynamic*.
   c. From the *Sub Type* dropdown list, select *Fabric Connector address*.
   d. From the *SDN Connector* dropdown list, select *azure1*.
   e. For *SDN address type*, select *Private*.
   f. From the *Filter* dropdown list, select the desired filter.
   g. For *Interface*, select *any*.

h. Click *OK*.

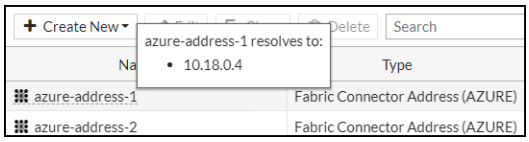


3. Click *Create new*. Configure the second SDN connector firewall address:
   a. In the *Name* field, enter azure-address-1.
   b. From the *Type* dropdown list, select *Dynamic*.
   c. From the *Sub Type* dropdown list, select *Fabric Connector address*.
   d. From the *SDN Connector* dropdown list, select *azure2*.
   e. For *SDN address type*, select *Private*.
   f. From the *Filter* dropdown list, select the desired filter.
   g. For *Interface*, select *any*.
   h. Click *OK*.

To check the resolved firewall addresses after the update interval:

By default, the update interval is 60 seconds.

1. Go to *Policy & Objects* > *Addresses* and select *Address*.
2. Hover over the created addresses. The firewall address that the configured SDN connectors resolved display.



To run diagnose commands:

Run the `show sdn connector status` command. Both SDN connectors should appear with a status of `connected`.

Run the `diagnose debug application azd -1` command. The output should look like the following:

```
Level2-downstream-D # diagnose debug application azd -1
...
azd sdn connector azure1 start updating IP addresses
azd checking firewall address object azure-address-1, vd 0
IP address change, new list:
10.18.0.4
...
```

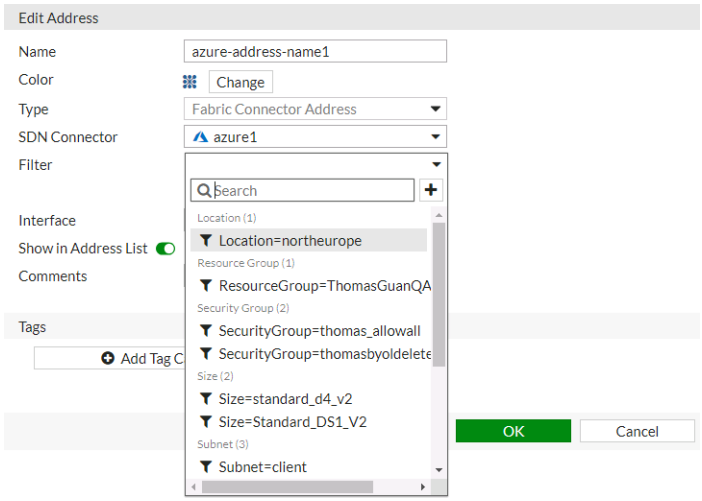
To restart the Azure SDN connector daemon, run the `diagnose test application azd 99` command.

Filter lookup in SDN connectors

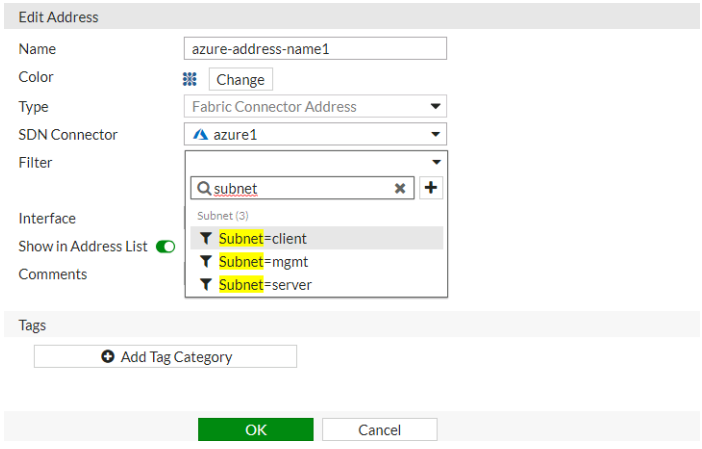
When configuring dynamic address mappings for filters in SDN connectors for Azure, GCP, OpenStack, Kubernetes, and AliCloud, FortiGate can query the filters automatically.

To use the filter lookup:

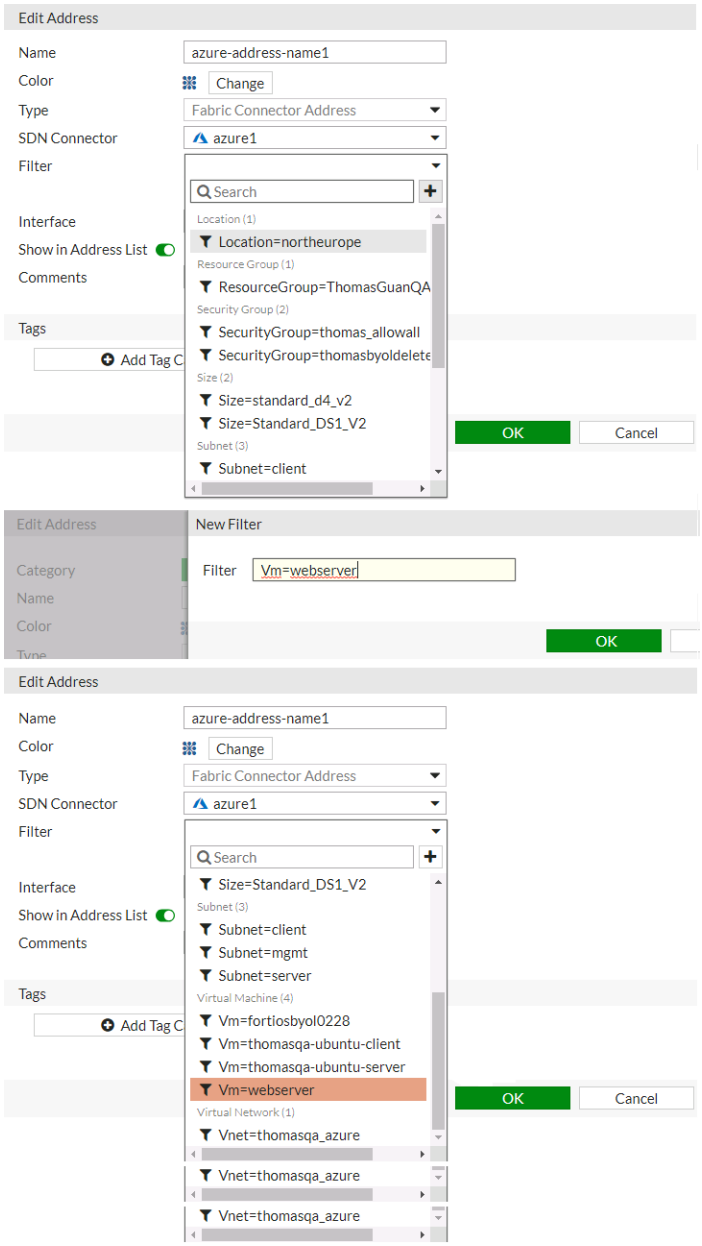
1. Navigate to *Policy & Objects* > *Addresses*.
2. Create or edit an SDN connector type dynamic IP address.
   Supported SDN connector types include: AWS, Azure, GCP, OpenStack, Kubernetes, and AliCloud. The example below is for an Azure SDN connector.
3. In the address *Filter* field, you can perform the following actions:
   - List all available filters.



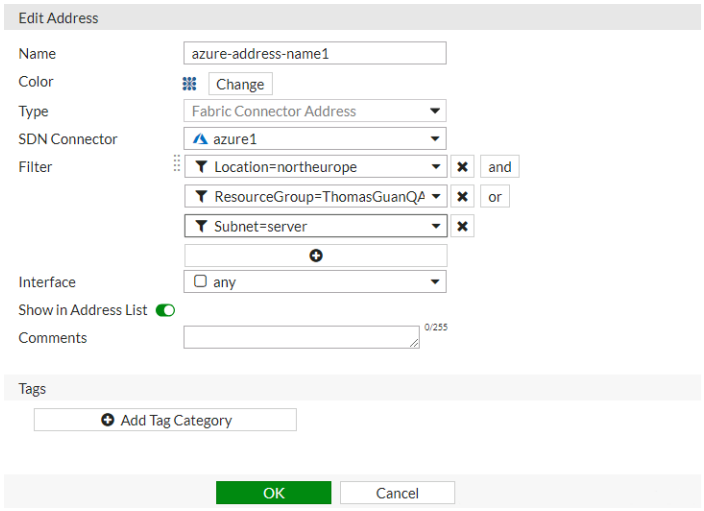
   - Search the available filters.



- Create custom filters.



- Set filter logic [and|or].



Support for wildcard SDN connectors in filter configurations

Wildcards are supported for SDN connectors when configuring dynamic address filters.

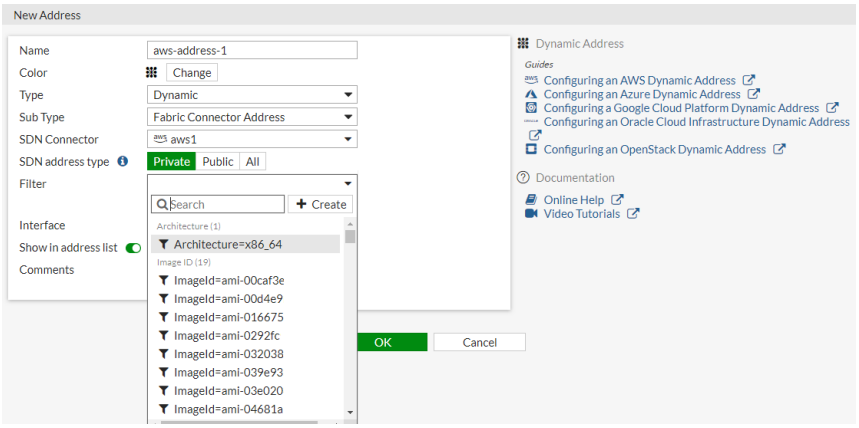
The following SDN connector types are currently supported:

- AWS
- Azure
- Google Cloud Platform
- Kubernetes
- OpenStack
- Oracle Cloud Infrastructure
- VMware ESXi

To configure a dynamic address filter for AWS in the GUI:

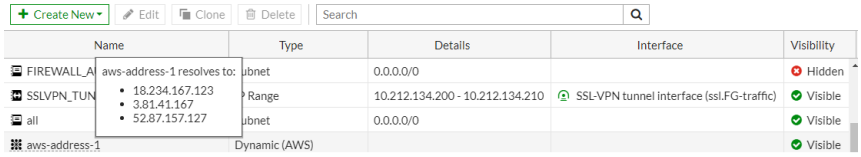
1. Create the SDN connector:
   a. Go to *Security Fabric* > *External Connectors*.
   b. Click *Create New*.
   c. In the *Public SDN* section, click *Amazon Web Services (AWS)*.
   d. Configure the settings as needed.
   e. Click *OK*.
2. Create the dynamic firewall address:
   a. Go to *Policy & Objects* > *Addresses* and select *Address*.
   b. Click *Create new*.
   c. Enter a name for the address, then configure the following settings:

- Set *Type* to *Dynamic*.
- Set *Sub Type* to *Fabric Connector Address*.
- Set *SDN Connector* to *aws1*.
- Set *SDN address type* to *Private*.
- For *Filter*, click *Create*, enter `Tag.Name=aws*`, the click *OK*.

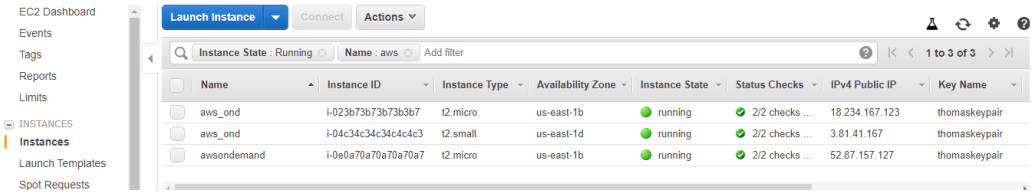


d. Click *OK*.

3. In the address table, hover over the address to view what IPs it resolves to.



4. In AWS, verify to confirm the IP addresses match.



To configure a dynamic address filter for AWS in the CLI:

1. Configure the SDN connector:

```
config firewall address
    edit "aws-address-1"
        set type dynamic
        set sdn "aws1"
        set filter "Tag.Name=aws*"
        set sdn-addr-type public
    next
end
```

2. Create the dynamic firewall address and verify where the IP addresses resolve to:

```
config firewall address
    edit "aws-address-1"
        set type dynamic
        set sdn "aws1"
        set filter "Tag.Name=aws*"
        set sdn-addr-type public
        config list
            edit "18.234.167.123"
            next
            edit "3.81.41.167"
            next
            edit "52.87.157.127"
            next
        end
    next
end
```

3. In AWS, verify that the IP addresses match.

Endpoint/Identity connectors

SSO fabric connectors integrate SSO authentication into the network. This allows users to enter their credentials only once, and have those credentials reused when accessing other network resources through the FortiGate.

The following fabric connectors are available:

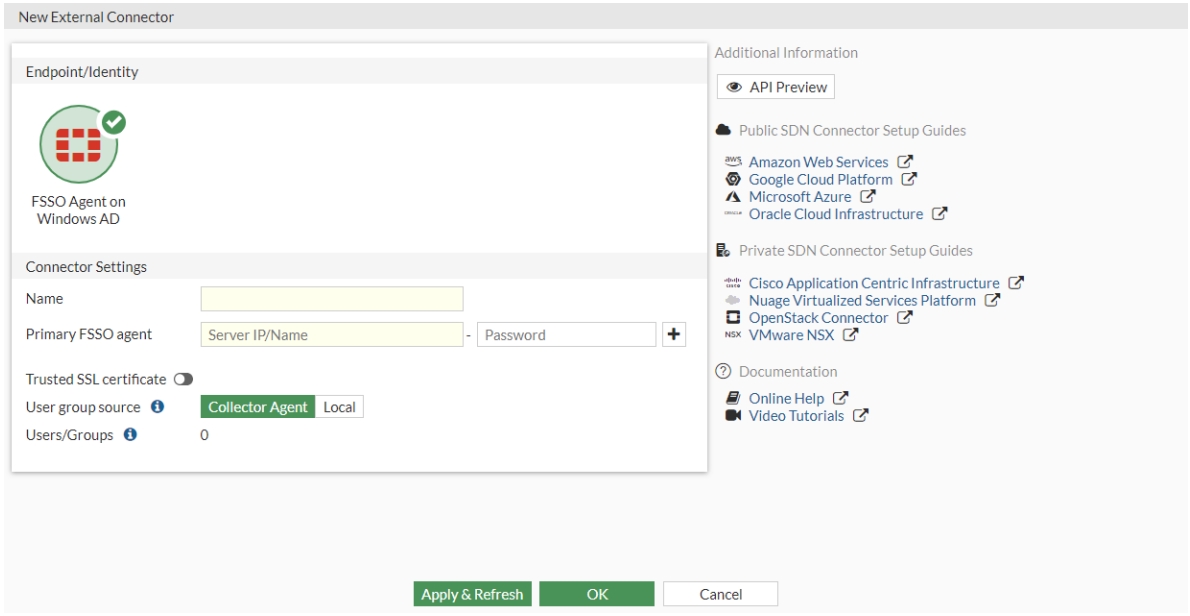
- Fortinet single sign-on agent on page 4679
- Poll Active Directory server on page 4680
- Symantec endpoint connector on page 4681
- RADIUS single sign-on agent on page 4688
- Exchange Server connector on page 4691

Fortinet single sign-on agent

To create an FSSO agent connector in the GUI:

1. Go to *Security Fabric > External Connectors*.
2. Click *Create New*.

3. In the *Endpoint/Identity* section, click *FSSO Agent on Windows AD*.



4. Fill in the *Name*, and *Primary FSSO Agent* server IP address or name and *Password*.
5. Optionally, add more FSSO agents by clicking the plus icon.
6. Optionally, enable *Trusted SSL certificate* and select or import a certificate.
7. Select the *User group source*:
   - *Collector Agent*: User groups will be pushed to the FortiGate from the collector agent. Click *Apply & Refresh* to fetch group filters from the collector agent.
   - *Local*: User groups will be specified in the FortiGate unit's configuration. Select the LDAP server from the list, then click *Edit* to select the *Users*, *Groups*, and *Organizational Units*. Optionally, enable *Proactively retrieve from LDAP server* and configure the *Search filter* and *Interval*.
8. Click *OK*.

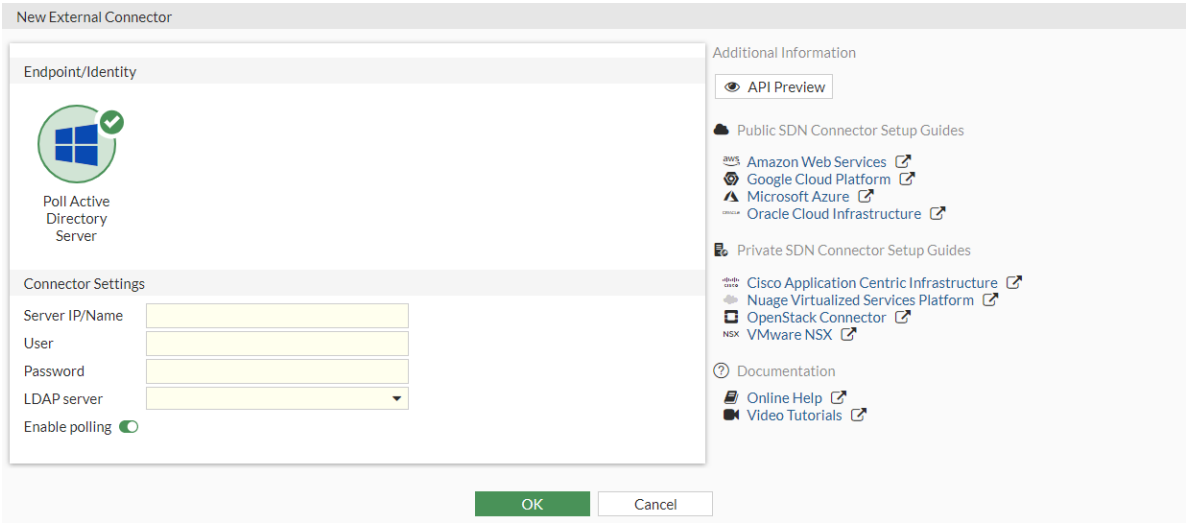
Poll Active Directory server

The FortiGate unit can authenticate users and allow them network access based on groups membership in Windows Active Directory (AD).

To create an AD server connector in the GUI:

1. Go to *Security Fabric* > *External Connectors*.
2. Click *Create New*.

3. In the *Endpoint/Identity* section, click *Poll Active Directory Server*.



4. Fill in the *Server IP/Name*, *User*, and *Password* for the AD server.
5. Select the LDAP server from the list.
6. If necessary, disable *Enable Polling*. This can be used to temporarily stop the FortiGate from polling security event logs on the Windows logon server, for troubleshooting purposes.
7. Click *OK*.

Symantec endpoint connector

With the Fabric connector for Symantec Endpoint Protection Manager (SEPM), you can use the client IP information from SEPM to assign to dynamic IP addresses on FortiOS.

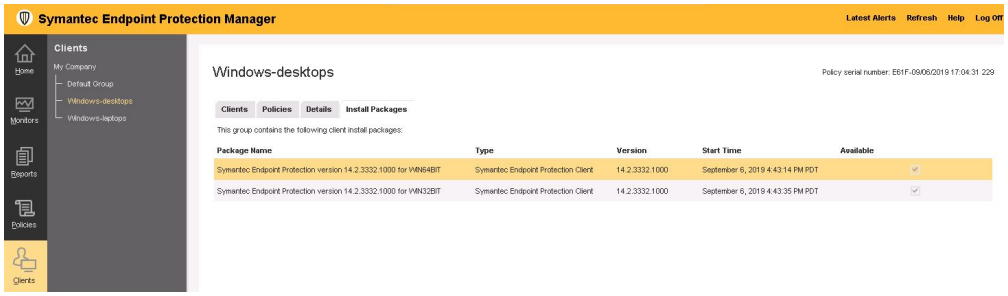
When communication between FortiGate and SEPM is established, FortiGate polls every minute for updates via TLS over port 8446. You can use the CLI to change the default one minute polling interval.

For example, you can create a dynamic Fabric Connector IP address subtype and use it in firewall policies as the source address. The dynamic IP address contains all IP addresses sent by SEPM.

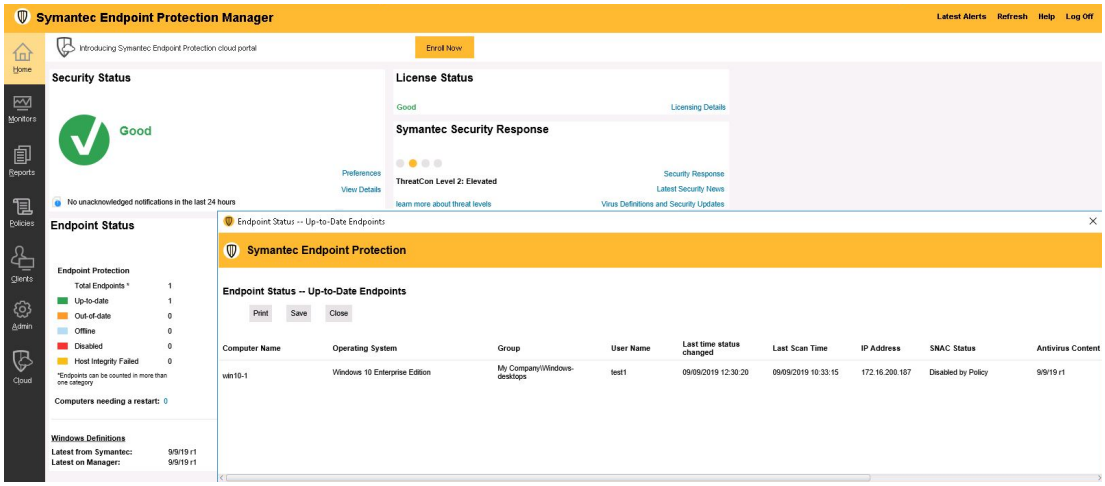
This example shows a dynamic IP address with SEPM and one client PC managed by SEPM using FortiGate as the default gateway.

To configure SEPM on a managed client PC:

1. In SEPM, create client packages for client hosts and group them into SEPM groups. You can install packages locally on clients or download them directly from SEPM.

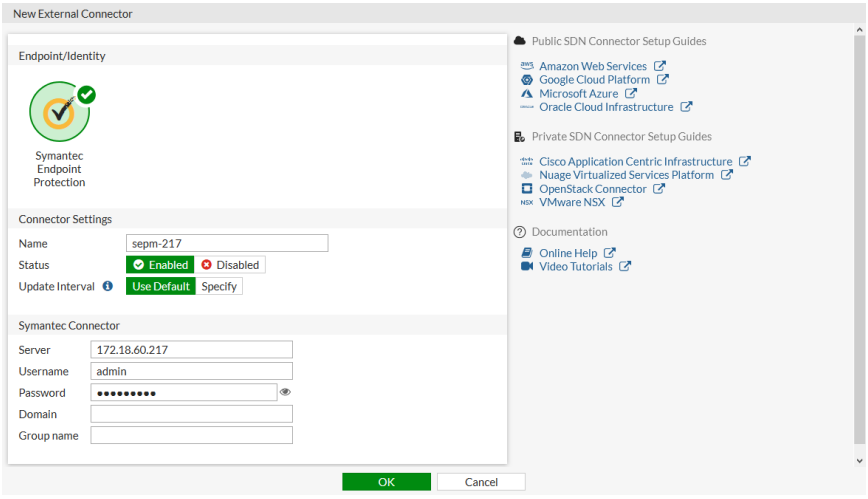


2. When a package is installed on the client host, the host is considered managed by SEPM. Even if the host has multiple interfaces, only one IP per host is displayed.



To configure Symantec endpoint connector on FortiGate in the GUI:

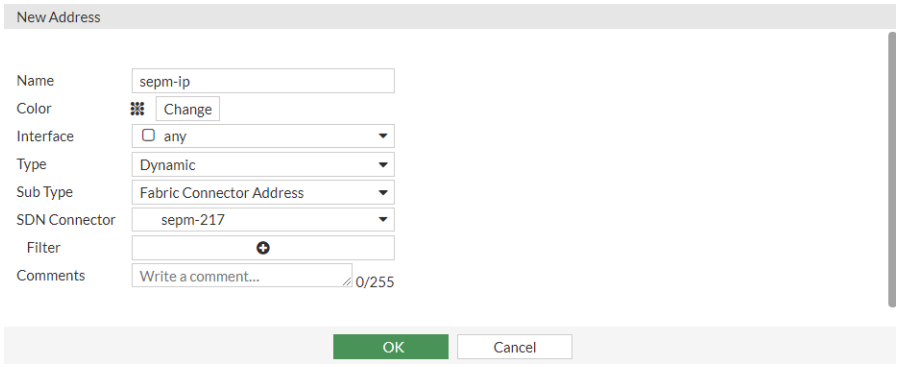
1. Go to *Security Fabric* > *External Connectors* and click *Create New*:
   a. In the *Endpoint/Identity* section, click *Symantec Endpoint Protection*.
   b. Fill in the *Name*, and set the *Status* and *Update Interval*.
   c. Set *Server* to the SEPM IP address.
   d. Enter the *Username* and *Password* for the server.
   e. To limit the domain or group that is monitored, enter them in the requisite fields.



   f. Click *OK*.
   When the connection is established, you can see a green up arrow in the bottom right of the card. You might need to refresh your browser to see the established connection.
2. Go to *Policy & Objects* > *Addresses* and select *Address*.
3. Click *Create new*:
   a. Fill in the address *Name*.
   b. Set *Type* to *Dynamic*.
   c. Set *Sub Type* to *Fabric Connector Address*.

d. Set *SDN Connector* to the fabric connector that you just created.

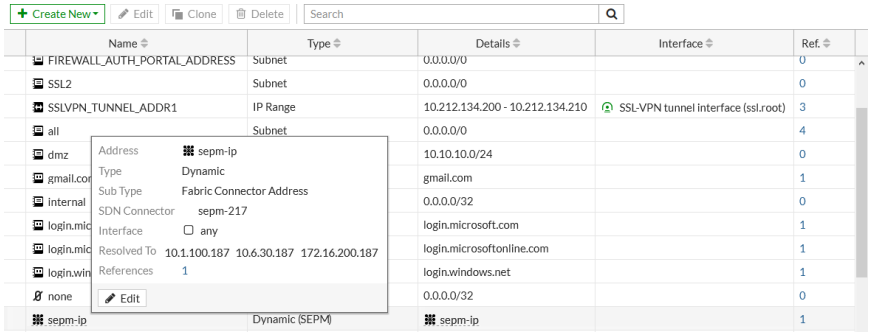
e. Add *Filters* as needed.



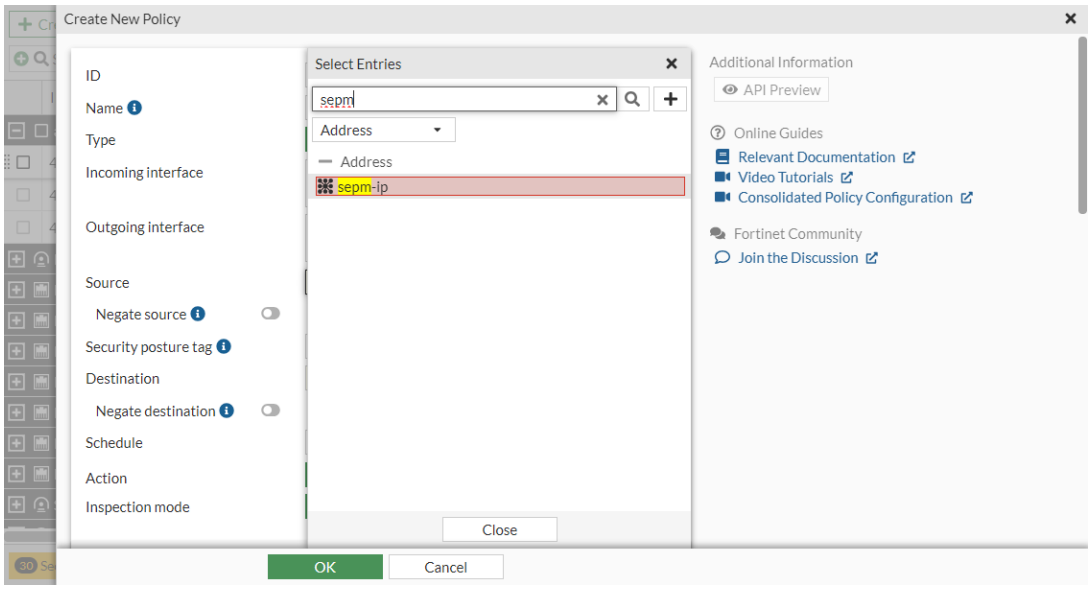
f. Click *OK*.

> Filter options are only available for active computers that are configured and registered in SEPM. Free-form filters can be created manually by clicking *Create* and entering the filter, in the format: `filter_type=value`.
>
> Possible manual filter types are: `GroupName`, `GroupID`, `ComputerName`, `ComputerUUID`, and `OSName`. For example: `GroupName=MyGroup`.

4. Go to *Policy & Objects* > *Addresses* and hover the cursor over the name of the new address to see the resolved IP addresses of the host.

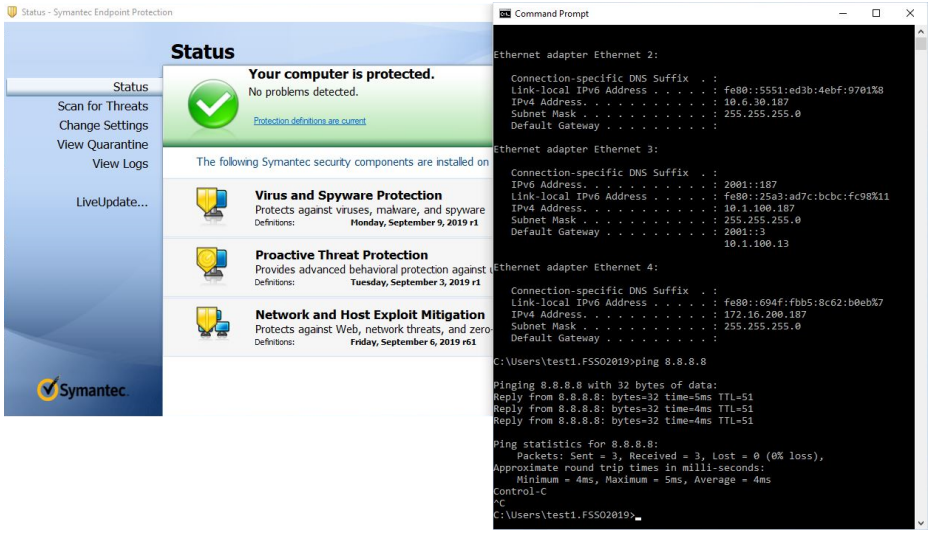


5. Go to *Policy & Objects* > *Firewall Policy*, click *Create New*, and add a policy that uses the dynamic IP address.

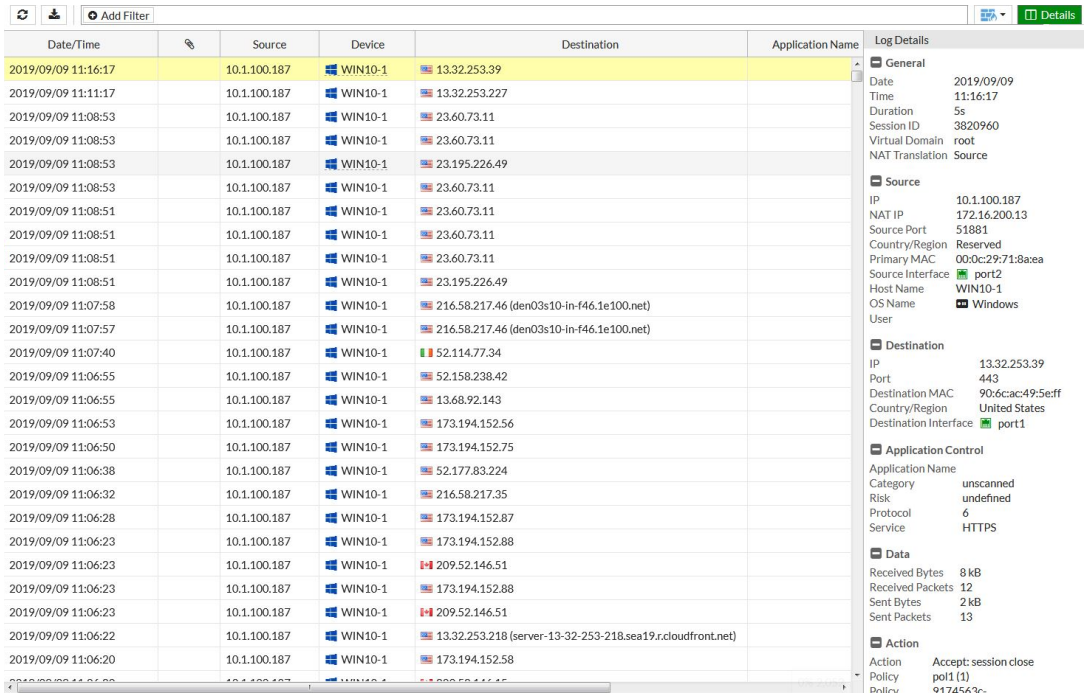


To verify the configuration:

1. On the client PC, check that it is managed by SEPM to access the Internet.



2. On the FortiGate, you can check in *Dashboard* > *FortiView Sources* and *Log & Report* > *Forward Traffic*.



> Because this traffic is not authenticated traffic but is based on source IP address only, it is not shown in the GUI firewall monitor or in the `diagnose firewall auth list` CLI command.

To configure Symantec endpoint connector on FortiGate in the CLI:

1. Create the fabric connector:

```
config system sdn-connector
    edit "sepm-217"
        set type sepm
        set server "172.18.60.217"
        set username "admin"
        set password *********
        set status enable
    next
end
```

2. Create the dynamic IP address:

```
config firewall address
    edit "sepm-ip"
        set type dynamic
        set sdn "sepm-217"
        set filter "ComputerName=win10-1"
        config list
            edit "10.1.100.187"
            next
            edit "10.6.30.187"
```

```
            next
            edit "172.16.200.187"
            next
        end
    next
end
```

3. Add the dynamic IP address to the firewall policy:

```
config firewall policy
    edit 1
        set name "pol1"
        set srcintf "port2"
        set dstintf "port1"
        set srcaddr "sepm-ip"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "certificate-inspection"
        set av-profile "default"
        set logtraffic all
        set fsso disable
        set nat enable
    next
end
```

To troubleshoot Symantec SD connector in the CLI:

```
# diagnose debug application sepmd -1

Output is sent every minute (default). All IPv4 learned from SEPM. IPv6 also sent but not yet
supported.

2019-09-09 12:01:09 sepmd sdn connector sepm-217 start updating IP addresses
2019-09-09 12:01:09 sepmd checking firewall address object sepm-ip, vd 0
2019-09-09 12:01:09 sepmd sdn connector sepm-217 finish updating IP addresses
2019-09-09 12:01:09 sepmd reap child pid: 18079
2019-09-09 12:02:09 sepmd sdn connector sepm-217 prepare to update
2019-09-09 12:02:09 sepmd sdn connector sepm-217 start updating
2019-09-09 12:02:09 sepm-217 sdn connector will retrieve token after 9526 secs
2019-09-09 12:02:09 sym_new_ip_addr ComputerName win10-1
    ComputerUuid AC894D56-BD86-A786-7DDB-7FD98B718AE0, OsName Windows 10
    IP 172.16.200.187
    GroupName My Company\Windows-desktops, GroupId E61FDEA2AC10C80E46D0B31BB58D7CB3
    DomainName Default, DomainId 6C507580AC10C80E5F3CAED5B1711A8E
2019-09-09 12:02:09 sym_new_ip_addr ComputerName win10-1
    ComputerUuid AC894D56-BD86-A786-7DDB-7FD98B718AE0, OsName Windows 10
    IP 10.6.30.187
    GroupName My Company\Windows-desktops, GroupId E61FDEA2AC10C80E46D0B31BB58D7CB3
    DomainName Default, DomainId 6C507580AC10C80E5F3CAED5B1711A8E
```

```
2019-09-09 12:02:09 sym_new_ip_addr ComputerName win10-1
    ComputerUuid AC894D56-BD86-A786-7DDB-7FD98B718AE0, OsName Windows 10
    IP 10.1.100.187
    GroupName My Company\Windows-desktops, GroupId E61FDEA2AC10C80E46D0B31BB58D7CB3
    DomainName Default, DomainId 6C507580AC10C80E5F3CAED5B1711A8E
2019-09-09 12:02:09 2001:0000:0000:0000:0000:0000:0000:0187 is not in IPv4 presentation format


2019-09-09 12:02:09 sepmd sdn connector sepm-217 start updating IP addresses
2019-09-09 12:02:09 sepmd checking firewall address object sepm-ip, vd 0
2019-09-09 12:02:09 sepmd sdn connector sepm-217 finish updating IP addresses
2019-09-09 12:02:09 sepmd reap child pid: 18089
2019-09-09 12:03:09 sepmd sdn connector sepm-217 prepare to update
2019-09-09 12:03:09 sepmd sdn connector sepm-217 start updating
2019-09-09 12:03:09 sepm-217 sdn connector will retrieve token after 9466 secs
2019-09-09 12:03:09 sym_new_ip_addr ComputerName win10-1
    ComputerUuid AC894D56-BD86-A786-7DDB-7FD98B718AE0, OsName Windows 10
    IP 172.16.200.187
    GroupName My Company\Windows-desktops, GroupId E61FDEA2AC10C80E46D0B31BB58D7CB3
    DomainName Default, DomainId 6C507580AC10C80E5F3CAED5B1711A8E
2019-09-09 12:03:09 sym_new_ip_addr ComputerName win10-1
    ComputerUuid AC894D56-BD86-A786-7DDB-7FD98B718AE0, OsName Windows 10
    IP 10.6.30.187
    GroupName My Company\Windows-desktops, GroupId E61FDEA2AC10C80E46D0B31BB58D7CB3
    DomainName Default, DomainId 6C507580AC10C80E5F3CAED5B1711A8E
2019-09-09 12:03:09 sym_new_ip_addr ComputerName win10-1
    ComputerUuid AC894D56-BD86-A786-7DDB-7FD98B718AE0, OsName Windows 10
    IP 10.1.100.187
    GroupName My Company\Windows-desktops, GroupId E61FDEA2AC10C80E46D0B31BB58D7CB3
    DomainName Default, DomainId 6C507580AC10C80E5F3CAED5B1711A8E
2019-09-09 12:03:09 2001:0000:0000:0000:0000:0000:0000:0187 is not in IPv4 presentation format
```

To list the SEPM daemon SDN connectors:

```
# diagnose test application sepmd 1
sepm SDN connector list:
  name: sepm-217, status: enabled, updater_interval: 60
```

To list the SEPM daemon SDN filters:

```
# diagnose test application sepmd 2
sepm SDN connector sepm-217 filter list:
  name: sepm-ip, vd 0, filter 'ComputerName=win10-1'
```

Using a self-signed certificate

Users can explicitly specify a certificate or series of certificates for FortiGate to trust during the connection to the Symantec Endpoint Protection Manager (SEPM) server. For example, a self-signed certificate without proper SAN.

The following new options are added in SEPM `sdn-connector`:

Option	Description
`server-cert`	Trust servers that contain this certificate only.
`server-ca-cert`	Trust only those servers whose certificate is directly or indirectly signed by this certificate.

When these options are enabled, only the specified certificate or series of certificates will be allowed for SEPM server connection ensuring some level of security by blocking off all unspecified certificates.

To specify SEPM certificates:

```
config system sdn-connector
    edit "sepm-217"
        set type sepm
        set server "172.18.60.217"
        set username "admin"
        set password *********
        set status enable
        set server-cert "REMOTE_Cert_1”
        set server-ca-cert "REMOTE_Cert_2”
    next
end
```

The `server-cert` and `server-ca-cert` options are independent of each other and can be set separately. However, when both options are set, both constraints are applied.

RADIUS single sign-on agent

With RADIUS single sign-on (RSSO), a FortiGate can authenticate users who have authenticated on a remote RADIUS server. Based on which user group the user belongs to, the security policy applies the appropriate UTM profiles.

The FortiGate does not interact with the remote RADIUS server; it only monitors RADIUS accounting records that the server forwards (originating from the RADIUS client). These records include the user IP address and user group. The remote RADIUS server sends the following accounting messages to the FortiGate:

Message	Action
Start	If the information in the start message matches the RSSO configuration on the FortiGate, the user is added to the local list of authenticated firewall users.
Stop	The user is removed from the local list of authenticated firewall users because the user session no longer exists on the RADIUS server.

You can configure an RSSO agent connector using the FortiOS GUI; however, in most cases, you will need to use the CLI. There are some default options you may need to modify that can only be done in the CLI.

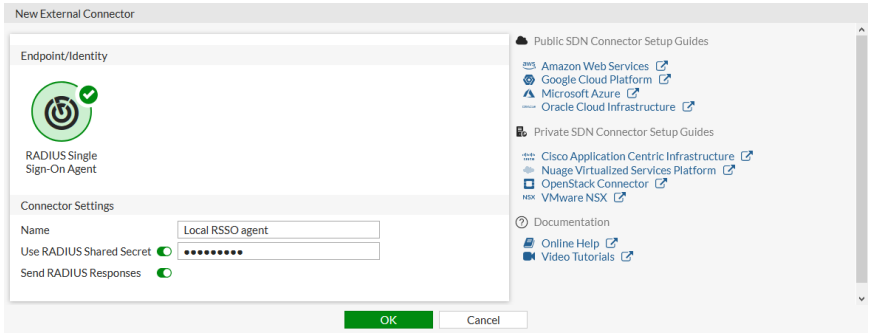
```
config user radius
    edit <name>
        set rsso enable
        set rsso-radius-response enable
        set rsso-secret <password>
        set rsso-endpoint-attribute <attribute>
        set sso-attribute <attribute>
        set delimiter {plus | comma}
    next
end
```

To configure an RSSO agent connector:

1. Create the new connector:
   a. Go to *Security Fabric* > *External Connectors*.
   b. Click *Create New*.
   c. In the *Endpoint/Identity* section, click *RADIUS Single Sign-On Agent*. The *New Fabric Connector* pane opens.
   d. Enter the connector name.
   e. Enable *Use RADIUS Shared Secret*.

   The value entered in *Use RADIUS Shared Secret* must be identical to what the remote RADIUS server uses to authenticate when it sends RADIUS accounting messages to the FortiGate.

   f. Enable *Send RADIUS Responses*.

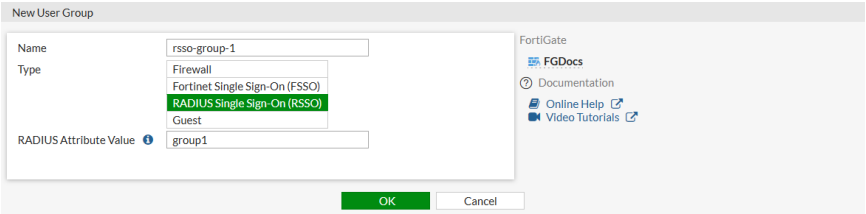


   You should enable *Send RADIUS Responses* because some RADIUS servers continue to send the same RADIUS accounting message several times if there is no response.

   g. Click *OK*.
2. Edit the network interface:
   a. Go to *Network* > *Interfaces*.
   b. Double-click the interface that will receive the RADIUS accounting messages. The *Edit Interface* pane opens.
   c. In the *Administrative Access* section, select the *RADIUS Accounting* checkbox. This will open listening for port 1813 on this interface. The FortiGate will then be ready to receive RADIUS accounting messages.

d. Click *OK*.

3. Create a local RSSO user group:
   a. Go to *User & Authentication* > *User Groups*.
   b. Click *Create New*.
   c. Enter the group name.
   d. For the *Type* field, click *RADIUS Single-Sign-ON (RSSO)*.
   e. Enter a value for *RADIUS Attribute Value*.
   This value by default is the class attribute. The FortiGate uses the content of this attribute in RADIUS accounting start messages to map a user to a FortiGate group, which then can be used in firewall policies.
   In this example configuration, the FortiGate will only add a remote RADIUS user to the local firewall user list if the class attribute in the RADIUS accounting START message contains the value group1.



> If your users are in multiple groups, you will need to add multiple local RSSO user group.

> If the RADIUS attribute value used to map users to a local RSSO group is different than the RADIUS attribute in the RADIUS accounting messages forwarded by the server, you must change it in the CLI.

   f. Click *OK*.

4. Edit the local RSSO agent to modify default options using the CLI.
   For example, the default value for `rsso-endpoint-attribute` might work in common remote access scenarios where users are identified by their unique `Calling-Station-Id`, but in other scenarios the user name might be in a different attribute.

```
config user radius
    edit "Local RSSO Agent"
        set rsso-endpoint-attribute <attribute>
        set sso-attribute <attribute>
    next
end
```

5. Add the local RSSO user group to a firewall policy.

Verifying the RSSO configuration

Verification requires a working remote RADIUS server configured for RADIUS accounting forwarding and wireless or wired clients that use RADIUS for user authentication.

For a quick test, you can use one of the publicly available RADIUS test tools to send RADIUS accounting start and stop messages to the FortiGate. You can also use radclient.

To verify the RSSO configuration:

1. In radclient, enter the RADIUS attributes. These attributes are then executed with the FortiGate IP parameters (sends accounting messages to port 1813) and shared password you configured. `-x` is used for verbose output:

```
root@ControlPC:~# echo "Acct-Status-Type =Start,Framed-Ip-Address=10.1.100.185,User-
Name=test2,Acct-Session-Id=0211a4ef,Class=group1,Calling-Station-Id=00-0c-29-44-BE-B8" |
radclient -x 10.1.100.1 acct 123456
Sending Accounting-Request of id 180 to 10.1.100.1 port 1813
        Acct-Status-Type = Start
        Framed-IP-Address = 10.1.100.185
        User-Name = "test2"
        Acct-Session-Id = "0211a4ef"
        Class = 0x67726f757031
        Calling-Station-Id = "00-0c-29-44-BE-B8"
rad_recv: Accounting-Response packet from host 10.1.100.1 port 1813, id=180, length=20
root@ControlPC:~#
```

2. Verify that the user is in the local firewall user list with the correct type (`rsso`) and local firewall group (`rsso-group1`):

```
# diagnose firewall auth l

10.1.100.185, test2
        type: rsso, id: 0, duration: 5, idled: 5
        flag(10): radius
        server: vdom1
        packets: in 0 out 0, bytes: in 0 out 0
        group_id: 3
        group_name: rsso-group-1

----- 1 listed, 0 filtered ------
```

Exchange Server connector

FortiGate can collect additional information about authenticated users from corporate Microsoft Exchange Servers. After a user logs in, the additional information can be viewed in various parts of the GUI.

The Exchange connector must be mapped to the LDAP server that is used for authentication.

The following attributes are retrieved:

USER_INFO_FULL_NAME	USER_INFO_COMPANY	USER_INFO_CITY
USER_INFO_FIRST_NAME	USER_INFO_DEPARTMENT	USER_INFO_STATE
USER_INFO_LAST_NAME	USER_INFO_GROUP	USER_INFO_POSTAL_CODE
USER_INFO_LOGON_NAME	USER_INFO_TITLE	USER_INFO_COUNTRY
USER_INFO_TELEPHONE	USER_INFO_MANAGER	USER_INFO_ACCOUNT_EXPIRES
USER_INFO_EMAIL	USER_INFO_STREET	
USER_INFO_USER_PHOTO	USER_INFO_POST_OFFICE_BOX	

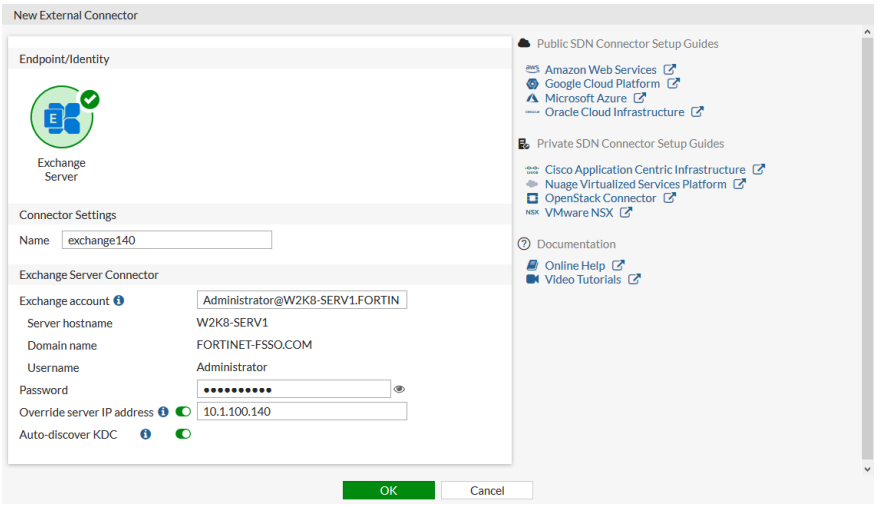
Kerberos Key Distribution Center (KDC) automatic discovery is enabled by default. The FortiGate must be able to use DNS to resolve the KDC IP addresses, otherwise the FortiGate will be unable to retrieve additional user information from the Exchange Server.

KDC automatic discovery can be disabled, and one or more internal IP addresses that the FortiGate can reach can be configured for KDC.

The Override server IP address is enabled when the IP address of the Exchange server cannot be resolved by DNS and must be entered manually.

To configure an Exchange connector in the GUI:

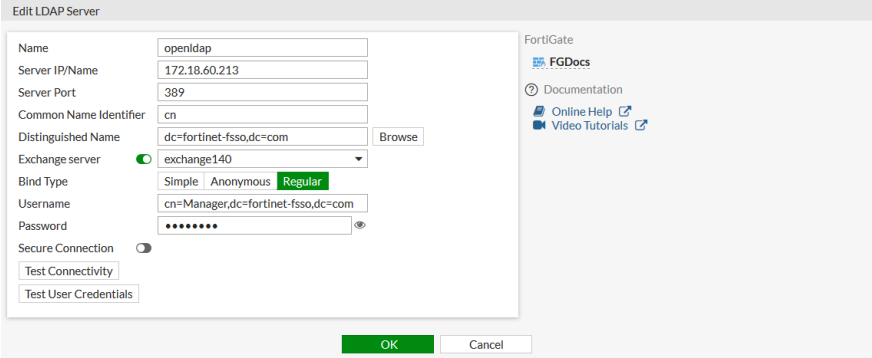
1. Go to *Security Fabric > External Connectors* and click *Create New*.
2. In the *Endpoint/Identity* section, click *Exchange Server*.
3. Set *Name* to *exchange140*.
4. Set *Exchange account* to *Administrator@W2K8-SERV1.FORTINET-FSSO.COM*.
   Administrator is the username, *W2K8-SERV1* is the exchange server name, and *FORTINET-FSSO.COM* is the domain name.
5. Set *Password* to the password.
6. Enable *Override server IP address* and set it to *10.1.100.140*.
7. Ensure that *Auto-discover KDC* is enabled.



   If *Auto-discover KDC* is disabled, one or more KDC IP addresses can be manually entered.
8. Click *OK*.

To link the connector to the LDAP server in the GUI:

1. Go to *User & Authentication > LDAP Servers*.
2. Edit an existing LDAP server, or click *Create New* to create a new one.
3. Enable *Exchange server*, and select the connector from the list.
4. Configure the remaining settings as required.



5. Click *OK*.

To configure an Exchange connector with automatic KDC discovery in the CLI:

```
config user exchange
    edit "exchange140"
        set server-name "W2K8-SERV1"
        set domain-name "FORTINET-FSSO.COM"
        set username "Administrator"
        set password **********
        set ip 10.1.100.140
        set auto-discover-kdc enable
    next
end
```

To link the connector to the LDAP server in the CLI:

```
config user ldap
    edit "openldap"
        set server "172.18.60.213"
        set cnid "cn"
        set dn "dc=fortinet-fsso,dc=com"
        set type regular
        set username "cn=Manager,dc=fortinet-fsso,dc=com"
        set password **********
        set group-member-check group-object
        set group-object-filter "(&(objectclass=groupofnames)(member=*))"
        set member-attr "member"
        set user-info-exchange-server "exchange140"
    next
end
```

Verification

To verify that KDC auto-discovery is working:

```
# diagnose wad debug enable category all
# diagnose wad debug enable level verbose
```

```
# diagnose debug enable
# diagnose wad user exchange test-auto-discover
```

```
wad_diag_session_acceptor(3115): diag socket 20 accepted.
__wad_fmem_open(557): fmem=0x12490bd8, fmem_name='cmem 9188 bucket', elm_sz=9188, block_sz=73728,
overhead=0, type=advanced
Starting auto-discover test for all configured user-exchanges.
[NOTE]: If any errors are returned, try manually configuring IPs for the reported errors.

wad_rpc_nspi_test_autodiscover_kdc(1835): Starting DNS SRV request for srv(0x7f938e052050) query(_
kerberos._udp.FORTINET-FSSO.COM)
wad_dns_send_srv_query(705): 1:0: sending DNS SRV request for remote peer _kerberos._udp.FORTINET-
FSSO.COM id=0
1: DNS response received for remote host _kerberos._udp.FORTINET-FSSO.COM req-id=0
wad_dns_parse_srv_resp(409): _kerberos._udp.FORTINET-FSSO.COM: resp_type(SUCCESS)
  srv[0]: name(w2k12-serv1.fortinet-fsso.com) port(88) priority(0) weight(100)
    addr[0]: 10.1.100.131
    addr[1]: 10.6.30.131
    addr[2]: 172.16.200.131
    addr[3]: 2003::131
    addr[4]: 2001::131
  srv[1]: name(fsso-core-DC.Fortinet-FSSO.COM) port(88) priority(0) weight(100)
    addr[0]: 10.6.30.16
    addr[1]: 172.16.200.16
  srv[2]: name(w2k12-serv1.Fortinet-FSSO.COM) port(88) priority(0) weight(100)
    addr[0]: 10.1.100.131
    addr[1]: 172.16.200.131
    addr[2]: 10.6.30.131
    addr[3]: 2001::131
    addr[4]: 2003::131
wad_rpc_nspi_dns_on_discover_kdc_done(1787): Received response for DNS autodiscover req
(0x7f938dfe8050) query(_kerberos._udp.FORTINET-FSSO.COM) n_rsp(3)

Completed auto-discover test for all configured user-exchanges.
```

To check the collected information after the user has been authenticated:

1. In the GUI, go to *Dashboard > Assets & Identities*, expand the *Firewall Users* widget, and hover over the user name.
2. In the CLI, run the following diagnose command:

```
# diagnose wad user info 20 test1
'username' = 'test1'
'sourceip' = '10.1.100.185'
'vdom' = 'root'
'cn' = 'test1'
'givenName' = 'test1'
'sn' = 'test101'
'userPrincipalName' = 'test1@Fortinet-FSSO.COM'
'telephoneNumber' = '604-123456'
'mail' = 'test1@fortinet-fsso.com'
```

```
'thumbnailPhoto' = '/tmp/wad/user_info/76665fff62ffffffffffffffffffff75ff68ffffffffa'
'company' = 'Fortinet'
'department' = 'Release QA'
'memberOf' = 'CN=group321,OU=Testing,DC=Fortinet-FSSO,DC=COM'
'memberOf' = 'CN=g1,OU=Testing,DC=Fortinet-FSSO,DC=COM'
'memberOf' = 'CN=group21,OU=Testing,DC=Fortinet-FSSO,DC=COM'
'memberOf' = 'CN=group1,OU=Testing,DC=Fortinet-FSSO,DC=COM'
'manager' = 'CN=test6,OU=Testing,DC=Fortinet-FSSO,DC=COM'
'streetAddress' = 'One Backend Street 1901'
'l' = 'Burnaby'
'st' = 'BC'
'postalCode' = '4711'
'co' = 'Canada'
'accountExpires' = '9223372036854
```

If the results are not as expected, verify what information FortiGate can collect from the Exchanger Server:

```
# diagnose test application wad 2500
# diagnose test application wad 162
```

External feeds

The FortiGate dynamically imports an external list from an HTTP/HTTPS server in the form of a plain text file. The imported list is then available as an external feed, which can be used to enforce special security requirements, such as long-term policies to always allow or block access to certain websites, or short-term requirements to block access to known compromised locations. The external feeds are dynamically synchronized and are updated periodically so that any changes are immediately imported by FortiOS.

If the FortiGate loses connectivity with the external server, the external feed will continue to function despite the *Connection Status* error or reboot. However, the external feed will not be updated and no new entries will be added until the connection is re-established.

FortiOS also supports STIX/TAXII format. See STIX format for external feeds on page 4726 for more information.

There are five types of external feeds:

FortiGuard Category	The FortiGate dynamically imports a text file from an external server, which contains one URL per line. See FortiGuard category external feed on page 4709 for more information.
IP Address	The FortiGate dynamically imports a text file from an external server, which contains one IP/IP range/subnet per line. See IP address external feed on page 4712 for more information.
Domain Name	The FortiGate dynamically imports a text file from an external server, which contains one domain per line. Simple wildcards are supported. See Domain name external feed on page 4716 for more information.

MAC Address	The FortiGate dynamically imports a text file from an external server, which contains one MAC address, MAC range, or MAC OUI per line. See MAC address external feed on page 4718 for more information.
Malware Hash	The FortiGate dynamically imports a text file from an external server, which contains one hash per line in the format `<hex hash> [optional hash description]`. Each line supports MD5, SHA1, and SHA256 hex hashes. See Malware hash external feed on page 4720 for more information.

Additionally, the EMS external feed is integrated with FortiClient EMS, but it is not configured in the same way as the preceding feeds:

EMS External Feed	A FortiGate can pull malware external feeds from FortiClient EMS, which in turn receives malware hashes detected by FortiClient. The malware hash can be used in an antivirus profile when AV scanning is enabled with block or monitor actions. See Malware external feed from EMS on page 2287 for an example.

FortiManager can host external feeds. See External resources in the FortiManager Administration Guide.

External resources file format

File format requirements for a HTTP/HTTPS external resources file:

- The file is in plain text format with each URL list, IP address, domain name, or malware hash occupying one line.

  Comments can be added by using the number sign, for example: `# This is a test`.
- The file is limited to a maximum size and entry limit, based on the device model; see External resource entry limit on page 4697.
- The external resources update period can be set to 1 minute, hourly, daily, weekly, or monthly (43200 min, 30 days).
- The external resources type as category (URL list) and domain (domain name list) share the category number range 192 to 221 (total of 30 categories).
- There is no duplicated entry validation for the external resources file (entry inside each file or inside different files).
- If the number of entries exceed the limit, a warning is displayed. Additional entries beyond the threshold will not be loaded.

For URL list (type = category):

- The scheme is optional, and will be truncated if found; https:// and http:// are not required.
- Wildcards are allowed at the beginning or end or the URL, for example: *.domain.com or domain.com.*.
- IDN and UTF encoding URL are supported .
- The URL can be an IPv4 or IPv6 address. An IPv6 URL must be in [ ] format.
- If using an IP address, it cannot be a private IP address.

For IP address list (type = address):

- The IP address can be a single IP address, subnet address, or address range. For example, 192.168.1.1, 192.168.10.0/24, or 192.168.100.1-192.168.100.254.

- The address can be an IPv4 or IPv6 address. An IPv6 address does not need to be in [ ] format.

For domain name list (type = domain):

- Simple wildcards are allowed in the domain name list, for example: *.test.com.
- IDN (international domain name) is supported.

For MAC address list (type = mac-address):

- The MAC address can be a single MAC address, MAC OUI, or MAC range. For example, 01:01:01:01:01:01, 8c:aa:b5, or 01:01:01:01:01:01-01:01:02:50:20:ff.
- The hexadecimal digits in MAC address must only be separated by colons.

For malware hash list (type = malware):

- The malware hash list follows a strict format in order for its contents to be valid. Malware hash signature entries must be separated into each line. A valid signature must follow this format:

```
# MD5 Entry with hash description
aa67243f746e5d76f68ec809355ec234  md5_sample1

# SHA1 Entry with hash description
a57983cb39e25ab80d7d3dc05695dd0ee0e49766  sha1_sample2

# SHA256 Entry with hash description
ae9bc0b4c5639d977d720e4271da06b50f7c60d1e2070e9c75cc59ab30e49379  sha256_sample1

# Entry without hash description
0289b0d967cb7b1fb1451339c7b9818a621903090e0020366ab415c549212521

# Invalid entries
7688499dc71b932feb126347289c0b8a_md5_sample2
7614e98badca10b5e2d08f8664c519b7a906fbd5180ea5d04a82fce9796a4b87sha256_sample3
```

To determine the external resource table size limit for your device:

```
# print tablesize
...
system.external-resource: 0 256 0 4
...
```

In this example, a FortiGate 60E has no global limit and a per-VDOM limit of 256. The total number of feeds is determined by the device's available memory, with no fixed maximum imposed.

External resource entry limit

The external resource entry limit is global, and file size restrictions change according to the device model. If VDOMs are enabled, global entries are counted first, then VDOM entries in alphabetical order based on the VDOMs' names.

If more than the maximum number of entries are added, the most recently added entries are truncated unless the order is manually changed. The entry order can be changed using the `move` CLI command. For example:

```
config system external-resource
    move "entry2" before "entry1"
end
```

The maximum number of each type of entry and the file size limit for each model range are as follows:

	High-End (Data Center)	Mid-Range (Campus)	Entry-Level (Branch)
Category	2 000 000	300 000	150 000
IP address	5 000 000	1 000 000	300 000
Domain	5 000 000	3 000 000	1 000 000
MAC	1 000 000	1 000 000	1 000 000
File size limit (MB)	128	64	32

For example, a FortiGate 601E, a mid-range device, is configured as follows:

- global VDOM: One external feed, g-category-push, with one entry.
- root VDOM: One external feed, r-category-push, with one entry.
- vd1 VDOM: Two external feeds, v-category-300000 with 300000 entries first, and v-category-push with one entry second.
- vd2 VDOM: One external feed, z-category-push, with one entry.

There are more than 300000 entries, so some of the entries will be truncated.

- The global VDOM is counted first, so its entry is kept:

```
FGT (global)# diagnose sys external-resource stats
name: g-category-push; uuid_idx: 606; type: category; update_method: push; total lines: 1;
valid lines: 1; error lines: 0; used: no; buildable: 1; total in count file: 1;
```

- The root VDOM is alphabetically before the vd1 and vd2 VDOMs, so its entry is kept:

```
FGT (root)# diagnose sys external-resource stats
name: g-category-push; uuid_idx: 606; type: category; update_method: push; total lines: 1;
valid lines: 1; error lines: 0; used: no; buildable: 1; total in count file: 1;
name: r-category-push; uuid_idx: 746; type: category; update_method: push; total lines: 1;
valid lines: 1; error lines: 0; used: no; buildable: 1; total in count file: 1;
```

- The vd1 VDOM is next alphabetically. The maximum number of entries is 300000, so 299998 entries from the v-category-3000000 external feed are kept, and no entries from the v-category-push feed:

```
FGT (vd1)# diagnose sys external-resource stats
name: g-category-push; uuid_idx: 606; type: category; update_method: push; total lines: 1;
valid lines: 1; error lines: 0; used: no; buildable: 1; total in count file: 1;
name: v-category-300000; uuid_idx: 863; type: category; update_method: feed; truncated total
lines: 300000; valid lines: 299999; error lines: 1; used: no; buildable: 299998; total in
count file: 300000;
name: v-category-push; uuid_idx: 868; type: category; update_method: push; total lines: 1;
valid lines: 1; error lines: 0; used: yes; buildable: 0; total in count file: 1;
```

- The vd2 VDOM is last alphabetically and the maximum number of entries has already been reached, so all of its entries are truncated:

```
FGT (vd2)# diagnose sys external-resource stats
name: g-category-push; uuid_idx: 606; type: category; update_method: push; total lines: 1;
valid lines: 1; error lines: 0; used: no; buildable: 1; total in count file: 1;
name: z-category-push; uuid_idx: 989; type: category; update_method: push; total lines: 1;
valid lines: 1; error lines: 0; used: no; buildable: 0; total in count file: 1;
```

Configuring an external feed

An external feed can be configured on the *Security Fabric > External Connectors* page. After clicking *Create New*, there are five external feed options available: *FortiGuard Category*, *IP Address*, *Domain Name*, *MAC Address*, and *Malware Hash*. When configuring the external feed settings, the *Update method* can be either a pull method (*External Feed*) or a push method (*PUSH API*).

This topic includes example external feed configurations:

- Configuring an external feed with an external feed update
- Configuring external feed authentication
- Configuring an external feed with a push API update
- Configuring mTLS client certification on page 4707

When multi-VDOM mode is enabled, external feed connectors can be defined in the global VDOM or within a VDOM. See External feed connectors per VDOM on page 4722 for example configurations.

Configuring an external feed with an external feed update

The external feed will periodically fetch entries from the URI using HTTP or HTTPS.

To configure the external feed in the GUI:

1. Go *Security Fabric > External Connectors* and click *Create New*.
2. In the *External Feeds* section, select the required feed type.
3. Configure the connector settings:

Status	Enable/disable the connector.
Name	Enter a name for the external feed connector.
Update method	Select *External Feed*.
URL of external resource	Enter the link to the external resource file. HTTP, HTTPS, and STIX protocols are supported.

HTTP basic authentication	Enable/disable basic HTTP authentication. When enabled, enter the username and password in the requisite fields. See Configuring external feed authentication for more information.
Refresh Rate	The time interval to refresh the external resource, in minutes (1 - 43200, default = 5). The applicable external feed will be triggered to refresh between 0 minutes and the configured value. When the refresh is triggered, if another task is being processed be the schedule worker, the refresh task will be added to the queue.
Comments	Optionally, enter a description of the connector.

4. Click *OK*.

To configure the external feed in the CLI:

```
config system external-resource
    edit <name>
        set status {enable | disable}
        set type {category | address | domain | malware}
        set update-method {feed | push}
        set category <integer, 192-221>
        set username <string>
        set password <string>
        set client-cert-auth {enable | disable}
        set comments <string>
        *set resource <resource-uri>
        set user-agent <string>
        set server-identity-check {none | basic | full}
        *set refresh-rate <integer>
        set source-ip <ip address>
        set source-ip-interface <string>
        set interface-select-method {auto | sdwan | specify}
    next
end
```

The parameter marked with an asterisk (*) is mandatory and must be filled in. The `category` parameter must be set when the `type` is either `category` or `domain`. Other parameters have either default values or are optional.

To improve the security of the connection, it is recommended to enable server certificate validation (`server-identity-check`) either in basic or full mode.

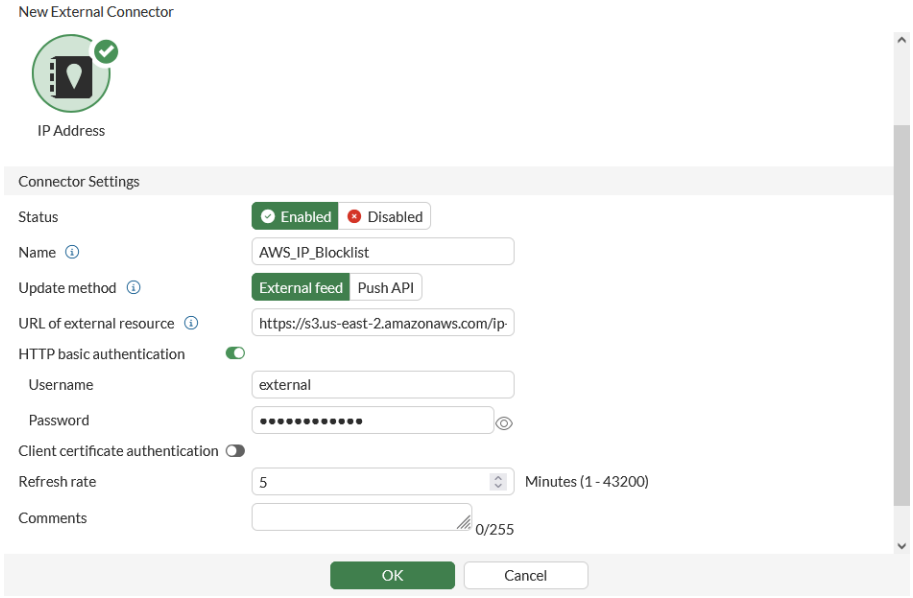
Configuring external feed authentication

External feed connectors support username and password authentication.

The *HTTP basic authentication* field is only visible when the *Update method* is set to *External Feed*.

To enable username and password authentication in an external feed connector:

1. Go *Security Fabric* > *External Connectors*.
2. Click *Create New*, or edit an existing external feed connector.
3. Enable *HTTP basic authentication*.
4. Enter the *Username* and *Password*.



5. Click *OK*.

HTTP header

Additional headers can be included in the `user-agent` field. Use `\r\n` to separate the URL headers, for example:

```
# set user-agent "Firefox\r\nheader1: test1\r\nheader2: test2"
```

Sample request:

```
HTTP request: http
GET /filetypes/test.tar.gz HTTP/1.1
Host: 172.17.219.10
User-Agent: Firefox
header1: test1
header2: test2
Accept: */*
Connection: close
```

External feed connectors use this functionality to support authentication using an API key. The API key authentication can only be configured in the CLI with the `set user-agent` command. The API key must be appended with user-agent in the following format: “`user-agent\r\nAPI-Key:SecretAPIkey`”. API keys are typically used for programmatic access to the resource by an authorized requester. See What Is an API Key in the Fortinet Cyber Glossary for more information.

To enable API key authentication in an external feed connector:

1. Configure the external feed. See Configuring an external feed with an external feed update on page 4699.
2. Configure the user-agent with an API key:

```
config system external resources
    edit <name>
        set user-agent "Firefox\r\nAPI-Key:abcdef12345"
    next
end
```

See Using the AusCERT malicious URL feed with an API key on page 4728 for an example.

Configuring an external feed with a push API update

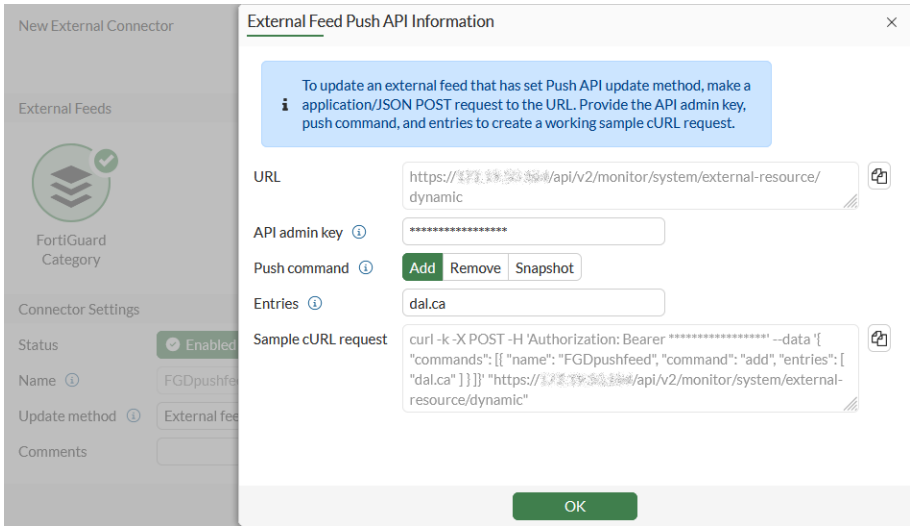
The external feed receives entry updates from webhook requests to the FortiGate REST API. This method provides the code samples needed to perform add, remove, and snapshot operations.

In the following example, a *FortiGuard Category* external feed is used to show the different API push options. A FortiGuard license is required when using a *FortiGuard Category* external feed in a web filter profile in a policy in NGFW profile or policy mode.

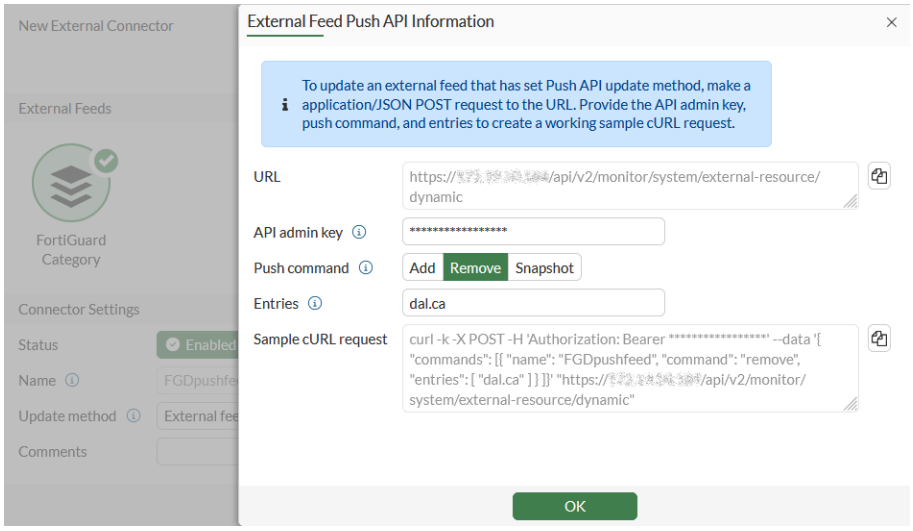
To configure the external feed in the GUI:

1. Go to *Security Fabric* > *External Connectors* and click *Create New*.
2. In the *External Feeds* section, click *FortiGuard Category*.
3. Enter a name.
4. Set the *Update method* to *Push API*.
5. Click *OK*. The *External Feed Push API Information* pane opens that contains the following fields:
   - *URL*: the FortiGate's API URL to call in order to perform the update.
   - *API admin key*: when an API administrator user is configured on the FortiGate, an *API admin key* will be associated with the API administrator. Input the API key to see the final cURL request.
   - *Push command*: select one of three push methods.

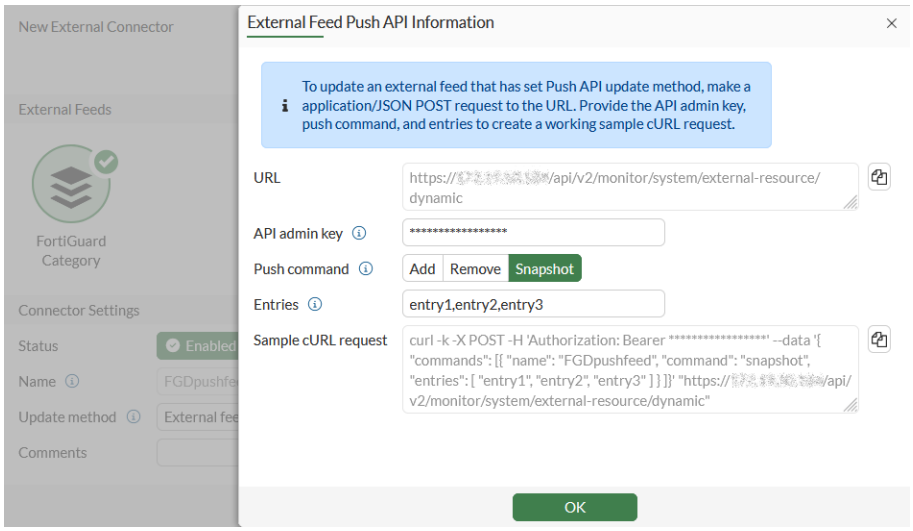
- *Add*: add the specified entries to the external feed.



- *Remove*: remove the specified entries from the external feed.



- *Snapshot*: replace the external feed with all specified entries.



- *Entries*: enter the entries separated by a comma (,) to be applied to the FortiGuard Category external feed list.
- *Sample cURL request*: copy this cURL command to perform the push API update on the FortiGate against the list (*cccccccc*).

See REST API administrator on page 3712 for more information.

6. Copy the content in the *Sample cURL request* field (*Add* is used in this example).
7. Click *OK*.
8. On a client, generate the API request for the external feed.

To configure the external feed in the CLI:

```
config system external-resource
    edit "cccccccc"
        set update-method push
        set category 201
    next
end
```

To use the API in the CLI:

```
# diagnose system external-resource {push-add | push-remove | push-snapshot} <ext_name> <entry>
```

To use the API with a JSON file:

```
# diagnose sys external-resource push-api-json-commands
{
  "commands": [ <Array: Mandatory>
    { <Object: Mandatory>
         "name": <String: Mandatory, Example: "AWS_MALWARE_FEED">
      "command": <String: mandatory, Options: "add", "remove", "snapshot">,
      "entries": [ <Array: Mandatory>
        <String: Mandatory, Example: "10.100.1.1">
```

```
      ]
    }
  ]
}
```

Sample:

```
# diagnose sys external-resource push-api-json-commands '{"commands":
[{"name":"test","command":"add","entries":["10.10.10.1","10.10.10.2"]},
{"name":"test","command":"whatever","entries":["10.10.10.3","10.10.10.4"]}]}'
command returned: EXT_RESOURCE_PUSH_CMD_RETURN_OK
Returned json:
[
  {
    "name":"test",
    "command":"add",
    "status":"success"
  },
  {
    "name":"test",
    "command":"whatever",
    "error":"Invalid command.",
    "status":"error"
  }
]
```

To use the API with a Postman REST client:

1. Create an API administrator in FortiOS with write access.
2. Ensure the API token is generated.
3. Configure the external resource list as needed.
4. In the Postman client, create a new request, set the HTTP method to *POST*, enter the URL.
5. Configure the access token using one of the following methods:
   - To use the bearer token: click the *Authorization* tab, set the *Type* to *Bearer*, and enter the REST API administrator token.
   - To use the access_token parameter: click the *Params* tab and enter the access_token key-value pair (*access_token* and *<key>*).
6. Click the *Body* tab and configure the following:
   a. Select *raw* and set the input type to *JSON*.
   b. Insert the JSON data payload.
7. Click *Send* to send the POST request. If there is a response, the response body appears. For example,

```
POST https://172.18.52.153/api/v2/monitor/system/external-resource/dynamic?access_
token=g1mnfs8bzxk5hf8Qwcz4kx7yn3jHmG&vdom=vd1
Content-Type: application/json
User-Agent: PostmanRuntime/7.29.2
Accept: */*
Postman-Token: 04e10736-190e-4119-92e1-04e91bf99c10
```

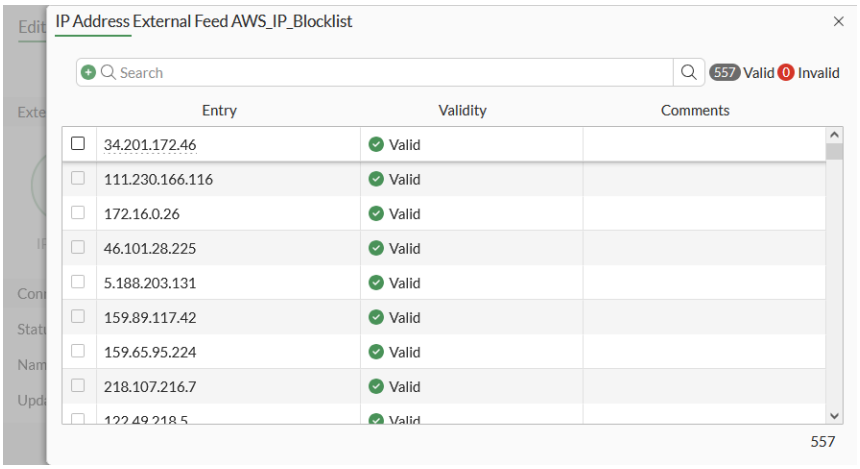
```
Host: 172.18.52.153
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Length: 485

{
   "commands":[
      {
         "name":"ip",
         "command":"add",
         "entries":[
            "10.10.10.1",
            "10.10.10.2"
         ]
      },
      {
         "name":"fqdn",
         "command":"remove",
         "entries":[
            "10.10.10.1",
            "10.10.10.2"
         ]
      },
      {
         "name":"fortiguard",
         "command":"snapshot",
         "entries":[
            "10.10.10.1",
            "10.10.10.2"
         ]
      }
   ]
}

HTTP/1.1 200 OK
date: Fri, 22 Jul 2022 21:10:39 GMT
x-frame-options: SAMEORIGIN
content-security-policy: frame-ancestors 'self'
x-xss-protection: 1; mode=block
cache-control: no-cache, must-revalidate
content-length: 480
content-type: application/json
Connection: keep-alive

{
   "http_method":"POST",
   "results":[
      {
         "name":"ip",
         "command":"add",
         "status":"success"
      },
```

```
      {
         "name":"fqdn",
         "command":"remove",
         "status":"success"
      },
      {
         "name":"fortiguard",
         "command":"snapshot",
         "status":"success"
      }
   ],
   "vdom":"vd1",
   "path":"system",
   "name":"external-resource",
   "action":"dynamic",
   "status":"success",
   "serial":"FG6H1E5819900000",
   "version":"v7.2.1",
   "build":1254
}
```


Viewing the update history

To review the update history of an external feed, go to *Security Fabric* > *External Connectors*, select a feed, and click *Edit*. The *Last Update* field shows the date and time that the feed was last updated.

Click *View Entries* to view the current entries in the list.



Configuring mTLS client certification

Administrators can configure and define a trusted client certificate for mutual TLS (mTLS) authentication in the CLI. This enhances security for the external feed server when connecting to an HTTPS external resource.

During configuration of external resources, a client certificate can be configured that is signed and trusted by a remote mTLS server. When the server asks for client certification upon connection, FortiOS will use the configured client certificate in the TLS handshake process and pass the mTLS authentication.

If FortiOS cannot provide the correct certificate, the server may choose to deny or accept the connection based on its authentication protocol. Therefore, it is crucial to specify client certificate authentication when connecting to an mTLS server. This requirement does not apply if the server is not using mTLS.

To configure client certification mTLS in the CLI:

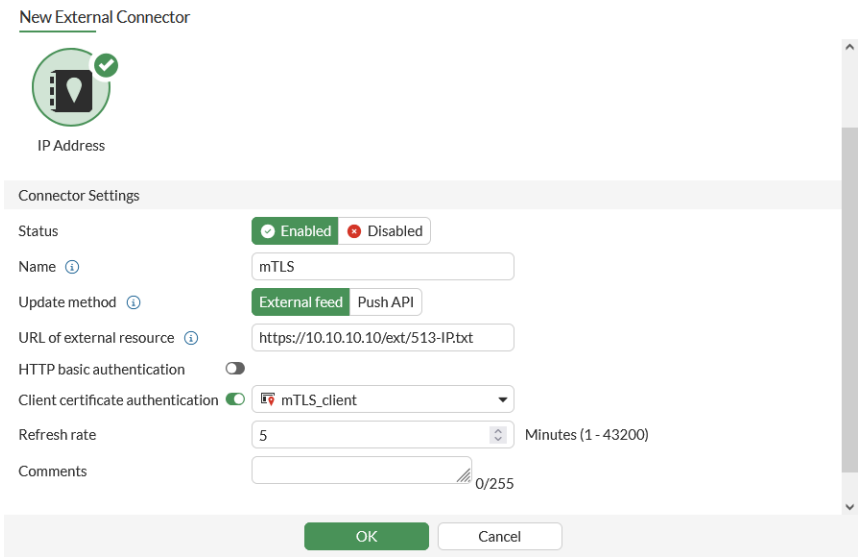
```
config system external-resource
    edit "mTLS"
        set type address
        set client-cert-auth enable
        set client-cert <mTLS_client>
        set resource <HTTPS external resource>
    next
end
```

To configure client certification mTLS in the GUI:

1. Go to *Security Fabric* > *External Connectors*.
2. Click *Create New*.
3. Select *External Feeds* > *IP Address*.
4. Set the *Status* to *Enabled*.
5. Enable *Client certificate authentication* and select the mTLS client certificate from the dropdown list.

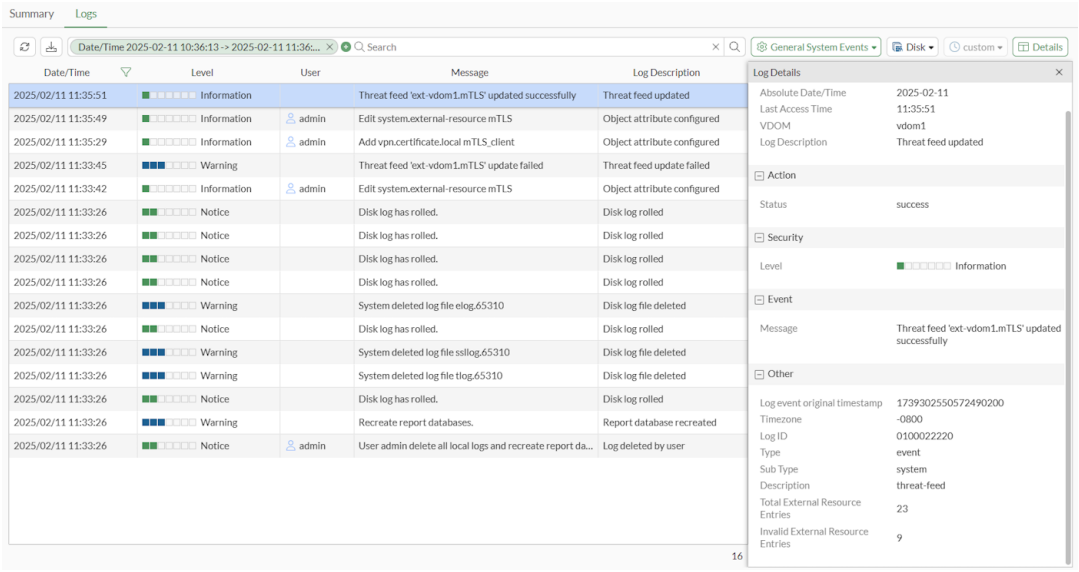
   If the mTLS client certificate has not yet been created, click *Create* and proceed with the configuration.

6. Configure other settings, as needed.



7. Click *OK*.

Logs related to the mTLS client certificate authentication can be found in *Log & Report* > *System Events* > *Logs*.



FortiGuard category external feed

A FortiGuard category external feed is a dynamic list that contains URLs and is periodically updated from an external server. The list is stored in text file format on an external server. After the FortiGate imports this list, it becomes available as a category in the *Remote Categories* group of web filter profiles that can be used to block or monitor URLs matching this category. A category external feed can also be used solely or grouped with other categories to be used for exemptions within an SSL/SSH profile that performs full SSL inspection.

Multiple custom categories can be defined by creating a FortiGuard category external feed for each category.

Text file example:

```
http://example/com.url
https://example.com/url
http://example.com:8080/url
```

The file contains one URL per line. See External resources file format for more information about the URL list formatting style.

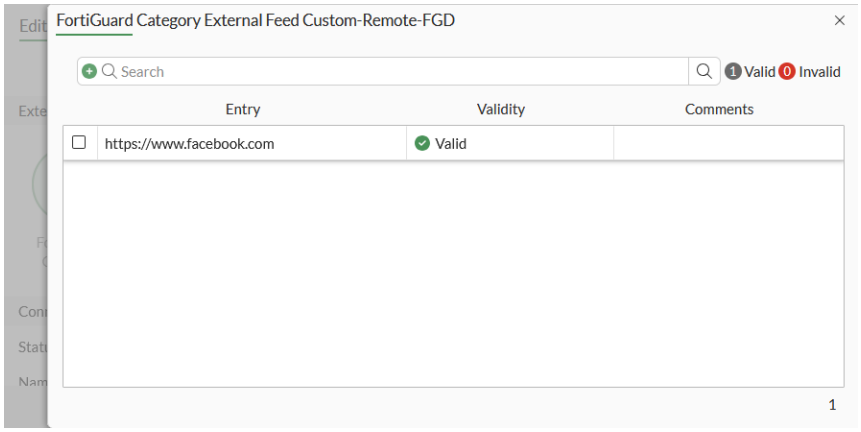
Example configuration

In this example, a list of URLs is imported using the FortiGuard category external feed. The newly created external feed is set to block in the web filter profile, and the web filter profile is applied to a firewall policy. Any traffic that passes through the FortiGate and matches the URLs in the external feed list will be dropped.

To configure a FortiGuard category external feed in the GUI:

1. Go to *Security Fabric* > *External Connectors* and click *Create New*.
2. In the *External Feeds* section, click *FortiGuard Category*.
3. Set the *Name* to *Custom-Remote-FGD*.

4. Set the *Update method* to *External Feed*.
5. Set the *URL of external resource* to *https://192.168.10.13/Override_URLs.txt*.
6. Configure the remaining settings as needed, then click *OK*.
7. Edit the connector, then click *View Entries* to view the URL in the feed, which is https://www.facebook.com.



To configure a FortiGuard category external feed in the CLI:

```
config system external-resource
    edit "Custom-Remote-FGD"
        set type category
        set category 192
        set resource "https://192.168.10.13/Override_URLs.txt"
        set server-identity-check {none | basic | full}
    next
end
```

To improve the security of the connection, it is recommended to enable server certificate validation (`server-identity-check`) either in basic or full mode. By default, it is set to none.

To apply a FortiGuard category external feed in a web filter profile:

1. Go to *Security Profiles* > *Web Filter* and create a new web filter profile, or edit an existing one.
2. Enable *FortiGuard Category Based Filter*.
3. In the *Remote Categories* group, set the action for the *Custom-Remote-FGD* category to *Block*.
4. Configure the remaining settings as needed, then click *OK*.

Selecting the *Allow* action for the *FortiGuard Category Based Filter* does not actually allow the category. It merely implies that no filter has been applied.

We recommend avoid using the *Allow* action for remote categories, as it will not override the original action specified in the *FortiGuard Category Based Filter*.

The *Monitor* and *Block* actions for remote categories can override the original action specified in the *FortiGuard Category Based Filter*.

To apply the web filter profile in a firewall policy:

1. Go to *Policy & Objects > Firewall Policy* and create a new policy, or edit an existing one.
2. Configure the policy fields as required.
3. Under *Security Profiles*, enable *Web Filter* and select the profile used in the previous procedure.
4. Enable *Log Allowed Traffic*.
5. Click *OK*.

URLs that match the FortiGuard category external feed list are rated as the FortiGuard category external feed, overriding their original domain rating. Use the FortiGuard Web Filter Lookup to check the original category of a URL.

To view the web filer logs:

1. Go to *Log & Report > Security Events* and select *Web Filter*.
2. View the log details in the GUI, or download the log file:

```
1: date=2023-02-06 time=09:31:04 eventtime=1675704664795395841 tz="-0800" logid="0316013056"
type="utm" subtype="webfilter" eventtype="ftgd_blk" level="warning" vd="root" policyid=1
poluuid="e8b310ba-914f-51ed-9014-7b2a116f29ad" policytype="policy" sessionid=509983
srcip=172.20.120.13 srcport=54645 srccountry="Reserved" srcintf="port2"
srcintfrole="undefined" srcuuid="3342cb44-9140-51ed-5dbe-8e0787bedeec" dstip=157.240.3.35
dstport=443 dstcountry="United States" dstintf="port3" dstintfrole="wan" dstuuid="3342cb44-
9140-51ed-5dbe-8e0787bedeec" proto=6 httpmethod="GET" service="HTTPS"
hostname="www.facebook.com" agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/64.0.3282.140 Safari/537.36 Edge/18.17763"
profile="default" action="blocked" reqtype="referral" url="https://www.facebook.com/"
referralurl="https://www.google.com/url?url=https://www.facebook.com/&q=facebook&rct=j&sa=X&so
urce=suggest&ct=res&oi=suggest_nav&usg=AOvVaw3XzIKieZE-CH5KqZaBe775&oq=facebook&gs_l=heirloom-
hp..0.5j0i512i433i131i10l3j0i512i433i10l3j0i512i433i131i10l2j0i512i433i10.1716.3397.0.5824.8.8
.0.0.0.0.85.609.8.8.0....0...1ac.1.34.heirloom-hp..0.8.608.798UUeJkbN0" sentbyte=527
rcvdbyte=0 direction="outgoing" msg="URL belongs to a denied category in policy"
ratemethod="domain" cat=192 catdesc="Custom-Remote-FGD"
```

Note that facebook.com, which was originally in the Social Networking category with a default action set to allow in the *FortiGuard Category Based Filter*, has been overridden by the block action of the remote category.

Applying a FortiGuard category external feed in an SSL/SSH profile

A FortiGuard category external feed can be applied in an SSL/SSH profile where full SSL inspection mode is used. The external feed category can be selected in the exempt category list. HTTPS requests that match the URLs in the external feed list will be exempted from SSL deep inspection. This example uses the *Custom-Remote-FGD* external feed configured in the previous example.

To configure the SSL/SSH profile:

1. Go to *Security Profiles > SSL/SSH Inspection* and create a new profile, or edit an existing one.
2. Set the *Inspection method* to *Full SSL Inspection*.

3. In the *Exempt from SSL Inspection* section, locate *Web categories*. Click the *+* and add *Custom-Remote-FGD* in the *FORTIGUARD CATEGORY EXTERNAL FEED* section.
4. Enable *Log SSL exemptions*.
5. Click *OK*.

To apply the SSL/SSH inspection profile in a firewall policy:

1. Go to *Policy & Objects > Firewall Policy* and create a new policy, or edit an existing one.
2. Configure the policy fields as required.
3. Under *Security Profiles*, set *SSL Inspection* to the profile used in the previous procedure.
4. Enable *Log Allowed Traffic*.
5. Click *OK*.

URLs that match the FortiGuard category external feed list are rated as the FortiGuard category external feed, overriding their original domain rating. Use the FortiGuard Web Filter Lookup to check the original category of a URL.

To view the SSL logs:

1. Go to *Log & Report > Security Events* and select *SSL*.
2. View the log details in the GUI, or download the log file:

```
1: date=2023-02-06 time=11:23:54 eventtime=1675711434094550877 tz="-0800" logid="1701062009"
type="utm" subtype="ssl" eventtype="ssl-exempt" level="notice" vd="root" action="exempt"
policyid=1 poluuid="e8b310ba-914f-51ed-9014-7b2a116f29ad" policytype="policy" sessionid=531331
service="SSL" profile="custom-deep-inspection" srcip=172.20.120.13 srcport=52805
srccountry="Reserved" dstip=157.240.3.35 dstport=443 dstcountry="United States"
srcintf="port2" srcintfrole="undefined" dstintf="port3" dstintfrole="wan" srcuuid="3342cb44-
9140-51ed-5dbe-8e0787bedeec" dstuuid="3342cb44-9140-51ed-5dbe-8e0787bedeec" proto=17
tlsver="tls1.3" sni="www.facebook.com" cipher="0x1301" authalgo="ecdsa" kxproto="ecdhe"
eventsubtype="user-category" cat=192 catdesc="Custom-Remote-FGD" hostname="www.facebook.com"
msg="SSL connection is exempted based on user category rating.
```

IP address external feed

An IP address external feed is a dynamic list that contains IPv4 and IPv6 addresses, address ranges, and subnets. The list is periodically updated from an external server and stored in text file format on an external server. After the FortiGate imports this list, it can be used as a source or destination in firewall policies, proxy policies, local-in policies, and ZTNA rules. It can also be used as an external IP block list in DNS filter profiles.

Text file example:

```
192.168.2.100
172.200.1.4/16
172.16.1.2/24
172.16.8.1-172.16.8.100
2001:0db8::eade:27ff:fe04:9a01/120
2001:0db8::eade:27ff:fe04:aa01-2001:0db8::eade:27ff:fe04:ab01
```

The file contains one IPv4 or IPv6 address, address range, or subnet per line. See External resources file format for more information about the IP list formatting style.

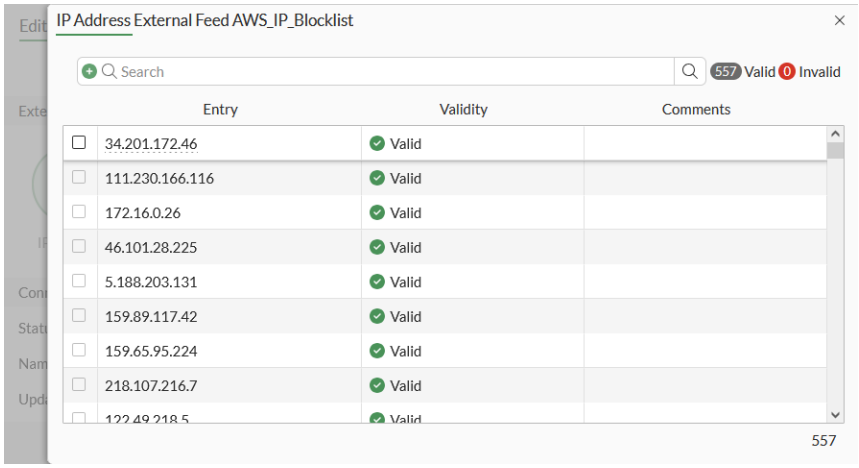
Example configuration

In this example, a list of destination IP addresses is imported using the IP address external feed. The newly created external feed is then used as a destination in a firewall policy with the action set to deny. Any traffic that passes through the FortiGate and matches the defined firewall policy will be dropped.

Please note that the URI provided in the example is solely for demonstration purposes and does not represent a reliable list of well-maintained IP addresses. It is recommended that you utilize a URI of a reputable external IP list that is regularly updated.

To configure an IP address external feed in the GUI:

1. Go to *Security Fabric* > *External Connectors* and click *Create New*.
2. In the *External Feeds* section, click *IP Address*.
3. Set the *Name* to *AWS_IP_Blocklist*.
4. Set the *Update method* to *External Feed*.
5. Set the *URL of external resource* to *https://blocklist-example.com/ip-blocklist/ip.txt*.
6. Configure the remaining settings as required, then click *OK*.
7. Edit the connector, then click *View Entries* to view the IP addresses in the feed.



To configure an IP address external feed in the CLI:

```
config system external-resource
   edit "AWS_IP_Blocklist"
      set type address
      set resource "https://blocklist-example.com/ip-blocklist/ip.txt"
      set server-identity-check {none | basic | full}
   next
end
```

To improve the security of the connection, it is recommended to enable server certificate validation (`server-identity-check`) either in basic or full mode. By default, it is set to none.

To apply an IP address external feed in a firewall policy:

1. Go to *Policy & Objects* > *Firewall Policy* and create a new policy, or edit an existing one.
2. Configure the policy fields as required.
3. In the *Destination* field, click the + and select *AWS_IP_Blocklist* from the list (in the *IP ADDRESS FEED* section).
4. Set *Action* to *DENY*.
5. Enable *Log Allowed Traffic*.
6. Click *OK*.

Applying an IP address external feed as an external IP block list in a DNS filter profile

An IP address external feed can be applied by enabling *External IP Block Lists* in a DNS filter profile. Any DNS query that passes through the FortiGate and resolves to any of the IP addresses in the external feed list will be dropped.

To configure the DNS filter profile:

1. Go to *Security Profiles* > *DNS Filter* and create a new profile, or edit an existing one.
2. Enable *External IP Block Lists*.
3. Click the + and select *AWS_IP_Blocklist* from the list.
4. Click *OK*.

To apply the DNS filter profile in a firewall policy:

1. Go to *Policy & Objects* > *Firewall Policy* and create a new policy, or edit an existing one.
2. Configure the policy fields as required.
3. Under *Security Profiles*, enable *DNS Filter* and select the profile used in the previous procedure.
4. Enable *Log Allowed Traffic*.
5. Click *OK*.

IP addresses that match the IP address external feed list will be blocked.

To view the DNS query logs:

1. Go to *Log & Report* > *Security Events* and select *DNS Query*.
2. View the log details in the GUI, or download the log file:

```
1: date=2023-02-06 time=15:06:50 eventtime=1675724810452621179 tz="-0800" logid="1501054400"
type="utm" subtype="dns" eventtype="dns-response" level="warning" vd="root" policyid=0
sessionid=555999 srcip=172.20.120.13 srcport=59602 srccountry="Reserved" srcintf="port2"
```

```
srcintfrole="undefined" dstip=172.20.120.12 dstport=53 dstcountry="Reserved" dstintf="root"
dstintfrole="undefined" proto=17 profile="default" xid=24532 qname="dns.google" qtype="A"
qtypeval=1 qclass="IN" ipaddr="208.91.112.55" msg="Domain was blocked because it is in the
domain-filter list" action="redirect" domainfilteridx=0 domainfilterlist="AWS_IP_Block_list"
```

Applying an IP address external feed in a local-in policy

An IP address external feed can be applied as a source or destination in a local-in policy.

In this example, a previously created IP address external feed named *AWS_IP_Blocklist* is used as a source address in a local-in-policy. Any traffic originating from any of the IP addresses in the external feed list and destined for the FortiGate will be dropped.

To apply an IP address external feed in a local-in policy:

```
config firewall local-in-policy
    edit 1
        set intf "any"
        set srcaddr "AWS_IP_Blocklist"
        set dstaddr "all"
        set service "ALL"
        set schedule "always"
    next
end
```

To test the configuration:

1. From one of the IP addresses listed in IP address external feed (in this case 172.16.200.2), start a continuous ping to port1:

   ```
   ping 172.16.200.1 –t
   ```

2. On the FortiGate, enable debug flow:

   ```
   # diagnose debug flow filter addr 172.16.200.2
   # diagnose debug flow filter proto 1
   # diagnose debug enable
   # diagnose debug flow trace start 10
   ```

3. The output of the debug flow shows that traffic is dropped by local-in policy 1:

   ```
   id=65308 trace_id=11 func=print_pkt_detail line=5939 msg="vd-root:0 received a packet(proto=1,
   172.16.200.2:0->172.16.200.1:2048) tun_id=0.0.0.0 from port1. type=8, code=0, id=0, seq=0."
   id=65308 trace_id=11 func=init_ip_session_common line=6121 msg="allocate a new session-
   0002f318, tun_id=0.0.0.0"
   id=65308 trace_id=11 func=__vf_ip_route_input_rcu line=2012 msg="find a route: flag=80000000
   gw-0.0.0.0 via root"
   id=65308 trace_id=11 func=fw_local_in_handler line=545 msg="iprope_in_check() check failed on
   policy 1, drop"
   ```

Domain name external feed

A domain name external feed is a dynamic list that contains domains and periodically updates from an external server. The list is stored in a text file format on an external server. After the FortiGate imports this list, it becomes available as a category in the *Remote Categories* group of DNS filter profiles that can be used to block or monitor domains matching this category. Multiple custom categories can be defined by creating a domain name external feed for each category.

Text file example:

```
mail.*.example.com
*-special.example.com
www.*example.com
example.com
```

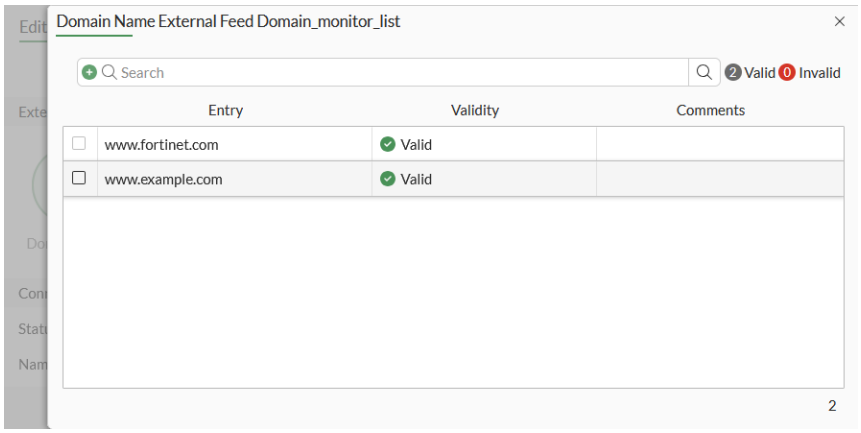
The file contains one domain name per line. See External resources file format for more information about the domain list formatting style.

Example configuration

In this example, a list of domain names is imported using the domain name external feed. The newly created external feed is set to monitor in the DNS filter profile, and the DNS filter profile is applied to a firewall policy. Any traffic that passes through the FortiGate and matches any of the domain names in the external feed list will be monitored.

To configure a domain name external feed in the GUI:

1. Go to *Security Fabric* > *External Connectors* and click *Create New*.
2. In the *External Feeds* section, click *Domain Name*.
3. Set the *Name* to *Domain_monitor_list*.
4. Set the *Update method* to *External Feed*.
5. Set the *URL of external resource* to *https://192.168.10.13/external_domain_list.txt*.
6. Configure the remaining settings as required, then click *OK*.
7. Edit the connector, then click *View Entries* to view the domain names in the feed (fortinet.com and example.com).



To configure a domain name external feed in the CLI:

```
config system external-resource
    edit "Domain_monitor_list”
        set type domain
        set category 194
        set resource "http://192.168.10.13/external_domain_list.txt"
        set server-identity-check {none | basic | full}
    next
end
```

To improve the security of the connection, it is recommended to enable server certificate validation (`server-identity-check`) either in basic or full mode. By default, it is set to none.

To apply a domain name external feed in a DNS filter profile:

1. Go to *Security Profiles* > *DNS Filter* and create a new DNS filter profile, or edit an existing one.
2. Enable *FortiGuard Category Based Filter*.
3. In the *Remote Categories* group, set the action for the *Domain_monitor_list* category to *Monitor*.
4. Configure the remaining settings as needed, then click *OK*.

Selecting the *Allow* action for the *FortiGuard Category Based Filter* does not actually allow the category. It merely implies that no filter has been applied.

We recommend avoid using the *Allow* action for remote categories, as it will not override the original action specified in the *FortiGuard Category Based Filter*.

The *Monitor* and *Block* actions for remote categories can override the original action specified in the *FortiGuard Category Based Filter*.

To apply the DNS filter profile in a firewall policy:

1. Go to *Policy & Objects* > *Firewall Policy* and create a new policy, or edit an existing one.
2. Configure the policy fields as required.
3. Under *Security Profiles*, enable *DNS Filter* and select the profile used in the previous procedure.
4. Enable *Log Allowed Traffic*.
5. Click *OK*.

Domains that match the domain external feed list are rated as domain external feed, overriding their original domain rating. Use the FortiGuard Secure DNS Service to check the original category of a domain name.

To view the DNS query logs:

1. Go to *Log & Report* > *Security Events* and select *DNS Query*.
2. View the log details in the GUI, or download the log file:

```
1: date=2023-02-03 time=10:44:16 eventtime=1675449856658521042 tz="-0800" logid="1501054802"
type="utm" subtype="dns" eventtype="dns-response" level="notice" vd="root" policyid=0
```

```
sessionid=265870 srcip=172.20.120.13 srcport=59662 srccountry="Reserved" srcintf="port2"
srcintfrole="undefined" dstip=172.20.120.12 dstport=53 dstcountry="Reserved" dstintf="root"
dstintfrole="undefined" proto=17 profile="default" xid=35624 qname="example.com" qtype="A"
qtypeval=1 qclass="IN" ipaddr="93.184.216.34" msg="Domain is monitored" action="pass" cat=194
catdesc="Domain_monitor_list"
```

```
2: date=2023-02-03 time=10:44:08 eventtime=1675449848683418535 tz="-0800" logid="1501054802"
type="utm" subtype="dns" eventtype="dns-response" level="notice" vd="root" policyid=0
sessionid=265537 srcip=172.20.120.13 srcport=57434 srccountry="Reserved" srcintf="port2"
srcintfrole="undefined" dstip=172.20.120.12 dstport=53 dstcountry="Reserved" dstintf="root"
dstintfrole="undefined" proto=17 profile="default" xid=31194 qname="fortinet.com" qtype="A"
qtypeval=1 qclass="IN" ipaddr="3.1.92.70, 52.220.222.172" msg="Domain is monitored"
action="pass" cat=194 catdesc="Domain_monitor_list"
```

Note that fortinet.com, which was originally in the Information Technology category with a default action set to allow in the *FortiGuard Category Based Filter*, has been overridden by the monitor action of the remote category.

MAC address external feed

A MAC address external feed is a dynamic list that contains MAC addresses, MAC ranges, and MAC OUIs. The list is periodically updated from an external server and stored in text file format on an external server. After the FortiGate imports this list, it can be used as a source in firewall policies, proxy policies, and ZTNA rules. For policies in transparent mode or virtual wire pair policies, the MAC address external feed can be used as a source or destination address.

Text file example:

```
01:01:01:01:01:01
01:01:01:01:01:01-01:01:02:50:20:ff
8c:aa:b5
```

The file contains one MAC address, MAC range, or MAC OUI per line. See External resources file format for more information about the MAC list formatting style.

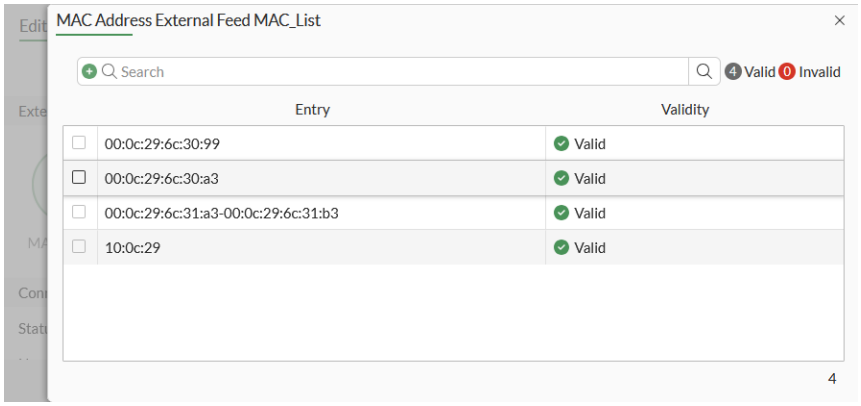
Example configuration

In this example, a list of MAC addresses is imported using the MAC address external feed. The newly created external feed is then used as a source in a firewall policy with the action set to accept. Any traffic from the client MAC addresses that match the defined firewall policy will be allowed.

To configure a MAC address external feed in the GUI:

1. Go to *Security Fabric* > *External Connectors* and click *Create New*.
2. In the *External Feeds* section, click *MAC Address*.
3. Set the *Name* to *MAC_List*.
4. Set the *Update method* to *External Feed*.

5. Set the *URL of external resource* to *http://172.16.200.55/external-resources/Ext-Resource-Type-as-Address-mac-1.txt*.
6. Configure the remaining settings as required, then click *OK*.
7. Edit the connector, then click *View Entries* to view the MAC addresses in the feed.



To configure a MAC address external feed in the CLI:

```
config system external-resource
    edit "MAC_List"
        set type mac-address
        set resource "http://172.16.200.55/external-resources/Ext-Resource-Type-as-Address-mac-
1.txt"
        set server-identity-check {none | basic | full}
    next
end
```

To improve the security of the connection, it is recommended to enable server certificate validation (`server-identity-check`) either in basic or full mode. By default, it is set to none.

To apply a MAC address external feed in a firewall policy in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy* and create a new policy, or edit an existing one.
2. Configure the policy fields as required.
3. In the *Source* field, click the *+* and select *MAC_List* from the list (in the *MAC ADDRESS FEED* section).
4. Set *Action* to *ACCEPT*.
5. Click *OK*.

To apply a MAC address external feed in a firewall policy in the CLI:

```
config firewall policy
    edit 1
        set name "MAC-traffic"
        set srcintf "port2"
        set dstintf "port1"
        set action accept
```

```
        set srcaddr "MAC_List"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set profile-protocol-options "protocol"
        set nat enable
    next
end
```

To verify the MAC addresses used in the firewall policy:

```
# diagnose sys external-mac-resource list MAC_List
MAC ranges of uuid-idx 574 (num=1)
be:d1:6b:0d:20:61-be:d1:6b:0d:20:61
```

Malware hash external feed

A malware hash external feed is a dynamic list that contains malware hashes and periodically updates from an external server. The list is stored in text file format on an external server. After the FortiGate imports this list, it is automatically used for virus outbreak prevention on antivirus profiles when *Use external malware block list* is enabled. Similar to FortiGuard outbreak prevention, the malware hash external feed is not supported in AV quick scan mode.

Text file example:

```
292b2e6bb027cd4ff4d24e338f5c48de
dda37961870ce079defbf185eeeef905 Trojan-Ransom.Win32.Locky.abfl
3fa86717650a17d075d856a41b3874265f8e9eab Trojan-Ransom.Win32.Locky.abfl
c35f705df9e475305c0984b05991d444450809c35dd1d96106bb8e7128b9082f Trojan-Ransom.Win32.Locky.abfl
```

The file contains one malware hash per line. See External resources file format for more information about the malware hash list formatting style.

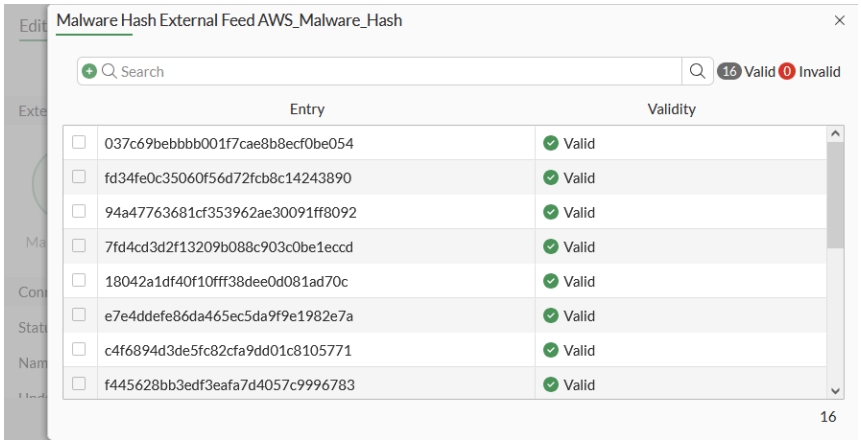
For optimal performance, do not mix different hashes in the list. Only use one MD5, SHA1, or SHA256.

Example configuration

In this example, a list of malware hashes is imported using the malware external feed. The newly created external feed is applied to an antivirus profile, and the antivirus profile is applied to a firewall policy. Any traffic that passes through the FortiGate and matches the malware hashes in the external feed list will be dropped.

To configure a malware hash external feed in the GUI:

1. Go to *Security Fabric* > *External Connectors* and click *Create New*.
2. In the *External Feeds* section, click *Malware Hash*.
3. Set the *Name* to *AWS_Malware_Hash*.
4. Set the *Update method* to *External Feed*.
5. Set the *URL of external resource* to *https://s3.us-west-2.amazonaws.com/malware.txt*.
6. Configure the remaining settings as required, then click *OK*.
7. Edit the connector, then click *View Entries* to view the hash list.



To configure a malware hash external feed in the CLI:

```
config system external-resource
    edit "AWS_Malware_Hash"
        set type malware
        set resource "https://s3.us-west-2.amazonaws.com/malware.txt"
        set server-identity-check {none | basic | full}
    next
end
```

To improve the security of the connection, it is recommended to enable server certificate validation (`server-identity-check`) either in basic or full mode. By default, it is set to none.

To apply a malware hash external feed in an antivirus profile:

1. Go to *Security Profiles* > *AntiVirus* and create a new web filter profile, or edit an existing one.
2. Enable *Use external malware block list*.
3. Click the + and select *AWS_Malware_Hash* from the list.
4. Configure the remaining settings as needed, then click *OK*.

To apply the antivirus profile in a firewall policy:

1. Go to *Policy & Objects* > *Firewall Policy* and create a new policy, or edit an existing one.
2. Configure the policy fields as required.

3. Under *Security Profiles*, enable *AntiVirus* and select the profile used in the previous procedure.
4. Set *SSL Inspection* to *deep-inspection* to inspect HTTPS traffic.
5. Enable *Log Allowed Traffic*.
6. Click *OK*.

To view the antivirus logs:

1. Go to *Log & Report > Security Events* and select *AntiVirus*.
2. View the log details in the GUI, or download the log file:

```
1: date=2023-02-03 time=15:42:41 eventtime=1675467761491047388 tz="-0800" logid="0207008212"
type="utm" subtype="virus" eventtype="malware-list" level="warning" vd="root" policyid=1
poluuid="e8b310ba-914f-51ed-9014-7b2a116f29ad" policytype="policy" msg="Blocked by local
malware list." action="blocked" service="HTTP" sessionid=293915 srcip=172.20.120.13
dstip=192.168.10.13 srcport=53515 dstport=80 srccountry="Reserved" dstcountry="Reserved"
srcintf="port2" srcintfrole="undefined" dstintf="port3" dstintfrole="wan" srcuuid="3342cb44-
9140-51ed-5dbe-8e0787bedeec" dstuuid="3342cb44-9140-51ed-5dbe-8e0787bedeec" proto=6
direction="incoming" filename="test.jpg" quarskip="Quarantine-disabled"
virus="a1a74a39788854b75d454dc9c83c612b" viruscat="File Hash" dtype="external-blocklist"
filehash="a1a74a39788854b75d454dc9c83c612b" filehashsrc="AWS_Malware_Hash"
url="http://192.168.10.13/test.jpg" profile="default" agent="curl/7.55.1" httpmethod="GET"
analyticssubmit="false" crscore=10 craction=2 crlevel="medium"
```

To verify the scanunit daemon:

```
# diagnose sys scanunit file-hash list
malware 'a1a74a39788854b75d454dc9c83c612b' vf_id 0 uuid 15752 profile 'AWS_Malware_Hash'
description ''
```

The list of external hashes has been updated.

External feed connectors per VDOM

When multi-VDOM mode is enabled, an external feed connector can be defined in global or within a VDOM. Global external feeds can be used in any VDOM, but cannot be edited within the VDOM. FortiGuard category and domain name-based external feeds have an added category number field to identify the external feed. The external feed name in global must start with `g-`. External feed names in VDOMs cannot start with `g-`.

FortiGuard category and domain name-based external feed entries must have a number assigned to them that ranges from 192 to 221. This number can be assigned to both external feed types. However, when a category number is used under a global entry, such as 192 with the name `g-cat-192`, this category number cannot be used in any other global or VDOM entries. If a category is used under a VDOM entry, such as 192 under VDOM1 with the name `cat-192`, the category 192 can be used in another VDOM or root with the name `cat-192`.

An external feed connector can only be used in profiles in the VDOM that it was created in. Global connectors can be used in all VDOMs.

Each VDOM can have a maximum of 256 external feed entries. But in total, a FortiGate can only have 511 external feed entries.

To improve the security of the connection, it is recommended to enable server certificate validation (`server-identity-check`) either in basic or full mode.

To configure a FortiGuard category external feed connector under global in the GUI:

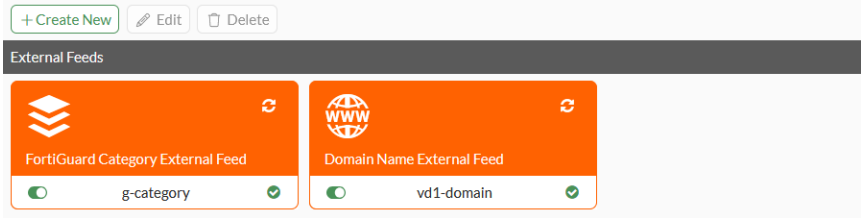
1. Go to *Security Fabric* > *External Connectors* and click *Create New*.
2. In the *External Feeds* section, click *FortiGuard Category*.
3. Enter a name that begins with `g-`.
4. Configure the other settings as needed.
5. Click *OK*.

To configure a FortiGuard category external feed connector under global in the CLI:

```
config global
    config system external-resource
        edit "g-category"
            set status enable
            set type category
            set category 192
            set comments ''
            set resource "http://172.16.200.55/external-resource-test/513-FDGCategory.txt"
            set server-identity-check {none | basic | full}
            set refresh-rate 5
        next
    end
end
```

To configure a domain name external feed connector under a VDOM in the GUI:

1. Go to *Security Fabric* > *External Connectors* and click *Create New*.
2. In the *External Feeds* section, click *Domain Name*.
3. Enter a name that does not begin with `g-`.
4. Configure the other settings as needed.
5. Click *OK*. The external feed connector created under global also appears, but it is not editable.



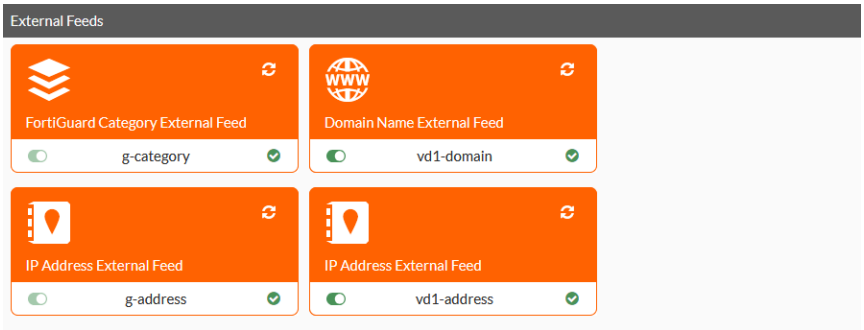
To configure a domain name external feed connector under a VDOM in the CLI:

```
config vdom
    edit vd1
        config system external-resource
            edit "vd1-domain"
```

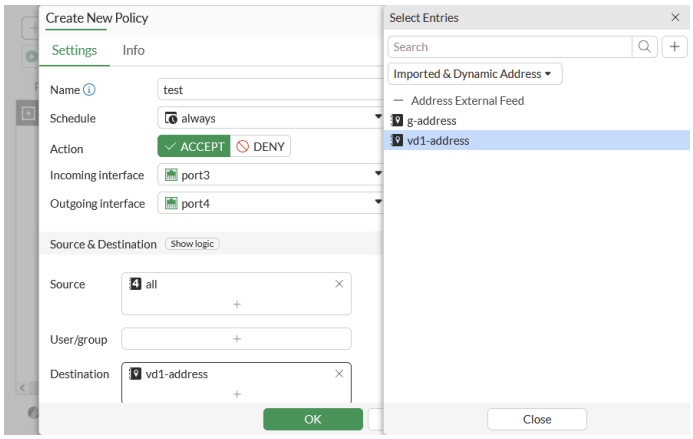
```
                set status enable
                set type domain
                set category 193
                set comments ''
                set resource "http://172.16.200.55/external-resource-test/513-Domain.txt"
                set server-identity-check {none | basic | full}
                set refresh-rate 5
            next
        end
    next
end
```

To use an IP address external feed in a policy in the GUI:

1. Configure an IP address connector in global:
   a. Go to *Security Fabric* > *External Connectors* and click *Create New*.
   b. In the *External Feeds* section, click *IP Address*.
   c. Enter a name that begins with `g-`.
   d. Configure the other settings as needed.
   e. Click *OK*.
2. Configure an IP address connector in the VDOM (vd1):
   a. Go to *Security Fabric* > *External Connectors* and click *Create New*.
   b. In the *External Feeds* section, click *IP Address*.
   c. Enter a name that does not begin with `g-`.
   d. Configure the other settings as needed.
   e. Click *OK*. The external feed connectors created under global also appear, but they are not editable.



3. Configure the firewall policy in the VDOM (vd1):
   a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
   b. For *Destination*, select *vd1-address*. Since this policy is configured under vd1, *g-address* can also be set as the destination.



c. Configure the other settings as needed.

d. Click *OK*.

To use an IP address external feed in a policy in the CLI:

1. Configure the IP address connectors:

```
config global
    config system external-resource
        edit "g-address"
            set status enable
            set type address
            set username ''
            set comments ''
            set resource "http://172.16.200.55/external-resource-test/513-IP.txt"
            set server-identity-check {none | basic | full}
            set refresh-rate 5
        next
    end
end
```

```
config vdom
    edit vd1
        config system external-resource
            edit "vd1-address"
                set status enable
                set type address
                set comments ''
                set resource "http://172.16.200.55/external-resource-test/513-IP.txt"
                set user-agent "curl/7.58.0"
                set server-identity-check {none | basic | full}
                set refresh-rate 5
            next
        end
    next
end
```

2. In the VDOM, configure a firewall policy with the external address as the destination address:

```
config vdom
    edit vd1
        config firewall policy
            edit 1
                set name "test"
                set srcintf "port10"
                set dstintf "port9"
                set srcaddr "all"
                set dstaddr "vd1-address"
                set action accept
                set schedule "always"
                set service "ALL"
                set profile-protocol-options "protocol"
                set nat enable
            next
        end
    next
end
```

Since this firewall policy is configured under `vd1`, `g-address` can also be set as the `dstaddr`.

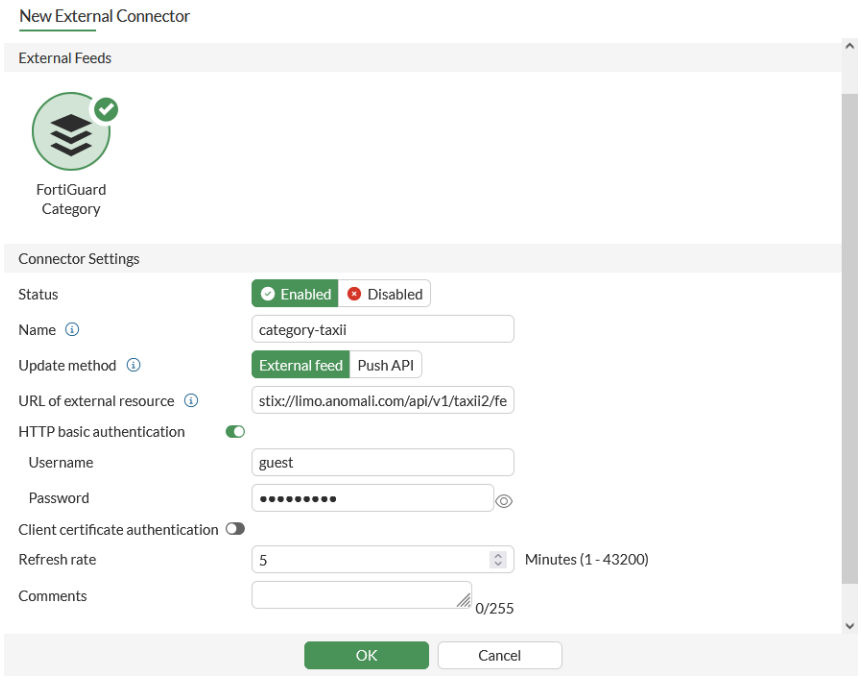
STIX format for external feeds

The FortiGate's external feeds support feeds that are in the STIX/TAXII format. Use the `stix://` prefix in the URI to denote the protocol.

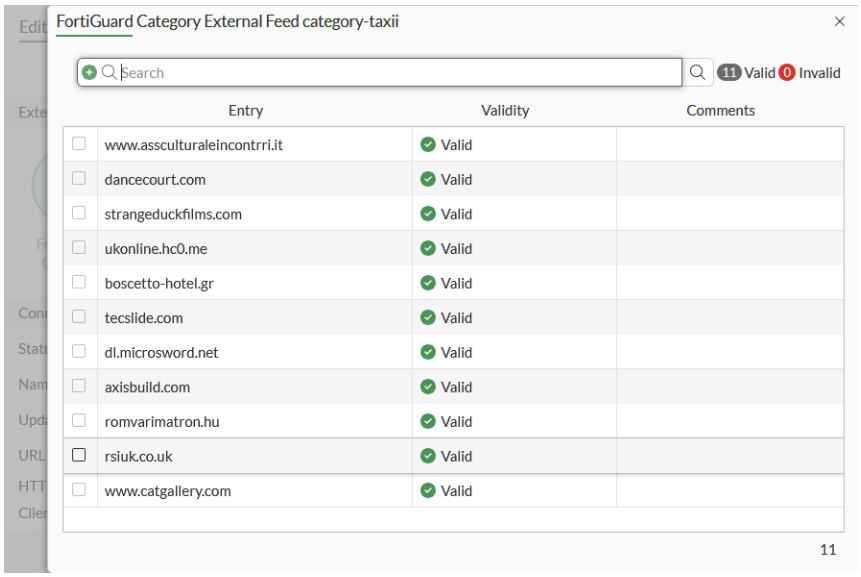
All external feeds support the STIX format. In this example, a FortiGuard Category external feed in the STIX format is configured.

To configure a FortiGuard Category external feed in the STIX format in the GUI:

1. Go to *Security Fabric* > *External Connectors* and click *Create New*.
2. Select *FortiGuard Category* from the *External Feeds* section.
3. Configure the connector:
   - *Name*: *category-taxii*
   - *Update method*: *External Feed*
   - *URL of external resource*: *stix://limo.anomali.com/api/v1/taxii2/feeds/collections/200/objects/*
   - *HTTP basic authentication*: Enable and enter the username and password, such as *guest* and *guest*.



4. Click *OK*.
5. Edit the connector, and click *View Entries* in the right side bar to view the retrieved entries.



To configure a FortiGuard Category external feed in the STIX format in the CLI:

```
config system external-resource
    edit "category-taxii"
        set category 194
        set username "guest"
        set password guest
        set resource "stix://limo.anomali.com/api/v1/taxii2/feeds/collections/200/objects/"
        set server-identity-check {none| basic | full}
```

```
        set update-method feed
    next
end
```

To improve the security of the connection, it is recommended to enable server certificate validation (`server-identity-check`) either in basic or full mode. By default, `server-identity-check` is set to none.

If the connector is used in webfilter that blocks category 194, the traffic that matches the retrieved URLs, such as *rsiuk.co.uk*, is blocked:

```
1: date=2021-10-06 time=18:07:46 eventtime=1633568867163763708 tz="-0700" logid="0316013056"
type="utm" subtype="webfilter" eventtype="ftgd_blk" level="warning" vd="vd1" policyid=1
sessionid=174974 srcip=10.1.100.12 srcport=48284 srcintf="port2" srcintfrole="undefined"
srcuuid="c6753ba2-231b-51ec-1675-090f2b5f1384" dstip=78.129.255.151 dstport=443 dstintf="port1"
dstintfrole="undefined" dstuuid="c6753ba2-231b-51ec-1675-090f2b5f1384" proto=6 service="HTTPS"
hostname="rsiuk.co.uk" profile="test" action="blocked" reqtype="direct" url="https://rsiuk.co.uk/"
sentbyte=75 rcvdbyte=0 direction="outgoing" msg="URL belongs to a denied category in policy"
method="domain" cat=194 catdesc="category-taxii"
```

Using the AusCERT malicious URL feed with an API key

In this example, a list of malicious URLs is imported from AUSCERT, an Australian not for profit organization. See AUSCERT for more information.

The FortiGuard external feed is used to import the malicious URL feed by appending the API key to the user-agent. See HTTP header on page 4701 for more information. The newly created external feed is set to block in the web filter profile, and the web filter profile is applied to a firewall policy. Any traffic that passes through the FortiGate and matches the URLs in the external feed list will be dropped, and a replacement message will be shown.

To configure the FortiGuard category external feed in the GUI:

1. Go *Security Fabric* > *External Connectors* and click *Create New*.
2. In the *External Feeds* section, select *FortiGuard Category*.
3. Configure the following settings:

Status	Enabled
Name	AusCERT_Feed
Update method	External Feed
URL of external resource	https://www.auscert.org.au/api/v1/malurl/combo-7-txt/

4. Click *OK*.
5. In the CLI, enter the following:

```
config system external-resource
    edit "AusCERT_Feed"
        set user-agent "Firefox\r\nAPI-Key:SECRETAPIKEY"
    next
end
```

6. In the GUI, edit the connector and configure the remaining settings as needed, then click *OK*.
7. Edit the connector again, and click *View Entries* in the right pane to view the URL list.

To configure the FortiGuard category external feed in the CLI:

```
config system external-resource
    edit "AusCERT_Feed"
        set category 194
        set resource "https://www.auscert.org.au/api/v1/malurl/combo-7-txt/"
        set user-agent "Firefox\r\nAPI-Key:SECRETAPIKEY"
    next
end
```

When configuring a FortiGuard category external feed in the GUI, the category is set automatically. When configuring an external feed in the CLI, the category must be set manually. The category must be unique and in the range of 192 - 221.

To improve the security of the connection, it is recommended to enable server certificate validation (`server-identity-check`) in either `basic` or `full` mode. By default, it is set to none.

To apply the FortiGuard category external feed to a web filter profile:

1. Go to *Security Profiles* > *Web Filter* and create a new web filter profile, or edit an existing one.
2. Enable *FortiGuard category based filter*.
3. In the *Remote Categories* group, set the action for the *AusCERT_Feed* category to *Block*.
4. Configure the remaining settings as needed, then click *OK*.

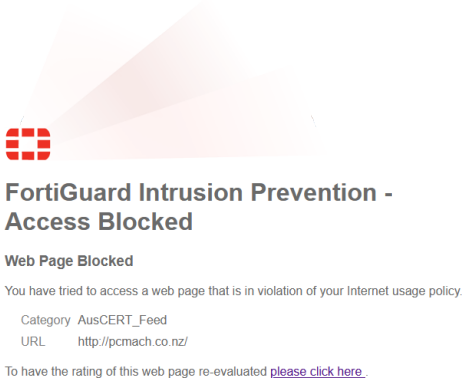
To apply the web filter profile in a firewall policy:

1. Go to *Policy & Objects* > *Firewall Policy* and create a new policy, or edit an existing one.
2. Configure the policy fields as required.
3. Under *Security Profiles*, enable *Web Filter* and select the profile used in the previous procedure.
4. Enable *Log Allowed Traffic*.
5. Click *OK*.

URLs that match the FortiGuard category external feed list are rated as the category matching the corresponding FortiGuard category external feed, overriding their original domain rating.

To verify that FortiGate is blocking URLs from the AusCERT feed list:

1. Visit one of the URLs from the AusCERT_Feed list.
   A replacement message should be shown.



2. Go to *Log & Report* > *Security Events* and select *Web Filter*.
3. View the log details in the GUI, or download the log file:

```
1: date=2023-04-11 time=14:18:02 eventtime=1681247882561766251 tz="-0700" logid="0316013056"
type="utm" subtype="webfilter" eventtype="ftgd_blk" level="warning" vd="root" policyid=1
poluuid="26540ed0-ae54-51ed-80eb-89af8af4d53f" policytype="policy" sessionid=3275
srcip=172.20.120.13 srcport=64151 srccountry="Reserved" srcintf="port2"
srcintfrole="undefined" srcuuid="3342cb44-9140-51ed-5dbe-8e0787bedeec" dstip=114.142.162.65
dstport=80 dstcountry="Australia" dstintf="port3" dstintfrole="wan" dstuuid="3342cb44-9140-
51ed-5dbe-8e0787bedeec" proto=6 httpmethod="GET" service="HTTP" hostname="pcmach.co.nz"
agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko)
Chrome/112.0.0.0 Safari/537.36" profile="default" action="blocked" reqtype="direct"
url="http://pcmach.co.nz/" sentbyte=427 rcvdbyte=0 direction="outgoing" msg="URL belongs to a
denied category in policy" ratemethod="domain" cat=194 catdesc="AusCERT_Feed"
```

Troubleshooting an external feed

In this example, the user entered the URL of external resource without the trailing slash. The following commands can be used to troubleshoot connectivity issues between a FortiGate and external resource:

```
diagnose debug app dnsproxy -1
diagnose debug app forticron -1
diagnose debug enable
```

This output shows that the DNS resolution is successful, indicating that the FortiGate has connectivity to the external server:

```
#diagnose debug app dnsproxy -1
[worker 0] dns_local_lookup()-2476: vfid=0, real_vfid=0, qname=www.auscert.org.au, qtype=1,
qclass=1, offset=36, map#=4 max_sz=512
[worker 0] dns_lookup_aa_zone()-608: vfid=0, fqdn=www.auscert.org.au
[worker 0] dns_send_request()-1398
[worker 0] dns_send_resol_request()-1234: orig id: 0xa002 local id: 0xa002
domain=www.auscert.org.au
```

```
[worker 0] dns_find_best_server()-595: found server: 96.45.46.46
…
id:0xa002 domain=www.auscert.org.au active
```

This output shows that the requested resource was missing a trailing slash:

```
#diagnose debug app forticron -1
fcron_timer_func()-23: Timer ext_upd fired
6745-before-init: fd=-1 name='ext-root.AusCERT_Feed' http_1=0 loc=0 state=send.body info=0-DNS
fail chunk=0 content-0=0 etag=0 csum=0 done=0 closed=0
    sync-0(len=0 note=0 err=0) buf-1(sz=8192 data=179 free=8013 pos=0 end=179 max=10485760)
6745-init-as: fd=-1 name='ext-root.AusCERT_Feed' http_1=0 loc=0 state=send.body info=0-None
chunk=0 content-0=0 etag=0 csum=0 done=0 closed=0
    sync-0(len=0 note=0 err=0) buf-1(sz=8192 data=0 free=8192 pos=0 end=0 max=10485760)
http_request_make()-2066: HTTP request: https

GET /api/v1/malurl/combo-7-txt HTTP/1.1
Host: www.auscert.org.au
User-Agent: Firefox
API-Key: <obfuscated>
Accept: */*
Connection: close
http_request_make()-2101: fcron_get_addr(www.auscert.org.au)
__update_ext()-187: Updating EXT 'AusCERT_Feed' with HTTP
fcron_update_ext_func()-611: update ver: 0
fcron_timer_func()-32: Timer ext_upd done
fcron_epoll_before_handle()-297: BEFORE READ fd 11 handle event 0x01 read 0xc55a40 epoll events
0x01
dns_parse_resp()-102: DNS www.auscert.org.au -> 54.253.78.74
dns_parse_resp()-102: DNS www.auscert.org.au -> 13.54.251.23
…
HTTP/1.1 301 Moved Permanently
…
Location: /api/v1/malurl/combo-7-txt/
```

After adding a trailing slash to the external resource URL, the connection is now working:

```
#diagnose debug app forticron -1
fcron_timer_func()-23: Timer ext_upd fired
2832-before-init: fd=-1 name='ext-root.AusCERT_Feed' http_1=0 loc=0 state=send.header info=0-None
chunk=0 content-0=0 etag=0 csum=0 done=0 closed=0
    sync-0(len=0 note=0 err=0) buf-0(sz=0 data=0 free=0 pos=0 end=0 max=10485760)
2832-init-as: fd=-1 name='ext-root.AusCERT_Feed' http_1=0 loc=0 state=send.header info=0-None
chunk=0 content-0=0 etag=0 csum=0 done=0 closed=0
    sync-0(len=0 note=0 err=0) buf-1(sz=8192 data=0 free=8192 pos=0 end=0 max=10485760)
http_request_make()-2066: HTTP request: https

GET /api/v1/malurl/combo-7-txt/ HTTP/1.1
Host: www.auscert.org.au
User-Agent: Firefox
API-Key: <obfuscated>
Accept: */*
Connection: close
```

```
...
HTTP/1.1 200 OK
```

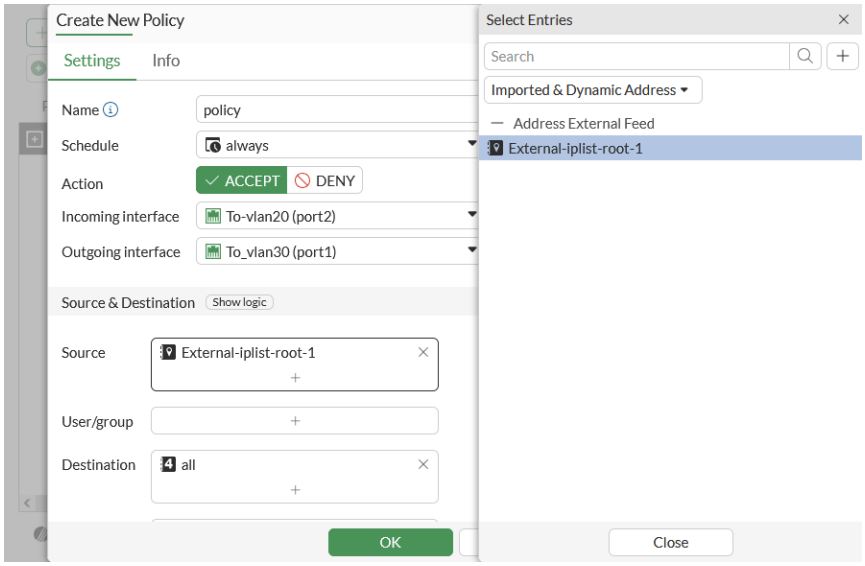
These troubleshooting commands can be used to resolve a variety of issues. they are not limited to this specific use case.

Apply external feed connectors as source addresses in central SNAT

FortiOS allows an IP address external feed to be applied as a source address in central SNAT. This enhancement allows for more dynamic and responsive network security configuration.

The IP address external feed can be applied in the GUI and the CLI:

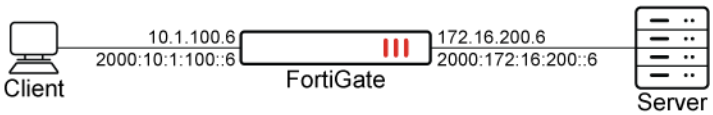
- In the GUI, select an external feed object from the *Address External Feed* section when creating and editing a policy.



- In the CLI, the IP address external feed connector can be applied when configuring the `central-snat-map`.

Example

In the following example, an external IP list external feed object will be created and used in a central SNAT map as the source address.



To apply an external feed connector in central SNAT:

1. Create an external feed IP list object:

```
config system external-resource
    edit "External-iplist-central-snat"
        set type address
        set resource "http://172.16.200.55/ip_list_test/test-external-iplist-central-snat.txt"
    next
end
```

The external feed list is as follows:

```
10.1.100.22
10.1.100.41
2000:10:1:100::22
2000:10:1:100::41
```

See IP address external feed on page 4712 for more information.

2. Apply the external feed connector in a central SNAT map as the source address:

```
config firewall central-snat-map
    edit 1
        set type ipv6
        set srcintf "port2"
        set dstintf "port1"
        set orig-addr6 "External-iplist-central-snat"
        set dst-addr6 "all"
    next
    edit 2
        set srcintf "port2"
        set dstintf "port1"
        set orig-addr "External-iplist-central-snat"
        set dst-addr "all"
    next
end
```

3. Verify that the external feed connector has been applied and taken effect:

```
# diagnose firewall iprope list 10000d
policy index=2 uuid_idx=8391 action=accept
flag (8041100): nat sport use_src pol_stats
flag3 (80): best-route
flag4 (200): port-preserve
schedule()
cos_fwd=0  cos_rev=0
group=0010000d av=00000000 au=00000000 split=00000000
host=0 chk_client_info=0x0 app_list=0 ips_view=0
misc=0
zone(1): 8 -> zone(1): 7
dest(1): 0.0.0.0-255.255.255.255, uuid_idx=8031,
source external ip pool(1): 8390
service(1):
```

```
        [0:0x0:0/(0,65535)->(0,65535)] flags:0 helper:auto

# diagnose firewall iprope6 list 10000d
policy id: 1, group: 0010000d, uuid_idx=8163
  action: accept, schedule:
  cos_fwd=0 cos_rev=0
  flag (08041100): nat sport use_src pol_stats
  flag3(00000080): best-route
  shapers: / per_ip=
  sub_groups: av 00000000 auth 00000000 split 00000000 misc 00000000
  app_list: 0 ips_view: 0
  vdom_id: 0
  zone_from(1): 8
  zone_to(1): 7
  address_dst(1):
      all uuid_idx=8045
  source external ip pool(1):
      8390
  service(1):
      [0:0x0:0/(0,65535)->(0,65535)] helper:auto
  nat(0):
  nat_64(0):
```

The source external IP pool is attached.

```
# diagnose sys external-address-resource list
List of external address resources:
name:External-iplist-central-snat, uuid-idx:8390, num of ipv4/ipv6 ranges:2/2, used:yes

# diagnose sys external-address-resource list External-iplist-central-snat
IPv4 ranges of uuid-idx 8390 (num=2)
10.1.100.22-10.1.100.22
10.1.100.41-10.1.100.41
IPv6 ranges of uuid-idx 8390 (num=2)
2000:10:1:100::22-2000:10:1:100::22
2000:10:1:100::41-2000:10:1:100::41
```

The external IP list UUID index matches.

4. Verify that sending packets from IP addresses included in the IP list will hit the central SNAT map and that SNAT will take effect:
   a. Send packets from an IPv4 address that is included in the IP list. In this example, the packets are sent from 10.1.100.41.

```
# diagnose sniffer packet any icmp 4
interfaces=[any]
filters=[icmp]
7.269689 port2 in 10.1.100.41 -> 172.16.200.55: icmp: echo request
7.269727 port1 out 172.16.200.6 -> 172.16.200.55: icmp: echo request
7.269850 port1 in 172.16.200.55 -> 172.16.200.6: icmp: echo reply
7.269861 port2 out 172.16.200.55 -> 10.1.100.41: icmp: echo reply
...
```

```
8 packets received by filter
0 packets dropped by kernel
```

SNAT will take effect. The outgoing packet is SNAT'd to the IP address of the port1 interface.

b. Send packets from an IPv4 address that is not included in the IP list. In this example, the packets are sent from 10.1.100.11.

```
# diagnose sniffer packet any icmp 4
interfaces=[any]
filters=[icmp]
2.323329 port2 in 10.1.100.11 -> 172.16.200.55: icmp: echo request
2.323362 port1 out 10.1.100.11 -> 172.16.200.55: icmp: echo request
...
4 packets received by filter
0 packets dropped by kernel
```

SNAT will not take effect.

c. Send packets from an IPv6 address that is included in the IP list. In this example, the packets are sent from 2000:10:1:100::41.

```
# diagnose sniffer packet any icmp6 4
interfaces=[any]
filters=[icmp6]
2.105798 port2 in 2000:10:1:100::41 -> 2000:172:16:200::55: icmp6: echo request seq 1
[flowlabel 0x204d4]
2.105844 port1 out 2000:172:16:200::6 -> 2000:172:16:200::55: icmp6: echo request seq 1
[flowlabel 0x204d4]
2.105959 port1 in 2000:172:16:200::55 -> 2000:172:16:200::6: icmp6: echo reply seq 1
[flowlabel 0xebd44]
2.105971 port2 out 2000:172:16:200::55 -> 2000:10:1:100::41: icmp6: echo reply seq 1
[flowlabel 0xebd44]
...
8 packets received by filter
0 packets dropped by kernel
```

SNAT will take effect. The outgoing packet is SNAT'd to the IPv6 address of the port1 interface.

d. Send packets from an IPv6 address that is not included in the IP list. In this example, the packets are sent from 2000:10:1:100::11.

```
# diagnose sniffer packet any icmp6 4
interfaces=[any]
filters=[icmp6]
1.917946 port2 in 2000:10:1:100::11 -> 2000:172:16:200::55: icmp6: echo request seq 1
1.917979 port1 out 2000:10:1:100::11 -> 2000:172:16:200::55: icmp6: echo request seq 1
...
8 packets received by filter
0 packets dropped by kernel
```

SNAT will not take effect.

Generic connector for importing addresses

This features allows for seamless integration with any third-party database using a JSON based REST API. Each JSON entry is converted into an address object on the FortiGate, which can be used in policies like any other address.

Each dynamic firewall address can parse up to 100,000 IP addresses and 3,000 MAC addresses. IPv6 addresses are not supported.

When VDOMs are enabled, a generic connector that is created in the Global VDOM must have *g-* prepended to it's name. The connector and imported addresses are synchronized to all VDOMs. A generic connector that is created in a specific VDOM is not synchronized to other VDOMs, and the address objects are only imported to that VDOM.

When VDOMs are not enabled, generic connectors cannot use the *g-* prefix in their name.

External feed update method example

In this example, the FortiGate pulls updates from an external resource: a REST API interface created using JSONBIN.io.

To create the REST API interface:

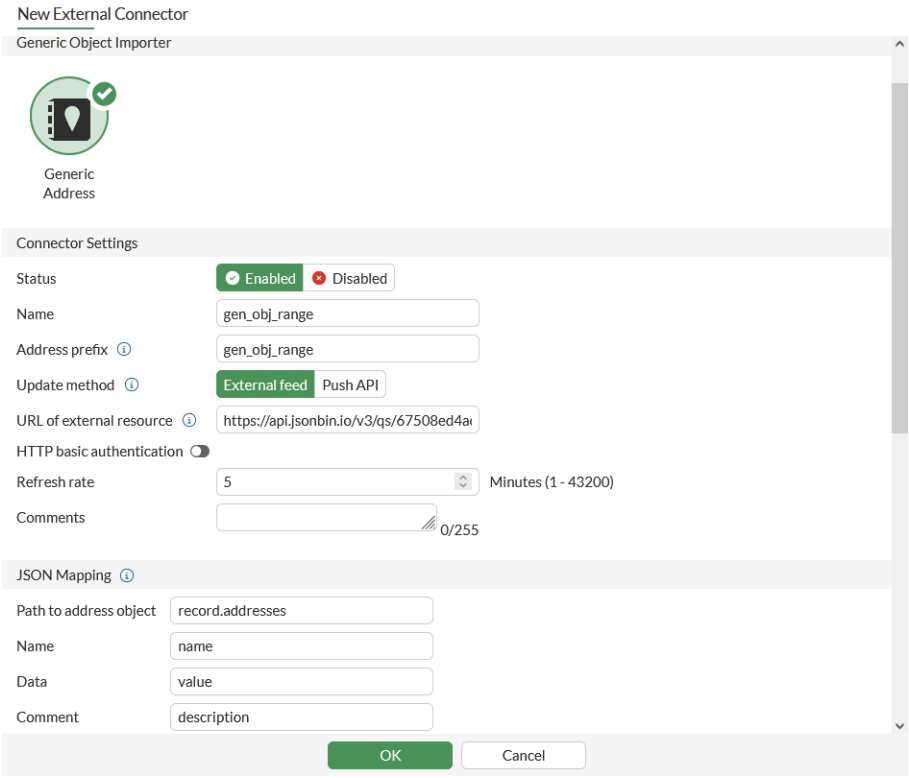
1. Go to JSONBIN.io and click Quick Create JSON.
2. Enter a name for the JSON file and select when it expires.
3. Copy in the following JSON then click *Create Bin*:

```
{
  "addresses": [
    {
      "name": "ip_address",
      "value": [
        "172.16.200.1-172.16.200.254",
        "192.168.4.1-192.168.4.254"
      ],
      "description": "generic object IP Address"
    },
    {
      "name": "mac_address",
      "value": [
        "00:0c:29:1b:40:c9",
        "00:0c:29:f6:0d:49",
        "00:0c:29:63:40:09"
      ],
      "description": "generic object MAC Address"
    }
  ]
}
```

4. Copy the generated *Access URL*.

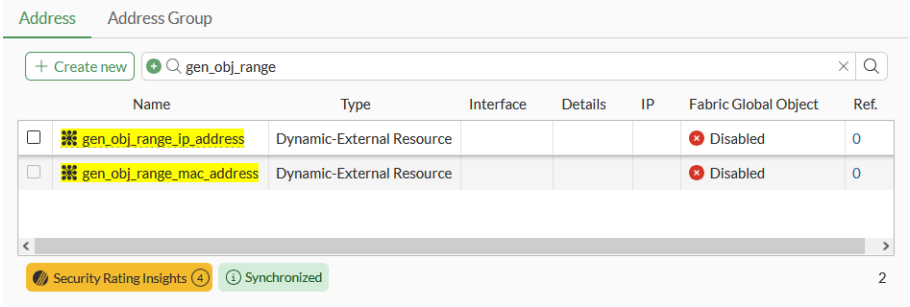
To create and test a generic connector that uses the external feed update method in the GUI:

1. On the FortiGate, go to *Security Fabric > External Connectors* and click *Create New*.
2. Enter a name for the connector, such as *gen_obj_range*.
3. Set *Update method* to *External feed*.
4. Set the *URL of external resource* to the Access URL copied from JSONBIN.io.
5. In the *JSON Mapping*, change the *Path to address object* to *record.addresses*.

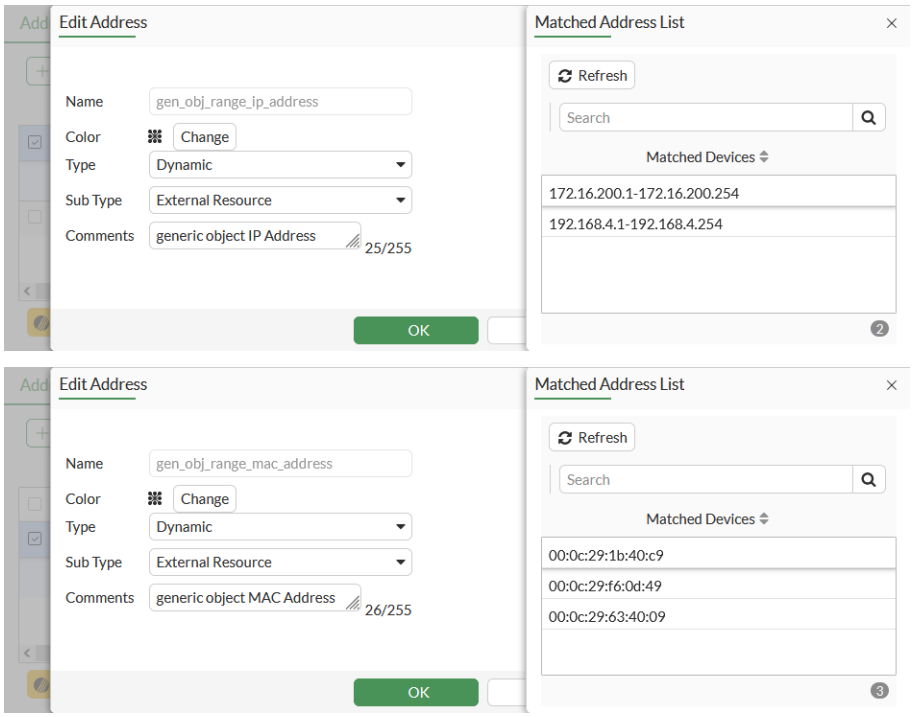


6. Click *OK*.

   The connector imports the IP and MAC addresses and automatically creates address objects on the FortiGate. The address object names are a combination of the connector name and the name of the content, for example *gen_obj_range_ip_address*.



7. Edit the address object then select *View Matched Addresses* from the right side bar, or hover over the object name then select *View Matched Addresses* in the popup message.



To create a generic connector that uses the external feed update method in the CLI:

1. Create the generic connector:

```
config system external-resource
    edit "gen_obj_range"
        set type generic-address
        set namespace "gen_obj_range"
        set object-array-path "$.record.addresses"
        set resource "https://api.jsonbin.io/v3/qs/6748a04dacd3cb34a8b09811"
    next
end
```

2. Check the matched IP addresses:

```
# show firewall address gen_obj_range_ip_address
config firewall address
    edit "gen_obj_range_ip_address"
        set uuid 711443a0-a6cc-51ef-9a0c-0db7194a28d7
        set type dynamic
        set sub-type external-resource
        set comment "generic object IP Address"
        set obj-tag "ip_address"
        set tag-type "classification"
    next
end
# diagnose firewall dynamic list gen_obj_range_ip_address
CMDB name: gen_obj_range_ip_address
gen_obj_range_ip_address: ID(88)
```

```
        RANGE(172.16.200.1-172.16.200.254)
        RANGE(192.168.4.1-192.168.4.254)
Total IP dynamic range blocks: 2.
Total IP dynamic addresses: 508.
```

3. Check the matched MAC addresses:

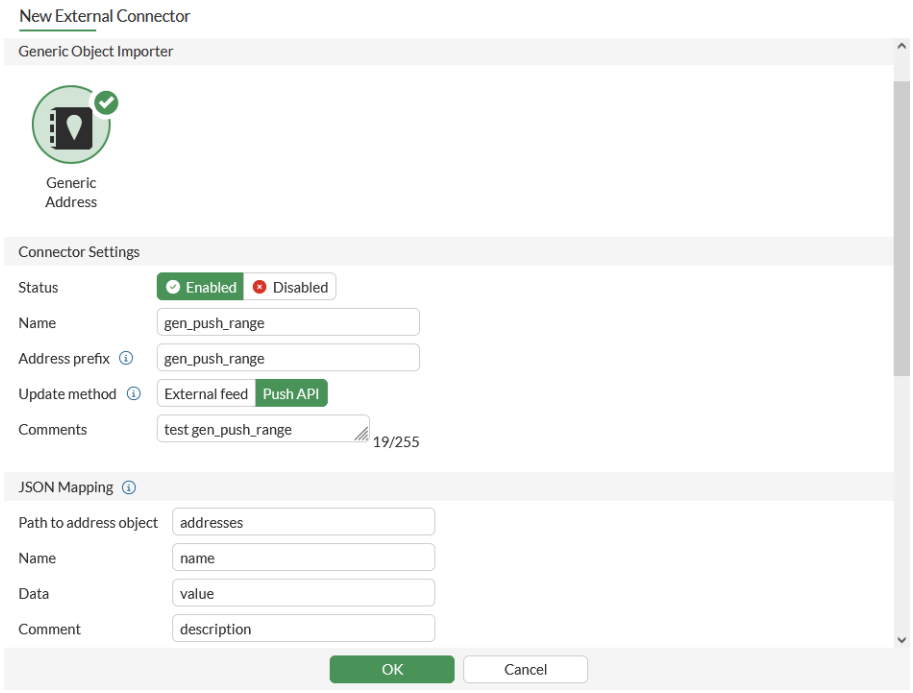
```
# show firewall address gen_obj_range_mac_address
config firewall address
    edit "gen_obj_range_mac_address"
        set uuid 7114802c-a6cc-51ef-c4d3-c98a769ccf33
        set type dynamic
        set sub-type external-resource
        set comment "generic object MAC Address"
        set obj-tag "mac_address"
        set obj-type mac
        set tag-type "classification"
    next
end
# diagnose firewall dynamic list gen_obj_range_mac_address
CMDB name: gen_obj_range_mac_address
gen_obj_range_mac_address: ID(220)
        MAC(00:0c:29:1b:40:c9)
        MAC(00:0c:29:f6:0d:49)
        MAC(00:0c:29:63:40:09)
Total MAC dynamic addresses: 3.
```

Push API update method example

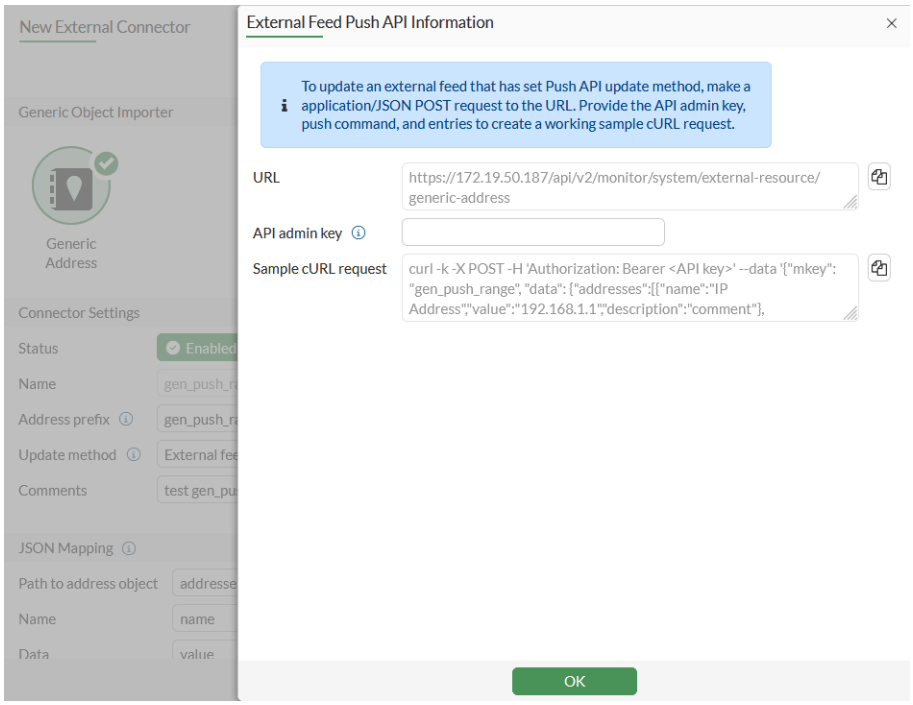
In this example, an external resource update is pushed to the FortiGate through the FortiGate's REST API. A Linux PC is connected to the FortiGate and used as the external resource.

To create and test a generic connector that uses the push API update method in the GUI:

1. On the FortiGate, go to *Security Fabric > External Connectors* and click *Create New*.
2. Enter a name for the connector, such as *gen_push_range*.
3. Set *Update method* to *Push API*.



4. Click *OK*.
   The *External Feed Push API Information* pane opens.



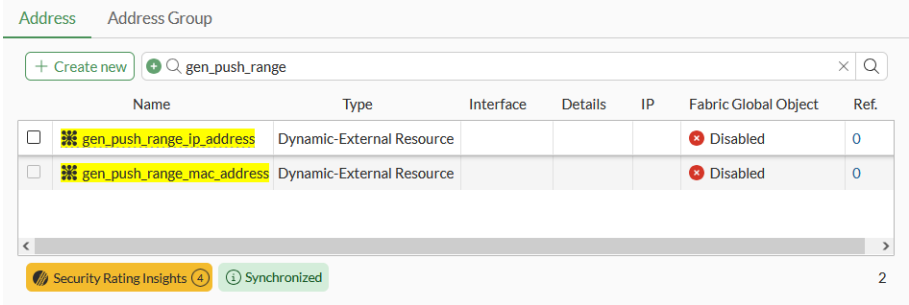
5. Copy the *Sample cURL request* and edit the entries, such as API key, IP Address, and son on.
   In this example, the cURL request is:

```
curl -k -X POST -H 'Authorization: Bearer xxxxxxxxxxx' --data '{"mkey": "gen_push_range",
"data": {"addresses":[{"name":"ip_address","value":["172.16.200.1-
```

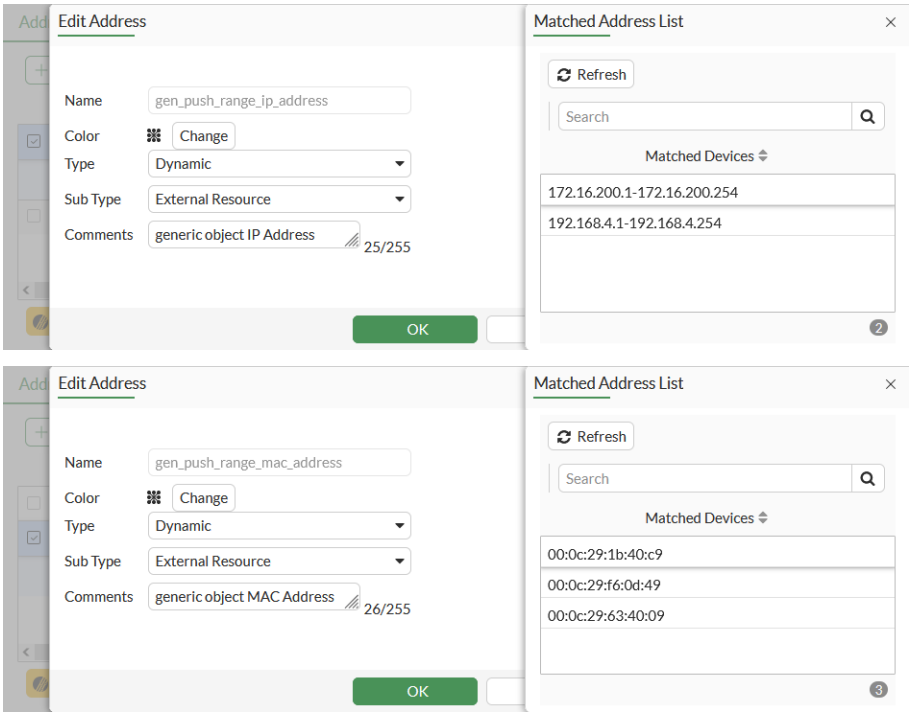
```
172.16.200.254","192.168.4.1-192.168.4.254"],"description":"generic object IP Address"},
{"name":"mac_address","value":
["00:0c:29:1b:40:c9","00:0c:29:f6:0d:49","00:0c:29:63:40:09"],"description":"generic object
MAC Address"}]}}' "https://172.16.116.210:48182/api/v2/monitor/system/external-
resource/generic-address"
```

6. Send the JSON request to the FortiGate through the Linux PC.

   The connector imports the IP and MAC addresses and automatically creates address objects on the FortiGate. The address object names are a combination of the connector name and the name of the content, for example *gen_obj_push_ip_address*.



7. Edit the address object then select *View Matched Addresses* from the right side bar, or hover over the object name then select *View Matched Addresses* in the popup message.



To create and test a generic connector that uses the push API update method in the CLI:

1. Create the generic connector:

```
config system external-resource
    edit "gen_push_range"
```

```
        set type generic-address
        set namespace "gen_push_range"
        set update-method push
        set comments "test gen_push_range"
    next
end
```

2. Send the JSON request to the FortiGate through the Linux client used in this example.

```
curl -k -X POST -H 'Authorization: Bearer xxxxxxxxxx' --data '{"mkey": "gen_push_range",
"data": {"addresses":[{"name":"ip_address","value":["172.16.200.1-
172.16.200.254","192.168.4.1-192.168.4.254"],"description":"generic object IP Address"},
{"name":"mac_address","value":
["00:0c:29:1b:40:c9","00:0c:29:f6:0d:49","00:0c:29:63:40:09"],"description":"generic object
MAC Address"}]}}' "https://172.16.116.210:48182/api/v2/monitor/system/external-
resource/generic-address"
```

3. Check the matched IP addresses:

```
# show firewall address gen_push_range_ip_address
config firewall address
    edit "gen_push_range_ip_address"
        set uuid b2012094-ac5e-51ef-354d-cd13120322c4
        set type dynamic
        set sub-type external-resource
        set comment "generic object IP Address"
        set obj-tag "ip_address"
        set tag-type "classification"
    next
end
# diagnose firewall dynamic list gen_push_range_ip_address
CMDB name: gen_push_range_ip_address
gen_push_range_ip_address: ID(254)
        RANGE(172.16.200.1-172.16.200.254)
        RANGE(192.168.4.1-192.168.4.254)
Total IP dynamic range blocks: 2.
Total IP dynamic addresses: 508.
```

4. Check the matched MAC addresses:

```
# show firewall address gen_push_range_mac_address
config firewall address
    edit "gen_push_range_mac_address"
        set uuid b2015c62-ac5e-51ef-75ef-8bc7586e5238
        set type dynamic
        set sub-type external-resource
        set comment "generic object MAC Address"
        set obj-tag "mac_address"
        set obj-type mac
        set tag-type "classification"
    next
end
```

```
# diagnose firewall dynamic list gen_push_range_mac_address
CMDB name: gen_push_range_mac_address
gen_push_range_mac_address: ID(98)
        MAC(00:0c:29:1b:40:c9)
        MAC(00:0c:29:f6:0d:49)
        MAC(00:0c:29:63:40:09)
Total MAC dynamic addresses: 3.
```

Troubleshooting

The following topics provide troubleshooting information for the Fortinet Security Fabric:

- Viewing a summary of all connected FortiGates in a Security Fabric on page 4743
- Diagnosing automation stitches on page 4746

Viewing a summary of all connected FortiGates in a Security Fabric

In downstream FortiGates, the `diagnose sys csf global` command shows a summary of all of the connected FortiGates in the Security Fabric.

To view a Security Fabric summary on a downstream FortiGate:

```
# diagnose sys csf global
Current vision:
[
  {
    "path":"FGVM01TM19000001",
    "mgmt_ip_str":"104.196.102.183",
    "mgmt_port":10403,
    "sync_mode":1,
    "saml_role":"identity-provider",
    "admin_port":443,
    "serial":"FGVM01TM19000001",
    "host_name":"admin-root",
    "firmware_version_major":6,
    "firmware_version_minor":2,
    "firmware_version_patch":0,
    "firmware_version_build":1010,
    "subtree_members":[
      {
        "serial":"FGVM01TM19000002"
      },
      {
        "serial":"FGVM01TM19000003"
```

```
      },
      {
        "serial":"FGVM01TM19000004"
      },
      {
        "serial":"FGVM01TM19000005"
      }
    ]
  },
  {
    "path":"FGVM01TM19000001:FGVM01TM19000002",
    "mgmt_ip_str":"104.196.102.183",
    "mgmt_port":10423,
    "sync_mode":1,
    "saml_role":"service-provider",
    "admin_port":443,
    "serial":"FGVM01TM19000002",
    "host_name":"Branch_Office_01",
    "firmware_version_major":6,
    "firmware_version_minor":2,
    "firmware_version_patch":0,
    "firmware_version_build":1010,
    "upstream_intf":"Branch-HQ-A",
    "upstream_serial":"FGVM01TM19000001",
    "parent_serial":"FGVM01TM19000001",
    "parent_hostname":"admin-root",
    "upstream_status":"Authorized",
    "upstream_ip":22569994,
    "upstream_ip_str":"10.100.88.1",
    "subtree_members":[
    ],
    "is_discovered":true,
    "ip_str":"10.0.10.2",
    "downstream_intf":"To-HQ-A",
    "idx":1
  },
  {
    "path":"FGVM01TM19000001:FGVM01TM19000003",
    "mgmt_ip_str":"104.196.102.183",
    "mgmt_port":10407,
    "sync_mode":1,
    "saml_role":"service-provider",
    "admin_port":443,
    "serial":"FGVM01TM19000003",
    "host_name":"Enterprise_Second_Floor",
    "firmware_version_major":6,
    "firmware_version_minor":2,
    "firmware_version_patch":0,
    "firmware_version_build":1010,
    "upstream_intf":"port3",
    "upstream_serial":"FGVM01TM19000001",
    "parent_serial":"FGVM01TM19000001",
```

```
    "parent_hostname":"admin-root",
    "upstream_status":"Authorized",
    "upstream_ip":22569994,
    "upstream_ip_str":"10.100.88.1",
    "subtree_members":[
    ],
    "is_discovered":true,
    "ip_str":"10.100.88.102",
    "downstream_intf":"port1",
    "idx":2
  },
  {
    "path":"FGVM01TM19000001:FGVM01TM19000004",
    "mgmt_ip_str":"104.196.102.183",
    "mgmt_port":10424,
    "sync_mode":1,
    "saml_role":"service-provider",
    "admin_port":443,
    "serial":"FGVM01TM19000004",
    "host_name":"Branch_Office_02",
    "firmware_version_major":6,
    "firmware_version_minor":2,
    "firmware_version_patch":0,
    "firmware_version_build":1010,
    "upstream_intf":"HQ-MPLS",
    "upstream_serial":"FGVM01TM19000001",
    "parent_serial":"FGVM01TM19000001",
    "parent_hostname":"admin-root",
    "upstream_status":"Authorized",
    "upstream_ip":22569994,
    "upstream_ip_str":"10.100.88.1",
    "subtree_members":[
    ],
    "is_discovered":true,
    "ip_str":"10.0.12.3",
    "downstream_intf":"To-HQ-MPLS",
    "idx":3
  },
  {
    "path":"FGVM01TM19000001:FGVM01TM19000005",
    "mgmt_ip_str":"104.196.102.183",
    "mgmt_port":10404,
    "sync_mode":1,
    "saml_role":"service-provider",
    "admin_port":443,
    "serial":"FGVM01TM19000005",
    "host_name":"Enterprise_First_Floor",
    "firmware_version_major":6,
    "firmware_version_minor":2,
    "firmware_version_patch":0,
    "firmware_version_build":1010,
    "upstream_intf":"port3",
```

```
    "upstream_serial":"FGVM01TM19000001",
    "parent_serial":"FGVM01TM19000001",
    "parent_hostname":"admin-root",
    "upstream_status":"Authorized",
    "upstream_ip":22569994,
    "upstream_ip_str":"10.100.88.1",
    "subtree_members":[
    ],
    "is_discovered":true,
    "ip_str":"10.100.88.101",
    "downstream_intf":"port1",
    "idx":4
  }
]
```


Diagnosing automation stitches

Diagnose commands are available to:

- Test an automation stitch
- Enable or disable log dumping for automation stitches
- Display the settings of every automation stitch
- Display statistics on every automation stitch

To test an automation stitch:

```
diagnose automation test <automation-stitch-name>
```

Example:

```
# diagnose automation test HA-failover
automation test is done. stitch:HA-failover
```

To toggle log dumping:

```
diagnose test application autod 1
```

Examples:

```
# diagnose test application autod 1
autod log dumping is enabled
```

```
# diagnose test application autod 1
autod log dumping is disabled

autod logs dumping summary:
autod dumped total:7 logs, num of logids:4
```

To display the settings for all of the automation stitches:

```
diagnose test application autod 2
```

Example:

```
# diagnose test application autod 2
csf: enabled root:yes
total stitches activated: 3

stitch: Compromised-IP-Banned
        destinations: all
        trigger: Compromised-IP-Banned

       local hit: 0 relayed to: 0 relayed from: 0
        actions:
               Compromised-IP-Banned_ban-ip type:ban-ip interval:0

stitch: HA-failover
        destinations: HA-failover_ha-cluster_25;
        trigger: HA-failover

       local hit: 0 relayed to: 0 relayed from: 0
        actions:
               HA-failover_email type:email interval:0
               subject: HA Failover
               mailto:admin@example.com;

stitch: rebooot
       destinations: all
       trigger: reboot

       local hit: 0 relayed to: 0 relayed from: 0
       actions:
               action1 type:alicloud-function interval:0
                       delay:1 required:yes
                       Account ID: id
                       Region: region
                       Function domain: fc.aliyuncs.com
                       Version: versoin
                       Service name: serv
                       Function name: funcy
                       headers:
```

To display statistic on all of the automation stitches:

```
diagnose test application autod 3
```

Example:

```
stitch: Compromised-IP-Banned
       local hit: 0 relayed to: 0 relayed from: 0
```

```
        last trigger:Wed Dec 31 20:00:00 1969
        last relay:Wed Dec 31 20:00:00 1969
        actions:
                Compromised-IP-Banned_ban-ip:
                        done: 1 relayed to: 0 relayed from: 0
                        last trigger:Wed Dec 31 20:00:00 1969
                        last relay:

stitch: HA-failover
        local hit: 0 relayed to: 0 relayed from: 0
        last trigger:Thu May 24 11:35:22 2018
        last relay:Thu May 24 11:35:22 2018
        actions:
                HA-failover_email:
                        done: 1 relayed to: 1 relayed from: 1
                        last trigger:Thu May 24 11:35:22 2018
                        last relay:Thu May 24 11:35:22 2018

stitch: rebooot
        local hit: 2 relayed to: 1 relayed from: 1
        last trigger:Fri May  3 13:30:56 2019
        last relay:Fri May  3 13:30:23 2019
        actions:
                action1
                        done: 1 relayed to: 0 relayed from: 0
                        last trigger:Fri May  3 13:30:56 2019
                        last relay:

logid2stitch mapping:
id:20103  local hit: 0 relayed to: 0 relayed from: 0
        License Expiry
        lambada

id:32138  local hit: 2 relayed to: 1 relayed from: 1
        Compromised-IP-Banned
        HA-failover
        rebooot

action run cfg&stats:
total:2 cur:0 done:1 drop:1
        email:
                flags:10
                stats: total:1 cur:0 done:1 drop:0
        fortiexplorer-notification:
                flags:1
                stats: total:0 cur:0 done:0 drop:0
        alert:
                flags:0
                stats: total:0 cur:0 done:0 drop:0
        disable-ssid:
                flags:7
                stats: total:0 cur:0 done:0 drop:0
```

```
        quarantine:
                flags:7
                stats: total:0 cur:0 done:0 drop:0
        quarantine-forticlient:
                flags:4
                stats: total:0 cur:0 done:0 drop:0
        quarantine-nsx:
                flags:4
                stats: total:0 cur:0 done:0 drop:0
        ban-ip:
                flags:7
                stats: total:0 cur:0 done:0 drop:0
        aws-lambda:
                flags:11
                stats: total:0 cur:0 done:0 drop:0
        webhook:
                flags:11
                stats: total:0 cur:0 done:0 drop:0
        cli-script:
                flags:10
                stats: total:0 cur:0 done:0 drop:0
        azure-function:
                flags:11
                stats: total:1 cur:0 done:0 drop:1
        google-cloud-function:
                flags:11
                stats: total:0 cur:0 done:0 drop:0
        alicloud-function:
                flags:11
                stats: total:0 cur:0 done:0 drop:0
```

Log and Report

Logging and reporting are useful components to help you understand what is happening on your network, and to inform you about certain network activities, such as the detection of a virus, a visit to an invalid website, an intrusion, a failed log in attempt, and myriad others.

Logging records the traffic that passes through, starts from, or ends on the FortiGate, and records the actions the FortiGate took during the traffic scanning process. After this information is recorded in a log message, it is stored in a log file that is stored on a log device (a central storage location for log messages). FortiGate supports sending all log types to several log devices, including FortiAnalyzer, FortiAnalyzer Cloud, FortiGate Cloud, and syslog servers. Approximately 5% of memory is used for buffering logs sent to FortiAnalyzer. The FortiGate system memory and local disk can also be configured to store logs, so it is also considered a log device. See Log settings and targets on page 4768 for more information.

Reports show the recorded activity in a more readable format. A report gathers all the log information that it needs, then presents it in a graphical format with a customizable design and automatically generated charts showing what is happening on the network. Reports can be generated on FortiGate devices with disk logging and on FortiAnalyzer devices.

FortiView is a more comprehensive network reporting and monitoring tool. It integrates real-time and historical data into a single view in FortiOS. For more information, see FortiView dashboard on page 166.

Performance statistics are not logged to disk. Performance statistics can be received by a syslog server or by FortiAnalyzer.

The following topics provide information about logging and reporting:

- Forward Traffic log page NEW on page 4750
- Viewing event logs on page 4753
- System Events log page on page 4755
- Security Events log page on page 4760
- Reports page on page 4764
- Log settings and targets on page 4768
- Logging to FortiAnalyzer on page 4774
- Advanced and specialized logging on page 4786
- Sample logs by log type on page 4830
- Troubleshooting on page 4854

Forward Traffic log page NEW

The *Log & Report* > *Forward Traffic* page includes a *Log Details* pane with summary panel. The summary panel shows the log timestamp, policy name, and a summary of security events with hover-enabled tooltips and access to quick actions.

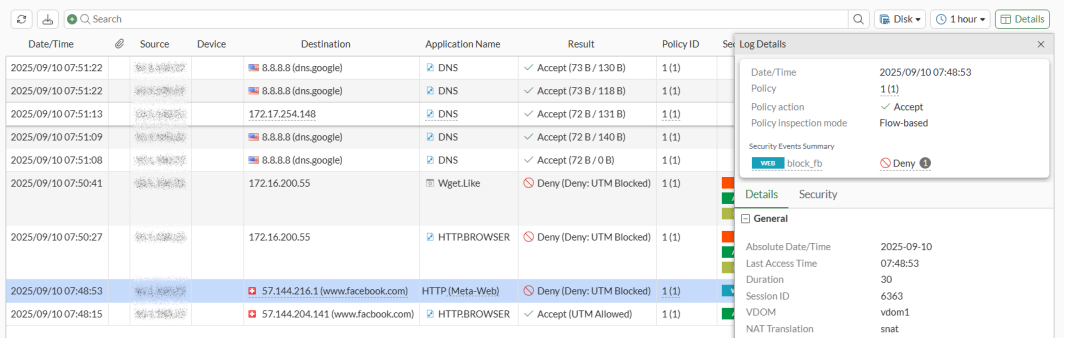
The summary panel:

- Is available for *Memory*, *Disk*, *FortiAnalyzer*, and *FortiGate Cloud* log sources.

  For *FortiGate Cloud* log source, the summary panel will always display *No security events found* because forward traffic logs from FortiGate Cloud do not contain the utmref field, which is necessary for populating security event data.
- Is only available in *Log Details* when the selected log has security events associated with it
- Only displays for non-implicit deny policy logs

To view the summary in Log Details:

1. Go to *Log & Report* > *Forward Traffic*.
2. Set the log source to *Memory*, *Disk*, or *FortiAnalyzer*.
3. Select a UTM log entry, and click *Details*. The *Log Details* pane is displayed with a summary panel at the top:

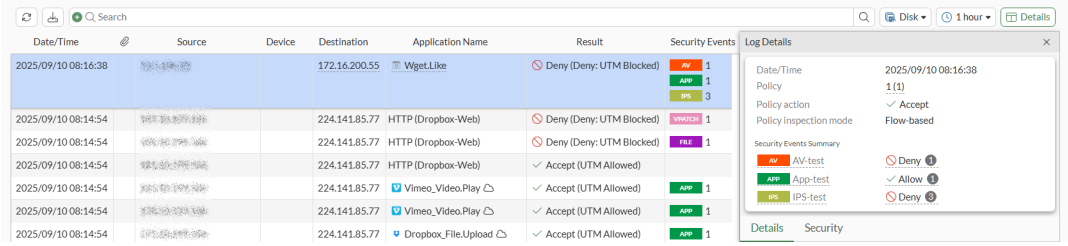


Option	Description
Date/Time	Date and time of the log entry
Policy	Name of the policy used
Policy action	Policy action (Accept or Deny)
Policy inspection mode	Policy inspection mode (Flow-based or Proxy-based)
Security Events Summary	Lists the type and name of the security profile(s) used, final action (Allow or Deny), and the number of security event logs for the event. If the same security profile has multiple security events, and if at least one (1) of the events is categorized as *Deny*, then the final action is *Deny*, and the number bubble indicates the number of deny logs out of the total logs.

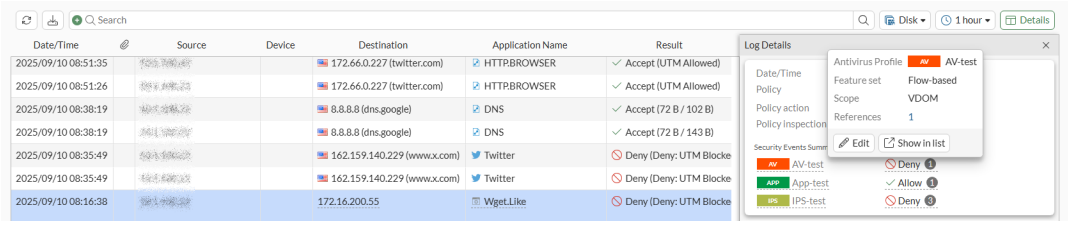
In the *Security Events Summary* section:

- When multiple security profiles (for example, APP, AV, and IPS) log security events, all security profile names and the final action (Allow or Deny) are displayed. The number bubble indicates how many security event logs are available for the event.

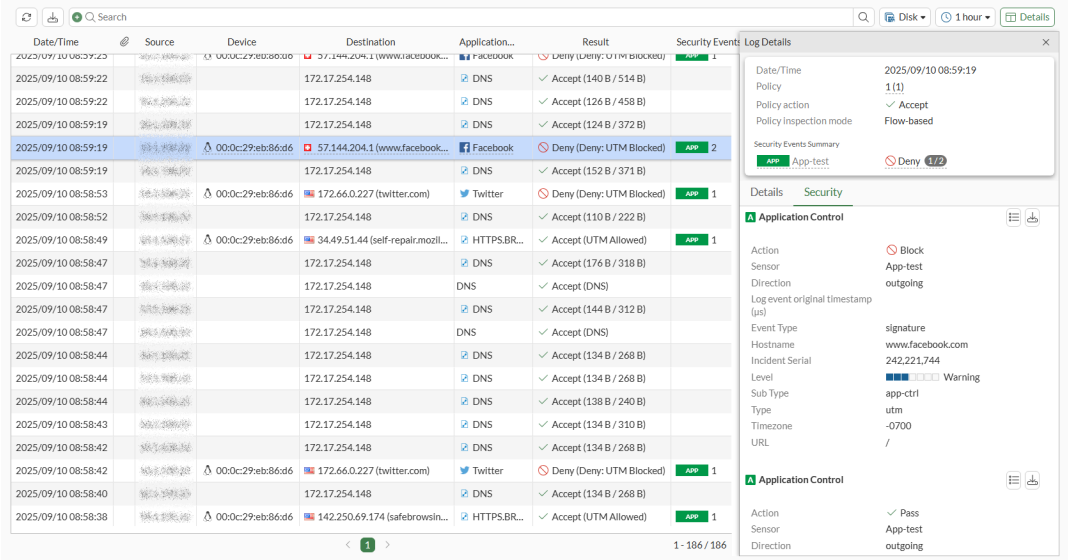
  In this example, multiple profiles (AV, APP, and IPS) logged security events, and each has one (1) security event log, except for IPS, which has three (3):



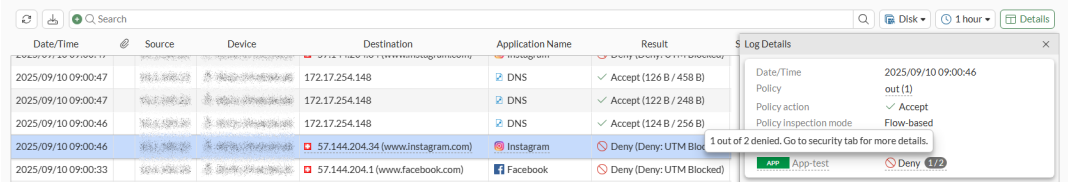
- Hover over any security profile to display a tooltip of information and access the *Edit* and *Show in list* buttons:



- When multiple security events are listed for the same security profile, the final verdict is *Deny*, if at least one of the events is categorized as *Deny*. More information about the deny logs is available on the *Security* tab.

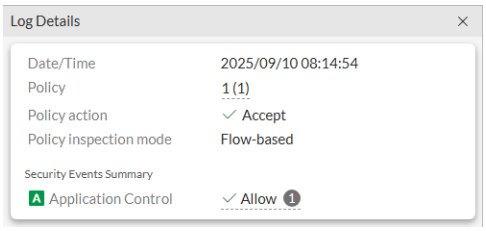


- Hover over the number bubble. A tooltip displays how the number of denied logs and *Go to security tab for more details*.



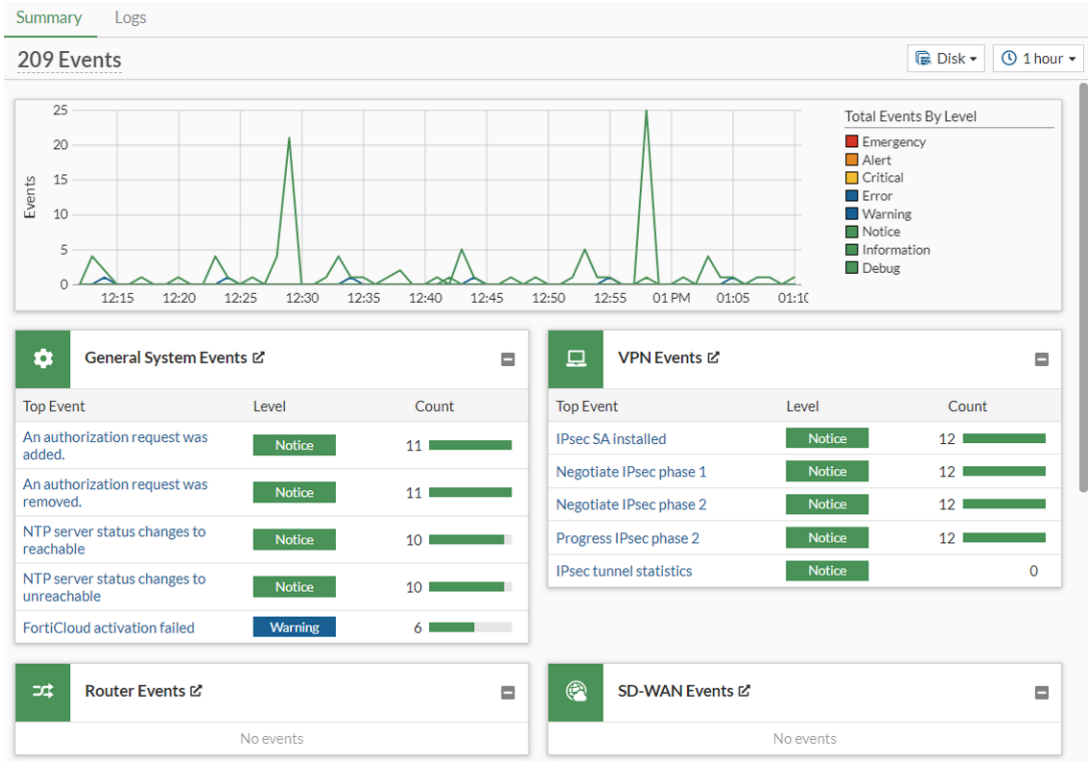
If any event has multiple logs with same security profile, but all logs are with same action (Allow or Deny), then the tooltip for the number bubble only displays *Go to security tab for more details*.

- When the security profile no longer exists, then the *Security Events Summary* section displays the generic security event name of *Application Control*.

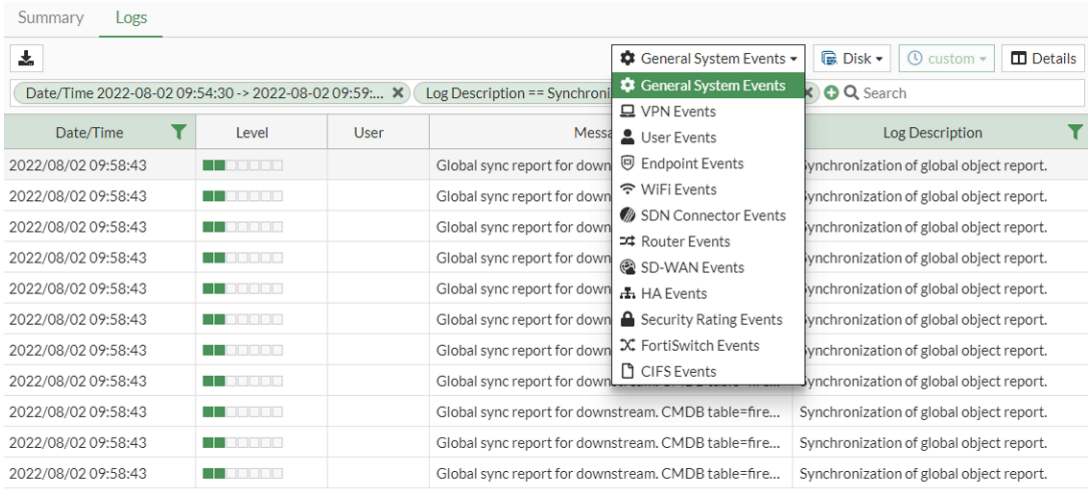


Viewing event logs

Event log subtypes are available on the *Log & Report* > *System Events* page. Not all of the event log subtypes are available by default. See System Events log page on page 4755 for more information.



When viewing event logs in the *Logs* tab, use the event log subtype dropdown list on the to navigate between event log types.



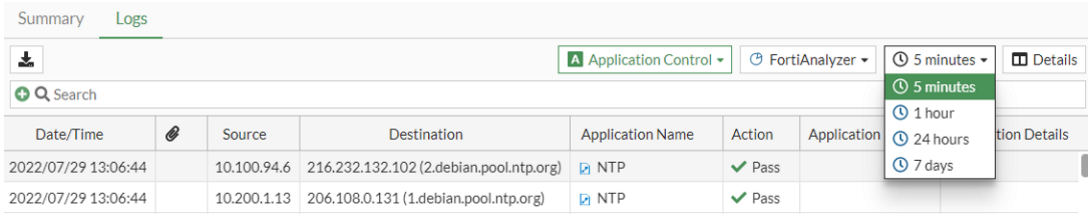
System Events	Always available.
Router Events	Always available.
VPN Events	Available when *VPN* is enabled in *System > Feature Visibility*.
SD-WAN Events	Always available.
User Events	Always available.
Endpoint Events	Available when *Endpoint Control* is enabled in *System > Feature Visibility*.
HA Events	Always available.
Security Rating Events	Always available, but logs are only generated when a Surface Attack Security Rating License is registered.
WAN Opt. & Cache Events	Available on devices with two hard disks by default. On devices with one hard disk, the disk usage must be set to `wanopt` and then *WAN Opt. & Cache* must be enabled in *System > Feature Visibility*.
WiFi Events	Available on hardware devices when *WiFi Controller* is enabled in *System > Feature Visibility*.
FortiExtender Events	Available when *FortiExtender* is enabled in *System > Feature Visibility*.
SDN Connector Events	Always available.
FortiSwitch Events	Available when *Switch Controller* is enabled in *System > Feature Visibility*.
CIFS Events	Always available.
REST API Events	Always available.

Logs can be filtered by date and time in the *Log & Report > System Events* page. The log viewer can be filtered with a custom range or with specific time frames.

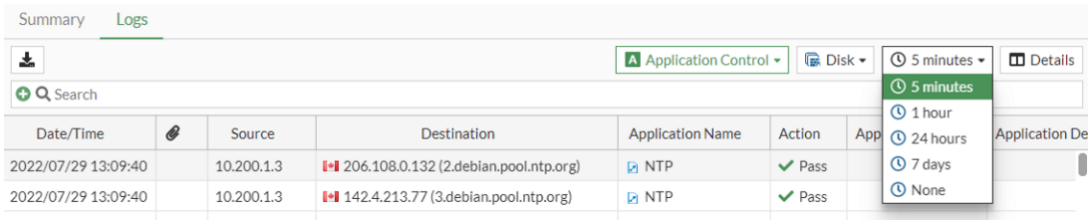
UTM logs can also be filtered by date and time in *Log & Report > Security Events*. See Security Events log page on page 4760.

The time frame available is dependent on the source:

- Logs sourced from FortiAnalyzer, FortiGate Cloud, and FortiAnalyzer Cloud have the same time frame options as FortiView (*5 minutes*, *1 hour*, *24 hours*, or *7 days*).

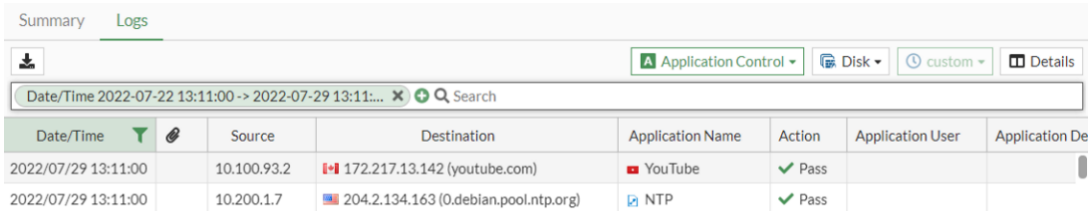


- Logs sourced from the Disk have the time frame options of *5 minutes*, *1 hour*, *24 hours*, *7 days*, or *None*.



- Logs source from Memory do not have time frame filters.

A custom time frame can be applied using the *Date/Time* filter. If the *Date/Time* filter is applied, the time frame will be disabled and set to *custom*.



Time frame settings for each *Log & Report* page are independent of each other. For example, if you change the time frame on the *System Events* page, the time frame will be different than that of the *Security Events* page unless it is also changed to match.

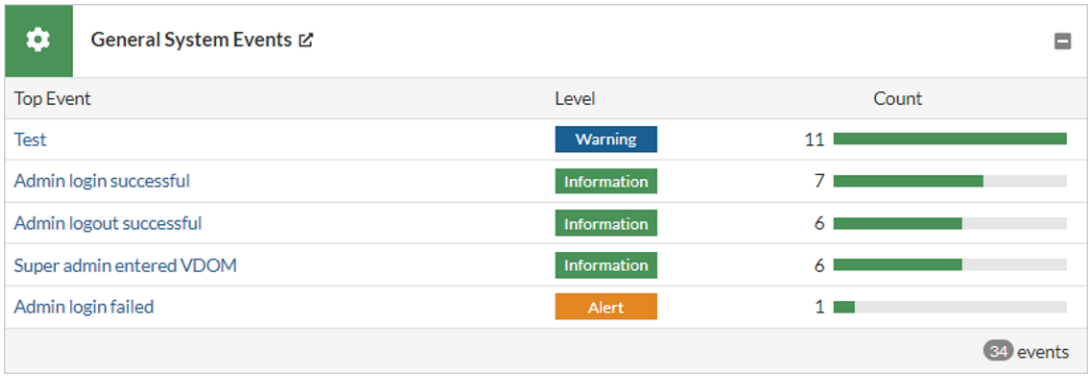
System Events log page

The *Log & Report* > *System Events* page includes:

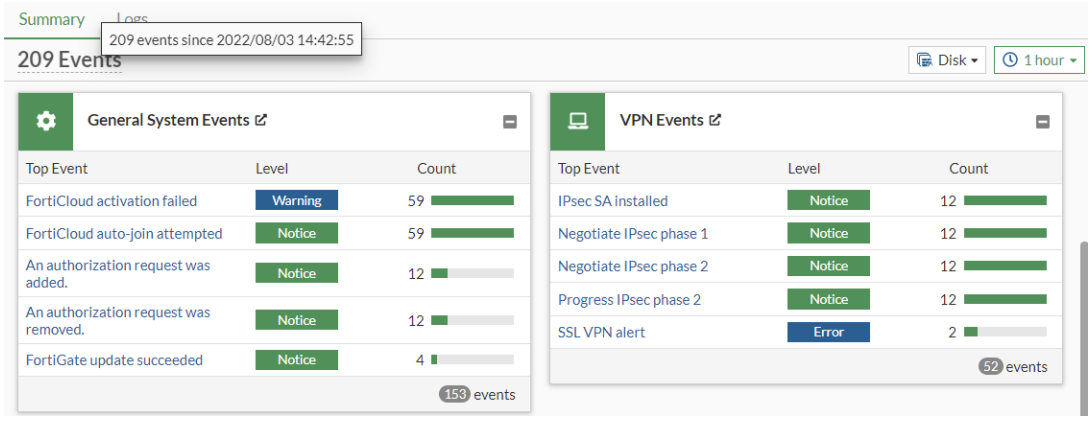
- A *Summary* tab that displays the top five most frequent events in each type of event log and a line chart to show aggregated events by each severity level. Clicking on a peak in the line chart will display the specific event count for the selected severity level.
- A *Logs* tab that displays individual, detailed log views for event type.

The *Summary* tab includes the following:

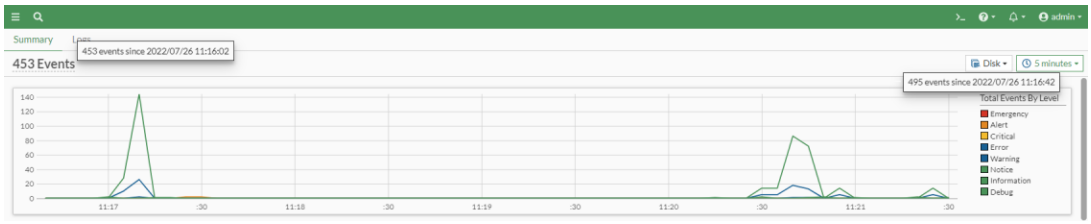
- Event list footers show a count of the events that relate to the type.



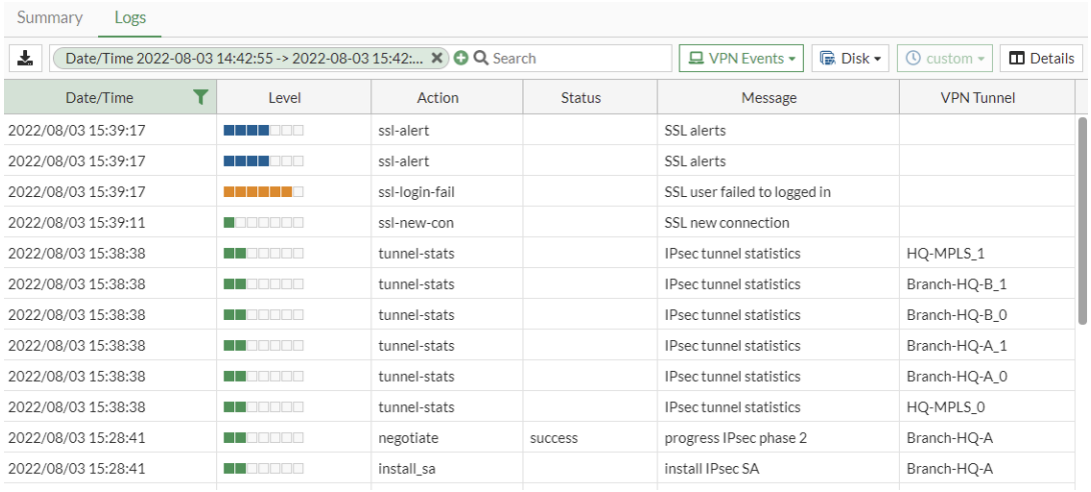
- A count of the total events is shown at the top of the *Summary*. Hovering over the count shows the number of events with a time stamp.



- Hovering over the *Total Events By Level* shows the shows the number of events with a time stamp.



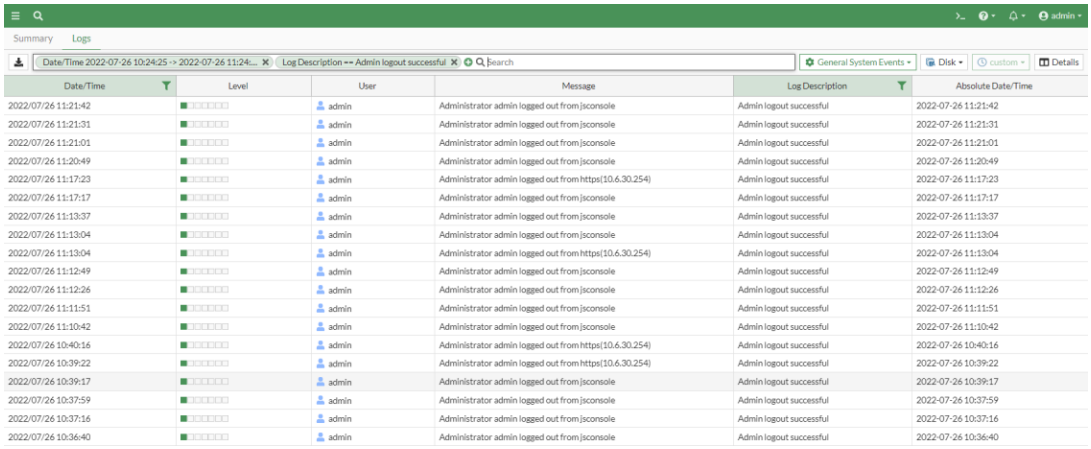
- Clicking on any event type title opens the *Logs* page for that event type filtered by the selected time span. For example, clicking *VPN Events* opens the following page:



Date/Time	Level	Action	Status	Message	VPN Tunnel
2022/08/03 15:39:17		ssl-alert		SSL alerts	
2022/08/03 15:39:17		ssl-alert		SSL alerts	
2022/08/03 15:39:17		ssl-login-fail		SSL user failed to logged in	
2022/08/03 15:39:11		ssl-new-con		SSL new connection	
2022/08/03 15:38:38		tunnel-stats		IPsec tunnel statistics	HQ-MPLS_1
2022/08/03 15:38:38		tunnel-stats		IPsec tunnel statistics	Branch-HQ-B_1
2022/08/03 15:38:38		tunnel-stats		IPsec tunnel statistics	Branch-HQ-B_0
2022/08/03 15:38:38		tunnel-stats		IPsec tunnel statistics	Branch-HQ-A_1
2022/08/03 15:38:38		tunnel-stats		IPsec tunnel statistics	Branch-HQ-A_0
2022/08/03 15:38:38		tunnel-stats		IPsec tunnel statistics	HQ-MPLS_0
2022/08/03 15:28:41		negotiate	success	progress IPsec phase 2	Branch-HQ-A
2022/08/03 15:28:41		install_sa		install IPsec SA	Branch-HQ-A

- Clicking on any event entry opens the *Logs* page for that event type filtered by the selected time span and log description.

  For example, in the *General System Events* box, clicking *Admin logout successful* opens the following page:

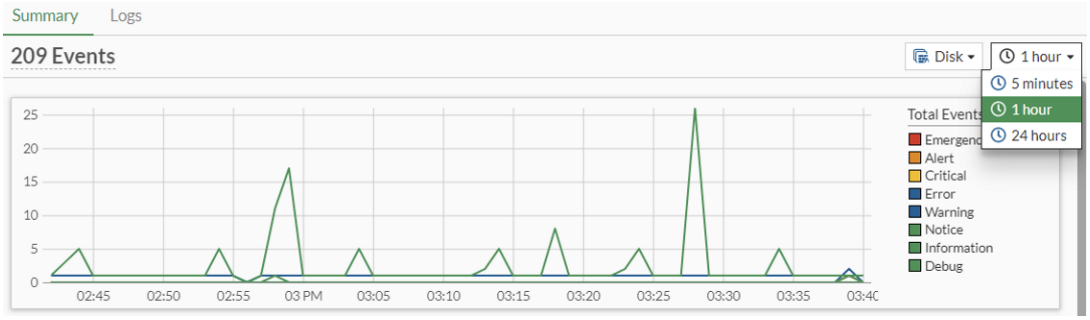


Date/Time	Level	User	Message	Log Description	Absolute Date/Time
2022/07/26 11:21:42		admin	Administrator admin logged out from jsconsole	Admin logout successful	2022-07-26 11:21:42
2022/07/26 11:21:31		admin	Administrator admin logged out from jsconsole	Admin logout successful	2022-07-26 11:21:31
2022/07/26 11:21:01		admin	Administrator admin logged out from jsconsole	Admin logout successful	2022-07-26 11:21:01
2022/07/26 11:20:49		admin	Administrator admin logged out from jsconsole	Admin logout successful	2022-07-26 11:20:49
2022/07/26 11:17:23		admin	Administrator admin logged out from https(10.6.30.254)	Admin logout successful	2022-07-26 11:17:23
2022/07/26 11:17:17		admin	Administrator admin logged out from jsconsole	Admin logout successful	2022-07-26 11:17:17
2022/07/26 11:13:37		admin	Administrator admin logged out from jsconsole	Admin logout successful	2022-07-26 11:13:37
2022/07/26 11:13:04		admin	Administrator admin logged out from jsconsole	Admin logout successful	2022-07-26 11:13:04
2022/07/26 11:13:04		admin	Administrator admin logged out from https(10.6.30.254)	Admin logout successful	2022-07-26 11:13:04
2022/07/26 11:12:49		admin	Administrator admin logged out from jsconsole	Admin logout successful	2022-07-26 11:12:49
2022/07/26 11:12:26		admin	Administrator admin logged out from jsconsole	Admin logout successful	2022-07-26 11:12:26
2022/07/26 11:11:51		admin	Administrator admin logged out from jsconsole	Admin logout successful	2022-07-26 11:11:51
2022/07/26 11:10:42		admin	Administrator admin logged out from jsconsole	Admin logout successful	2022-07-26 11:10:42
2022/07/26 10:40:16		admin	Administrator admin logged out from https(10.6.30.254)	Admin logout successful	2022-07-26 10:40:16
2022/07/26 10:39:22		admin	Administrator admin logged out from https(10.6.30.254)	Admin logout successful	2022-07-26 10:39:22
2022/07/26 10:39:17		admin	Administrator admin logged out from jsconsole	Admin logout successful	2022-07-26 10:39:17
2022/07/26 10:37:59		admin	Administrator admin logged out from jsconsole	Admin logout successful	2022-07-26 10:37:59
2022/07/26 10:37:16		admin	Administrator admin logged out from jsconsole	Admin logout successful	2022-07-26 10:37:16
2022/07/26 10:36:40		admin	Administrator admin logged out from jsconsole	Admin logout successful	2022-07-26 10:36:40

Disk logging and historical FortiView must be enabled for the *Summary* tab to display valid data. See Log settings and targets on page 4768 for more information.

A FortiOS Event Log trigger can be created using the shortcut on any *Logs* tab. Select a log, then right-click and select *Create Automation Trigger*. See System Events page shortcut on page 4520 for more information.

A time frame can be selected from the dropdown.



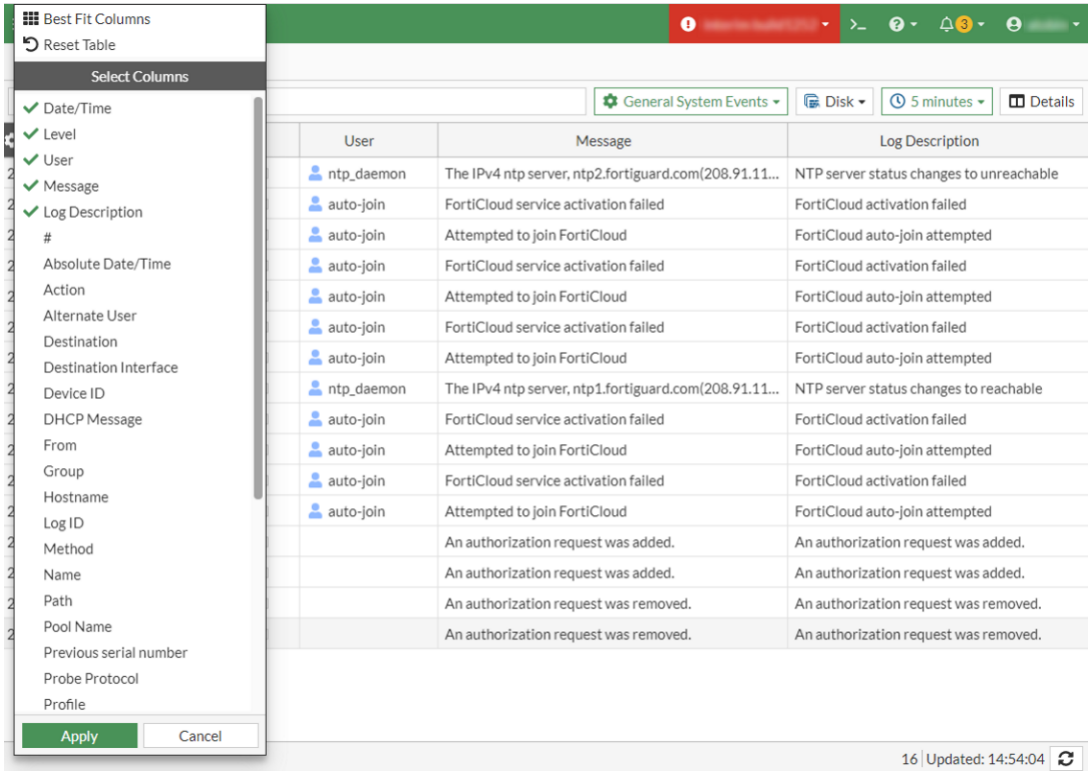
The line chart will display all of the system events, and the non-empty event cards will list up to five *Top Event* entries within the time frame set.

> Data is retrieved from FortiView with the *5 minutes* range updated first. When selecting either the *1 hour* or *24 hours* time range, there may be a delay to update *Top Event* entries.

Up to 100 *Top Event* entries can be listed in the CLI using the `diagnose fortiview result event-log` command.

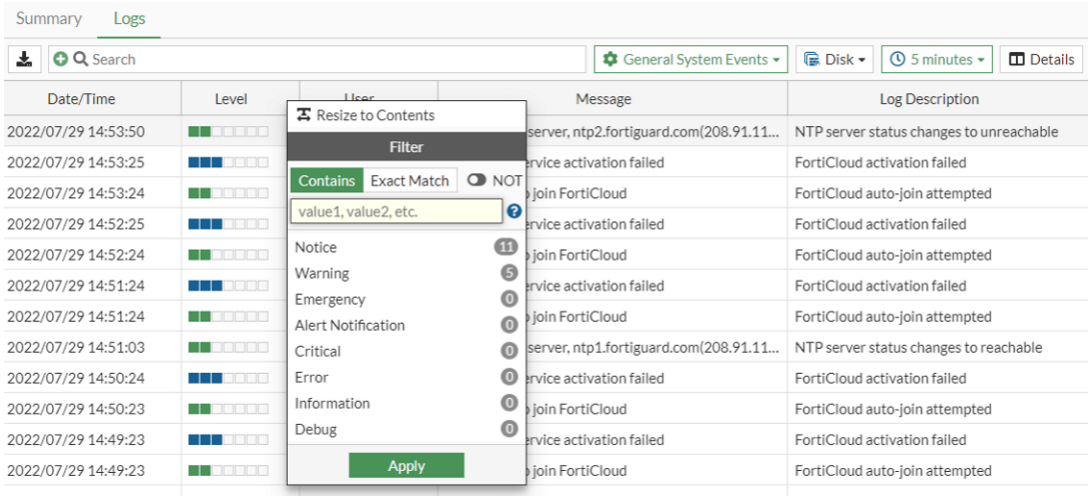
To view filtered log information:

1. Go to *Log & Report* > *System Events*.
2. Select the *Logs* tab.
3. Hover over the leftmost column and click the gear icon. A list of column you can filter is displayed.

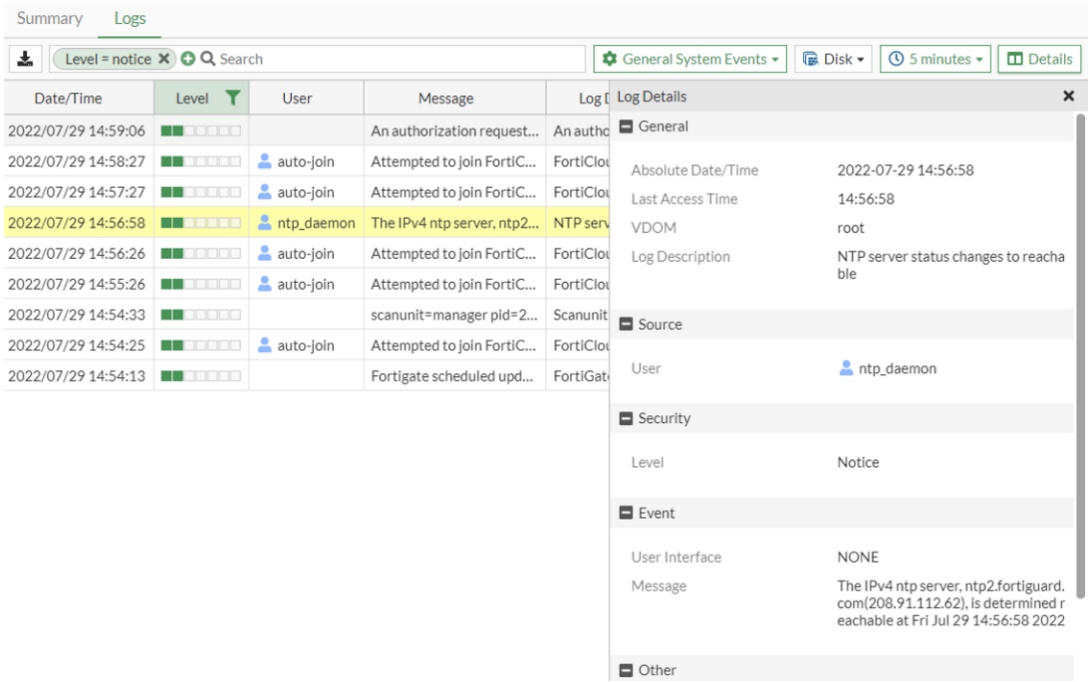


4. Select the columns you want displayed.
5. Click *Apply*. The selected columns are displayed.

6. Click the filter icon for the column you want to filter. The filter dialog is displayed and the number of logs for each filter type is listed.



7. Select the filters you want and click *Apply*. The logs that match the set filters are displayed and the filter is listed in the search bar.
8. Select the log you want to see more information on.
9. Click *Details*. The *Log Details* pane is displayed.



To list system events in the CLI:

```
# diagnose fortiview result event-log

    data(1646760000-1646846401):
    0). subtype-ha | eventname-HA device interface failed | level-warning | count-1 |
    1). subtype-system | eventname-DHCP statistics | level-information | count-40 |
```

```
    2). subtype-system | eventname-Super admin left VDOM | level-information | count-13 |
    3). subtype-system | eventname-Admin performed an action from GUI | level-warning | count-5 |
    4). subtype-system | eventname-Super admin entered VDOM | level-information | count-4 |
    5). subtype-system | eventname-Global setting changed | level-notice | count-3 |
    6). subtype-system | eventname-Attribute configured | level-information | count-2 |
    7). subtype-system | eventname-Clear active sessions | level-warning | count-2 |
    8). subtype-system | eventname-Disk log rolled | level-notice | count-2 |
    9). subtype-system | eventname-Log rotation requested by FortiCron | level-notice | count-1 |
    10). subtype-system | eventname-Report generated successfully | level-notice | count-1 |
    11). subtype-system | eventname-Test | level-warning | count-1 |
    12). subtype-system | eventname-VDOM added | level-notice | count-1 |
    13). subtype-user | eventname-Authentication failed | level-notice | count-1 |
    14). subtype-user | eventname-Authentication lockout | level-warning | count-1 |
    15). subtype-user | eventname-FortiGuard override failed | level-warning | count-1 |
```

The data is collected from FortiView for the last 24 hours by default. To specify a specific time range, customize the time filter using the `diagnose fortiview time` command.

To filter the time range of system events in the CLI:

```
# diagnose fortiview time <arg1> <arg2>
```

Where `<arg1>` is the start time in YYYY-MM-DD HH:MM:SS and `<arg2>` is the end time in YYYY-MM-DD HH:MM:SS.

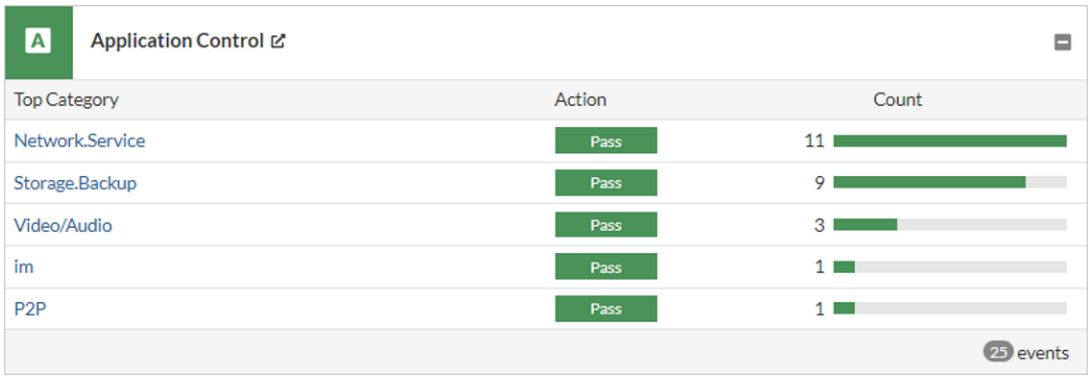
Security Events log page

The *Log & Report* > *Security Events* log page includes:

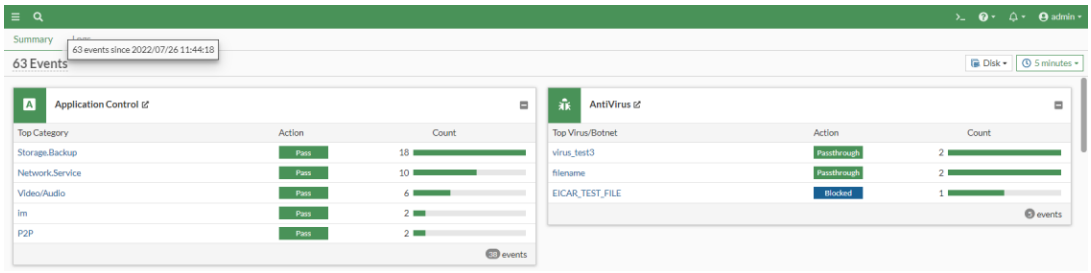
- A *Summary* tab that displays the five most frequent events for all of the enabled UTM security events.
- A *Logs* tab that displays individual, detailed logs for each UTM type.

The *Summary* tab includes the following:

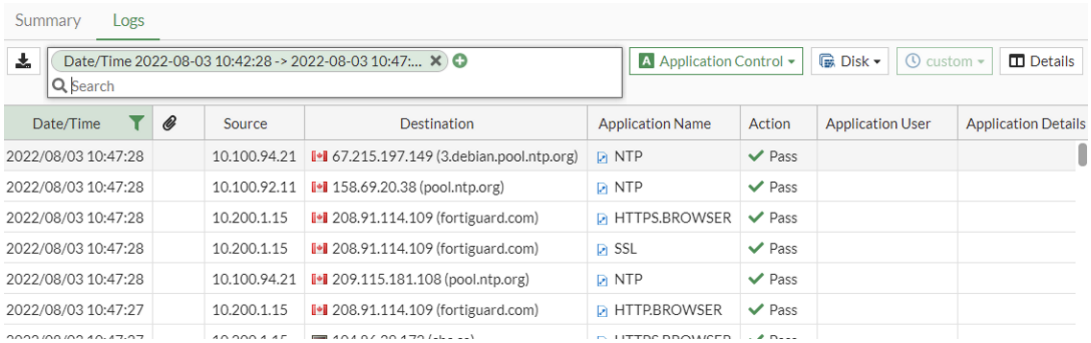
- Event list footers show a count of the events that relate to the type.



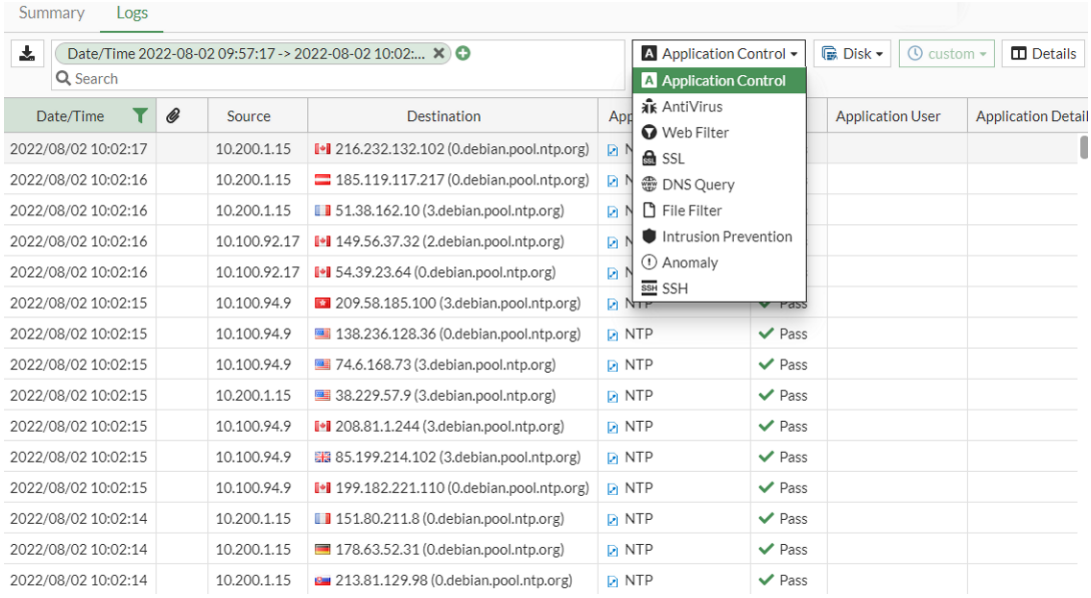
- A count of the total events is shown at the top of the *Summary*. Hovering over the count shows the number of events with a time stamp.



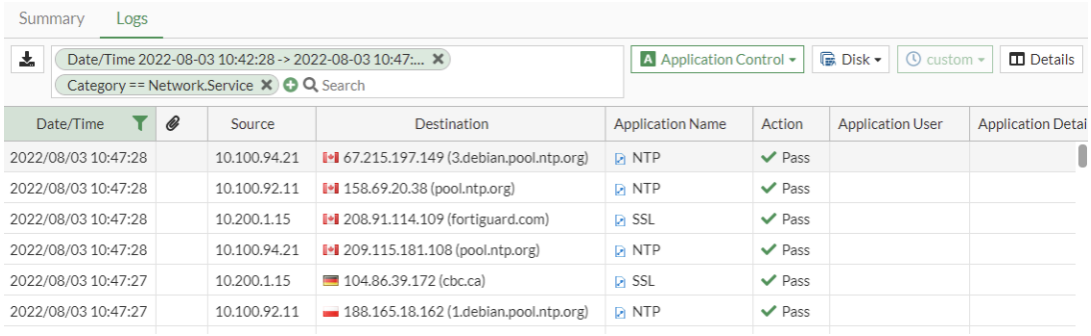
- Clicking on any event type title opens the *Logs* page for that event type filtered by the selected time span.
  For example, clicking *Application Control* opens the following page:



The security event type can be changed in the top-right dropdown list.

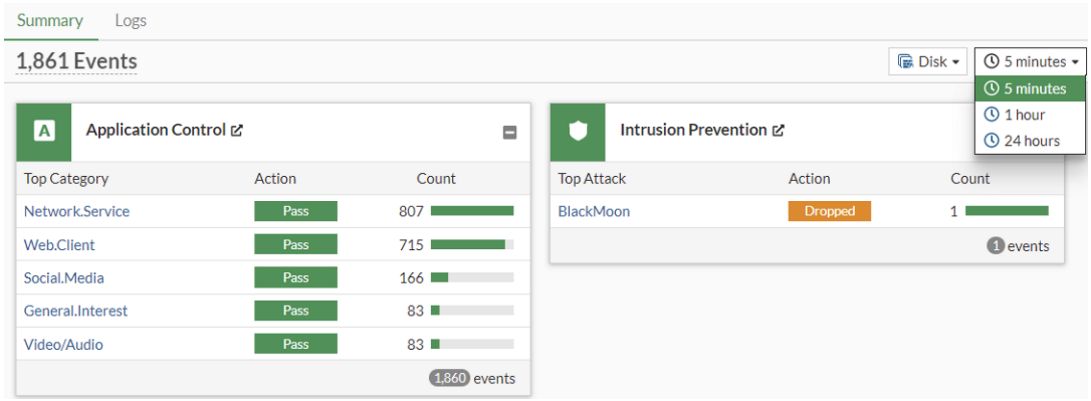


- Clicking on any event entry opens the *Logs* page for that event type filtered by the selected time span and log description.
  For example, in the *Application Control* box, clicking *Network.Service* opens the following page:



Disk logging and historical FortiView must be enabled for the *Summary* tab to display valid data. See Log settings and targets on page 4768 for more information.

A time frame can be selected from the dropdown.



The non-empty security event cards will list up to five top entries within the time range set.

Data is retrieved from FortiView with the *5 minutes* range updated first. When selecting either the *1 hour* or *24 hours* time range, there may be a delay to update top security event entries. Logs sourced from the Disk have different time frames available for filtering. See Viewing event logs on page 4753.

Up to 100 top security event entries can be listed in the CLI using the `diagnose fortiview result security-log` command.

To list security events in the CLI:

```
# diagnose fortiview result security-log [<filters>]
```

To list security events in the CLI with no filters applied:

```
# diagnose fortiview result security-log

    data(1646862300-1646948701):
    0). logcat-2 | logcatname-virus | logid-0211008192 | eventname-EICAR_TEST_FILE | eventname_
field-virus | action-blocked | count-1 |
```

```
    1). logcat-2 | logcatname-virus | logid-0211008192 | eventname-virus_test3 | eventname_field-
virus | action-passthrough | count-1 |
    2). logcat-2 | logcatname-virus | logid-0212008448 | eventname-filename | eventname_field-
virus | action-passthrough | count-1 |
    3). logcat-3 | logcatname-webfilter | logid-0318012800 | eventname- | eventname_field-catdesc
| action-blocked | count-2 |
    4). logcat-3 | logcatname-webfilter | logid-0316013056 | eventname-Information Technology |
eventname_field-catdesc | action-blocked | count-1 |
    5). logcat-3 | logcatname-webfilter | logid-0316013056 | eventname-Malicious Websites |
eventname_field-catdesc | action-blocked | count-1 |
    6). logcat-4 | logcatname-ips | logid-0419016384 | eventname-Eicar.Virus.Test.File |
eventname_field-attack | action-dropped | count-3 |
    7). logcat-4 | logcatname-ips | logid-0422016400 | eventname-test_botnet | eventname_field-
attack | action-detected | count-1 |
    8). logcat-7 | logcatname-anomaly | logid-0720018432 | eventname-tcp_syn_flood | eventname_
field-attack | action-clear_session | count-1 |
    9). logcat-10 | logcatname-app-ctrl | logid-1059028704 | eventname-Storage.Backup | eventname_
field-appcat | action-pass | count-9 |
    10). logcat-10 | logcatname-app-ctrl | logid-1059028704 | eventname-Video/Audio | eventname_
field-appcat | action-pass | count-3 |
    11). logcat-10 | logcatname-app-ctrl | logid-1059028672 | eventname-im | eventname_field-
appcat | action-pass | count-1 |
    12). logcat-10 | logcatname-app-ctrl | logid-1059028704 | eventname-P2P | eventname_field-
appcat | action-pass | count-1 |
    13). logcat-15 | logcatname-dns | logid-1501054400 | eventname-Domain blocked because it is in
the domain-filter list | eventname_field-logid | action-block | count-1 |
    14). logcat-17 | logcatname-ssl | logid-1700062300 | eventname-SSL connection is blocked due
to the server certificate is blocklisted | eventname_field-logid | action-blocked | count-1 |
    15). logcat-16 | logcatname-ssh | logid-1600061002 | eventname-SSH shell command is detected |
eventname_field-logid | action-passthrough | count-1 |
    16). logcat-16 | logcatname-ssh | logid-1601061010 | eventname-SSH channel is blocked |
eventname_field-logid | action-blocked | count-1 |
    17). logcat-12 | logcatname-waf | logid-1200030248 | eventname-Web application firewall
blocked application by signature | eventname_field-logid | action-blocked | count-1 |
    18). logcat-8 | logcatname-voip | logid-0814044032 | eventname-Logid_44032 | eventname_field-
logid | action-permit | count-1 |
    19). logcat-5 | logcatname-emailfilter | logid-0513020480 | eventname-SPAM notification |
eventname_field-logid | action-blocked | count-1 |
```

To list blocked security events in the CLI:

```
# diagnose fortiview result security-log action=blocked

    data(1646862600-1646949001):
    0). logcat-2 | logcatname-virus | logid-0211008192 | eventname-EICAR_TEST_FILE | eventname_
field-virus | action-blocked | count-1 |
    1). logcat-3 | logcatname-webfilter | logid-0318012800 | eventname- | eventname_field-catdesc
| action-blocked | count-2 |
    2). logcat-3 | logcatname-webfilter | logid-0316013056 | eventname-Information Technology |
eventname_field-catdesc | action-blocked | count-1 |
    3). logcat-3 | logcatname-webfilter | logid-0316013056 | eventname-Malicious Websites |
eventname_field-catdesc | action-blocked | count-1 |
```

```
    4). logcat-17 | logcatname-ssl | logid-1700062300 | eventname-SSL connection is blocked due to
the server certificate is blocklisted | eventname_field-logid | action-blocked | count-1 |
    5). logcat-16 | logcatname-ssh | logid-1601061010 | eventname-SSH channel is blocked |
eventname_field-logid | action-blocked | count-1 |
    6). logcat-12 | logcatname-waf | logid-1200030248 | eventname-Web application firewall blocked
application by signature | eventname_field-logid | action-blocked | count-1 |
    7). logcat-5 | logcatname-emailfilter | logid-0513020480 | eventname-SPAM notification |
eventname_field-logid | action-blocked | count-1 |
```

Reports page

The *Log & Report* > *Reports* page consolidates FortiAnalyzer, FortiGate Cloud, and Local log reports. Administrators can generate, delete, and edit report schedules, and view and download generated reports. The *Reports* page is organized into dedicated tabs:

- FortiAnalyzer on page 4764
- FortiGate Cloud on page 4766
- Local on page 4767

FortiAnalyzer

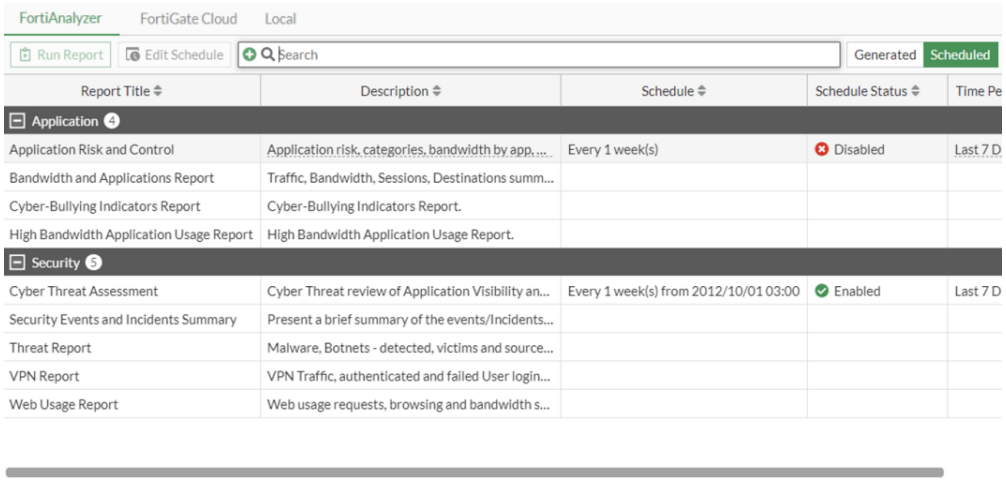
FortiAnalyzer reports can be viewed in the *FortiAnalyzer* tab.

FortiAnalyzer must be configured in FortiOS. If the FortiGate is unauthorized on FortiAnalyzer, or the connection to FortiAnalyzer is down, the *FortiAnalyzer* tab loads with *No results*.

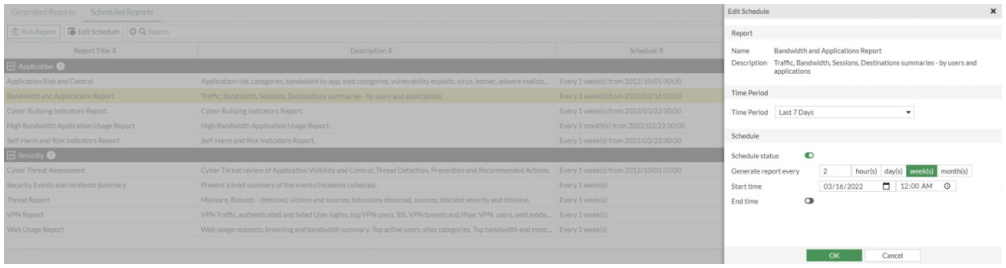
When the Security Fabric is enabled, only the root FortiGate can run, edit, and delete FortiAnalyzer reports. Downstream FortiGates can only view the generated reports.

To edit a report schedule:

1. Go to *Log & Report* > *Reports* and select the *FortiAnalyzer* tab.
2. Select *Scheduled*.

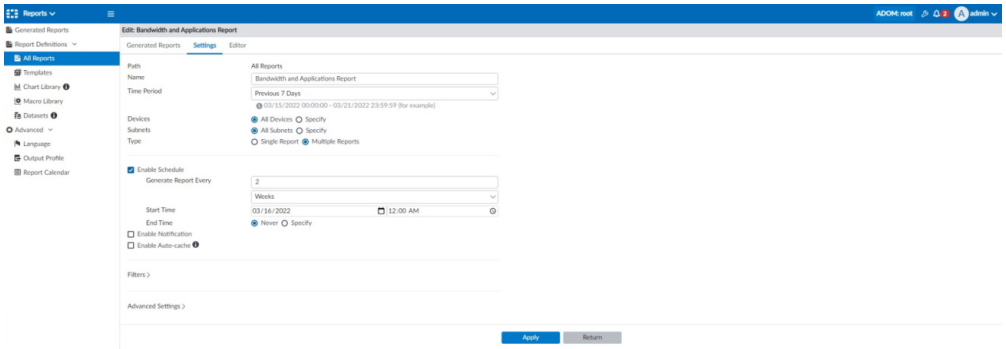


3. Select a report and click *Edit Schedule*. The *Edit Schedule* pane opens. In this example, the schedule for the Bandwidth and Applications report is changed to run from every week to every two weeks.
4. In the *Schedule* section, set the values for *Generate report every* to *2 week(s)*.



5. Click *OK*.

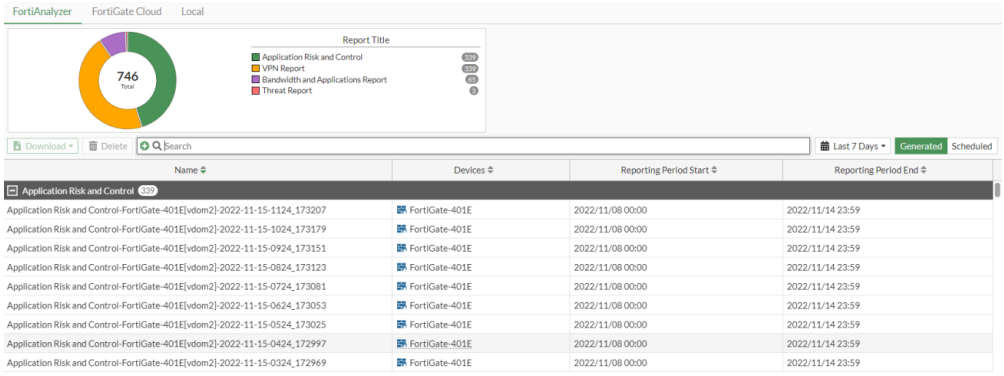
   The schedule is also updated automatically in FortiAnalyzer for the same report (go to *Reports* > *Report Definitions* > *All Reports* and edit the report to view the settings).



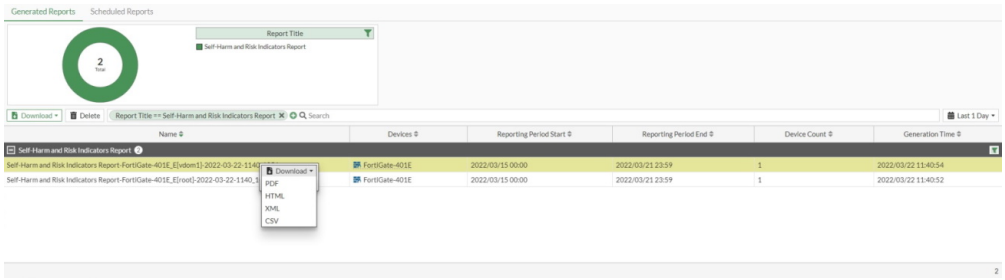
To view and download reports:

1. Go to *Log & Report* > *Reports* and select the *FortiAnalyzer* tab.

   A pie chart displays the total count of FortiAnalyzer reports, categorized by report title. Generated reports are listed below and arranged by title, which includes reports from all VDOMs.



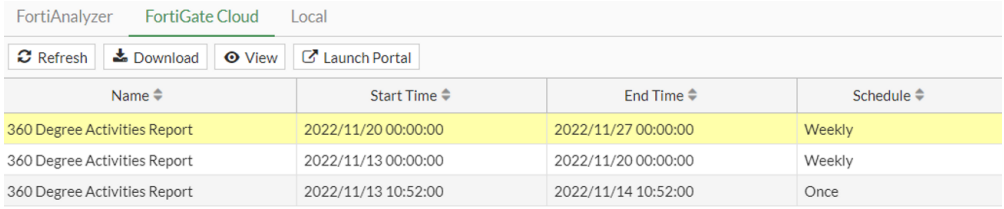
2. In the pie chart, click the a segment to filter reports.
3. In the filtered results, select the report. Right-click and select *Download*.



4. Select a file format. The report is saved to the default download location.

FortiGate Cloud

Reports can be viewed and downloaded from the *FortiGate Cloud* tab. Select *Refresh* to regenerate the available reports.



To download a report:

1. Go to *Log & Report* > *Reports* and select the *FortiGate Cloud* tab.
2. Select the report.
3. Click *Download*. The report is saved to the default download location.

To view a report:

1. Go to *Log & Report* > *Reports* and select the *FortiGate Cloud* tab.
2. Select the report.
3. Click *View*. The report is displayed.

You can launch the FortiGate Cloud portal from the *FortiGate Cloud* tab.

To launch the FortiGate Cloud portal:

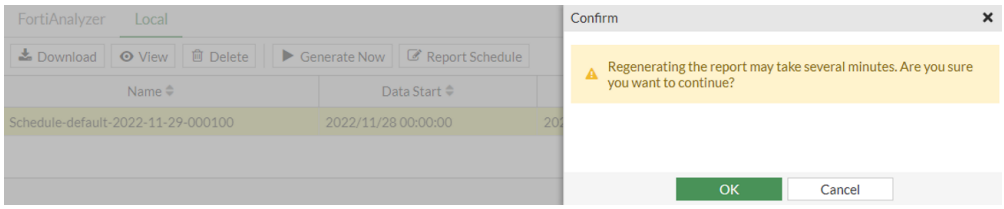
1. Go to *Log & Report* > *Reports* and select the *FortiGate Cloud* tab.
2. Select *Launch Portal*. The FortiGate Cloud landing page opens.

Local

Reports can be generated, scheduled, viewed, and downloaded in the *Local* tab.

To generate a report:

1. Go to *Log & Report* > *Reports* and select the *Local* tab.
2. Select *Generate Now*. The *Confirm* pane opens.



3. Click *OK*.

To view a report:

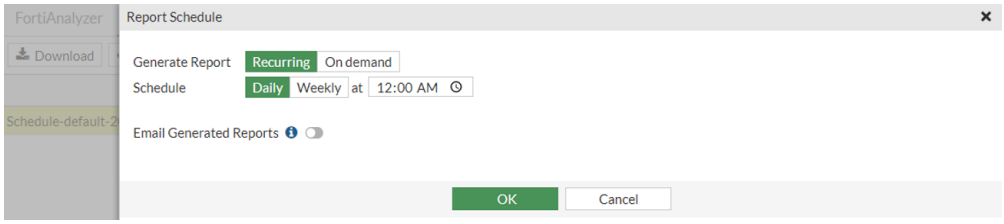
1. Go to *Log & Report* > *Reports* and select the *Local* tab.
2. Select the report.
3. Click *View*. The report is displayed.

To download a report:

1. Go to *Log & Report* > *Reports* and select the *Local* tab.
2. Select the report.
3. Click *Download*. The report is saved to the default download location.

To schedule reports:

1. Go to *Log & Report* > *Reports* and select the *Local* tab.
2. Click *Report Schedule*. The *Report Schedule* pane opens.



3. Set the schedule details.

4. Click *OK*.

Log settings and targets

Log settings determine what information is recorded in logs, where the logs are stored, and how often storage occurs. Log settings can be configured in the GUI and CLI. In the GUI, *Log & Report > Log Settings* provides the settings for local and remote logging.

Log & Report > Log Settings is organized into tabs:

- Global Settings on page 4768
- Local Logs on page 4769
- Threat Weight on page 4770

Global Settings

Settings available in the *Global Settings* tab include:

UUIDs in Traffic Log		
	Policy	Define the use of policy UUIDs in traffic logs: • *Enable*: Policy UUIDs are stored in traffic logs. UUIDs can be matched for each source and destination that match a policy in the traffic log. See Source and destination UUID logging on page 4790 for more information. • *Disable*: Policy UUIDs are excluded from the traffic logs.
	Address	Define the use of address UUIDs in traffic logs: • *Enable*: Address UUIDs are stored in traffic logs. When viewing *Forward Traffic* logs, a filter is automatically set based on UUID. • *Disable*: Address UUIDs are excluded from traffic logs.
Log Settings		
	Event Logging	Define the allowed set of event logs to be recorded: • *All*: All event logs will be recorded. • *Customize*: Select specific event log types to be recorded. Deselect all options to disable event logging.
	Local Traffic Logging	Define the allowed set of traffic logs to be recorded: • *All*: All traffic logs to and from the FortiGate will be recorded. • *Specify*: Select specific traffic logs to be recorded. Deselect all options to disable traffic logging. Local traffic logging is disabled by default due to the high volume of logs generated. Enable *Log local-in traffic* to log local traffic for local-in policies globally or per policy. See Local-in policy on page 1886.

Syslog logging	Enable/disable sending logs to a syslog server: • *Disable*: No logs are sent to a syslog server. • *Enable*: Select to specify the FQDN or IP address of the syslog server and send logs to the syslog server.
Preferences	

Resolve Hostnames	Define the translation of IP addresses to host names:
	• *Enable*: Two options are available:
	• *On log creation*: use hostnames that are resolved on log creation. This option enables the following CLI setting:
	`config log gui-display`
	`set resolve-ip enable`
	`end`
	• *When viewed*: IP addresses are translated to host names using reverse DNS lookup while the entries are fetched and viewed. If the DNS server is not available or is slow to reply, requests may time out. This option enables the following CLI setting:
	`config log gui-display`
	`set resolve-hosts enable`
	`end`
	• If both options are enabled in the CLI, then `resolve-ip` is preferred.
	• *Disable*: IP addresses are not translated to host names. Both CLI options are disabled.

Resolve Unknown Applications	Define the resolution of unknown applications: • *Enable*: Unknown applications are resolved using the Internet Service Database. • *Disable*: Unknown applications are not resolved.
Download full geographic IP database	Enable/disable downloading the full geographic database to support FortiView Countries: • *Enable*: Download the full geographic database. • *Disable*: Do not download the full geographic IP database. Conserves memory.

Local Logs

Settings available in the *Local Logs* tab include:

Local Logs	
Disk logging	Define local log storage on the FortiGate: • *Enable*: Logs will be stored on a local disk. Local disk logging is not available in the GUI if the Security Fabric is enabled. When the Security

		Fabric is enabled, disk logging can still be configured on the root FortiGate in the CLI but is not available for downstream FortiGates. • *Disable*: Logs will be stored remotely to FortiAnalyzer/FortiManager or to a Cloud logging device.
	Local Reports	Define log reporting on the FortiGate: • *Enable*: Local reports will be available on the FortiGate. Reports can be reviewed in *Log & Report > Reports* in the *Local* tab. • *Disable*: Local reports will not be available on the FortiGate.
	Historical FortiView	Define the presentation of log information on FortiView: • *Enable*: Historical log data will be available on a FortiView monitor. By default, logs older than seven days are deleted. Disk logging must be enabled. • *Disable*: Historical log data will not be available on FortiView.
	Disk Usage	Presents the disk space used and the total disk space available on the disk.

Threat Weight

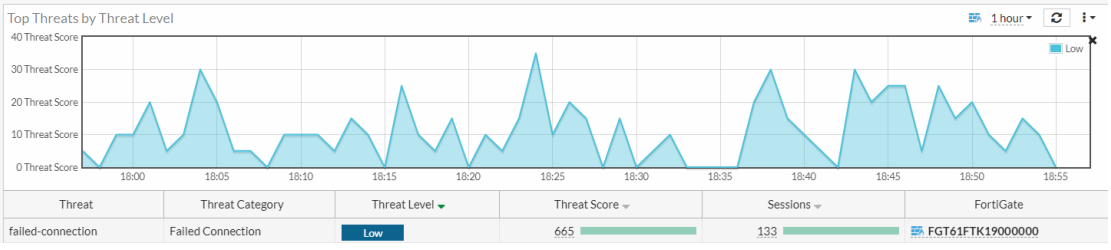
Threat weight helps aggregate and score threats based on user-defined severity levels. It adds several fields such as threat level (`crlevel`), threat score (`crscore`), and threat type (`craction`) to traffic logs. Threat weight logging is enabled by default and the settings can be customized. Threats can be viewed from the *Top Threats* FortiView dashboard.

To configure threat weight settings:

1. Go to *Log & Report > Log Settings* and select the *Threat Weight* tab.
2. Adjust the settings as needed, such as individual weights per threat type and risk level values.
3. Click *Apply*.

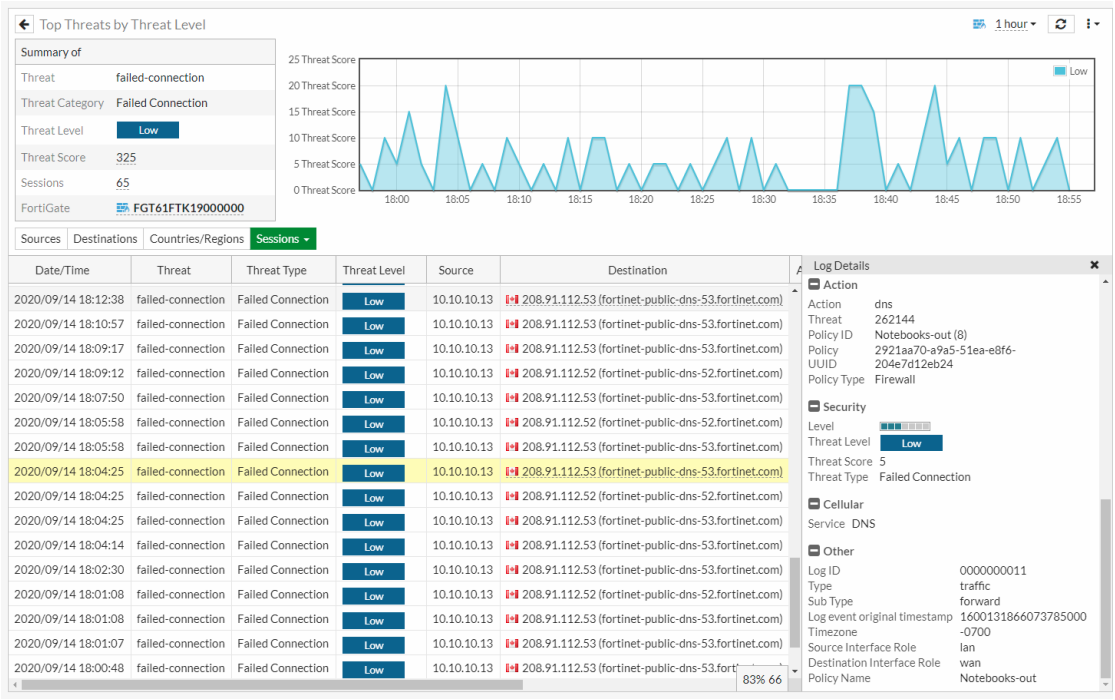
To add the Top Threats monitor to the dashboard:

1. In the tree menu, click *Dashboard* and in the FortiView section, click the *+* sign (*Add Monitor*).
2. In the *Security* section, enable *Show More* and click *Top Threats*.
3. Configure the settings as needed.
4. Click *Add Monitor*.
5. Go to *Dashboard > Top Threats*. The *Top Threats* monitor displays threats based on the scores in the traffic logs.



6. Double-click a threat to view the summary.

7. Click *Sources*, *Destinations*, *Countries/Regions*, or *Sessions* to view more information. Double-click an entry to view the log details.



Configuring logs in the CLI

The FortiGate can store logs locally to its system memory or a local disk. Logs can also be stored externally on a storage device, such as FortiAnalyzer, FortiAnalyzer Cloud, FortiGate Cloud, or a syslog server.

Disk logging

Disk logging must be enabled for logs to be stored locally on the FortiGate. By default, logs older than seven days are deleted from the disk. Log age can be configured in the CLI. Approximately 75% of disk space is available for log storage. Log storage space can be determined using the `diagnose sys logdisk usage` command.

To configure local disk logging:

```
config log disk setting
    set status enable
    set maximum-log-age <integer>
    set max-log-file-size <integer>
end
```

Remote logging

The process to configure FortiGate to send logs to FortiAnalyzer or FortiManager is identical. Remote logging to FortiAnalyzer and FortiManager can be configured using both the GUI and CLI. When using the CLI, use the `config log fortianalyzer setting` command for both FortiAnalyzer and FortiManager.

If VDOMs are configured on the FortiGate, multiple FortiAnalyzers and syslog servers can be added globally. See Configuring multiple FortiAnalyzers (or syslog servers) per VDOM on page 4777 and Configuring multiple FortiAnalyzers on a FortiGate in multi-VDOM mode on page 4779 for more information.

To configure remote logging to FortiAnalyzer:

```
config log fortianalyzer setting
    set status enable
    set server <server_IP>
    set upload option {store-and-upload | realtime | 1-minute | 5-minute}
end
```

Remote logging can also be configured to FortiCloud, FortiSIEM, and syslog servers. Up to four syslog servers or FortiSIEM devices can be configured using the `config log syslogd` command and can send logs to syslog in CSV and CEF formats.

To configure remote logging to FortiCloud:

```
config log fortiguard setting
    set status enable
    set source-ip <source IP used to connect FortiCloud>
end
```

To configure remote logging to a syslog server:

```
config log syslogd setting
    set status enable
    set server <syslog_IP>
    set format {default | csv | cef | rfc5424 | json |custom}
    set custom-log-format <string>
end
```

To configure a custom log format for logging to a syslog server:

```
config log custom-format
    edit <name>
        set field-exclusion-list {date | devid | devname | eventtime | level | logid | slot |
subtype | time | type | tz | vd | vsn}
        empty-value-indicator <string>
        config log-templates
            edit <name>
                set category <string>
                set subtypes <string>
```

```
                set template <string>
            next
        end
    next
end
```

See Customizing log formats for syslog servers NEW on page 4812 for examples.

Log filters

Log filter settings can be configured to determine which logs are recorded to the FortiAnalyzer, FortiManager, and syslog servers. This allows certain logging levels and types of logs to be directed to specific log devices.

To configure log filters for FortiAnalyzer:

```
config log fortianalyzer filter
    set severity <level>
    set forward-traffic {enable | disable}
    set local-traffic {enable | disable}
    set multicast-traffic {enable | disable}
    set sniffer-traffic {enable | disable}
end
```

To configure log filters for a syslog server:

```
config log syslogd filter
    set severity <level>
    set forward-traffic {enable | disable}
    set local-traffic {enable | disable}
    set multicast-traffic {enable | disable}
    set sniffer-traffic {enable | disable}
end
```

Email alerts

FortiGate events can be monitored at all times using email alerts. Email alerts send notifications to up to three recipients and can be triggered based on log event and severity level. Email alerts will be sent every five minutes by default but this can be configured in the CLI.

To configure email alerts:

```
config alertemail setting
    set username <name>
    set mailto1 <email>
    set filter-mode {category | threshold}
    set email-interval <integer>
    set IPS-logs {enable | disable}
```

```
    set HA-logs {enable | disable}
    set antivirus-logs {enable | disable}
    set webfilter-logs {enable | disable}
    set log-disk-usage-warning {enable | disable}
end
```

FortiTelemetry events

When enabled, the following events for FortiTelemetry are captured in the FortiOS system event log:

- FortiTelemetry agent discovered, online, or offline
- FortiTelemetry agent authorized, unauthorized, rejected, or deleted
- FortiGate connected to or disconnected from FortiTelemetry cloud server
- Telemetry tasks pushed to FortiTelemetry agents

To enable telemetry event logging:

```
config log eventfilter
    set telemetry enable
end
```

Logging to FortiAnalyzer

The following topics provide instructions on logging to FortiAnalyzer:

- FortiAnalyzer log caching on page 4774
- Configuring multiple FortiAnalyzers (or syslog servers) per VDOM on page 4777
- Configuring multiple FortiAnalyzers on a FortiGate in multi-VDOM mode on page 4779
- Switching to an alternate FortiAnalyzer if the main FortiAnalyzer is unavailable on page 4782

FortiAnalyzer log caching

Reliable logging to FortiAnalyzer prevents lost logs when the connection between FortiOS and FortiAnalyzer is disrupted. When reliable mode is enabled:

1. Logs are cached in a FortiOS memory queue.
2. FortiOS sends logs to FortiAnalyzer, and FortiAnalyzer uses `seq_no` to track received logs.
3. After FortiOS sends logs to FortiAnalyzer, logs are moved to a confirm queue in FortiOS.
4. FortiOS periodically queries FortiAnalyzer for the latest `seq_no` of the last log received, and clears logs from the confirm queue up to the `seq_no`.
5. If the connection between FortiOS and FortiAnalyzer is disrupted, FortiOS resends the logs in the confirm queue to FortiAnalyzer when the connection is reestablished.

FortiAnalyzer 7.2.0 and later is required.

To enable reliable mode:

```
config log fortianalyzer setting
    set reliable enable
end
```

To view the memory and confirm queues:

1. Verify that log synchronization is enabled for FortiAnalyzer:

```
# diagnose test application fgtlogd 1
vdom-admin=0
mgmt=root

fortilog:
faz: global , enabled
    server=172.16.200.251, realtime=1, ssl=1, state=connected
    server_log_status=Log is allowed.,
    src=, mgmt_name=FGh_Log_root_172.16.200.251, reliable=1, sni_prefix_type=none,
    required_entitlement=none, region=ca-west-1,,
    logsync_enabled:1, logsync_conn_id:65535, seq_no:790
...
```

2. When a network disruption disconnects FortiOS from FortiAnalyzer and FortiOS continues to generate logs, the logs are cached in the memory queue.
   - View the number of logs in the cache and queue:

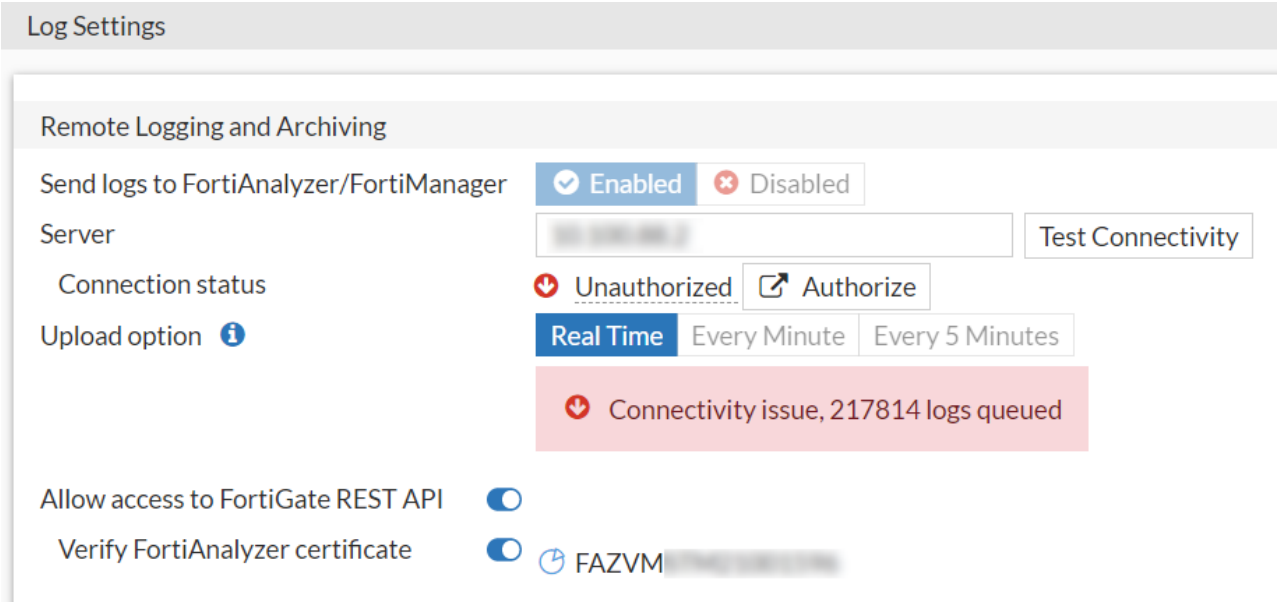
```
# diagnose test application fgtlogd 41

cache maximum: 189516595(180MB) objects: 40 used: 27051(0MB) allocated: 29568(0MB)

VDOM:root
Memory queue for: global-faz
    queue:
        num:9 size:6976(0MB) total size:26068(0MB) max:189516595(180MB) logs:28
Confirm queue for: global-faz
    queue:
        num:29 size:19092(0MB) total size:27051(0MB) max:189516595(180MB) logs:7
```

```
# diagnose test application fgtlogd 30
VDOM:root
Memory queue for: global-faz
        queue:
                num:9 size:6976(0MB) total size:26068(0MB) max:189516595(180MB)
                        type:3, cat=1, log_count=1, seq_no=0, data len=359 size:435
                        type:3, cat=1, log_count=1, seq_no=0, data len=307 size:383
                        ......
                        type:3, cat=0, log_count=4, seq_no=0, data len=1347 size:1423
```

```
                        type:3, cat=4, log_count=1, seq_no=0, data len=653 size:729
                'total log count':28,  'total data len':6292

Confirm queue for: global-faz
        queue:
                num:29 size:19092(0MB) total size:26068(0MB) max:189516595(180MB)
                        type:3, cat=1, log_count=1, seq_no=1, data len=290 size:366
                        type:3, cat=1, log_count=1, seq_no=2, data len=233 size:309
                        ......
                        type:3, cat=0, log_count=1, seq_no=28, data len=524 size:600
                        type:3, cat=1, log_count=1, seq_no=29, data len=307 size:383
                'total log count':76,  'total data len':16888
```

There are nine OFTP items cached to the memory queue, and 29 OFTP items to send from FortiOS to FortiAnalyzer that are waiting for confirmation from FortiAnalyzer.

- Go to *Log & Report* > *Log Settings* to view the queue in the GUI:



3. Re-establish the connection between FortiOS and FortiAnalyzer and confirm that the queue has cleared by checking the seq_no, which indicates the latest confirmation log from FortiAnalyzer:

```
# diagnose test application fgtlogd 30
VDOM:root
Memory queue for: global-faz
    queue:
        num:0 size:0(0MB) total size:0(0MB) max:189516595(180MB)
        'total log count':0,  'total data len':0

Confirm queue for: global-faz
    queue:
        num:0 size:0(0MB) total size:0(0MB) max:189516595(180MB)
        'total log count':0,  'total data len':0
```

The queue has been cleared, meaning that FortiOS received confirmation from FortiAnalyzer and cleared the confirm queue.

```
# diagnose test application fgtlogd 1
vdom-admin=0
mgmt=root

fortilog:
faz: global , enabled
        server=172.16.200.251, realtime=1, ssl=1, state=connected
        server_log_status=Log is allowed.,
        src=, mgmt_name=FGh_Log_root_172.16.200.251, reliable=1, sni_prefix_type=none,
        required_entitlement=none, region=ca-west-1,
        logsync_enabled:1, logsync_conn_id:65535, seq_no:67
            status: ver=6, used_disk=0, total_disk=0, global=0, vfid=0 conn_verified=Y
            SNs: last sn update:38 seconds ago.
                Sn list:
               (FAZ-VMTM21000000,age=38s)
            queue: qlen=0.
```

OFTP items with a `seq_no` lower than 67 have been sent to FortiAnalyzer and were confirmed.

Configuring multiple FortiAnalyzers (or syslog servers) per VDOM

In a VDOM, multiple FortiAnalyzer and syslog servers can be configured as follows:

- Up to three override FortiAnalyzer servers
- Up to four override syslog servers

If the VDOM `faz-override` and/or `syslog-override` setting is enabled or disabled (default) before upgrading, the setting remains the same after upgrading.

If the override setting is disabled, the GUI displays the global FortiAnalyzer1 or syslog1 setting. If the override setting is enabled, the GUI displays the VDOM override FortiAnalyzer1 or syslog1 setting.

You can only use CLI to enable the override to support multiple log servers.

To enable FortiAnalyzer and syslog server override under VDOM:

```
config log setting
    set faz-override enable
    set syslog-override enable
end
```

When `faz-override` and/or `syslog-override` is enabled, the following CLI commands are available for configuring VDOM override:

To configure VDOM override for FortiAnalyzer:

1. Configure the FortiAnalyzer override settings:

```
config log fortianalyzer/fortianalyzer2/fortianalyzer3 override-setting
    set status enable
    set server "123.12.123.123"
    set reliable enable
end
```

2. Configure the override filters:

```
config log fortianalyzer/fortianalyzer2/fortianalyzer3 override-filter
    set severity information
    set forward-traffic enable
    set local-traffic enable
    set multicast-traffic enable
    set sniffer-traffic enable
    set anomaly enable
    set voip enable
    set dlp-archive enable
    set dns enable
    set ssh enable
    set ssl enable
end
```

To configure VDOM override for a syslog server:

1. Configure the syslog override settings:

```
config log syslogd/syslogd2/syslogd3/syslogd4 override-setting
    set status enable
    set server "123.12.123.12"
    set facility local1
end
```

2. Configure the override filters:

```
config log syslogd/syslogd2/syslogd3/syslogd4 override-filter
    set severity information
    set forward-traffic enable
    set local-traffic enable
    set multicast-traffic enable
    set sniffer-traffic enable
    set anomaly enable
    set voip enable
    set dns enable
    set ssh enable
    set ssl enable
end
```

Configuring multiple FortiAnalyzers on a FortiGate in multi-VDOM mode

This topic shows a sample configuration of multiple FortiAnalyzers on a FortiGate in multi-VDOM mode.

In this example:

- The FortiGate has three VDOMs:
  - Root (management VDOM)
  - VDOM1
  - VDOM2
- There are four FortiAnalyzers.
  These IP addresses are used as examples in the instructions below.
  - FAZ1: `172.16.200.55`
  - FAZ2: `172.18.60.25`
  - FAZ3: `192.168.1.253`
  - FAZ4: `192.168.1.254`
- Set up FAZ1 and FAZ2 under global.
  - These two collect logs from the root VDOM and VDOM2.
  - FAZ1 and FAZ2 must be accessible from management VDOM root.
- Set up FAZ3 and FAZ4 under VDOM1.
  - These two collect logs from VDOM1.
  - FAZ3 and FAZ4 must be accessible from VDOM1.

To set up FAZ1 as global FortiAnalyzer 1 from the GUI:

Prerequisite: FAZ1 must be reachable from the management root VDOM.

1. Go to *Global* > *Log & Report* > *Log Settings*.
2. Enable *Send logs to FortiAnalyzer/FortiManager*.
3. Enter the FortiAnalyzer IP.
   In this example: `172.16.200.55`.
4. For *Upload option*, select *Real Time*.
5. Click *Apply*.

To set up FAZ2 as global FortiAnalyzer 2 from the CLI:

Prerequisite: FAZ2 must be reachable from the management root VDOM.

```
config log fortianalyzer2 setting
    set status enable
    set server "172.18.60.25"
    set upload-option realtime
end
```

To set up FAZ3 and FAZ4 as VDOM1 FortiAnalyzer 1 and FortiAnalyzer 2:

Prerequisite: FAZ3 and FAZ4 must be reachable from VDOM1.

```
config log setting
   set faz-override enable
end

config log fortianalyzer override-setting
   set status enable
   set server "192.168.1.253"
   set upload-option realtime
end

config log fortianalyzer2 override-setting
   set status enable
   set server "192.168.1.254"
   set upload-option realtime
end
```

Checking FortiAnalyzer connectivity

To use the diagnose command to check FortiAnalyzer connectivity:

1. Check the global FortiAnalyzer status:

```
FGTA(global) # diagnose test application fgtlogd 1
vdom-admin=1
mgmt=root
faz: global , enabled
        server=172.16.200.55, realtime=1, ssl=1, state=connected
        server_log_status=Log is allowed.,
        src=, mgmt_name=FGh_Log_root_172.16.200.55, reliable=0, sni_prefix_type=none,
        required_entitlement=none, region=ca-west-1,
        logsync_enabled:1, logsync_conn_id:65535, seq_no:0
                status: ver=6, used_disk=0, total_disk=0, global=0, vfid=0 conn_verified=Y
                SNs: last sn update:383 seconds ago.
                        Sn list:
                        (FAZ-VMTM2200****,age=383s)
                queue: qlen=0.
filter: severity=6, sz_exclude_list=0
         traffic virus webfilter ips emailfilter anomaly voip dlp app-ctrl waf dns ssh ssl
file-filter icap sctp-filter
subcategory:
        traffic: forward local multicast sniffer ztna
        anomaly:all subcategories are enabled.
        server: global, id=0, ready=1, name=172.16.200.55 addr=172.16.200.55:514
        oftp-state=connected
faz2: global , enabled
        server=172.18.60.25, realtime=3, ssl=1, state=connected
        server_log_status=Log is allowed.,
```

```
        src=, mgmt_name=FGh_Log_root_172.18.60.25, reliable=0, sni_prefix_type=none,
        required_entitlement=none, region=ca-west-1,
        logsync_enabled:1, logsync_conn_id:131071, seq_no:0
                status: ver=6, used_disk=0, total_disk=0, global=0, vfid=0 conn_verified=Y
                SNs: last sn update:383 seconds ago.
                        Sn list:
                        (FAZ-VMTM2201****,age=383s)
                queue: qlen=0.
filter: severity=6, sz_exclude_list=0
         traffic virus webfilter ips emailfilter anomaly voip dlp app-ctrl waf dns ssh ssl
file-filter icap sctp-filter
subcategory:
        traffic: forward local multicast sniffer ztna
        anomaly:all subcategories are enabled.
        server: global, id=1, ready=1, name=172.18.60.25 addr=172.18.60.25:514
        oftp-state=connected
```

2. Check the VDOM1 override FortiAnalyzer status:

```
FGTA(global) # diagnose test application fgtlogd 3101
vdom VDOM1: id=3
event filter:
         event
system vpn user router wireless wanopt endpoint ha security-rating fortiextender connector
sdwan cifs-auth-fail switch-controller rest-api webproxy
faz: vdom, enabled, override
        server=192.168.1.253, realtime=3, ssl=1, state=connected
        server_log_status=Log is allowed.,
        src=, mgmt_name=FGh_Log_VDOM1_192.168.1.253, reliable=0, sni_prefix_type=none,
        required_entitlement=none, region=ca-west-1,
        logsync_enabled:1, logsync_conn_id:3, seq_no:0
                status: ver=6, used_disk=0, total_disk=0, global=0, vfid=0 conn_verified=Y
                SNs: last sn update:16 seconds ago.
                        Sn list:
                        (FAZ-VMTM2200****,age=16s)
                queue: qlen=0.
filter: severity=6, sz_exclude_list=0
         traffic virus webfilter ips emailfilter anomaly voip dlp app-ctrl waf dns ssh ssl
file-filter icap sctp-filter
subcategory:
        traffic: forward local multicast sniffer ztna
        anomaly:all subcategories are enabled.
        server: vdom, id=0, ready=1, name=192.168.1.253 addr=192.168.1.253:514
        oftp-state=connected
faz2: vdom, enabled, override
        server=192.168.1.254, realtime=3, ssl=1, state=connected
        server_log_status=Log is allowed.,
        src=, mgmt_name=FGh_Log_VDOM1_192.168.1.254, reliable=0, sni_prefix_type=none,
        required_entitlement=none, region=ca-west-1,
        logsync_enabled:1, logsync_conn_id:65539, seq_no:0
                status: ver=6, used_disk=0, total_disk=0, global=0, vfid=0 conn_verified=Y
                SNs: last sn update:16 seconds ago.
```

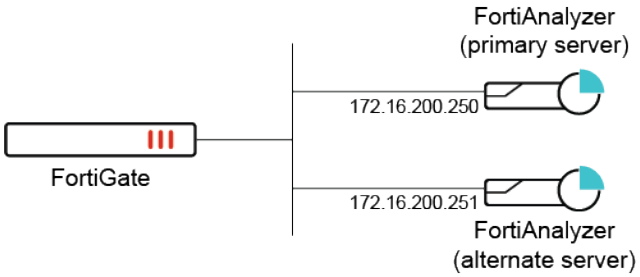
```
                        Sn list:
                        (FAZ-VMTM2201****,age=16s)
                queue: qlen=0.
filter: severity=6, sz_exclude_list=0
         traffic virus webfilter ips emailfilter anomaly voip dlp app-ctrl waf dns ssh ssl
file-filter icap sctp-filter
subcategory:
        traffic: forward local multicast sniffer ztna
        anomaly:all subcategories are enabled.

        server: vdom, id=1, ready=1, name=192.168.1.254 addr=192.168.1.254:514
        oftp-state=connected
faz3: vdom, disabled, override
```

Switching to an alternate FortiAnalyzer if the main FortiAnalyzer is unavailable

FortiOS supports switching to an alternate FortiAnalyzer if the main FortiAnalyzer is unavailable. Once the connectivity is restored, it will automatically fall back to the primary FortiAnalyzer.

The FortiAnalyzer units are required to operate in geo-redundant HA mode. See Geo-redundant HA for more information.

This feature can be used in multi-VDOM mode when FortiAnalyzer override settings are configured.



To configure switching to an alternate FortiAnalyzer when the main FortiAnalyzer is unavailable:

1. Ensure the FortiAnalyzer units are operating in geo-redundant HA mode. See Geo-redundant HA for more information.
2. Configure primary and alternate FortiAnalyzer servers:

```
config log fortianalyzer setting
    set status enable
    set server "172.16.200.250"
    set alt-server "172.16.200.251"
    set fallback-to-primary enable
    set serial "FAZ-VMTM22000000" "FAZ-VMTM23000003"
end
```

3. Verify the primary and alternate FortiAnalyzer server IPs:

```
# diagnose test application fgtlogd 1
vdom-admin=1
mgmt=vdom1

fortilog:
faz: global , enabled
        server=172.16.200.250, alt-server=172.16.200.251, active-server=172.16.200.250,
realtime=3, ssl=1, state=connected
        server_log_status=Log is allowed.,
        src=, mgmt_name=FGh_Log_vdom1_172.16.200.250, reliable=0, sni_prefix_type=none,
        required_entitlement=none, region=ca-west-1,
        logsync_enabled:1, logsync_conn_id:65535, seq_no:0
        disconnect_jiffies:0
                status: ver=6, used_disk=0, total_disk=0, global=0, vfid=0 conn_verified=Y
                SNs: last sn update:11 seconds ago.
                        Sn list:
                        (FAZ-VMTM22000000,age=11s)      (FAZ-VMTM23000003,age=12s)
                queue: qlen=0.
filter: severity=6, sz_exclude_list=0
         traffic virus webfilter ips emailfilter anomaly voip dlp app-ctrl waf dns ssh ssl
file-filter icap sctp-filter virtual-patch
subcategory:
        traffic: forward local multicast sniffer ztna
        virus:all subcategories are enabled.
        webfilter:all subcategories are enabled.
        ips:all subcategories are enabled.
        emailfilter:all subcategories are enabled.
        anomaly:all subcategories are enabled.
        voip:all subcategories are enabled.
        dlp:all subcategories are enabled.
        app-ctrl:all subcategories are enabled.
        waf:all subcategories are enabled.
        dns:all subcategories are enabled.
        ssh:all subcategories are enabled.
        ssl:all subcategories are enabled.
        file-filter:all subcategories are enabled.
        icap:all subcategories are enabled.
        sctp-filter:all subcategories are enabled.
        virtual-patch:all subcategories are enabled.

        server: global, id=0, ready=1, name=172.16.200.250 addr=172.16.200.250:514
        oftp-state=connected
        primary oftp status:null
        probe oftp status:null, 442
```

   The 172.16.200.250 server is currently active and acting as the primary FortiAnalyzer.
4. Make the primary FortiAnalyzer server go down. The FortiGate will automatically connect to the alternate FortiAnalyzer server.
5. Verify the FortiAnalyzer server status information:

```
# diagnose  test application fgtlogd 1
vdom-admin=1
mgmt=vdom1

fortilog:
faz: global , enabled
        server=172.16.200.250, alt-server=172.16.200.251, active-server=172.16.200.251,
realtime=3, ssl=1, state=connected
        server_log_status=Log is allowed.,
        src=, mgmt_name=FGh_Log_vdom1_172.16.200.250, reliable=0, sni_prefix_type=none,
        required_entitlement=none, region=ca-west-1,
        logsync_enabled:1, logsync_conn_id:65535, seq_no:0
        disconnect_jiffies:0
                status: ver=6, used_disk=0, total_disk=0, global=0, vfid=0 conn_verified=Y
                SNs: last sn update:30 seconds ago.
                        Sn list:
                        (FAZ-VMTM22000000,age=30s)      (FAZ-VMTM23000003,age=31s)
                queue: qlen=0.
filter: severity=6, sz_exclude_list=0
         traffic virus webfilter ips emailfilter anomaly voip dlp app-ctrl waf dns ssh ssl
file-filter icap sctp-filter virtual-patch
subcategory:
        traffic: forward local multicast sniffer ztna
        virus:all subcategories are enabled.
        webfilter:all subcategories are enabled.
        ips:all subcategories are enabled.
        emailfilter:all subcategories are enabled.
        anomaly:all subcategories are enabled.
        voip:all subcategories are enabled.
        dlp:all subcategories are enabled.
        app-ctrl:all subcategories are enabled.
        waf:all subcategories are enabled.
        dns:all subcategories are enabled.
        ssh:all subcategories are enabled.
        ssl:all subcategories are enabled.
        file-filter:all subcategories are enabled.
        icap:all subcategories are enabled.
        sctp-filter:all subcategories are enabled.
        virtual-patch:all subcategories are enabled.

        server: global, id=0, ready=1, name=172.16.200.250 addr=172.16.200.250:514
        oftp-state=connected
        probe oftp status:null, 38
```

The 172.16.200.251 server is currently active and acting as the primary FortiAnalyzer.

6. Restore the connection to the 172.16.200.250 server. The FortiGate will automatically reconnect to this FortiAnalyzer server.
7. Verify the FortiAnalyzer server status information:

```
# diagnose test application fgtlogd 1
vdom-admin=1
```

```
mgmt=vdom1

fortilog:
faz: global , enabled
        server=172.16.200.250, alt-server=172.16.200.251, active-server=172.16.200.250,
realtime=3, ssl=1, state=connected
        server_log_status=Log is allowed.,
        src=, mgmt_name=FGh_Log_vdom1_172.16.200.250, reliable=0, sni_prefix_type=none,
        required_entitlement=none, region=ca-west-1,
        logsync_enabled:1, logsync_conn_id:65535, seq_no:0
        disconnect_jiffies:0
                status: ver=6, used_disk=0, total_disk=0, global=0, vfid=0 conn_verified=Y
                SNs: last sn update:11 seconds ago.
                        Sn list:
                        (FAZ-VMTM22000000,age=58s)      (FAZ-VMTM23000003,age=59s)
                queue: qlen=0.
filter: severity=6, sz_exclude_list=0
         traffic virus webfilter ips emailfilter anomaly voip dlp app-ctrl waf dns ssh ssl
file-filter icap sctp-filter virtual-patch
subcategory:
        traffic: forward local multicast sniffer ztna
        virus:all subcategories are enabled.
        webfilter:all subcategories are enabled.
        ips:all subcategories are enabled.
        emailfilter:all subcategories are enabled.
        anomaly:all subcategories are enabled.
        voip:all subcategories are enabled.
        dlp:all subcategories are enabled.
        app-ctrl:all subcategories are enabled.
        waf:all subcategories are enabled.
        dns:all subcategories are enabled.
        ssh:all subcategories are enabled.
        ssl:all subcategories are enabled.
        file-filter:all subcategories are enabled.
        icap:all subcategories are enabled.
        sctp-filter:all subcategories are enabled.
        virtual-patch:all subcategories are enabled.

        server: global, id=0, ready=1, name=172.16.200.250 addr=172.16.200.250:514
        oftp-state=connected
        primary oftp status:null
        probe oftp status:null, 530
```

The 172.16.200.250 server is currently active and acting as the primary FortiAnalyzer again.

To manually switch from the primary to alternate FortiAnalyzer (and vice-versa):

```
# execute log {fortianalyzer | fortianalyzer2 | fortianalyzer3} manual-failover
```

If the primary server is still up, the behavior resulting from running this command is based on the `fallback-to-primary` setting configured in the global FortiAnalyzer log settings.

- If `fallback-to-primary` is enabled (default), running `execute log fortianalyzer manual-failover` will switch to the alternate FortiAnalyzer, but it will switch back to the primary since it is not actually down.
- If `fallback-to-primary` is disabled, running `execute log fortianalyzer manual-failover` will switch to the alternate FortiAnalyzer, and it will not switch back to the primary.

Advanced and specialized logging

The following topics provide information on advanced and specialized logging:

- Logs for the execution of CLI commands on page 4786
- Log buffer on FortiGates with an SSD disk on page 4788
- Source and destination UUID logging on page 4790
- Configuring and debugging the free-style filter on page 4792
- Logging the signal-to-noise ratio and signal strength per client on page 4794
- RSSO information for authenticated destination users in logs on page 4796
- Destination user information in UTM logs on page 4799
- Message ID in UTM logs on page 4803
- Log fields for long-live sessions on page 4804
- Generate unique user name for anonymized logs on page 4806
- Configuring syslog overrides for VDOMs on page 4809
- Customizing log formats for syslog servers NEW on page 4812
- Logging MAC address flapping events on page 4818
- Incorporating endpoint device data in the web filter UTM logs on page 4818
- Logging detection of duplicate IPv4 addresses on page 4819
- Including zone information fields in logs on page 4824
- Local in and local out logging on page 4827
- TACACS+ accounting details in audit logs NEW on page 4828

Logs for the execution of CLI commands

The `cli-audit-log` option records the execution of CLI commands in system event logs (log ID 44548). In addition to `execute` and `config` commands, `show`, `get`, and `diagnose` commands are recorded in the system event logs.

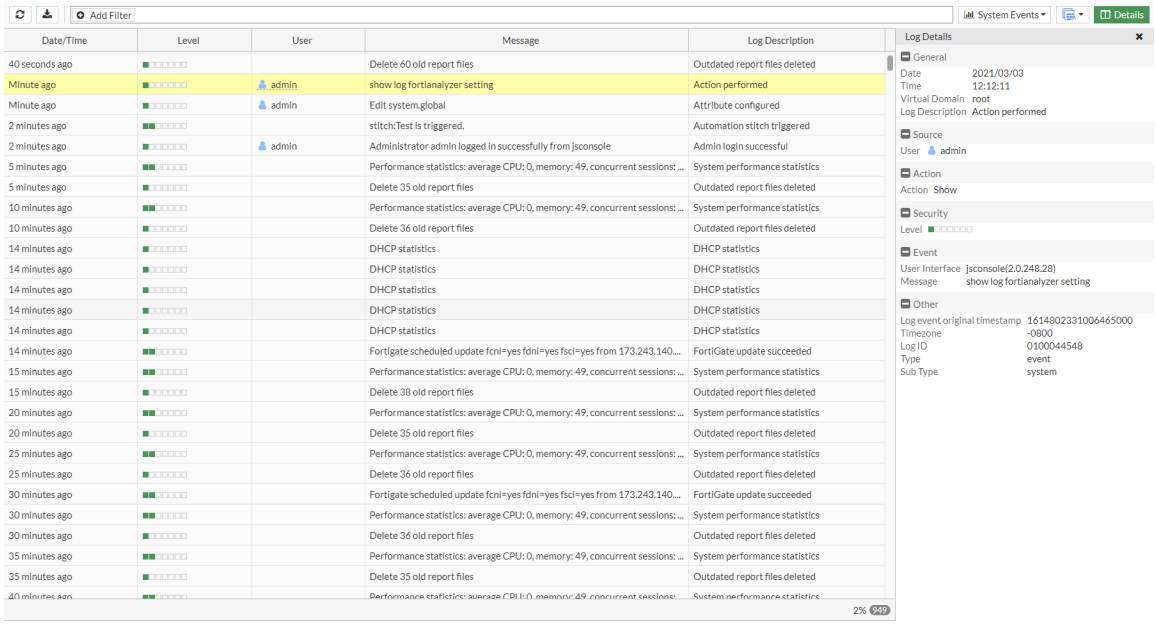
The `cli-audit-log` data can be recorded on memory or disk, and can be uploaded to FortiAnalyzer, FortiGate Cloud, or a syslog server.

To enable the CLI audit log option:

```
config system global
    set cli-audit-log enable
end
```

To view system event logs in the GUI:

1. Run the command in the CLI (`# show log fortianalyzer setting`).
2. Go to *Log & Report* > *System Events*.
3. Select *General System Events*.
4. Select the log entry and click *Details*.



To display the logs:

```
# execute log filter device disk
```

```
# execute log filter category event
```

```
# execute log filter field subtype system
```

```
# execute log filter field logid 0100044548
```

```
# execute log display
```

Sample log:

```
1: date=2020-11-16 time=10:43:00 eventtime=1605552179970875703 tz="-0800" logid="0100044548"
type="event" subtype="system" level="information" vd="root" logdesc="Action performed"
user="admin" ui="jsconsole(2.0.225.112)" action="Show" msg="show log fortianalyzer setting"
```

```
2: date=2020-11-16 time=10:42:43 eventtime=1605552163502003054 tz="-0800" logid="0100044548"
type="event" subtype="system" level="information" vd="root" logdesc="Action performed"
user="admin" ui="jsconsole(2.0.225.112)" action="Get" msg="get sys status"
```

```
3: date=2020-11-16 time=09:47:04 eventtime=1605548824762387718 tz="-0800" logid="0100044548"
type="event" subtype="system" level="information" vd="root" logdesc="Action performed"
user="admin" ui="jsconsole(2.0.228.202)" action="Diagnose" msg="diagnose log test"
```

Log buffer on FortiGates with an SSD disk

FortiGates with an SSD disk have a configurable log buffer. When the connection to FortiAnalyzer is unreachable, the FortiGate is able to buffer logs on disk if the memory log buffer is full. The logs queued on the disk buffer can be sent successfully once the connection to FortiAnalyzer is restored.

The queued logs are buffered to the memory first and then disk. If the total buffer is full, new logs will overwrite the old logs.

To configure the log buffer:

1. Allocate disk space (MB) to temporarily store logs to FortiAnalyzer:

```
config system global
    set faz-disk-buffer-size 200
end
```

2. Check the `fgtlogd` statistics. The 200 MB disk buffer has been set, and there are currently no logs buffered in memory or on disk when FortiAnalyzer is reachable:

```
# diagnose test application fgtlogd 41
cache maximum: 19569745(18MB) objects: 0 used: 0(0MB) allocated: 0(0MB)
   VDOM:root
Memory queue for: global-faz
        queue:
                num:0 size:0(0MB) total size:0(0MB) max:15375441(14MB) logs:0
        queue disk total size:0MB, max size:200MB
                total items:0
                        devid:-1-13-0-0
                        buffer path:/var/log/log/qbuf/13.0/0
                        saved size:0MB, lost files: 0
                        save roll:0 restore roll:0
Confirm queue for: global-faz
        queue:
                num:0 size:0(0MB) total size:0(0MB) max:15375441(14MB) logs:0
```

3. Disable the connection between the FortiGate and FortiAnalyzer. For example, delete the FortiGate from the FortiAnalyzer authorized device list.
   Assuming a massive number of logs (~ 300000) are recorded during this downtime, the logs will be queued in the memory buffer first. If the memory buffer is full, then the remaining logs will be queued on the disk buffer.
4. Recheck the `fgtlogd` statistics. Currently, there are logs buffered in both memory and disk:

```
# diagnose test application fgtlogd 41
cache maximum: 19569745(18MB) objects: 14391 used: 10450754(9MB) allocated: 12089088(11MB)
```

```
   VDOM:root
Memory queue for: global-faz
        queue:
                num:14245 size:9306505(8MB) total size:10450754(9MB) max:15375441(14MB)
logs:14245
        queue disk total size:199MB, max size:200MB
                total items:321085
                        devid:-1-13-0-0
                        buffer path:/var/log/log/qbuf/13.0/0
                        saved size:199MB, lost files: 10
                        save roll:60 restore roll:10
Confirm queue for: global-faz
        queue:
                num:0 size:0(0MB) total size:10450754(9MB) max:15375441(14MB) logs:0
```

The overall `fgtlogd` statistics shows the total cached logs is the sum of the logs buffered in memory and on disk:

```
# diagnose test application fgtlogd 4
Queues in all miglogds: cur:1  total-so-far:727973
global log dev statistics:
faz=399411, faz_cloud=0, fds_log=399411
faz 0: sent=0, failed=0, cached=335480, dropped=0
Num of REST URLs: 0
```

5. Enable the connection between FortiAnalyzer and the FortiGate.
6. After a while, check the `fgtlogd` statistics to confirm that all buffered logs are being sent to FortiAnalyzer successfully:

```
# diagnose test application fgtlogd 4
Queues in all miglogds: cur:1  total-so-far:727973
global log dev statistics:
faz=399411, faz_cloud=0, fds_log=399411
faz 0: sent=335487, failed=0, cached=0, dropped=0
Num of REST URLs: 0
```

```
# diagnose test application fgtlogd 41
cache maximum: 19569745(18MB) objects: 0 used: 0(0MB) allocated: 0(0MB)
VDOM:root
Memory queue for: global-faz
        queue:
                num:0 size:0(0MB) total size:0(0MB) max:15375441(14MB) logs:0
        queue disk total size:0MB, max size:200MB
                total items:0
                        devid:-1-13-0-0
                        buffer path:/var/log/log/qbuf/13.0/0
                        saved size:0MB, lost files: 10
                        save roll:60 restore roll:60
Confirm queue for: global-faz
        queue:
                num:0 size:0(0MB) total size:0(0MB) max:15375441(14MB) logs:0
```

Source and destination UUID logging

The traffic log setting includes three UUID fields: Source UUID (srcuuid), Destination UUID (dstuuid), and Policy UUID (poluuid). It also includes two `internet-service` name fields: *Source Internet Service* (`srcinetsvc`) and *Destination Internet Service* (`dstinetsvc`).

Log UUIDs

All policy types have a UUID field that is auto-generated by FortiOS when the policy is created, and can be viewed in the CLI using the show command. For example:

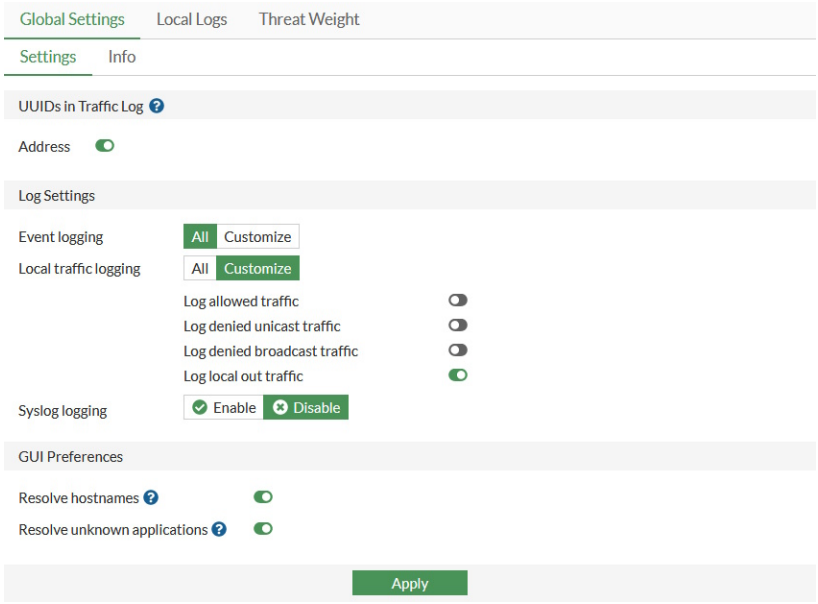
```
# show firewall policy 1
config firewall policy
    edit 1
        set name "client_yt_v4"
        set uuid f4fe48a4-938c-51ee-8856-3e84e3b24af4
        ...
    next
end
```

UUIDs can be matched for each source and destination that match a policy that is added to the traffic log. This allows the address objects to be referenced in log analysis and reporting.

As this may consume a significant amount of storage space, this feature is optional. By default, address UUID insertion is disabled.

To enable address UUID insertion in traffic logs in the GUI:

1. Go to *Log & Report* > *Log Settings*.
2. Under *UUIDs in Traffic Log*, enable *Address*.



3. Click *Apply*.

To enable address UUID insertion in traffic logs in the CLI:

```
config system global
   set log-uuid-address enable
end
```

Sample log

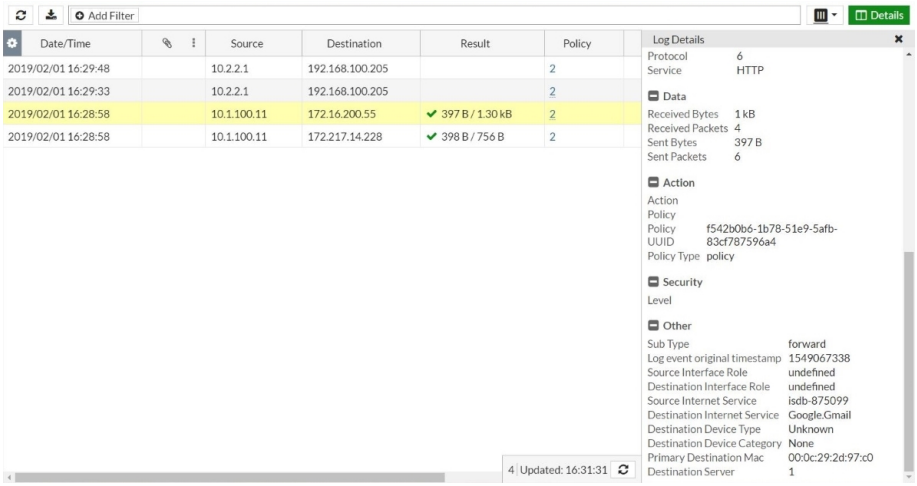
```
date=2019-01-25 time=11:32:55 logid="0000000013" type="traffic" subtype="forward" level="notice"
vd="vdom1" eventtime=1528223575srcip=192.168.1.183 srcname="PC24" srcport=33709 srcintf="lan"
srcintfrole="lan" dstip=192.168.70.184 dstport=80 dstintf="wan1" dstintfrole="wan"
srcuuid="27dd503e 883c 51e7-ade1-7e015d46494f" dstuuid="27dd503e-883c-51e7-ade1-7e015d46494f"
poluuid="9e0fe24c-1808-51e8-1257-68ce4245572c" sessionid=5181 proto=6 action="client-rst"
policyid=4 policytype="policy" service="HTTP" trandisp="snat" transip=192.168.70.228
transport=33709 appid=38783 app="Wget" appcat="General.Interest" apprisk="low" applist="default"
duration=5 sentbyte=450 rcvdbyte=2305 sentpkt=6 wanin=368 wanout=130 lanin=130 lanout=130
utmaction="block" countav=2 countapp=1 crscore=50 craction=2 devtype="Linux PC" devcategory="None"
osname="Linux" mastersrcmac="00:0c:29:36:5c:c3" srcmac="00:0c:29:36:5c:c3" srcserver=0
utmref=65523-1018
```

Internet service name fields

Traffic logs for `internet-service` include two fields: *Source Internet Service* and *Destination Internet Service*.

To view the internet service fields using the GUI:

1. Go to *Log & Report* > *Forward Traffic*.
2. Double-click on an entry to view the *Log Details*. The *Source Internet Service* and *Destination Internet Service* fields are visible in the *Log Details* pane.



Sample log

```
date=2019-01-25 time=14:17:04 logid="0000000013" type="traffic" subtype="forward" level="notice"
vd="vdom1" eventtime=1548454622
srcip=10.1.100.11 srcport=51112 srcintf="port3" srcintfrole="undefined" dstip=172.217.14.228
dstport=80 dstintf="port1" dstintfrole="undefined" poluuid="af519380-2094-51e9-391c-b78e8edbddfc"
srcinetsvc="isdb-875099" dstinetsvc="Google.Gmail" sessionid=6930 proto=6 action="close"
policyid=2 policytype="policy" service="HTTP" dstcountry="United States" srccountry="Reserved"
trandisp="snat" transip=172.16.200.2 transport=51112 duration=11 sentbyte=398 rcvdbyte=756
sentpkt=6 rcvdpkt=4 appcat="unscanned" devtype="Router/NAT Device" devcategory="Fortinet Device"
mastersrcmac="90:6c:ac:41:7a:24" srcmac="90:6c:ac:41:7a:24" srcserver=0 dstdevtype="Unknown"
dstdevcategory="Fortinet Device" masterdstmac="08:5b:0e:1f:ed:ed" dstmac="08:5b:0e:1f:ed:ed"
dstserver=0
```

Configuring and debugging the free-style filter

Free-style filters allow users to define a filter for logs that are captured to each individual logging device type. Filters can include log categories and specific log fields. The filters can be created as an inclusive list or exclusive list.

Free-style filters can also be used to filter logs that have been captured on logging devices already to narrow down the list of logs to view.

```
config log syslogd filter
    config free-style
        edit <id>
            set category <option>
            set filter <string>
            set filter-type {include | exclude}
        next
    end
end
```

`category <option>`	Set the log category. The following options are available: `traffic`, `event`, `virus`, `webfilter`, `attack`, `spam`, `anomaly`, `voip`, `dlp`, `app-ctrl`, `waf`, `dns`, `ssh`, `ssl`, `file-filter`, `icap`, and `ztna`.
`filter <string>`	Enter the filter criteria. Multiple values can be added, for example: `set filter "logid <id> <id>"`
`filter-type {include | exclude}`	Include/exclude logs that match the filter.

Use the following commands to view the results when multiple fields are used:

```
# execute log filter free-style "logid <id> <id>"
```

```
# execute log filter free-style "srcip <IP_address> <IP_address>"
```

```
# execute log filter free-style "(logid <id>) or (srcip <IP_address> <IP_address>)"
```

```
# execute log filter free-style "(srcip <IP_address>) and (dstip <IP_address>)"
```

In this example, the free-style filter is set to filter log IDs 0102043039 and 0102043040. The source IPs, 192.168.2.5 and 192.168.2.205, are also checked.

To configure the syslogd free-style filter with multiple values:

```
config log syslogd filter
    config free-style
        edit 1
            set category event
            set filter "logid 0102043039 0102043040"
        next
    end
end
```

To view the syslogd free-style filter results:

```
# execute log filter free-style "logid 0102043039 0102043040"
# execute log filter dump
category: event
device: disk
start-line: 1
view-lines: 10
max-checklines: 0
HA member:
log search mode: on-demand
pre-fetch-pages: 2
Filter: logid 0102043039 0102043040
Oftp search string: (and (or logid=="0102043039" not-exact logid=="0102043040" not-exact))
```

```
# execute log filter free-style "(logid 0102043039) or (srcip 192.168.2.5 192.168.2.205)"
# execute log filter dump
category: event
device: disk
start-line: 1
view-lines: 10
max-checklines: 0
HA member:
log search mode: on-demand
pre-fetch-pages: 2
Filter: (logid 0102043039) or (srcip 192.168.2.5 192.168.2.205)
Oftp search string: (or (or (or srcip==192.168.2.5) (or srcip==192.168.2.205)) (or
logid=="0102043039" not-exact))
```

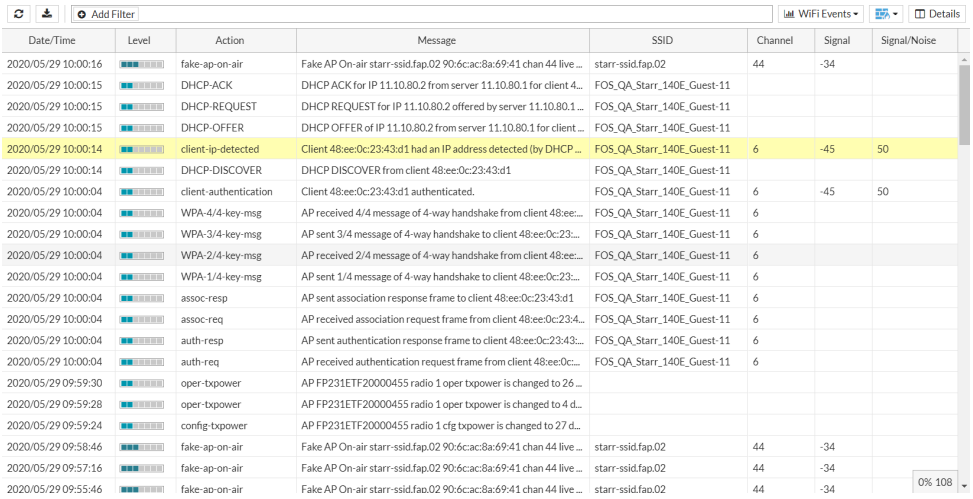
Logging the signal-to-noise ratio and signal strength per client

The signal-to-noise ratio (`snr`) and signal strength (`signal`) are logged per client in the WiFi event and traffic logs.

When a WiFi client connects to a tunnel or local-bridge mode SSID on an FortiAP that is managed by a FortiGate, signal-to-noise ratio and signal strength details are included in WiFi event logs for local-bridge traffic statistics and authentication, and in forward traffic logs for tunnel traffic. This allows you to store and view clients' historical signal strength and signal-to-noise ratio information.

To verify when a client is connecting to an SSID:

1. Go to *Log & Report* > *System Events* and select *WiFi Events*.
   The *Signal* and *Signal/Noise* columns show the signal strength and signal-to-noise ratio for each applicable client.



Date/Time	Level	Action	Message	SSID	Channel	Signal	Signal/Noise
2020/05/29 10:00:16		fake-ap-on-air	Fake AP On-air starr-ssid.fap.02 90:6c:ac:8a:69:41 chan 44 live ...	starr-ssid.fap.02	44	-34	
2020/05/29 10:00:15		DHCP-ACK	DHCP ACK for IP 11.10.80.2 from server 11.10.80.1 for client 4...	FOS_QA_Starr_140E_Guest-11			
2020/05/29 10:00:15		DHCP-REQUEST	DHCP REQUEST for IP 11.10.80.2 offered by server 11.10.80.1 ...	FOS_QA_Starr_140E_Guest-11			
2020/05/29 10:00:15		DHCP-OFFER	DHCP OFFER of IP 11.10.80.2 from server 11.10.80.1 for client ...	FOS_QA_Starr_140E_Guest-11			
2020/05/29 10:00:14		client-ip-detected	Client 48:ee:0c:23:43:d1 had an IP address detected (by DHCP ...	FOS_QA_Starr_140E_Guest-11	6	-45	50
2020/05/29 10:00:14		DHCP-DISCOVER	DHCP DISCOVER from client 48:ee:0c:23:43:d1	FOS_QA_Starr_140E_Guest-11			
2020/05/29 10:00:04		client-authentication	Client 48:ee:0c:23:43:d1 authenticated.	FOS_QA_Starr_140E_Guest-11	6	-45	50
2020/05/29 10:00:04		WPA-4/4-key-msg	AP received 4/4 message of 4-way handshake from client 48:ee:...	FOS_QA_Starr_140E_Guest-11	6		
2020/05/29 10:00:04		WPA-3/4-key-msg	AP sent 3/4 message of 4-way handshake to client 48:ee:0c:23:...	FOS_QA_Starr_140E_Guest-11	6		
2020/05/29 10:00:04		WPA-2/4-key-msg	AP received 2/4 message of 4-way handshake from client 48:ee:...	FOS_QA_Starr_140E_Guest-11	6		
2020/05/29 10:00:04		WPA-1/4-key-msg	AP sent 1/4 message of 4-way handshake to client 48:ee:0c:23:...	FOS_QA_Starr_140E_Guest-11	6		
2020/05/29 10:00:04		assoc-resp	AP sent association response frame to client 48:ee:0c:23:43:d1	FOS_QA_Starr_140E_Guest-11	6		
2020/05/29 10:00:04		assoc-req	AP received association request frame from client 48:ee:0c:23:4...	FOS_QA_Starr_140E_Guest-11	6		
2020/05/29 10:00:04		auth-resp	AP sent authentication response frame to client 48:ee:0c:23:43:...	FOS_QA_Starr_140E_Guest-11	6		
2020/05/29 10:00:04		auth-req	AP received authentication request frame from client 48:ee:0c:...	FOS_QA_Starr_140E_Guest-11	6		
2020/05/29 09:59:30		oper-txpower	AP FP231ETF20000455 radio 1 oper txpower is changed to 26 ...				
2020/05/29 09:59:28		oper-txpower	AP FP231ETF20000455 radio 1 oper txpower is changed to 4 d...				
2020/05/29 09:59:24		config-txpower	AP FP231ETF20000455 radio 1 cfg txpower is changed to 27 d...				
2020/05/29 09:58:46		fake-ap-on-air	Fake AP On-air starr-ssid.fap.02 90:6c:ac:8a:69:41 chan 44 live ...	starr-ssid.fap.02	44	-34	
2020/05/29 09:57:16		fake-ap-on-air	Fake AP On-air starr-ssid.fap.02 90:6c:ac:8a:69:41 chan 44 live ...	starr-ssid.fap.02	44	-34	
2020/05/29 09:55:46		fake-ap-on-air	Fake AP On-air starr-ssid.fap.02 90:6c:ac:8a:69:41 chan 44 live ...	starr-ssid.fap.02	44	-34	

2. WiFi event log messages include the `signal` and `snr` values:

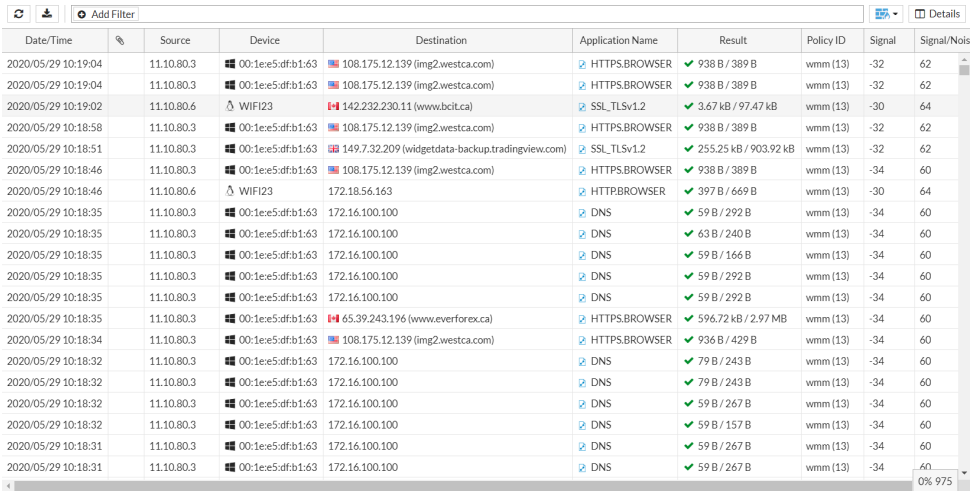
```
date=2020-05-27 time=11:26:28 logid="0104043579" type="event" subtype="wireless"
level="notice" vd="vdom1" eventtime=1590603988877156921 tz="-0700" logdesc="Wireless client IP
assigned" sn="FP231ETF20000455" ap="FP231ETF20000455" vap="stability3" ssid="FOS_QA_Starr_
140E_Guest-11" radioid=1 user="N/A" group="N/A" stamac="1c:87:2c:b6:a8:49" srcip=11.10.80.2
channel=6 radioband="802.11n,g-only" signal=-45 snr=50 security="WPA2 Personal"
encryption="AES" action="client-ip-detected" reason="Reserved 0" mpsk="N/A" msg="Client
1c:87:2c:b6:a8:49 had an IP address detected (by DHCP packets)."
```

```
date=2020-05-27 time=11:26:11 logid="0104043573" type="event" subtype="wireless"
level="notice" vd="vdom1" eventtime=1590603970962702892 tz="-0700" logdesc="Wireless client
authenticated" sn="FP231ETF20000455" ap="FP231ETF20000455" vap="stability3" ssid="FOS_QA_
```

```
Starr_140E_Guest-11" radioid=1 user="N/A" group="N/A" stamac="1c:87:2c:b6:a8:49" srcip=0.0.0.0
channel=6 radioband="802.11n,g-only" signal=-45 snr=50 security="WPA2 Personal"
encryption="AES" action="client-authentication" reason="Reserved 0" mpsk="N/A" msg="Client
1c:87:2c:b6:a8:49 authenticated."
```

To verify tunnel traffic when a client is connecting to a tunnel mode SSID:

1. Go to *Log & Report* > *Forward Traffic*.
   The *Signal* and *Signal/Noise* columns show the signal strength and signal-to-noise ratio for each applicable client.



Date/Time	Source	Device	Destination	Application Name	Result	Policy ID	Signal	Signal/Nois
2020/05/29 10:19:04	11.10.80.3	00:1e:e5:df:b1:63	108.175.12.139 (img2.westca.com)	HTTPS.BROWSER	938 B / 389 B	wmm (13)	-32	62
2020/05/29 10:19:04	11.10.80.3	00:1e:e5:df:b1:63	108.175.12.139 (img2.westca.com)	HTTPS.BROWSER	938 B / 389 B	wmm (13)	-32	62
2020/05/29 10:19:02	11.10.80.6	WIFI23	142.232.230.11 (www.bcit.ca)	SSL_TLSv1.2	3.67 kB / 97.47 kB	wmm (13)	-30	64
2020/05/29 10:18:58	11.10.80.3	00:1e:e5:df:b1:63	108.175.12.139 (img2.westca.com)	HTTPS.BROWSER	938 B / 389 B	wmm (13)	-32	62
2020/05/29 10:18:51	11.10.80.3	00:1e:e5:df:b1:63	149.7.32.209 (widgetdata-backup.tradingview.com)	SSL_TLSv1.2	255.25 kB / 903.92 kB	wmm (13)	-32	62
2020/05/29 10:18:46	11.10.80.3	00:1e:e5:df:b1:63	108.175.12.139 (img2.westca.com)	HTTPS.BROWSER	938 B / 389 B	wmm (13)	-34	60
2020/05/29 10:18:46	11.10.80.6	WIFI23	172.18.56.163	HTTP.BROWSER	397 B / 669 B	wmm (13)	-30	64
2020/05/29 10:18:35	11.10.80.3	00:1e:e5:df:b1:63	172.16.100.100	DNS	59 B / 292 B	wmm (13)	-34	60
2020/05/29 10:18:35	11.10.80.3	00:1e:e5:df:b1:63	172.16.100.100	DNS	63 B / 240 B	wmm (13)	-34	60
2020/05/29 10:18:35	11.10.80.3	00:1e:e5:df:b1:63	172.16.100.100	DNS	59 B / 166 B	wmm (13)	-34	60
2020/05/29 10:18:35	11.10.80.3	00:1e:e5:df:b1:63	172.16.100.100	DNS	59 B / 292 B	wmm (13)	-34	60
2020/05/29 10:18:35	11.10.80.3	00:1e:e5:df:b1:63	172.16.100.100	DNS	59 B / 292 B	wmm (13)	-34	60
2020/05/29 10:18:35	11.10.80.3	00:1e:e5:df:b1:63	65.39.243.196 (www.everforex.ca)	HTTPS.BROWSER	596.72 kB / 2.97 MB	wmm (13)	-34	60
2020/05/29 10:18:34	11.10.80.3	00:1e:e5:df:b1:63	108.175.12.139 (img2.westca.com)	HTTPS.BROWSER	936 B / 429 B	wmm (13)	-34	60
2020/05/29 10:18:32	11.10.80.3	00:1e:e5:df:b1:63	172.16.100.100	DNS	79 B / 243 B	wmm (13)	-34	60
2020/05/29 10:18:32	11.10.80.3	00:1e:e5:df:b1:63	172.16.100.100	DNS	79 B / 243 B	wmm (13)	-34	60
2020/05/29 10:18:32	11.10.80.3	00:1e:e5:df:b1:63	172.16.100.100	DNS	59 B / 267 B	wmm (13)	-34	60
2020/05/29 10:18:32	11.10.80.3	00:1e:e5:df:b1:63	172.16.100.100	DNS	59 B / 157 B	wmm (13)	-34	60
2020/05/29 10:18:31	11.10.80.3	00:1e:e5:df:b1:63	172.16.100.100	DNS	59 B / 267 B	wmm (13)	-34	60
2020/05/29 10:18:31	11.10.80.3	00:1e:e5:df:b1:63	172.16.100.100	DNS	59 B / 267 B	wmm (13)	-34	60

2. Forward traffic log messages include the `signal` and `snr` values:

```
date=2020-05-27 time=11:30:26 logid="0000000013" type="traffic" subtype="forward"
level="notice" vd="vdom1" eventtime=1590604226533016978 tz="-0700" srcip=11.10.80.2
srcname="WIFI23" srcport=53926 srcintf="stability3" srcintfrole="lan" srcssid="FOS_QA_Starr_
140E_Guest-11" apsn="FP231ETF20000455" ap="FP231ETF20000455" channel=6 radioband="802.11n,g-
only" signal=-31 snr=64 dstip=91.189.91.157 dstport=123 dstintf="wan1" dstintfrole="wan"
srccountry="United States" dstcountry="United States" sessionid=322069 proto=17
action="accept" policyid=13 policytype="policy" poluuid="7c14770c-1456-51e9-4c57-806e9c499782"
policyname="wmm" service="NTP" trandisp="snat" transip=172.16.200.111 transport=53926
appid=16270 app="NTP" appcat="Network.Service" apprisk="elevated" applist="g-default"
duration=180 sentbyte=76 rcvdbyte=76 sentpkt=1 rcvdpkt=1 utmaction="allow" countapp=1
osname="Linux" mastersrcmac="1c:87:2c:b6:a8:49" srcmac="1c:87:2c:b6:a8:49" srcserver=0
utmref=65534-66
```

To verify local-bridge traffic statistics when a client is connecting to a local-bridge mode SSID:

1. Go to *Log & Report* > *System Events* and select *WiFi Events*.
   The *Signal* and *Signal/Noise* columns show the signal strength and signal-to-noise ratio for each applicable client.

Date/Time	Level	Action	Message	SSID	Channel	Signal	Signal/Noise
2020/05/29 10:44:44		sta-wl-bridge-traffic-stats	Traffic stats for bridge ssid client 00:1e:e5:df:b1:63	FOS_QA_Starr-140E-LB		-53	51
2020/05/29 10:39:44		sta-wl-bridge-traffic-stats	Traffic stats for bridge ssid client 00:1e:e5:df:b1:63	FOS_QA_Starr-140E-LB		-54	50
2020/05/29 10:34:44		sta-wl-bridge-traffic-stats	Traffic stats for bridge ssid client 00:1e:e5:df:b1:63	FOS_QA_Starr-140E-LB		-54	51
2020/05/29 10:29:44		sta-wl-bridge-traffic-stats	Traffic stats for bridge ssid client 00:1e:e5:df:b1:63	FOS_QA_Starr-140E-LB		-52	52

2. WiFi event log messages include the `signal` and `snr` values:

```
date=2020-05-26 time=17:48:57 logid="0104043687" type="event" subtype="wireless"
level="information" vd="vdom1" eventtime=1590540537841497433 tz="-0700" logdesc="Traffic stats
for station with bridge wlan" sn="FP231ETF20000455" ap="FP231ETF20000455" vap="wifi.fap.01"
ssid="FOS_QA_Starr-140E-LB-cap-2" srcip=10.128.100.4 user="N/A" stamac="00:1e:e5:df:b1:63"
signal=-53 snr=52 sentbyte=8970016 rcvdbyte=985910 nextstat=300 action="sta-wl-bridge-traffic-
stats" msg="Traffic stats for bridge ssid client 00:1e:e5:df:b1:63"
```

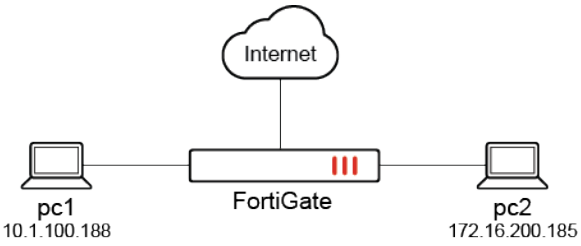
RSSO information for authenticated destination users in logs

FortiGate can use RSSO accounting information from authenticated RSSO users to populate destination users and groups, along with source users and groups.

RSSO user login information can be forwarded by the RADIUS server to the FortiGate that is listening for incoming RADIUS accounting start messages on the RADIUS accounting port. Accounting start messages usually contain the IP address, user name, and user group information. FortiGate uses this information in traffic logs, which include *dstuser* and *dstgroup* fields for user and group destination information .

For instructions on configuring RSSO, see RADIUS single sign-on agent on page 4688.

The three following scenarios show traffic between pc1 and the internet, and pc1 and pc2.



Scenario 1

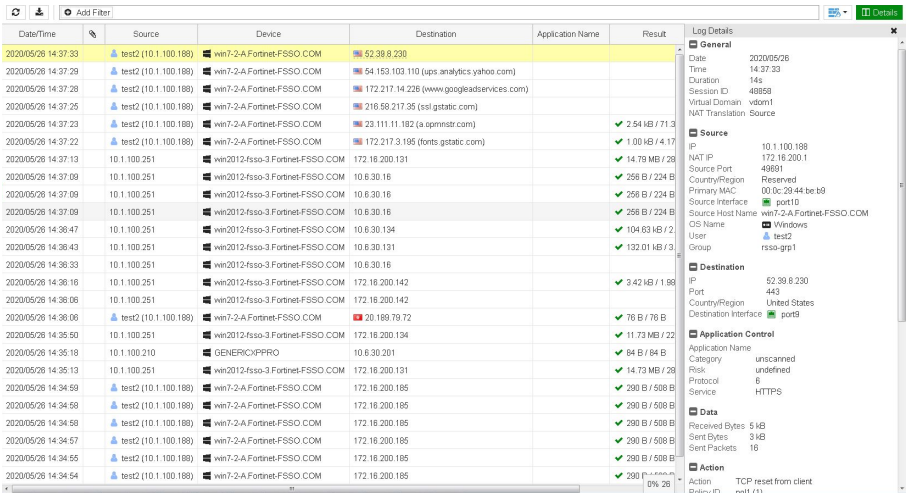
In this scenario, RSSO user *test2* in group *rsso-grp1* is authenticated on pc1. Traffic flows from pc1 to the internet.

Expected result:

In the logs, user *test2* is shown as the source user in the *rsso-grp1* group.

To verify the results:

1. In the GUI, go to *Log & Report* > *Forward Traffic* and view the details of an entry with test2 as the source.
2. In the *Source* section, *User* is *test2* and *Group* is the *rsso-grp1*.



3. The log message shows the user and group:

```
10: date=2020-05-25 time=15:34:43 logid="0000000013" type="traffic" subtype="forward"
level="notice" vd="vdom1" eventtime=1590446083718007055 tz="-0700" srcip=10.1.100.188
srcname="win7-2-A.Fortinet-FSSO.COM" srcport=56982 srcintf="port10" srcintfrole="undefined"
dstip=172.217.3.195 dstport=443 dstintf="port9" dstintfrole="undefined" srccountry="Reserved"
dstcountry="United States" sessionid=120651 proto=17 action="accept" policyid=1
policytype="policy" poluuid="d130f886-9ec6-51ea-206e-8c561c5244c6" policyname="pol1"
user="test2" group="rsso-grp1" authserver="vdom1" service="udp/443" trandisp="snat"
transip=172.16.200.1 transport=56982 duration=181 sentbyte=2001 rcvdbyte=1820 sentpkt=6
rcvdpkt=4 appcat="unscanned" sentdelta=0 rcvddelta=0 srchwvendor="VMware" osname="Windows"
srcswversion="7" mastersrcmac="00:0c:29:44:be:b9" srcmac="00:0c:29:44:be:b9" srcserver=0
```

Scenario 2

In this scenario, RSSO user *test2* is authenticated on pc1. Traffic is initialized on pc2 (172.16.200.185) going to pc1 (10.1.100.188).

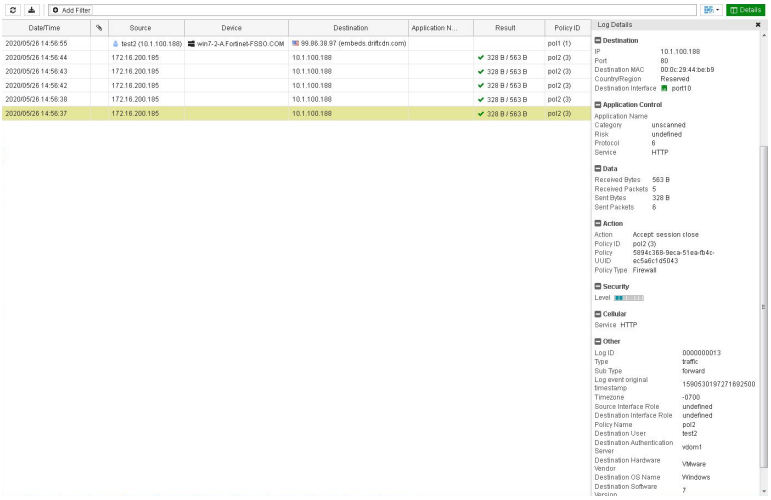
Expected result:

In the logs, user *test2* is shown as the destination user (`dstuser`). No destination group (`dstgroup`) is logged because no RSSO user is logged in on pc2, so the traffic from pc2 is unauthenticated.

To verify the results:

1. In the GUI, go to *Log & Report* > *Forward Traffic* and view the details of an entry with 172.16.200.185 (pc2) as the source.

2. In the *Other* section, *Destination User* is *test2* and no destination group is shown.



3. The log message shows the destination user:

```
1: date=2020-05-22 time=07:38:06 logid="0000000020" type="traffic" subtype="forward"
level="notice" vd="root" eventtime=1590158286585506922 tz="-0700" srcip=172.16.200.185
identifier=1 srcintf="port9" srcintfrole="undefined" dstip=10.1.100.188 dstintf="port10"
dstintfrole="undefined" srccountry="Reserved" dstcountry="Reserved" sessionid=4395 proto=1
action="accept" policyid=3 policytype="policy" poluuid="d4f18e1e-9c36-51ea-6ec0-3a354d5910ee"
policyname="pol2" dstuser="test2" dstauthserver="root" service="PING" trandisp="snat"
transip=10.1.100.1 transport=0 duration=128 sentbyte=7620 rcvdbyte=5220 sentpkt=127 rcvdpkt=87
appcat="unscanned" sentdelta=7620 rcvddelta=5220
```

Scenario 3

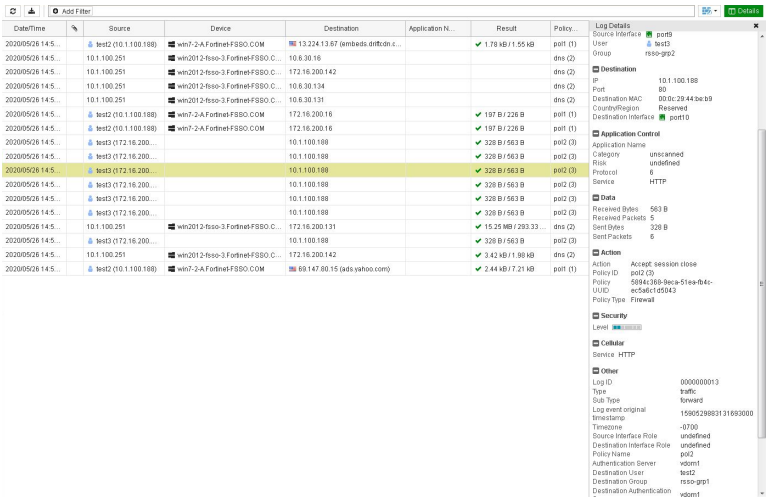
In this scenario, RSSO user *test2* in group *rsso-grp1* is authenticated on pc1, and user *test3* in group *rsso-grp2* is authenticated on pc2. Traffic flows from pc2 to pc1.

Expected result:

In the logs, user *test3* is shown as the source user in the *rsso-grp1* group. User *test2* is shown as destination user (`dstuser`) in the *rsso-grp1* destination group (`dstgroup`). The destination group is logged because an RSSO user is logged in to pc2.

To verify the results:

1. In the GUI, go to *Log & Report* > *Forward Traffic* and view the details of an entry with 172.16.200.185 (pc2) as the source.
2. In the *Source* section, *User* is *test3* and *Group* is the *rsso-grp2*. In the *Other* section, *Destination User* is *test2* and *Destination Group* is *rsso-grp1*.



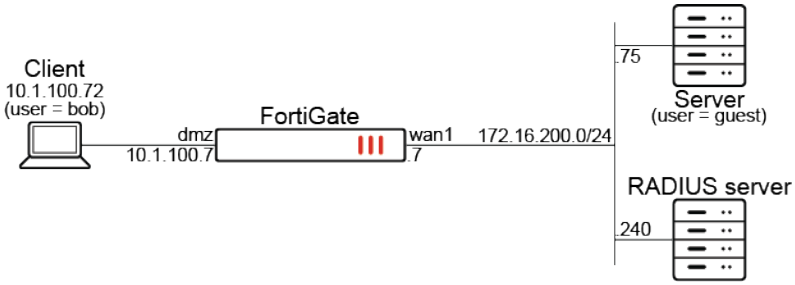
3. The log message shows both the source and the destination users and groups:

```
8: date=2020-05-25 time=14:23:07 logid="0000000013" type="traffic" subtype="forward"
level="notice" vd="vdom1" eventtime=1590441786958007914 tz="-0700" srcip=172.16.200.185
srcport=64096 srcintf="port9" srcintfrole="undefined" dstip=10.1.100.188 dstport=80
dstintf="port10" dstintfrole="undefined" srccountry="Reserved" dstcountry="Reserved"
sessionid=112445 proto=6 action="close" policyid=3 policytype="policy" poluuid="5894c368-9eca-
51ea-fb4c-ec5a6c1d5043" policyname="pol2" user="test3" group="rsso-grp2" authserver="vdom1"
dstuser="test2" dstgroup="rsso-grp1" dstauthserver="vdom1" service="HTTP" trandisp="snat"
transip=10.1.100.1 transport=64096 duration=1 sentbyte=328 rcvdbyte=563 sentpkt=6 rcvdpkt=5
appcat="unscanned" dsthwvendor="VMware" dstosname="Windows" dstswversion="7"
masterdstmac="00:0c:29:44:be:b9" dstmac="00:0c:29:44:be:b9" dstserver=0
```

Destination user information in UTM logs

The `dstuser` field in UTM logs records the username of a destination device when that user has been authenticated on the FortiGate.

In the following example topology, the user, bob, is authenticated on a client computer. The user, guest, is authenticated on the server. Log are collected for AV and IPS in flow inspection mode. Logs are collected for application control and web filter in proxy mode.



To configure the RADIUS user and user groups:

1. Configure the RADIUS server:

```
config user radius
    edit "Ubuntu_docker"
        set server "172.16.200.240"
        set secret ************
    next
end
```

2. Configure the local user:

```
config user local
    edit "guest"
        set type password
        set passwd ************
    next
end
```

3. Configure the RADIUS user groups:

```
config user group
    edit "RADIUS_User_Group"
        set member "Ubuntu_docker"
    next
    edit "Local_User"
        set member "guest"
    next
end
```

Flow inspection mode

To verify AV and IPS logs in flow mode:

1. Configure the firewall policies:

```
config firewall policy
    edit 1
        set name "WAN_out"
        set srcintf "dmz"
        set dstintf "wan1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set ssl-ssh-profile "deep-inspection"
        set av-profile "g-default"
        set ips-sensor "sensor-11"
```

```
        set nat enable
        set groups "RADIUS_User_Group" "Local_User"
    next
    edit 3
        set name "WAN_in"
        set srcintf "wan1"
        set dstintf "dmz"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set logtraffic all
        set nat enable
        set groups "RADIUS_User_Group" "Local_User"
    next
end
```

2. Verify the AV log:

```
date=2021-09-14 time=16:37:25 eventtime=1631662646131356720 tz="-0700" logid="0211008192"
type="utm" subtype="virus" eventtype="infected" level="warning" vd="vdom1" policyid=1
msg="File is infected." action="blocked" service="HTTP" sessionid=4613 srcip=10.1.100.72
dstip=172.16.200.75 srcport=60086 dstport=80 srcintf="dmz" srcintfrole="undefined"
dstintf="wan1" dstintfrole="undefined" srcuuid="877d43a4-c2f9-51eb-f78f-e09794924d8a"
dstuuid="877d43a4-c2f9-51eb-f78f-e09794924d8a" proto=6 direction="incoming"
filename="eicar.com" quarskip="Quarantine-disabled" virus="EICAR_TEST_FILE" viruscat="Virus"
dtype="av-engine" ref="http://www.fortinet.com/ve?vn=EICAR_TEST_FILE" virusid=2172
url="http://172.16.200.75/eicar.com" profile="g-default" user="bob" group="RADIUS_User_Group"
authserver="Ubuntu_docker" dstuser="guest" agent="Wget/1.17.1"
analyticscksum="275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f"
analyticssubmit="false" crscore=50 craction=2 crlevel="critical"
```

3. Verify the IPS log:

```
date=2021-09-14 time=16:56:06 eventtime=1631663765992499880 tz="-0700" logid="0419016384"
type="utm" subtype="ips" eventtype="signature" level="alert" vd="vdom1" severity="info"
srcip=10.1.100.72 srccountry="Reserved" dstip=172.16.200.75 srcintf="dmz"
srcintfrole="undefined" dstintf="wan1" dstintfrole="undefined" sessionid=7881 action="dropped"
proto=6 service="HTTP" policyid=1 attack="Eicar.Virus.Test.File" srcport=60092 dstport=80
direction="incoming" attackid=29844 profile="sensor-11"
ref="http://www.fortinet.com/ids/VID29844" user="bob" group="RADIUS_User_Group"
authserver="Ubuntu_docker" dstuser="guest" incidentserialno=17825794 attackcontextid="2/2"
attackcontext="dGVudC1MZW5ndGg6IDY4DQpLZWVwLUFsaXZlOiB0aW1lb3V0PTUsIG1heD0xMDANCkNvbm5lY3Rpb24
6IEtlZXAtQWxpdmUNCkNvbnRlbnQtVHlwZTogYXBwbGljYXRpb24veC1tc2Rvcy1wcm9ncmFtDQoNClg1TyFQJUBBUFs0X
FBaWDU0KFBeKTdDQyk3fSRFSUNBUi1TVEFOREFSRC1BTlRJVklSVVMtVEVTVC1GSUxFISRIK0gqPC9QQUNLRVQ+"
```

Proxy inspection mode

To verify application control and web filter logs in proxy mode:

1. Configure the firewall policies:

```
config firewall policy
    edit 1
        set name "WAN_out"
        set srcintf "dmz"
        set dstintf "wan1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "deep-inspection"
        set av-profile "g-default"
        set application-list "g-default"
        set webfilter-profile "1"
        set nat enable
        set groups "RADIUS_User_Group" "Local_User"
    next
    edit 3
        set name "WAN_in"
        set srcintf "wan1"
        set dstintf "dmz"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set schedule "always"
        set service "ALL"
        set inspection-mode proxy
        set logtraffic all
        set nat enable
        set groups "RADIUS_User_Group" "Local_User"
    next
end
```

2. Verify the application control log:

```
date=2021-09-14 time=17:05:45 eventtime=1631664345570951500 tz="-0700" logid="1059028704"
type="utm" subtype="app-ctrl" eventtype="signature" level="information" vd="vdom1" appid=38783
user="bob" group="RADIUS_User_Group" authserver="Ubuntu_docker" dstuser="guest"
srcip=10.1.100.72 dstip=172.16.200.75 srcport=60098 dstport=80 srcintf="dmz"
srcintfrole="undefined" dstintf="wan1" dstintfrole="undefined" proto=6 service="HTTP"
direction="outgoing" policyid=1 sessionid=10871 applist="g-default" action="pass"
appcat="General.Interest" app="Wget" hostname="172.16.200.75" incidentserialno=17825796
url="/eicar.com" msg="General.Interest: Wget," apprisk="low"
```

3. Verify the web filter log:

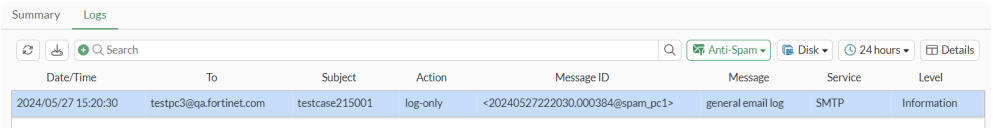
```
date=2021-09-14 time=17:14:46 eventtime=1631664886585770420 tz="-0700" logid="0315012546"
type="utm" subtype="webfilter" eventtype="urlfilter" level="information" vd="vdom1"
urlfilteridx=1 urlfilterlist="Auto-webfilter-urlfilter_caex0ojl5" policyid=1 sessionid=15251
user="bob" group="RADIUS_User_Group" authserver="Ubuntu_docker" dstuser="guest"
srcip=10.1.100.72 srcport=60106 srcintf="dmz" srcintfrole="undefined" srcuuid="877d43a4-c2f9-
51eb-f78f-e09794924d8a" dstip=172.16.200.75 dstport=80 dstintf="wan1" dstintfrole="undefined"
dstuuid="877d43a4-c2f9-51eb-f78f-e09794924d8a" proto=6 service="HTTP" hostname="172.16.200.75"
profile="1" action="passthrough" reqtype="direct" url="http://172.16.200.75/eicar.com"
sentbyte=149 rcvdbyte=0 direction="outgoing" msg="URL was allowed because it is in the URL
filter list"
```

Message ID in UTM logs

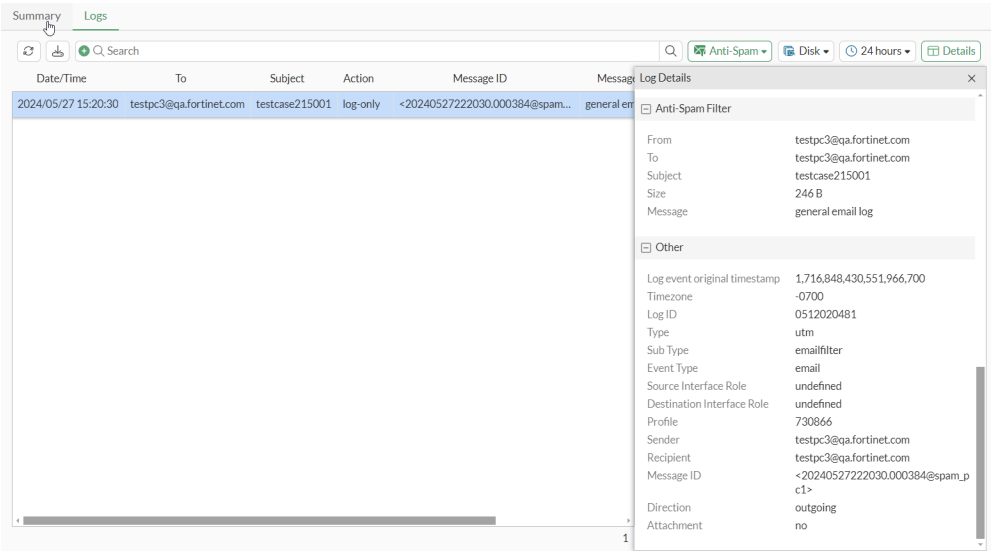
FortiOS logs the message ID (`messageid`) field in UTM logs under the email filter, file filter, and DLP subtypes. The message ID can be used with FortiMail to locate an undesired email. The message ID can also be used with FortiAnalyzer to trace the email and locate the device that sent the undesired traffic.

To view the message ID:

1. Go to *Log & Report* > *Security Events* and select *Logs*.
2. Set the filters to display the email filter, file filter, or DLP subtypes. In this example, the *Anti-Spam* and *Disk* filters are set to display an entry with the *Message ID* field.



3. Select the log, and click *Details*. The *Message ID* field is displayed.



Following are examples of the message ID (`messageid`) field in email filter, file filter, and DLP logs:

- Email filter logs:

```
1: date=2024-05-27 time=15:20:30 eventtime=1716848430551966694 tz="-0700" logid="0512020481"
type="utm" subtype="emailfilter" eventtype="email" level="information" vd="vdom1" policyid=1
poluuid="12c1682e-18a5-51ef-dc3d-459a4231c9e6" policytype="policy" sessionid=162
srcip=10.1.100.22 srcport=41344 srccountry="Reserved" srcintf="port21" srcintfrole="undefined"
srcuuid="f29a920a-18a4-51ef-4fca-8c6dc5db9e26" dstip=172.16.200.55 dstport=25
dstcountry="Reserved" dstintf="port17" dstintfrole="undefined" dstuuid="f29a920a-18a4-51ef-
4fca-8c6dc5db9e26" proto=6 service="SMTP" profile="730866" action="log-only"
from="testpc3@qa.fortinet.com" to="testpc3@qa.fortinet.com" sender="testpc3@qa.fortinet.com"
recipient="testpc3@qa.fortinet.com" messageid="<20240527222030.000384@spam_pc1>"
direction="outgoing" msg="general email log" subject="testcase215001" size="246"
attachment="no"
```

- File filter logs:

```
1: date=2024-05-27 time=18:52:21 eventtime=1716861141191537397 tz="-0700" logid="1900064001"
type="utm" subtype="file-filter" eventtype="file-filter" level="notice" vd="vdom1" policyid=1
poluuid="12c1682e-18a5-51ef-dc3d-459a4231c9e6" policytype="policy" sessionid=536
srcip=10.1.100.22 srcport=55966 srccountry="Reserved" srcintf="port21" srcintfrole="undefined"
srcuuid="f29a920a-18a4-51ef-4fca-8c6dc5db9e26" dstip=172.16.200.55 dstport=25
dstcountry="Reserved" dstintf="port17" dstintfrole="undefined" dstuuid="f29a920a-18a4-51ef-
4fca-8c6dc5db9e26" proto=6 service="SMTP" profile="msgId_test" direction="outgoing"
action="log-only" from="testpc3@qa.fortinet.com" to="testpc3@qa.fortinet.com"
sender="testpc3@qa.fortinet.com" recipient="testpc3@qa.fortinet.com"
messageid="<20240528015221.001679@spam_pc1>" subject="703400" attachment="no"
rulename="bannedFiles" filesize=105749 filetype="jpeg" msg="File was detected by file filter."
```

- DLP logs:

```
1: date=2024-05-28 time=11:59:50 eventtime=1716922790136310107 tz="-0700" logid="0954024576"
type="utm" subtype="dlp" eventtype="dlp" level="warning" vd="vdom1" ruleid=1 rulename="test_
exe" dlpextra="Sensor 'test' matching any: ('Test-Dictionary'=1) >= 1; match."
filtertype="sensor" filtercat="message" severity="medium" policyid=1 poluuid="12c1682e-18a5-
51ef-dc3d-459a4231c9e6" policytype="policy" sessionid=1976 epoch=1523674618 eventid=2
srcip=10.1.100.22 srcport=52998 srccountry="Reserved" srcintf="port21" srcintfrole="undefined"
srcuuid="f29a920a-18a4-51ef-4fca-8c6dc5db9e26" dstip=172.16.200.55 dstport=25
dstcountry="Reserved" dstintf="port17" dstintfrole="undefined" dstuuid="f29a920a-18a4-51ef-
4fca-8c6dc5db9e26" proto=6 service="SMTP" filetype="N/A" direction="outgoing" action="block"
from="testpc3@qa.fortinet.com" to="testpc3@qa.fortinet.com" sender="testpc3@qa.fortinet.com"
recipient="testpc3@qa.fortinet.com" messageid="<20240528185950.007871@spam_pc1>"
subject="731047" attachment="no" profile="testing"
```

Log fields for long-live sessions

Logging of long-live session statistics can be enabled or disabled in traffic logs.

```
config log setting
     set long-live-session-stat {enable | disable}
end
```

When enabled, traffic logs include the following fields of statistics for long-live sessions:

Duration delta (`durationdelta`)	Displays the time in seconds between the last session log and the current session log.
Sent packet delta (`sentpktdelta`)	Displays the number of sent packets. When the number of packets reported in the `sentpktdelta` field matches the number of bytes reported in the `sentpkt` field, it shows no missing logs.
Received packet delta (`rcvdpktdelta`)	Displays the number of received packets. When the number of packets reported in the `rcvdpktdelta` field matches the number of bytes reported in the `rcvdpkt` field, it shows no missing logs.

The long-live session fields enhance the granularity and accuracy of traffic longs to aid troubleshooting and analysis.

Example

In this example, logging is enabled for long-live session statistics. Log ID 20 includes the new fields for long-live sessions.

To log long-live session statistics:

1. Enable logging of long-live session statistics:

```
config log setting
     set long-live-session-stat enable
end
```

2. View information in the logs:

   In the following example, log fields are filtered for log ID 0000000020 to displays the new fields of data. The `sentpkt` field displays `205` bytes, and the `rcvdpkt` field displays `1130` bytes. The new fields (`sentpktdelta=205` and `rcvdpktdelta=1130`) display the same number of packets, which shows no logs have been lost. The `durationdelta` shows `120` seconds between the last session log and the current session log.

```
# execute log filter device Disk

# execute log filter category 0

# execute log filter field subtype forward

# execute log filter field logid 0000000020

# execute log display

1 logs found.

1 logs returned.

1: date=2023-12-07 time=14:19:59 eventtime=1701987599439429340 tz="-0800" logid="0000000020"
type="traffic" subtype="forward" level="notice" vd="vdom1" srcip=10.1.100.22 srcport=53540
```

```
srcintf="wan2" srcintfrole="undefined" dstip=172.16.200.55 dstport=80 dstintf="wan1"
dstintfrole="undefined" srccountry="Reserved" dstcountry="Reserved" sessionid=296 proto=6
action="accept" policyid=1 policytype="policy" poluuid="e538d622-53eb-51ee-8adc-f8fbb0f22fdd"
policyname="B-out" service="HTTP" trandisp="snat" transip=172.16.200.2 transport=53540
duration=120 sentbyte=10855 rcvdbyte=1397640 sentpkt=205 rcvdpkt=1130 appcat="unscanned"
sentdelta=10855 rcvddelta=1397640 durationdelta=120 sentpktdelta=205 rcvdpktdelta=1130
```

Generate unique user name for anonymized logs

With the `anonymization-hash` option, user fields in logs can be anonymized by generating a hash based on the user name and salt value. The hash for the same user will generate the same hash value, allowing the anonymized user to be correlated between logs.

```
config log setting
    set user-anonymize enable
    set anonymization-hash <salt string>
end
```

Example

In this example, user names are encrypted in traffic and event logs using the `anonymization-hash` option.

To encrypt the user name for logs in the GUI:

1. Configure the hash anonymization in the CLI:

```
config log setting
    set user-anonymize enable
    set anonymization-hash "random"
end
```

2. Configure a firewall policy with a user as a source:
   a. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
   b. For *Source*, select a user.
   c. In the *Security Profiles* section, enable *AntiVirus* and select a profile.
   d. Configure the other settings as needed.
   e. Click *OK*.
3. Verify the forward traffic log:
   a. Go to *Log & Report* > *Forward Traffic*.
   b. Select an entry and double-click to view the log details.
      The user name has a hashed value of e8557d12f6551b2d.
4. Verify the antivirus traffic log:
   a. Go to *Log & Report* > *AntiVirus*.
   b. Select an entry and double-click to view the log details.

The user name has the same hashed value. Hovering over the user name displays a *No user information* tooltip.

5. Verify the event log:
   a. Go to *Log & Report > Events > System Events*.
   b. Select an entry and double-click to view the log details.
   The administrative user has a hashed value of 6a4d668735f5517a.

To encrypt the user name for logs in the CLI:

1. Configure the hash anonymization:

```
config log setting
    set user-anonymize enable
    set anonymization-hash "random"
end
```

2. Configure a firewall policy with a user as a source:

```
config firewall policy
    edit 1
        set name "WAN_out"
        set srcintf "dmz"
        set dstintf "wan1"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssl-ssh-profile "deep-inspection"
        set av-profile "g-default"
        set nat enable
        set users "bob"
    next
end
```

3. Verify the forward traffic log. The user name has a hashed value of e8557d12f6551b2d:

```
date=2021-09-09 time=15:24:52 eventtime=1631226292981803646 tz="-0700" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="vdom1" srcip=10.1.100.72 srcport=33250
srcintf="dmz" srcintfrole="undefined" dstip=172.16.200.75 dstport=80 dstintf="wan1"
dstintfrole="undefined" srccountry="Reserved" dstcountry="Reserved" sessionid=383 proto=6
action="client-rst" policyid=1 policytype="policy" poluuid="12f6f924-c2fb-51eb-6e06-
3b997d55d5f4" policyname="WAN_out" user="e8557d12f6551b2d" dstuser="e8557d12f6551b2d"
service="HTTP" trandisp="snat" transip=172.16.200.7 transport=33250 duration=1 sentbyte=469
rcvdbyte=6331 sentpkt=6 rcvdpkt=8 appcat="unscanned" wanin=369 wanout=149 lanin=149 lanout=149
utmaction="block" countav=1 crscore=50 craction=2 srchwvendor="VMware" osname="Linux"
mastersrcmac="**:**:**:**:**:**" srcmac="**:**:**:**:**:**" srcserver=0 dsthwvendor="VMware"
```

```
dstosname="Linux" masterdstmac="**:**:**:**:**:**" dstmac="**:**:**:**:**:**" dstserver=0
utmref=0-28
```

4. Verify the antivirus traffic log. The user name has the same hashed value:

```
date=2021-09-09 time=15:24:51 eventtime=1631226291945007723 tz="-0700" logid="0211008192"
type="utm" subtype="virus" eventtype="infected" level="warning" vd="vdom1" policyid=1
msg="File is infected." action="blocked" service="HTTP" sessionid=383 srcip=10.1.100.72
dstip=172.16.200.75 srcport=33250 dstport=80 srcintf="dmz" srcintfrole="undefined"
dstintf="wan1" dstintfrole="undefined" srcuuid="877d43a4-c2f9-51eb-f78f-e09794924d8a"
dstuuid="877d43a4-c2f9-51eb-f78f-e09794924d8a" proto=6 direction="incoming"
filename="eicar.com" quarskip="File-was-not-quarantined" virus="EICAR_TEST_FILE"
viruscat="Virus" dtype="av-engine" ref="http://www.fortinet.com/ve?vn=EICAR_TEST_FILE"
virusid=2172 url="http://172.16.200.75/eicar.com" profile="g-default" user="e8557d12f6551b2d"
dstuser="e8557d12f6551b2d" agent="Wget/1.17.1" analyticssubmit="false" crscore=50 craction=2
crlevel="critical"
```

5. Verify the event log. The administrative user has a hashed value of 6a4d668735f5517a:

```
date=2021-09-09 time=09:59:09 eventtime=1631206750109938510 tz="-0700" logid="0100032001"
type="event" subtype="system" level="information" vd="vdom1" logdesc="Admin login successful"
sn="**********" user="6a4d668735f5517a" ui="https(10.6.30.254)" method="https"
srcip=10.6.30.254 dstip=10.6.30.107 action="login" status="success" reason="none"
profile="super_admin" msg="Administrator 6a4d668735f5517a logged in successfully from https
(10.6.30.254)"
```

If `user-anonymize` is enabled in the log settings and `anonymization-hash` is left blank, the user name is displayed as anonymous in the logs.

Sample traffic log:

```
date=2021-09-09 time=11:27:44 eventtime=1631212064444723180 tz="-0700" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="vdom1" srcip=10.1.100.72 srcport=33246
srcintf="dmz" srcintfrole="undefined" dstip=172.16.200.75 dstport=80 dstintf="wan1"
dstintfrole="undefined" srccountry="Reserved" dstcountry="Reserved" sessionid=1337 proto=6
action="client-rst" policyid=1 policytype="policy" poluuid="12f6f924-c2fb-51eb-6e06-3b997d55d5f4"
policyname="WAN_out" user="anonymous" dstuser="anonymous" service="HTTP" trandisp="snat"
transip=172.16.200.7 transport=33246 duration=1 sentbyte=469 rcvdbyte=6331 sentpkt=6 rcvdpkt=8
appcat="unscanned" wanin=369 wanout=149 lanin=149 lanout=149 utmaction="block" countav=1
crscore=50 craction=2 srchwvendor="VMware" osname="Linux" mastersrcmac="**:**:**:**:**:**"
srcmac="**:**:**:**:**:**" srcserver=0 dsthwvendor="VMware" dstosname="Linux"
masterdstmac="**:**:**:**:**:**" dstmac="**:**:**:**:**:**" dstserver=0 utmref=0-14
```

Sample UTM log:

```
date=2021-09-09 time=11:27:43 eventtime=1631212063400129220 tz="-0700" logid="0211008192"
type="utm" subtype="virus" eventtype="infected" level="warning" vd="vdom1" policyid=1 msg="File is
infected." action="blocked" service="HTTP" sessionid=1337 srcip=10.1.100.72 dstip=172.16.200.75
srcport=33246 dstport=80 srcintf="dmz" srcintfrole="undefined" dstintf="wan1"
dstintfrole="undefined" srcuuid="877d43a4-c2f9-51eb-f78f-e09794924d8a" dstuuid="877d43a4-c2f9-
51eb-f78f-e09794924d8a" proto=6 direction="incoming" filename="eicar.com" quarskip="File-was-not-
```

```
quarantined" virus="EICAR_TEST_FILE" viruscat="Virus" dtype="av-engine"
ref="http://www.fortinet.com/ve?vn=EICAR_TEST_FILE" virusid=2172
url="http://172.16.200.75/eicar.com" profile="g-default" user="anonymous" dstuser="anonymous"
agent="Wget/1.17.1" analyticssubmit="false" crscore=50 craction=2 crlevel="critical"
```

Sample event log:

```
date=2021-09-09 time=11:27:26 eventtime=1631212046861637260 tz="-0700" logid="0100032102"
type="event" subtype="system" level="alert" vd="vdom1" logdesc="Configuration changed"
user="anonymous" ui="jsconsole" msg="Configuration is changed in the anonymous session"
```

Configuring syslog overrides for VDOMs

Logs can be sent from non-management VDOMs to both global and VDOM-override syslog servers.

```
config log syslogd override-setting
    set use-management-vdom {enable | disable}
end
```

Command	Description
`use-management-vdom {enable | disable}`	Enable/disable use of management VDOM as source VDOM for logs sent to syslog server. • `enable`: Send logs through the management VDOM. • `disable`: Do not send logs through the management VDOM.

When `use-management-vdom` is enabled under non-management VDOM, only the management VDOM is used to forward logs to configured syslog servers. Non-management VDOM override syslog servers must be reachable through the management VDOM.

Example

This example covers the following scenarios:

- Non-management VDOM with use-management-vdom disabled on page 4810
- Non-management VDOM with use-management-vdom enabled on page 4810
- Non-management VDOM with mix of use-management-vdom enabled and disabled on page 4811

All scenarios use the following IP addresses:

- Global syslog server: 10.6.30.22
- Root VDOM gateway: 192.168.5.254
- Management VDOM (vdom1) gateway: 172.16.200.254

Non-management VDOM with use-management-vdom disabled

In this example, a global syslog server is enabled. For the root VDOM, an override syslog server is enabled with `use-management-vdom` disabled. For the management VDOM, an override syslog server is enabled. With this configuration, logs are sent to the following locations:

- All VDOMs, except the root and management VDOMs, send logs to the global syslog server (10.6.30.22).
- The root VDOM sends logs to its override syslog server at 192.168.5.44.
- The management VDOM sends logs to its override syslog server at 172.16.200.55.

To configure syslog servers:

1. Enable the global syslog server:

```
config log syslogd setting
    set status enable
    set server "10.6.30.22"
    set facility local6
end
```

2. For the root VDOM, enable an override syslog server and disable `use-management-vdom`:

```
config log syslogd override-setting
    set status enable
    set server "192.168.5.44"
    set use-management-vdom disable
    set facility local6
end
```

3. For the management VDOM (vdom1), enable an override syslog server:

```
config log syslogd override-setting
    set status enable
    set server "172.16.200.55"
    set facility local6
end
```

Non-management VDOM with use-management-vdom enabled

In this example, a global syslog server is enabled. For the root VDOM, an override syslog server and `use-management-vdom` are enabled. For the management VDOM, two override syslog servers are enabled. With this configuration, logs are sent to the following locations:

- All VDOMs, except root and management VDOMs, send logs to the global syslog server (10.6.30.22).
- The root VDOM cannot send logs to syslog servers because the servers are not reachable through the management VDOM.

  To send logs to 192.168.5.44, set *use-management-vdom* to *disable* for the root VDOM. Alternately, configure the root VDOM to use an override syslog server that is reachable through the management VDOM.
- The management VDOM sends logs to the override syslog server at 172.16.200.55.

To configure syslog servers:

1. Enable the global syslog server:

```
config log syslogd setting
    set status enable
    set server "10.6.30.22"
    set facility local6
end
```

2. For the root VDOM, enable an override syslog server and enable `use-management-vdom`:

```
config log syslogd override-setting
    set status enable
    set server "192.168.5.44"
    set use-management-vdom enable
    set facility local6
end
```

3. For the management VDOM, enable an override syslog server:

```
config log syslogd override-setting
    set status enable
    set server "172.16.200.55"
    set facility local6
end
```

Non-management VDOM with mix of use-management-vdom enabled and disabled

In this example, a global syslog server is enabled. For the root VDOM, three override syslog servers are enabled with a mix of `use-management-vdom` set to `enabled` and `disabled`. For the management VDOM, an override syslog server is enabled. With this configuration, logs are sent from non-management VDOMs to both global and VDOM-override syslog servers. The logs are sent to the following locations:

- All VDOMs, except the root and management VDOMs, send logs to the global syslog server (10.6.30.22).
- The root VDOM sends logs to the following syslog servers:
  - For syslogd, logs are sent to the root VDOM override server at 192.168.5.44 because `use-management-vdom` is disabled.
  - For syslogd2, logs are sent through the management VDOM to the root VDOM override server at 172.16.200.55 and to the syslog server reachable by the management VDOM because `use-management-vdom` is enabled.
  - For syslogd3, logs are sent through the management VDOM to the root VDOM override syslog server at 10.6.30.22 and to the syslog server reachable by the management VDOM because `use-management-vdom` is enabled.
- The management VDOM (vdom1) sends logs to the override syslog server at 172.16.200.55.

To configure syslog servers:

1. Enable the global syslog server:

```
config log syslogd setting
    set status enable
    set server "10.6.30.22"
    set facility local6
end
```

2. For root, configure three override syslog servers:
   a. For syslogd, enable an override syslog server and disable `use-management-vdom`:

```
config log syslogd override-setting
    set status enable
    set server "192.168.5.44"
    set use-management-vdom disable
    set facility local6
end
```

   b. For syslog2, enable an override syslog server and enable `use-management-vdom`:

```
config log syslogd2 override-setting
    set status enable
    set server "172.16.200.55"
    set use-management-vdom enable
    set facility local6
end
```

   c. For syslog3, enable an override syslog server and enable `use-management-vdom`:

```
config log syslogd3 override-setting
    set status enable
    set server "10.6.30.22"
    set use-management-vdom enable
end
```

3. For the management VDOM, configure an override syslog server:

```
config log syslogd override-setting
    set status enable
    set server "172.16.200.55"
    set facility local6
end
```

Customizing log formats for syslog servers NEW

You can configure and send custom log formats to syslog servers using custom-format and log-template configurations, allowing you to tailor syslog output to match the exact requirements of your external logging system.

Use the feature by:

1. Configuring the custom log format.

```
config log custom-format
    edit <name>
        set field-exclusion-list {date | devid | devname | eventtime | level | logid | slot |
subtype | time | type | tz | vd | vsn}
        empty-value-indicator <string>
        config log-templates
            edit <name>
                set category <string>
                set subtypes <string>
                set template <string>
            next
        end
    next
end
```

2. Configuring the syslog logging devices to use the custom log format.

```
config log syslogd setting
    set status enable
    set server <syslog_IP>
    set format {default | csv | cef | rfc5424 | json |custom}
    set custom-log-format <string>
en
```

Custom log formats can also be configured for syslogd2, syslogd3, and syslogd-override under multi-VDOM:

- For syslogd2 and syslogd3 and the `config log syslogd2 setting`/`config log syslogd3 setting` commands, set `format` to `custom`, and select the log format.
- For syslogd override under vdom, enable `syslog-override` under `config log setting`. Then set `format` as `custom`, and select log format under `config log syslogd override-setting`.

This topic includes the following examples:

- Example 1: exclude fields on page 4813
- Example 2: configure specific categories, subtypes, and fields on page 4815
- Example 3: include UTM-related fields in traffic logs on page 4816
- Example 4: use wildcard syntax on page 4816

Example 1: exclude fields

In this example, logs sent from FortiGate to a syslog server are customized as follows:

- The percentage (%) character indicates empty values in the logs.
- The “date”, “eventtime”, “time” and “tz” fields are excluded from the logs.

To customize logs for syslog servers:

1. Set the custom log format:
   The `empty-value-indicator` can be a single character symbol, such as %.

```
config log custom-format
    edit “test”
        set empty-value-indicator  “%”
        set field-exclusion-list  "date" "eventtime" "time" "tz"
    end
end
```

2. Set the log template:

```
config log syslogd setting
    set status enable
    set server "172.16.200.55"
    set  facility local6
    set format custom
    set custom-log-format “test”
end
```

3. Send logs to a syslog server.
   Logs received by the syslog server exclude “date”, “eventtime”, “time”, and “tz” fields.
   - Traffic log example:

     ```
     Feb  3 10:03:31 172.16.200.2 devname=FGT-B-LOG devid=FG201FT920901892  logid="0006000026"
     type="traffic" subtype="http-transaction" level="notice" vd="vdom1" srcip=10.1.100.22
     srcport=33012 dstip=172.16.200.55 dstport=80 sessionid=100663312 transid=100663315
     action="accept" policyid=1 policytype="proxy-policy" poluuid="e66b90f6-008d-51f1-e5f4-
     799c5495850b" url="http://172.16.200.55/" agent="curl/7.68.0" appcat="unscanned"
     duration=0 reqlength=127 resplength=1145 rcvdbyte=1145 sentbyte=127 scheme="http"
     hostname="172.16.200.55" resptype="normal" httpmethod="GET" statuscode="200"
     reqtime=1770134611 resptime=1770134611 respfinishtime=1770134611
     ```

   - UTM log example:

     ```
     Feb  2 15:31:47 172.16.200.2 devname=FGT-B-LOG devid=FG201FT920901892  logid="0211008192"
     type="utm" subtype="virus" eventtype="infected" level="warning" vd="vdom1" policyid=1
     poluuid="2a0da664-90c8-51ef-c42e-31dd5cc56aea" policytype="policy" msg="File is infected."
     action="blocked" service="HTTP" sessionid=1009 srcip=10.1.100.22 dstip=172.16.200.55
     srcport=56586 dstport=80 srccountry="Reserved" dstcountry="Reserved" srcintf="port1"
     srcintfrole="undefined" dstintf="port2" dstintfrole="undefined" srcuuid="85bf1200-90ae-
     51ef-7a00-759244bb35e4" dstuuid="85bf1200-90ae-51ef-7a00-759244bb35e4" proto=6
     direction="incoming" filename="eicar.com" checksum="6851cf3c" quarskip="No-skip"
     virus="EICAR_TEST_FILE" viruscat="Virus" dtype="cached" itype="infected"
     ref="https://fortiguard.com/encyclopedia/virus/2172" virusid=2172
     url="http://172.16.200.55/virus/eicar.com" profile="av-test" agent="curl/7.68.0"
     httpmethod="GET"
     analyticscksum="275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f"
     analyticssubmit="false" crscore=50 craction=2 crlevel="critical"
     ```

   - Event log example:

     ```
     Feb  2 15:31:41 172.16.200.2 devname=FGT-B-LOG devid=FG201FT920901892  logid="0100020025"
     type="event" subtype="system" level="notice" vd="vdom1" logdesc="Report generated
     successfully" msg="Report generation succeeded for layout:default."
     ```

```
file="/var/log/log/vdom1/report/generated/On-Demand-default-2026-02-02-123140/default.pdf"
filesize=272974 datarange="01-Feb-2026 12:00 To 02-Feb-2026 11:59" reporttype="On-demand"
processtime=0"
```

Example 2: configure specific categories, subtypes, and fields

In this example, logs sent from FortiGate to a syslog server are customized as follows:

- The percentage (%) character indicates empty values in the logs.
- A log type category (`event`), subtype (`system`), and specific fields are configured for the logs.

FortiGate only sends logs for this category and subtype with the configured fields to the syslog server. This feature is supported for all categories.

To customize logs for syslog servers:

1. Configure a custom log format:

```
config log custom-format
    edit “test”
        set empty-value-indicator "%"
            config log-templates
                edit "1"
                   set category event
                   set subtypes "system"
                   set template “eve_type=$(subtype) eve_msg=$(msg) eve_vdom=$(vd) eve_
level=$(level) eve_status=$(status) eve_profile=$(profile)”
                next
            end
        next
    end
end
```

2. Set the log template:

```
config log syslogd setting
    set status enable
    set server "172.16.200.55"
    set  facility local6
    set format custom
    set custom-log-format “test”
end
```

3. Send logs to a syslog server.

   Logs received by the syslog server only include logs configured in template, and fields with empty value display %.

```
Feb  2 16:00:55 172.16.200.2 eve_type="system" eve_msg="Administrator admin login failed from
https(10.6.30.254) because of invalid password" eve_vdom="vdom1" eve_level="alert" eve_
status="failed" eve_profile=%
```

```
Feb  2 16:01:12 172.16.200.2 eve_type="system" eve_msg="Edit system.global " eve_vdom="vdom1"
eve_level="information" eve_status=% eve_profile=%
```

Example 3: include UTM-related fields in traffic logs

This example demonstrates how to include UTM-related fields in traffic logs. This feature is only available for traffic logs.

In this example, `category` is set to `traffic`, `subtype` is set to `forward`, and specific fields are configured for the logs FortiGate sends to a syslog server.

When a UTM field does not exist for the category, the `invalid log field name` error is displayed.

To customize logs for syslog servers:

1. Configure a custom log format:

```
config log custom-format
    edit "test"
        set empty-value-indicator ' '
        config log-templates
            edit "1"
                set category traffic
                set subtypes "forward"
                set template "type=$(type) subtype=$(subtype) http_status_code=$(statuscode)
url=$(virus.url) virus=$(virus.virus) agent=$(virus.agent) profile=$(virus.profile)"
            next
        end
    next
end
```

2. Set the log template:

```
config log syslogd setting
    set status enable
    set server "172.16.200.55"
    set  facility local6
    set format custom
    set custom-log-format “test”
end
```

3. Send logs to a syslog server.
   Logs received by the syslog server include selected traffic and UTM log fields.

```
Feb  3 11:44:07 172.16.200.2 type="traffic" subtype="forward" http_status_code=
url="http://172.16.200.55/virus/eicar.com" virus="EICAR_TEST_FILE" agent="curl/7.68.0"
profile="av-test"
```

Example 4: use wildcard syntax

This example demonstrates how to include `WILDCARD_SYNTAX: $(*)` in the template. Note:

- `$(*)` will be replaced by default-format log content.
- You can append additional UTM fields with the `WILDCARD_SYNTAX`.
- `$(*)` behaves like regular field in custom-format.log-templates and can include both prefixes and postfixes.
- Each template can only have one `WILDCARD_SYNTAX $(*)`. An error is displayed when more than one `$(*)` are used in the template.

To customize logs for syslog servers:

1. Configure a custom log format:
   When the custom log format has `exclusion-list` set, and the template is set for a specific category, then the template overrides the `exclusion-list` for the category.

```
config log custom-format
    edit "test"
        set field-exclusion-list "date" "eventtime" "time" "tz"
        set empty-value-indicator "%"
        config log-templates
            edit "1"
                set category traffic
                set subtypes "forward"
                set template "url=$(virus.url) virus=$(virus.virus) http_status_
code=$(statuscode) $(*) agent=$(virus.agent) profile=$(virus.profile)"
            next
        end
    next
end
```

2. Set the log template:

```
config log syslogd setting
    set status enable
    set server "172.16.200.55"
    set  facility local6
    set format custom
    set custom-log-format “test”
end
```

3. Send logs to a syslog server.
   Logs received by the syslog server includes all traffic log fields with selected UTM log fields.

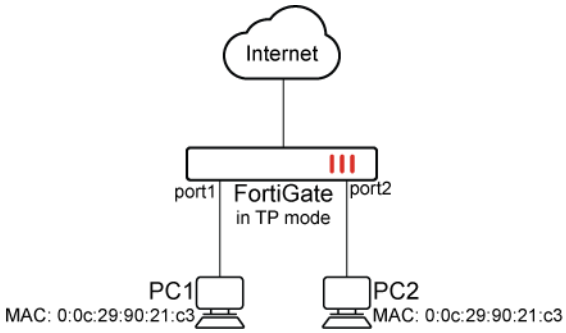
```
Feb  3 09:27:45 172.16.200.2 date=2026-02-03 time=09:27:45 devname=FGT-B-LOG
devid=FG201FT920901892 url="http://172.16.200.55/virus/eicar.com" virus="EICAR_TEST_FILE"
http_status_code=% eventtime=1770139665020444596 tz="-0800" logid="0000000013" type="traffic"
subtype="forward" level="notice" vd="vdom1" srcip=10.1.100.22 srcname="10.1.100.22"
srcport=40028 srcintf="port1" srcintfrole="undefined" dstip=172.16.200.55
dstname="172.16.200.55" dstport=80 dstintf="port2" dstintfrole="undefined"
srccountry="Reserved" dstcountry="Reserved" sessionid=5120 proto=6 action="close" policyid=1
policytype="policy" poluuid="2a0da664-90c8-51ef-c42e-31dd5cc56aea" policyname="B-out"
service="HTTP" trandisp="snat" transip=172.16.200.2 transport=40028 appcat="unscanned"
duration=30 sentbyte=1784 rcvdbyte=1242 sentpkt=31 rcvdpkt=6 utmaction="block" countav=1
crscore=50 craction=2 agent="curl/7.68.0" profile="av-test"
```

Logging MAC address flapping events

FortiOS logs MAC address flapping events when a device's MAC address is learned on different interfaces within the MAC address table in transparent mode. The log provides comprehensive details about the event, such as the specific MAC address involved, the ports where the flapping occurred, and the exact time of the event. This enhancement assists network administrators in quickly identifying and addressing related issues, thereby enhancing network stability and performance.

Example

In this example, the end user initiates internet traffic from PC1, which has an authentic MAC address. Subsequently, the user generates internet traffic from PC2 using a packet manipulation tool, such as Scapy, but with the spoofed MAC address of PC1. This event is successfully identified and logged by FortiGate running in transparent (TP) mode.



To view the logs:

```
# execute log filter category 1
# execute log filter start-line 1
# execute log display

36 logs found.
10 logs returned.
1: date=2024-03-26 time=14:05:33 eventtime=1711487133347757075 tz="-0700" logid="0100022970"
type="event" subtype="system" level="information" vd="vdom1" logdesc="MAC flapping"
service="kernel" mac="00:0c:29:90:21:c3" src_int="port1" msg="The incoming port of MAC address
00:0c:29:90:21:c3 has been switched from port2 to port1"
```

Incorporating endpoint device data in the web filter UTM logs

Endpoint device data, including hostname and MAC address, have been incorporated in the web filter UTM logs. Endpoint device data can be incorporated in the logs using the following:

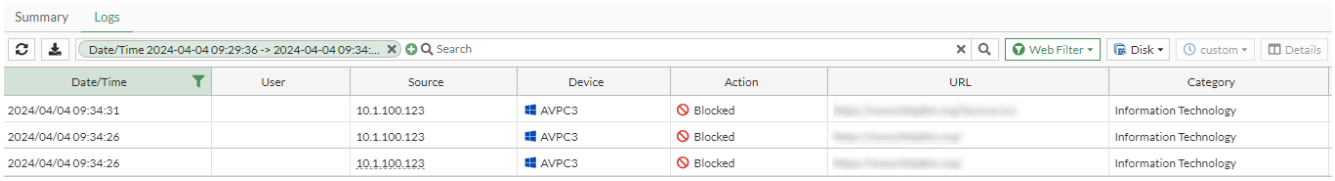
```
config log setting
    set extended-utm-log {enable | disable}
end
```

To incorporate endpoint device data in the web filter UTM logs, ensure a firewall policy with a web filter profile is configured and *Device detection* is configured on the interfaces. *Device detection* can be configured in *Network > Interfaces* and the CLI.

When this command is enabled, the `srcmac` and `srcname` fields are included in the web filter UTM logs:

```
1: date=2024-04-04 time=09:34:31 eventtime=1712248470720798942 tz="-0700" logid="0316013056"
type="utm" subtype="webfilter" eventtype="ftgd_blk" level="warning" vd="vdom1" policyid=1
poluuid="9f550138-ed67-51ee-b593-e4c9c3cd549f" policytype="policy" sessionid=20910
srcip=10.1.100.123 srcport=59705 srccountry="Reserved" srcintf="port2" srcintfrole="undefined"
srcuuid="04df25b6-ed67-51ee-3006-8c2d12813f90" srcmac="00:0c:29:06:7e:5b" srcname="AVPC3"
dstip=52.201.199.27 dstport=443 dstcountry="United States" dstintf="port1" dstintfrole="undefined"
dstuuid="04df25b6-ed67-51ee-3006-8c2d12813f90" proto=6 httpmethod="GET" service="HTTPS"
hostname="www.httpbin.org" agent="Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36
(KH" profile="webfilter" action="blocked" reqtype="referral"
url="https://www.httpbin.org/favicon.ico" referralurl="https://www.httpbin.org/" sentbyte=2088
rcvdbyte=5709 direction="outgoing" msg="URL belongs to a denied category in policy"
ratemethod="domain" cat=52 catdesc="Information Technology"
```

Likewise, the *Device* column is populated with the endpoint hostname information in the *Log & Report > Security Events > Logs* table:



When this command is disabled, the new fields are excluded from the web filter UTM logs and the *Device* column does not display the client hostname information. The command is disabled by default.

Logging detection of duplicate IPv4 addresses

FortiOS can log each detection of duplicate IPv4 addresses on physical interfaces and VLAN interfaces in the event log under the new log ID 32701.

```
config system global
    set ip-conflict-detection {enable | disable}
end
```

Command	Description
`ip-conflict-detection {enable | disable}`	Enable/disable logging of IPv4 address conflict detection.

FortiOS uses the following methods to detect duplicate IPv4 addresses, and can generate a log for each detection:

Detection method	Description
Active	Detection of duplicate physical and VLAN IPv4 addresses on FortiGate is triggered when: • FortiOS starts • The miglogd daemon restarts • A new physical interface is created • A physical interface status changes to up • A physical interface configuration changes • The `diagnose test app miglogd 55` command is run
Passive	Detection of duplicate IPv4 address on a client is triggered when a device connected to FortiGate attempts to use an IPv4 address that is already in use. FortiOS identifies duplicate IPv4 addresses by monitoring the Gratuitous ARP packet. When the source IP in the Gratuitous ARP packet is duplicated in the cache, but the source MAC address is different, then the IP address is considered a duplicate. In addition to physical interfaces, passive detection is also valid for VLAN.

In addition, a packet can be sent to the SNMP host when the SNMP event is set to `interface`.

The `config system snmp community` command includes a new `interface` event:

```
config system snmp community
    edit 1
        set name "test"
        config hosts
            edit 1
                set ip 172.18.71.107 255.255.255.0
            next
        end
        config hosts6
            edit 1
            next
        end
        set events {interface}
    next
end
```

Command	Description
`events {interface}`	Send a trap for interface events.

The following new debug commands are also available:

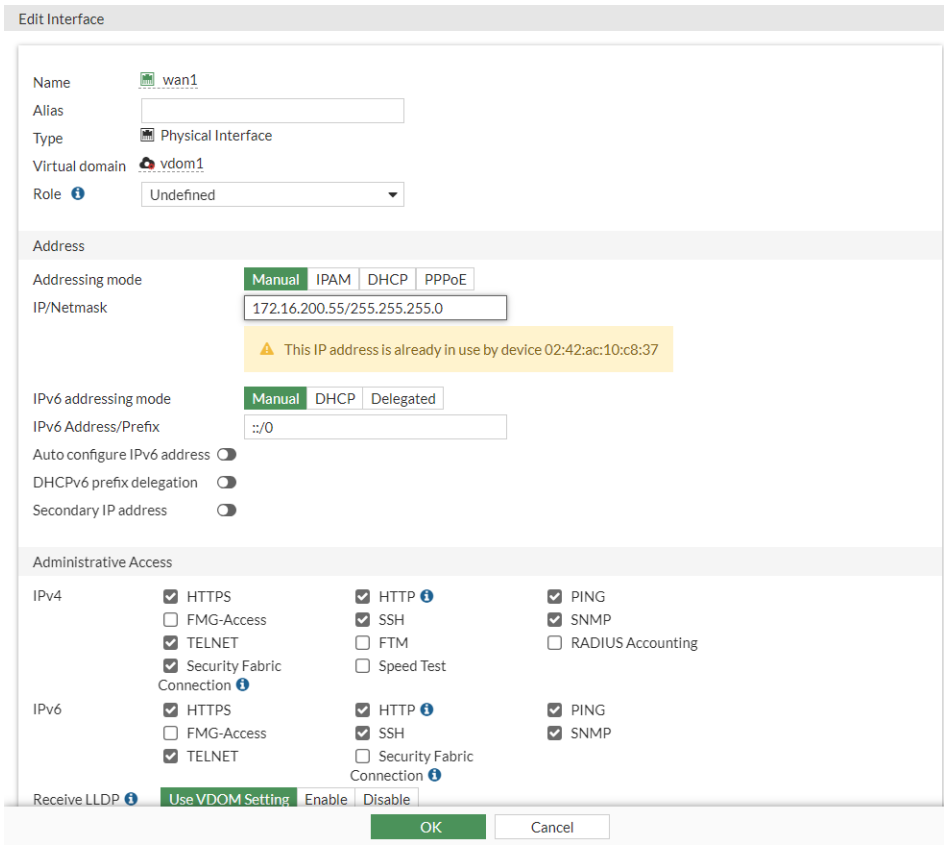
`diagnose test application miglogd 54`	Shows the cache for IPv4 address conflict detection.
`diagnose test application miglogd 55`	Executes an IPv4 address conflict detection.

To enable logging of IPv4 address conflicts:

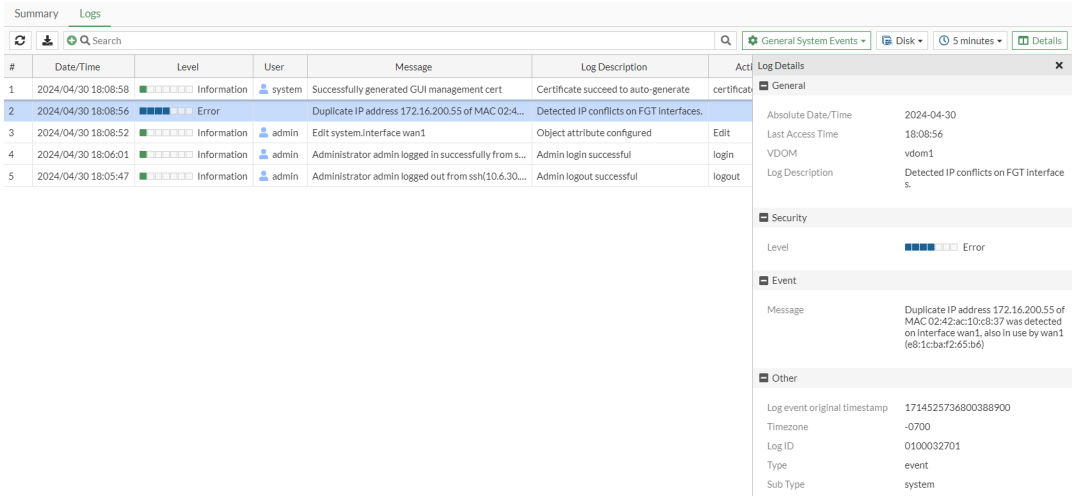
```
config system global
    set ip-conflict-detection enable
end
```

To trigger an active detection and log of IPv4 conflicts:

1. In FortiOS, go to *Network* > *Interfaces*, and double-click an interface, such as *wan1*, to open it for editing.
2. In the *IP/Netmask* box, change the IP address from *172.16.200.2/24* to *172.16.200.55/24*. The following warning is displayed: *This IP address is already in use by device <MAC address>*.



3. Go to *Log & Report* > *System Events*. A *Duplicate IP address* log entry is displayed.

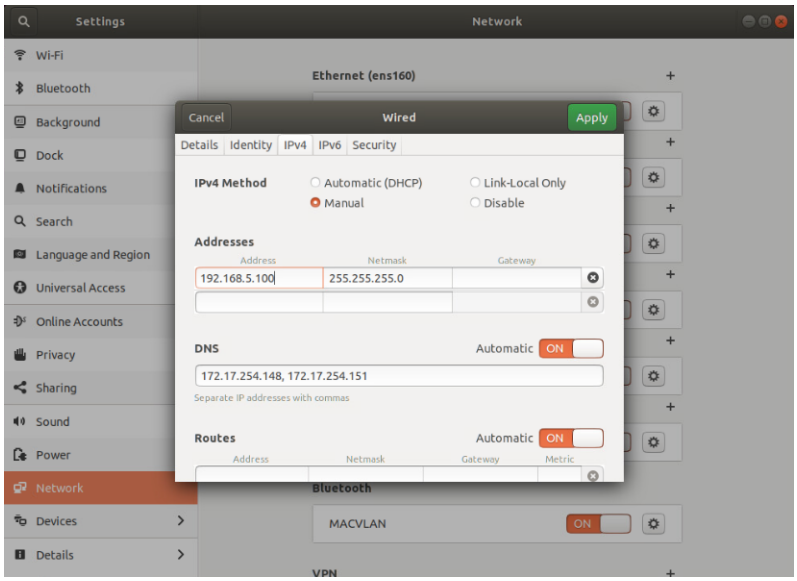


The same information is visible in the raw log

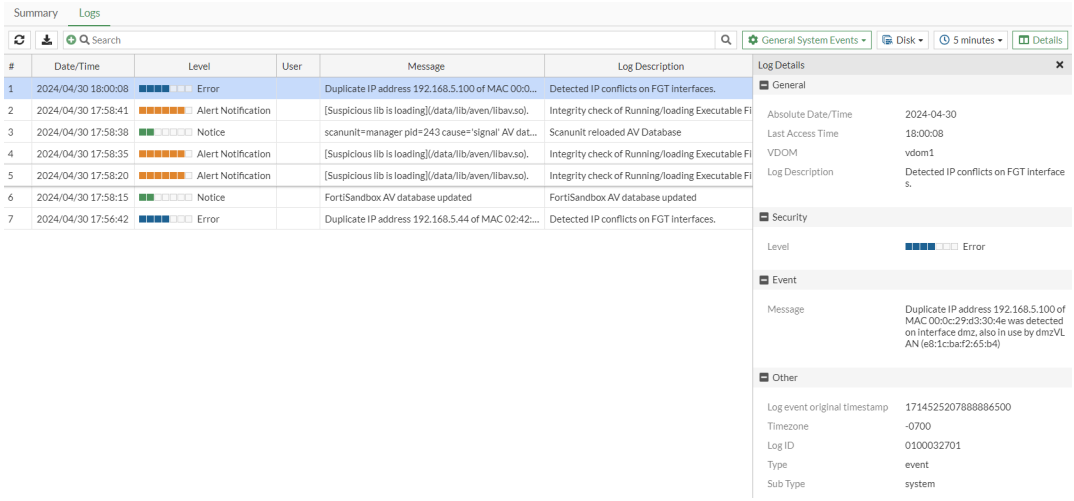
```
date=2024-04-30 time=18:08:56 eventtime=1714525736800388800 tz="-0700" logid="0100032701"
type="event" subtype="system" level="error" vd="vdom1" logdesc="Detected IP conflicts on FGT
interfaces." msg="Duplicate IP address 172.16.200.55 of MAC 02:42:ac:10:c8:37 was detected on
interface wan1, also in use by wan1 (e8:1c:ba:f2:65:b6)"
```

To trigger a passive detection and log of IPv4 conflicts:

1. On a client, change the interface. For example, change the interface from *192.168.5.44* to *192.168.5.100/24*.



2. Go to *Log & Report* > *System Events*. A *Duplicate IP address* log entry is displayed.



The same information is visible in the raw log

```
date=2024-04-30 time=18:00:08 eventtime=1714525207888886460 tz="-0700" logid="0100032701"
type="event" subtype="system" level="error" vd="vdom1" logdesc="Detected IP conflicts on FGT
interfaces." msg="Duplicate IP address 192.168.5.100 of MAC 00:0c:29:d3:30:4e was detected on
interface dmz, also in use by dmzVLAN (e8:1c:ba:f2:65:b4)"
```

To configure SNMP traps for interfaces:

1. Configure an SNMP host and set up an event trap for interface.

```
config system snmp community
    edit 1
        set name "test"
        config hosts
            edit 1
                set ip 172.16.200.55 255.255.255.0
            next
        end
        set events interface
    next
end
```

To show the IPv4 address conflict detection cache:

```
# diagnose test application miglogd 54
index   IPv4 address    MAC     dev     vlanid
50      10.10.100.2     00:00:00:00:00:00       Loopback2
58      192.168.5.44    e8:1c:ba:f2:65:b4       dmzVLAN 50
48      10.255.1.1      00:00:00:00:00:00       fortilink
51      10.1.10.3       00:00:00:00:00:00       Loopback3
8       10.1.100.2      e8:1c:ba:f2:65:b7       wan2
49      10.10.10.2      00:00:00:00:00:00       loopback
6       10.6.30.107     e8:1c:ba:f2:65:b5       mgmt
52      192.168.100.99  e8:1c:ba:f2:65:bb       lan
```

```
23      10.2.2.2        e8:1c:ba:f2:65:c6       x1
7       172.16.200.2    e8:1c:ba:f2:65:b6       wan1
```

To execute an IPv4 address conflict detection:

```
# diagnose test application miglogd 55
Sending probe for 10.10.100.2 via Loopback2.
Sending probe for 192.168.5.44 via dmzVLAN.
Sending probe for 10.255.1.1 via fortilink.
Sending probe for 10.1.10.3 via Loopback3.
Sending probe for 10.1.100.2 via wan2.
Sending probe for 10.10.10.2 via loopback.
Sending probe for 10.6.30.107 via mgmt.
Sending probe for 192.168.100.99 via lan.
Sending probe for 10.2.2.2 via x1.
Sending probe for 172.16.200.2 via wan1.
```

To verify the trap has been sent:

```
# diagnose debug application snmp -1
# diagnose debug enable
snmpd: queue is 2 entries long.
snmpd: queueing trap 8008000000000000@4295121097 (4295121097)
snmpd: queue is 3 entries long.
snmpd: dequeueing trap 8008000000000000@4295121097 (4295121097)
snmpd: sending to hosts: interface(1601)
snmpd: attempting v1 trap: interface(1601)
snmpd: trap from (172.16.200.1 -> 172.16.200.55)
snmpd: trap send(172.16.200.1:162 -> 172.16.200.55:162) bytes sent=264 total=264
snmpd: attempting v2c trap: interface(1601)
snmpd: get     : system.3.0 -> () -> 0
snmpd: trap send(172.16.200.1:162 -> 172.16.200.55:162) bytes sent=288 total=288
```

Including zone information fields in logs

Source and destination zone fields can be enabled for logs to enhance scalability and efficiency in log management. This reduces the need to search logs by individual interfaces within zones.

CLI syntax

The *srczone* and *dstzone* fields can be enabled in logs using the following command:

```
config log setting
    set zone-name {enable | disable}
end
```

Example

The following example demonstrates configuring zone information in a firewall policy and enabling it so that this information is displayed in the logs.

To enable zone information fields in the logs:

1. Enable the inclusion of the zone fields in logs:

```
config log setting
    set zone-name enable
end
```

2. Configure the zones:

```
config system zone
    edit "zone-1"
        set interface "port1" "port15"
    next
    edit "zone-2"
        set interface "port2" "port16"
    next
end
```

3. Apply the zones in a firewall policy:

```
config firewall policy
    edit 1
        set name "out"
        set srcintf "zone-1"
        set dstintf "zone-2"
        set action accept
        set srcaddr "all"
        set dstaddr "all"
        set srcaddr6 "all"
        set dstaddr6 "all"
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set av-profile "av"
        set logtraffic all
        set nat enable
    next
end
```

4. Review the traffic logs:

```
#execute log display
date=2025-05-29 time=11:51:21 eventtime=1748544681417779289 tz="-0700" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="vdom1" srcip=10.1.100.22 srcport=45970
srcintf="port1" srcintfrole="undefined" srczone="zone-1" dstip=172.16.200.44 dstport=22
dstintf="port2" dstintfrole="undefined" dstzone="zone-2" srccountry="Reserved"
```

```
dstcountry="Reserved" sessionid=2754 proto=6 action="close" policyid=1 policytype="policy"
poluuid="e12d3e48-3cb4-51f0-3e20-28d63a9401de" policyname="out" service="SSH" trandisp="snat"
transip=172.16.200.2 transport=45970 appcat="unscanned" duration=4 sentbyte=2621 rcvdbyte=2437
sentpkt=17 rcvdpkt=14

date=2025-05-29 time=11:52:15 eventtime=1748544734697773392 tz="-0700" logid="0000000013"
type="traffic" subtype="forward" level="notice" vd="vdom1" srcip=10.1.100.22 srcport=35680
srcintf="port1" srcintfrole="undefined" srczone="zone-1" dstip=172.16.200.55 dstport=80
dstintf="port2" dstintfrole="undefined" dstzone="zone-2" srccountry="Reserved"
dstcountry="Reserved" sessionid=2756 proto=6 action="close" policyid=1 policytype="policy"
poluuid="e12d3e48-3cb4-51f0-3e20-28d63a9401de" policyname="out" service="HTTP" trandisp="snat"
transip=172.16.200.2 transport=35680 appcat="unscanned" duration=30 sentbyte=1732
rcvdbyte=1972 sentpkt=30 rcvdpkt=8 utmaction="block" countav=1 crscore=50 craction=2
utmref=65528-28
```

5. Review the UTM logs:

```
#execute log display
date=2025-05-29 time=11:51:45 eventtime=1748544704623697175 tz="-0700" logid="0211008192"
type="utm" subtype="virus" eventtype="infected" level="warning" vd="vdom1" policyid=1
poluuid="e12d3e48-3cb4-51f0-3e20-28d63a9401de" policytype="policy" msg="File is infected."
action="blocked" service="HTTP" sessionid=2756 srcip=10.1.100.22 dstip=172.16.200.55
srcport=35680 dstport=80 srccountry="Reserved" dstcountry="Reserved" srcintf="port1"
srcintfrole="undefined" srczone="zone-1" dstintf="port2" dstintfrole="undefined"
dstzone="zone-2" srcuuid="85bf1200-90ae-51ef-7a00-759244bb35e4" dstuuid="85bf1200-90ae-51ef-
7a00-759244bb35e4" proto=6 direction="incoming" filename="eicar.com" checksum="6851cf3c"
quarskip="No-skip" virus="EICAR_TEST_FILE" viruscat="Virus" dtype="av-engine" itype="infected"
ref="https://fortiguard.com/encyclopedia/virus/2172" virusid=2172
url="http://172.16.200.55/virus/eicar.com" profile="av" agent="curl/7.68.0" httpmethod="GET"
analyticscksum="275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f"
analyticssubmit="false" crscore=50 craction=2 crlevel="critical"
```

6. Review the interface cache for zone information:

```
# diagnose test application miglogd 18
Network interface cache (version=6):
        ID=50, name=fortilink, phy=, role=undefined, vfid=0 type=2 detail=0
        ID=4, name=ha, phy=, role=undefined, vfid=0 type=0 detail=0
        ID=40, name=l2t.root, phy=, role=undefined, vfid=0 type=3 detail=0
        ID=44, name=l2t.vdom1, phy=, role=undefined, vfid=1 type=3 detail=0
        ID=48, name=l2t.vdom2, phy=, role=undefined, vfid=2 type=3 detail=0
        ID=3, name=mgmt, phy=, role=lan, vfid=1 type=0 detail=0
        ID=37, name=modem, phy=, role=undefined, vfid=0 type=0 detail=0
        ID=39, name=naf.root, phy=, role=undefined, vfid=0 type=3 detail=0
        ID=43, name=naf.vdom1, phy=, role=undefined, vfid=1 type=3 detail=0
        ID=47, name=naf.vdom2, phy=, role=undefined, vfid=2 type=3 detail=0
        ID=35, name=npu0_vlink0, phy=, role=undefined, vfid=0 type=0 detail=0
        ID=36, name=npu0_vlink1, phy=, role=undefined, vfid=0 type=0 detail=0
        ID=7, name=port1, phy=, role=undefined, vfid=1 type=0 detail=0, zone-name: zone-1,
zone_id:1, zone:zone-1
        ID=8, name=port2, phy=, role=undefined, vfid=1 type=0 detail=0, zone-name: zone-2,
zone_id:2, zone:zone-2
```

```
        ID=9, name=port3, phy=, role=undefined, vfid=0 type=0 detail=0
        ID=10, name=port4, phy=, role=undefined, vfid=0 type=0 detail=0
        ID=11, name=port5, phy=, role=undefined, vfid=2 type=0 detail=0
        ID=12, name=port6, phy=, role=undefined, vfid=0 type=0 detail=0
        ID=13, name=port7, phy=, role=undefined, vfid=0 type=0 detail=0
        ID=14, name=port8, phy=, role=undefined, vfid=0 type=0 detail=0
        ID=15, name=port9, phy=, role=undefined, vfid=0 type=0 detail=0
        ID=16, name=port10, phy=, role=undefined, vfid=0 type=0 detail=0
        ID=17, name=port11, phy=, role=undefined, vfid=0 type=0 detail=0
        ID=18, name=port12, phy=, role=undefined, vfid=0 type=0 detail=0
        ID=19, name=port13, phy=, role=undefined, vfid=0 type=0 detail=0
        ID=20, name=port14, phy=, role=undefined, vfid=0 type=0 detail=0
        ID=21, name=port15, phy=, role=undefined, vfid=1 type=0 detail=0, zone-name: zone-1,
zone_id:1, zone:zone-1
        ID=22, name=port16, phy=, role=undefined, vfid=1 type=0 detail=0, zone-name: zone-2,
zone_id:2, zone:zone-2
```

Local in and local out logging

Traffic generated by the FortiGate (local out) or traffic destined for the FortiGate (local in) is not handled by the same policies as forward traffic (traffic that is intended to pass through the FortiGate).

While forward traffic is handled by the policies found in firewall policies, local in traffic is controlled using interface configuration, such as enabling or disabling Administrative access traffic (HTTPS, PING, SSH, and so on), as well as the local in policy. By default, local in policies exist to allow traffic enabled by interface settings and FortiGate services such as IPsec and central management. See Firewall policy on page 1842 and Local-in policy on page 1886.

Default local in policies do not have a policy ID and their logs will show `policyid=0`. This is not the same as the forward traffic firewall policies, where `policyid=0` is the implicit deny.

User-defined local in policies will include a policy ID.

Local out logs are not filtered by the FortiGate and by default rely on routing table lookups to determine the egress interface that is used to initiate the connection. See Local out traffic on page 1468.

To enable local traffic logging and to review a sample local traffic log, see Traffic Logs > Local Traffic on page 4831.

Logging local traffic per local-in policy

Starting in 7.6.0, local traffic logging can be configured for each local-in policy. This enables more precise and targeted logging by focusing on specific local-in policies that are most relevant to your needs.

Logging can be configured per local-in policy in the *Log & Report > Log Settings* page or by using the following commands:

```
config log setting
    set local-in-policy-log {enable | disable}
```

```
end
config firewall local-in-policy
    edit <id>
        set logtraffic {enable | disable}
    next
end
config firewall local-in-policy6
    edit <id>
        set logtraffic {enable | disable}
    next
end
```

For more information, see Logging local traffic per local-in policy on page 1895.

TACACS+ accounting details in audit logs NEW

TACACS+ accounting logs include CLI change details in the `reason` field.

To view the log details:

1. Configure the FortiGate and TACACS+ accounting server.
2. Verify that audit logging is enabled:

```
config system global
    set cli-audit-log enable
end
config log tacacs+accounting filter
    set login-audit enable
    set config-change-audit enable
    set cli-cmd-audit enable
end
```

3. Log on to the FortiGate using remote TACACS+ authentication.
4. Execute some tasks, such as adding a local user and changing port settings, then compare the information from the FortiGate log files with information collected in a PCAP file and from the external TACACS+accounting server.

   Adding a local user:

   - FortiOS log:

```
# execute log display
14 logs found.
10 logs returned.

1: date=2026-01-16 time=07:04:38 eventtime=1768575878906101594 tz="-0800"
logid="0100032102" type="event" subtype="system" level="alert" vd="vdom1"
logdesc="Configuration changed" user="tac1" ui="https(172.16.200.254)" msg="Configuration
is changed in the admin session"
...
9: date=2026-01-16 time=07:03:01 eventtime=1768575781267058847 tz="-0800"
```

```
logid="0100032132" type="event" subtype="system" level="notice" vd="vdom1" logdesc="Local
user added" user="tac1" ui="https(172.16.200.254)" name="test101" status="enable"
msg="User tac1 added local user test101 from https(172.16.200.254)"
```

- Message extracted from PCAP file:

```
Frame 195: 210 bytes on wire (1680 bits), 210 bytes captured (1680 bits)
Ethernet II, Src: Fortinet_a3:50:f3 (80:80:2c:a3:50:f3), Dst: VMware_e2:b1:02
(00:0c:29:e2:b1:02)
Internet Protocol Version 4, Src: 10.1.100.8, Dst: 10.1.100.142
Transmission Control Protocol, Src Port: 6188, Dst Port: 49, Seq: 1, Ack: 1, Len: 144
TACACS+
    Major version: TACACS+
    Minor version: 0
    Type: Accounting (3)
    Sequence number: 1
    Flags: 0x00 (Encrypted payload, Multiple Connections)
    Session ID: 3656374799
    Packet length: 132
    Encrypted Request
    Decrypted Request
        Flags: 0x04
        Auth Method: NOT_SET (0x00)
        Privilege Level: 0
        Authentication type: Unknown (0)
        Service: Login (1)
        User len: 4
        User: tac1
        Port len: 0
        Remaddr len: 0
        Arg count: 4
        Arg[0] length: 17
        Arg[0] value: service=fortigate
        Arg[1] length: 14
        Arg[1] value: event=sys_acct
        Arg[2] length: 29
        Arg[2] value: stop_time=1768575783514241967
        Arg[3] length: 55
        Arg[3] value: reason="Add user.local test101 type[password]passwd[*]"
```

Changing port settings:

- Message sent to TACACS+ accounting server:

```
Frame 4: 300 bytes on wire (2400 bits), 300 bytes captured (2400 bits)
...
        Arg count: 4
        Arg[0] length: 17
        Arg[0] value: service=fortigate
        Arg[1] length: 14
        Arg[1] value: event=sys_acct
        Arg[2] length: 29
        Arg[2] value: stop_time=1768579536565487819
```

```
        Arg[3] length: 144
        Arg[3] value: reason="Edit system.interface port10 allowaccess[ping https ssh snmp
http telnet radius-acct->ping https ssh snmp http telnet radius-acct scim]"
```

Sample logs by log type

This topic provides a sample raw log for each subtype and the configuration requirements.

Traffic Logs > Forward Traffic

Log configuration requirements

```
config firewall policy
    edit 1
        set srcintf "port12"
        set dstintf "port11"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set logtraffic all
        set application-list "g-default"
        set ssl-ssh-profile "certificate-inspection"
        set nat enable
    next
end
```

Sample log

```
date=2019-05-10 time=11:37:47 logid="0000000013" type="traffic" subtype="forward" level="notice"
vd="vdom1" eventtime=1557513467369913239 srcip=10.1.100.11 srcport=58012 srcintf="port12"
srcintfrole="undefined" dstip=23.59.154.35 dstport=80 dstintf="port11" dstintfrole="undefined"
srcuuid="ae28f494-5735-51e9-f247-d1d2ce663f4b" dstuuid="ae28f494-5735-51e9-f247-d1d2ce663f4b"
poluuid="ccb269e0-5735-51e9-a218-a397dd08b7eb" sessionid=105048 proto=6 action="close" policyid=1
policytype="policy" service="HTTP" dstcountry="Canada" srccountry="Reserved" trandisp="snat"
transip=172.16.200.2 transport=58012 appid=34050 app="HTTP.BROWSER_Firefox" appcat="Web.Client"
apprisk="elevated" applist="g-default" duration=116 sentbyte=1188 rcvdbyte=1224 sentpkt=17
rcvdpkt=16 utmaction="allow" countapp=1 osname="Ubuntu" mastersrcmac="a2:e9:00:ec:40:01"
srcmac="a2:e9:00:ec:40:01" srcserver=0 utmref=65500-742
```

Traffic Logs > Local Traffic

Log configuration requirements

```
config log setting
    set local-in-allow enable
    set local-in-deny-unicast enable
    set local-in-deny-broadcast enable
    set local-out enable
end
```

Sample log

```
date=2019-05-10 time=11:50:48 logid="0001000014" type="traffic" subtype="local" level="notice"
vd="vdom1" eventtime=1557514248379911176 srcip=172.16.200.254 srcport=62024 srcintf="port11"
srcintfrole="undefined" dstip=172.16.200.2 dstport=443 dstintf="vdom1" dstintfrole="undefined"
sessionid=107478 proto=6 action="server-rst" policyid=0 policytype="local-in-policy"
service="HTTPS" dstcountry="Reserved" srccountry="Reserved" trandisp="noop" app="Web Management
(HTTPS)" duration=5 sentbyte=1247 rcvdbyte=1719 sentpkt=5 rcvdpkt=6 appcat="unscanned"
```

Traffic Logs > Multicast Traffic

Log configuration requirements

```
config firewall multicast-policy
    edit 1
        set dstaddr 230-1-0-0
        set dstintf port3
        set srcaddr 172-16-200-0
        set srcintf port25
        set action accept
        set logtraffic all
    next
end
```

```
config system setting
    set multicast-forward enable
end
```

Sample log

```
date=2019-03-31 time=06:42:54 logid="0002000012" type="traffic" subtype="multicast" level="notice"
vd="vdom1" eventtime=1554039772 srcip=172.16.200.55 srcport=60660 srcintf="port25"
srcintfrole="undefined" dstip=230.1.1.2 dstport=7878 dstintf="port3" dstintfrole="undefined"
sessionid=1162 proto=17 action="accept" policyid=1 policytype="multicast-policy"
service="udp/7878" dstcountry="Reserved" srccountry="Reserved" trandisp="noop" duration=22
sentbyte=5940 rcvdbyte=0 sentpkt=11 rcvdpkt=0 appcat="unscanned"
```

Traffic Logs > Sniffer Traffic

Log configuration requirements

```
config firewall sniffer
    edit 3
        set logtraffic all
        set interface "port1"
        set ips-sensor-status enable
        set ips-sensor "sniffer-profile"
    next
end
```

Sample log

```
date=2019-05-10 time=14:18:54 logid="0004000017" type="traffic" subtype="sniffer" level="notice"
vd="root" eventtime=1557523134021045897 srcip=208.91.114.4 srcport=50463 srcintf="port1"
srcintfrole="undefined" dstip=104.80.88.154 dstport=443 dstintf="port1" dstintfrole="undefined"
sessionid=2193276 proto=6 action="accept" policyid=3 policytype="sniffer" service="HTTPS"
dstcountry="United States" srccountry="Canada" trandisp="snat" transip=0.0.0.0 transport=0
duration=10 sentbyte=0 rcvdbyte=0 sentpkt=0 rcvdpkt=0 appcat="unscanned" utmaction="allow"
countips=1 crscore=5 craction=32768 sentdelta=0 rcvddelta=0 utmref=65162-7772
```

```
config system global
    set log-uuid-address enable
end
```

```
config firewall sniffer
    edit 1
        set logtraffic all
        set ipv6 enable
        set interface "port3"
        set ip-threatfeed-status enable
        set ip-threatfeed "g-source"
    next
end
```

Sample log

```
1: date=2021-01-26 time=15:51:37 eventtime=1611705097880421908 tz="-0800" logid="0004000017"
type="traffic" subtype="sniffer" level="notice" vd="vd1" srcip=10.1.100.12 srcport=34604
srcintf="port3" srcintfrole="undefined" dstip=172.16.200.55 dstport=80 dstintf="port3"
dstintfrole="undefined" srcthreatfeed="g-source" srccountry="Reserved" dstcountry="Reserved"
sessionid=30384 proto=6 action="accept" policyid=1 policytype="sniffer" service="HTTP"
trandisp="snat" transip=0.0.0.0 transport=0 duration=0 sentbyte=0 rcvdbyte=0 sentpkt=0 rcvdpkt=0
appcat="unscanned"
```

Event Logs > SD-WAN Events

Log configuration requirements

```
config log eventfilter
    set event enable
    set sdwan enable
end
```

Sample log

```
date=2020-03-29 time=16:41:30 logid="0113022923" type="event" subtype="sdwan" level="notice"
vd="root" eventtime=1585525290513555981 tz="-0700" logdesc="Virtual WAN Link status"
eventtype="Health Check" healthcheck="ping1" slatargetid=1 oldvalue="1" newvalue="2" msg="Number
of pass member changed."
```

```
date=2020-03-29 time=16:51:27 logid="0113022925" type="event" subtype="sdwan" level="notice"
vd="root" eventtime=1585525888177637570 tz="-0700" logdesc="Virtual WAN Link SLA information"
eventtype="SLA" healthcheck="ping1" slatargetid=1 interface="R150" status="up" latency="0.013"
jitter="0.001" packetloss="100.000%" inbandwidth="0kbps" outbandwidth="0kbps" bibandwidth="0kbps"
slamap="0x0" metric="packetloss" msg="Health Check SLA status. SLA failed due to being over the
performance metric threshold."
```

Event Logs > System Events

Log configuration requirements

```
config log eventfilter
    set event enable
    set system enable
end
```

Sample log

```
date=2019-05-13 time=11:20:54 logid="0100032001" type="event" subtype="system" level="information"
vd="vdom1" eventtime=1557771654587081441 logdesc="Admin login successful" sn="1557771654"
user="admin" ui="ssh(172.16.200.254)" method="ssh" srcip=172.16.200.254 dstip=172.16.200.2
action="login" status="success" reason="none" profile="super_admin" msg="Administrator admin
logged in successfully from ssh(172.16.200.254)"
```

Event Logs > Router Events

Log configuration requirements

```
config log eventfilter
    set event enable
```

```
    set router enable
end
```

```
config router bgp
    set log-neighbour-changes enable
end
```

```
config router ospf
    set log-neighbour-changes enable
end
```

Sample log

```
date=2019-05-13 time=14:12:26 logid="0103020301" type="event" subtype="router" level="warning"
vd="root" eventtime=1557781946677737955 logdesc="Routing log" msg="OSPF: RECV[Hello]: From
31.1.1.1 via port9:172.16.200.1: Invalid Area ID 0.0.0.0"
```

Event Logs > VPN Events

Log configuration requirements

```
config log eventfilter
    set event enable
    set vpn enable
end
```

Sample log

```
date=2019-05-13 time=14:21:42 logid="0101037127" type="event" subtype="vpn" level="notice"
vd="root" eventtime=1557782502722231889 logdesc="Progress IPsec phase 1" msg="progress IPsec phase
1" action="negotiate" remip=50.1.1.101 locip=50.1.1.100 remport=500 locport=500 outintf="port14"
cookies="9091f4d4837ea71c/0000000000000000" user="N/A" group="N/A" xauthuser="N/A"
xauthgroup="N/A" assignip=N/A vpntunnel="test" status="success" init="local" mode="main"
dir="outbound" stage=1 role="initiator" result="OK"
```

Event Logs > User Events

Log configuration requirements

```
config log eventfilter
    set event enable
    set user enable
end
```

Sample log

```
date=2019-05-13 time=15:55:56 logid="0102043008" type="event" subtype="user" level="notice"
vd="root" eventtime=1557788156913809277 logdesc="Authentication success" srcip=10.1.100.11
dstip=172.16.200.55 policyid=1 interface="port10" user="bob" group="local-group1"
authproto="TELNET(10.1.100.11)" action="authentication" status="success" reason="N/A" msg="User
bob succeeded in authentication"
```

Event Logs > Endpoint Events

Log configuration requirements

```
config log eventfilter
    set event enable
    set endpoint enable
end
```

Sample log

```
date=2019-05-14 time=08:32:13 logid="0107045057" type="event" subtype="endpoint"
level="information" vd="root" eventtime=1557847933900764210 logdesc="FortiClient connection added"
action="add" status="success" license_limit="unlimited" used_for_type=4 connection_type="sslvpn"
count=1 user="skubas" ip=172.18.64.250 name="VAN-200957-PC"
fctuid="52C66FE08F724FE0B116DAD5062C96CD" msg="Add a FortiClient Connection."
```

```
date=2019-05-14 time=08:19:38 logid="0107045058" type="event" subtype="endpoint"
level="information" vd="root" eventtime=1557847179037488154 logdesc="FortiClient connection
closed" action="close" status="success" license_limit="unlimited" used_for_type=5 connection_
type="sslvpn" count=1 user="skubas" ip=172.18.64.250 name="VAN-200957-PC"
fctuid="52C66FE08F724FE0B116DAD5062C96CD" msg="Close a FortiClient Connection."
```

Event Logs > HA Events

Log configuration requirements

```
config log eventfilter
    set event enable
    set ha enable
end
```

Sample log

```
date=2019-05-10 time=09:53:18 logid="0108037894" type="event" subtype="ha" level="critical"
vd="root" eventtime=1557507199208575235 logdesc="Virtual cluster member joined" msg="Virtual
cluster detected member join" vcluster=1 ha_group=0 sn="FG2K5E3916900286"
```

Event Logs > Security Rating Events

Log configuration requirements

```
config log eventfilter
    set event enable
    set security-rating enable
end
```

Sample log

```
date=2019-05-13 time=14:40:59 logid="0110052000" type="event" subtype="security-rating"
level="notice" vd="root" eventtime=1557783659536252389 logdesc="Security Rating summary"
auditid=1557783648 audittime=1557783659 auditscore="5.0" criticalcount=1 highcount=6 mediumcount=8
lowcount=0 passedcount=38
```

Event Logs > WAN Opt & Cache Events

Log configuration requirements

```
config log eventfilter
    set event enable
    set wan-opt enable
end
```

Sample log

```
date=2019-05-14 time=09:37:46 logid="0105048039" type="event" subtype="wad" level="error"
vd="root" eventtime=1557851867382676560 logdesc="SSL fatal alert sent" session_id=0 policyid=0
srcip=0.0.0.0 srcport=0 dstip=208.91.113.83 dstport=636 action="send" alert="2" desc="certificate
unknown" msg="SSL Alert sent"
```

```
date=2019-05-10 time=15:48:31 logid="0105048038" type="event" subtype="wad" level="error"
vd="root" eventtime=1557528511221374615 logdesc="SSL Fatal Alert received" session_id=5f88ddd1
policyid=0 srcip=172.18.70.15 srcport=59880 dstip=91.189.89.223 dstport=443 action="receive"
alert="2" desc="unknown ca" msg="SSL Alert received"
```

Event Logs > Wireless

Log configuration requirements

```
config log eventfilter
    set event enable
    set wireless-activity enable
end
```

```
config wireless-controller log
    set status enable
end
```

Sample log

```
date=2019-05-13 time=11:30:08 logid="0104043568" type="event" subtype="wireless" level="warning"
vd="vdom1" eventtime=1557772208134721423 logdesc="Fake AP on air" ssid="fortinet"
bssid="90:6c:ac:89:e1:fa" aptype=0 rate=130 radioband="802.11n" channel=6 action="fake-ap-on-air"
manuf="Fortinet, Inc." security="WPA2 Personal" encryption="AES" signal=-93 noise=-95 live=353938
age=505 onwire="no" detectionmethod="N/A" stamac="N/A" apscan="N/A" sndetected="N/A"
radioiddetected=0 stacount=0 snclosest="FP320C3X17001909" radioidclosest=0 apstatus=0 msg="Fake AP
On-air fortinet 90:6c:ac:89:e1:fa chan 6 live 353938 age 505"
```

Event Logs > SDN Connector

Log configuration requirements

```
config log eventfilter
    set event enable
    set connector enable
end
```

Sample log

```
date=2019-05-13 time=16:09:43 logid="0112053200" type="event" subtype="connector"
level="information" vd="root" eventtime=1557788982 logdesc="IP address added" cfgobj="aws1"
action="object-add" addr="54.210.36.196" cldobjid="i-0fe5a1ef16bb94796" netid="vpc-97e81cee"
msg="connector object discovered in addr-obj aws1, 54.210.36.196"
```

```
date=2019-05-13 time=16:09:43 logid="0112053201" type="event" subtype="connector"
level="information" vd="root" eventtime=1557788982 logdesc="IP address removed" cfgobj="aws1"
action="object-remove" addr="172.31.31.101" cldobjid="i-0fe5a1ef16bb94796" netid="vpc-97e81cee"
msg="connector object removed in addr-obj aws1, 172.31.31.101"
```

Event Logs > FortiExtender Events

Log configuration requirements

```
config log eventfilter
    set event enable
    set fortiextender enable
end
```

Sample log

```
date=2019-02-20 time=09:57:22 logid="0111046400" type="event" subtype="fortiextender"
level="notice" vd="root" eventtime=1550685442 logdesc="FortiExtender system activity"
action="FortiExtender Authorized"    msg="ext SN:FX04DN4N16002352 authorized"
```

```
date=2019-02-20 time=09:51:42 logid="0111046401" type="event" subtype="fortiextender"
level="notice" vd="root" eventtime=1550685102 logdesc="FortiExtender controller activity"
sn="FX04DN4N16002352" ip=11.11.11.2 action="ext session-deauthed" msg="ext SN:FX04DN4N16002352
deauthorized"
```

```
date=2019-02-20 time=10:02:26 logid="0111046409" type="event" subtype="fortiextender"
level="information" vd="root" eventtime=1550685746 logdesc="Remote FortiExtender info activity"
sn="FX04DN4N16002352" ip=11.11.11.2 action="Cellular Connected" imei="359376060442770"
imsi="302720502331361" iccid="89302720403038146410" phonenumber="+16045067526" carrier="Rogers"
plan="Rogers-plan" apn="N/A" service="LTE" msg="FX04DN4N16002352 STATE: sim with
imsi:302720502331361 in slot:2 on carrier:Rogers connected"
```

```
date=2019-02-20 time=10:33:57 logid="0111046407" type="event" subtype="fortiextender"
level="warning" vd="root" eventtime=1550687636 logdesc="Remote FortiExtender warning activity"
sn="FX04DN4N16002352" ip=11.11.11.2 action="Cellular Disconnected" imei="359376060442770"
imsi="N/A" iccid="N/A" phonenumber="N/A" carrier="N/A" plan="N/A" apn="N/A" service="LTE"
msg="FX04DN4N16002352 STATE: sim with imsi: in slot:2 on carrier:N/A disconnected"
```

```
date=2019-02-20 time=10:02:24 logid="0111046409" type="event" subtype="fortiextender"
level="information" vd="root" eventtime=1550685744 logdesc="Remote FortiExtender info activity"
sn="FX04DN4N16002352" ip=11.11.11.2 action="Cellular Connecting" imei="359376060442770"
imsi="302720502331361" iccid="89302720403038146410" phonenumber="+16045067526" carrier="Rogers"
plan="Rogers-plan" apn="N/A" service="N/A" msg="FX04DN4N16002352 STATE: sim with
imsi:302720502331361 in slot:2 on carrier:Rogers connecting
```

```
date=2019-02-20 time=10:47:19 logid="0111046407" type="event" subtype="fortiextender"
level="warning" vd="root" eventtime=1550688438 logdesc="Remote FortiExtender warning activity"
sn="FX04DN4N16002352" ip=11.11.11.2 action="SIM Change" imei="N/A" slot=2 msg="FX04DN4N16002352
SIM: SIM2 is inserted"
```

```
date=2019-02-20 time=10:57:50 logid="0111046407" type="event" subtype="fortiextender"
level="warning" vd="root" eventtime=1550689069 logdesc="Remote FortiExtender warning activity"
sn="FX04DN4N16002352" ip=11.11.11.2 action="SIM Change" imei="359376060442770" slot=1
msg="FX04DN4N16002352 SIM: SIM2 is plucked out"
```

```
date=2019-02-20 time=12:02:24 logid="0111046407" type="event" subtype="fortiextender"
level="warning" vd="root" eventtime=1550692942 logdesc="Remote FortiExtender warning activity"
sn="FX04DN4N16002352" ip=11.11.11.2 action="SIM Switch" imei="359376060442770" reason="sim-switch
can't take effect due to unavailability of 2 sim cards" msg="FX04DN4N16002352 SIM: sim-switch
can't take effect due to unavailability of 2 sim cards"
```

```
date=2019-02-19 time=18:08:46 logid="0111046409" type="event" subtype="fortiextender"
level="information" vd="root" eventtime=1550628524 logdesc="Remote FortiExtender info activity"
sn="FX04DN4N16002352" ip=11.11.11.2 action="Cellular Signal Statistics" imei="359376060442770"
imsi="302720502331361" iccid="89302720403038146410" phonenumber="+16045067526" carrier="Rogers"
```

```
plan="Rogers-plan" service="LTE" sinr="7.0 dB" rsrp="-89 dBm" rsrq="-16 dB" signalstrength="92
dBm" rssi="-54" temperature="40 C" apn="N/A" msg="FX04DN4N16002352 INFO:  LTE RSSI=-54dBm,RSRP=-
89dBm,RSRQ=-
16dB,SINR=7.0dB,BAND=B2,CELLID=061C700F,BW=15MHz,RXCH=1025,TXCH=19025,TAC=8AAC,TEMPERATURE=40 C"
```

```
date=2019-02-19 time=18:09:46 logid="0111046409" type="event" subtype="fortiextender"
level="information" vd="root" eventtime=1550628585 logdesc="Remote FortiExtender info activity"
sn="FX04DN4N16002352" ip=11.11.11.2 action="Cellular Data Statistics" imei="359376060442770"
imsi="302720502331361" iccid="89302720403038146410" phonenumber="+16045067526" carrier="Rogers"
plan="Rogers-plan" service="LTE" rcvdbyte=7760 sentbyte=3315 msg="FX04DN4N16002352 INFO: SIM2 LTE,
rx=7760, tx=3315, rx_diff=2538, tx_diff=567"
```

Event Logs > FortiSwitch Events

Log configuration requirements

```
config log eventfilter
    set event enable
    set switch-controller enable
end
```

Sample log

```
date=2020-09-28 time=15:37:02 eventtime=1601332622257714795 tz="-0700" logid="0114032695"
type="event" subtype="switch-controller" level="notice" vd="vdom1" logdesc="FortiSwitch link"
user="Fortilink" sn="S248EPTF18001384" name="S248EPTF18001384" msg="port51 Module re-initialized
to recover from ERROR state."
```

```
date=2020-09-28 time=15:37:02 eventtime=1601332622255619520 tz="-0700" logid="0114032697"
type="event" subtype="switch-controller" level="warning" vd="vdom1" logdesc="FortiSwitch switch"
user="Fortilink" sn="S248EPTF18001384" name="S248EPTF18001384" msg="FortiLink: internal echo reply
timed out"
```

```
date=2020-09-28 time=15:37:01 eventtime=1601332621664809633 tz="-0700" logid="0114032605"
type="event" subtype="switch-controller" level="information" vd="vdom1" logdesc="Switch-Controller
Tunnel Up" user="Switch-Controller" ui="cu_acd" sn="S248EPTF18001384" name="S248EPTF18001384"
msg="CAPWAP Tunnel Up (169.254.1.3)"
```

```
date=2020-09-28 time=15:36:59 eventtime=1601332619501461995 tz="-0700" logid="0114022904"
type="event" subtype="switch-controller" level="notice" vd="vdom1" logdesc="CAPUTP session status
notification" user="Switch-Controller" ui="cu_acd" sn="S248EPTF18001384" name="S248EPTF18001384"
msg="S248EPTF18001384 Connected via session join" action="session-join" srcip=169.254.1.3
```

```
date=2020-09-28 time=15:36:26 eventtime=1601332560434649361 tz="-0700" logid="0114032601"
type="event" subtype="switch-controller" level="information" vd="vdom1" logdesc="Switch-Controller
discovered" user="daemon_admin" ui="cmdbsvr" sn="S524DN4K16000116" name="S524DN4K16000116"
msg="S524DN4K16000116 Discovered"
```

```
date=2020-09-28 time=15:36:26 eventtime=1601332560405228924 tz="-0700" logid="0114032601"
type="event" subtype="switch-controller" level="information" vd="vdom1" logdesc="Switch-Controller
discovered" user="daemon_admin" ui="cmdbsvr" sn="S248EPTF18001827" name="S248EPTF18001827"
msg="S248EPTF18001827 Discovered"
```

```
date=2020-09-28 time=15:36:26 eventtime=1601332560336851635 tz="-0700" logid="0114032601"
type="event" subtype="switch-controller" level="information" vd="vdom1" logdesc="Switch-Controller
discovered" user="daemon_admin" ui="cmdbsvr" sn="S248EPTF18001384" name="S248EPTF18001384"
msg="S248EPTF18001384 Discovered"
```

Event Logs > REST API Events

Log configuration requirements

```
config log setting
    set rest-api-set enable
    set rest-api-get enable
end
```

Sample log

```
date=2022-02-02 time=15:52:09 eventtime=1643845930263415066 tz="-0800" logid="0116047301"
type="event" subtype="rest-api" level="information" vd="root" logdesc="REST API request success"
user="admin" ui="GUI(192.168.1.69)" method="GET" path="system.usb-log" status="200"
url="/api/v2/monitor/system/usb-log?vdom=root"
```

```
date=2022-02-02 time=15:52:06 eventtime=1643845926774931021 tz="-0800" logid="0116047301"
type="event" subtype="rest-api" level="information" vd="root" logdesc="REST API request success"
user="admin" ui="GUI(192.168.1.69)" method="GET" path="license.status" status="200"
url="/api/v2/monitor/license/status?vdom=root"
```

```
date=2022-02-02 time=15:52:06 eventtime=1643845926764579729 tz="-0800" logid="0116047301"
type="event" subtype="rest-api" level="information" vd="root" logdesc="REST API request success"
user="admin" ui="GUI(192.168.1.69)" method="GET" path="log.fortianalyzer.setting" status="200"
url="/api/v2/cmdb/log.fortianalyzer/setting?vdom=root"
```

```
date=2022-02-02 time=15:52:06 eventtime=1643845926762372766 tz="-0800" logid="0116047301"
type="event" subtype="rest-api" level="information" vd="root" logdesc="REST API request success"
user="admin" ui="GUI(192.168.1.69)" method="GET" path="system.sandbox" action="connection"
status="200" url="/api/v2/monitor/system/sandbox/connection?vdom=root"
```

```
date=2022-02-02 time=15:52:06 eventtime=1643845926755869998 tz="-0800" logid="0116047301"
type="event" subtype="rest-api" level="information" vd="root" logdesc="REST API request success"
user="admin" ui="GUI(192.168.1.69)" method="GET" path="system.firmware" status="200"
url="/api/v2/monitor/system/firmware?vdom=root"
```

Event Logs > IOC Detection

Log configuration requirements

```
config log setting
    set local-out enable
    set local-out-ioc-detection enable
end
```

Sample log

```
date=2021-12-20 time=16:43:54 eventtime=1640047434839814226 tz="-0800" logid="0100020214"
type="event" subtype="system" level="warning" vd="root" logdesc="Locally generated traffic goes to
IoC location" srcip=172.16.200.2 srcport=18047 dstip=223.205.1.54 dstport=514 session_id=23563
proto=6
```

```
# Corresponding Traffic Log #
date=2021-12-20 time=16:45:18 eventtime=1640047518959313316 tz="-0800" logid="0001000014"
type="traffic" subtype="local" level="notice" vd="root" srcip=172.16.200.2 srcport=18047
srcintf="unknown-0" srcintfrole="undefined" dstip=223.205.1.54 dstport=514 dstintf="port2"
dstintfrole="undefined" srccountry="Reserved" dstcountry="Thailand" sessionid=23632 proto=6
action="timeout" policyid=0 service="tcp/514" trandisp="noop" app="tcp/514" duration=17
sentbyte=240 rcvdbyte=0 sentpkt=4 rcvdpkt=0 appcat="unscanned" dsthwvendor="Fortinet"
masterdstmac="e8:1c:ba:c2:86:63" dstmac="e8:1c:ba:c2:86:63" dstserver=0
```

Security Logs > Antivirus

Log configuration requirements

```
config antivirus profile
    edit "test-av"
        config http
            set av-scan block
        end
        set av-virus-log enable
        set av-block-log enable
    next
end
```

```
config firewall policy
    edit 1
        set srcintf "port12"
        set dstintf "port11"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
```

```
        set utm-status enable
        set av-profile "test-av"
        set logtraffic utm
        set nat enable
    next
end
```

Sample log

```
date=2019-05-13 time=11:45:03 logid="0211008192" type="utm" subtype="virus" eventtype="infected"
level="warning" vd="vdom1" eventtime=1557773103767393505 msg="File is infected." action="blocked"
service="HTTP" sessionid=359260 srcip=10.1.100.11 dstip=172.16.200.55 srcport=60446 dstport=80
srcintf="port12" srcintfrole="undefined" dstintf="port11" dstintfrole="undefined" policyid=4
proto=6 direction="incoming" filename="eicar.com" quarskip="File-was-not-quarantined."
virus="EICAR_TEST_FILE" dtype="Virus" ref="http://www.fortinet.com/ve?vn=EICAR_TEST_FILE"
virusid=2172 url="http://172.16.200.55/virus/eicar.com" profile="g-default" agent="curl/7.47.0"
analyticscksum="275a021bbfb6489e54d471899f7db9d1663fc695ec2fe2a2c4538aabf651fd0f"
analyticssubmit="false" crscore=50 craction=2 crlevel="critical"
```

```
# Corresponding Traffic Log #
date=2019-05-13 time=11:45:04 logid="0000000013" type="traffic" subtype="forward" level="notice"
vd="vdom1" eventtime=1557773104815101919 srcip=10.1.100.11 srcport=60446 srcintf="port12"
srcintfrole="undefined" dstip=172.16.200.55 dstport=80 dstintf="port11" dstintfrole="undefined"
srcuuid="48420c8a-5c88-51e9-0424-a37f9e74621e" dstuuid="187d6f46-5c86-51e9-70a0-fadcfc349c3e"
poluuid="3888b41a-5c88-51e9-cb32-1c32c66b4edf" sessionid=359260 proto=6 action="close" policyid=4
policytype="policy" service="HTTP" dstcountry="Reserved" srccountry="Reserved" trandisp="snat"
transip=172.16.200.2 transport=60446 appid=15893 app="HTTP.BROWSER" appcat="Web.Client"
apprisk="medium" applist="g-default" duration=1 sentbyte=412 rcvdbyte=2286 sentpkt=6 rcvdpkt=6
wanin=313 wanout=92 lanin=92 lanout=92 utmaction="block" countav=1 countapp=1 crscore=50
craction=2 osname="Ubuntu" mastersrcmac="a2:e9:00:ec:40:01" srcmac="a2:e9:00:ec:40:01" srcserver=0
utmref=65497-770
```

Security Logs > Web Filter

Log configuration requirements

```
config webfilter profile
    edit "test-webfilter"
        set web-content-log enable
        set web-filter-activex-log enable
        set web-filter-command-block-log enable
        set web-filter-cookie-log enable
        set web-filter-applet-log enable
        set web-filter-jscript-log enable
        set web-filter-js-log enable
        set web-filter-vbs-log enable
        set web-filter-unknown-log enable
        set web-filter-referer-log enable
        set web-filter-cookie-removal-log enable
```

```
        set web-url-log enable
        set web-invalid-domain-log enable
        set web-ftgd-err-log enable
        set web-ftgd-quota-usage enable
    next
end
```

```
config firewall policy
    edit 1
        set name "v4-out"
        set srcintf "port12"
        set dstintf "port11"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set logtraffic utm
        set utm-status enable
        set webfilter-profile "test-webfilter"
        set nat enable
    next
end
```

Sample log

```
date=2019-05-13 time=16:29:45 logid="0316013056" type="utm" subtype="webfilter" eventtype="ftgd_
blk" level="warning" vd="vdom1" eventtime=1557790184975119738 policyid=1 sessionid=381780
srcip=10.1.100.11 srcport=44258 srcintf="port12" srcintfrole="undefined" dstip=185.244.31.158
dstport=80 dstintf="port11" dstintfrole="undefined" proto=6 service="HTTP"
hostname="morrishittu.ddns.net" profile="test-webfilter" action="blocked" reqtype="direct" url="/"
sentbyte=84 rcvdbyte=0 direction="outgoing" msg="URL belongs to a denied category in policy"
method="domain" cat=26 catdesc="Malicious Websites" crscore=30 craction=4194304 crlevel="high"
```

```
# Corresponding traffic log #
date=2019-05-13 time=16:29:50 logid="0000000013" type="traffic" subtype="forward" level="notice"
vd="vdom1" eventtime=1557790190452146185 srcip=10.1.100.11 srcport=44258 srcintf="port12"
srcintfrole="undefined" dstip=185.244.31.158 dstport=80 dstintf="port11" dstintfrole="undefined"
srcuuid="ae28f494-5735-51e9-f247-d1d2ce663f4b" dstuuid="ae28f494-5735-51e9-f247-d1d2ce663f4b"
poluuid="ccb269e0-5735-51e9-a218-a397dd08b7eb" sessionid=381780 proto=6 action="close" policyid=1
policytype="policy" service="HTTP" dstcountry="Germany" srccountry="Reserved" trandisp="snat"
transip=172.16.200.2 transport=44258 duration=5 sentbyte=736 rcvdbyte=3138 sentpkt=14 rcvdpkt=5
appcat="unscanned" utmaction="block" countweb=1 crscore=30 craction=4194304 osname="Ubuntu"
mastersrcmac="a2:e9:00:ec:40:01" srcmac="a2:e9:00:ec:40:01" srcserver=0 utmref=65497-796
```

Security Logs > DNS Query

Log configuration requirements

```
config dnsfilter profile
    edit "dnsfilter_fgd"
        config ftgd-dns
            set options error-allow
        end
        set log-all-domain enable
        set block-botnet enable
    next
end
```

```
config firewall policy
    edit 1
        set srcintf "port12"
        set dstintf "port11"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set dnsfilter-profile "dnsfilter_fgd"
        set logtraffic utm
        set nat enable
    next
end
```

Sample log

```
date=2019-05-15 time=15:05:49 logid="1501054802" type="utm" subtype="dns" eventtype="dns-response"
level="notice" vd="vdom1" eventtime=1557957949740931155 policyid=1 sessionid=6887
srcip=10.1.100.22 srcport=50002 srcintf="port12" srcintfrole="undefined" dstip=172.16.100.100
dstport=53 dstintf="port11" dstintfrole="undefined" proto=17 profile="dnsfilter_fgd"
srcmac="a2:e9:00:ec:40:41" xid=57945 qname="changelogs.ubuntu.com" qtype="AAAA" qtypeval=28
qclass="IN" ipaddr="2001:67c:1560:8008::11" msg="Domain is monitored" action="pass" cat=52
catdesc="Information Technology"
```

```
date=2019-05-15 time=15:05:49 logid="1500054000" type="utm" subtype="dns" eventtype="dns-query"
level="information" vd="vdom1" eventtime=1557957949653103543 policyid=1 sessionid=6887
srcip=10.1.100.22 srcport=50002 srcintf="port12" srcintfrole="undefined" dstip=172.16.100.100
dstport=53 dstintf="port11" dstintfrole="undefined" proto=17 profile="dnsfilter_fgd"
srcmac="a2:e9:00:ec:40:41" xid=57945 qname="changelogs.ubuntu.com" qtype="AAAA" qtypeval=28
qclass="IN"
```

```
# Corresponding traffic log #
date=2019-05-15 time=15:08:49 logid="0000000013" type="traffic" subtype="forward" level="notice"
vd="vdom1" eventtime=1557958129950003945 srcip=10.1.100.22 srcport=50002 srcintf="port12"
```

```
srcintfrole="undefined" dstip=172.16.100.100 dstport=53 dstintf="port11" dstintfrole="undefined"
srcuuid="ae28f494-5735-51e9-f247-d1d2ce663f4b" dstuuid="ae28f494-5735-51e9-f247-d1d2ce663f4b"
poluuid="ccb269e0-5735-51e9-a218-a397dd08b7eb" sessionid=6887 proto=17 action="accept" policyid=1
policytype="policy" service="DNS" dstcountry="Reserved" srccountry="Reserved" trandisp="snat"
transip=172.16.200.2 transport=50002 duration=180 sentbyte=67 rcvdbyte=207 sentpkt=1 rcvdpkt=1
appcat="unscanned" utmaction="allow" countdns=1 osname="Linux" mastersrcmac="a2:e9:00:ec:40:41"
srcmac="a2:e9:00:ec:40:41" srcserver=0 utmref=65495-306
```

Security Logs > Application Control

Log configuration requirements

```
# log enabled by default in application profile entry

config application list
    edit "block-social.media"
        set other-application-log enable
        config entries
            edit 1
                set category 2 5 6 23
                set log enable
            next
        end
    next
end
```

```
config firewall policy
    edit 1
        set name "to_Internet"
        set srcintf "port10"
        set dstintf "port9"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set logtraffic utm
        set application-list "block-social.media"
        set ssl-ssh-profile "deep-inspection"
        set nat enable
    next
end
```

Sample log

```
date=2019-05-15 time=18:03:36 logid="1059028704" type="utm" subtype="app-ctrl" eventtype="app-
ctrl-all" level="information" vd="root" eventtime=1557968615 appid=40568 srcip=10.1.100.22
dstip=195.8.215.136 srcport=50798 dstport=443 srcintf="port10" srcintfrole="lan" dstintf="port9"
```

```
dstintfrole="wan" proto=6 service="HTTPS" direction="outgoing" policyid=1 sessionid=4414
applist="block-social.media" appcat="Web.Client" app="HTTPS.BROWSER" action="pass"
hostname="www.dailymotion.com" incidentserialno=1962906680 url="/" msg="Web.Client:
HTTPS.BROWSER," apprisk="medium" scertcname="*.dailymotion.com" scertissuer="DigiCert SHA2 High
Assurance Server CA"
```

```
date=2019-05-15 time=18:03:35 logid="1059028705" type="utm" subtype="app-ctrl" eventtype="app-
ctrl-all" level="warning" vd="root" eventtime=1557968615 appid=16072 srcip=10.1.100.22
dstip=195.8.215.136 srcport=50798 dstport=443 srcintf="port10" srcintfrole="lan" dstintf="port9"
dstintfrole="wan" proto=6 service="HTTPS" direction="incoming" policyid=1 sessionid=4414
applist="block-social.media" appcat="Video/Audio" app="Dailymotion" action="block"
hostname="www.dailymotion.com" incidentserialno=1962906682 url="/" msg="Video/Audio: Dailymotion,"
apprisk="elevated"
```

```
date=2019-05-15 time=18:03:35 logid="1059028705" type="utm" subtype="app-ctrl" eventtype="app-
ctrl-all" level="warning" vd="root" eventtime=1557968615 appid=16072 srcip=10.1.100.22
dstip=195.8.215.136 srcport=50798 dstport=443 srcintf="port10" srcintfrole="lan" dstintf="port9"
dstintfrole="wan" proto=6 service="HTTPS" direction="incoming" policyid=1 sessionid=4414
applist="block-social.media" appcat="Video/Audio" app="Dailymotion" action="block"
hostname="www.dailymotion.com" incidentserialno=1962906681 url="/" msg="Video/Audio: Dailymotion,"
apprisk="elevated"
```

```
# Corresponding Traffic Log #  date=2019-05-15 time=18:03:41 logid="0000000013" type="traffic"
subtype="forward" level="notice" vd="root" eventtime=1557968619 srcip=10.1.100.22 srcport=50798
srcintf="port10" srcintfrole="lan" dstip=195.8.215.136 dstport=443 dstintf="port9"
dstintfrole="wan" poluuid="d8ce7a90-7763-51e9-e2be-741294c96f31" sessionid=4414 proto=6
action="client-rst" policyid=1 policytype="policy" service="HTTPS" dstcountry="France"
srccountry="Reserved" trandisp="snat" transip=172.16.200.10 transport=50798 appid=16072
app="Dailymotion" appcat="Video/Audio" apprisk="elevated" applist="block-social.media"
appact="drop-session" duration=5 sentbyte=1150 rcvdbyte=7039 sentpkt=13 utmaction="block"
countapp=3 devtype="Unknown" devcategory="None" mastersrcmac="00:0c:29:51:38:5e"
srcmac="00:0c:29:51:38:5e" srcserver=0 utmref=0-330
```

Security Logs > Intrusion Prevention

Log configuration requirements

```
# log enabled by default in ips sensor

config ips sensor
    edit "block-critical-ips"
        config entries
            edit 1
                set severity critical
                set status enable
                set action block
                set log enable
            next
        end
```

```
   next
end
```

```
config firewall policy
    edit 1
        set name "to_Internet"
        set srcintf "port10"
        set dstintf "port9"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set logtraffic utm
        set ips-sensor "block-critical-ips"
        set nat enable
    next
end
```

Sample log

```
date=2019-05-15 time=17:56:41 logid="0419016384" type="utm" subtype="ips" eventtype="signature"
level="alert" vd="root" eventtime=1557968201 severity="critical" srcip=10.1.100.22
srccountry="Reserved" dstip=172.16.200.55 srcintf="port10" srcintfrole="lan" dstintf="port9"
dstintfrole="wan" sessionid=4017 action="dropped" proto=6 service="HTTP" policyid=1
attack="Adobe.Flash.newfunction.Handling.Code.Execution" srcport=46810 dstport=80
hostname="172.16.200.55" url="/ips/sig1.pdf" direction="incoming" attackid=23305 profile="block-
critical-ips" ref="http://www.fortinet.com/ids/VID23305" incidentserialno=582633933
msg="applications3: Adobe.Flash.newfunction.Handling.Code.Execution," crscore=50 craction=4096
crlevel="critical"
```

```
# Corresponding Traffic Log #  date=2019-05-15 time=17:58:10 logid="0000000013" type="traffic"
subtype="forward" level="notice" vd="root" eventtime=1557968289 srcip=10.1.100.22 srcport=46810
srcintf="port10" srcintfrole="lan" dstip=172.16.200.55 dstport=80 dstintf="port9"
dstintfrole="wan" poluuid="d8ce7a90-7763-51e9-e2be-741294c96f31" sessionid=4017 proto=6
action="close" policyid=1 policytype="policy" service="HTTP" dstcountry="Reserved"
srccountry="Reserved" trandisp="snat" transip=172.16.200.10 transport=46810 duration=89
sentbyte=565 rcvdbyte=9112 sentpkt=9 rcvdpkt=8 appcat="unscanned" utmaction="block" countips=1
crscore=50 craction=4096 devtype="Unknown" devcategory="None" mastersrcmac="00:0c:29:51:38:5e"
srcmac="00:0c:29:51:38:5e" srcserver=0 utmref=0-302
```

Security Logs > Anomaly

Log configuration requirements

```
config firewall DoS-policy
    edit 1
        set interface "port12"
```

```
        set srcaddr "all"
        set dstaddr "all"
        set service "ALL"
        config anomaly
            edit "icmp_flood"
                set status enable
                set log enable
                set action block
                set threshold 50
            next
        end
    next
end
```

Sample log

```
date=2019-05-13 time=17:05:59 logid="0720018433" type="utm" subtype="anomaly" eventtype="anomaly"
level="alert" vd="vdom1" eventtime=1557792359461869329 severity="critical" srcip=10.1.100.11
srccountry="Reserved" dstip=172.16.200.55 srcintf="port12" srcintfrole="undefined" sessionid=0
action="clear_session" proto=1 service="PING" count=1 attack="icmp_flood" icmpid="0x1474"
icmptype="0x08" icmpcode="0x00" attackid=16777316 policyid=1 policytype="DoS-policy"
ref="http://www.fortinet.com/ids/VID16777316" msg="anomaly: icmp_flood, 51 > threshold 50"
crscore=50 craction=4096 crlevel="critical"
```

Security Logs > Data Loss Prevention

Log configuration requirements

```
config dlp profile
    edit "dlp-file-type-test"
        set comment ''
        set replacemsg-group ''
        config filter
            edit 1
                set name ''
                set severity medium
                set type file
                set proto http-get http-post ftp
                set filter-by file-type
                set file-type 1
                set archive enable
                set action block
            next
        end
        set dlp-log enable
    next
end
```

```
config firewall policy
    edit 1
        set name "to_Internet"
        set srcintf "port10"
        set dstintf "port9"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set logtraffic utm
        set dlp-profile "dlp-file-type-test"
        set ssl-ssh-profile "deep-inspection"
        set nat enable
    next
end
```

Sample log

```
date=2022-02-03 time=17:45:30 eventtime=1643938572487964255 tz="-0800" logid="0954024576"
type="utm" subtype="dlp" eventtype="dlp" level="warning" vd="root" filteridx=1 dlpextra="dlp-file-
size11" filtertype="file-type" filtercat="file" severity="medium" policyid=1 policytype="policy"
sessionid=156237 epoch=300501327 eventid=0 srcip=10.1.100.22 srcport=50354 srccountry="Reserved"
srcintf="port10" srcintfrole="lan" dstip=52.216.177.83 dstport=443 dstcountry="United States"
dstintf="port9" dstintfrole="wan" proto=6 service="HTTPS" filetype="pdf" direction="incoming"
action="block" hostname="fortinetweb.s3.amazonaws.com"
url="https://172.16.200.88/dlp/files/fortiauto.pdf" forwardedfor="192.168.0.99"
agent="curl/7.56.0" httpmethod="GET" referralurl="https://example.com/referer.html"
filename="fortiauto.pdf" filesize=285442 profile="dlp-file-type-test" rawdata="x-forwarded-
for=192.168.0.99|Response-Content-Type=application/pdf"
```

```
# Corresponding Traffic Log #
date=2022-02-03 time=17:45:34 logid="0000000013" type="traffic" subtype="forward" level="notice"
vd="root" eventtime=1557967534 srcip=10.1.100.22 srcport=50354 srcintf="port10" srcintfrole="lan"
dstip=52.216.177.83 dstport=443 dstintf="port9" dstintfrole="wan" poluuid="d8ce7a90-7763-51e9-
e2be-741294c96f31" sessionid=3423 proto=6 action="server-rst" policyid=1 policytype="policy"
service="HTTPS" dstcountry="United States" srccountry="Reserved" trandisp="snat"
transip=172.16.200.10 transport=50354 duration=5 sentbyte=2314 rcvdbyte=5266 sentpkt=33 rcvdpkt=12
appcat="unscanned" wanin=43936 wanout=710 lanin=753 lanout=753 utmaction="block" countdlp=1
crscore=5 craction=262144 crlevel="low" devtype="Unknown" devcategory="None"
mastersrcmac="00:0c:29:51:38:5e" srcmac="00:0c:29:51:38:5e" srcserver=0 utmref=0-152
```

Security Logs > SSH and Security Logs > SSL

Log configuration requirements

```
config ssh-filter profile
    edit "ssh-deepscan"
        set block shell
        set log shell
        set default-command-log disable
    next
end
```

```
config firewall policy
    edit 1
        set srcintf "port21"
        set dstintf "port23"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set ssh-filter-profile "ssh-deepscan"
        set profile-protocol-options "protocol"
        set ssl-ssh-profile "ssl"
        set nat enable
    next
end
```

For SSL-Traffic-log, enable `logtraffic all`

```
config firewall policy
    edit 1
        set srcintf "dmz"
        set dstintf "wan1"
        set srcaddr "all"
        set dstaddr "all"
        set action accept
        set schedule "always"
        set service "ALL"
        set utm-status enable
        set inspection-mode proxy
        set logtraffic all
        set ssl-ssh-profile "deep-inspection"
        set nat enable
    next
end
```

For SSL-UTM-log

```
#EVENTTYPE="SSL-ANOMALIES"
```

By default, `ssl-anomaly-log` is enabled.

```
config firewall ssl-ssh-profile
    edit "deep-inspection"
        set comment "Read-only deep inspection profile."
        set server-cert-mode re-sign
        set caname "Fortinet_CA_SSL"
        set untrusted-caname "Fortinet_CA_Untrusted"
        set ssl-anomaly-log enable
        set ssl-exemption-log disable
        set ssl-negotiation-log disable
        set rpc-over-https disable
        set mapi-over-https disable
        set use-ssl-server disable
    next
end
```

```
# EVENTTYPE="SSL-EXEMPT"
```

Enable `ssl-exemption-log` to generate ssl-utm-exempt log.

```
config firewall ssl-ssh-profile
    edit "deep-inspection"
        set comment "Read-only deep inspection profile."
        set server-cert-mode re-sign
        set caname "Fortinet_CA_SSL"
        set untrusted-caname "Fortinet_CA_Untrusted"
        set ssl-anomaly-log enable
        set ssl-exemption-log enable
        set ssl-negotiation-log disable
        set rpc-over-https disable
        set mapi-over-https disable
        set use-ssl-server disable
    next
end
```

```
# EVENTTYPE="SSL-negotiation"
```

Enable `ssl-negotiation-log` to log SSL negotiation. Enable `ssl-server-cert-log` to log server certificate information. Enable `ssl-handshake-log` to log TLS handshakes.

```
config firewall ssl-ssh-profile
    edit "deep-inspection"
        set comment "Read-only deep inspection profile."
        set server-cert-mode re-sign
        set caname "Fortinet_CA_SSL"
        set untrusted-caname "Fortinet_CA_Untrusted"
        set ssl-anomaly-log enable
```

```
        set ssl-exemption-log enable
        set ssl-negotiation-log enable
        set rpc-over-https disable
        set mapi-over-https disable
        set use-ssl-server disable
        set ssl-server-cert-log enable
        set ssl-handshake-log enable
    next
end
```

Sample log for SSH

```
date=2019-05-15 time=16:18:17 logid="1601061010" type="utm" subtype="ssh" eventtype="ssh-channel"
level="warning" vd="vdom1" eventtime=1557962296 policyid=1 sessionid=344 profile="ssh-deepscan"
srcip=10.1.100.11 srcport=43580 dstip=172.16.200.44 dstport=22 srcintf="port21"
srcintfrole="undefined" dstintf="port23" dstintfrole="undefined" proto=6 action="blocked"
direction="outgoing" login="root" channeltype="shell"
```

```
# Corresponding Traffic Log #
date=2019-05-15 time=16:18:18 logid="0000000013" type="traffic" subtype="forward" level="notice"
vd="vdom1" eventtime=1557962298 srcip=10.1.100.11 srcport=43580 srcintf="port21"
srcintfrole="undefined" dstip=172.16.200.44 dstport=22 dstintf="port23" dstintfrole="undefined"
poluuid="49871fae-7371-51e9-17b4-43c7ff119195" sessionid=344 proto=6 action="close" policyid=1
policytype="policy" service="SSH" dstcountry="Reserved" srccountry="Reserved" trandisp="snat"
transip=172.16.200.171 transport=43580 duration=8 sentbyte=3093 rcvdbyte=2973 sentpkt=18
rcvdpkt=16 appcat="unscanned" utmaction="block" countssh=1 utmref=65535-0
```

Sample log for SSL

For SSL-Traffic-log

```
date=2019-05-16 time=10:08:26 logid="0000000013" type="traffic" subtype="forward" level="notice"
vd="root" eventtime=1558026506763925658 srcip=10.1.100.66 srcport=38572 srcintf="dmz"
srcintfrole="dmz" dstip=104.154.89.105 dstport=443 dstintf="wan1" dstintfrole="wan"
poluuid="a17c0a38-75c6-51e9-4c0d-d547347b63e5" sessionid=100 proto=6 action="server-rst"
policyid=1 policytype="policy" service="HTTPS" dstcountry="United States" srccountry="Reserved"
trandisp="snat" transip=172.16.200.11 transport=38572 duration=5 sentbyte=930 rcvdbyte=6832
sentpkt=11 rcvdpkt=19 appcat="unscanned" wanin=1779 wanout=350 lanin=754 lanout=754
utmaction="block"  countssl=1 crscore=5 craction=262144 crlevel="low" utmref=65467-0
```

For SSL-UTM-log

```
#EVENTTYPE="SSL-ANOMALIES"
```

```
date=2019-03-28 time=10:44:53 logid="1700062002" type="utm" subtype="ssl" eventtype="ssl-
anomalies" level="warning" vd="vdom1" eventtime=1553795092 policyid=1 sessionid=10796
service="HTTPS" srcip=10.1.100.66 srcport=43602 dstip=104.154.89.105 dstport=443 srcintf="port2"
srcintfrole="undefined" dstintf="port3" dstintfrole="undefined" proto=6 action="blocked"
msg="Server certificate blocked" reason="block-cert-invalid"
```

```
date=2019-03-28 time=10:51:17 logid="1700062002" type="utm" subtype="ssl" eventtype="ssl-
anomalies" level="warning" vd="vdom1" eventtime=1553795476 policyid=1 sessionid=11110
service="HTTPS" srcip=10.1.100.66 srcport=49076 dstip=172.16.200.99 dstport=443 srcintf="port2"
srcintfrole="undefined" dstintf="port3" dstintfrole="undefined" proto=6 action="blocked"
msg="Server certificate blocked" reason="block-cert-untrusted"
```

```
date=2019-03-28 time=10:55:43 logid="1700062002" type="utm" subtype="ssl" eventtype="ssl-
anomalies" level="warning" vd="vdom1" eventtime=1553795742 policyid=1 sessionid=11334
service="HTTPS" srcip=10.1.100.66 srcport=49082 dstip=172.16.200.99 dstport=443 srcintf="port2"
srcintfrole="undefined" dstintf="port3" dstintfrole="undefined" proto=6 action="blocked"
msg="Server certificate blocked" reason="block-cert-req"
```

```
date=2019-03-28 time=10:57:42 logid="1700062053" type="utm" subtype="ssl" eventtype="ssl-
anomalies" level="warning" vd="vdom1" eventtime=1553795861 policyid=1 sessionid=11424
service="SMTPS" profile="block-unsupported-ssl" srcip=10.1.100.66 srcport=41296
dstip=172.16.200.99 dstport=8080 srcintf="port2" srcintfrole="undefined" dstintf=unknown-0
dstintfrole="undefined" proto=6 action="blocked" msg="Connection is blocked due to unsupported SSL
traffic" reason="malformed input"
```

```
date=2019-03-28 time=11:00:17 logid="1700062002" type="utm" subtype="ssl" eventtype="ssl-
anomalies" level="warning" vd="vdom1" eventtime=1553796016 policyid=1 sessionid=11554
service="HTTPS" srcip=10.1.100.66 srcport=49088 dstip=172.16.200.99 dstport=443 srcintf="port2"
srcintfrole="undefined" dstintf="port3" dstintfrole="undefined" proto=6 action="blocked"
msg="Server certificate blocked" reason="block-cert-sni-mismatch"
```

```
# EVENTTYPE="SSL-EXEMPT"
```

```
date=2019-03-28 time=11:09:14 logid="1701062003" type="utm" subtype="ssl" eventtype="ssl-exempt"
level="notice" vd="vdom1" eventtime=1553796553 policyid=1 sessionid=12079 service="HTTPS"
srcip=10.1.100.66 srcport=49102 dstip=172.16.200.99 dstport=443 srcintf="port2"
srcintfrole="undefined" dstintf="port3" dstintfrole="undefined" proto=6 action="exempt" msg="SSL
connection exempted" reason="exempt-addr"
```

```
date=2019-03-28 time=11:10:55 logid="1701062003" type="utm" subtype="ssl" eventtype="ssl-exempt"
level="notice" vd="vdom1" eventtime=1553796654 policyid=1 sessionid=12171 service="HTTPS"
srcip=10.1.100.66 srcport=47390 dstip=50.18.221.132 dstport=443 srcintf="port2"
srcintfrole="undefined" dstintf="port3" dstintfrole="undefined" proto=6 action="exempt" msg="SSL
connection exempted" reason="exempt-ftgd-cat"
```

```
# EVENTYPE="SSL-NEGOTIATION"
```

```
date=2020-02-07 time=11:10:58 logid="1702062101" type="utm" subtype="ssl" eventtype="ssl-
negotiation" level="warning" vd="vdom1" eventtime=1581102658589415731 tz="-0800" action="blocked"
policyid=1 sessionid=141224 service="HTTPS" profile="deep-inspection-clone" srcip=10.1.100.66
srcport=33666 dstip=172.16.200.99 dstport=8080 srcintf="port2" srcintfrole="undefined"
dstintf="port3" dstintfrole="undefined" proto=6 eventsubtype="unexpected-protocol" msg="SSL
connection is blocked."
```

```
date=2021-06-17 time=16:55:26 eventtime=1623974126384215772 tz="-0700" logid="1702062103"
type="utm" subtype="ssl" eventtype="ssl-negotiation" level="information" vd="vdom1" action="info"
policyid=1 sessionid=6361 service="HTTPS" profile="deep-inspection-clone" srcip=10.1.100.11
```

```
srcport=48892 dstip=18.140.21.233 dstport=443 srcintf="port2" srcintfrole="undefined"
dstintf="port3" dstintfrole="undefined" srcuuid="8666f70e-cfb9-51eb-4991-9012417d69da"
dstuuid="8666f70e-cfb9-51eb-4991-9012417d69da" proto=6 sni="www.fortinet.com"
eventsubtype="server-cert-info" hostname="www.fortinet.com" notbefore="2021-03-13T00:00:00Z"
notafter="2022-04-13T23:59:59Z" issuer="DigiCert TLS RSA SHA256 2020 CA1" cn="*.fortinet.com"
san="*.fortinet.com;www.fortinet.com;fortinet.com" sn="000aa00a00000a00000a00a00aa000a0"
ski="df9152b605cc18b346efb34de6907275dbdb2b3c" certhash="1d55cd34a1ed5d3f69bd825a45e04fbd2efba937"
keyalgo="rsa" keysize=2048
```

```
date=2021-06-17 time=16:55:26 eventtime=1623974126411127210 tz="-0700" logid="1702062103"
type="utm" subtype="ssl" eventtype="ssl-negotiation" level="information" vd="vdom1" action="info"
policyid=1 sessionid=6361 service="HTTPS" profile="deep-inspection-clone" srcip=10.1.100.11
srcport=48892 dstip=18.140.21.233 dstport=443 srcintf="port2" srcintfrole="undefined"
dstintf="port3" dstintfrole="undefined" srcuuid="8666f70e-cfb9-51eb-4991-9012417d69da"
dstuuid="8666f70e-cfb9-51eb-4991-9012417d69da" proto=6 tlsver="tls1.3" sni="www.fortinet.com"
cipher="0x1302" authalgo="rsa" kxproto="ecdhe" kxcurve="secp256r1" eventsubtype="handshake-done"
hostname="www.fortinet.com" handshake="full" mitm="yes"
```

Troubleshooting

The following topics provide information about troubleshooting logging and reporting:

- Log-related diagnostic commands on page 4854
- Backing up log files or dumping log messages on page 4860
- SNMP OID for logs that failed to send on page 4862

Log-related diagnostic commands

This topic contains examples of commonly used log-related diagnostic commands. Local logging is handled by the locallogd daemon, and remote logging is handled by the fgtlogd daemon.

Log search debugging

The `diagnose debug application miglogd 0x1000` command is used is to show log filter strings used by the log search backend. It also shows which log files are searched.

To run log search debugging:

```
# diagnose debug application miglogd 0x1000
Debug messages will be on for 28 minutes.
# diagnose debug enable

# Files to be searched:
file_no=65422, start line=0, end_line=805
```

```
file_no=65423, start line=0, end_line=221
session ID=2, total logs=1028
back ground search. process ID=2913, session_id=2
 start line=1 view line=10 pre-fetch-pages=2

back ground search. next log file roll number is: 65422
ID=2, total=1028, checked=806, found=806
on-demand back ground search exit. process ID=2913, session_id=2, status=process_on-demand_pending
```

Log filtering

The `execute log filter` command can be used to define and display specific log messages based on the parameters entered.

To display all login system event logs:

```
# execute log filter device disk
# execute log filter category event
# execute log filter field action login
# execute log display

Files to be searched:
file_no=65523, start line=0, end_line=237
file_no=65524, start line=0, end_line=429
file_no=65525, start line=0, end_line=411
file_no=65526, start line=0, end_line=381
file_no=65527, start line=0, end_line=395
file_no=65528, start line=0, end_line=458
file_no=65529, start line=0, end_line=604
file_no=65530, start line=0, end_line=389
file_no=65531, start line=0, end_line=384
session ID=1, total logs=3697
back ground search. process ID=26240, session_id=1
 start line=1  view line=10
( action "login" )
ID=1, total=3697, checked=238, found=5
ID=1, total=3697, checked=668, found=13
ID=1, total=3697, checked=1080, found=23
ID=1, total=3697, checked=1462, found=23
ID=1, total=3697, checked=1858, found=23
ID=1, total=3697, checked=2317, found=54
ID=1, total=3697, checked=2922, found=106
ID=1, total=3697, checked=3312, found=111
ID=1, total=3697, checked=3697, found=114
```

Checking the FortiGate to FortiAnalyzer connection

To check the FortiGate to FortiAnalyzer connection status:

```
# diagnose test application fgtlogd 1
faz: global , enabled
        server=172.16.200.251, realtime=3, ssl=1, state=connected
        server_log_status=Log is allowed.,
        src=, mgmt_name=FGh_Log_vdom1_172.16.200.251, reliable=0, sni_prefix_type=none,
        required_entitlement=none, region=ca-west-1,
        logsync_enabled:1, logsync_conn_id:65535, seq_no:0
                status: ver=6, used_disk=0, total_disk=0, global=0, vfid=0 conn_verified=Y
                SNs: last sn update:56 seconds ago.
                        Sn list:
                        (FAZ-VMTM2200****,age=56s)
                queue: qlen=0.
filter: severity=6, sz_exclude_list=0
         traffic virus webfilter ips emailfilter anomaly voip dlp app-ctrl waf dns ssh ssl file-
filter icap sctp-f
ilter
subcategory:
        traffic: forward local multicast sniffer ztna
        virus:all subcategories are enabled.
        webfilter:all subcategories are enabled.
        ips:all subcategories are enabled.
        emailfilter:all subcategories are enabled.
        anomaly:all subcategories are enabled.
        voip:all subcategories are enabled.
        dlp:all subcategories are enabled.
        app-ctrl:all subcategories are enabled.
        waf:all subcategories are enabled.
        dns:all subcategories are enabled.
        ssh:all subcategories are enabled.
        ssl:all subcategories are enabled.
        file-filter:all subcategories are enabled.
        icap:all subcategories are enabled.
        sctp-filter:all subcategories are enabled.

        server: global, id=0, ready=1, name=172.16.200.251 addr=172.16.200.251:514
        oftp-state=connected
```

To collect debug information when FortiAnalyzer is enabled:

```
# diagnose debug application fgtlogd 0x100

# <2026> __fgtlog_parse_featset()-1680: No featset data in login packet,init the device with
default value
<2026> __on_connect()-1620: oftp is ready.
<2026> __on_connect()-1621: status connected for global-faz.
<2026> _check_oftp_certificate()-206: checking sn:FAZVMSTM2200**** vs cert sn:FAZVMSTM2200****
<2026> _check_oftp_certificate()-208: Verified the certificate of peer (10.100.88.2) to match
```

```
sn=FAZVMSTM2200****
<2026> _faz_post_connection()-249: Certificate verification:enabled, Faz verified:1
<2026> _send_queue_item()-549: Disconnect global-faz until receiving disk usage response.
<2026> _send_queue_item()-555: type=0, cat=0, logcount=0, len=0
<2026> __on_pkt_recv()-1590: dev=global-faz type=252 pkt_len=1099
<2026> __on_pkt_recv()-1590: opt=204, opt_len=91
<2026> __on_pkt_recv()-1590: opt=252, opt_len=996
<2026> _process_hainfo_response()-1206: hainfo opt code=204
<2026> _faz_process_oftp_resp_hainfo_json()-447: ha mode: standalone
<2026> __is_sn_known()-315: MATCHED: idx:0 sn:FAZVMSTM2200****
<2026> _faz_process_oftp_resp_hainfo_json()-481: Received SN:FAZVMSTM2200**** should update:0
<2026> _process_hainfo_response()-1206: hainfo opt code=252
<2026> _faz_process_oftp_resp_hainfo_struct()-553: ha nmember:1 nvcluster:0 mode:1
<2026> __is_sn_known()-315: MATCHED: idx:0 sn:FAZVMSTM2200****
<2026> _faz_process_oftp_resp_hainfo_struct()-559: Received SN:FAZVMSTM2200**** should update:0
<2026> __on_pkt_recv()-1590: dev=global-faz type=1 pkt_len=1356
<2026> __on_pkt_recv()-1590: opt=12, opt_len=16
<2026> __on_pkt_recv()-1590: opt=51, opt_len=9
...
<2026> _build_ack()-867: global-faz ready to send data.
<2026> _process_response()-1152: checking opt code=81
<2026> _process_response()-1152: checking opt code=81
<2026> _process_response()-1152: checking opt code=81
...
<2026> _send_queue_item()-555: type=1, cat=0, logcount=0, len=0
<2026> _send_queue_item()-555: type=7, cat=0, logcount=0, len=58
<2026> _send_queue_item()-555: type=3, cat=10, logcount=1, len=790
<2026> _send_queue_item()-555: type=3, cat=10, logcount=1, len=807
<2026> __on_pkt_recv()-1590: dev=global-faz type=60 pkt_len=474
...
<2026> __on_pkt_recv()-1590: opt=80, opt_len=16
<2026> __on_pkt_recv()-1590: opt=7, opt_len=446
<2026> __on_pkt_recv()-1590: dev=global-faz type=11 pkt_len=37
...
<2026> _send_queue_item()-555: type=3, cat=0, logcount=1, len=1037
<2026> _send_queue_item()-555: type=3, cat=0, logcount=1, len=1033
```

To check the FortiGate to FortiGate Cloud connection status:

```
# diagnose test application fgtlogd 20
Home log server:
    Address: 173.243.132.57:514
Alternative log server:
    Address: 173.243.132.121:514
FazCloud log server:
    Address:
    oftp status: connected
Debug zone info:
    Server IP:          173.243.132.57
    Server port:        514
    Server status:      up
    Server log status:  enabled
```

```
    Log quota:          500000000MB
    Log used:           599MB
    Daily volume:       1000000MB
    FDS arch pause:     0
    fams archive pause: 0
```

locallogd diagnostics

To check real-time log statistics by log type since the locallogd daemon start:

```
# diagnose test application locallogd 3
info for vdom: root
memory
traffic: logs=18289 len=15921725, Sun=0 Mon=18289 Tue=0 Wed=0 Thu=0 Fri=0 Sat=0
event: logs=286 len=115729, Sun=0 Mon=286 Tue=0 Wed=0 Thu=0 Fri=0 Sat=0
app-ctrl: logs=10018 len=7051278, Sun=0 Mon=10018 Tue=0 Wed=0 Thu=0 Fri=0 Sat=0

disk
traffic: logs=18289 len=15921725, Sun=0 Mon=18289 Tue=0 Wed=0 Thu=0 Fri=0 Sat=0 compressed=1620003
event: logs=280 len=112390, Sun=0 Mon=280 Tue=0 Wed=0 Thu=0 Fri=0 Sat=0 compressed=13157
app-ctrl: logs=10018 len=7051278, Sun=0 Mon=10018 Tue=0 Wed=0 Thu=0 Fri=0 Sat=0 compressed=836906
```

fgtlogd diagnostics

To check real-time log statistics by log type since the fgtlogd daemon start:

```
# diagnose test application fgtlogd 3
info for vdom: root
faz
traffic: logs=11763 len=6528820, Sun=2698 Mon=3738 Tue=0 Wed=0 Thu=0 Fri=2523 Sat=2804
compressed=1851354
event: logs=2190 len=891772, Sun=500 Mon=400 Tue=0 Wed=0 Thu=0 Fri=786 Sat=504 compressed=713129
app-ctrl: logs=1 len=692, Sun=0 Mon=0 Tue=0 Wed=0 Thu=0 Fri=1 Sat=0 compressed=384

faz-cloud
traffic: logs=11763 len=6528820, Sun=2698 Mon=3738 Tue=0 Wed=0 Thu=0 Fri=2523 Sat=2804
event: logs=2190 len=891772, Sun=500 Mon=400 Tue=0 Wed=0 Thu=0 Fri=786 Sat=504
app-ctrl: logs=1 len=692, Sun=0 Mon=0 Tue=0 Wed=0 Thu=0 Fri=1 Sat=0
```

To check the remote queue and see the maximum buffered memory size:

```
# diagnose test application fgtlogd 41
cache maximum: 19569745(18MB) objects: 0 used: 0(0MB) allocated: 0(0MB)
VDOM:root
Memory queue for: global-faz
        queue:
                num:0 size:0(0MB) total size:0(0MB) max:19569745(18MB) logs:0
Confirm queue for: global-faz
```

```
        queue:
                num:0 size:0(0MB) total size:0(0MB) max:19569745(18MB) logs:0
Memory queue for: fds
        queue:
                num:0 size:0(0MB) total size:0(0MB) max:19569745(18MB) logs:0
Confirm queue for: fds
        queue:
                num:0 size:0(0MB) total size:0(0MB) max:19569745(18MB) logs:0
```

miglogd diagnostics

The miglogd daemon includes a publisher/subscriber framework that separates functions into different daemons. The miglogd daemon is responsible for building and publishing logs, while device-related details are managed by subscriber daemons.

To enable debugging the miglogd (log daemon) at the proper debug level:

```
# diagnose debug application miglogd <integer>
# diagnose debug enable
```

To display the status or statistics at the proper debug level:

```
# diagnose test application miglogd <integer>
# diagnose debug enable
```

When using the preceding commands, press `Enter` after `diagnose debug application miglogd` or `diagnose test application miglogd` to view the list of available levels.

To check log statistics to the local/remote log device since the miglogd daemon start:

```
# diagnose test application miglogd 6
mem=4288, disk=4070, alert=0, alarm=0, sys=5513, faz=4307, webt=0, fds=0
interface-missed=208
```

To check the miglogd daemon number:

```
# diagnose test application miglogd 15
Main miglogd: ID=0, children=2, active-children=2
        ID=1, duration=70465.
        ID=2, duration=70465.
```

To increase one miglogd child:

```
# diagnose test application miglogd 13
# diagnose test application miglogd 15
Main miglogd: ID=0, children=3, active-children=3
```

```
        ID=1, duration=70486.
        ID=2, duration=70486.
        ID=3, duration=1.
```

To decrease one miglogd child:

```
# diagnose test application miglogd 14
# diagnose test application miglogd 15
Main miglogd: ID=0, children=2, active-children=2
        ID=1, duration=70604.
        ID=2, duration=70604.
```

Backing up log files or dumping log messages

When a log issue is caused by a particular log message, it is very help to get logs from that FortiGate. This topic provides steps for using `execute log backup` or dumping log messages to a USB drive.

Backing up full logs using execute log backup

This command backs up all disk log files and is only available on FortiGates with an SSD disk.

Before running `execute log backup`, we recommend temporarily stopping `miglogd` and `reportd`.

To stop and kill miglogd and reportd:

```
diagnose sys process daemon-auto-restart disable miglogd
diagnose sys process daemon-auto-restart disable reportd
```

Or

1. Determine the process, or thread, ID (PID) of `miglogd` and `reportd`:

   ```
   # diagnose sys top 10 99
   ```

2. Kill each process:

   ```
   # diagnose sys kill 9 <PID>
   ```

To store the log file on a USB drive:

1. Plug in a USB drive into the FortiGate.
2. Run this command:

   ```
   execute log backup local /usb/log.tar
   ```

To restart miglogd and reportd:

```
diagnose sys process daemon-auto-restart enable miglogd
diagnose sys process daemon-auto-restart enable reportd
```

Dumping log messages

To dump log messages:

1. Enable log dumping for `miglogd` daemon:

```
(global) # diagnose test application miglogd 26 1
miglogd(1) log dumping is enabled
```

2. Display all `miglogd` dumping status:

```
global) # diagnose test application miglogd 26 0 255
miglogd(0) log dumping is disabled
miglogd(1) log dumping is enabled
miglogd(2) log dumping is disabled
```

```
(global) # diagnose test application miglogd 26 2
miglogd(2) log dumping is enabled

(global) # diagnose test application miglogd 26 0
miglogd(0) log dumping is enabled

(global) # diagnose test application miglogd 26 0 255
miglogd(0) log dumping is enabled
miglogd(1) log dumping is enabled
miglogd(2) log dumping is enabled
```

3. Let the FortiGate run and collect log messages.
4. List the log dump files:

```
(global) # diagnose test application miglogd 33
2019-04-17 15:50:02          20828      log-1-0.dat
2019-04-17 15:48:31           4892      log-2-0.dat
```

5. Back up log dump files to the USB drive:

```
(global) # diagnose test application miglogd 34

Dumping file miglog1_index0.dat copied to USB disk OK.

Dumping file miglog2_index0.dat copied to USB disk OK.
```

6. Disable log dumping for `miglogd` daemon:

```
(global) # diagnose test application miglogd 26 0
miglogd(0) log dumping is disabled
```

```
(global) # diagnose test application miglogd 26 1
miglogd(1) log dumping is disabled

(global) # diagnose test application miglogd 26 2
miglogd(2) log dumping is disabled

(global) # diagnose test application miglogd 26 0 255
miglogd(0) log dumping is disabled
miglogd(1) log dumping is disabled
miglogd(2) log dumping is disabled
```

SNMP OID for logs that failed to send

When a syslog server encounters low-performance conditions and slows down to respond, the buffered syslog messages in the kernel might overflow after a certain number of retransmissions, causing the overflowed messages to be lost. OIDs track the lost messages or failed logs.

SNMP query OIDs include log statistics for global log devices:

- FORTINET-FORTIGATE-MIB:fortinet.fnFortiGateMib.fgLog.fgLogDeviceNumber 1.3.6.1.4.1.12356.101.21.1.1
- FORTINET-FORTIGATE-MIB:fortinet.fnFortiGateMib.fgLog.fgLogDevices.fgLogDeviceTable.fgLogDeviceEntry.fgLogDeviceEntryIndex 1.3.6.1.4.1.12356.101.21.2.1.1.1
- FORTINET-FORTIGATE-MIB:fortinet.fnFortiGateMib.fgLog.fgLogDevices.fgLogDeviceTable.fgLogDeviceEntry.fgLogDeviceEnabled 1.3.6.1.4.1.12356.101.21.2.1.1.2
- FORTINET-FORTIGATE-MIB:fortinet.fnFortiGateMib.fgLog.fgLogDevices.fgLogDeviceTable.fgLogDeviceEntry.fgLogDeviceName 1.3.6.1.4.1.12356.101.21.2.1.1.3
- FORTINET-FORTIGATE-MIB:fortinet.fnFortiGateMib.fgLog.fgLogDevices.fgLogDeviceTable.fgLogDeviceEntry.fgLogDeviceSentCount 1.3.6.1.4.1.12356.101.21.2.1.1.4
- FORTINET-FORTIGATE-MIB:fortinet.fnFortiGateMib.fgLog.fgLogDevices.fgLogDeviceTable.fgLogDeviceEntry.fgLogDeviceRelayedCount 1.3.6.1.4.1.12356.101.21.2.1.1.5
- FORTINET-FORTIGATE-MIB:fortinet.fnFortiGateMib.fgLog.fgLogDevices.fgLogDeviceTable.fgLogDeviceEntry.fgLogDeviceCachedCount 1.3.6.1.4.1.12356.101.21.2.1.1.6
- FORTINET-FORTIGATE-MIB:fortinet.fnFortiGateMib.fgLog.fgLogDevices.fgLogDeviceTable.fgLogDeviceEntry.fgLogDeviceFailedCount 1.3.6.1.4.1.12356.101.21.2.1.1.7
- FORTINET-FORTIGATE-MIB:fortinet.fnFortiGateMib.fgLog.fgLogDevices.fgLogDeviceTable.fgLogDeviceEntry.fgLogDeviceDroppedCount 1.3.6.1.4.1.12356.101.21.2.1.1.8

Where:

- fgLogDeviceNumber is the number of devices in the table.
- fgLogDeviceEnabled is either 1 or 0, indicating whether the device is enabled.

- fgLogDeviceName is the name of the device.

A FortiGate connected to a syslog server or FortiAnalyzer generates statistics that can be seen using the `diagnose test application syslogd` command:

```
(global) # diagnose test application syslogd 4
syslog=437, nulldev=0, webtrends=0, localout_ioc=258, alarms=0
global log dev statistics:
syslog 0: sent=222, failed=0, cached=0, dropped=0
syslog 1: sent=215, failed=0, cached=0, dropped=0
syslog 2: sent=95, failed=0, cached=0, dropped=0
```

The same statistics are also available in `snmpwalk/snmpget` on the OID 1.3.6.1.4.1.12356.101.21.

```
snmpwalk -v2c -c REGR-SYS 172.16.200.1 1.3.6.1.4.1.12356.101.21
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.1.1.0 = INTEGER: 9
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.1.0 = INTEGER: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.1.1 = INTEGER: 1
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.1.2 = INTEGER: 2
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.1.3 = INTEGER: 3
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.1.4 = INTEGER: 4
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.1.5 = INTEGER: 5
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.1.6 = INTEGER: 6
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.1.7 = INTEGER: 7
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.1.8 = INTEGER: 8
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.2.0 = INTEGER: 1
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.2.1 = INTEGER: 1
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.2.2 = INTEGER: 1
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.2.3 = INTEGER: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.2.4 = INTEGER: 1
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.2.5 = INTEGER: 1
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.2.6 = INTEGER: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.2.7 = INTEGER: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.2.8 = INTEGER: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.3.0 = STRING: "syslog"
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.3.1 = STRING: "syslog2"
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.3.2 = STRING: "syslog3"
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.3.3 = STRING: "syslog4"
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.3.4 = STRING: "faz"
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.3.5 = STRING: "faz2"
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.3.6 = STRING: "faz3"
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.3.7 = STRING: "webtrends"
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.3.8 = STRING: "fds"
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.4.0 = Counter32: 254
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.4.1 = Counter32: 220
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.4.2 = Counter32: 95
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.4.3 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.4.4 = Counter32: 282
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.4.5 = Counter32: 272
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.4.6 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.4.7 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.4.8 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.5.0 = Counter32: 0
```

```
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.5.1 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.5.2 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.5.3 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.5.4 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.5.5 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.5.6 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.5.7 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.5.8 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.6.0 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.6.1 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.6.2 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.6.3 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.6.4 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.6.5 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.6.6 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.6.7 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.6.8 = Gauge32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.7.0 = Counter32: 139
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.7.1 = Counter32: 139
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.7.2 = Counter32: 73
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.7.3 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.7.4 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.7.5 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.7.6 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.7.7 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.7.8 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.8.0 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.8.1 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.8.2 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.8.3 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.8.4 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.8.5 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.8.6 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.8.7 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.8.8 = Counter32: 0
```

To get the type of logging device that is attached to the FortiGate:

```
root@PC05:/home/tester/autolib/trunk# snmpwalk -v2c -c REGR-SYS 172.16.200.1
1.3.6.1.4.1.12356.101.21.2.1.1.3
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.3.0 = STRING: "syslog"
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.3.1 = STRING: "syslog2"
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.3.2 = STRING: "syslog3"
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.3.3 = STRING: "syslog4"
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.3.4 = STRING: "faz"
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.3.5 = STRING: "faz2"
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.3.6 = STRING: "faz3"
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.3.7 = STRING: "webtrends"
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.3.8 = STRING: "fds"
```

To get the present state of the logging device that is attached to the FortiGate:

```
root@PC05:/home/tester/autolib/trunk# snmpwalk -v2c -c REGR-SYS 172.16.200.1
1.3.6.1.4.1.12356.101.21.2.1.1.2
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.2.0 = INTEGER: 1
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.2.1 = INTEGER: 1
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.2.2 = INTEGER: 1
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.2.3 = INTEGER: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.2.4 = INTEGER: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.2.5 = INTEGER: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.2.6 = INTEGER: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.2.7 = INTEGER: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.2.8 = INTEGER: 0
```

To get the failed log count value:

```
root@PC05:/home/tester/autolib/trunk# snmpwalk -v2c -c REGR-SYS 172.16.200.1
1.3.6.1.4.1.12356.101.21.2.1.1.7
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.7.0 = Counter32: 139
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.7.1 = Counter32: 139
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.7.2 = Counter32: 73
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.7.3 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.7.4 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.7.5 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.7.6 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.7.7 = Counter32: 0
FORTINET-FORTIGATE-MIB::fnFortiGateMib.21.2.1.1.7.8 = Counter32: 0
```

WAN optimization

Many multi-location enterprise environments reduce costs and consolidate resources by centralizing applications or providing applications in the cloud. Applications that work fine on a local LAN, such as Windows File Sharing (CIFS), email exchange (MAPI), and many others, suffer from bandwidth limitations and latency issues when accessed over a WAN. This results in a loss of productivity and a perceived need for expensive network upgrades. WAN optimization reduces your network overhead and removes unnecessary traffic for a better overall performance experience and eliminates the need for costly WAN link upgrades between data centers and other expensive solutions for your network traffic growth.

FortiOS WAN optimization provides an inexpensive and comprehensive solution that maximizes your WAN performance and provides intelligent bandwidth management and unmatched consolidated security performance.

FortiOS includes license-free WAN optimization on most current FortiGate devices with internal storage that also support SSL offloading.

This feature is not supported on FortiGate models with 2 GB RAM or less. See Proxy-related features not supported on FortiGate 2 GB RAM models on page 126 for more information.

Features

The following features are available through WAN optimization:

Protocol optimization

Protocol optimization is effective for applications designed for the LAN that do not function well on low bandwidth, high latency networks. See Protocol optimization on page 4874 for more information.

Byte caching

Byte caching improves caching by accelerating the transfer of similar, but not identical content. See Byte caching on page 4875 for more information.

Compression

The WAN optimization feature uses gzip compression to reduce the data transfer size. See Compression on page 4876 for more information.

SSL offloading

SSL is used by many organizations to keep WAN communications private. WAN optimization utilizes the SSL offloading capabilities of the FortiGate FortiASIC hardware to accelerate SSL traffic across the WAN. The FortiGate unit handles SSL encryption and decryption for corporate servers providing SSL encrypted connections over the WAN. See SSL Offloading on page 4870 for more information.

WAN optimization and HA

You can configure WAN optimization on a FortiGate HA cluster. See HA on page 4879 for more information.

Secure tunneling

FortiOS WAN optimization supports secure SSL-encrypted tunnels between FortiGate units on the WAN. See Secure tunneling on page 4872 for more information.

Prerequisites

FortiGate WAN optimization is proprietary to Fortinet Inc.. It will not work with other vendors' WAN optimization or offloading features.

Before you begin to configure WAN optimization, please go through the following steps:

1. To use WAN optimization, your FortiGate unit must support it and not all FortiGate units do. In general, your FortiGate unit must include a hard disk to support these features. See Feature Platform Matrix.
2. If the physical FortiGate has only one hard disk, make sure it is selected for WAN optimization. See Disk usage on page 4868 for more information.
3. For FortiGate-VM, ensure you create two virtual disks besides the boot disk for WAN optimization to work.
4. To be able to configure WAN optimization from the GUI you should begin by going to *System* > *Feature Visibility* and turning on *WAN Opt. & Cache*.
5. If you enable virtual domains (VDOMs) on the FortiGate unit, WAN optimization is available separately for each VDOM.

At this stage, the following installation and configuration conditions are assumed:

- For WAN optimization you have already successfully installed two or more FortiGate units at various locations across your WAN.
- You have administrative access to the GUI or CLI.
- The FortiGate units are integrated into your WAN or other networks.
- The system time, DNS settings, administrator password, and network interfaces have been configured.
- Firmware, FortiGuard Antivirus, and FortiGuard Antispam updates are completed.

- Your Fortinet products have been registered. Register your Fortinet Inc. products at the Fortinet Technical Support website, https://support.fortinet.com.

Disk usage

Both logging and WAN optimization use hard disk space to save data. In FortiOS, you cannot use the same hard disk for both WAN optimization and logging.

- If the FortiGate has one hard disk, then it can be used for either disk logging or WAN optimization, but not both. By default, the hard disk is used for disk logging.

  Only physical FortiGate devices can switch between disk logging and WAN optimization in the case of a single hard disk. FortiGate-VM must have two virtual disks apart from the boot disk for WAN optimization to work.

- If the FortiGate has two hard disks, then one disk is always used for disk logging and the other disk is always used for WAN optimization.

On the FortiGate, go to *System* > *Settings* > *Disk Settings* to switch between *Local Log* and *WAN Optimization*.

You can also change disk usage from the CLI using the following command:

```
config system storage
    edit <name>
        set usage  {log | wanopt}
        set wanopt-mode {mix | wanopt | webcache}
    next
end
```

Option	Description
`wanopt-mode`	WAN optimization mode: • `mix`: This is the default mode. • `wanopt`: Recommended if only the WANopt feature is enabled. • `webcache`: Recommended if only the webcache feature is enabled. If only one of the two features is being used, using the applicable recommended mode will give a higher cache capacity and improve performance.

Enabling WAN optimization affects more than just disk logging. The following table shows other features affected by the FortiGate disk configuration.

Feature	1 hard disk	2 hard disks
Logging	Not supported	Supported
Report/Historical FortiView	Not supported	Supported

Feature	1 hard disk	2 hard disks
Firewall Packet Capture (Policy Capture and Interface Capture)	Not supported	Supported
AV Quarantine	Not supported	Supported
IPS Packet Capture	Not supported	Supported
DLP Archive	Not supported	Supported
Sandbox DB & Results	FortiSandbox database and results are also stored on disk, but will not be affected by this feature.	
Remote Logging	Remote logging (including logging to FortiAnalyzer and Syslog servers) is not affected by this features.	

⚠ Changing the disk setting formats the disk, erases current data stored on the disk, and disables either disk logging or WAN optimization.

The following sections provide information about WAN optimization:

- Overview on page 4869
- Example topologies on page 4881
- Configuration examples on page 4883

Overview

The following topics provide an overview on WAN optimization:

- Peers and authentication groups on page 4871
- Tunnels on page 4872
- Transparent mode on page 4873
- Protocol optimization on page 4874
- Cache service and video caching on page 4876
- Manual and active-passive on page 4877
- Monitoring performance on page 4878
- System and feature operation with WAN optimization on page 4879
- Best practices on page 4881

Client/server architecture

Traffic across a WAN typically consists of clients on a client network communicating across a WAN with a remote server network. The clients do this by starting communication sessions from the client network across a

WAN to the server network. When you have FortiGates on each end, you can optimize these sessions by adding a WAN optimization profile.

To use WAN optimization, the FortiGate units can operate in either NAT or transparent mode. The client-side and server-side FortiGate units do not have to be operating in the same mode. The client-side FortiGate unit is located between the client network and the WAN. The server-side FortiGate unit is located between the server network and the WAN.

WAN optimization profiles are only added to the client-side. The server-side FortiGate unit employs the WAN optimization settings set in the WAN optimization profile on the client-side FortiGate unit.

Profiles

Use WAN optimization profiles to apply WAN optimization techniques to traffic to be optimized. In a WAN optimization profile you can select the protocols to be optimized and for HTTP protocol. You can also enable SSL offloading (if supported), secure tunneling, byte caching, transparent mode, and optionally select an authentication group. You can edit the default WAN optimization profile or create new ones. See Configuration examples on page 4883 for sample configuration.

Transparent mode	Servers receiving packets after WAN optimization see different source addresses depending on whether or not you select *Transparent Mode*. See Transparent mode on page 4873 for more information.
Authentication group	Select this option and select an authentication group so that the client and server-side FortiGate units must authenticate with each other before starting the WAN optimization tunnel. See Peers and authentication groups on page 4871 for more information.
Protocol	Select CIFS, FTP, HTTP, MAPI or TCP to apply protocol optimization for the selected protocols. See Protocol optimization on page 4874 for more information.
SSL Offloading	Select to apply SSL offloading for HTTPS traffic. You can use SSL offloading to offload SSL encryption and decryption from one or more HTTP servers to the FortiGate unit. If you enable this option, you must also use one of the following option to achieve SSL offloading: • Enable ssl profile with *ssl deep-inspection* in the WAN optimization firewall policy on the client-side and use the CLI command `config firewall ssl-server` to add an SSL server on the server-side for each HTTP server that you want to offload SSL encryption and decryption for. • Enable ssl profile with *ssl deep-inspection* in the WAN optimization firewall policy on client-side and WAN optimization proxy policy on server-side to accept SSL encrypted traffic.
SSL Secure Tunneling	The WAN optimization tunnel is encrypted using SSL encryption. You must also add an authentication group to the profile. See Secure tunneling on page 4872 for more information.

Byte Caching	Select to apply WAN optimization byte caching to the sessions accepted by this rule. See Byte caching on page 4875 for more information.

Peers and authentication groups

The client-side and server-side FortiGate units are called WAN optimization peers. The client and server roles relate to how a session is started. Any FortiGate unit configured for WAN optimization can be a client-side and a server-side FortiGate unit at the same time, depending on the direction of the traffic. Client-side FortiGate units initiate WAN optimization sessions and server-side FortiGate units respond to the session requests.

During this process, the WAN optimization peers identify and optionally authenticate each other. The authentication group is optional unless the tunnel is a secure tunnel. You need to add authentication groups to support secure tunneling between WAN optimization peers.

Peer requirements

WAN optimization requires the following configuration on each peer:

- The peer must have a unique host ID.
- Unless authentication groups are used, peers authenticate each other using host ID values. Do not leave the local host ID at its default value.
- The peer must know the host IDs and IP addresses of all of the other peers that it can start WAN optimization tunnels with. This does not apply if you use authentication groups that accept all peers.
- If a FortiGate unit or VDOM is operating in transparent mode, WAN optimization uses the management IP address as the peer IP address of the FortiGate unit instead of the address of an interface.
- All peers must have the same local certificate installed on their FortiGate units if the units authenticate by local certificate. Furthermore, system time must be enabled to ensure that SSL/TLS certificate expiry can be validated. Similarly, if the units authenticate by pre-shared key (password), administrators must know the password. The type of authentication is selected in the authentication group. This applies only if you use authentication groups.

Tunnel requests for peer authentication

When a client-side FortiGate unit attempts to start a WAN optimization tunnel with a peer server-side FortiGate unit, the tunnel request includes the following information:

- The client-side host ID.
- The name of an authentication group, if included in the rule that initiates the tunnel.
- The authentication method it specifies (pre-shared key or certificate), if an authentication group is used.
- The type of tunnel (secure or not).

If the tunnel request does not include an authentication group, authentication will be based on the client-side host ID in the tunnel request.

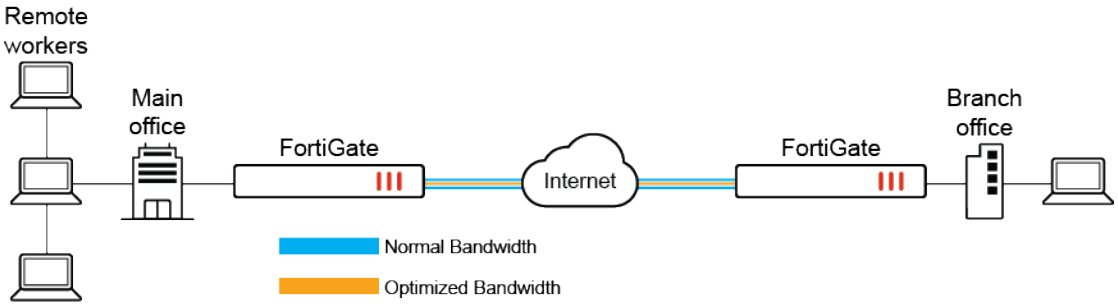
If the tunnel request includes an authentication group, the authentication will be based on the settings of this group as follows:

- The server-side FortiGate unit searches its own configuration for the name of the authentication group in the tunnel request. If no match is found, the authentication fails.
- If a match is found, the server-side FortiGate unit compares the authentication method in the client and server authentication groups. If the methods do not match, the authentication fails.
- If the authentication methods match, the server-side FortiGate unit tests the peer acceptance settings in its copy of the authentication group.
  - If the *Accept peer(s)* setting is *Any* , the authentication is successful.
  - If the *Accept peer(s)* setting is *One*, the server-side FortiGate unit compares the client-side host ID in the tunnel request with the peer name in the server-side authentication group. If the names match, authentication is successful. If a match is not found, authentication fails.
  - If the *Accept peer(s)* setting is *Defined Peers Only*, the server-side FortiGate unit compares the client-side host ID in the tunnel request with the server-side peer list. If a match is found, authentication is successful. If a match is not found, authentication fails.

After a tunnel is established, multiple WAN optimization sessions can start and stop between peers without restarting the tunnel.

Tunnels

All optimized traffic passes between the FortiGate units over a WAN optimization tunnel. Traffic in the tunnel can be sent in plain text or encrypted. Both plain text and the encrypted tunnels use TCP destination port 7810.



Secure tunneling

You can configure a WAN optimization profile to use SSL secure tunneling to encrypt the traffic in the WAN optimization tunnel using AES-128bit-CBC SSL. WAN optimization uses FortiASIC acceleration to accelerate SSL decryption and encryption of the secure tunnel. Secure tunneling can be enabled in both manual and active-passive WAN optimization configuration. See Manual and active-passive on page 4877 for more information.

To use secure tunneling, you must add an authentication group and enable *SSL Secure Tunneling* in a WAN optimization profile. The *Accept peers(s)* setting of the authentication group does not affect secure tunneling. See Secure tunneling configuration example on page 4894 for sample configuration.

Tunnel sharing

Tunnel sharing means multiple WAN optimization sessions share the same tunnel. It can improve performance by reducing the number of WAN optimization tunnels between FortiGate units. Having fewer tunnels means less data to manage. Also, tunnel setup requires more than one exchange of information between the ends of the tunnel. Once the tunnel is set up, each new session that shares the tunnel avoids tunnel setup delays.

Tunnel sharing also uses bandwidth more efficiently by reducing the chances that small packets will be sent down the tunnel. For example, suppose a FortiGate unit is processing five WAN optimization sessions and each session has 100 bytes to send. If these sessions use a shared tunnel, WAN optimization combines the packets from all five sessions into one 500-byte packet. If each session uses its own private tunnel, five 100-byte packets will be sent instead. Each packet also requires a TCP ACK reply. The combined packet in the shared tunnel requires one TCP ACK packet. The separate packets in the private tunnels require five.

Tunnel sharing is not always recommended and may not always be the best practice. For instance, aggressive and non-aggressive protocols should not share the same tunnel. An aggressive protocol can be defined as a protocol that is able to get more bandwidth than a non-aggressive protocol, for example, HTTP and FTP. If aggressive and non-aggressive protocols share the same tunnel, the aggressive protocols may take all of the available bandwidth. As a result, the performance of less aggressive protocols could be reduced.

To avoid this problem, rules for HTTP and FTP traffic should have their own tunnel. To do this, set `tunnel-sharing` to `private` for WAN optimization rules that accept HTTP or FTP traffic.

It is also useful to set `tunnel-sharing` to `express-shared` for applications, such as Telnet, that are very interactive but not aggressive.

Set `tunnel-sharing` to `shared` for applications that are not aggressive and are not sensitive to latency or delays.

Example configuration

To configure tunnel sharing for HTTP traffic in a WAN optimization profile:

```
config wanopt profile
    edit default
        config http
            set tunnel-sharing {express-shared | private | shared}
        end
    next
end
```

Transparent mode

WAN optimization is transparent to users. This means that with WAN optimization in place, clients connect to servers in the same way as they would without WAN optimization. However, servers receiving packets after WAN optimization see different source addresses depending on whether or not transparent mode is selected for WAN optimization. If transparent mode is selected, WAN optimization keeps the original source address of the packets, so servers appear to receive traffic directly from clients. Routing on the server network should be configured to route the traffic back to the client network.

Some protocols, for example CIFS, may not function as expected if transparent mode is not selected. In most cases, for CIFS WAN optimization you should select transparent mode and confirm the server network can route traffic as described to support transparent mode.

If transparent mode is not selected, the source address of the packets received by servers is changed to the address of the server-side FortiGate unit interface that sends the packets to the servers. So servers appear to receive packets from the server-side FortiGate unit. Routing on the server network is simpler in this case because client addresses are not involved. All traffic appears to come from the server-side FortiGate unit and not from individual clients.

Do not confuse WAN optimization transparent mode with FortiGate transparent mode. WAN optimization transparent mode is similar to source NAT. FortiGate's transparent mode is a system setting that controls how the FortiGate unit (or a VDOM) processes traffic.

Configuring transparent mode

You can configure transparent mode by selecting *Transparent* in a WAN optimization profile. The profile is added to an active WAN optimization policy.

When you configure a passive WAN optimization policy you can accept or override the active policy transparent setting. From the GUI you can do this by setting the *Passive* option as follows:

- *Default*: Use the transparent setting in the WAN optimization profile added to the active policy (client-side configuration).
- *Transparent*: Override the active policy transparent mode setting and impose transparent mode. Packets exiting the FortiGate keep their original source addresses.
- *Non-transparent*: Override the active policy transparent mode setting and impose non-transparent mode. Packets exiting the FortiGate have their source address changed to the address of the server-side FortiGate unit interface that sends the packets to the servers.

To configure a passive wan optimization policy in the CLI:

```
config firewall policy
    edit <policy ID>
        set srcintf <Incoming interface>
        set wanopt-passive-opt {default | transparent | non-transparent}
    next
end
```

Protocol optimization

Protocol optimization techniques optimize bandwidth use across the WAN. These techniques can improve the efficiency of communication across the WAN optimization tunnel by reducing the amount of traffic required by communication protocols. You can apply protocol optimization to CIFS, FTP, HTTP, MAPI, and general TCP sessions. You can apply general TCP optimization to MAPI sessions.

FortiGate supports WAN optimization for SMBv1, SMBv2, and SMBv3 (unencrypted only) protocols.

For example, CIFS provides file access, record locking, read/write privileges, change notification, server name resolution, request batching, and server authentication. CIFS requires many background transactions to successfully transfer a single file. This is usually not a problem across a LAN. However, across a WAN, latency and bandwidth reduction can slow down CIFS performance.

When you select the CIFS protocol in a WAN optimization profile, the FortiGate units at both ends of the WAN optimization tunnel use a number of techniques to reduce the number of background transactions that occur over the WAN for CIFS traffic.

If a policy accepts a range of different types of traffic, you can set *Protocol* to *TCP* to apply general optimization techniques to TCP traffic. However, applying this TCP optimization is not as effective as applying more protocol-specific optimization to specific types of traffic. TCP protocol optimization uses techniques such as TCP SACK support, TCP window scaling and window size adjustment, and TCP connection pooling to remove TCP bottlenecks.

Byte caching

Byte caching breaks large units of application data (for example, a file being downloaded from a web page) into small chunks of data, labeling each chunk of data with a hash of the chunk and storing those chunks and their hashes in a database. The database is stored on a WAN optimization storage device. Then, instead of sending the actual data over the WAN tunnel, the FortiGate unit sends the hashes. The FortiGate unit at the other end of the tunnel receives the hashes and compares them with the hashes in its local byte caching database. If any hashes match, that data does not have to be transmitted over the WAN optimization tunnel. The data for any hashes that does not match is transferred over the tunnel and added to that byte caching database. Then the unit of application data (the file being downloaded) is reassembled and sent to its destination.

The stored byte caches are not application specific. Byte caches from a file in an email can be used to optimize downloading that same file or a similar file from a web page.

The result is less data transmitted over the WAN. Initially, byte caching may reduce performance until a large enough byte caching database is built up.

To enable byte caching, select *Byte Caching* in a WAN optimization profile.

Byte caching cannot determine whether or not a file is compressed (for example a zip file), and caches compressed and non-compressed versions of the same file separately.

Dynamic data chunking for byte caching

Dynamic data chunking can improve byte caching by improving detection of data chunks that are already cached in changed files or in data embedded in traffic using an unknown protocol. Dynamic data chunking can only be enabled from the CLI and is available for HTTP, CIFS and FTP.

Dynamic data chunking is disabled by default and `prefer-chunking` is set to `fix`.

To enable dynamic data chunking for HTTP in the default WAN optimization profile:

```
config wanopt profile
    edit default
        config http
            set prefer-chunking dynamic
        end
    next
end
```

Compression

The WAN optimization feature uses gzip compression to reduce the data transfer size. This compression operates independently of byte caching, meaning that even if byte caching is disabled, gzip compression can still be applied, depending on the nature of the traffic.

- Compression is applied selectively. For example, gzip is not effective on already-compressed or random data (such as. SSL traffic, binary files, or already gzip-encoded HTTP content).
- Byte caching is still required for optimal WANOpt results. gzip alone may not provide measurable gains, and its effectiveness depends on the compressibility of the data.
- When byte caching is disabled, compression may still occur, but protocol-specific optimization techniques remain the primary means of improving performance for such traffic.

Cache service and video caching

Two features that can only be configured in the CLI include cache service and video caching.

Cache service

The `config wanopt cache-service` command is used to configure cache-service clusters between multiple FortiGates. The result is that the cache-service daemons of the different FortiGates can collaborate for serving web cache entries.

See config wanopt cache-service in the CLI Reference guide for more configuration information.

Video caching

The `config wanopt content-delivery-network-rule` command configures web-caching, including the video-cache matching rules.

See config wanopt content-delivery-network-rule in the CLI Reference guide for more configuration information.

Manual and active-passive

You can create manual (peer-to-peer) and active-passive WAN optimization configurations.

There are a few key differences between manual and active-passive mode:

- For manual mode, the tunnels are always up which makes it more resource extensive as compared to active-passive.
- The performance of active-passive mode is lower than manual mode for the new connection.
- The active-passive mode can be used to deploy tunnel dynamically using *Authentication groups* set to accept *Any* peers which eliminates the need of defining peers manually. This is not possible with manual mode.

  This setting is only recommended when you do not know the peer host IDs or IP addresses of the peers that will use this authentication group. This setting is most often used with FortiGate units that do not have static IP addresses, for example units that use DHCP.

- For manual mode, traffic shaping cannot be applied to traffic on the server-side. See Traffic shaping on page 4881 for more information.

Manual (peer to peer) configurations

Manual configurations allow for WAN optimization between one client-side FortiGate unit and one server-side FortiGate unit. Manual WAN optimization requires a manual WAN optimization firewall policy on the client-side FortiGate unit and a WAN optimization proxy policy on the server-side FortiGate unit.

In a manual mode configuration, the client-side peer can only connect to the named server side peer. When the client-side peer initiates a tunnel with the server-side peer, the packets that initiate the tunnel include extra information so that the server-side peer can determine that it is a peer-to-peer tunnel request. This extra information is required because the server-side peer does not require a WAN optimization firewall policy; however, you need to add the client peer host ID and IP address to the server-side FortiGate unit peer list. See Manual (peer-to-peer) WAN optimization configuration example on page 4884 for a sample configuration.

Active-passive configurations

Active-passive WAN optimization requires an active WAN optimization firewall policy on the client-side FortiGate unit and a passive WAN optimization firewall policy on the server-side FortiGate unit. The server-side FortiGate unit also requires a WAN optimization proxy policy.

You can use the passive policy to control WAN optimization address translation by specifying transparent mode or non-transparent mode. You can also use the passive policy to apply security profiles, web caching, and other FortiGate features at the server-side FortiGate unit. For example, if a server-side FortiGate unit is protecting a web server, the passive policy could enable web caching.

A single passive policy can accept tunnel requests from multiple FortiGate units as long as the server-side FortiGate unit includes their peer IDs and all of the client-side FortiGate units include the server-side peer ID. See Active-passive WAN optimization configuration example on page 4889 for a sample configuration.

The WAN optimization proxy policy can only be added from the CLI and policies with `proxy` set to `wanopt` do not appear on the GUI.

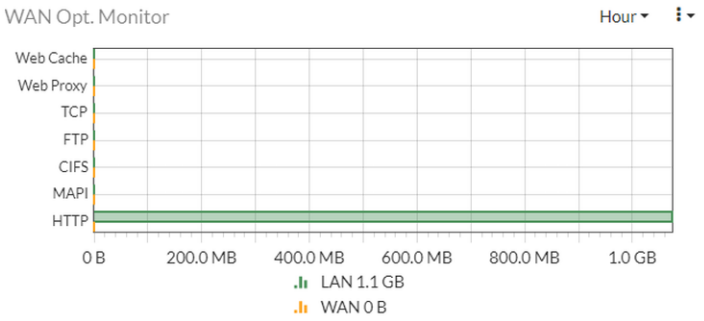
Monitoring performance

Using *WAN Opt. & Cache* widgets, you can confirm that a FortiGate unit is optimizing traffic and view estimates of the amount of bandwidth saved. These include peers manually added to the configuration as well as discovered peers.

To add *WAN Opt. & Cache* widgets go to *Dashboard* > *Status* > *Add Widget* > *WAN Opt. & Cache* and add *WAN Opt. Monitor* and *Peer Monitor*.

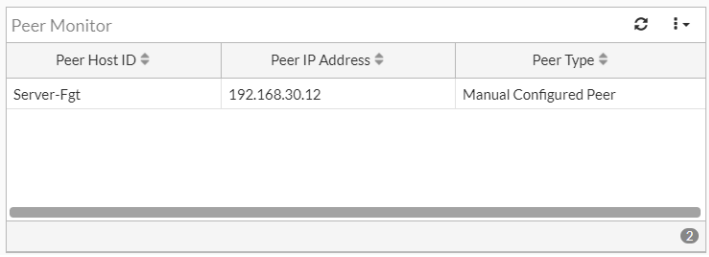
Wan Opt. Monitor

The *Wan Opt. Monitor* shows how WAN optimization is reducing the amount of traffic on the WAN for each WAN optimization protocol by showing the amount of WAN and LAN traffic. If WAN optimization is being effective, the amount of WAN traffic should be lower than the amount of LAN traffic.



Peer Monitor

The *Peer Monitor* lists all of the WAN optimization peers that a FortiGate unit can perform WAN optimization with.



System and feature operation with WAN optimization

This section contains the following information:

- HA on page 4879
- Memory usage on page 4879
- Distributing WAN optimization processing on page 4879
- Distributing WAN optimization to multiple CPU cores on page 4880
- Identity policies and load balancing on page 4880
- Traffic shaping on page 4881

HA

You can configure WAN optimization on a FortiGate HA cluster. The recommended best practice HA configuration for WAN optimization is active-passive mode. When the cluster is operating, all WAN optimization sessions are processed by the primary unit only. Even if the cluster is operating in active-active mode, HA does not load-balance WAN optimization sessions.

You can also form a WAN optimization tunnel between a cluster and a standalone FortiGate unit or between two clusters.

In a cluster, only the primary unit stores the byte cache database. This database is not synchronized to the subordinate units. So, after a failover, the new primary unit must rebuild its byte cache. Rebuilding the byte cache can happen relatively quickly because the new primary unit gets byte cache data from the other FortiGate unit that it is participating with in WAN optimization tunnels.

Memory usage

To accelerate and optimize disk access and to provide better throughput and less latency, FortiOS WAN optimization uses provisioned memory to reduce disk I/O and increase disk I/O efficiency. In addition, WAN optimization requires a small amount of additional memory per session for comprehensive flow control logic and efficient traffic forwarding.

When WAN optimization is enabled you will see a reduction in available memory. The reduction increases when more WAN optimization sessions are being processed. If you are thinking of enabling WAN optimization on an operating FortiGate unit, make sure its memory usage is not maxed out during high traffic periods.

In addition to using the system dashboard to see the current memory usage you can use the `get test wad 2` command to see how much memory is currently being used by WAN optimization.

Distributing WAN optimization processing

The `wad-worker` balancing algorithm supports a more balanced dispersal of traffic to the wad processes even if the bulk of the traffic is coming from a small set of sources or single source.

By default, dispatching traffic to WAD workers is based on source affinity. This may negatively affect performance when users have another explicit proxy in front of the FortiGate. Source affinity causes the

FortiGate to process the traffic as if it originated from the single (or small set of) IP address of the outside proxy. This results in the use of one, or a small number, of WAD processes.

By disabling `wad-source-affinity` the traffic is balanced over all of the WAD processes. The WAD dispatcher will not assign the traffic based on the source IP address, but will assign the traffic to available workers in a round-robin fashion.

To configure WAD source affinity:

```
config system global
    set wad-source-affinity {enable | disable}
end
```

⚠ Handling the traffic by different WAD workers results in losing cached related benefits of using source affinity, as there is the memory cache on the current wad worker and if a new connection is handled by another worker, the cache will not be hit.

This is explained by the warning message that appears when it is disabled:

```
WARNING: Disabling this option results in some features to be unsupported. IP-based user
authentication, disclaimer messages, security profile override, authentication cookies,
MAPI scanning, and some video caches such as YouTube are not supported.
```

```
Do you want to continue? (y/n)
```

Distributing WAN optimization to multiple CPU cores

By default WAN optimization is handled by half of the CPU cores in a FortiGate unit. For example, if your FortiGate unit has 4 CPU cores, by default two will be used for WAN optimization. You can use the following command to change the number of CPU cores that are used.

```
config system global
    set wad-worker-count <number>
end
```

The `wad-worker-count` can be between `1` and the total number of CPU cores in your FortiGate unit. Adding more cores may enhance WAN optimization but reduce the performance of other FortiGate systems.

Identity policies and load balancing

WAN optimization and firewall policies compatibility varies depending on the type of policy:

- WAN optimization is not compatible with firewall load balancing.
- WAN optimization is compatible with source and destination NAT options in firewall policies (including firewall virtual IPs). If a virtual IP is added to a policy, the traffic that exits the WAN optimization tunnel has its destination address changed to the virtual IPs mapped to IP address and port.
- WAN optimization is compatible with user identity-based and device identity security policies. If a session is allowed after authentication or device identification the session can be optimized.

Traffic shaping

Traffic shaping works for WAN optimization traffic that is not in a WAN optimization tunnel. So traffic accepted by a WAN optimization policy on a client-side FortiGate unit can be shaped on ingress. However, when the traffic enters the WAN optimization tunnel, traffic shaping is not applied.

In manual mode:

- Traffic shaping works as expected on the client-side FortiGate unit.
- Traffic shaping cannot be applied to traffic on the server-side FortiGate unit.

In active-passive mode:

- Traffic shaping works as expected on the client-side FortiGate unit.
- If transparent mode is enabled in the WAN optimization profile, traffic shaping also works as expected on the server-side FortiGate unit.
- If transparent mode is not enabled, traffic shaping works partially on the server-side FortiGate unit.

Best practices

WAN optimization and explicit proxy best practices include:

- WAN optimization tunnel sharing is recommended for similar types of WAN optimization traffic. However, tunnel sharing for different types of traffic is not recommended. For example, aggressive and non-aggressive protocols should not share the same tunnel.
- Active-passive HA is the recommended HA configuration for WAN optimization.
- Configure WAN optimization authentication with specific peers. Accepting any peer is not recommended as this can be less secure.

Example topologies

All FortiGate WAN optimization topologies consist of two FortiGate units operating as WAN optimization peers intercepting and optimizing traffic crossing the WAN between the private networks.

The following topics cover a few of the example topologies:

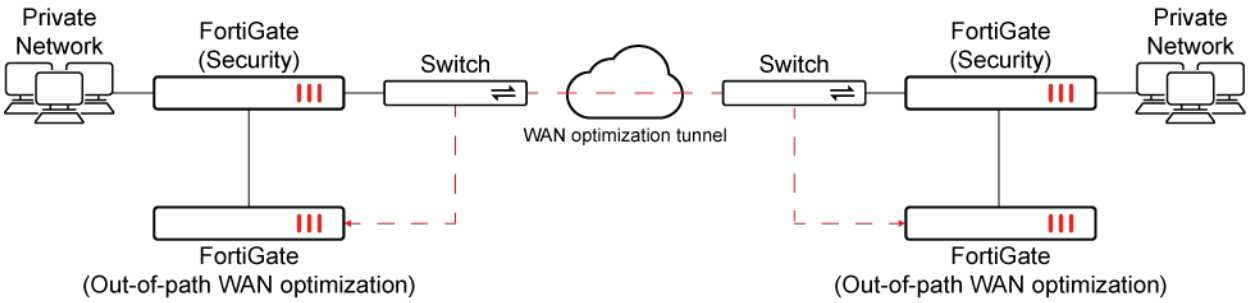
- In-path WAN optimization topology on page 4882
- Out-of-path WAN optimization topology on page 4882
- Topology for multiple networks on page 4882

In-path WAN optimization topology

FortiGate units can be deployed as typical security devices that protect private networks connected to the WAN and also perform WAN optimization. The WAN optimization configuration intercepts traffic to be optimized as it passes through the FortiGate unit and uses a WAN optimization tunnel with another FortiGate unit to optimize the traffic that crosses the WAN.

You can add web caching to any WAN optimization topology when users on a private network communicate with web servers located across the WAN on another private network.

Out-of-path WAN optimization topology



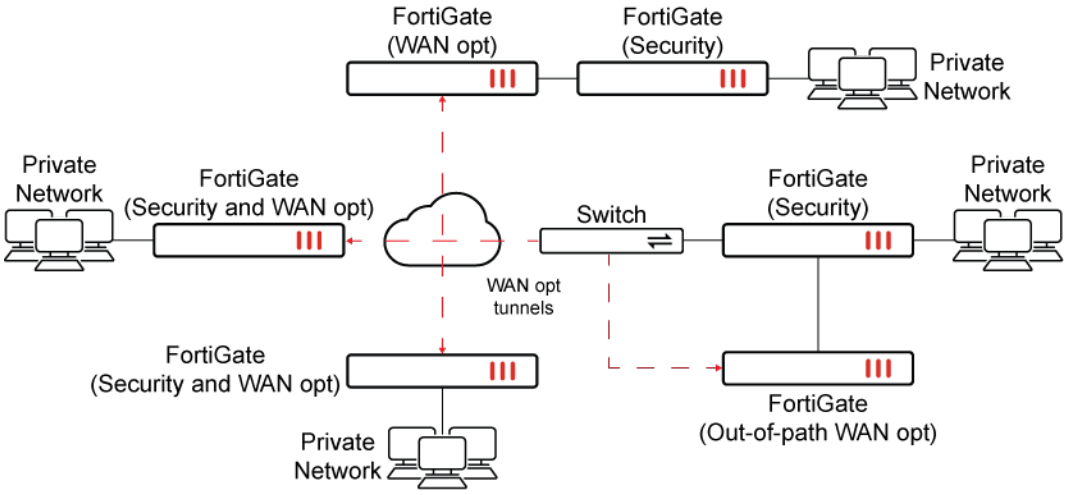
In an out-of-path topology, one or both of the FortiGate units configured for WAN optimization are not directly in the main data path. Instead, the out-of-path FortiGate unit is connected to FortiGate units in the data path, and the device is configured to redirect sessions to be optimized to the out-of-path FortiGate unit. The FortiGate units in the data path use a method such as policy routing to redirect traffic to be optimized to the out-of-path FortiGate units. The out-of-path FortiGate units establish a WAN optimization tunnel between each other and optimize the redirected traffic.

One of the benefits of out-of-path WAN optimization is that out-of-path FortiGate units only perform WAN optimization and do not have to process other traffic. An in-path FortiGate unit configured for WAN optimization also has to process other non-optimized traffic on the data path.

Topology for multiple networks

As shown below, you can create multiple WAN optimization configurations between many private networks. Whenever WAN optimization occurs, it is always between two FortiGate units, but you can configure any

FortiGate unit to perform WAN optimization with any of the other FortiGate units that are part of your WAN.

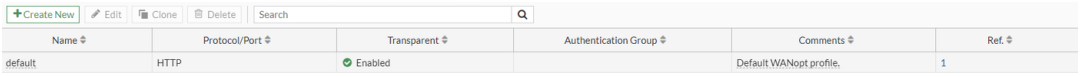


You can also configure WAN optimization between FortiGate units with different roles on the WAN. FortiGate units configured as security devices and for WAN optimization can perform WAN optimization as if they are single-purpose FortiGate units just configured for WAN optimization. The WAN optimization configuration is the same for FortiGate units deployed as security devices and for single-purpose WAN optimization FortiGate units. The only differences would result from the different network topologies.

Configuration examples

The following pages are used in the WAN optimization configuration examples demonstrated in the subsequent sections:

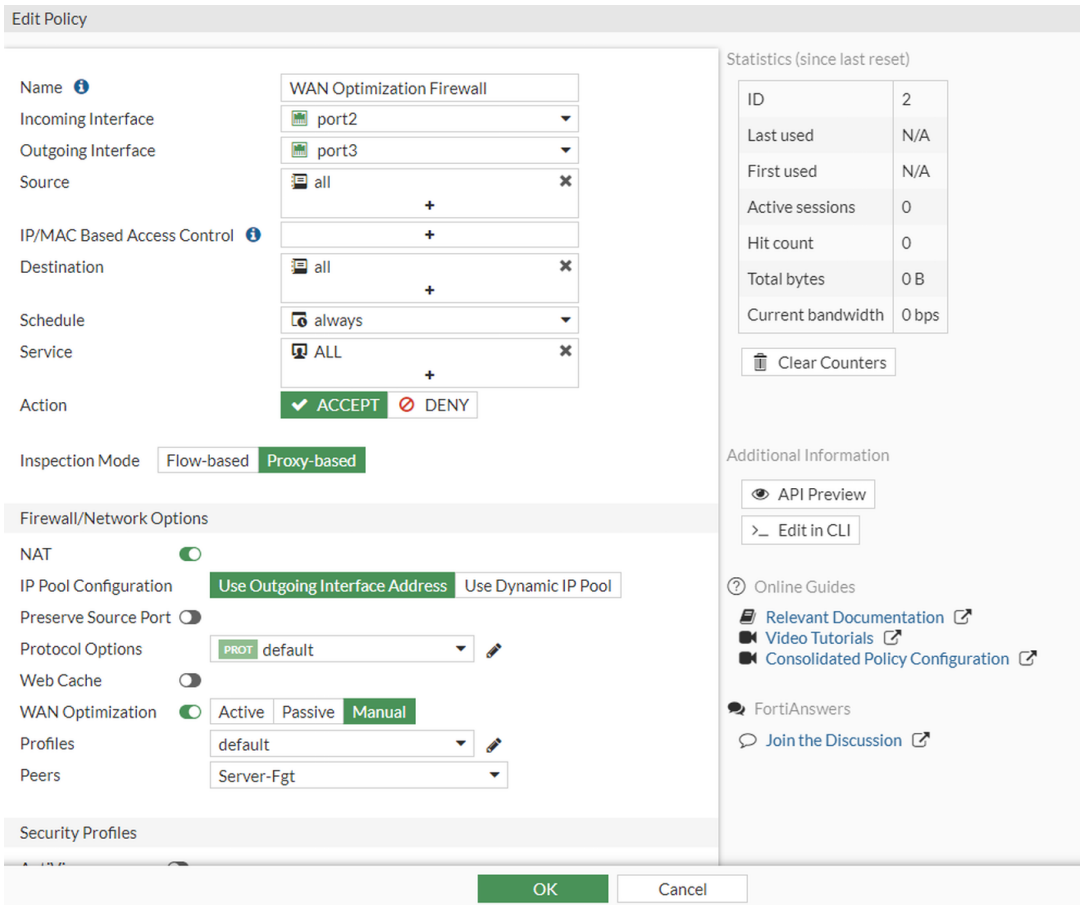
- *WAN Opt. & Cache* > *Profiles*: Configure the default WAN optimization profile to optimize HTTP traffic on client side.



- *WAN Opt. & Cache* > *Peers*: Change the *Host ID* and add *Peer Host ID* and *IP address* on both client and server side.

- *WAN Opt. & Cache* > *Authentication Groups*: Add an authentication group for the authentication purpose on both client and server side. (Optional)

- *Policy & Objects* > *Firewall Policy*: Add a WAN optimization firewall policy on the client side or on both client and server side depending on the WAN optimization configuration. See the examples for more information.



A WAN optimization firewall policy is a firewall policy running in *Proxy-based* inspection mode with *WAN Optimization* enabled. A WAN optimization firewall policy cannot be configured with inspection mode set to *Flow-based*.

The following topics provide instructions on different WAN optimization configuration examples:

- Manual (peer-to-peer) WAN optimization configuration example on page 4884
- Active-passive WAN optimization configuration example on page 4889
- Secure tunneling configuration example on page 4894
- Testing and troubleshooting the configuration on page 4900

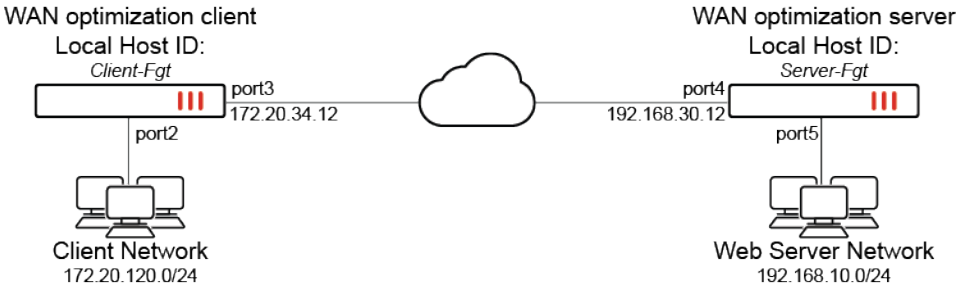
Manual (peer-to-peer) WAN optimization configuration example

Please ensure that the Prerequisites on page 4867 are met before proceeding with the configuration example.

See Manual (peer to peer) configurations on page 4877 for conceptual information.

This example configuration includes a client-side FortiGate unit called Client-Fgt with a WAN IP address of 172.20.34.12. This unit is in front of a network with IP address 172.20.120.0. The server-side FortiGate unit is called Server-Fgt with a WAN IP address of 192.168.30.12. This unit is in front of a web server network with IP address 192.168.10.0.

This example customizes the default WAN optimization profile on the client-side FortiGate unit and adds it to the WAN optimization firewall policy. You can also create a new WAN optimization profile.



General configuration steps

This section breaks down the configuration for this example into smaller procedures:

1. Configure the client-side FortiGate unit:
   - Add peers.
   - Configure the default WAN optimization profile to optimize HTTP traffic.
   - Add a manual WAN optimization firewall policy.
2. Configure the server-side FortiGate unit:
   - Add peers.
   - Add a WAN optimization proxy policy.

Configuring manual WAN optimization from the GUI

Use the following steps to configure the example configuration from the GUI:

To configure the client-side FortiGate unit:

1. Go to *WAN Opt. & Cache* > *Peers* and change the *Host ID* of the client-side FortiGate unit:
   a. Click *Change*. The *Host ID* pane opens.
   b. Enter a new *Host ID*:

Host ID	Client-Fgt

   c. Click *OK*.
2. Create the server-side FortiGate unit peer:
   a. Select *Create New*. The *New WAN Optimization Peer* opens.
   b. Configure the following settings:

Peer Host ID	Server-Fgt

IP address	192.168.30.12

c. Click *OK*.

3. Go to *WAN Opt. & Cache > Profiles* and edit the default profile:
   a. Select the default profile and click *Edit*.
   b. Under *Protocol Options*, edit *HTTP*.
   c. Set *Status* to *Enable* and click *Apply*.
   d. Click *OK*.
4. Go to *Policy & Objects > Firewall Policy* to add a manual WAN optimization firewall policy to the client-side FortiGate unit that accepts traffic to be optimized:
   a. Click *Create New*.
   b. Enter a *Name* and configure the following settings:

Incoming Interface	port2
Outgoing Interface	port3
Source	all
Destination	all
Schedule	always
Service	ALL
Action	ACCEPT

   c. Set *Inspection Mode* to *Proxy-based*.
   d. Enable *WAN Optimization* and configure the following settings:

WAN Optimization	Manual
Profiles	default
Peers	Server-Fgt

   e. Click *OK* to save the policy.

To configure the server-side FortiGate unit:

1. Go to *WAN Opt. & Cache > Peers* and change the *Host ID* of the server-side FortiGate unit:
   a. Click *Change*. The *Host ID* pane opens.
   b. Enter a new *Host ID*:

Host ID	Server-Fgt

   c. Click *OK*.
2. Create the client-side FortiGate unit peer:
   a. Select *Create New*. The *New WAN Optimization Peer* opens.
   b. Configure the following settings:

Peer Host ID	Client-Fgt
IP address	172.20.34.12

c. Click *OK*.

3. Enter the following CLI command to add a WAN optimization proxy policy to accept WAN optimization tunnel connections:

```
config firewall proxy-policy
    edit 0
        set proxy wanopt
        set dstintf port5
        set srcaddr all
        set dstaddr all
        set action accept
        set schedule always
        set service ALL
    next
end
```

Configuring basic WAN optimization from the CLI

Use the following steps to configure the example configuration from the CLI.

To configure the client-side FortiGate unit:

1. Change the Host ID of the client-side FortiGate:

```
config wanopt settings
    set host-id Client-Fgt
end
```

2. Add the Host ID of the server-side FortiGate:

```
config wanopt peer
    edit Server-Fgt
        set ip 192.168.30.12
    next
end
```

3. Edit the *default* WAN optimization profile and enable HTTP WAN optimization:

```
config wanopt profile
    edit default
        config http
            set status enable
        end
    next
end
```

4. Add a WAN optimization firewall policy to accept the traffic to be optimized:

```
config firewall policy
    edit 0
        set srcintf port2
        set dstintf port3
        set srcaddr all
        set dstaddr all
        set action accept
        set service ALL
        set schedule always
        set inspection-mode proxy
        set wanopt enable
        set wanopt-profile default
        set wanopt-detection off
        set wanopt-peer Server-Fgt
    next
end
```

When you set the detection mode to off, the policy becomes a manual mode WAN optimization firewall, which is reflected on the GUI.

To configure the server-side FortiGate unit:

1. Change the Host ID of the server-side FortiGate:

```
config wanopt settings
    set host-id Server-Fgt
end
```

2. Add the Host ID of the client-side FortiGate:

```
config wanopt peer
    edit Client-Fgt
        set ip 172.20.34.12
    next
end
```

3. Add a WAN optimization proxy policy:

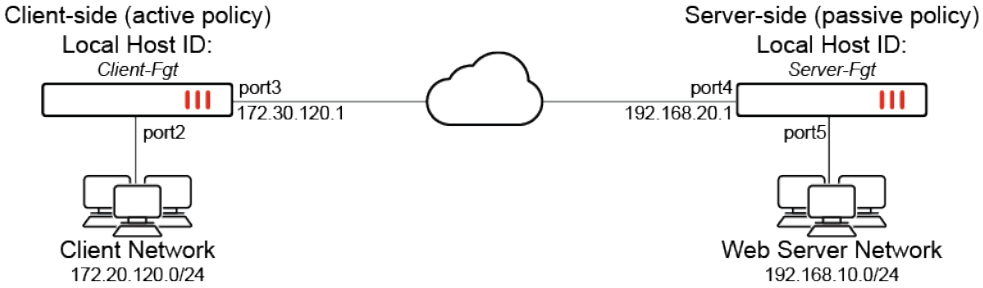
```
config firewall proxy-policy
    edit 0
        set proxy wanopt
        set dstintf port5
        set srcaddr all
        set dstaddr all
        set action accept
        set schedule always
        set service ALL
    next
end
```

Active-passive WAN optimization configuration example

Please ensure that the Prerequisites on page 4867 are met before proceeding with the configuration example.

See Active-passive configurations on page 4877 for conceptual information.

This example configuration includes a client-side FortiGate unit called Client-Fgt with a WAN IP address of 172.30.120.1. This unit is in front of a network with IP address 172.20.120.0. The server-side FortiGate unit is called Server-Fgt and has a WAN IP address of 192.168.20.1. This unit is in front of a web server network with IP address 192.168.10.0.



General configuration steps

This section breaks down the configuration for this example into smaller procedures:

1. Configure the client-side FortiGate unit:
   - Add peers.
   - Add a WAN optimization profile to optimize CIFS, FTP, and HTTP traffic.
   - Add an active WAN optimization firewall policy.
2. Configure the server-side FortiGate unit:
   - Add peers.
   - Add a passive WAN optimization firewall policy.
   - Add a WAN optimization proxy policy.

Configuring active-passive WAN optimization from the GUI

Use the following steps to configure the example configuration from the GUI.

To configure the client-side FortiGate unit:

1. Go to *WAN Opt. & Cache* > *Peers* and change the *Host ID* of the client-side FortiGate unit:
   a. Click *Change*. The *Host ID* pane opens.
   b. Enter a new *Host ID*:

Host ID	Client-Fgt

c. Click *OK*.

2. Create the server-side FortiGate unit peer:
   a. Select *Create New*. The *New WAN Optimization Peer* opens.
   b. Configure the following settings:

Peer Host ID	Server-Fgt
IP address	192.168.20.1

   c. Click *OK*.
3. Go to *WAN Opt & Cache > Profiles* to add a WAN optimization profile to optimize CIFS, HTTP, and FTP traffic:
   a. Select *Create New*.
   b. Enter the profile name:

Name	Custom-wan-opt-pro

   c. In the *Protocol Options* section:
      i. Edit *CIFS*.
      ii. Set *Status* to *Enable*.
      iii. Click *Apply*.
      iv. Repeat these steps to edit and enable *FTP* and *HTTP*.
   d. Click *OK*.
4. Go to *Policy & Objects > Firewall Policy* to add an active WAN optimization firewall policy:
   a. Click *Create New*.
   b. Enter a *Name* and configure the following settings:

Incoming Interface	port2
Outgoing Interface	port3
Source	all
Destination	all
Schedule	always
Service	HTTP FTP SMB
Action	ACCEPT

   c. Set *Inspection Mode* to *Proxy-based*.
   d. Enable WAN optimization and configure the following settings:

WAN Optimization	Active
Profile	Custom-wan-opt-pro

   e. Click *OK*.

To configure the server-side FortiGate unit:

1. Go to *WAN Opt. & Cache > Peers* and change the *Host ID* of the server-side FortiGate unit:
   a. Click *Change*. The *Host ID* pane opens.
   b. Enter a new *Host ID*:

Host ID	Server-Fgt

   c. Click *OK*.
2. Create the client-side FortiGate unit peer:
   a. Select *Create New*. The *New WAN Optimization Peer* opens.
   b. Configure the following settings:

Peer Host ID	Client-Fgt
IP address	172.30.120.1

   c. Click *OK*.
3. Go to *Policy & Objects > Firewall Policy* to add a passive WAN optimization firewall policy:
   a. Click *Create New*.
   b. Enter a *Name* and configure the following settings:

Incoming Interface	port4
Outgoing Interface	port5
Source	all
Destination	all
Schedule	always
Service	ALL
Action	ACCEPT

   c. Set *Inspection Mode* to *Proxy-based*.
   d. Enable *WAN Optimization* and configure the following settings:

WAN Optimization	Passive
Passive Option	Default

   e. Click *OK*.
4. Add a WAN optimization proxy policy from the CLI:

```
config firewall proxy-policy
    edit 0
        set proxy wanopt
        set dstintf port5
        set srcaddr all
        set dstaddr all
        set action accept
        set schedule always
```

```
        set service ALL
    next
end
```

Configuring basic active-passive WAN optimization from the CLI

Use the following steps to configure the example configuration from the CLI.

To configure the client-side FortiGate unit:

1. Change the Host ID of the client-side FortiGate:

```
config wanopt settings
    set host-id Client-Fgt
end
```

2. Add the Host ID of the server-side FortiGate:

```
config wanopt peer
    edit Server-Fgt
        set ip 192.168.20.1
    next
end
```

3. Add a WAN optimization profile to optimize CIFS, HTTP, and FTP traffic:

```
config wanopt profile
    edit Custom-wan-opt-pro
        config cifs
            set status enable
        end
        config http
            set status enable
        end
        config ftp
            set status enable
        end
    next
end
```

4. Add an active WAN optimization firewall policy:

```
config firewall policy
    edit 0
        set srcintf port2
        set dstintf port3
        set srcaddr all
        set dstaddr all
        set action accept
        set service HTTP FTP SMB
        set schedule always
```

```
        set inspection-mode proxy
        set wanopt enable
        set wanopt-detection active
        set wanopt-profile Custom-wan-opt-pro
    next
end
```

To configure the server-side FortiGate unit:

1. Change the Host ID of the server-side FortiGate:

```
config wanopt settings
    set host-id Server-Fgt
end
```

2. Add the Host ID of the client-side FortiGate:

```
config wanopt peer
    edit Client-Fgt
        set ip 172.30.120.1
    next
end
```

3. Add a passive WAN optimization firewall policy:

```
config firewall policy
    edit 0
        set srcintf port4
        set dstintf port5
        set srcaddr all
        set dstaddr all
        set action accept
        set service ALL
        set schedule always
        set inspection-mode proxy
        set wanopt enable
        set wanopt-detection passive
        set wanopt-passive-opt default
    next
end
```

4. Add a WAN optimization proxy policy:

```
config firewall proxy-policy
    edit 0
        set proxy wanopt
        set dstintf port5
        set srcaddr all
        set dstaddr all
        set action accept
        set schedule always
        set service ALL
```

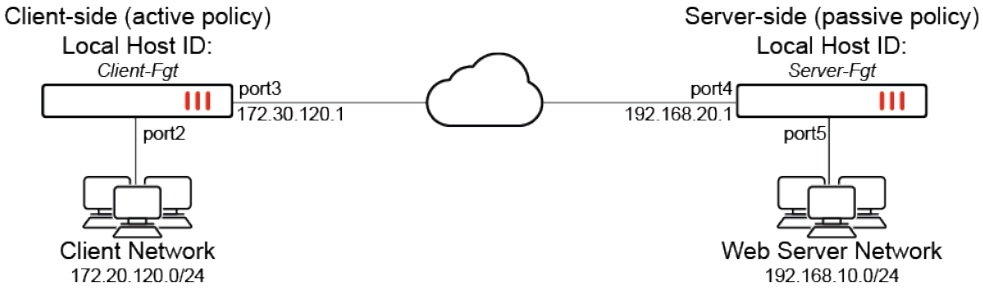
```
    next
end
```

Secure tunneling configuration example

Please ensure that the Prerequisites on page 4867 are met before proceeding with the configuration example.

See Secure tunneling on page 4872 for conceptual information.

This example configuration includes a client-side FortiGate unit called Client-Fgt with a WAN IP address of 172.30.120.1. This unit is in front of a network with IP address 172.20.120.0. The server-side FortiGate unit is called Server-Fgt and has a WAN IP address of 192.168.20.1. This unit is in front of a web server network with IP address 192.168.10.0.



General configuration steps

This section breaks down the configuration for this example into smaller procedures:

1. Configure the client-side FortiGate unit:
   - Add peers.
   - Add an authentication group.
   - Add an active WAN optimization firewall policy.
2. Configure the server-side FortiGate unit:
   - Add peers.
   - Add the same authentication group.
   - Add a passive WAN optimization firewall policy.
   - Add a WAN optimization proxy policy.

Configuring WAN optimization with secure tunneling from the GUI

Use the following steps to configure the example configuration from the GUI.

To configure the client-side FortiGate unit:

1. Go to *WAN Opt. & Cache > Peers* and change the *Host ID* of the client-side FortiGate unit:
   a. Click *Change*. The *Host ID* pane opens.
   b. Enter a new *Host ID*:

Host ID	Client-Fgt

   c. Click *OK*.
2. Create the server-side FortiGate unit peer:
   a. Select *Create New*. The *New WAN Optimization Peer* opens.
   b. Configure the following settings:

Peer Host ID	Server-Fgt
IP address	192.168.20.1

   c. Click *OK*.
3. Go to *WAN Opt. & Cache > Authentication Groups* to add the authentication group to be used for secure tunneling:
   a. Click *Create New* and configure the following settings:

Name	Auth-Secure-Tunnel
Authentication Method	Pre-shared key
Pre-shared key	********
Accept peer(s)	Defined Peers Only

   b. Click *OK*.
4. Go to *WAN Opt. & Cache > Profiles* to add a WAN optimization profile that enables secure tunneling and includes the authentication group:
   a. Click *Create New*.
   b. Enter a *Name*:

Name	Secure-wan-opt-pro

   c. Enable *Authentication group*:

Authentication group	Auth-Secure-Tunnel

   d. In the *Protocol Options* section, edit *HTTP*:
      i. Set *Status* to *Enable*.
      ii. Click *Apply*.
      iii. Set *SSL Secure Tunneling* to *Enable*.
      iv. Click *Apply*.
   e. Click *OK*.
5. Go to *Policy & Objects > Firewall Policy* to add an active WAN optimization firewall policy:
   a. Click *Create New*.
   b. Enter a *Name* and configure the following settings:

Incoming Interface	port2
Outgoing Interface	port3
Source	all
Destination	all
Schedule	always
Service	HTTP
Action	ACCEPT

c. Set *Inspection Mode* to *Proxy-based*.
d. Enable *WAN Optimization* and configure the following settings:

WAN Optimization	Active
Profile	Secure-wan-opt-pro

e. Click *OK*.

To configure the server-side FortiGate unit:

1. Go to *WAN Opt. & Cache > Peers* and change the *Host ID* of the server-side FortiGate unit:
   a. Click *Change*. The *Host ID* pane opens.
   b. Enter a new *Host ID*:

Host ID	Server-Fgt

   c. Click *OK*.
2. Create the client-side FortiGate unit peer:
   a. Select *Create New*. The *New WAN Optimization Peer* opens.
   b. Configure the following settings:

Peer Host ID	Client-Fgt
IP address	172.30.120.1

   c. Click *OK*.
3. Go to *WAN Opt. & Cache > Authentication Groups* to add the authentication group to be used for secure tunneling:
   a. Click *Create New* and configure the following settings:

Name	Auth-Secure-Tunnel
Authentication Method	Pre-shared key
Pre-shared key	********
Accept peer(s)	Defined Peers Only

   b. Click *OK*.
4. Go to *Policy & Objects > Firewall Policy* to add an passive WAN optimization firewall policy:

a. Click *Create New*.
b. Enter a *Name* and configure the following settings:

Incoming Interface	port4
Outgoing Interface	port5
Source	all
Destination	all
Schedule	always
Service	ALL
Action	ACCEPT

c. Set *Inspection Mode* to *Proxy-based*.
d. Enable *WAN Optimization* and configure the following settings:

WAN Optimization	Passive
Passive Option	Default

e. Click *OK*.

5. Add a WAN optimization proxy policy from the CLI:

```
config firewall proxy-policy
    edit 0
        set proxy wanopt
        set dstintf port5
        set srcaddr all
        set dstaddr all
        set action accept
        set schedule always
        set service ALL
    next
end
```

Configuring WAN optimization with secure tunneling from the CLI

Use the following steps to configure the example configuration from the CLI.

To configure the client-side FortiGate unit:

1. Change the Host ID of the client-side FortiGate:

```
config wanopt settings
    set host-id Client-Fgt
end
```

2. Add the Host ID of the server-side FortiGate:

```
config wanopt peer
    edit Server-Fgt
        set ip 192.168.20.1
    next
end
```

3. Add a new authentication group to be used for secure tunneling:

```
config wanopt auth-group
    edit Auth-Secure-Tunnel
        set auth-method psk
        set peer-accept defined
        set psk ********
    next
end
```

4. Add a WAN optimization profile that enabled secure tunneling and includes the authentication group, enables HTTP protocol optimization, and enables secure tunneling for HTTP traffic:

```
config wanopt profile
    edit Secure-wan-opt-pro
        set auth-group Auth-Secure-Tunnel
        config http
            set status enable
            set secure-tunnel enable
        end
    next
end
```

5. Add an active WAN optimization firewall policy that enables secure tunneling:

```
config firewall policy
    edit 0
        set srcintf port2
        set dstintf port3
        set srcaddr all
        set dstaddr all
        set action accept
        set service HTTP
        set schedule always
        set inspection-mode proxy
        set wanopt enable
        set wanopt-detection active
        set wanopt-profile Secure-wan-opt-pro
    next
end
```

To configure the server-side FortiGate unit:

1. Change the Host ID of the server-side FortiGate:

```
config wanopt settings
    set host-id Server-Fgt
end
```

2. Add the Host ID of the client-side FortiGate:

```
config wanopt peer
    edit Client-Fgt
        set ip 172.30.120.1
    next
end
```

3. Add an authentication group to be used for secure tunneling:

```
config wanopt auth-group
    edit Auth-Secure-Tunnel
        set auth-method psk
        set peer-accept defined
        set psk ********
    next
end
```

4. Add a passive WAN optimization firewall policy:

```
config firewall policy
    edit 0
        set srcintf port4
        set dstintf port5
        set srcaddr all
        set dstaddr all
        set action accept
        set service ALL
        set schedule always
        set inspection-mode proxy
        set wanopt enable
        set wanopt-detection passive
        set wanopt-passive-opt default
    next
end
```

5. Add a WAN optimization proxy policy:

```
config firewall proxy-policy
    edit 0
        set proxy wanopt
        set dstintf port5
        set srcaddr all
        set dstaddr all
        set action accept
```

```
        set schedule always
        set service ALL
    next
end
```

Testing and troubleshooting the configuration

To test the configuration attempt, start a web browsing session between the client network and the web server network. For example, from a PC on the client network browse to the IP address of a web server on the web server network, for example http://192.168.10.13. Even though this address is not on the client network, you should be able to connect to this web server over the WAN optimization tunnel.

If you can connect, the *WAN Opt. Monitor* widget should show the protocol that has been optimized (in this case HTTP) and the *Peer Monitor* widget displays the *Peer* information. To add the *WAN Opt. Monitor* and the *Peer Monitor*, go to *Dashboard > Status > Add Widget > WAN Opt. & Cache* and add *WAN Opt. Monitor* and *Peer Monitor*. See Monitoring performance on page 4878 for more information.

If you cannot connect, try the following to diagnose the problem:

- Review your configuration and make sure all details such as address ranges, peer names, and IP addresses are correct.
- Check routing on the FortiGate units and on the client and web server networks to make sure packets can be forwarded as required. The FortiGate units must be able to communicate with each other, routing on the client network must allow packets destined for the web server network to be received by the client-side FortiGate unit, and packets from the server-side FortiGate unit must be able to reach the web servers. See Routing concepts on page 575 for more information.

You can use `get` and `diagnose` commands to display information about how WAN optimization is operating.

Example output

The command output for the client-side FortiGate unit shows 10 tunnels all created by the manual WAN optimization configuration:

```
# diagnose wad tunnel list

Tunnel: id=100 type=manual
vd=0 shared=no uses=0 state=2
peer name=Server-Fgt id=100 ip=192.168.30.12
SSL-secured-tunnel=no auth-grp=
bytes_in=348 bytes_out=384

Tunnel: id=99 type=manual
vd=0 shared=no uses=0 state=2
peer name=Server-Fgt id=99 ip=192.168.30.12
SSL-secured-tunnel=no auth-grp=
bytes_in=348 bytes_out=384

Tunnel: id=98 type=manual
```

```
vd=0 shared=no uses=0 state=2
peer name=Server-Fgt id=98 ip=192.168.30.12
SSL-secured-tunnel=no auth-grp=
bytes_in=348 bytes_out=384

Tunnel: id=39 type=manual
vd=0 shared=no uses=0 state=2
peer name=Server-Fgt id=39 ip=192.168.30.12
SSL-secured-tunnel=no auth-grp=
bytes_in=1068 bytes_out=1104

Tunnel: id=7 type=manual
vd=0 shared=no uses=0 state=2
peer name=Server-Fgt id=7 ip=192.168.30.12
SSL-secured-tunnel=no auth-grp=
bytes_in=1228 bytes_out=1264

Tunnel: id=8 type=manual
vd=0 shared=no uses=0 state=2
peer name=Server-Fgt id=8 ip=192.168.30.12
SSL-secured-tunnel=no auth-grp=
bytes_in=1228 bytes_out=1264

Tunnel: id=5 type=manual
vd=0 shared=no uses=0 state=2
peer name=Server-Fgt id=5 ip=192.168.30.12
SSL-secured-tunnel=no auth-grp=
bytes_in=1228 bytes_out=1264

Tunnel: id=4 type=manual
vd=0 shared=no uses=0 state=2
peer name=Server-Fgt id=4 ip=192.168.30.12
SSL-secured-tunnel=no auth-grp=
bytes_in=1228 bytes_out=1264

Tunnel: id=1 type=manual
vd=0 shared=no uses=0 state=2
peer name=Server-Fgt id=1 ip=192.168.30.12
SSL-secured-tunnel=no auth-grp=
bytes_in=1228 bytes_out=1264

Tunnel: id=2 type=manual
vd=0 shared=no uses=0 state=2
peer name=Server-Fgt id=2 ip=192.168.30.12
SSL-secured-tunnel=no auth-grp=
bytes_in=1228 bytes_out=1264

Tunnels total=10 manual=10 auto=0
```

The command output shows three tunnels all created by the active-passive WAN optimization configuration:

```
# diagnose wad tunnel list
```

```
Tunnel: id=22 type=auto
    vd=0 shared=no uses=1 state=2
    peer name=Server-Fgt id=42 ip=192.168.20.1 (best guess)
    SSL-secured-tunnel=no auth-grp=
    bytes_in=56693 bytes_out=10831

Tunnel: id=24 type=auto
    vd=0 shared=no uses=1 state=2
    peer name=Server-Fgt id=44 ip=192.168.20.1 (best guess)
    SSL-secured-tunnel=no auth-grp=
    bytes_in=14833 bytes_out=3896

Tunnel: id=26 type=auto
    vd=0 shared=no uses=1 state=2
    peer name=Server-Fgt id=46 ip=192.168.20.1 (best guess)
    SSL-secured-tunnel=no auth-grp=
    bytes_in=481 bytes_out=176

Tunnels total=3 manual=0 auto=3
```

The command output shows a tunnel created by active passive WAN optimization configuration with secure tunneling:

```
# diagnose wad tunnel list

Tunnel: id=3 type=auto
    vd=0 shared=no uses=1 state=2
    peer name=Server-Fgt id=49 ip=192.168.20.1 (best guess)
    SSL-secured-tunnel=yes auth-grp=Auth-Secure-Tunnel
    bytes_in=95810 bytes_out=39597

Tunnels total=1 manual=0 auto=1
```

Unlike manual mode, for active-passive configurations, each session will negotiate an active-passive tunnel so an open session is required the display the corresponding output above.

For example, continuous data transfer such as uploading or downloading will display tunnel output in the active-passive configuration, which is in contrast to manual mode where tunnels are always open and ready to use.

VM

This section contains topics on deploying FortiGate-VM:

- Amazon Web Services on page 4903
- Microsoft Azure on page 4903
- Google Cloud Platform on page 4903
- OCI on page 4904
- AliCloud on page 4904
- Private cloud on page 4904
- VM license on page 4904
- Permanent trial mode for FortiGate-VM on page 4913
- PF and VF SR-IOV driver and virtual SPU support on page 4916
- Using OCI IMDSv2 on page 4918
- FIPS cipher mode for AWS, Azure, OCI, and GCP FortiGate-VMs on page 4921
- Cloud-init on page 4923
- TPM support for FortiGate-VM on page 4925

Amazon Web Services

See the FortiOS 8.0.0 AWS Administration Guide.

Microsoft Azure

See the FortiOS 8.0.0 Azure Administration Guide.

For Azure vWAN deployments, see:

- Azure vWAN NGFW Deployment Guide
- Azure vWAN SD-WAN NGFW Deployment Guide

Google Cloud Platform

See the 8.0.0 FortiOS GCP Administration Guide.

OCI

See the 8.0.0 FortiOS OCI Administration Guide.

AliCloud

See the 8.0.0 FortiOS AliCloud Administration Guide.

Private cloud

See FortiGate Private Cloud in the document library.

VM license

You can access *FortiGate VM License* from *Dashboard > Status* in the *Virtual Machine* widget. Click the device license and select *FortiGate VM License*.

FortiGate VM License displays the following information:

Field	Description
License status	Displays one of the following statuses: • *Valid*: VM can connect and validate the license against a FortiManager or FortiGuard server. All features are available. • *Validation overdue*: VM cannot connect and validate against a FortiManager or FortiGuard server. A check is made against how many days the warning status is continuous. If the number is less than 30 days, the status does not change. You may be seeing this status because the network environment does not allow the FortiGate-VM to connect to the FortiGuard server within 30 days. • *30 day Grace Period*: license is expired but within the 30-day grace period. Check the expiration date for evaluation or term-based licenses. • *Duplicate copy*: license is a duplicate copy. FortiGuard returns code 401 and FortiOS sets the license status as an invalid copy. FortiGate firewall policy continues to work during this state. If the FortiGate keeps the duplicate copy status for more than 24 hours, the status changes to invalid.

Field	Description

	As you cannot access *Dashboard > Status* in the *Virtual Machine* widget when the license is in one of the following statuses, they do not display in the *License status* field:
	• *Invalid*: VM cannot connect and validate against a FortiManager or FortiGuard server. A check is made against how many days the warning status is continuous. If the number is 30 days or more, the status changes to invalid. This status also occurs if the duplicate copy status persists for more than 24 hours. FortiOS restricts GUI access until a valid license is uploaded. Firewall policies do not work. FortiGuard downloads are unavailable. When the status is invalid, upon login, FortiOS redirects you to the VM license upload page.
	Reasons for having an invalid status include:
	  • The VM license is expired and has passed the grace period.
	  • Another VM has been already validated with FortiGuard using the same license. See Technical Note: VM License activation for details about duplicated VM instances.
	• *Pending*: temporary state where the VM attempts to validate its license. The GUI displays a loading page with the message *License is being validated by FortiGuard*.

Allocated vCPUs	Number of allocated and total allowable vCPUs
Allocated RAM	Amount of allocated RAM. There are no RAM restrictions.
Expires on	Expiry date (value depends on the type of license)

This information is visible in the CLI by running `get system status`. See CLI troubleshooting.

Uploading a license file

After you submit an order for a FortiGate-VM, Fortinet sends a license registration code to the email address that you entered in the order form. Use this code on the FortiCloud portal to register the FortiGate-VM.

Once the VM is registered, you can download the license file in .LIC format. On the *FortiGate VM License* page, click *Upload*. The system prompts you to reboot and validate the license with the FortiGuard server. Once validated, your FortiGate-VM is fully functional.

The VM license window may also appear immediately after logging in if you are running a VM with an evaluation license that has expired.

In cases where the GUI is inaccessible, you can upload the license using secure copy (SCP).

For information about injecting FortiFlex licenses, see Injecting the FortiFlex license.

To upload the license using SCP:

1. Enable SCP:

```
config system global
    set admin-scp enable
end
```

2. Enable SSH in the administrative access for the interface where the transfer will take place:

```
config system interface
    edit <interface>
        append allowaccess ssh
    next
end
```

3. On your computer, upload the VM license. This example is for Linux:

```
scp <filename> <admin-user>@<FortiGate_IP>:vmlicense
```

License activation and validation

Once the license file is uploaded using the GUI or SCP, the FortiGate-VM initiates a validation process with the FortiGuard servers. For successful activation, the following requirements must be met:

- **Connectivity**: The FortiGate-VM must have a valid route to the internet.
- **Upstream Firewall Rules**: If the FortiGate-VM is behind another firewall, ensure that traffic is allowed to the following Fully Qualified Domain Names (FQDNs):

Anycast FQDNs (Recommended)	Non-Anycast FQDNs (Legacy)
vmactivation1.fortinet.net	update.fortiguard.net
vmactivation2.fortinet.net	usupdate.fortiguard.net
vmactivation2.fortinet.net	eupdate.fortiguard.net

Monitoring the activation process

You can monitor the license registration and communication with FortiGuard in real-time using the CLI. This is particularly useful for troubleshooting connectivity issues during initial setup.

To enable the debug output for the VM license validation process:

```
# diagnose debug application cloudinitd -1
# diagnose debug enable
```

Validation with anycast enabled

When anycast is enabled (default), the FortiGate attempts to resolve and connect to the anycast update servers.

```
# diagnose debug application cloudinitd -1
# diagnose debug enable
```

```
vmlic status:success, valid:1
vmlic load config
vmlic setup vfid:0
vmlic resolve:vmactivation1.fortinet.net
vmlic server:192.168.100.85
vmlic add 192.168.100.85
vmlic setup 192.168.100.85
[222] ssl_add_ftgd_hostname_check: Add hostname checking 'vmactivation1.fortinet.net'
...
Protocol=3.0|Command=VMSetup|Firmware=FGVM64-FW-7.4.x|SerialNumber=FGVMSLTMxxxxxxxx
vmlic setup result:200
```

After the process completes, verify the final license status using the `get system status` command. See CLI troubleshooting on page 4909 for more information.

Validation with anycast disabled

If anycast is disabled, the FortiGate attempts to resolve and connect to the non-anycast update servers.

```
# diagnose debug application cloudinitd -1
# diagnose debug enable
```

```
vmlic status:success, valid:1
vmlic setup vfid:0
vmlic resolve:update.fortiguard.net
vmlic server:192.168.100.205
vmlic add 192.168.100.205
[219] ssl_add_ftgd_hostname_check: Add hostname checking 'update.fortiguard.net'
...
vmlic setup result:200
```

After the process completes, verify the final license status using the `get system status` command. See CLI troubleshooting on page 4909 for more information.

VM license types

FortiGate-VM offers annual subscription licensing (S-series). SKUs are based on the number of vCPUs (1, 2, 4, 8, 16, 32, or unlimited).

FortiGate-VM has a permanent trial license. See Permanent trial mode for FortiGate-VM on page 4913.

The FortiFlex program allows qualified enterprise and MSSP customers to create as many VM entitlements as required. Resource consumption is based upon predefined points that are calculated on a daily basis. See Program guide.

Feature	Trial	S-series	FortiFlex
Licensing and support	Hardware configuration restrictions apply. Support is not available.	Single annually contracted SKU that contains a VM base and a FortiCare/FortiGuard service bundle. Service bundles and a la carte services are available.	Annually contracted program to create multiple sets of a single entitlement per VM. Entitlements contain a VM base and FortiCare/FortiGuard bundle. Service bundles and a la carte services are available.
vCPU number upgrade or downgrade	Not supported.	Supported via co-term. You can also upgrade the service bundle. Contact a Fortinet sales representative to upgrade.	Natively supported via the scale up/down feature.
VDOM support[1]	VDOMs are supported, but restricted by the CPU allowance for the trial license. See the FortiGate-VM data sheet for default limits.	By default, all CPU levels do not support adding VDOMs, but you can add additional VDOMs via a subscription VDOM license. See the FortiGate-VM data sheet for default limits.	By default, all CPU levels do not support adding VDOMs, but you can add additional VDOMs via the scale up/down feature. See the FortiGate-VM data sheet for default limits.

[1] FortiGate-VMs with one VDOM license (S-series, FortiFlex) have a maximum of two VDOMs: an administrative type root VDOM and a traffic type VDOM. To add a second VDOM, first change the root VDOM to administrative type and then add the traffic VDOM.

Applying a FortiFlex token

You can apply a FortiFlex token in the *FortiGate VM License* page for the following VM instance types:

- Newly deployed or expired FortiGate-VM instances. After logging into the FortiOS GUI, a *FortiFlex token* option is available when the license popup appears.
- Already licensed FortiGate-VM instances. You can go to this page from the *Virtual Machine* dashboard widget or from *System* > *FortiGuard*. *FortiFlex token* option is available for migrating into FortiFlex.

Consuming a new vCPU

FortiGate-VM supports automatic vCPU hot-add/hot-remove to the limit of the license entitlement after activating an S-series license or a FortiFlex license. This enhancement removes the requirement of running the CLI command `execute cpu add` or performing a reboot when the FortiGate-VM has a lower number of vCPUs allocated than the licensed number of vCPUs.

CLI troubleshooting

In some cases, you can view more information from the CLI to diagnose issues with VM licensing. This is also useful when the GUI is inaccessible due to an invalid contract.

Before you begin, ensure your FortiGate has the proper routes to connect to the internet. Run all following debug commands for a full picture of the issue.

To view the license status, expiration date, and VM resources:

```
# get system status
Version: FortiGate-VM64-KVM v8.0.0,buildXXXX,200730 (GA)
...
Serial-Number: FGVM08**********
....
License Status: Valid
License Expiration Date: 2026-12-10
VM Resources: 1 CPU/8 allowed, 2010 MB RAM
...
```

To display license details:

```
# diagnose debug vm-print-license
SerialNumber: FGVM08**********
CreateDate: Tue Dec 10 00:57:32 2019
License expires: Thu Dec 10 00:00:00 2026
Expiry: 366
Key: yes
Cert: yes
Key2: yes
Cert2: yes
Model: 08 (11)
CPU: 8
MEM: 2147483647
```

To display license information from FortiGuard:

```
# diagnose hardware sysinfo vm full
UUID:     abbe**************************
valid:    1
```

```
status:   1
code:     200
warn:     0
copy:     0
received: 4604955037
warning:  4600905081
recv:     202009152207
dup:
```

Field	Value	Description
valid	0	Invalid
	1	Valid
status	0	Startup
	1	Success
	2	Warning
	3	Error
	4	Invalid copy
	5	Evaluation license expired.
	6	Grace period. For FortiFlex, there is a two-hour grace period to begin passing traffic upon retrieving the license from FortiCloud.
code	2xx, 3xx	Success
	200	Valid
	202	Accepted (treated as correct response code)
	4xx	Error
	400	Expired
	401	Duplicate
	5xx, 500	Warning
	502	Invalid. Cannot connect to FortiGuard distribution servers.
	6xx	Evaluation license expired
	Other codes	Error

The following are examples of common combinations:

Combination	Indicates...
valid: 1 status: 1 code: 200	License is valid and functioning normally.

Combination	Indicates...
valid: 1 status: 4 code: 401	License is valid but running on a duplicate instance.
valid: 0 status: 2 code: 502	System cannot connect to FortiGuard.
valid: 0 status: 3 code: 400	License is expired and invalid.
valid: 0 status: 3 code: 0	VM is unlicensed.

For FortiFlex licenses, the following command allows you to enter the license token and proxy information:

```
# execute vm-license <token> https://<username>:<password>@<proxy IP address>:<proxy port>
```

FortiOS can receive the following error codes from the FortiCare server:

```
1 - Runtime error (server unhandled error on FortiCare sever)
57 - License Token is invalid
58 - License Token is already used and cannot be used again to retrieve license key
```

The FortiGate can generate the following error code:

```
60 - Failed to request forticare license. Failed to download VM license.
```

Contact Fortinet Support for assistance if your licensing issue persists.

Customizing the FortiFlex license token activation retry parameters

FortiOS supports the customization of the retries for FortiFlex license token activation. You can configure the token activation number of retries and the interval between each attempt using the following commands, respectively:

```
execute vm-license-options count <integer>
execute vm-license-options interval <interval length in seconds>
```

If you set `vm-license-options count` to zero, the token activation retries indefinitely until success.

To define the FortiFlex token activation parameters:

1. Set the number of retries allowed:

```
execute vm-license-options count 4
```

2. Set the retry interval:

```
execute vm-license-options interval 5
```

3. Activate the license. FortiOS requests the FortiFlex license token four times, with an interval of five seconds in between, as set.
   - If FortiOS cannot verify the license within the set amount of retries, the download fails:

```
# execute vm-license F4FC697D65428013FAKE
This operation will reboot the system !
Do you want to continue? (y/n)y

Requesting FortiCare license token: *******, proxy:(null)
Requesting FortiCare license token: *******, proxy:(null)
Requesting FortiCare license token: *******, proxy:(null)
Requesting FortiCare license token: *******, proxy:(null)
Failed to download VM license.
```

   - If FortiOS can verify the license within the set number of retries, FortiOS successfully installs the VM license:

```
# execute vm-license 227602862F7E6E9XXXX
This operation will reboot the system !
Do you want to continue? (y/n)y

Requesting FortiCare license token: *******, proxy:(null)
VM license install succeeded. Rebooting firewall.
```

FortiFlex token activation parameters can also be defined in an ISO file using the mime user-data.

To define the parameters in an ISO file:

1. Create a config drive ISO with a MIME file:

```
Content-Type: text/plain; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment; filename="license.txt
"LICENSE-TOKEN: 334ADF7B49F2FEC1XXXX INTERVAL: 5 COUNT: 4
```

   See Cloud-init using config drive for more information.
2. Attach the ISO config drive at boot time. See Cloud-init for more information.
3. Boot up the VM and verify the token activation parameters:

```
# diagnose debug cloudinit show
 >> Found config drive /dev/sr0
```

```
>> Successfully mount config drive
>> MIME parsed preconfig script
>> MIME parsed VM token
>> MIME parsed config script
>> Found metadata source: config drive
>> Run preconfig script
>> FortiGate-VM64  conf sys global
…
>> Trying to install vmlicense ...
>> License-token:334ADF7B49F2FEC1XXXX INTERVAL:5 COUNT:4
>> Run config script
```

Permanent trial mode for FortiGate-VM

FortiGate-VM has a permanent evaluation VM license. The evaluation VM license applies to all private cloud (VMware ESXi, KVM, and so on) and all bring your own license (BYOL) public cloud instances.

When spinning up a new FortiGate-VM, you choose to log in to FortiCloud to activate the VM trial or upload a new license.

Limitations of the evaluation VM license include the following:

- Maximum of one free evaluation copy per FortiCloud account
- Support for low encryption operation only, except for GUI management access and FortiManager communications
- Maximum of 1 CPU and 2 GB of memory
- Maximum of three interfaces, firewall policies, and routes
- No FortiCare support
- No FortiGuard support
- Support for a maximum of two virtual domains (VDOM). When using multi-VDOM mode, the root VDOM must be an admin type and the other can be a traffic VDOM. See VDOM types on page 3836.

To obtain the permanent VM trial license from FortiCare using the CLI:

1. A newly deployed FortiGate-VM no longer has a valid evaluation license, even if the instance has only 1 CPU and 2 GB of memory. Run `get sys stat`. The following output is expected:
   ```
   Version: FortiGate-VM64 v8.0.0,buildXXXX,220715 (interim)
   ...
   Serial-Number: FGVMEVNXFLTGKOBC
   License Status: Invalid
   VM Resources: 1 CPU/1 allowed, 2007 MB RAM/2048 MB allowed
   ```
2. Obtain the permanent VM trial license from FortiCare:

   ```
   execute vm-license-options account-id xxxx@fortinet.com
   ```

   ```
   execute vm-license-options account-password xxxxxxx
   ```

```
execute vm-license
This VM is using the evaluation license. This license does not expire.
Limitations of the Evaluation VM license include:
  1.Support for low encryption operation only
  2.Maximum of 1 CPU and 2GiB of memory
  3.Maximum of three interfaces, firewall policies, and routes each
  4.No FortiCare Support
This operation will reboot the system !
Do you want to continue? (y/n)y

Connection to 10.6.30.74 closed.
```

3. You can run the following commands to check that the permanent VM trial license is valid:
   - `get system status`. The following output is expected:

     ```
     Version: FortiGate-VM64 v8.0.0,buildXXXX,220715 (interim)
     ...
     Serial-Number: FGVMEVNXFLTGKOBC
     License Status: Valid
     VM Resources: 1 CPU/1 allowed, 2007 MB RAM/2048 MB allowed
     ```

   - `diagnose hardware sysinfo vm full`. The following output is expected:

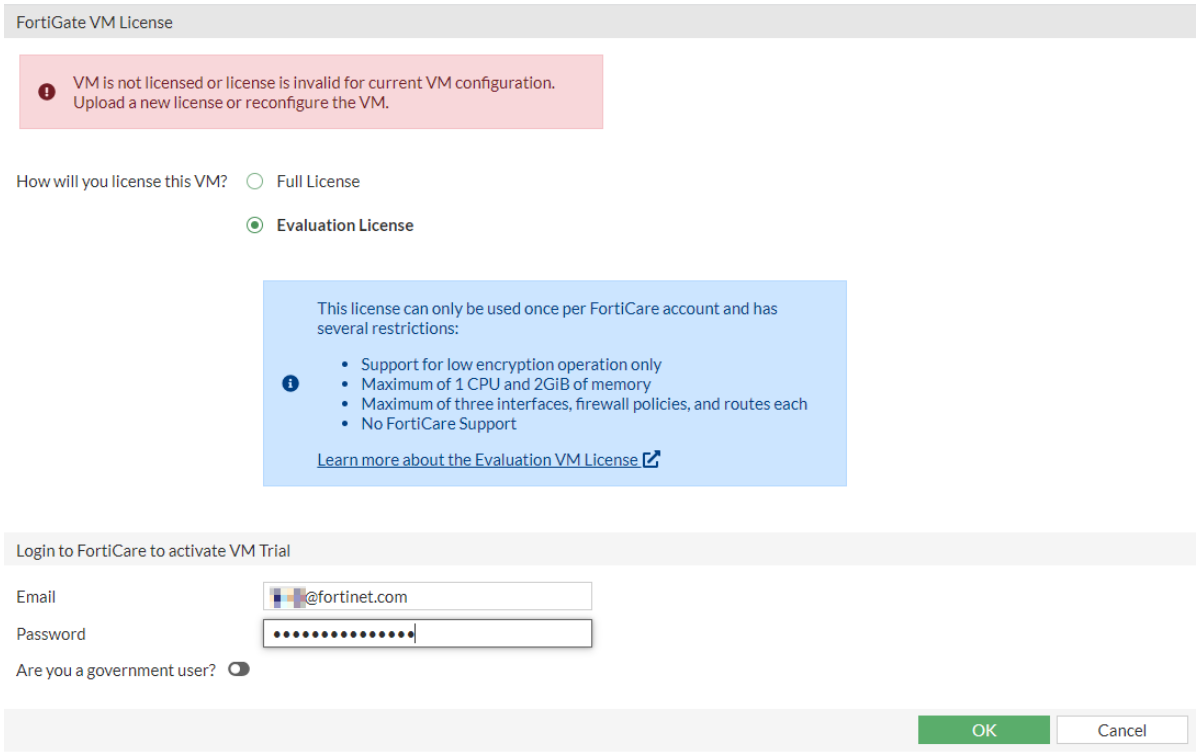
     ```
     UUID:     4213dbbc94f2520b0d75eeafe1b319c7
     valid:    1
     status:   1
     code:     0
     warn:     0
     copy:     0
     received: 4294939472
     warning:  4294939472
     recv:     202207162014
     dup:
     ```

   - `diagnose debug vm-print-license`. The following output is expected:

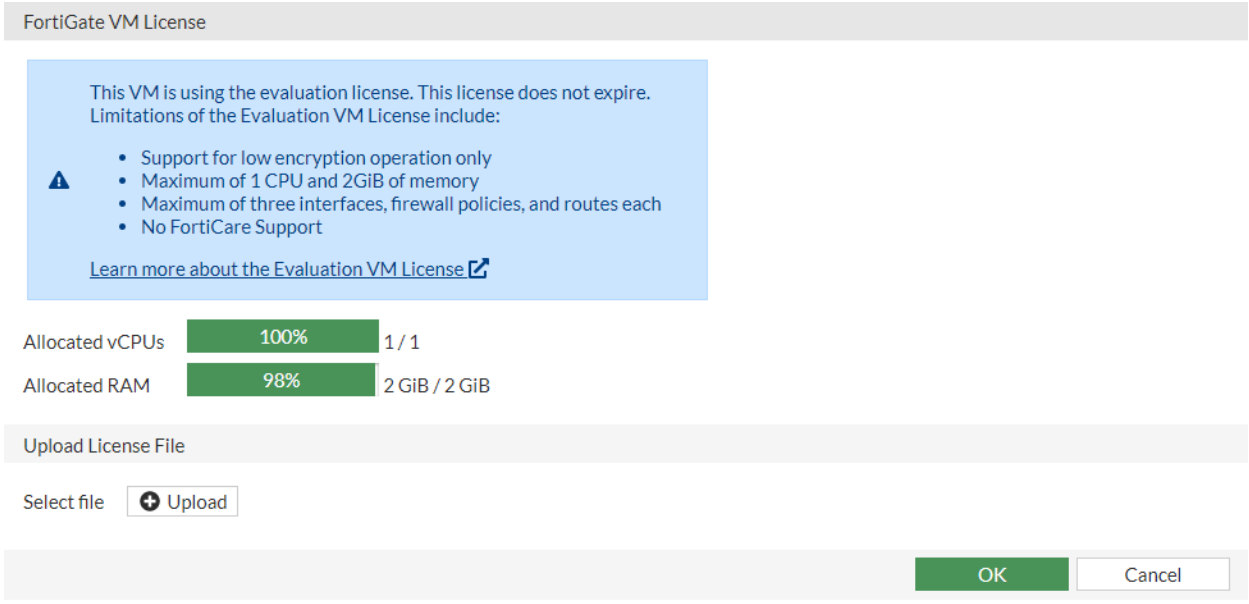
     ```
     SerialNumber: FGVMEVNXFLTGKOBC
     CreateDate: Sat Jul 16 20:11:15 2022
     UUID: 4213dbbc94f2520b0d75eeafe1b319c7
     Key: yes
     Cert: yes
     Key2: yes
     Cert2: yes
     Model: EVAL (1)
     CPU: 1
     MEM: 2048
     VDOM license:
       permanent: 2
       subscription: 0
     ```

To obtain the permanent VM trial license from FortiCare using the GUI:

1. When unlicensed, the FortiOS GUI allows you to download the permanent VM trial license from FortiCare with your FortiCare account credentials. In the FortiGate VM License page, for *How will you license this VM?*, select *Evaluation License*.



2. In the *Email* field, enter your FortiCare account email address.
3. In the *Password* field, enter your FortiCare account password.
4. Click *OK*. When a permanent VM trial license is applied, the FortiOS, the GUI shows a summary of the license limitations and allows you to upload a paid VM license.



To allow FortiManager to apply a license to an unlicensed FortiGate-VM instance:

1. Confirm that the FortiGate is unlicensed by running `get system status` in the FortiOS CLI. The following shows expected output for this command:

```
Version: FortiGate-VM64-AZURE v8.0.0,buildXXXX,220728 (interim)
...
Serial-Number: FGVMEVTN8UP4KIA6
License Status: Invalid
VM Resources: 1 CPU/1 allowed, 1945 MB RAM/2048 MB allowed
```

2. In the FortiOS CLI, configure the FortiManager as central management:

```
config system central-management
    set type fortimanager
    set fmg "<FortiManager IP address>"
end
```

3. In FortiManager, configure the VM license as Installing VM licenses on managed devices describes.

PF and VF SR-IOV driver and virtual SPU support

Physical function (PF) and virtual function (VF) PCI Passthrough and SR-IOV drivers in FortiGate guest VM are supported.

PF provides the ability for PCI Passthrough, but requires an entire Network Interface Card (NIC) for a VM. It can usually achieve greater performance than a Virtual Function (VF) based SR-IOV. PF is also expensive. While VF allows one NIC to be shared among multiple guests VMs, PF is allocated to one port on a VM.

The supported driver versions are:

Driver	Version	Hypervisor	PCI-Passthrough/SR-IOV	vSPU (In-guest DPDK)	Note
ixgbe	5.3.7	ESXi, KVM	Yes	Yes	
Ixgbevf	4.3.5				
i40e	2.12.6			Yes	
i40evf	3.6.15				Available in FortiOS 6.4.0 and earlier versions.
Iavf	4.5.3				Replaces i40evf in FortiOS 6.4.1 and later versions. Supports Intel E810-C 100G adapters.
Mlx5	24.04-0.6.6				Supports Nvidia ConnectX-5 and ConnectX-6 100G adapters.
Bnxt_en	1.10.1-216.0.416.1				Available in FortiOS 6.4.3 and later versions. Supports Broadcom P2100G 100G adapters.
Vmxnet3	1.4.16.0-k-NAPI	ESXi			The combination of VMware ESXi and NSX-T does not support virtual SPU (vSPU).
ICE	1.9.11	ESXi, KVM	Yes	Yes	Added support to Intel 25GbE E-810 card and its variants (E810-XXVDA2 and E810-XXVDA4)

Other hypervisors, such as Xen or Microsoft Hyper-V, may work with vSPU, although they are unverified.

All tools and software utilities for UEFI 1.X have been removed from 6.2.0 and later releases. Update to UEFI 2.x to use the UEFI tools or software utilities.

You perform the configuration to use PF or VF on the hypervisor, and do not configure it on the FortiGate.

To check what driver is being used on the FortiGate:

```
# diagnose hardware deviceinfo nic port2
Name:        port2
Driver:      i40e
Version:     2.4.10
Bus:         0000:03:00.0
Hwaddr:      3c:fd:fe:1e:98:02
Permanent Hwaddr:3c:fd:fe:1e:98:02
State:       up
Link:        up
Mtu:         1500
Supported:   auto 1000full 10000full
Advertised:  auto 1000full 10000full
Auto:        disabled
Rx packets:      0
Rx bytes:        0
Rx compressed:       0
...
```

Using OCI IMDSv2

OCI IMDSv2 offers increased security for accessing instance metadata compared to IMDSv1. IMDSv2 is used in OCI SDN connectors and on instance deployments with bootstrap metadata. When upgrading from previous FortiOS builds with legacy IMDSv1 endpoints, the endpoints will be updated to IMDSv2, and the same calls can be made.

The following use cases illustrate IMDSv2 support on the FortiGate-VM.

To configure the Oracle OCI instance to use IMDSv2:

1. In OCI, deploy an instance using IMDSv2 with bootstrap metadata. There are two methods to enable IMDSv2 :
   - Use the OCI command line to deploy an instance using `user-data`. This example uses a MIME file that contains the license and configuration, as well as a JSON file that specifies to disable V1 metadata.

     ```
     oci compute instance launch
     --availability-domain wwwl:US-ASHBURN-AD-1
     --compartment-id
     ocid1.tenancy.oc1..aaaaaaaaaaa3aaaaaaaaaaaaaaaaa7xxxxxxx54aaaaaa4xxxxxxxx55xxxa
     --display-name fos-byol-v6.4.6-b2290-emulated
     --image-id
     ocid1.image.oc1.iad.aaaaaaaa6xxx43xxxxxxxxx7aaaaaaaaaaaaaaaaaaaa3xxxxxxxxxxxxxxx
     --subnet-id
     ocid1.subnet.oc1.iad.aaaaaaaaxxxxxxxxx2xxxxxxxxxxxxxxxxxxxxx5aaa4xxxxxxxxxxxx42aaa
     --shape VM.Standard1.4
     --assign-public-ip true
     --user-data-file /home/oci/userdata/mime.txt
     ```

```
--ssh-authorized-keys-file /home/oci/userdata/myfirstkeypair.pub
--instance-options file://home/oci/scripts/metadatav2.json
```

```
root@mail:/home/oci/scripts# cat metadatav2.json
{
  "areLegacyImdsEndpointsDisabled": true
}
```

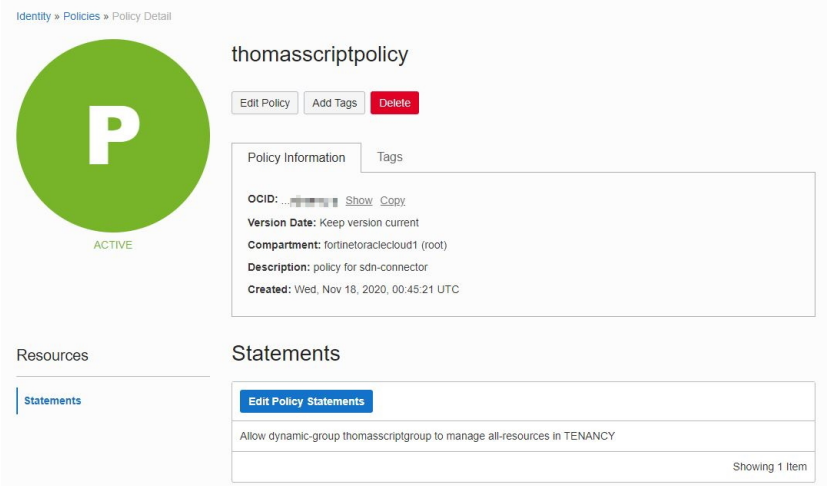
- While the instance is running, edit the instance metadata service version in the GUI ,and change the allowed IMDS version to *VERSION 2 ONLY* (see Getting Instance Metadata in the OCI documentation).

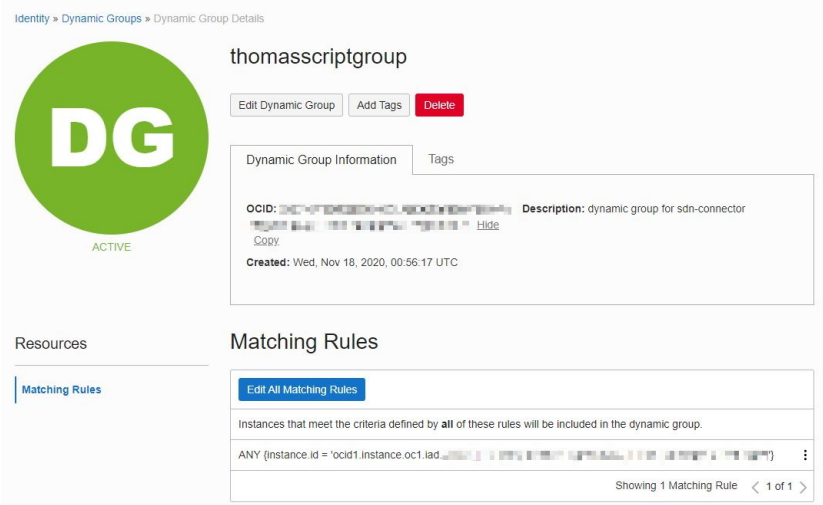
2. The FortiGate will use the metadata v2 endpoints to get the metadata bootstrap information. In FortiOS, verify this by running the following after bootup:

```
# diagnose debug cloudinit show
```

To configure an SDN connector with meta-IAM enabled and firewall addresses to obtain dynamic addresses:

1. Configure an IAM policy and dynamic group (see How Policies Work and Managing Dynamic Groups in the OCI documentation).





2. In FortiOS, configure the OCI Fabric connector (see OCI SDN connector using certificates on page 4656 for detailed instructions):
   a. Create the SDN connector.
   b. Verify that the OCI connector comes up (*Security Fabric > External Connectors* page indicates the status is up).
   c. Configure a dynamic firewall address with a filter.
   d. Verify the dynamic firewall address is resolved by the SDN connector.

To manually update the external IP:

```
# execute update-eip
instance: fos-byol-v6.4.6-b2290-emulated
    vnic0: fos-byol-v6.4.6-b2290-emulated
           10.0.0.58 (129.213.138.192)
port1: 10.0.0.58, eip: 129.213.138.192
EIP is updated successfully
```

To verify the OCI daemon debugs related to metadata:

```
# diagnose test application ocid 4
instance: fos-byol-v6.4.6-b2290-emulated
    vnic0: fos-byol-v6.4.6-b2290-emulated
           10.0.0.58
```

```
# diagnose test application ocid 5
Compartment Id:ocid1.tenancy.oc1..aaaaaaaaaaa3aaaaaaaaaaaaaaaaa7xxxxxxx54aaaaaa4xxxxxxxx55xxxa
Instance Id:ocid1.instance.oc1.iad.axxxxxxxxxxxxxxxxxxx4aaaaa5aaaaaaaaa4xxxxxxx2aaaaaaaa
Instance Name:fos-byol-v6.4.6-b2290-emulated
OCI Regarxiehlion:us-ashburn-1
```

```
# diagnose test application ocid 6
Instance Principal Token has been refreshed
```

FIPS cipher mode for AWS, Azure, OCI, and GCP FortiGate-VMs

AWS, Azure, OCI, and GCP FortiGate-VMs support FIPS cipher mode. You must remove all VPN configurations before you can enable FIPS CC mode.

FIPS cipher mode only allows a restricted set of ciphers for features that require encryption, such as SSH, IPsec and Agentless VPN, and HTTPS. You cannot use insecure protocols such as Telnet, TFTP, and HTTP to access the FortiGate-VM.

You must perform a factory reset to disable `fips-ciphers` mode.

To enable fips-cipher mode:

```
config system fips-cc
    set status fips-ciphers
end
Warning: entering fips-ciphers mode. To exit this mode, factory reset is required.
Do you want to continue? (y/n) y
```

FIPS-CC cipher mode is silently enabled when configured via cloud-init.

The following behavior occurs when you enable FIPS cipher mode:

- You can restore a license, image, configuration, and so on from an FTP server.
- The following options are available:

SSH algorithms	• aes128-gcm@openssh.com • aes256-gcm@openssh.com • hmac-sha2-256 • hmac-sha2-512
IKE/IPsec phase1 proposals	• aes128-sha256 • aes128-sha256 • aes128-sha384 • aes128-sha384 • aes128-sha512 • aes128-sha512 • aes128gcm-prfsha256 • aes128gcm-prfsha256 • aes128gcm-prfsha384 • aes128gcm-prfsha384 • aes128gcm-prfsha512 • aes128gcm-prfsha512 • aes256-sha256 • aes256-sha256

	• aes256-sha384 • aes256-sha384 • aes256-sha512 • aes256-sha512 • aes256gcm-prfsha256 • aes256gcm-prfsha256 • aes256gcm-prfsha384 • aes256gcm-prfsha384 • aes256gcm-prfsha512 • aes256gcm-prfsha512
IKE/IPsec phase2 proposals	• aes128-sha256 • aes128-sha256 • aes128-sha384 • aes128-sha384 • aes128-sha512 • aes128-sha512 • aes128gcm • aes128gcm • aes256-sha256 • aes256-sha256 • aes256-sha384 • aes256-sha384 • aes256-sha512 • aes256-sha512 • aes256gcm • aes256gcm
IKE/IPsec DH groups	• Default = 19, or any three from 14 - 21, 27 - 32
HTTPS for admin and Agentless VPN (with RSA server certificate) TLS suites	PFS: • TLS_AES_256_GCM_SHA384 • ECDHE-RSA-AES256-GCM-SHA384 • DHE-RSA-AES256-GCM-SHA384 • TLS_AES_128_GCM_SHA256 • ECDHE-RSA-AES128-GCM-SHA256 • DHE-RSA-AES128-GCM-SHA256 Elliptic curves: • prime256v1 • secp384r1 • secp521r1 DH group: • RFC3526/Oakley group 14 (2048 bits)

HTTPS for admin and Agentless VPN (with ECC server certificate) TLS suites	PFS: • TLS_AES_256_GCM_SHA384 • ECDHE-ECDSA-AES256-GCM-SHA384 • TLS_AES_128_GCM_SHA256 • ECDHE-ECDSA-AES128-GCM-SHA256 Elliptic curves: • prime256v1 • secp384r1 • secp521r1

- The FortiCare license is validated.
- FortiGuard databases and engines are updated.
- The DH-RSA-AES128-GCM-SHA256 and DH-RSA-AES256-GCM-SHA384 ciphers are not supported.

Cloud-init

You can use cloud-init to preconfigure a FortiGate-VM instance before bootup using a text file. For example, you can include a license or configuration information in the cloud-init file, so that the license and configuration is already present on the FortiGate-VM after initialization. All FortiGate-VM public and private cloud platforms support cloud-init. You can provide the cloud-init file when initializing the FortiGate-VM through the GUI of your desired cloud platform.

When providing FortiOS configuration in the cloud-init text file, you can include a full backed up FortiOS configuration or a partial configuration. For a partial configuration, provide the configuration in the form of CLI commands. The example in this topic includes a partial configuration that consists of the following CLI commands:

```
config system global
    set hostname mimecheck
    set admintimeout 480
end
```

```
config system admin
    edit admin
        set password 12345678
    next
end
```

The following shows the content of an example cloud-init MIME file that includes the FortiGate-VM license and some configuration. The example omits most of the license file content for security purposes:

```
Content-Type: multipart/mixed; boundary="===============0740947994048919689=="
MIME-Version: 1.0

--===============0740947994048919689==
Content-Type: text/plain; charset="us-ascii"
```

```
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment; filename="config"

config system global
set hostname mimecheck
set admintimeout 480
end
config system admin
edit admin
set password 12345678
end

----------------==0740947994048919689==
Content-Type: text/plain; charset="us-ascii"
MIME-Version: 1.0
Content-Transfer-Encoding: 7bit
Content-Disposition: attachment; filename="license"

-----BEGIN FGT VM LICENSE-----
QAAAAM9mmdw0/C5oquSAhgYYurEyOAeTChjuEt8NlvQZszsj6FMpzv9FFLl8DuV8
C7JSP1JqFSjTEeSLa/crtO84Df7gGQAAgZ3RwxjOeKPEgl4i4cQKpcECVcXM4hcb
...
uqoVJ7Nca1B4mZUE3v4BuO07fZZJCdO2
-----END FGT VM LICENSE-----

----------------==0740947994048919689==--
```

To debug the cloud-init configuration, use the `diagnose debug cloud-init show` command. The following shows example output for this command:

```
>> Checking metadata source config drive
>> Found config drive /dev/sr0
>> Successfully mount config drive
>> MIME parsed preconfig script
>> Found metadata source: config drive
>> Trying to install vmlicense ...
>> Run config script
>> FGVMULTM12345678 $
>> FGVMULTM12345678 $ config system global
>> FGVMULTM12345678 (global) $ set hostname vFGTvm00
>> FGVMULTM12345678 (global) $ end
>> vFGTvm00 $ config system admin
>> vFGTvm00 (admin) $ edit admin
>> vFGTvm00 (admin) $ set password 12345678
>> vFGTvm00 (admin) $ end
>> vFGTvm00 $
>> vFGTvm00 $ config system interface
>> vFGTvm00 (interface) $ edit port1
>> vFGTvm00 (port1) $ set mode static
>> vFGTvm00 (port1) $ set ip 10.6.30.169/24
>> vFGTvm00 (port1) $ set allowaccess ping https ssh snmp http telnet fgfm radius-acct probe-
response ftm
```

```
>> vFGTvm00 (port1) $ next
>> vFGTvm00 (interface) $ edit port2
>> vFGTvm00 (port2) $ set mode static
>> vFGTvm00 (port2) $ set ip 10.1.100.169/24
>> vFGTvm00 (port2) $ set allowaccess ping https ssh snmp http telnet fgfm radius-acct probe-
response ftm
>> vFGTvm00 (port2) $ next
>> vFGTvm00 (interface) $ edit port3
>> vFGTvm00 (port3) $ set mode static
>> vFGTvm00 (port3) $ set ip 172.16.200.169/24
>> vFGTvm00 (port3) $ set allowaccess ping https ssh snmp http telnet fgfm radius-acct probe-
response ftm
>> vFGTvm00 (port3) $ next
>> vFGTvm00 (interface) $ end
>> vFGTvm00 $
>> vFGTvm00 $ config firewall policy
>> vFGTvm00 (policy) $ edit 0
>> vFGTvm00 (0) $ set srcintf "port2"
>> vFGTvm00 (0) $ set dstintf "port3"
>> vFGTvm00 (0) $ set srcaddr "all"
>> vFGTvm00 (0) $ set dstaddr "all"
>> vFGTvm00 (0) $ set action accept
>> vFGTvm00 (0) $ set schedule "always"
>> vFGTvm00 (0) $ set service "ALL"
>> vFGTvm00 (0) $ set nat enable
>> vFGTvm00 (0) $ next
>> vFGTvm00 (policy) $ end
>> vFGTvm00 $
>> vFGTvm00 $ config router static
>> vFGTvm00 (static) $ edit 1
>> vFGTvm00 (1) $ set gateway 172.16.200.254
>> vFGTvm00 (1) $ set device "port3"
>> vFGTvm00 (1) $ next
>> The destination is set to 0.0.0.0/0 which means all IP addresses.
>> vFGTvm00 (static) $ end
>> vFGTvm00 $
>> Finish running config script
```

TPM support for FortiGate-VM

Using the TPM module, the FortiGate can generate, store, and authenticate cryptographic keys. When TPM is enabled on a FortiGate,a master-encryption-password is automatically generated to encrypt sensitive data on the FortiGate such as IPsec VPN preshared keys (PSK), and other passwords and keys as this document lists. In turn, a TPM-generated primary key, which is stored on the TPM, encrypts this master-encryption-password.

When the FortiGate backs up configurations to a configuration file, the master-encryption-password encrypts passwords and keys. The primary key also encrypts the master-encryption-password. Therefore, when restoring a config file, if the FortiGate unit does not have TPM enabled, or does not have the same master-encryption-key, you cannot upload the configuration file.

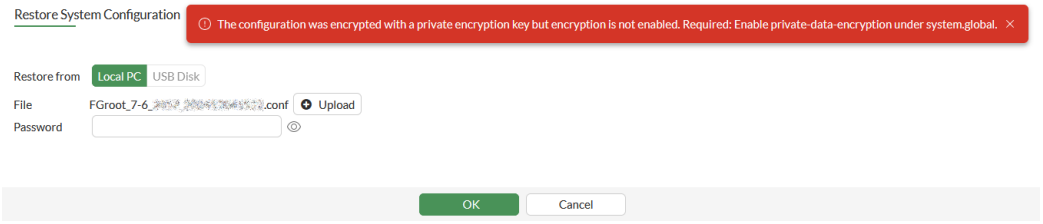
This enhancement adds TPM support to FGT-VM64 platforms. Hypervisors with software TPM emulator packages installed can support the TPM feature in FortiOS. This feature supports KVM/QEMU.

For information about TPM, see Trusted platform module support on page 3830.

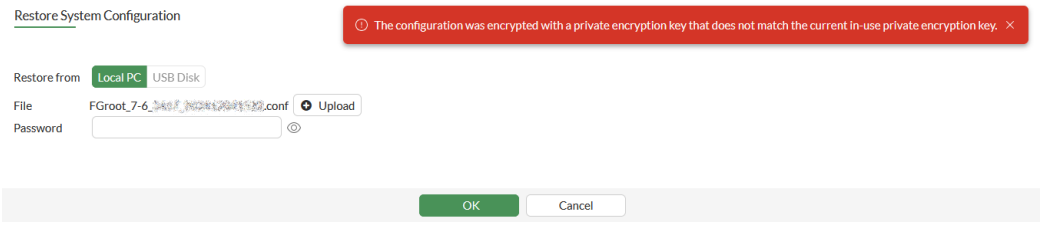
Passwords and keys that the masterencryptionkey can encrypt include:

- Alert email user password
- BGP and other routing-related configurations
- External resource
- FortiGuard proxy password
- FortiToken/FortiToken Mobile seed
- High availability password
- Link Monitor server-side password
- IPsec VPN PSK
- Local certificate private key
- SDN connector server-side password
- Local, LDAP, RADIUS, FSSO, and other user category-related passwords
- Modem/PPPoE
- NST password
- NTP Password
- SNMP
- Wireless security-related password

You cannot restore a private key-encrypted configuration via the FortiOS GUI if `private-data-encryption` is disabled. The following shows the GUI in this scenario:



Also, if a configuration backup was taken while the private-data-encryption is enabled, then private-data-encryption is disabled and re-enabled again, the configuration backup will no longer be able to be restored.



To check if your FortiGate has a TPM:

1. Verify that the required packages are installed on the Linux KVM host:

```
packet@kvm-s01:~$ lsb_release -a
No LSB modules are available.
Distributor ID:        Ubuntu
Description:   Ubuntu 22.04.1 LTS
Release:       22.04
Codename:      jammy
packet@kvm-s01:~$
packet@kvm-s01:~$ apt list swtpm swtpm-tools qemu libvirt0 virtinst
Listing... Done
libvirt0/jammy-updates,jammy-updates,now 8.0.0-1ubuntu7.1 amd64 [installed,automatic]
qemu/jammy-updates,jammy-updates,now 1:6.2+dfsg-2ubuntu6.3 amd64 [installed]
swtpm-tools/jammy,jammy,now 0.6.3-0ubuntu3 amd64 [installed]
swtpm/jammy,jammy,now 0.6.3-0ubuntu3 amd64 [installed]
virtinst/jammy,jammy,jammy,jammy,now 1:4.0.0-1 all [installed]
```

2. Import a FGT_VM64_KVM VM to the host. You may want to change the following script to fit your setup:

```
UUID="$(uuid)"
SKU="FGT_VM64_KVM"
VER=7
NUM=0418
CPU=2
RAM=2048
CONTROLLER="type=ide,index=0"
BUS="ide"
MODEL="virtio"
RND_MAC() { printf '90:6C:AC:%02X:%02X\n' $((RANDOM%256)) $((RANDOM%256)) ;}
MACADDR=$(RND_MAC)
DOMAIN=$SKU-v$VER-b$NUM

qemu-img create -f qcow2 $DOMAIN-log.qcow2 1024M
qemu-img create -f qcow2 $DOMAIN-wanopt.qcow2 1024M

virt-install --connect qemu:///system \
       --name $DOMAIN \
       --uuid $UUID \
       --virt-type kvm \
       --arch=x86_64 \
       --hvm \
       --osinfo linux \
       --os-variant=generic \
       --graphics vnc,listen=0.0.0.0 --noautoconsole \
       --cpu host-passthrough \
       --vcpus=$CPU \
       --ram $RAM \
       --sysinfo host \
       --controller $CONTROLLER \
       --boot hd,menu=on \
       --disk fortios.qcow2,device=disk,bus=$BUS,format=qcow2,cache=none,io=native \
```

```
        --disk $DOMAIN-log.qcow2,device=disk,bus=$BUS,format=qcow2,cache=none,io=native \
        --disk $DOMAIN-wanopt.qcow2,device=disk,bus=$BUS,format=qcow2,cache=none,io=native \
        --features kvm_hidden=on,smm=on \
        --tpm backend.type=emulator,backend.version=2.0,model=tpm-tis \
        --network bridge=br1,model=$MODEL,mac=$MACADDR:01 \
        --network bridge=br2,model=$MODEL,mac=$MACADDR:02 \
        --network bridge=br3,model=$MODEL,mac=$MACADDR:03 \
        --network bridge=br4,model=$MODEL,mac=$MACADDR:04 \
        --import
```

Key pairs are created on the host when the VM with TPM is imported:

```
packet@kvm-s01:~$ sudo ls -al /var/lib/swtpm-localca/
total 56
drwxr-x---  2 swtpm root  4096 Sep 21 08:09 .
drwxr-xr-x 49 root  root  4096 Sep 19 12:42 ..
-rwxr-xr-x  1 swtpm swtpm    0 Sep 21 08:09 .lock.swtpm-localca
-rw-r--r--  1 swtpm swtpm 5519 Sep 21 08:09 01.pem
-rw-r--r--  1 swtpm swtpm    1 Sep 21 08:19 certserial
-rw-r--r--  1 swtpm swtpm   48 Sep 21 08:09 index.txt
-rw-r--r--  1 swtpm swtpm   21 Sep 21 08:09 index.txt.attr
-rw-r--r--  1 swtpm swtpm    0 Sep 21 08:09 index.txt.old
-rw-r--r--  1 swtpm swtpm 5519 Sep 21 08:09 issuercert.pem
-rw-r--r--  1 swtpm swtpm    3 Sep 21 08:09 serial
-rw-r--r--  1 swtpm swtpm    3 Sep 21 08:09 serial.old
-rw-r-----  1 swtpm swtpm 2459 Sep 21 08:09 signkey.pem
-rw-r--r--  1 swtpm swtpm 1468 Sep 21 08:09 swtpm-localca-rootca-cert.pem
-rw-r-----  1 swtpm swtpm 2459 Sep 21 08:09 swtpm-localca-rootca-privkey.pem
packet@kvm-s01:~$
packet@kvm-s01:~$ sudo cat /var/log/swtpm/libvirt/qemu/FGT_VM64_KVM_v7.0.8_b0418-swtpm.log
Starting vTPM manufacturing as swtpm:swtpm @ Wed 21 Sep 2023 08:09:30 AM PDT
Successfully created RSA 2048 EK with handle 0x81010001.
  Invoking /usr/lib/x86_64-linux-gnu/swtpm/swtpm-localca --type ek --ek
b0a85bad0cb79ef673f05f4d3fdb4f65da3171d86a392e60435c18a431a3062aafaadb22e2af06b2522cfcf959ca33
4ba38684859beb8064f2ba610735cb1dccee1388b9da840a4732d626358e383f0d089592d04dfc15b7e82285f1fa1b
4a73bd1bfdbf0d75a02f94f069ae1546d2f28f984046f384f4b35ef1451a191628b2a1329f138dad4e4407d0d03b2f
71defc568642fe74d98f0e383e8ac1a5c94b4c30c1a0aae0cfe96bc9316397582cbbb834557a2112aad32d3f1e825e
8dfbd569bb9b2492728c425609515568f17d42aee8a5fdaf973a441aaf8bf20762101a9e2507ee0b4e876280e36474
b4c10179df18fe066db708d0c11e741a8e722154c9 --dir /var/lib/libvirt/swtpm/eb3c65cc-d354-11ea-
a7dc-08002799a4d5/tpm2 --logfile /var/log/swtpm/libvirt/qemu/FGT_VM64_KVM_v7.0.8_b0418-
swtpm.log --vmid FGT_VM64_KVM_v7.0.8_b0418:eb3c65cc-d354-11ea-a7dc-08002799a4d5 --tpm-spec-
family 2.0 --tpm-spec-level 0 --tpm-spec-revision 164 --tpm-manufacturer id:00001014 --tpm-
model swtpm --tpm-version id:20191023 --tpm2 --configfile /etc/swtpm-localca.conf --optsfile
/etc/swtpm-localca.options
Creating root CA and a local CA's signing key and issuer cert.
Successfully created EK certificate locally.
  Invoking /usr/lib/x86_64-linux-gnu/swtpm/swtpm-localca --type platform --ek
b0a85bad0cb79ef673f05f4d3fdb4f65da3171d86a392e60435c18a431a3062aafaadb22e2af06b2522cfcf959ca33
4ba38684859beb8064f2ba610735cb1dccee1388b9da840a4732d626358e383f0d089592d04dfc15b7e82285f1fa1b
4a73bd1bfdbf0d75a02f94f069ae1546d2f28f984046f384f4b35ef1451a191628b2a1329f138dad4e4407d0d03b2f
71defc568642fe74d98f0e383e8ac1a5c94b4c30c1a0aae0cfe96bc9316397582cbbb834557a2112aad32d3f1e825e
```

```
8dfbd569bb9b2492728c425609515568f17d42aee8a5fdaf973a441aaf8bf20762101a9e2507ee0b4e876280e36474
b4c10179df18fe066db708d0c11e741a8e722154c9 --dir /var/lib/libvirt/swtpm/eb3c65cc-d354-11ea-
a7dc-08002799a4d5/tpm2 --logfile /var/log/swtpm/libvirt/qemu/FGT_VM64_KVM_v7.0.8_b0418-
swtpm.log --vmid FGT_VM64_KVM_v7.0.8_b0418:eb3c65cc-d354-11ea-a7dc-08002799a4d5 --tpm-spec-
family 2.0 --tpm-spec-level 0 --tpm-spec-revision 164 --tpm-manufacturer id:00001014 --tpm-
model swtpm --tpm-version id:20191023 --tpm2 --configfile /etc/swtpm-localca.conf --optsfile
/etc/swtpm-localca.options
Successfully created platform certificate locally.
Successfully created NVRAM area 0x1c00002 for RSA 2048 EK certificate.
Successfully created NVRAM area 0x1c08000 for platform certificate.
Successfully created ECC EK with handle 0x81010016.
  Invoking /usr/lib/x86_64-linux-gnu/swtpm/swtpm-localca --type ek --ek
x=d28e9411dbe9aa0ada17c179c0854bebcf2d7ef2f94f42ef92f4e2deb28b568c9ecabd847fd36a974efceb7b0d54
893e,y=6b777ed060459c7907eb639665b3e64d9a93e692b7a4c0d20a18acafb6a2ae8e1284e948060266b96c1c23c
c883e7634,id=secp384r1 --dir /var/lib/libvirt/swtpm/eb3c65cc-d354-11ea-a7dc-08002799a4d5/tpm2
--logfile /var/log/swtpm/libvirt/qemu/FGT_VM64_KVM_v7.0.8_b0418-swtpm.log --vmid FGT_VM64_KVM_
v7.0.8_b0418:eb3c65cc-d354-11ea-a7dc-08002799a4d5 --tpm-spec-family 2.0 --tpm-spec-level 0 --
tpm-spec-revision 164 --tpm-manufacturer id:00001014 --tpm-model swtpm --tpm-version
id:20191023 --tpm2 --configfile /etc/swtpm-localca.conf --optsfile /etc/swtpm-localca.options
Successfully created EK certificate locally.
Successfully created NVRAM area 0x1c00016 for ECC EK certificate.
Successfully activated PCR banks sha1,sha256 among sha1,sha256,sha384,sha512.
Successfully authored TPM state.
Ending vTPM manufacturing @ Wed 21 Sep 2023 08:09:33 AM PDT
Starting vTPM manufacturing as swtpm:swtpm @ Wed 21 Sep 2023 08:19:44 AM PDT
Successfully created RSA 2048 EK with handle 0x81010001.
  Invoking /usr/lib/x86_64-linux-gnu/swtpm/swtpm-localca --type ek --ek
b49eb6d250c2add268fe448098b458f57e3a47719c3fbcc49fb85ecddd937f2f662a238eee0b8814ea3c07a4beeeba
d5a4ef30fd224e9051fad2ae29256ba7b85b03aef004ec05d2fd1e8139edcb3396b0b2b0a2adfb6b29fd975a9daf38
5aa3ffc0739fbc2d6b5850b9f424c787074ac56571fc15564b3dfbd847f2c79d310dfea27f2a694bb2c49d3bbb2e2d
2a61c29d4214140358dfe23b97562ea8c756da7942e8be3b260da9dfccb26383c4734c76d6e8e47e55055c1a697c13
79faf3b41400034b201115fb0913151f0a1d4b963208e5f758ad9c59ee1da145d2bc740069768545085d18a0010891
5214014b8b99fb47611f8b9260c70a4e2cef3ce1c7 --dir /var/lib/libvirt/swtpm/eb3c65cc-d354-11ea-
a7dc-08002799a4d5/tpm2 --logfile /var/log/swtpm/libvirt/qemu/FGT_VM64_KVM_v7.0.8_b0418-
swtpm.log --vmid FGT_VM64_KVM_v7.0.8_b0418:eb3c65cc-d354-11ea-a7dc-08002799a4d5 --tpm-spec-
family 2.0 --tpm-spec-level 0 --tpm-spec-revision 164 --tpm-manufacturer id:00001014 --tpm-
model swtpm --tpm-version id:20191023 --tpm2 --configfile /etc/swtpm-localca.conf --optsfile
/etc/swtpm-localca.options
Successfully created EK certificate locally.
  Invoking /usr/lib/x86_64-linux-gnu/swtpm/swtpm-localca --type platform --ek
b49eb6d250c2add268fe448098b458f57e3a47719c3fbcc49fb85ecddd937f2f662a238eee0b8814ea3c07a4beeeba
d5a4ef30fd224e9051fad2ae29256ba7b85b03aef004ec05d2fd1e8139edcb3396b0b2b0a2adfb6b29fd975a9daf38
5aa3ffc0739fbc2d6b5850b9f424c787074ac56571fc15564b3dfbd847f2c79d310dfea27f2a694bb2c49d3bbb2e2d
2a61c29d4214140358dfe23b97562ea8c756da7942e8be3b260da9dfccb26383c4734c76d6e8e47e55055c1a697c13
79faf3b41400034b201115fb0913151f0a1d4b963208e5f758ad9c59ee1da145d2bc740069768545085d18a0010891
5214014b8b99fb47611f8b9260c70a4e2cef3ce1c7 --dir /var/lib/libvirt/swtpm/eb3c65cc-d354-11ea-
a7dc-08002799a4d5/tpm2 --logfile /var/log/swtpm/libvirt/qemu/FGT_VM64_KVM_v7.0.8_b0418-
swtpm.log --vmid FGT_VM64_KVM_v7.0.8_b0418:eb3c65cc-d354-11ea-a7dc-08002799a4d5 --tpm-spec-
family 2.0 --tpm-spec-level 0 --tpm-spec-revision 164 --tpm-manufacturer id:00001014 --tpm-
model swtpm --tpm-version id:20191023 --tpm2 --configfile /etc/swtpm-localca.conf --optsfile
/etc/swtpm-localca.options
```

```
Successfully created platform certificate locally.
Successfully created NVRAM area 0x1c00002 for RSA 2048 EK certificate.
Successfully created NVRAM area 0x1c08000 for platform certificate.
Successfully created ECC EK with handle 0x81010016.
  Invoking /usr/lib/x86_64-linux-gnu/swtpm/swtpm-localca --type ek --ek
x=56a69f0827e7f4fc237dffb8202573f910140516ced4d85f62b443b627d6eb3075993a5e757119ed56ab43daa76e
5f23,y=c38364e2663bcb8cab92a658c2f4054826ca36d6cff99ea0a7a2ef9f600bf5902902482a67ad90101930ed7
f17cc613d,id=secp384r1 --dir /var/lib/libvirt/swtpm/eb3c65cc-d354-11ea-a7dc-08002799a4d5/tpm2
--logfile /var/log/swtpm/libvirt/qemu/FGT_VM64_KVM_v7.0.8_b0418-swtpm.log --vmid FGT_VM64_KVM_
v7.0.8_b0418:eb3c65cc-d354-11ea-a7dc-08002799a4d5 --tpm-spec-family 2.0 --tpm-spec-level 0 --
tpm-spec-revision 164 --tpm-manufacturer id:00001014 --tpm-model swtpm --tpm-version
id:20191023 --tpm2 --configfile /etc/swtpm-localca.conf --optsfile /etc/swtpm-localca.options
Successfully created EK certificate locally.
Successfully created NVRAM area 0x1c00016 for ECC EK certificate.
Successfully activated PCR banks sha1,sha256 among sha1,sha256,sha384,sha512.
Successfully authored TPM state.
Ending vTPM manufacturing @ Wed 21 Sep 2023 08:19:44 AM PDT
```

3. Log in to FGT_VM64_KVM and check TPM status:

```
FGT_VM64_KVM # diagnose hardware deviceinfo tpm

TPM capability information of fixed properties:
=======================================================
TPM_PT_FAMILY_INDICATOR:        2.0
TPM_PT_LEVEL:                   0
TPM_PT_REVISION:                164
TPM_PT_DAY_OF_YEAR:             75
TPM_PT_YEAR:                    2021
TPM_PT_MANUFACTURER:            IBM
TPM_PT_VENDOR_STRING:           SW   TPM
TPM_PT_VENDOR_STRING_1 in HEX:  0x53572020
TPM_PT_VENDOR_STRING_2 in HEX:  0x2054504d
TPM_PT_VENDOR_STRING_3 in HEX:  0x00000000
TPM_PT_VENDOR_STRING_4 in HEX:  0x00000000
TPM_PT_VENDOR_TPM_TYPE:         1
TPM_PT_FIRMWARE_VERSION:        8217.4131.22.13878
TPM_PT_FIRMWARE_VERSION in HEX: 0x2019102300163636

TPM_PT_MEMORY:
=======================================================
Shared RAM:                     0 CLEAR
Shared NV:                      1 SET
Object Copied To Ram:           1 SET

TPM_PT_PERMANENT:
=======================================================
Owner Auth Set:                 0 CLEAR
Sendorsement Auth Set:          0 CLEAR
Lockout Auth Set:               0 CLEAR
Disable Clear:                  0 CLEAR
```

```
In Lockout:                     0 CLEAR
TPM Generated EPS:              1 SET
```

```
FGT_VM64_KVM # diagnose tpm get-property

TPM capability information of fixed properties:
============================================================
TPM_PT_FAMILY_INDICATOR:        2.0
TPM_PT_LEVEL:                   0
TPM_PT_REVISION:                164
TPM_PT_DAY_OF_YEAR:             75
TPM_PT_YEAR:                    2021
TPM_PT_MANUFACTURER:            IBM
TPM_PT_VENDOR_STRING:           SW   TPM
TPM_PT_VENDOR_STRING_1 in HEX:  0x53572020
TPM_PT_VENDOR_STRING_2 in HEX:  0x2054504d
TPM_PT_VENDOR_STRING_3 in HEX:  0x00000000
TPM_PT_VENDOR_STRING_4 in HEX:  0x00000000
TPM_PT_VENDOR_TPM_TYPE:         1
TPM_PT_FIRMWARE_VERSION:        8217.4131.22.13878
TPM_PT_FIRMWARE_VERSION in HEX: 0x2019102300163636

TPM_PT_MEMORY:
============================================================
Shared RAM:                     0 CLEAR
Shared NV:                      1 SET
Object Copied To Ram:           1 SET

TPM_PT_PERMANENT:
============================================================
Owner Auth Set:                 0 CLEAR
Sendorsement Auth Set:          0 CLEAR
Lockout Auth Set:               0 CLEAR
Disable Clear:                  0 CLEAR
In Lockout:                     0 CLEAR
TPM Generated EPS:              1 SET
```

```
FGT_VM64_KVM # diagnose tpm get-var-property

TPM capability information of variable properties:

TPM_PT_STARTUP_CLEAR:
============================================================
Ph Enable:                      1 SET
Sh Enable:                      1 SET
Eh Enable:                      1 SET
Orderly:                        0 CLEAR
```

```
FGT_VM64_KVM # diagnose tpm read-clock

Clock info:
============================================================
```

```
Time since the last TPM_Init:
2375158 ms  =  0 y, 0 d, 0 h, 39 min, 35 s, 158 ms

Time during which the TPM has been powered:
2375319 ms  =  0 y, 0 d, 0 h, 39 min, 35 s, 319 ms

TPM Reset since the last TPM2_Clear:            5
Number of times that TPM2_Shutdown:             0
Safe:                                           1 = Yes
```

```
FGT_VM64_KVM # diagnose tpm shutdown-prepare

Shutdown works as expected.
```

```
FGT_VM64_KVM # diagnose tpm selftest

Successfully tested. Works as expected.
```

```
FGT_VM64_KVM # diagnose tpm generate-random-number

Random value:

0x00000000:   0x73  0xF1  0x9F  0x31
```

```
FGT_VM64_KVM # diagnose tpm SHA-1 1234567890abcdef1234567890abcdef
1234567890abcdef1234567890abcdef

TPM2_Hash of '1234567890abcdef1234567890abcdef' with SHA-1:

0x00000000:   62  0A  31  15  69  9A  42  2B
0x00000008:   D8  74  DE  31  D3  E6  91  1C
0x00000010:   58  3A  76  75
```

```
FGT_VM64_KVM # diagnose tpm SHA-256 1234567890abcdef1234567890abcdef
1234567890abcdef1234567890abcdef

TPM2_Hash of '1234567890abcdef1234567890abcdef' with SHA-256:

0x00000000:   C5  12  D9  2E  35  45  B2  F1
0x00000008:   22  2E  4B  4C  6A  F6  D3  30
0x00000010:   EC  30  02  A0  4B  CA  A4  1D
0x00000018:   F9  CC  2C  49  62  84  96  D6
```

4. Enable TPM and input the master encryption password. This is an example. Using 0123456789abcdef0123456789abcdef as your private key is not recommended:

```
FGT_VM64_KVM # execute private-encryption-key sample
Private encryption is not enabled.
Command fail. Return code 7
```

```
FGT_VM64_KVM # config system global
FGT_VM64_KVM (global) # set private-data-encryption enable
FGT_VM64_KVM (global) # end
This operation will generate a random private data encryption key!
Previous config files encrypted with the system default key cannot be restored after this
operation!
Do you want to continue? (y/n)y
Private data encryption key generation succeeded!
```

The following shows an example of a successful activation:

```
FGT_VM64_KVM # execute private-encryption-key sample
B64TEXT: u7oOx1iBjPFu4XLZVq5/RpoZrDJ9htRo6Jjhfts4BaI=
B64HMAC: FHmUhzSyT0IEfyoRnfdTFbY2l0o=
```

Note the B64TEXT and B64HMAC sample keys. Run the following to verify the feature:

```
FGT_VM64_KVM # execute private-encryption-key verify
u7oOx1iBjPFu4XLZVq5/RpoZrDJ9htRo6Jjhfts4BaI= FHmUhzSyT0IEfyoRnfdTFbY2l0o=
Verification passed.
```

5. Back up the config:

```
FGT_VM64_KVM # execute backup config tftp FGVM02TM12345678.conf 172.18.70.161
Please wait...
Connect to tftp server 172.18.70.161 ...
#
Send config file to tftp server OK.
```

6. Verify that the backup config has private-encryption-key:

```
packet@1804:/mnt/incoming$ less FGVM02TM12345678.conf
  #config-version=FGVMK6-7.0.8-FW-build0418-220920:opmode=0:vdom=0:user=admin
  #conf_file_ver=2079893748141389
  #buildno=0418
  #global_vdom=1
  #private-encryption-key=oY5GhQK3w0Ddn0EX+8hp6UYpjB4=
  config system global
      set admin-server-cert "Fortinet_Factory"
      set alias "FortiGate-VM64-KVM"
      set hostname "FGT_VM64_KVM"
      set private-data-encryption enable
      set timezone 04
  end
```

Hyperscale firewall

Hyperscale hardware and facilities can scale a distributed computing environment up to thousands of servers. It is about achieving massive scale in computing, typically for big data or cloud computing.

A hyperscale firewall secures hyperscale data centers and 5G networks.

See the Hyperscale Firewall Guide. For information about the difference between hyperscale and standard FortiOS CGNAT, see Hyperscale and standard FortiOS CGNAT feature comparison.

Troubleshooting

This section is intended for administrators with super_admin permissions who require assistance with basic and advanced troubleshooting. Administrators with other types of permissions may not be able to perform all of the tasks in this section.

This section contains the following troubleshooting topics:

- Troubleshooting methodologies on page 4936
- Connectivity Fault Management on page 4939
- Troubleshooting scenarios on page 4942
  - System date and time settings on page 4944
  - Checking the hardware connections on page 4945
  - Checking FortiOS network settings on page 4946
  - Checking CPU and memory resources on page 4948
  - FortiGuard server settings on page 5001
  - Troubleshooting high CPU usage on page 4951
  - Checking the modem status on page 4955
  - Running ping and traceroute on page 4956
  - Checking the logs on page 4961
  - Verifying routing table contents in NAT mode on page 4962
  - Verifying the correct route is being used on page 4962
  - Verifying the correct firewall policy is being used on page 4963
  - Checking the bridging information in transparent mode on page 4963
  - Checking wireless information on page 4965
  - Performing a sniffer trace or packet capture on page 4966
  - Debugging the packet flow on page 4968
  - Testing a proxy operation on page 4972
  - Displaying detail Hardware NIC information on page 4973
  - Performing a traffic trace on page 4978
  - Using a session table on page 4979
  - Finding object dependencies on page 4987
  - Diagnosing NPU-based interfaces on page 4988
  - Identifying the XAUI link used for a specific traffic stream on page 4990
  - Running the TAC report on page 4991
  - Using the process monitor on page 4991
  - Computing file hashes on page 4993
  - Other commands on page 4995
  - FortiGuard troubleshooting on page 4999
  - View open and in use ports on page 5004
  - IPS and AV engine version on page 5005
  - print tablesize on page 5006

- CLI troubleshooting cheat sheet on page 5007
- CLI error codes on page 5007
- Additional resources on page 5007

Troubleshooting methodologies

The sections in this topic provide an overview of how to prepare to troubleshoot problems in FortiGate. They include verifiying your user permissions, establishing a baseline, defining the problem, and creating a plan.

Verify user permissions

Before you begin troubleshooting, verify the following:

- You have administrator privileges for the FortiGate.
- The FortiGate is integrated into your network.
- The operation mode is configured.
- The system time, DNS settings, administrator password, and network interfaces are configured.
- Firmware, FortiGuard AntiVirus, FortiGuard Application Control, and FortiGuard IPS are up to date.

If you are using a FortiGate that has virtual domains (VDOMs) enabled, you can often troubleshoot within your own VDOM. However, you should inform the super_admin for the FortiGate that you will be performing troubleshooting tasks.

You may also need access to other networking equipment, such as switches, routers, and servers to carry out tests. If you do not have access to this equipment, contact your network administrator for assistance.

Establish a baseline

FortiGate operates at all layers of the OSI model. For this reason, troubleshooting can be complex. Establishing baseline parameters for your system before a problem occurs helps to reduce the complexity when you need to troubleshoot.

A best practice is to establish and record the normal operating status. Regular operation data shows trends, and allows you to see where changes occur when problems arise. You can gather this data by using logs and SNMP tools to monitor the system performance or by regularly running information gathering commands and saving the output.

You should back up your FortiOS configuration on a regular basis even when you are not troubleshooting. You can restore the backed up configuration as needed to save time recreating it from the factory default settings.

Use the following CLI commands to obtain normal operating data for a FortiGate:

`get system status`	Displays firmware versions and FortiGuard engine versions, and other system information.
`get system performance status`	Displays CPU and memory states, average network usage, average sessions and session setup rate, viruses caught, IPS attacks blocked, and uptime.
`get hardware memory`	Displays information about memory.
`get system session status`	Displays total number of sessions.
`get router info routing-table all`	Displays all the routes in the routing table, including their type, source, and other useful data.
`get ips session`	Displays memory used and maximum amount available to IPS as well as counts
`get webfilter ftgd-statistics`	Displays a list of FortiGuard related counts of status, errors, and other data.
`diagnose sys session list`	Displays the list of current detailed sessions.
`show sys dns`	Displays the configured DNS servers.
`diagnose sys ntp status`	Displays information about NTP servers.

You can run any commands that apply to your system for information gathering. For example, if you have active VPN connections, use the `get vpn` series of commands to get more information about them.

Use `execute tac report` to get an extensive snapshot of your system. This command runs many diagnostic commands for specific configurations. It also records the current state of each feature regardless of the features deployed on your FortiGate. If you need to troubleshoot later, you can run the same command again and compare the differences to identify any suspicious output.

Define the problem

The following questions are intended to compare the current behavior of the FortiGate with normal operations to help you define the problem. Be specific with your answers. After you define the problem, search for a solution in the troubleshooting scenarios section, and then create a plan to resolve it.

What is the problem?	The problem being observed may not be the actual problem. You should determine where the problem lies before starting to troubleshoot the FortiGate.
Was the device working before?	If the device never worked, it might be defective. For more information, see Troubleshooting your installation on page 52.
Can the problem be reproduced?	If the problem is intermittent, it may be dependent on system load. Intermittent problems are challenging to troubleshoot because they are difficult to reproduce.
What has changed?	Use the FortiGate event log to identify possible configuration changes.

	There may be changes in the operating environment. For example, there might be a gradual increase in load as more sites are forwarded through the firewall. If something has changed, roll back the change and assess the impact.
What is the scope of the problem?	After you isolate the problem, determine what applications, users, devices, and operating systems the problem affects. The following questions are intended to narrow the scope of the problem and identify what to check during troubleshooting. The more factors you can eliminate, the less you need to check. For this reason, be as specific and accurate as possible when gathering information. • What is not working? • Is more than one thing not working? • Is it partly working? If so, what parts are working? • Is it a connectivity issue for the entire device, or is there an application that isn't reaching the Internet? • Where did the problem occur? • When did the problem occur and to which users or groups of users? • What components are involved? • What applications are affected? • Can you use a packet sniffer to trace the problem? • Can you use system debugging or look in the session table to trace the problem? • Do any of the log files indicate a failure has occurred?

Create a troubleshooting plan

After you define the problem and its scope, develop a troubleshooting plan.

Create checklist	Make a list all the possible causes of the problem and how you can test for each cause. Create a checklist to keep track of what has been tried and what is left to test. Checklists are useful when more than one person is performing troubleshooting tasks.
Obtain the required equipment	Testing your solution may require additional networking equipment, computers, or other devices. Network administrators usually have additional networking equipment available to loan you, or a lab where you can bring the FortiGate unit to test. If you do not have access to equipment, check for shareware applications that can perform the same tasks. Often, there are software solutions you can use when hardware is too expensive.
Consult Fortinet troubleshooting resources	After the checklist is created, refer to the troubleshooting scenarios sections to assist with implementing your plan. See Troubleshooting scenarios on page 4942.

Gather information for technical support	If you still require technical assistance after the plan is implemented, be prepared to provide Fortinet technical support with following information: • Firmware build version (use the `get system status` command) • Network topology diagram • Recent configuration file • Recent debug log (optional) • Summary of troubleshooting steps you have taken and the results. Do not provide the output from the `execute tac` report unless the support team requests it. The output from this command is very large and is not required in many cases.
Contact technical support	Before contacting technical support, ensure you have login access (preferably with full read/write privileges) to all networking devices that could be relevant to troubleshooting. If you are using VMs, be prepared to have someone who can log in to the virtual hosting platform in case it is necessary to check and possibly modify resource allocation. For information about contacting technical support, go to FortiCare Support Service page.

Connectivity Fault Management

Some FortiGate hardware models support Connectivity Fault Management (CFM) technology. With CFM, administrators can easily diagnose and resolve issues in Ethernet networks. CFM provides tools for monitoring, testing, and verifying the connectivity and performance of network segments.

The following platforms support CFM:

FortiGate	FG-40F, FG-40F-3G4G, FG-60E, FG-60F, FG-61E, FG-61F, FG-80E, FG-80E-POE, FG-80F-POE, FG-81E, FG-81E-POE, FG-81F, FG-90E-POE, FG-100F, FG-101F, FG-200E, FG-200F, FG-1100E
FortiWiFi	FWF-40F, FWF-60E, FWF-60F, FWF-61E, FWF-61F

Use the `config ethernet-oam cfm` command to configure the CFM protocol.

```
config ethernet-oam cfm
    edit <domain-id>
        set domain-name <string>
        set domain-level <integer>
        config service
            edit <service-id>
                set service-name <string>
                set interface "<string>"
                set mepid <integer>
                set message-interval <integer>
```

```
                set cos <integer>
                set sender-id {none | Hostname}
            next
        end
    next
end
```

Command	Description
`<domain-id>`	Specify the domain ID for the Ethernet layer operation, administration, and management (OAM) protocol. A unique domain ID is used to communicate with other peers under the same domain ID and domain level.
`domain-level <integer>`	Specify the OAM maintenance level (0 to 7, with 0 being the smallest and 7 being the largest). A unique domain level is used to communicate with other devices under the same domain ID and domain level.
`domain-name <string>`	Specify the OAM domain name or maintenance domain identifier (MDID). Other peer devices recognize the domain name. All devices in the same domain with the same service level can communicate with each other.

A domain can provide multiple services. Each service uses a special service ID. The following items describe a service:

Command	Description
`<service-id>`	Specify the ID for the service.
`service-name <string>`	Specify the name of the service.
`interface <string>`	Specify the name of the VLAN interface where the service is enabled. The service is associated with a particular VLAN network port and can't be accessed by other network ports.
`mepid <integer>`	Specify the unique ID of the maintenance association endpoints (MEP) (1 - 8191). The service is associated with a unique MEP ID and can't respond to other service requests of a different MEP ID.
`message interval <integer>`	Specify the continuity-check message frequency interval in milliseconds. Determines how long to send a continuity-check message to determine whether the service is alive.
`cos <integer>`	Specify the class of service (COS) bit for continuity-check messages (0 to 7). CoS is an optional, special bit in the packet of continuity-check messages.
`sender-id {none | hostname}`	Specify the type, length, value (TLV) sender ID: • `none`: indicates no sender ID. • `hostname`: uses the Fortinet production name of the device as the sender ID, for example, FortiGate-80F. The sender ID is an optional column that includes a hostname in the packet of continuity-check messages.

The following diagnose commands can be used with this feature:

Command	Description
`diagnose ethernet-oam cfmpeer`	Locate peers configured with `config ethernet-oam cfm` that are using the CFM Continuity Check Protocol (CCP) protocol to connect to the CCP daemon (CCD).
`diagnose debug application cfmd {enable | disable}`	Enable or disable debugging messages of the CFM protocol. • `enable`: enable debugging messages for the CFM protocol. Messages appear on the console. • `disable`: disable debugging messages.

The following execute commands can be used with this feature:

Command	Description
`execute ethernet ping`	Check if an interface has a peer with mac address and level available under CFM support.
`execute ethernet traceroute`	Check the Ethernet traceroute with the peer FortiGate. The traceroute is instructed to achieve a peer through an interface with mac_address and level available under CFM support.

Example

In this example, an interface (vlan101) connects FortiGate 81F to FortiGate 101F. CFM is configured for the interface (vlan101) on the FortiGate 81F. All steps are performed on the FortiGate 101F.

Because this feature is based on IEEE 802.1Q, an IP address is not needed to connect the interface.

To configure and use CFM :

1. Configure CFM for the interface named `vlan101`:

```
config ethernet-oam cfm
    edit 1
        set domain-name cfm-test
        set domain-level 1
        config service
            edit 1
                set service-name vlan-101
                set interface "vlan101"
                set mepid 101
                set message-interval 10000
                set cos 7
                set sender-id Hostname
            next
        end
    next
end
```

2. On the FortiGate 101F, show the peers connecting to the device:

```
# diagnose ethernet-oam cfmpeer
wait for the responses from CCD daemons ...

========                MEPs (pid 11251)               ========
======== domain_name: cfm-test service_name: vlan-101 mepid: 101 ========
1  MAC = e0:23:ff:9b:07:0a, state = UP, mdlevel = 1, domain_name = cfm-test, service_name =
vlan-101, mepid = 81, TLV_port_status = PsUP, TLV_interface_status = isUp
========                     END                     ========
```

3. On FortiGate 101F, check whether the interface has a peer under CFM support:

```
# execute ethernet ping vlan101 1 5 e0:23:ff:9b:07:0a
Sending CFM LBM to e0:23:ff:9b:07:0a
64 bytes from e0:23:ff:9b:07:0a, sequence 422603820, 1 ms
64 bytes from e0:23:ff:9b:07:0a, sequence 422603821, 1 ms
64 bytes from e0:23:ff:9b:07:0a, sequence 422603822, 1 ms
64 bytes from e0:23:ff:9b:07:0a, sequence 422603823, 1 ms
64 bytes from e0:23:ff:9b:07:0a, sequence 422603824, 1 ms
```

4. Execute the Ethernet traceroute:

```
# execute ethernet traceroute vlan101 1 e0:23:ff:9b:07:0a
Sending CFM LTM probe to e0:23:ff:9b:07:0a
ethtrace_main: flags = 0, usefdbonly = 0
ttl 1: LTM with id 984984516
cfm_matchltr - 384
cfm_matchltr - 404
        reply from e0:23:ff:9b:07:0a, id=984984516, ttl=0, RlyHit
```

Troubleshooting scenarios

The following table is intended to help you diagnose common problems and provides links to the corresponding troubleshooting topics:

Problem	Probable cause	Recommended action
Hardware connections	• Are all of the cables and interfaces connected properly? • Is the LED for the interface green?	Checking the hardware connections on page 4945
FortiOS network settings	• If you are having problems connecting to the management interface, is your protocol enabled on the interface for administrative access? • Does the interface have an IP address?	Checking FortiOS network settings on page 4946

Problem	Probable cause	Recommended action
CPU and memory resources	• Is the CPU running at almost 100 percent usage? • Is your FortiGate running low on memory?	Checking CPU and memory resources on page 4948
Modem status	• Is the modem connected? • Are there PPP issues?	Checking the modem status on page 4955
Ping and traceroute	Is the FortiGate experiencing complete packet loss?	Running ping and traceroute on page 4956
Logs	Do you need to identify a problem?	Checking the logs on page 4961
Contents of the routing table (in NAT mode)	• Are there routes in the routing table for default and static routes? • Do all connected subnets have a route in the routing table? • Does a route have a higher priority than it should?	Verifying routing table contents in NAT mode on page 4962
Traffic routes	Is the traffic routed correctly?	Verifying the correct route is being used on page 4962
Firewall policies	Is the correct firewall policy applied to the expected traffic?	Verifying the correct firewall policy is being used on page 4963
Bridging information in transparent mode	Are you having problems in transparent mode?	Checking the bridging information in transparent mode on page 4963
Firewall session list	Are there active firewall sessions?	Using a session table on page 4979
Wireless Network	Is the wireless network working properly?	Checking wireless information on page 4965
FortiGuard connectivity	Is the FortiGate communicating properly with FortiGuard?	Verifying connectivity to FortiGuard on page 4999
Sniffer trace	• Is traffic entering the FortiGate? Does the traffic arrive on the expected interface? • Is the ARP resolution correct for the next-hop destination? • Is the traffic exiting the FortiGate to the destination as expected? • Is the FortiGate sending traffic back to the originator?	Performing a sniffer trace or packet capture on page 4966
Packet flow	Is traffic entering or leaving the FortiGate as expected?	Debugging the packet flow on page 4968

Problem	Probable cause	Recommended action
FortiGate is frozen or halted	The FortiGate may have experienced a kernel issue.	On supported models, use the NMI button to troubleshoot kernel issues.

System date and time settings

The system date and time are important for FortiGuard services, logging events, sending alerts, and certificates that have a time requirement to check for validity.

The wrong time makes the log entries confusing and difficult to use. If all devices have the same time, it helps to correlate log entries from different devices.

When possible, use Network Time Protocol (NTP) to set the date and time. This is an automatic method that does not require manual intervention. However, you must ensure that the port is allowed through the firewalls on your network. FortiToken synchronization requires NTP in many situations.

For information about setting the system date and time, see Setting the system time on page 3797.

To view and configure the date and time in the GUI:

1. Go to *Dashboard* > *Status*. The date and time are displayed in the *System Information* widget, next to *System Time*.
2. Go to *System* > *Settings* > *General*.
3. In the *System time* section, select *NTP*, and then configure the *Time zone*, and *Set time* settings as required.

To view the date and time in the CLI:

```
# execute date
current date is: 2024-07-23
```

```
# execute time
current time is: 14:17:00
last ntp sync:Tue Jul 23 13:25:21 2024
```

To configure the date and time in the CLI:

Use the `set timezone ?` command to display a list of timezones and the integers that represent them.

```
config system global
    set timezone <integer>
end
config system ntp
    set type custom
    config ntpserver
        edit 1
            set server “ntp1.fortinet.net”
        next
```

```
        edit 2
            set server “ntp2.fortinet.net”
        next
    end
    set ntpsync enable
    set syncinterval 60
end
```

Checking the hardware connections

If traffic is not flowing from the FortiGate, there may be a problem with the hardware connection.

To check hardware connections:

1. Ensure the network cables are plugged into the interfaces.
2. Verify the LED connection lights for the network cables indicate there is a connection. The lights are typically green when there is a connection.
3. Change the cable when:
   - The cable or its connector are damaged.
   - You are unsure of the type or quality of the cable, such as straight through or crossover.
   - You see exposed wires at the connector.
4. Connect the FortiGate to different hardware.
5. Go to *Network* > *Interfaces* to ensure the link status for the interface is set to *Up*.
   The link status is based on the physical connection and cannot be set in FortiOS.

To enable an interface in the GUI:

You should still perform basic software connectivity tests to ensure complete connectivity even if there was a problem with the hardware connection. The interface might also be disabled, or its *Status* might be set to *Down*.
See Interfaces on page 222.

1. Go to *Network* > *Interfaces*.
2. Select an interface, such as *Port1*, and click *Edit*.
3. In the *Miscellaneous* area, next to *Status*, click *Enabled*.
4. Click *OK*.

To enable an interface in the CLI:

```
config system interface
    edit port1
        set status up
    next
end
```

Checking FortiOS network settings

Check the FortiOS network settings if you have problems connecting to the management interface. FortiOS network settings include, interface settings, DNS Settings, and DHCP settings.

Interface settings

If you can access the FortiGate with the management cable only, you can view the interface settings in the GUI.

To view the interface settings in the GUI:

1. Go to *Network* > *Interfaces*.
2. Select an interface and click *Edit*.
3. Check the following interfaces to ensure they are not blocking traffic.

Setting	Description
Link Status	The status is *Up* when a valid cable is plugged in. The status is *Down* when an invalid cable is plugged in. The Link Status is shown physically by the connection LED for the interface. If the LED is green, the connection is good. If Link Status is *Down*, the interface does not work. Link status also appears in the *Network* > *Interfaces* page by default.
Addressing mode	Do not use *DHCP* if you do not have a DHCP server. You will not be able to log into an interface in DHCP mode as it will not have an IP address.
IP/Network Mask	An interface requires an IP address to connect to other devices. Ensure there is a valid IP address in this field. The one exception is when *DHCP* is enabled for this interface to get its IP address from an external DHCP server.
IPv6 address	The same protocol must be used by both ends to complete the connection. Ensure this interface and the remote connection are both using IPv4 or both are using IPv6 addresses.
Administrative access	If no protocols are selected, you will have to use the local management cable to connect to the unit. If you are using IPv6, configure the IPv6 administrative access protocols.
Status	Ensure the status is set to *Up* or the interface will not work.

To display the internal interface settings in the CLI:

```
FGT# show system interface <interface_name>
```

To view the list of possible interface settings:

```
config system interface
    edit <interface_name>
```

```
        get
end
```

DNS settings

To view DNS settings in the GUI:

Go to *Network* > *DNS*.

You can trace many networking problems back to DNS issues. Check the following items:

1. Are there values for both the *Primary DNS server* and *Secondary DNS server* fields.
2. Is the *Local Domain Name* correct?
3. Are you using IPv6 addressing? If so, are the IPv6 DNS settings correct?
4. Are you using Dynamic DNS (DDNS)? If so, is it using the correct server, credentials, and interface?
5. Can you contact both DNS servers to verify the servers are operational?
6. If an interface addressing mode is set to DHCP and is set to override the internal DNS, is that interface receiving a valid DNS entry from the DHCP server? Is it a reasonable address and can it be contacted to verify it is operational?
7. Are there any DENY security policies that need to allow DNS?
8. Can any internal device perform a successful traceroute to a location using the FQDN?

DHCP server settings

DHCP servers are common on internal and wireless networks. The DHCP server will cause problems if it is not configured correctly.

To view DHCP server settings in the GUI:

1. Go to *Network* > *Interfaces*.
2. Select an interface, and click *Edit*.

Check the following items:

1. Is the DHCP server enabled?
2. Is the DHCP server entry set to *Relay*? If so, verify there is another DHCP server to which requests can be relayed. Otherwise, set it to *Server*.
3. Does the DHCP server use a valid IP address range? Are other devices using the addresses? If one or more devices are using IP addresses in this range, you can use the IP reservation feature to ensure the DHCP server does not use these addresses. See DHCP servers and relays on page 550
4. Is there a gateway entry? If not, add a gateway entry to ensure that the server's clients have a default route.
5. Is the system DNS setting being used? A best practice is to avoid confusion by using the system DNS whenever possible. However, you can specify up to three custom DNS servers, and you should use all three entries for redundancy.

> ⚠ There are some situations, such as a new wireless interface, or during the initial FortiGate configuration, where interfaces override the system DNS entries. When this happens, it often shows up as intermittent Internet connectivity.
>
> To fix the problem, go to *Network* > *DNS*, and enable *Use FortiGuard Servers*.

Checking CPU and memory resources

Check the CPU and memory resources when the FortiGate is not working, the network is slow, or there is a reduced firewall session setup rate. All processes share the system resources in FortiOS, including CPU and memory.

To view system resources in the GUI:

Go to *Dashboard* > *Status*.

The resource information is located in the *CPU* and *Memory* widgets. For information, see Dashboard on page 142.

To view system resources in the CLI:

```
get system performance status
```

Sample output:

```
# get system performance status
CPU states: 0% user 0% system 0% nice 100% idle 0% iowait 0% irq 0% softirq
CPU0 states: 0% user 0% system 0% nice 100% idle 0% iowait 0% irq 0% softirq
CPU1 states: 0% user 0% system 0% nice 100% idle 0% iowait 0% irq 0% softirq
CPU2 states: 0% user 0% system 0% nice 100% idle 0% iowait 0% irq 0% softirq
CPU3 states: 0% user 0% system 0% nice 100% idle 0% iowait 0% irq 0% softirq
Memory: 4050332k total, 527148k used (13%), 3381312k free (83%), 141872k freeable (3%)
Average network usage: 41 / 28 kbps in 1 minute, 54 / 44 kbps in 10 minutes, 42 / 34 kbps in 30
minutes
Average sessions: 33 sessions in 1 minute, 48 sessions in 10 minutes, 38 sessions in 30 minutes
Average session setup rate: 0 sessions per second in last 1 minute, 0 sessions per second in last
10 minutes, 0 sessions per second in last 30 minutes
Virus caught: 0 total in 1 minute
IPS attacks blocked: 0 total in 1 minute
Uptime: 0 days,  22 hours,  59 minutes
```

The first line of the output shows the CPU usage by category:

```
CPU states: 0% user 0% system 0% nice 100% idle 0% iowait 0% irq 0% softirq
```

The second section of the output shows the memory usage:

```
Memory: 4050332k total, 527148k used (13%), 3381312k free (83%), 141872k freeable (3%)
```

Memory usage should not exceed 90%. Using too much memory prevents some processes from functioning properly. For example, if the system is running low on memory, antivirus scanning enters into *failopen* mode where it drops connections or bypasses the antivirus system.

Other lines of output, such as `average network usage`, `average session setup rate`, `viruses caught`, and `IPS attacks blocked`, help determine why system resource usage is high.

For example:

- A high `average network usage` may indicate high traffic processing on the FortiGate,
- A very low or zero, `average session setup rate` may indicate the proxy is overloaded and unable to do its job.

Troubleshooting CPU and network resources

FortiGate has stopped working

If the FortiGate has stopped working, the first line of the output will look similar to this:

```
CPU states: 0% user 0% system 0% nice 100% idle
```

Network is slow

If your network is running slow, the first line of the output will look similar to this:

```
CPU states: 1% user 98% system 0% nice 1% idle
```

This example shows that all of the CPU is being used by system processes, and the FortiGate is overloaded. When overloading occurs, it is possible a process such as `scanunitid` is using all the resources to scan traffic. In this case you need to reduce the amount of traffic being scanned by blocking unwanted protocols, configuring more security policies to limit scanning to certain protocols, or similar actions.

It is also possible a hacker has accessed your network and is overloading it with malicious activity, such as running a spam server or using zombie PCs to attack other networks on the Internet.

You can use the following command to investigate the problem with the CPU:

```
get system performance top
```

This command shows all of the top processes that are running on the FortiGate and their CPU usage. The process names are on the left. If a process is using most of the CPU cycles, investigate it to determine whether the activity is normal.

Reduced firewall session setup rate

A reduced firewall session setup rate can be caused by a lack of system resources on the FortiGate, or reaching the session count limit for a VDOM.

As a best practice, administrators should record the session setup rate during normal operation to establish a baseline to help define a problem when your are troubleshooting.

The session setup rate appears in the `average sessions` section of the output.

A reduced firewall session setup rate will look similar to this:

```
Average sessions: 80 sessions in 1 minute, 30 sessions in 10 minutes, 42 sessions in 30 minutes

Average session setup rate: 3 sessions per second in last 1 minute, 0 sessions per second in last
10 minutes, 0 sessions per second in last 30 minutes
```

In the example above, there were 80 sessions in 1 minute, or an average of 3 sessions per second.

The values for `10 minutes` and `30 minutes` allow you to take a longer average for a more reliable value if your FortiGate is working at maximum capacity. The smallest FortiGate can have 1,000 sessions established per second across the unit.

> The session setup rate is a global command. If you have multiple VDOMs configured with many sessions in each VDOM, the session setup rate per VDOM will be slower than if there are no VDOMs configured.

High memory usage

As with any system, a FortiGate has limited hardware resources, such as memory, and all processes running on the FortiGate share the memory. Each process uses more or less memory, depending on its workload. For example, a process usually uses more memory in high traffic situations. If some processes use all of the available memory, other processes will not be able to run.

When high memory usage occurs, the services may freeze up, connections may be lost, or new connections may be refused.

If you see high memory usage in the *Memory* widget, the FotiGate may be handling high traffic volumes. Alternatively, the FortiGate may have problems with connection pool limits that are affecting a single proxy. If the FortiGate receives large volumes of traffic on a specific proxy, the unit may exceed the connection pool limit. If the number of free connections within a proxy connection pool reaches zero, issues may occur.

To view current memory usage information in the CLI:

```
diagnose hardware sysinfo memory
```

Sample output:

```
total: used: free: shared: buffers: cached: shm:
Mem: 2074185728 756936704 1317249024 0 20701184 194555904 161046528
Swap:            0            0            0
MemTotal:    2025572 kB
MemFree:      1286376 kB
MemShared:       0 kB
Buffers:        20216 kB
Cached:          189996 kB
SwapCached:      0 kB
Active:          56644 kB
Inactive:   153648 kB
HighTotal:       0 kB
HighFree:          0 kB
LowTotal:    2025572 kB
LowFree:    1286376 kB
```

```
SwapTotal:        0 kB
SwapFree:         0 kB
```

Troubleshooting high CPU usage

Connection-related problems may occur when FortiGate's CPU resources are over extended. This occurs when you deploy too many FortiOS features at the same time.

Examples of CPU intensive features:

- VPN high-level encryption
- Intensive scanning of all traffic
- Logging all traffic and packets
- Dashboard widgets that frequently perform data updates

For information on customizing the CPU use threshold, see Execute a CLI script based on memory and CPU thresholds on page 4571.

Determining the current level of CPU usage

You can view CPU usage levels in the GUI or CLI. For precise usage values for both overall usage and specific processes, use the CLI.

To view CPU usage in the GUI:

Go to *Dashboard* > *Status*. Real-time CPU usage information is located in the *CPU* widget.

To view CPU usage in the CLI:

- Show top processes information:

```
diagnose sys top
```

- Show top threads information:

```
diagnose sys top-all
```

Sample output:

```
Run Time: 86 days, 0 hours and 10 minutes
        0U, 0N, 0S, 100I, 0WA, 0HI, 0SI, 0ST; 3040T, 2437F
                bcm.user 93 S < 3.1 0.4
                httpsd 18922 S 1.5 0.5
                httpsd 19150 S 0.3 0.5
                newcli 20195 R 0.1 0.1
                cmdbsvr 115 S 0.0 0.8
                pyfcgid 20107 S 0.0 0.6
```

```
                forticron 146 S 0.0 0.5
                httpsd 139 S 0.0 0.5
                cw_acd 166 S 0.0 0.5
                miglogd 136 S 0.0 0.5
                pyfcgid 20110 S 0.0 0.4
                pyfcgid 20111 S 0.0 0.4
                pyfcgid 20109 S 0.0 0.4
                httpsd 20192 S 0.0 0.4
                miglogd 174 S 0.0 0.4
                miglogd 175 S 0.0 0.4
                fgfmd 165 S 0.0 0.3
                newcli 20191 S 0.0 0.3
                initXXXXXXXXXXX 1 S 0.0 0.3
                httpsd 184 s 0.0 0.3
```

The following table explains the codes in the second line of the output:

Code	Description
U	Percentage of user space applications that are currently using the CPU
N	Percentage of time that the CPU spent on low priority processes since the last shutdown
S	Percentage of system processes (or kernel processes) that are using the CPU
I	Percentage of idle CPU resources
WA	Percentage of time that the CPU spent waiting on IO peripherals since the last shutdown
HI	Percentage of time that the CPU spent handling hardware interrupt routines since the last shutdown
SI	Percentage of time that the CPU spent handling software interrupt routines since the last shutdown
ST	Steal time: Percentage of time a virtual CPU waits for the physical CPU when the hypervisor is servicing another virtual processor
T	Total FortiOS system memory in MB
F	Free memory in MB

Each additional line of the command output displays information specific to processes or threads that are running on the FortiGate unit. For example, the sixth line of the output is: `newcli 20195 R 0.1 0.1`

The following table describes the data in the sixth line of the output:

Item	Description
newcli	The process (or thread) name. Duplicate process or thread names indicate that separate instances of that process or thread are running.
20195	The process or thread ID, which can be any number.
R	Current state of the process or thread. The process or thread state can be:

Item	Description
	• R - running • S - sleep • Z - zombie • D- disk sleep
0.1	The percentage of CPU capacity that the process or thread is using. CPU usage can range from 0.0 for a process or thread that is sleeping to higher values for a process or thread that's taking a lot of CPU time.
0.1	The amount of memory that the process or thread is using. Memory usage can range from 0.1 to 5.5 and higher.

You can use the following single-key commands when running `diagnose sys top` or `diagnose sys top-all`:

- `q` to quit and return to the normal CLI prompt.
- `p` to sort the processes by the amount of CPU that the processes are using.
- `m` to sort the processes by the amount of memory that the processes are using.

The output only displays the top processes or threads that are running. For example, if 20 are listed, they are the top 20 currently running, sorted by either CPU or memory usage. You can configure the number of processes or threads displayed, using the following CLI commands:

```
diagnose sys top <integer_seconds> <integer_maximum_lines>
```

```
diagnose sys top-all <integer_seconds> <integer_maximum_lines>
```

Where:

- `<integer_seconds>` is the delay in seconds (default is 5)
- `<integer_maximum_lines>` is the maximum number of lines (or processes) to list (default is 20)

Determining which features are using the most CPU resources

You can use the CLI to view the top few processes that are currently running and using the most CPU resources.

To view processes using the most CPU resources:

```
get system performance top
```

The entries at the top are using the most CPU resources. The second column from the right shows CPU usage by percentage. Note which processes are using the most resources and try to reduce their CPU load.

Processes you will see include:

- `ipsengine`: the IPS engine that scans traffic for intrusions
- `scanunitd`: antivirus scanner
- `httpsd`: secure HTTP
- `iked`: internet key exchange (IKE) in use with IPsec VPN tunnels
- `newcli`: active whenever you're accessing the CLI

- `sshd`: there are active secure socket connections
- `cmdbsrv`: the command database server application

Go to the features that are at the top of the list and look for evidence of CPU overuse. Generally, the monitor for a feature is a good place to start.

Checking for unnecessary CPU “wasters”

These are some best practices that will reduce your CPU usage, even if the FortiGate is not experiencing high CPU usage. Note that if the following information instructs you to turn off a feature that you require, disregard that part of the instructions.

- Use hardware acceleration wherever possible to offload tasks from the CPU. Offloading tasks, such as encryption, frees up the CPU for other tasks.
- Schedule antivirus, IPS, and firmware updates during off-peak hours. These updates do not usually consume CPU resources but they can disrupt normal operation.
- Check the log levels and which events are being logged. This is the severity of the messages that are recorded. Consider going up one level to reduce the amount of logging. Also, if there are events you do not need to monitor, remove them from the list.
- Log to FortiCloud instead of logging to memory or disk. Logging to memory quickly uses up resources and logging to local disk impacts overall performance and reduces the lifetime of the unit.

  Fortinet recommends logging to FortiCloud to avoid using too much CPU.
- If the disk is almost full, transfer the logs or data off the disk to free up space. When a disk is almost full it consumes a lot of resources to find free space and organize the files.
- If packet logging is enabled on the FortiGate, consider disabling it. When packet logging is enabled, it records every packet that comes through that policy.
- Halt all sniffers and traces.
- Ensure the FortiGate isn't scanning traffic twice. Traffic does not need to be rescanned if it enters the FortiGate on one interface, goes out another, and then comes back in again. Doing so is a waste of resources. However, ensure that traffic truly is being scanned once.
- Reduce the session timers to close unused sessions faster. Enter the following CLI commands, which reduce the default values. Note that, by default, the system adds 10 seconds to `tcp-timewait`.

```
config system global
    set tcp-halfclose-timer 30
    set tcp-halfopen-timer 30
    set tcp-timewait-timer 0
    set udp-idle-timer 60
end
```

- Go to *System* > *Feature Visibility*, and enable only features that you need.

SNMP monitoring

When CPU usage is under control, use SNMP to monitor CPU usage. Alternatively, use logging to record CPU and memory usage every 5 minutes.

Once the system is back to normal, you should set up a warning system that sends alerts when CPU resources are used excessively. A common method to do this is using SNMP. SNMP monitors many values in FortiOS and allows you to set high water marks that generate events. You can run an application on your computer to watch for and record these events.

To enable SNMP:

1. Go to *System* > *SNMP*.
2. Configure an SNMP community.

See SNMP on page 3877.

You can use the *System Resources* widget to record CPU usage if SNMP is too complicated. However, the widget only records problems as they happen and will not send you alerts for problems.

Checking the modem status

You can use the CLI to troubleshoot a modem that is not working properly, or troubleshoot a FortiGate that does not detect the modem.

To diagnose modem issues in the CLI:

```
diagnose sys modem {cmd | com | detect | history | external-modem | query| reset}
```

You should always run the following command after you connect a USB modem to FortiGate:

```
diagnose sys modem detect
```

Use the following command to view the modem's configuration, vendor and custom product identification number:

```
get system modem
```

Use the following commands to resolve connectivity issues:

- `diagnose debug enable`: Activates the debug on the console
- `diagnose debug application modemd`: Dumps communication between the modem and the unit.
- `diagnose debug application ppp`: Dumps the PPP negotiating messages.
- `execute modem dial`: Displays modem debug output.

The modem diagnose output should not contain errors when initializing. You should also verify the number used to dial into your ISP.

Running ping and traceroute

Ping and traceroute are useful tools in network troubleshooting. Alone, either tool can determine network connectivity between two points. However, ping can be used to generate simple network traffic that you can view using `diagnose` commands in FortiGate. This combination can be very powerful when you are trying to locate network problems.

Ping and traceroute can also tell you if your computer or network device has access to a domain name server (DNS). Both tools can use IP addresses or device domain names to determine why particular services, such as email or web browsing, may not work properly.

If ping does not work, it may be disabled on at least one of the interface settings and security policies for that interface.

Both ping and traceroute require particular ports to be open on firewalls to function. Since you typically use these tools to troubleshoot, you can allow them in the security policies and on interfaces only when you need them. Otherwise, keep the ports disabled for added security.

Ping

The ping command sends a very small packet to a destination, and waits for a response. The response has a timer that expires when the destination is unreachable.

Ping is part of layer 3 on the OSI Networking Model. Ping sends Internet Control Message Protocol (ICMP) “echo request” packets to the destination, and listens for “echo response” packets in reply. However, many public networks block ICMP packets because ping can be used in a denial of service (DoS) attack (such as Ping of Death or a smurf attack), or by an attacker to find active locations on the network. By default, FortiGate units have ping enabled while broadcast-forward is disabled on the external interface.

What ping can tell you

Beyond the basic connectivity information, ping can tell you the amount of packet loss (if any), how long it takes the packet to make the round trip, and the variation in that time from packet to packet.

If packet loss is detected, you should investigate the following:

- Possible ECMP, split horizon, or network loops.
- Cabling, to ensure there are no loose connections.
- Verify which security policy was used. To do this:
  Go to *Policy & Objects* > *Firewall Policy* and view the packet count column.

If there is total packet loss, you should investigate the following:

1. Ensure cabling is correct, and all equipment between the two locations is accounted for.
2. Ensure all IP addresses and routing information along the route is configured as expected.
3. Ensure all firewalls, including FortiGate security policies allow PING to pass through.

FortiGate ping options

Use the following CLI to view all the FortiGate ping options:

```
execute ping-options ?
adaptive-ping     Adaptive ping <enable|disable>.
data-size         Integer value to specify datagram size in bytes.
df-bit            Set DF bit in IP header <yes | no>.
interface         Auto | <outgoing interface>.
interval          Integer value to specify seconds between two pings.
pattern           Hex format of pattern, e.g. 00ffaabb.
repeat-count      Integer value to specify how many times to repeat PING.
reset             Reset settings.
source            Auto | <source interface IP>.
timeout           Integer value to specify timeout in seconds.
tos               IP type-of-service option.
ttl               Integer value to specify time-to-live.
use-sdwan         Use SD-WAN rules to get output interface <yes | no>.
validate-reply    Validate reply data <yes | no>.
view-settings     View the current settings for PING option.
```

`adaptive-ping {enable | disable}`	Enable or disable adaptive ping. FortiGate sends the next packet after the last response is received.
`data-size <integer>`	Specify the size of data in bytes. The size excludes ICMP header information. The data size allows you to control the ICMP datagram size to test the effects of different packet sizes on the connection. Enter an integer value from <0> to <65507>.
`df-bit {yes | no}`	Don't fragment bit. Prevent or allow the ICMP packet to fragment: • `yes`: Prevent the ICMP packet from being fragmented. • `no`: Allow the ICMP packet to be fragmented.
`interface {auto | <outgoing interface>}`	Specify the outgoing interface: • `auto`: Automatically select the interface based on the destination and routing table. • `<outgoing interface>`: Manually specify the interface to be used for ping. When `source` is specified, the IP address will be used as the ping source IP.
`interval <integer>`	Specify the time between pings in seconds. Enter an integer value from <1> to <2147483647>.
`pattern <string>`	Specify a pattern in hex format, for example, 00ffaabb, to be used in the data section of the ICMP packet.
`repeat-count <integer>`	Specify how many times to repeat the ping attempt. Enter an integer value from <1> to <2147483647>.
`reset`	Reset ping options to default values.
`source {auto | <source interface IP>}`	Specify the source IP address to use for sending out ping: • `auto`: Selects the primary IP address of the source interface. • `<source interface ip>`: Specify a source IP address for the interface used.
`timeout <integer>`	Specify in seconds how long to wait until the ping times out.

	Enter an integer value from <0> to <2147483647>.
`tos`	Set the type of service (ToS) field in the packet header to indicate the desired quality of service: • `default`: Defaults to 0. • `lowcost`: Minimize the cost. • `lowdelay`: Minimize the delay. • `reliability`: Maximize reliability. • `throughput`: Maximize throughput.
`ttl <integer>`	Specify the time to live. Time to live is the number of hops the ping packet should be allowed to make before being discarded or returned. Enter an integer value from <1> to <1255>.
`use-sdwan {yes | no}`	Specify whether to use SD-WAN rules and policy routes. • `yes`: The ping follows SD-WAN rules and policy routes. Usually used with other options, such as source, to match a specific SD-WAN rule that is based on a specific source address. • `no`: Do not follow SD-WAN rules and policy routes.
`validate-reply {yes | no}`	Specify whether to validate reply data: • `yes`: Validate reply data. • `no`: Do not validate reply data.
`view-settings`	Display the current ping option settings.

How to use ping

Ping syntax is the same for nearly every type of system on a network.

To ping from a FortiGate unit:

1. Go to *Dashboad*, and connect to the CLI through either telnet or the CLI widget.
2. Enter `execute ping 10.11.101.101` to send 5 ping packets to the destination IP address.
   To modify ping options, first apply your changes using the command `execute ping-options <option> <setting>`.

```
# execute ping 10.11.101.101
PING 10.11.101.101 (10.11.101.101): 56 data bytes
64 bytes from 10.11.101.101: icmp_seq=0 ttl=255 time=0.3 ms
64 bytes from 10.11.101.101: icmp_seq=1 ttl=255 time=0.2 ms
64 bytes from 10.11.101.101: icmp_seq=2 ttl=255 time=0.2 ms
64 bytes from 10.11.101.101: icmp_seq=3 ttl=255 time=0.2 ms
64 bytes from 10.11.101.101: icmp_seq=4 ttl=255 time=0.2 ms

--- 10.11.101.101 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 0.2/0.2/0.3 ms
```

3. Enter the domain name to test name resolution.

```
# execute ping google.com
PING google.com (142.250.179.78): 56 data bytes
```

To ping from a Microsoft Windows PC:

1. Open a command window.
2. Enter `ping 10.11.101.100` to ping the default internal interface of the FortiGate with four packets. Other options include:
   - `-t` to send packets until you press `Ctrl+C`
   - `-a` to resolve addresses to domain names where possible
   - `-n X` to send X ping packets and stop

```
C:\>ping 10.11.101.101

Pinging 10.11.101.101 with 32 bytes of data:
Reply from 10.11.101.101: bytes=32 time=10ms TTL=255
Reply from 10.11.101.101: bytes=32 time<1ms TTL=255
Reply from 10.11.101.101: bytes=32 time=1ms TTL=255
Reply from 10.11.101.101: bytes=32 time=1ms TTL=255

Ping statistics for 10.11.101.101:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 10ms, Average = 3ms
```

To ping from a Linux PC:

1. Go to a shell prompt.
2. Enter "`ping 10.11.101.101`".

Traceroute

Where ping will only tell you if it reached its destination and returned successfully, traceroute shows each step of the journey to its destination and how long each step takes. If ping finds an outage between two points, you can use traceroute to locate exactly where the problem is.

Traceroute works by sending ICMP packets to test each hop along the route. It sends three packets, and then increases the time to live (TTL) setting by one each time. This effectively allows the packets to go one hop farther along the route. This is why most traceroute commands display their maximum hop count before they start tracing the route, which is the maximum number of steps it takes before it declares the destination unreachable. Also, the TTL setting may result in steps along the route timing out due to slow responses. There are many possible reasons for this to occur.

By default, traceroute uses UDP datagrams with destination ports numbered from 33434 to 33534. The traceroute utility may also offer the option to select use of ICMP echo request (type 8) instead, which the Windows tracert utility uses. If you must, allow both protocols inbound through the FortiGate security policies (UDP with ports from 33434 to 33534 and ICMP type 8).

To track traceroute packets in the GUI:

Go to *Policy & Objects* > *Firewall Policy* and view the packet count column.

This allows you to verify the connection and confirm which security policy the traceroute packets are using.

What traceroute can tell you

Both ping and traceroute verify connectivity between two points. However, only traceroute shows you each step in the connection path. Also, ping and traceroute use different protocols and ports, so one may succeed where the other fails.

You can verify your DNS connection using traceroute. If you enter an FQDN instead of an IP address for the traceroute, DNS tries to resolve that domain name. If the name isn't resolved, you have DNS issues.

Using traceroute

The traceroute command varies slightly between operating systems. In Microsoft Windows, the command name is shortened to "`tracert`". Also, your output lists different domain names and IP addresses along your route.

To use traceroute on a Microsoft Windows PC:

1. Open a command window.
2. Enter `tracert fortinet.com` to trace the route from the PC to the Fortinet web site.

```
C:\>tracert fortinet.com
Tracing route to fortinet.com [208.70.202.225]
over a maximum of 30 hops:
1 <1 ms <1 ms <1 ms 172.20.120.2
2 66 ms 24 ms 31 ms 209-87-254-xxx.storm.ca [209.87.254.221]
3 52 ms 22 ms 18 ms core-2-g0-0-1104.storm.ca [209.87.239.129]
4 43 ms 36 ms 27 ms core-3-g0-0-1185.storm.ca [209.87.239.222]
5 46 ms 21 ms 16 ms te3-x.1156.mpd01.cogentco.com [38.104.158.69]
6 25 ms 45 ms 53 ms te8-7.mpd01.cogentco.com [154.54.27.249]
7 89 ms 70 ms 36 ms te3-x.mpd01.cogentco.com [154.54.6.206]
8 55 ms 77 ms 58 ms sl-st30-chi-.sprintlink.net [144.232.9.69]
9 53 ms 58 ms 46 ms sl-0-3-3-x.sprintlink.net [144.232.19.181]
10 82 ms 90 ms 75 ms sl-x-12-0-1.sprintlink.net [144.232.20.61]
11 122 ms 123 ms 132 ms sl-0-x-0-3.sprintlink.net [144.232.18.150]
12 129 ms 119 ms 139 ms 144.232.20.7
13 172 ms 164 ms 243 ms sl-321313-0.sprintlink.net [144.223.243.58]
14 99 ms 94 ms 93 ms 203.78.181.18
15 108 ms 102 ms 89 ms 203.78.176.2
16 98 ms 95 ms 97 ms 208.70.202.225
```

The first column on the left is the hop count, which cannot exceed 30 hops. When that number is reached, the traceroute ends.

The second, third, and fourth columns display how much time each of the three packets takes to reach this stage of the route. These values are in milliseconds and normally vary quite a bit. Typically a value of `<1ms` indicates a local connection.

The fifth column (farthest to the right) shows the domain name of the device and its IP address, or possibly only the IP address.

To perform a traceroute on a Linux PC:

1. Go to a command line prompt.
2. Enter "`traceroute fortinet.com`".
   The Linux traceroute output is very similar to the Windows `tracert` output.

To trace a route from a FortiGate to a destination IP address in the CLI:

```
# execute traceroute www.fortinet.com

traceroute to www.fortinet.com (66.171.121.34), 32 hops max, 84 byte packets
1 172.20.120.2 0.637 ms 0.653 ms 0.279 ms
2 209.87.254.221 <static-209-87-254-221.storm.ca> 2.448 ms 2.519 ms 2.458 ms
3 209.87.239.129 <core-2-g0-2.storm.ca> 2.917 ms 2.828 ms 9.324 ms
4 209.87.239.199 <core-3-bdi1739.storm.ca> 13.248 ms 12.401 ms 13.009 ms
5 216.66.41.113 <v502.core1.tor1.he.net> 17.181 ms 12.422 ms 12.268 ms
6 184.105.80.9 <100ge1-2.core1.nyc4.he.net> 21.355 ms 21.518 ms 21.597 ms
7 198.32.118.41 <ny-paix-gni.twgate.net> 83.297 ms 84.416 ms 83.782 ms
8 203.160.228.217 <217-228-160-203.TWGATE-IP.twgate.net> 82.579 ms 82.187 ms 82.066 ms
9 203.160.228.229 <229-228-160-203.TWGATE-IP.twgate.net> 82.055 ms 82.455 ms 81.808 ms
10 203.78.181.2 82.262 ms 81.572 ms 82.015 ms
11 203.78.186.70 83.283 ms 83.243 ms 83.293 ms
12 66.171.127.177 84.030 ms 84.229 ms 83.550 ms
13 66.171.121.34 <www.fortinet.com> 84.023 ms 83.903 ms 84.032 ms
14 66.171.121.34 <www.fortinet.com> 83.874 ms 84.084 ms 83.810 ms
```

Checking the logs

A log message records the traffic passing through FortiGate to your network and the action FortiGate takes when it scans the traffic. You should log as much information as possible when you first configure FortiOS. If FortiGate logs are too large, you can turn off or scale back the logging for features that are not in use.

It is difficult to troubleshoot logs without a baseline. Before you can determine if the logs indicate a problem, you need to know what logs result from normal operation.

When troubleshooting with log files

- Compare current logs to a recorded baseline of normal operation.
- If you need to, increase the level of logging (such as from Warning to Information) to obtain more information.
  When increasing logging levels, ensure that you configure email alerts and select both disk usage and log quota. This ensures that you will be notified if the increase in logging causes problems.

To configure the log settings in the GUI:

Go to *Log & Report* > *Log Settings*.

Determine the activities that generate the most log entries:

- Check all logs to ensure important information is not overlooked.
- Filter or order log entries based on different fields, such as level, service, or IP address, to look for patterns that may indicate a specific problem, such as frequent blocked connections on a specific port for all IP addresses.

Logs can help identify and locate any problems, but they do not solve them. The purpose of logs is to speed up your problem solving and save you time and effort.

For more information about logging and log reports, see Log and Report on page 4750.

Verifying routing table contents in NAT mode

Verify the contents of the routing table when a FortiGate has limited or no connectivity.

The routing table stores the routes currently in use for both static and dynamic protocols. Storing a route in the routing table saves time and resources performing a lookup. To ensure the most recently used routes remain in the table, old routes are bumped to make room for new ones. You cannot perform this task when FortiGate is in transparent mode.

If FortiGate is running in NAT mode, verify that all desired routes are in the routing table, including local subnets, default routes, specific static routes, and dynamic routing protocols.

To view the routing table in the CLI:

```
get router info routing-table all
```

Sample output:

```
FGT# get router info routing-table all
Codes:
    K - kernel, C - connected, S - static, R - RIP, B - BGP
           O - OSPF, IA - OSPF inter area
           N1 - OSPF NSSA external type 1, N2 - OSPF NSSA external type 2
           E1 - OSPF external type 1, E2 - OSPF external type 2
           i - IS-IS, L1 - IS-IS level-1, L2 - IS-IS level-2, ia - IS-IS inter area
           * - candidate default
S*      0.0.0.0/0 [10/0] via 172.20.120.2, wan1
C       10.31.101.0/24 is directly connected, internal
C       172.20.120.0/24 is directly connected, wan1
```

Verifying the correct route is being used

Run a trace route from a machine in the local area network (LAN) to ensure traffic is flowing as expected through the correct route when there is more than one default route.

In the following example output:

- The first hop contains the IP address `10.10.1.99`, which is the internal interface of the FortiGate.
- The second hop contains the IP address `172.20.120.2`, to which the wan1 interface of the FortiGate is connected.

This means the route through the wan1 interface is being used for this traffic.

```
C:\>tracert www.fortinet.com
Tracing route to www.fortinet.com [66.171.121.34]
over a maximum of 30 hops:
  1    <1 ms    <1 ms    <1 ms  10.10.1.99
  2     1 ms    <1 ms    <1 ms  172.20.120.2
  3     3 ms     3 ms     3 ms  static-209-87-254-221.storm.ca [209.87.254.221]
  4     3 ms     3 ms     3 ms  core-2-g0-2.storm.ca [209.87.239.129]
  5    13 ms    13 ms    13 ms  core-3-bdi1739.storm.ca [209.87.239.199]
  6    12 ms    19 ms    11 ms  v502.core1.tor1.he.net [216.66.41.113]
  7    22 ms    22 ms    21 ms  100ge1-2.core1.nyc4.he.net [184.105.80.9]
  8    84 ms    84 ms    84 ms  ny-paix-gni.twgate.net [198.32.118.41]
  9    82 ms    84 ms    82 ms  217-228-160-203.TWGATE-IP.twgate.net [203.160.228.217]
 10    82 ms    81 ms    82 ms  229-228-160-203.TWGATE-IP.twgate.net [203.160.228.229]
 11    82 ms    82 ms    82 ms  203.78.181.2
 12    84 ms    83 ms    83 ms  203.78.186.70
 13    84 ms     *       85 ms  66.171.127.177
 14    84 ms    84 ms    84 ms  fortinet.com [66.171.121.34]
 15    84 ms    84 ms    83 ms  fortinet.com [66.171.121.34]
```

You can also see the route taken for each session by debugging the packet flow in the CLI. For more information, see Debugging the packet flow on page 4968.

Verifying the correct firewall policy is being used

If you have more than one firewall policy, you can check which policy is being used in the *Policy & Objects* module in the GUI.

To verify the firewall policy in the GUI:

1. Go to *Policy & Objects* > *Firewall Policy*.
2. Look in the *Count* column to see which policy is being used. The count must show traffic increasing.

Debugging the packet flow in the CLI shows the policy ID that's allowing the traffic. For information, see Debugging the packet flow on page 4968.

Checking the bridging information in transparent mode

Checking the bridging information is useful when you are experiencing connectivity problems. When FortiGate is set to transparent mode, it acts like a bridge and sends all incoming traffic out on the other interfaces. Each bridge is a link between interfaces.

When traffic is flowing between the interfaces, you can see the bridges listed in the CLI. If no bridges are listed, this is the likely cause of the connectivity issue. When investigating bridging information, check for the MAC address of the interface or device in question.

How to check the bridging information

To view the list of bridge instances in the CLI:

```
diagnose netlink brctl list
```

Sample output:

```
#diagnose netlink brctl list
list bridge information
1. root.b fdb: size=256  used=6    num=7    depth=2    simple=no
Total 1 bridges
```

How to display forwarding domain information

You can use forwarding domains, or collision domains, in routing to limit where packets are forwarded on the network. Layer 2 broadcasts are limited to the same group. By default, all interfaces are in group 0. For example, if the FortiGate has 12 interfaces, only two may be in the same forwarding domain, which limits packets that are broadcast to those two interfaces. This reduces traffic on the rest of the network.

Collision domains prevent the forwarding of ARP packets to all VLANs on an interface. Without collision domains, duplicate MAC addresses on VLANs may cause ARP packets to be duplicated. Duplicate ARP packets can cause some switches to reset. It's important to know what interfaces are part of which forwarding domains because this determines which interfaces can communicate with each other.

To manually configure forwarding domains in transparent mode in the CLI:

```
config system interface
    edit <interface_name>
        set forward-domain <integer>
    next
end
```

To display the forward domains information in the CLI:

```
diagnose netlink brctl domain <name> <id>
```

Where `<name>` is the name of the forwarding domain to display and `<id>` is the domain ID.

Sample output:

```
diagnose netlink brctl domain ione 101
show bridge root.b ione forward domain.
```

```
id=101 dev=trunk_1 6
```

To list the existing bridge MAC table in the CLI:

```
diagnose netlink brctl name host <name>
```

Sample output:

```
show bridge control interface root.b host.
fdb: size=256, used=6, num=7, depth=2, simple=no
Bridge root.b host table
```

port no	device	devname	mac addr	ttl	attributes
2	7	wan2	02:09:0f:78:69:00	0	Local Static
5	6	vlan_1	02:09:0f:78:69:01	0	Local Static
3	8	dmz	02:09:0f:78:69:01	0	Local Static
4	9	internal	02:09:0f:78:69:02	0	Local Static
3	8	dmz	00:80:c8:39:87:5a	194	
4	9	internal	02:09:0f:78:67:68	8	
1	3	wan1	00:09:0f:78:69:fe	0	Local Static

To list the existing bridge port list in the CLI:

```
diagnose netlink brctl name port <name>
```

Sample output:

```
show bridge root.b data port.
trunk_1  peer_dev=0
internal  peer_dev=0
dmz  peer_dev=0
wan2  peer_dev=0
wan1  peer_dev=0
```

Checking wireless information

Check wireless connections, stations, and interfaces when the problem is not caused by a physical interface.

Troubleshooting station connection issues

To check if a station entry is created on access control in the CLI:

```
FG600B3909600253 # diagnose wireless-controller wlac -d sta
* vf=0 wtp=70 rId=2 wlan=open ip=0.0.0.0 mac=00:09:0f:db:c4:03 rssi=0 idle=148 bw=0 use=2
  vf=0 wtp=70 rId=2 wlan=open ip=172.30.32.122 mac=00:25:9c:e0:47:88 rssi=-40 idle=0 bw=9 use=2
```

Enabling diagnostics for a specific station

This example uses the station MAC address to find where it is failing:

```
FG600B3909600253 # diagnose wireless-controller wlac sta_filter 00:25:9c:e0:47:88 1
Set filter sta 00:25:9c:e0:47:88 level 1
FG600B3909600253 # 71419.245 <ih> IEEE 802.11 mgmt::disassoc <== 00:25:9c:e0:47:88 vap open rId 1
wId 0 00:09:0f:db:c4:03
71419.246 <dc> STA del 00:25:9c:e0:47:88 vap open rId 1 wId 0
71419.246 <cc> STA_CFG_REQ(34) sta 00:25:9c:e0:47:88 del ==> ws (0-192.168.35.1:5246) rId 1 wId 0
71419.246 <cc> STA del 00:25:9c:e0:47:88 vap open ws (0-192.168.35.1:5246) rId 1 wId 0
00:09:0f:db:c4:03 sec open reason I2C_STA_DEL
71419.247 <cc> STA_CFG_RESP(34) 00:25:9c:e0:47:88 <== ws (0-192.168.35.1:5246) rc 0 (Success).
```

Performing a sniffer trace or packet capture

When you troubleshoot networks and routing in particular, it helps to look inside the headers of packets to determine if they are traveling the route that you expect them to take. Packet sniffing is also known as network tap, packet capture, or logic analyzing.

System event logs are generated when a packet capture is started or stopped in the GUI, and when a sniffer operation is started or stopped in the CLI.

For more information on controlling GUI packet captures in the CLI, see Using the packet capture tool on page 1025.

> ⚠ For FortiGates with NP2, NP4, or NP6 interfaces that are offloading traffic, disable offloading on these interfaces before you perform a trace or it will change the sniffer trace.

Sniffing packets

To perform a sniffer trace in the CLI:

Before you start sniffing packets, you should prepare to capture the output to a file. A large amount of data may scroll by and you will not be able to see it without saving it first. One method is to use a terminal program like PuTTY to connect to the FortiGate CLI. Once the packet sniffing count is reached, you can end the session and analyze the output in the file.

The general form of the internal FortiOS packet sniffer command is:

```
# diagnose sniffer packet <interface_name> <'filter'> <verbose> <count> <tsformat>
```

To stop the sniffer, type CTRL+C.

<interface_name>	The name of the interface to sniff, such as port1 or internal. This can also be any to sniff all interfaces.
<'filter'>	What to look for in the information the sniffer reads. none indicates no filtering, and all packets are displayed as the other arguments indicate. The filter must be inside single quotes (‘).
<verbose>	The level of verbosity as one of: • 1 - print header of packets • 2 - print header and data from IP of packets • 3 - print header and data from Ethernet of packets • 4 - print header of packets with interface name • 5 - print header and data from IP of packets with interface name • 6 - print header and data from Ethernet of packets with interface name
<count>	The number of packets the sniffer reads before stopping. If you don't put a number here, the sniffer will run until you stop it with <CTRL+C>.
<tsformat>	The timestamp format. • a: absolute UTC time, yyyy-mm-dd hh:mm:ss.ms • l: absolute LOCAL time, yyyy-mm-dd hh:mm:ss.ms • otherwise: relative to the start of sniffing, ss.ms

Simple sniffing example:

```
# diagnose sniffer packet port1 none 1 3.
```

This displays the next three packets on the port1 interface using no filtering, and verbose level 1. At this verbosity level, you can see the source IP and port, the destination IP and port, action (such as ack), and sequence numbers.

In the output below, port 443 indicates these are HTTPS packets and that 172.20.120.17 is both sending and receiving traffic.

```
Head_Office_620b # diagnose sniffer packet port1 none 1 3
interfaces=[port1]
filters=[none]
0.545306 172.20.120.17.52989 -> 172.20.120.141.443: psh 3177924955 ack 1854307757
0.545963 172.20.120.141.443 -> 172.20.120.17.52989: psh 1854307757 ack 3177925808
0.562409 172.20.120.17.52988 -> 172.20.120.141.443: psh 4225311614 ack 3314279933
```

Using packet capture in a firewall policy

FortiGate can capture packets matching a firewall policy. You can enable capture-packet in the firewall policy.

To use packet capture, the FortiGate must have a disk and logging must be enabled in the firewall policy.

For information about using the packet capture tool in the GUI, see Using the packet capture tool on page 1025.

To enable packet capture in a policy in the GUI:

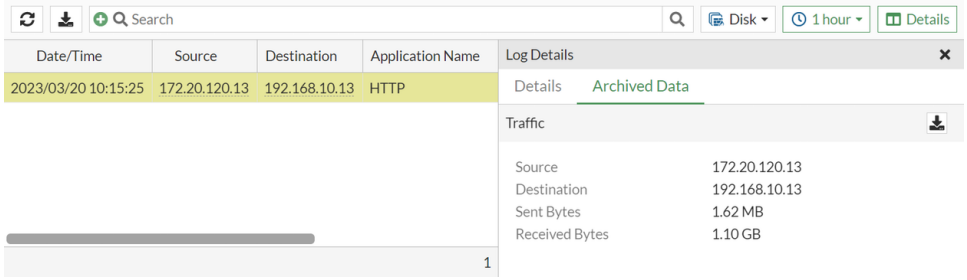
1. Go to *Policy & Objects* > *Firewall Policy* and click *Create New*.
2. Enter a name for the policy and configure the required settings.
3. Enable *Log Allowed Traffic* and select *Security Events* or *All Sessions*.
4. Enable *Capture Packets*.
5. Click *OK*.

To enable packet capture in a policy in the CLI:

```
config firewall policy
    edit <id>
        set action accept
        set logtraffic {all | utm}
        set capture-packet enable
    next
end
```

To view the packet capture:

1. Go to *Log & Report* > *Forward Traffic* and select the log that matches the firewall policy.
2. Select *Details* > *Archived Data* and click on the download button.



3. Open the downloaded PCAP file in a packet analyzer tool, such as Wireshark.

Debugging the packet flow

Debug the packet flow when network traffic is not entering and leaving the FortiGate as expected. When debugging the packet flow in the CLI, each command configures a part of the debug action. The final command starts the debug.

For information about using the debug flow tool in the GUI, see Using the debug flow tool on page 1032.

To trace the packet flow in the CLI:

```
# diagnose debug flow trace start
```

To follow packet flow by setting a flow filter:

```
# diagnose debug flow {filter | filter6} <option>
```

- Enter `filter` if your network uses IPv4.
- Enter `filter6` if your network uses IPv6.

Replace `<option>` with one of the following variables:

Variable	Description
`addr`	IPv4 or IPv6 address
`clear`	clear filter
`daddr`	destination IPv4 or IPv6 address
`dport`	destination port
`negate`	inverse IPv4 or IPv6 filter
`port`	port
`proto`	protocol number
`saddr`	source address
`sport`	source port
`vd`	index of virtual domain; `-1` matches all

If FortiGate is connected to FortiAnalyzer or FortiCloud, the diagnose debug flow output will be recorded as event log messages and then sent to the devices. Do not run this command longer than necessary, as it generates a significant amount of data.

Flow monitoring does not work for traffic offloaded to NP6 or NP7 processors. To use the diagnose debug flow commands with sessions offloaded to NP6 or NP7 processors you can test the traffic flow using ICMP (ICMP traffic is not offloaded) or you can disable NP6 or NP7 offloading.

You can use the following command to temporarily disable NP6 offloading of all traffic:

```
diagnose npu {np6 | np6xlite | np6lite} fastpath disable
```

You must disable NP7 offloading in the firewall policy that accepts the traffic that you are tracing, see Tracing packet flow on FortiGates with NP7 processors.

You can also use the NP7 packet sniffer to sniff NP7 offloaded traffic without disabling NP7 offloading, see NP7 packet sniffer.

To start flow monitoring with a specific number of packets:

```
# diagnose debug flow trace start <N>
```

To stop flow tracing at any time:

```
# diagnose debug flow trace stop
```

The following example shows the flow trace for a device with an IP address of `203.160.224.97`:

```
# diagnose debug enable
# diagnose debug flow filter addr 203.160.224.97
# diagnose debug flow show function-name enable
# diagnose debug flow trace start 100
```

Sample output: HTTP

To observe the debug flow trace, connect to the website at the following address:

```
https://www.fortinet.com
```

Comment: SYN packet received:

```
id=20085 trace_id=209 func=resolve_ip_tuple_fast
line=2700 msg="vd-root received a packet(proto=6,
192.168.3.221:1487->203.160.224.97:80) from port5."
```

SYN sent and a new session is allocated:

```
id=20085 trace_id=209 func=resolve_ip_tuple line=2799
msg="allocate a new session-00000e90"
```

Lookup for next-hop gateway address:

```
id=20085 trace_id=209 func=vf_ip4_route_input line=1543
msg="find a route: gw-192.168.11.254 via port6"
```

Source NAT, lookup next available port:

```
id=20085 trace_id=209 func=get_new_addr line=1219
msg="find SNAT: IP-192.168.11.59, port-31925"
direction"
```

Matched security policy. Check to see which policy this session matches:

```
id=20085 trace_id=209 func=fw_forward_handler line=317
msg="Allowed by Policy-3: SNAT"
```

Apply source NAT:

```
id=20085 trace_id=209 func=fw_forward_handler line=317
msg="Allowed by Policy-3: SNAT"
```

SYN ACK received:

```
id=20085 trace_id=210 func=resolve_ip_tuple_fast line=2700
msg="vd-root received a packet(proto=6, 203.160.224.97:80-
>192.168.11.59:31925) from port6."
```

Found existing session ID. Identified as the reply direction:

```
id=20085 trace_id=210 func=resolve_ip_tuple_fast line=2727
msg="Find an existing session, id-00000e90, reply direction"
```

Apply destination NAT to inverse source NAT action:

```
id=20085 trace_id=210 func=__ip_session_run_tuple
line=1516 msg="DNAT 192.168.11.59:31925-
>192.168.3.221:1487"
```

Lookup for next-hop gateway address for reply traffic:

```
id=20085 trace_id=210 func=vf_ip4_route_input line=1543
msg="find a route: gw-192.168.3.221 via port5"
```

ACK received:

```
id=20085 trace_id=211 func=resolve_ip_tuple_fast line=2700
msg="vd-root received a packet(proto=6,
192.168.3.221:1487->203.160.224.97:80) from port5."
```

Match existing session in the original direction:

```
id=20085 trace_id=211 func=resolve_ip_tuple_fast line=2727
msg="Find an existing session, id-00000e90, original
direction"
```

Apply source NAT:

```
id=20085 trace_id=211 func=__ip_session_run_tuple
line=1502 msg="SNAT 192.168.3.221->192.168.11.59:31925"
```

Receive data from client:

```
id=20085 trace_id=212 func=resolve_ip_tuple_fast
line=2700 msg="vd-root received a packet(proto=6,
192.168.3.221:1487->203.160.224.97:80) from port5."
```

Match existing session in the original direction:

```
id=20085 trace_id=212 func=resolve_ip_tuple_fast
line=2727 msg="Find an existing session, id-00000e90,
original direction"
```

Apply source NAT:

```
id=20085 trace_id=212 func=__ip_session_run_tuple
line=1502 msg="SNAT 192.168.3.221->192.168.11.59:31925"
```

Receive data from server:

```
id=20085 trace_id=213 func=resolve_ip_tuple_fast
line=2700 msg="vd-root received a packet(proto=6,
203.160.224.97:80->192.168.11.59:31925) from port6."
```

Match existing session in reply direction:

```
id=20085 trace_id=213 func=resolve_ip_tuple_fast
line=2727 msg="Find an existing session, id-00000e90,
reply direction"
```

Apply destination NAT to inverse source NAT action:

```
id=20085 trace_id=213 func=__ip_session_run_tuple
line=1516 msg="DNAT 192.168.11.59:31925-
>192.168.3.221:1487"
```

Sample output: IPsec (policy-based)

```
id=20085 trace_id=1 msg="vd-root received a packet(proto=1, 10.72.55.240:1->10.71.55.10:8) from
internal."
id=20085 trace_id=1 msg="allocate a new session-00001cd3"
id=20085 trace_id=1 msg="find a route: gw-66.236.56.230 via wan1"
id=20085 trace_id=1 msg="Allowed by Policy-2: encrypt"
id=20085 trace_id=1 msg="enter IPsec tunnel-RemotePhase1"
id=20085 trace_id=1 msg="encrypted, and send to 15.215.225.22 with source 66.236.56.226"
id=20085 trace_id=1 msg="send to 66.236.56.230 via intf-wan1“
id=20085 trace_id=2 msg="vd-root received a packet (proto=1, 10.72.55.240:1-1071.55.10:8) from
internal."
id=20085 trace_id=2 msg="Find an existing session, id-00001cd3, original direction"
id=20085 trace_id=2 msg="enter IPsec ="encrypted, and send to 15.215.225.22 with source
66.236.56.226“ tunnel-RemotePhase1"
id=20085 trace_id=2 msgid=20085 trace_id=2 msg="send to 66.236.56.230 via intf-wan1"
```

Testing a proxy operation

To monitor proxy operations in the CLI:

```
diagnose test application <application> <option>
```

To display a list of available application values:

```
diagnose test application ?
```

To display a list of available option values:

```
diagnose test application <application> ?
```

The `<option>` value will depend on the application value used in the command.

For example, if the application is `http`, the CLI command that displays the `<option>` values is:

```
diagnose test application http ?
```

Displaying detail Hardware NIC information

Monitoring the hardware NIC is important because interface errors indicate data link or physical layer issues which may impact the performance of the FortiGate.

To monitor hardware network operations in the CLI:

```
diagnose hardware deviceinfo nic <interface>
```

or

```
diagnose netlink interface list <interface>
```

Sample output:

The following is sample output when the `<interface>` is set to `port1`:

```
# diagnose netlink interface list port1
```

```
if=port1 family=00 type=1 index=3 mtu=1500 link=0 master=0
flags=up broadcast run multicast
Qdisc=mq hw_addr=00:0c:29:fc:18:54 broadcast_addr=ff:ff:ff:ff:ff:ff
stat: rxp=61149 txp=81109 rxb=5839308 txb=52396373 rxe=0 txe=0 rxd=0 txd=0 mc=95 collision=0 @
time=1678486883
re: rxl=0 rxo=0 rxc=0 rxf=0 rxfi=0 rxm=0
te: txa=0 txc=0 txfi=0 txh=0 txw=0
misc rxc=0 txc=0
```

Field descriptions

The `diagnose hardware deviceinfo nic` and `diagnose netlink interface list` commands display lists of error names and values that are related to hardware.

The following table describes possible hardware errors:

Field	Description
Rx packets (rxp)	Number of received packets.
Tx packets (txp)	Number of transmitted packets.
Rx bytes (rxb)	Number of received bytes.
Tx bytes (txb)	Number of transmitted bytes.

Field	Description
Rx_Errors (rxe) = rx error count	Bad frame was marked as error by PHY.
Tx_Errors (txe) = Tx_Aborted_Errors	ECOL (Excessive Collisions Count); only valid in half-duplex mode.
Rx_CRC_Errors + Rx_Length_Errors - Rx_Align_Errors	This error is only valid in 10/100M mode.
Rx_Dropped (rxd) or Rx_No_Buffer_ Count	Running out of buffer space.
Tx_Dropped (txd)	Number of dropped packet.
Multicast (mc)	Number of multicast packets received.
Collisions	Total number of collisions experienced by the transmitter; valid in half-duplex mode.
Rx_Length_Errors (rxl)	Number of packets dropped due to invalid length.
Rx_Over_Errors (rxo)	Receive FIFO overflow event counter.
Rx_CRC_Errors (rxc)	Number of received packets with Frame CRC error.
Rx_Frame_Errors (rxf)	Same as Rx_Align_Errors. This error is only valid in 10/100M mode.
Rx_FIFO_Errors (rxfi)	Same as Rx_Missed_Errors - a missed packet count
Rx_Missed_Errors (rxm)	Equals Rx_FIFO_Errors + CEXTERR (Carrier Extension Error Count). Only valid in 1000M mode, which is marked by PHY.
Tx_Aborted_Errors (txa)	See Tx_Errors.
Tx_Carrier_Errors (txc)	The PHY should assert the internal carrier sense signal during every transmission. Failure to do so may indicate that the link has failed or the PHY has an incorrect link configuration. This register only increments if transmits are enabled. This register isn't valid in internal SerDes 1 mode (TBI mode for the 82544GC/EI) and is valid only when the Ethernet controller is operating at full duplex.
Tx_FIFO_Errors (txfi)	Number of Frame transmission error due to underflow.
Tx_Heartbeat_ Errors (txh)	Number of heartbeat error.

Field	Description
Tx_Window_Errors (txw)	Late Collisions (LATECOL) Count Late collisions are collisions that occur after 64-byte time into the transmission of the packet while working in 10 to 100 Mb/s data rate and 512-byte time into the transmission of the packet while working in the 1,000 Mb/s data rate. This register only increments if transmits are enabled and the device is in half-duplex mode.
Rx compressed (misc rxc)	Number of received compressed packets.
Tx compressed (misc txc)	Number of transmitted compressed packets.
Tx_Single_Collision_ Frames	Counts the number of times that a successfully transmitted packet encountered a single collision. The value increments only if transmits are enabled and the Ethernet controller is in half-duplex mode.
Tx_Multiple_ Collision_Frames	A Multiple Collision Count which indicates the number of times that a transmit encountered more than one collision, but less than 16. The value increments only if transmits are enabled and the Ethernet controller is in half-duplex mode.
Tx_Deferred	Counts defer events. A deferred event occurs when the transmitter cannot immediately send a packet due to: • The medium being busy because another device is transmitting. • The IPG timer has not expired. • Half-duplex deferral events are occurring. • XOFF frames are being received . • The link is not up. This register only increments if transmits are enabled. This counter does not increment for streaming transmits that are deferred due to TX IPG.
Rx_Frame_Too_ Longs	The Rx frame is oversized.
Rx_Frame_Too_ Shorts	The Rx frame is too short.
Rx_Align_Errors	This error is only valid in 10/100M mode.
Symbol Error Count	Counts the number of symbol errors (SYMERRS) between reads. The count increases for every bad symbol that's received, whether or not a packet is currently being received and whether or not the link is up. This register increments only in internal SerDes mode.

Specialized NIC interface diagnostics

Commands are available for displaying Forward Error Correction (FEC) status, RX/TX bits per second (bps), packets per second (pps), and host-level RX drop statistics in NIC interface diagnostics. This provides enhanced

visibility to assist with debugging and performance analysis.

> This feature is currently only supported on FortiGate model 1801F and interfaces that support FEC CL91.

The `diagnose hardware deviceinfo nic-details port<number>` command was added to display the current FEC status. The following example demonstrates reviewing the FEC status for port 37.

```
# diagnose hardware deviceinfo nic-details port37
System time: Thu Jun 12 17:25:22 2025
                fec_stats_acc   changes         last change
corr codewds    :0               0               never
uncorr codewds  :0               0               never
lane 0          :0               0               never
lane 1          :0               0               never
lane 2          :0               0               never
lane 3          :0               0               never
FEC BER: 0.000000
FEC symbol err rate: 0.000000
```

The `diagnose hardware deviceinfo nic port<number>` command was updated to add additional counters for a debug-related analysis. The `htx_drop`, `RX pps`, `Rx bps`, `Tx pps`, and `Tx bps` counters have been added. The `htx_drop` counter indicates if there is an FCS, oversized, or jabber error.

The following example demonstrates reviewing the hardware NIC information for port17.

```
# diagnose hardware deviceinfo nic port17
Description     :FortiASIC NP7 Adapter
Driver Name     :FortiASIC Unified NPU Driver
pid             :17
oid             :145
vid             :18
macid           :16
eif_id          :127
promiscous      :1
local_port      :0
vlan_wa_done    :0
mtu             :9216
netdev oid      :145
dev-flags       :1903
dev-promis      :1
Current_HWaddr   e0:23:ff:50:31:6a
Permanent_HWaddr e0:23:ff:50:31:6a
==== Default Link Settings =====
auto-nego       :Disable
s_speed         :1000
s_duplex        :Full
==== Current Link Settings =====
auto-nego       :Enable
s_status        :Up
s_speed         :1000
s_duplex        :Full
```

```
FEC             :OFF (0x2)
FEC_cap         :None (0x0)
SerDes_if       :SGMII
SerDes_if_cap   :SGMII (0x4)
SerDes_dflt     :2
pm_mode_setting :0
pm_mode         :0x0
pm_mode_dflt    :0
pm_port         :No
medium_mode     :0
==== Link Status ===============
Admin           :Up
link_status     :Up
Speed           :1000
Duplex          :Full
==== Netdev Status =============
dev_running     :Yes
dev_carrier     :On
==== Host Counters =============
hrx_pkts        :233
hrx_bytes       :26831
htx_pkts        :1940
htx_bytes       :210405
htx_drop        :0
htx_e_busy      :0
htx_e_noheadroom:0
htx_e_oid       :0
htx_e_adapter   :0
htx_e_pad       :0
htx_e_frag      :0
htx_e_other     :0
==== Netdev Counters ===========
Rx Pkts         :233
Rx Bytes        :31025
Tx Pkts         :1939
Tx Bytes        :218713
Rx pps          :1
Rx bps          :1483
Tx pps          :1
Tx bps          :1313
==== Switch Poll Counters ======
sw_tx_pkts      :1939
sw_tx_bytes     :218713
sw_tx_bc_pkts   :0
sw_tx_mc_pkts   :0
sw_rx_pkts      :233
sw_rx_bytes     :31025
sw_rx_bc_bytes  :27
sw_rx_mc_bytes  :159
==== Switch Error Counters =====
rx_err          :0
rx_crc_err      :0
```

```
rx_len_err      :0
rx_carrier_err  :0
rx_over_err     :0
rx_under_err    :0
rx_drop_pkts    :0
tx_collision_err:0
tx_drop_pkts    :0
======Queue  Counters ===========
uc_que0_pkts    :0
uc_que1_pkts    :0
uc_que2_pkts    :0
uc_que3_pkts    :0
uc_que4_pkts    :0
uc_que5_pkts    :0
uc_que6_pkts    :0
uc_que7_pkts    :0
-------------------------------
uc_drop_que0_pkts:0
uc_drop_que1_pkts:0
uc_drop_que2_pkts:0
uc_drop_que3_pkts:0
uc_drop_que4_pkts:0
uc_drop_que5_pkts:0
uc_drop_que6_pkts:0
uc_drop_que7_pkts:0
==== Transceiver Info ==========
tx_disable      : N/A
rx_los          : N/A
tx_fault        : N/A
present         : N/A
last_spd        : 0
```

Performing a traffic trace

Traffic tracing allows you to follow a specific packet stream. This is useful when you want to confirm that packets are using the route you expect them to take on your network.

To view traffic sessions:

Use this command to view the characteristics of a traffic session though specific security policies.

```
diagnose sys session
```

To trace per-packet operations for flow tracing:

```
diagnose debug flow
```

To trace per-Ethernet frame:

```
diagnose sniffer packet
```

To trace a route from a FortiGate to a destination IP address:

```
# execute traceroute www.fortinet.com
traceroute to www.fortinet.com (66.171.121.34), 32 hops max, 84 byte packets
1  172.20.120.2  0.637 ms  0.653 ms  0.279 ms
2  209.87.254.221 <static-209-87-254-221.storm.ca>  2.448 ms  2.519 ms  2.458 ms
3  209.87.239.129 <core-2-g0-2.storm.ca>  2.917 ms  2.828 ms  9.324 ms
4  209.87.239.199 <core-3-bdi1739.storm.ca>  13.248 ms  12.401 ms  13.009 ms
5  216.66.41.113 <v502.core1.tor1.he.net>  17.181 ms  12.422 ms  12.268 ms
6  184.105.80.9 <100ge1-2.core1.nyc4.he.net>  21.355 ms  21.518 ms  21.597 ms
7  198.32.118.41 <ny-paix-gni.twgate.net>  83.297 ms  84.416 ms  83.782 ms
8  203.160.228.217 <217-228-160-203.TWGATE-IP.twgate.net>  82.579 ms  82.187 ms  82.066 ms
9  203.160.228.229 <229-228-160-203.TWGATE-IP.twgate.net>  82.055 ms  82.455 ms  81.808 ms
10  203.78.181.2  82.262 ms  81.572 ms  82.015 ms
11  203.78.186.70  83.283 ms  83.243 ms  83.293 ms
12  66.171.127.177  84.030 ms  84.229 ms  83.550 ms
13  66.171.121.34 <www.fortinet.com>  84.023 ms  83.903 ms  84.032 ms
14  66.171.121.34 <www.fortinet.com>  83.874 ms  84.084 ms  83.810 ms
```

Using a session table

A session is a communication channel between two devices or applications across the network. Sessions allow FortiOS to inspect and act on a sequential group of packets in a session all at once instead of inspecting each packet individually. Each session has an entry in the session table that includes important information about the session.

You can view FortiGate session tables from the FortiGate GUI or CLI. The most useful troubleshooting data comes from the CLI. The session table in the GUI also provides useful summary information, particularly the current policy number that the session is using.

When to use a session table

Session tables are useful when verifying open connections. For example, if you have a web browser open to browse the Fortinet website, you would expect a session entry from your computer on port 443 to the IP address for the Fortinet website.

You can also use a session table to investigate why there are too many sessions for FortiOS to process.

To view session information in the GUI, go to *Dashboard > FortiView Sessions*. This view requires that a logging device is configured, and traffic log is enabled on a policy.

Finding the security policy for a specific connection

Every application or device that connects to another application or device must open a communication channel to exchange information. A stateful firewall tracks these sessions to ensure only packets exchanged in accordance to the session state are allowed to pass through the firewall. This includes verifying there is a policy match for such traffic, where features such as NATing, UTM scanning, and traffic shaping can be applied. Each session will have an entry in the session table.

If a secure web browser session is not working properly, you can check the session table to ensure the session is still active and going to the proper address. The session table can also tell you the security policy number it matches, so you can check what is happening in that policy.

1. Get the connection information.

To identify the session you want, you may need:

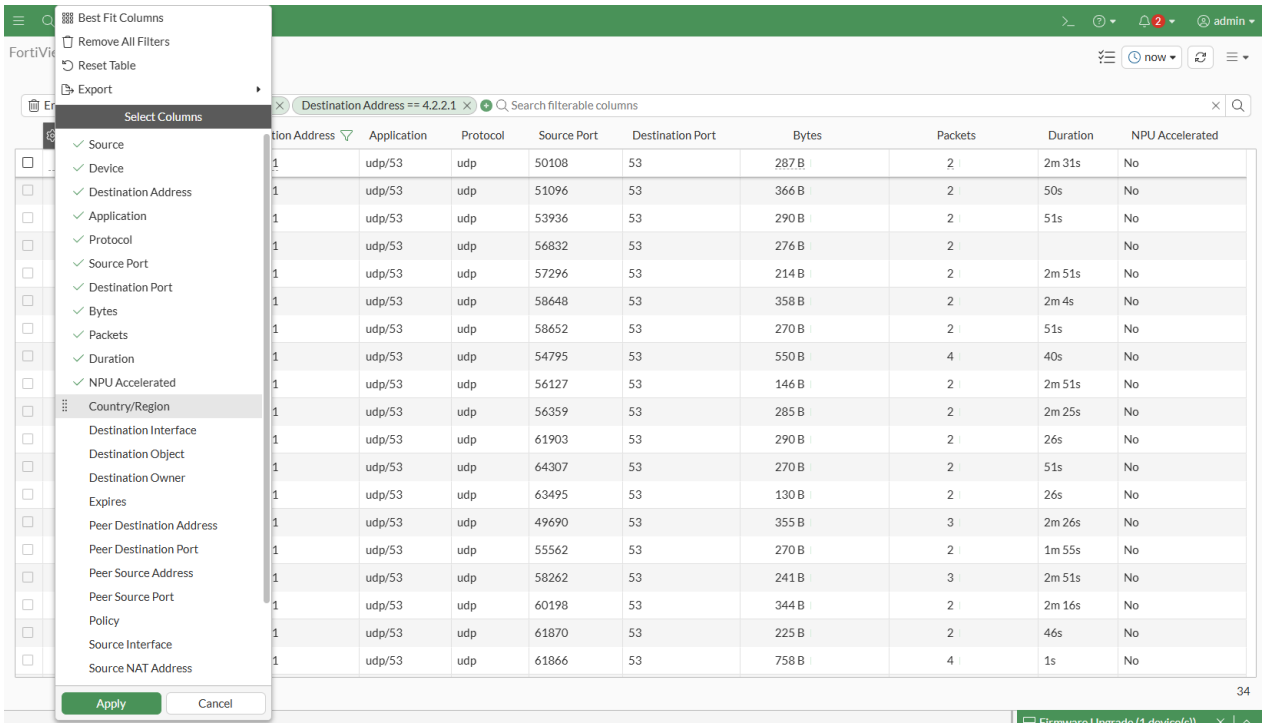
- The source IP address
- The destination IP address
- The port number used by the protocol. Common ports are:
  - Port 443 (HTTPS for SSL encrypted web browsing)
  - Port 22 (SSH for Secure Shell)
  - Port 25 (SMTP for Mail Transfer)

2. Find the session and policy ID

Go to *Dashboard > FortiView Sessions*. You can filter on each column header or you can type in a search by filterable columns.

For example, to filter on source, destination and port number, either go to each column header and select your value, or go to the search bar and select the column name and value one by one.

Additionally, you can select more columns to display by mousing over the column header, and clicking on the gear icon that displays.



This menu allows you to add or remove columns, as well as export the entire displayed sessions table to either a CSV or JSON file for further analysis.

Select *Policy* to add *Policy* to the view and click *Apply*. You can now identify the policy ID based on your search criteria.

CLI

Alternatively, instead of using the GUI you can also view the sessions from the CLI. The CLI has many filtering options and it displays granular information about each session.

To view session data in the CLI:

```
diagnose sys session list
```

To filter session data:

```
diagnose sys session filter <option>
```

The values for `<option>` include the following:

Value	Definition
`clear`	Clear session filter
`dintf`	Destination interface
`dport`	Destination port
`dst`	Destination IP address

Value	Definition
duration	Duration of the session
expire	Expire
ext-src	Add a source address to the extended match list.
ext-dst	Add a destination address to the extended match list.
ext-src-negate	Add a source address to the negated extended match list.
ext-dst-negate	Add a destination address to the negated extended match list.
negate	Inverse filter
nport	NAT'd source port
nsrc	NAT'd source ip address
policy	Policy ID
proto	Protocol number
proto-state	Protocol state
session-state1	Session state1
session-state2	Session state2
sintf	Source interface
sport	Source port
src	Source IP address
vd	Index of virtual domain, -1 matches all
vd-name	Name of virtual domain. -1 or any matches all.

An example of a TCP session:

```
# diagnose sys session list
session info: proto=6 proto_state=11 duration=59 expire=3586 timeout=3600 refresh_dir=both
flags=00000000 socktype=0 sockport=443 av_idx=9 use=5
origin-shaper=low-priority prio=4 guarantee 0Bps max 131072000Bps traffic 1826Bps drops 0B
reply-shaper=low-priority prio=4 guarantee 0Bps max 131072000Bps traffic 1826Bps drops 0B
per_ip_shaper=
class_id=0 shaping_policy_id=2 ha_id=0 policy_dir=0 tunnel=/ vlan_cos=0/255
state=redir local may_dirty os rs url_cat_valid
statistic(bytes/packets/allow_err): org=10941/16/1 reply=3633/11/1 tuples=3
tx speed(Bps/kbps): 0/0 rx speed(Bps/kbps): 0/0
orgin->sink: org pre->post, reply pre->post dev=6->3/3->6 gwy=172.19.50.1/0.0.0.0
hook=post dir=org act=snat 192.168.2.210:49397->54.177.212.176:443(172.19.50.87:49397)
hook=pre dir=reply act=dnat 54.177.212.176:443->172.19.50.87:49397(192.168.2.210:49397)
hook=post dir=reply act=noop 54.177.212.176:443->192.168.2.210:49397(0.0.0.0:0)
pos/(before,after) 0/(0,0), 0/(0,0)
misc=0 policy_id=9 pol_uuid_idx=15905 auth_info=0 chk_client_info=0 vd=0
serial=000016a0 tos=40/40 app_list=0 app=0 url_cat=142
```

```
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000100
no_ofld_reason:  redir-to-av
```

In the above example, a TCP/443 session is opened between the source (192.168.2.210) and the destination fortinet.com (54.177.212.176). To further understand the information, we will examine some fields in the output.

Session info:

```
session info: proto=6 proto_state=11 duration=59 expire=3586 timeout=3600 refresh_dir=both
flags=00000000 socktype=0 sockport=443 av_idx=9 use=5
```

Session info field	Description
`proto`	The protocol number. Common protocols used are: ICMP (proto 1) TCP (proto 6) UDP (proto 17)
`proto_state`	The state of the connection with respect to their protocol.
`duration`	Duration of the session (value in seconds).
`expire`	A countdown from the 'timeout' since the last packet passing via session (value in seconds).
`timeout`	An indicator of how long the session can stay open in the current state (value in seconds).

Session state:

```
state=redir local may_dirty os rs url_cat_valid
```

State	Explanation[KL1]
`may-dirty`	Session details are allowed to be altered.
`dirty`	The session has been altered (requires may-dirty).
`npu`	The session goes through an acceleration ship.
`npd`	The session is denied for hardware acceleration.
`npr`	The session is eligible for hardware acceleration (more information with `npu info: offload=x/y`).
`rem`	The session is allowed to be reset in case of a memory shortage.
`eph`	The session is ephemeral.
`oe`	The session is part of the IPsec tunnel (from the originator).
`re`	The session is part of the IPsec tunnel (from the responder).

State	Explanation[KL1]
`local`	The session is attached to the local FortiGate IP stack.
`br`	The session is bridged (VDOM is in transparent mode).
`redir`	The session is redirected to an internal FortiGate proxy.
`wccp`	The session is intercepted by wccp process.
`nlb`	The session is from a load-balanced vip.
`log`	The session is being logged.
`log-start`	The session is being logged when it starts as well (generate two logs for one session, `start` and `end`).
`os`	The session is shaped in the origin direction.
`rs`	The session is shaped by the reply direction.
`ndr`	The session is inspected by IPS signature.
`nds`	The session is inspected by IPS anomaly.
`auth`	The session is subject to authentication.
`authed`	The session was successfully authenticated.
`block`	The session was re-evaluated to block (policy changed).
`ext`	(Deprecated) The session is handled by a session helper.
`app_ntf`	Session matched a policy entry that contains `set block-notification enable`.
`F00`	After enabling traffic log in policy, the session will have this flag.
`pol_sniff`	After enabling packet capture in policy, session will have this flag.
`rst_tcp`	Flag visible when firewall policy has `timeout-send-rst enabled`.
`synced`	The session has been synchronized to HA peers, as seen on session-pickup enabled master.
`syn_ses`	The session has been synchronized from master to this peer, as seen on session-pickup enabled backup or HA-peer device.
`need_sync`	With 30 second HA sync delay. The session will be synced when reaching 30 seconds of lifetime.
`complex`	The session is handled by a session helper.
`app_valid`	The relevant rule has app control profile applied and FortiGate IPS engine was able to identify the application. (The session will have a field such as app= indicating the application.)
`url_cat_valid`	The relevant rule has web filter profile applied and FortiGate was able to identify the web category. (The session will have a field such as `url_cat=` `indicating the category`.)

Traffic directions:

```
orgin->sink: org pre->post, reply pre->post dev=6->3/3->6 gwy=172.19.50.1/0.0.0.0
hook=post dir=org act=snat 192.168.2.210:49397->54.177.212.176:443(172.19.50.87:49397)
hook=pre dir=reply act=dnat 54.177.212.176:443->172.19.50.87:49397(192.168.2.210:49397)
hook=post dir=reply act=noop 54.177.212.176:443->192.168.2.210:49397(0.0.0.0:0)
```

`dev=6->3/3->6` indicates the ingress and egress interfaces in both directions. The device and interface index can be obtained from:

```
# diagnose netlink interface list
```

In this case:

```
# diagnose netlink interface list | grep “index=6 “
if=port2 family=00 type=1 index=6 mtu=1500 link=0 master=0

# diagnose netlink interface list | grep "index=3 "
if=port1 family=00 type=1 index=3 mtu=1500 link=0 master=0
```

`gwy=172.19.50.1/0.0.0.0` indicates the gateway used in the original direction.

The next two lines display the source and destination, and NAT'd address in each direction:

```
<source_IP>:<source_port>-><destination_IP>:<destination_port>(<NAT_IP>:<NAT_port>)
```

Miscellaneous:

```
misc=0 policy_id=9 pol_uuid_idx=15905 auth_info=0 chk_client_info=0 vd=0
serial=000016a0 tos=40/40 app_list=0 app=0 url_cat=142
rpdb_link_id=00000000 ngfwid=n/a
npu_state=0x000100
no_ofld_reason:  redir-to-av
```

Misc fields	Description
`policy_id`	Policy ID matching this session
`pol_uuid_idx`	Policy index
`auth_info`	Indicates if the session holds any authentication data (`1`) or not (`0`).
`vd`	VDOM index. In the case where multiple VRFs are configured, VRF IDs can be obtained from the VD value: `vd=0:20`
`reflect info`	If this exists, it indicates an auxiliary or reflect session that created for the existing session.
`npu_state`	Provides the offloading details for the session, with the flag field which indicates whether this session is handling regular traffic (`flag=0x81`) or IPsec traffic (`flag=0x82`).
`no_ofld_reason`	The reason this session is not offloaded to NPU. For example, `redir-to-av` means the session requires AV inspection

Protocol states

Each protocol has its unique state which the firewall learns by examining the protocol headers. It is important for the firewall to keep the protocol state in order to anticipate the next packet it receives and to reject any unsolicited packets.

ICMP (proto 1)

There are no states for ICMP. It always shows `proto_state=00`.

TCP (proto 6)

The `proto_state` is a 2-digit number because the FortiGate is a keeps track of both directions of the session. The state values are:

State	Value
NONE	0
ESTABLISHED	1
SYN_SENT	2
SYN & SYN/ACK	3
FIN_WAIT	4
TIME_WAIT	5
CLOSE	6
CLOSE_WAIT	7
LAST_ACK	8
LISTEN	9

UDP (proto 17)

Even though UDP is a stateless protocol, FortiGate keeps track of the following states:

State	Value
UDP Reply not seen	0
UDP Reply seen	1

Examples

To examine the firewall session list in the CLI:

You can use a filter to limit the sessions displayed by source, destination address, port, or NAT'd address. To use more than one filter, enter a separate line for each value.

The following example filters the session list based on a source address of 10.11.101.112:

```
FGT# diagnose sys session filter src 10.11.101.112
FGT# diagnose sys session list
```

The following example filters the session list based on a destination address of 172.20.120.222.

```
FGT# diagnose sys session filter dst 172.20.120.222
FGT# diagnose sys session list
```

To clear all sessions corresponding to a filter:

```
FGT# diagnose sys session filter dst 172.20.120.222
FGT# diagnose sys session clear
```

Checking source NAT information

Checking source NAT is important when you are troubleshooting from the remote end of the connection outside the firewall.

To check the source NAT information in the CLI:

When you display the session list in the CLI, you can match the NAT'd source address (`nsrc`) and port (`nport`). This is useful when multiple internal IP addresses are NAT'd to a common external-facing source IP address.

```
FGT# diagnose sys session filter nsrc 172.20.120.122
FGT# diagnose sys session filter nport 8888
FGT# diagnose sys session list
```

Finding object dependencies

You may be prevented from deleting a configuration object when other configuration objects depend on it. You can use the GUI or CLI to identify objects which depend on, or make reference to the configuration you are trying to delete. Additionally, if you have a virtual interface with dependent objects, you will need to find and remove those dependencies before deleting the interface.

To remove interface object dependencies in the GUI:

1. Go to *Network* > *Interfaces*. The *Ref.* column displays the number of objects that reference this interface.
2. Select the number in the *Ref* . column for the interface. A window listing the dependencies appears.
3. Use these detailed entries to locate and remove object references to this interface. The trash can icon is enabled after all the object dependencies are removed.
4. Remove the interface by selecting the check box for the interface, and select *Delete*.

To find object dependencies in the CLI:

When running multiple VDOMs, use the following command in the global configuration only.

```
diagnose sys cmdb refcnt show <path.object.mkey>
```

The command searches for the named object in both the most recently used global and VDOM configurations.

Examples

To verify which objects a security policy with an ID of 1 refers to:

```
diagnose sys cmdb refcnt show firewall.policy.policyid 1
```

To check what is referred to by interface `port1`:

```
diagnose sys cmdb refcnt show system.interface.name port1
```

To show all dependencies for an interface:

```
diagnose sys cmdb refcnt show system.interface.name <interface name>
```

Sample output:

In this example , the interface has dependent objects, including four address objects, one VIP, and three security policies.

```
entry used by table firewall.address:name '10.98.23.23_host’
entry used by table firewall.address:name 'NAS'
entry used by table firewall.address:name 'all'
entry used by table firewall.address:name 'fortinet.com'
entry used by table firewall.vip:name 'TORRENT_10.0.0.70:6883'
entry used by table firewall.policy:policyid '21'
entry used by table firewall.policy:policyid '14'
entry used by table firewall.policy:policyid '19'
```

Diagnosing NPU-based interfaces

You can use the commands in this section to diagnose sessions offloaded to network processors (also called NPUs or NPs) in your FortiGate. Most FortiGates contain one or more of the following NPUs:

- NP7 or NP7Lite
- NP6, NP6XLite or NP6Lite

You can find your FortiGate unit in the Hardware Acceleration Guide to determine its NPU configuration.

Normally you can use the `diagnose debug flow` command to view sessions. However, this command only displays sessions processed by the CPU (also called software sessions). To view sessions offloaded to NPUs (also called hardware sessions), you must use the commands and techniques described in this section.

Alternatively, you can disable NPU offloading and then use the `diagnose debug flow` command. You should only disable the NPU functionality for troubleshooting purposes.

Diagnosing NP7 or NP7Lite sessions

Use the following command to list the NP7 processors in your FortiGate unit and the interfaces that they connect to:

```
diagnose npu np7 port-list
```

Use the following command to list the NP7Lite processors in your FortiGate unit and the interfaces that they connect to:

```
diagnose npu np7lite port-list
```

To use the NP7 packet sniffer

On FortiGates with NP7 and NP7Lite processors, you can use the following command to view sessions:

```
diagnose npu sniffer {start | stop | filter}
```

Here is a basic example to sniff offloaded TCP packets received by the port23 interface. In the following example:

- The first line clears the filter.
- The second line sets the sniffer to look for packets on port23.
- The third line looks for packets exiting the interface.
- The fourth line looks for TCP packets.
- The fifth line starts the sniffer.
- The sixth line starts displaying the packets on the CLI.

```
diagnose npu sniffer filter
diagnose npu sniffer filter intf port23
diagnose npu sniffer filter dir 1
diagnose npu sniffer filter protocol 6
diagnose npu sniffer start
diagnose sniffer packet npudbg
```

For more information, see NP7 packet sniffer or Tracing packet flow on FortiGates with NP7 processors.

See this Fortinet Community article for an NP7 packet sniffer example: Troubleshooting Tip: Collecting NP7 packet capture without disabling offload.

Diagnosing NP6, NP6XLite or NP6Lite sessions

Use either of the following commands to list the NP6 processors in your FortiGate unit and the interfaces that they connect to:

```
get hardware npu np6 port-list
diagnose npu np6 port-list
```

Use the following command to list the NP6XLite processors in your FortiGate unit:

```
get hardware npu np6xlite port-list
```

Use either of the following commands to list the NP6Lite processors in your FortiGate unit:

```
get hardware npu np6lite port-list
diagnose npu np6lite port-list
```

The output of all of these commands includes the device ID or `dev_id` of each NP processor. Only FortiGates with NP6 processors have multiple `dev_ids`. On FortiGates with one NP6, NP6Xlite, or NP6Lite processor, `dev_id` is always `0`.

To diagnose NP6, NP6XLite, or NP6 sessions, disable NPU offloading.

```
diagnose npu <processor> fastpath disable <dev_id>
```

Then use the `diagnose debug flow` command to view sessions.

Identifying the XAUI link used for a specific traffic stream

The `diagnose npu np6 xaui-hash` command takes a 6-tuple input of the traffic stream to identify the NP6 XAUI link that the traffic passes through.

This command is only available on the 38xxD, 39xxD, 34xxE, 36xxE, and 5001E series devices.

Syntax

```
diagnose npu np6 xaui-hash <interface> <proto> <src_ip> <dst_ip> <src_port> <dst_port>
```

Variable	Description
`<interface>`	The network interface that the packets are coming from.
`<proto>`	The proto number, 6 for TCP or 17 for UDP.
`<src_ip>`	The source IP address.
`<dst_ip>`	The destination IP address.
`<src_port>`	The source port.
`<dst_port>`	The destination port.

Examples

```
# diagnose npu np6 xaui-hash port1 6 1.1.1.1 2.2.2.1 4567 80
 NP6_ID: 0, XAUI_LINK: 2
```

```
# diagnose npu np6 xaui-hash port1 6 1.1.1.1 2.2.2.1 4567 200
NP6_ID: 6, XAUI_LINK: 2
```

```
# diagnose npu np6 xaui-hash port1 6 1.1.1.1 2.2.2.1 4567 20
NP6_ID: 1, XAUI_LINK: 2
```

```
# diagnose npu np6 xaui-hash port1 6 1.1.1.1 2.2.2.1 4567 23
NP6_ID: 1, XAUI_LINK: 1
```

The `NP6_ID` is the NP index of the model that is being used. It can be found with the `diagnose npu np6 port-list` command.

Running the TAC report

The Technical Assistance Center (TAC) report runs an exhaustive series of diagnostic commands. Some of the commands are only needed if you are using features, such as HA, VPN tunnels, or a modem. Fortinet support my ask you to use the report output to provide information about the current state of your FortiGate.

Due the amount of output generated, the report may take a few minutes to run. If you are logging CLI output to a file, you can run this command to familiarize yourself with the diagnostic commands.

To run the TAC report in the CLI:

```
execute tac report
```

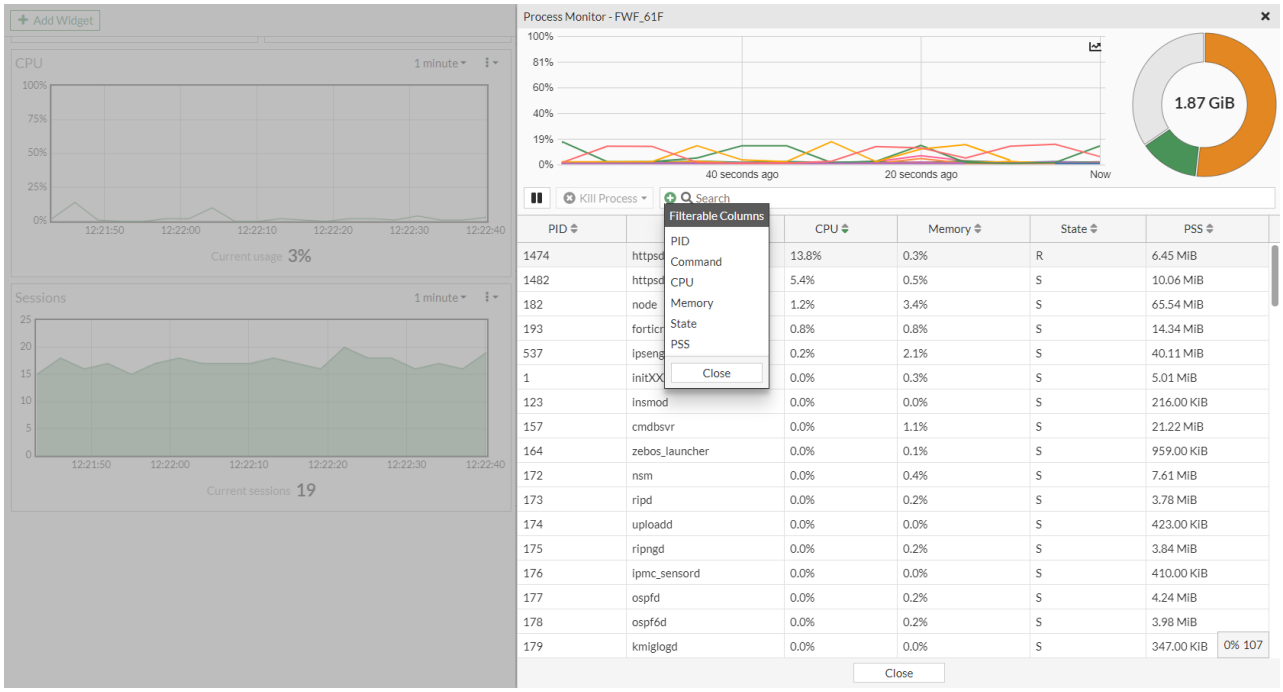
Using the process monitor

The *Process Monitor* displays running processes with their CPU and memory usage levels. Administrators can sort, filter, and terminate processes within the *Process Monitor* pane.

To access the process monitor:

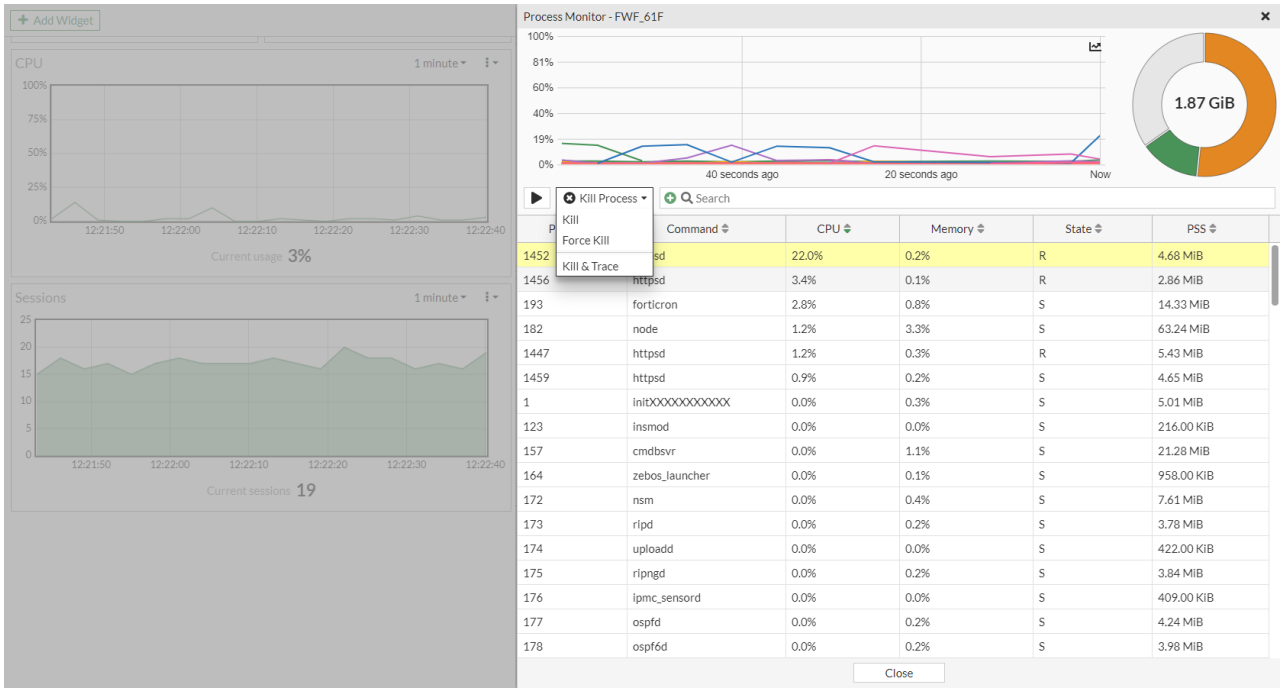
1. Go to *Dashboard* > *Status*:
   - Left-click in the *CPU* or *Memory* widget and select *Process Monitor*.
   - Click the user name in the upper right-hand corner of the screen, then go to *System* > *Process Monitor*.

   The *Process Monitor* appears, which includes a line graph, donut chart, and process list.
2. Click the *+* beside the search bar to view which columns can be filtered.

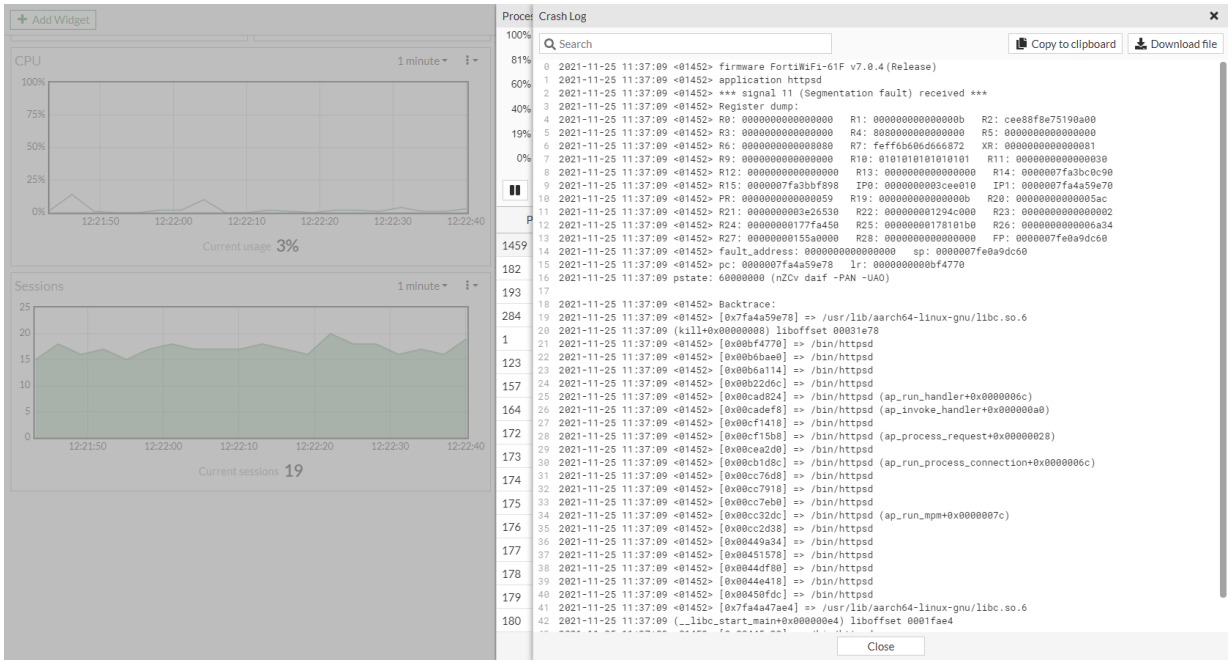


To kill a process within the process monitor:

1. Select a process.
2. Click the *Kill Process* dropdown.



3. Select one of the following options:
   - *Kill*: the standard kill option that produces one line in the crash log (`diagnose debug crashlog read`).
   - *Force Kill*: the equivalent to `diagnose sys kill 9 <pid>`. This can be viewed in the crash log.
   - *Kill & Trace*: the equivalent to `diagnose sys kill 11 <pid>`. This generates a longer crash log and backtrace. A crash log is displayed afterwards.



Computing file hashes

The following command computes the SHA256 file hashes for all of the files in a directory or directories:

```
# diagnose sys filesystem hash <paths> -d [depth]
```

<paths>	Add up to 25 paths to show only the hash for the files at those paths.
-d [depth]	Specify the maximum depth of the traversal.

This command can be used for troubleshooting and debugging the system. The file hashes of system files can be compared against known good system files to help identify any compromises made on the system files.

To hash all filesystems:

```
# diagnose sys filesystem hash
Hash contents: /bin
5132b40a66fd4cf062adb42e2af43cb9aea0672cf885f12978e8de2f3137834b        /bin/syslogd -> /bin/init
5132b40a66fd4cf062adb42e2af43cb9aea0672cf885f12978e8de2f3137834b        /bin/acd -> /bin/init
5132b40a66fd4cf062adb42e2af43cb9aea0672cf885f12978e8de2f3137834b        /bin/httpsnifferd ->
/bin/init
5132b40a66fd4cf062adb42e2af43cb9aea0672cf885f12978e8de2f3137834b        /bin/merged_daemons ->
/bin/init
...
/bin/init
6e2e07782dc17b8693268989f8ba1a8858a73d5291fb521e315011731cefe412        /bin/setpci
5132b40a66fd4cf062adb42e2af43cb9aea0672cf885f12978e8de2f3137834b        /bin/wad_csvc_cs ->
/bin/init
5132b40a66fd4cf062adb42e2af43cb9aea0672cf885f12978e8de2f3137834b        /bin/fds_notify ->
/bin/init
...
Hash contents: /lib
3dae8f9c15da465ffda24cebc1328725e98ee7c94a20e54af6ead7eaada45d9d        /lib/libusb-1.0.so.0
e50c6b5cad36b200d4903e4d7d5e5eac1f5c618d27fd6961011e28a892ed8866        /lib/libk5crypto.so.3
b021ad6fb16ce1e881ca586036687c1b2ae9555805817ef394284528d9e71612        /lib/libgomp.so.1
...
```

To hash specific filesystem, add the name of the filesystem:

```
# diagnose  sys  filesystem  hash  /sbin
Hash contents: /sbin
c1f81e67a53bcf70720748fe31c2380e95b4c3dfdb96957fd116fcf702bd797b        /sbin/init
Filesystem hash complete. Hashed 1 files.
```

To hash multiple filesystems, add the names of the filesystems:

Up to 25 file systems can be added.

```
# diagnose sys filesystem hash /sbin /bin
Hash contents: /sbin
```

```
c1f81e67a53bcf70720748fe31c2380e95b4c3dfdb96957fd116fcf702bd797b        /sbin/init
Hash contents: /bin
5132b40a66fd4cf062adb42e2af43cb9aea0672cf885f12978e8de2f3137834b        /bin/syslogd -> /bin/init
5132b40a66fd4cf062adb42e2af43cb9aea0672cf885f12978e8de2f3137834b        /bin/acd -> /bin/init
5132b40a66fd4cf062adb42e2af43cb9aea0672cf885f12978e8de2f3137834b        /bin/httpsnifferd ->
/bin/init
5132b40a66fd4cf062adb42e2af43cb9aea0672cf885f12978e8de2f3137834b        /bin/merged_daemons ->
/bin/init
```

To specify the maximum depth of the traversal:

```
# diagnose sys filesystem hash /data2 -d 1
Hash contents: /data2
a0166e804dc3d9a68fcc8015cb2d214ec40f0609e8e2aecc0eb2e5bdffc45524        /data2/new_alert_msg
7270b43899e0f72c7b9c94e66d64fd0e19881d91f74bd5ae6556eba045222e84        /data2/vir
8092e73c6a68f3cb02c86155bf3e55b2c1ab793eafcdd538beb5aa998d4b6b82        /data2/vir.x
2e29084d86f3925a0fb6bf96c4d83a6d3025fdd9cf8059ebcfc307153b9fd63b        /data2/virext
48ac27b0b5b10b3b0f3ab2f847406d524709c32117f6b721bb10448742bd5eb6        /data2/virext.x
2e29084d86f3925a0fb6bf96c4d83a6d3025fdd9cf8059ebcfc307153b9fd63b        /data2/virexdb
601316a029b28757c44515e37f48de2985d9fe8ef5c318e5f67e51369cba09f0        /data2/virexdb.x
7270b43899e0f72c7b9c94e66d64fd0e19881d91f74bd5ae6556eba045222e84        /data2/virfldb
896b71b3d9b209d339213f9d4af4088d3addd891cd292e93b5168eddb36b599a        /data2/virfldb.x
0af98283f9bcb7dff4974197f1c7f1b1013ec741c8cc6c1425119fb88f9a351b        /data2/ffdb_map_default_
res
627d2aed79770f698dbfc2bc0889f8285d1ea596c2dace8e6d3e7f00e040d990        /data2/madb.dat
96a296d224f285c67bee93c30f8a309157f0daa35dc5b87e410b78630a09cfc7        /data2/signature_result
ceab5e70a5368aa834842973241e1ae6ca49ff5c88afb6199e5d87e1749caeb1        /data2/revision_info_db
7eb70257593da06f682a3ddda54a9d260d4fc514f645237f5ca74b08f8da61a6        /data2/alci.dat
5840dfcf66d296be775e4e4d08bcdd014d1c91bd45e070587907d9eedab53e3e        /data2/uwdb
dc64fb8a291c7fc6d655474d00e2c42e7bb2b466de4489d33301f3ba82f64794        /data2/ffdb_pkg.tgz.x
c66a6ccc586ce29d38854a6afee49c0464fdc0064b59c4a104544325fd1ff03f        /data2/afdb
Filesystem hash complete. Hashed 17 files.
```

```
# diagnose sys filesystem hash /data2 -d 2
Hash contents: /data2
a0166e804dc3d9a68fcc8015cb2d214ec40f0609e8e2aecc0eb2e5bdffc45524        /data2/new_alert_msg
7270b43899e0f72c7b9c94e66d64fd0e19881d91f74bd5ae6556eba045222e84        /data2/vir
8092e73c6a68f3cb02c86155bf3e55b2c1ab793eafcdd538beb5aa998d4b6b82        /data2/vir.x
2e29084d86f3925a0fb6bf96c4d83a6d3025fdd9cf8059ebcfc307153b9fd63b        /data2/virext
48ac27b0b5b10b3b0f3ab2f847406d524709c32117f6b721bb10448742bd5eb6        /data2/virext.x
2e29084d86f3925a0fb6bf96c4d83a6d3025fdd9cf8059ebcfc307153b9fd63b        /data2/virexdb
601316a029b28757c44515e37f48de2985d9fe8ef5c318e5f67e51369cba09f0        /data2/virexdb.x
7270b43899e0f72c7b9c94e66d64fd0e19881d91f74bd5ae6556eba045222e84        /data2/virfldb
896b71b3d9b209d339213f9d4af4088d3addd891cd292e93b5168eddb36b599a        /data2/virfldb.x
0af98283f9bcb7dff4974197f1c7f1b1013ec741c8cc6c1425119fb88f9a351b        /data2/ffdb_map_default_
res
627d2aed79770f698dbfc2bc0889f8285d1ea596c2dace8e6d3e7f00e040d990        /data2/madb.dat
96a296d224f285c67bee93c30f8a309157f0daa35dc5b87e410b78630a09cfc7        /data2/signature_result
5ce22b4398f63fea2b47b7c1f00813a29851714993aee1269d3e95cbf43f4252        /data2/geodb/geoip.1
81ad258e278019dbd34fd07ba33966a6ff04e3fa352dddfe9ff362ac26d3cc88
/data2/config/cfg0000000001
```

```
e0067eb3d67b21cf39f27cb3558c5fbdafbc2c17c2afc29ab776b08e9c777a13
/data2/config/cfg0000000002
e77ad7c6b5d620d49f0f11933baf633335621de848a4229c3724152fff9aa4fa
/data2/config/cfg0000000003
228a7ed52779ba23f41a2423bfa7dbe858f24433f1702161f27678df4894f358
/data2/config/cfg0000000004
fe9e7afe7a6ccb739cb45c8d8f3b985377242ab61cc8199fa33dd475db49420f
/data2/config/cfg0000000005
b632b77348a54a2479453ab0f2c9f8e3c1e910badc8fbfb3fb841acf8eb4e35e
/data2/config/cfg0000000006
baeccb81d75f1f31503d42d3526f8831044144051f562486a89f1c5e4dd46d9c
/data2/config/cfg0000000007
ceab5e70a5368aa834842973241e1ae6ca49ff5c88afb6199e5d87e1749caeb1        /data2/revision_info_db
7eb70257593da06f682a3ddda54a9d260d4fc514f645237f5ca74b08f8da61a6        /data2/alci.dat
5840dfcf66d296be775e4e4d08bcdd014d1c91bd45e070587907d9eedab53e3e        /data2/uwdb
dc64fb8a291c7fc6d655474d00e2c42e7bb2b466de4489d33301f3ba82f64794        /data2/ffdb_pkg.tgz.x
c66a6ccc586ce29d38854a6afee49c0464fdc0064b59c4a104544325fd1ff03f        /data2/afdb
Filesystem hash complete. Hashed 25 files.
```

An error message is shown if an incorrect value is entered:

```
# diagnose sys filesystem hash /test-path
ERROR: Could not fetch info for path /test-path (No such file or directory)
Filesystem hash complete. Hashed 0 files.
```

```
# diagnose sys filesystem hash /bin -d 0
ERROR: depth must be greater than zero. (0)
Command fail. Return code -651
```

Other commands

You may be asked to provide the following information when you contact Fortinet support.

- ARP table on page 4995
- IP address on page 4999

ARP table

The ARP table is used to determine the destination MAC addresses of the network nodes, as well as the VLANs and ports that the nodes are reached from.

To view the ARP table:

```
# get system arp

Address           Age(min)   Hardware Addr      Interface
10.10.1.3         1          50:b7:c3:75:ea:dd internal7
```

```
192.168.0.190     0          28:f1:0e:03:2a:97 wan1
192.168.0.97      0          f4:f2:6d:37:b0:99 wan1
```

To view the ARP cache in the system:

```
# diagnose ip arp list

index=14 ifname=internal7 10.10.1.3 50:b7:c3:75:ea:dd state=00000004 use=2494 confirm=1995
update=374 ref=3
index=5 ifname=wan1 192.168.0.190 28:f1:0e:03:2a:97 state=00000002 use=88 confirm=86 update=977639
ref=2
index=22 ifname=internal 192.168.1.111 00:0c:29:c6:79:3d state=00000004 use=3724 confirm=9724
update=3724 ref=0
index=5 ifname=wan1 224.0.1.140 01:00:5e:00:01:8c state=00000040 use=924202 confirm=930202
update=924202 ref=1
index=5 ifname=wan1 192.168.0.97 f4:f2:6d:37:b0:99 state=00000002 use=78 confirm=486 update=614
ref=26
index=14 ifname=internal7 10.10.1.11 state=00000020 use=172 confirm=1037790 update=78 ref=2
```

To view a summary of the ARP table:

```
# diagnose sys device list root

list virtual firewall root info:
ip4 route_cache: table_size=65536 max_depth=2 used=31 total=34
arp: table_size=16 max_depth=2 used=5 total=6
proxy_arp: table_size=256 max_depth=0 used=0 total=0
arp6: table_size=32 max_depth=1 used=3 total=3
proxy_arp6: table_size=256 max_depth=0 used=0 total=0
local table version=00000000 main table version=0000002b
vf=root dev=root vrf=0
vf=root dev=ssl.root vrf=0
...
vf=root dev=internal5 vrf=0
ses=0/0 ses6=0/0 rt=0/0 rt6=0/0
```

ARP request, cache, and reachable time

When FortiGate tries to communication with a new destination it must resolve the destination's MAC address. The resolution depends on the IP address of the destination:

- Directly connected destination: ARP request is sent for the destination IP address.
- Non-directly connected destination: ARP request is sent for the gateway IP address of the exit interface.

After the ARP request is sent and an ARP reply is received, the corresponding ARP entry is added to the ARP cache, allowing FortiOS to use the cached entries for subsequent traffic and reducing the sending of frequent ARP request.

Entries in the ARP cache are valid for a duration equal to the actual ARP reachable time. The actual ARP reachable time is a random number between 50% and 150% of the base reachable time. The default base reachable time is 30 seconds, so the actual ARP reachable time is from 15 to 45 seconds. The actual ARP

reachable time is recalculated and updated every five minutes. The frequency of ARP requests to populate the ARP cache and unicast ARP probes to refresh the ARP cache entries depend on the base ARP reachable time.

Each ARP entry in the ARP cache includes its state and the number of objects that are currently using it. For example:

```
index=5 ifname=wan1 224.0.1.140 01:00:5e:00:01:8c state=00000040 use=924202 confirm=930202
update=924202 ref=1
```

Based on the state assigned to the ARP entry, the ARP cache determines the validity of the entry. There are multiple possible states for an ARP entry, and the state-transition mechanism can be complex. Common states include the following:

State	Meaning	Description
000000002 or 0x02	REACHABLE	An ARP response was received
000000004 or 0x04	STALE	No ARP response within the expected time
000000008 or 0x08	DELAY	A transition state between STALE and REACHABLE before Probes are sent out
000000020 or 0x20	FAILED	Did not manage to resolve within the maximum configured number of probes
000000040 or 0x40	NOARP	Device does not support ARP, e.g. IPsec interface
000000080 or 0x80	PERMANENT	A statically defined ARP entry

Multiple factors affect the state-transition mechanism and whether an entry is used by other subsystems. ARP creation, ARP request/reply, neighbor lookup, routing, and others can cause an ARP entry to be in use or referenced. Regular and successful unicast ARP probes initiated by FortiGate help maintain an ARP entry in Reachable state. By default, FortiGate disables regular unicast ARP probes for the sessions that are offloaded to the network processor (NP).

To set the ARP reachable time on an interface:

```
config system interface
    edit port1
        set reachable-time <integer>
    next
end
```

reachable-time <integer>	The reachable time (30000 to 3600000, default = 30000).

ARP cache purging

An ARP cache entry that is in the STALE (0x04) or FAILED (0x20) states with no references to it (ref=0) is eligible to be removed from the ARP cache using the garbage collection mechanism. The garbage collection mechanism runs every 30 seconds, and checks and removes ARP entries that have been stale, failed, or unreferenced longer than 60 seconds. Garbage collection is also triggered when the number of ARP entries exceeds the configured threshold. If the threshold is exceeded, no entries can be added to the ARP table.

To set the maximum number of ARP entries threshold:

```
config system global
    set arp-max-entry <integer>
end
```

arp-max-entry <integer>	The maximum number of dynamically learned MAC addresses that can be added to the ARP table (131072 to 2147483647, default = 131072).

ARP/ICMP6 probing for offloaded sessions

By default, FortiOS does not regularly send ARP probes for sessions offloaded to the NP. This can cause issues in environments where a session is established and offloaded to the NP, but no traffic flows through the FortiGate for an extended period of time, causing ARP entries on the FortiGate to expire and the MAC address table of upstream or downstream devices, such as switches, to age out. When this happens, an ARP broadcast request is triggered for any subsequent traffic passing through the session that generates a broadcast ARP request within the upstream or downstream switch network. To mitigate this, enable sending regular ARP probes for offloaded sessions.

To enable sending ARP/ICMP6 probing packets to update neighbors for offloaded sessions:

```
config system global
       set npu-neighbor-update enable
end
```

The npu-neighbor-update option is disabled by default.

Manually adding and removing ARP cache entries

To add static ARP entries:

```
config system arp-table
    edit 1
        set interface "internal"
        set ip 192.168.50.8
        set mac bc:14:01:e9:77:02
    next
end
```

To delete a single ARP entry from the ARP table:

```
diagnose ip arp delete <interface name> <IP address>
```

To clear all of the entries in the ARP table:

```
execute clear system arp table
```

IP address

You may want to verify the IP addresses assigned to the FortiGate interfaces are what you expect them to be.

To verify IP addresses:

```
diagnose ip address list
```

The output lists the:

- IP address and mask (if available)
- `index` of the interface (a type of ID number)
- `devname` (the interface name)

While physical interface names are set, virtual interface names can vary. A good way to use this command is to list all of the virtual interface names. For `vsys_ha` and `vsys_fgfm`, the IP addresses are the local host, which are virtual interfaces that are used internally.

Sample output:

```
# diagnose ip address list
IP=10.31.101.100->10.31.101.100/255.255.255.0 index=3 devname=internal
IP=172.20.120.122->172.20.120.122/255.255.255.0 index=5 devname=wan1
IP=127.0.0.1->127.0.0.1/255.0.0.0 index=8 devname=root
IP=127.0.0.1->127.0.0.1/255.0.0.0 index=11 devname=vsys_ha
IP=127.0.0.1->127.0.0.1/255.0.0.0 index=13 devname=vsys_fgfm
```

FortiGuard troubleshooting

The FortiGuard service provides updates to AntiVirus (AV), Antispam (AS), Intrusion Protection Services (IPS), Webfiltering (WF), and more. The FortiGuard Distribution System (FDS) consists of a number of servers across the world that provide updates to your FortiGate unit. Problems can occur with the connection to FDS and its configuration on your local FortiGate unit.

Some of the more common troubleshooting methods are listed here, including:

- Verifying connectivity to FortiGuard on page 4999
- Troubleshooting process for FortiGuard updates on page 5000
- FortiGuard server settings on page 5001
- FortiGuard web filter error logs on page 5003

Verifying connectivity to FortiGuard

You can verify FortiGuard connectivity in the GUI and CLI.

To verify FortiGuard connectivity in the GUI:

1. Got to *Dashboard* > *Status*.
2. Check the *Licenses* widget. When FortiGate is connected to FortiGuard, licensed services are in green icons.

To verify FortiGuard connectivity in the CLI:

```
execute ping service.fortiguard.net
```

```
execute ping update.fortiguard.net
```

Sample output:

```
FG100D# execute ping service.fortiguard.net
PING guard.fortinet.net (208.91.112.196): 56 data bytes
64 bytes from 208.91.112.196: icmp_seq=0 ttl=51 time=61.0 ms
64 bytes from 208.91.112.196: icmp_seq=1 ttl=51 time=60.0 ms
64 bytes from 208.91.112.196: icmp_seq=2 ttl=51 time=59.6 ms
64 bytes from 208.91.112.196: icmp_seq=3 ttl=51 time=58.9 ms
64 bytes from 208.91.112.196: icmp_seq=4 ttl=51 time=59.2 ms

--- guard.fortinet.net ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max = 58.9/59.7/61.0 ms

FG100D# execute ping update.fortiguard.net
PING fds1.fortinet.com (208.91.112.68): 56 data bytes
64 bytes from 208.91.112.68: icmp_seq=0 ttl=53 time=62.0 ms
64 bytes from 208.91.112.68: icmp_seq=1 ttl=53 time=61.8 ms
64 bytes from 208.91.112.68: icmp_seq=2 ttl=53 time=61.3 ms
64 bytes from 208.91.112.68: icmp_seq=3 ttl=53 time=61.9 ms
64 bytes from 208.91.112.68: icmp_seq=4 ttl=53 time=61.8 ms
```

Troubleshooting process for FortiGuard updates

The following process shows the logical steps you should take when troubleshooting problems with FortiGuard updates:

1. **Does the device have a valid license that includes these services?**
   Each device requires a valid FortiGuard license to access updates for some or all of these services. You can verify the status of the support contract for your devices at the Fortinet Support website.
2. **If the device is part of a high availability (HA) cluster, do all members of the cluster have the same level of support?**
   You can verify the status of the support contract for all of the devices in your HA cluster at the Fortinet Support website.
3. **Are services enabled on the device?**
   To see the FortiGuard information and status for a device in the GUI, go to *System* > *FortiGuard*.
   Use this page to verify the status of each component, and enable each service.

4. **Can the device communicate with FortiGuard servers?**
   Go to *System* > *FortiGuard* in the GUI, and try to update AntiVirus and IPS, or test the availability of Web Filtering and AS default and alternate ports.
5. **Is there proper routing to reach the FortiGuard servers?**
   Ensure there is a static or dynamic route that allows your FortiGate to reach the FortiGuard servers. Usually a generic default route to the internet is enough, but you may need to verify this if your network is complex.
6. **Are there issues with DNS?**
   An easy way to test this is to attempt a traceroute from behind the FortiGate to an external network using the Fully Qualified Domain Name (FQDN) for a location. If the traceroute FQDN name doesn't resolve, you have general DNS problems.
7. **Is there anything upstream that might be blocking FortiGuard traffic, either on the network or ISP side?**
   Many firewalls block all ports, by default, and ISPs often block ports that are low. There may be a firewall between the FortiGate and the FortiGuard servers that's blocking the traffic. By default, FortiGuard uses port 53. If that port is blocked you need to either open a hole for it or change the port it is using.
8. **Is there an issue with source ports?**
   It is possible that ports that FortiGate uses to contact FortiGuard are being changed before they reach FortiGuard or on the return trip before they reach FortiGate. A possible solution for this is to use a fixed-port at NAT'd firewalls to ensure the port remains the same. You can use packet sniffing to find more information about what's happening with ports.
9. **Are there security policies that include antivirus?**
   If none of the security policies include antivirus, the antivirus database will not be updated. If antivirus is included, only the database type that's used will be updated.

FortiGuard server settings

Your local FortiGate connects to remote FortiGuard servers to get updates to FortiGuard information, such as new viruses that may have been found or other new threats.

The default setting to reach FortiGuard is anycast. However, FortiGate can be configured to use unicast server. See FortiGuard on page 3906 for more information.

This section provides methods to display FortiGuard server information on your FortiGate, and how to use that information and update it to fix potential problems.

Displaying the server list

To get a list of FDS servers FortiGate uses to send web filtering requests:

```
get webfilter status
```

or

```
diagnose debug rating
```

Rating requests are only sent to the server at the top of the list in normal operation. Each server is probed for Round Trip Time (RTT) every two minutes. Rating may not be enabled on your FortiGate.

Optionally, you can add a refresh rate to the end of the command to determine how often the server list is refreshed.

Sample output:

```
Locale       : English

Service      : Web-filter
Status       : Enable
License      : Contract

Service      : Antispam
Status       : Disable

Service      : Virus Outbreak Prevention
Status       : Disable

Num. of servers : 2
Protocol        : https
Port            : 443
Anycast         : Disable
Default servers : Included

-=- Server List (Wed Nov 16 14:42:08 2022) -=-
IP                                            Weight    RTT Flags   TZ  FortiGuard-requests  Curr
Lost Total Lost             Updated Time
140.174.22.68                                     30    866         -5                    13
    0          0 Wed Nov 16 14:41:35 2022
12.34.97.18                                       30    878 DI      -5                    12
    0          0 Wed Nov 16 14:41:35 2022
```

Output details

The server list includes the IP addresses of alternate servers if the first entry cannot be reached. In this example, the IP addresses are not public addresses.

The following flags in `get webfilter status` indicate the server status:

Flag	Description
D	The server was found through the DNS lookup of the hostname. If the hostname returns more than one IP address, all of them are flagged with D and are used first for INIT requests before falling back to the other servers.
I	The server to which the last INIT request was sent
F	The server hasn't responded to requests and is considered to have failed
T	The server is currently being timed
S	Rating requests can be sent to the server. The flag is set for a server only in two cases: • The server exists in the servers list received from the FortiManager or any other INIT server. • The server list received from the FortiManager is empty so the FortiManager is the only server that the FortiGate knows and it should be used as the rating server.

Please note that the example output displays `Anycast` as `Disable` because the CLI commands above work with the FortiGuard unicast server case and not with the FortiGuard anycast servers case.

Also, in the example output above, the server 12.34.97.18 was found through a DNS lookup (`D` flag) and was sent the last INIT request (`I` flag).

Sorting the server list

The server list is sorted first by weight. The server with the smallest RTT appears at the top of the list, regardless of weight. When a packet is lost (there has been no response in 2 seconds), it is re-sent to the next server in the list. Therefore, the top position in the list is selected based on RTT, while the other positions are based on weight.

Calculating weight

The weight for each server increases with failed packets and decreases with successful packets. To lower the possibility of using a remote server, the weight isn't allowed to dip below a base weight. The base weight is calculated as the difference in hours between the FortiGate and the server multiplied by 10. The farther away the server is, the higher its base weight is and the lower it appears in the list.

FortiGuard web filter error logs

When troubleshooting issues related to web filter rating errors, it is helpful to review the following logs. These logs can be found in the FortiOS GUI under *Log & Report > Security Events > Web Filter*.

Shortened log	Message	Error	Reason
logid="0318012800" type="utm" subtype="webfilter" eventtype="ftgd_err" level="error"... msg="A rating error occurs" error="rating timeout"	A rating error occurs	rating timeout	Occurs if urlfilter sends a request to FortiGuard but does not receive a response before the packet retransmission interval.
logid="0318012801" type="utm" subtype="webfilter" eventtype="ftgd_err" level="warning" ... msg="A rating error occurs" error="bad network connection"	A rating error occurs	bad network connection	Occurs if urlfilter sends a request to FortiGuard but does not receive a response before the packet re-transmission interval. The number of lost packets for a given FortiGuard server exceeds a threshold

Shortened log	Message	Error	Reason
logid="0318012800" type="utm" subtype="webfilter" eventtype="ftgd_err" level="error" ... msg="A rating error occurs" error="no correct FortiGuard information"	A rating error occurs	no correct FortiGuard information	Occurs when urlfilter tried to send a request but was not able to reach any of the FortiGuard servers in its server list. This may be because the list is empty, or urlfilter has already tried all servers but did not get any response.
logid="0318012800" type="utm" subtype="webfilter" eventtype="ftgd_err" level="error" ... msg="A rating error occurs" error="unknown"	A rating error occurs	unknown	Occurs when the wad daemon does not specify a reason in the log or there is an internal urlfilter daemon issue.
logid="0318012800" type="utm" subtype="webfilter" eventtype="ftgd_err" level="error" ... msg="A rating error occurs" error="invalid license"	A rating error occurs	invalid license	Occurs when urlfilter daemon receives "invalid license" error response from Fortiguard server side when it is doing rating request.
logid="0318012800" type="utm" subtype="webfilter" eventtype="ftgd_err" level="error" ... msg="A rating error occurs" error="no rating service is enabled"	A rating error occurs	no rating service is enabled	Occurs when urlfilter service is not found or enabled.

Further actions may also be helpful in monitoring and troubleshooting the underlying web filtering issue:

- Set up Event Handlers in FortiAnalyzer to send alerts upon the appearance of the above logs. See the FortiAnalyzer Event Handler for more information.
- Check the FortiGuard Anycast Query status page for any current outages. An outage in the Web Filter Query will display under the line *FGT Anycast Web Filter Query Outage*.

View open and in use ports

Traffic destined for the FortiGate itself, and not being passed through or dropped, is called local-in traffic. It can be from a variety of services, such as HTTPS for administrative access, or BGP for inter-router communication.

Local-in traffic is controlled by local-in policies. To enable viewing local-in policies in the GUI, go to *System* > *Feature Visibility* and enable *Local In Policy*.

The *Policy & Objects > Local In Policy* page shows a read-only list of the local policies, populated with default values, and values that are automatically enabled when the related service is enabled, for example, enabling BGP opens TCP port 179. For more information, see Local-in policy on page 1886.

To view ports that are being listened on, and active connections and the services or processes using them:

```
# diagnose sys tcpsock | grep 0.0.0.0
0.0.0.0:10400->0.0.0.0:0->state=listen err=0 socktype=4 rma=0 wma=0 fma=0 tma=0 inode=10621
process=142/authd
...
0.0.0.0:53->0.0.0.0:0->state=listen err=0 socktype=1 rma=0 wma=0 fma=0 tma=0 inode=8067
process=177/dnsproxy
0.0.0.0:22->0.0.0.0:0->state=listen err=0 socktype=1 rma=0 wma=0 fma=0 tma=0 inode=13390
process=225/sshd
0.0.0.0:541->0.0.0.0:0->state=listen err=0 socktype=1 rma=0 wma=0 fma=0 tma=0 inode=13155
process=215/fgfmd
...
0.0.0.0:9980->0.0.0.0:0->state=listen err=0 socktype=1 rma=0 wma=0 fma=0 tma=0 inode=5063
process=129/httpsd
0.0.0.0:179->0.0.0.0:0->state=listen err=0 socktype=1 rma=0 wma=0 fma=0 tma=0 inode=10583
process=148/bgpd
...
```

For more information on incoming and outgoing ports, see the FortiOS Ports guide.

IPS and AV engine version

The IPS engine is an important module that processes traffic in policies configured with flow-based inspection, next generation firewall policies, as well as any policies that have IPS and application control defined. Just like its counterpart, the WAD daemon in proxy-based inspection, the IPS engine can invoke other daemons to perform additional processing such as certificate inspection, authentication, and other functions.

For each FortiOS release, an IPS engine is built into the firmware. You can find information about the IPS engine in its corresponding Release Notes.

When a FortiGate is configured for automatic FortiGuard updates and has policies configured to use the IPS engine, it downloads new releases of the IPS engine that are available through the FortiGuard Distribution Network. The IPS Engine package released to FortiGuard is unavailable for manual download.

The FortiGate supports manual upgrade/downgrade of the IPS engine in special cases, such as for troubleshooting or resolving a temporary issue that Technical Support deems necessary. In these cases, Technical Support distributes the IPS engine package.

Likewise, the AV engine is also built into the FortiOS firmware and available as an automatic update through FortiGuard. You can find information about the AV engine in its corresponding Release Notes.

Finally, for compatibility information between IPS and AV engines with FortiOS, see IPS Engine and AV Engine Support for FortiOS and FortiAPS

print tablesize

You can use the `print tablesize` command to view a list of the maximum values for your device. The four columns of values provide the following information, respectively:

1. Maximum number of variables allowed for this object type
2. Maximum number of objects of this type allowed per virtual domain (VDOM)
3. System global limit for maximum number of objects of this type
4. Current total number of objects of this type existing in FortiOS

A value of 0 indicates that there is no maximum value limit.

To check the maximum number of variables that you can append to all log messages:

```
# print tablesize
...
log.setting:custom-log-fields: 5 0 0 0
...
```

This means that you can add up to five variables:

```
config log setting
    set custom-log-fields <field_1>  <field_2>  <field_3>  <field_4>  <field_5>
end
```

To check the external resource table size limit for your device:

In this example, the device has a global limit of 512, a per-VDOM limit of 256, and there are currently no external resources existing in FortiOS. the available memory on the device limits the total number of feeds.

```
# print tablesize
...
system.external-resource: 0 256 512 0
...
```

To check the current total number of objects of this type existing in FortiOS:

```
# print tablesize
system.vdom: 0 0 10 3
system.datasource: 0 0 0 3
system.timezone: 0 0 0 597
system.accprofile: 0 0 10 5
system.np6xlite: 0 256 512 1
system.vdom-link: 0 0 0 3
...
```

CLI troubleshooting cheat sheet

See CLI troubleshooting cheat sheet.

CLI error codes

CLI error codes are shown in the command line if the command execution fails. The message includes a summary, followed by `Command fail. Return code -X`, where `-X` is the error code. For example:

```
# set test
Command parse error before 'test'
Command fail. Return code -61
```

The following table lists common error codes and their descriptions.

Return Code	Description
1	Generic CLI syntax error
-1	Invalid length of value.
-4	Maximum number of entries has been reached.
-5	A duplicate entry already exists.
-8	Invalid IP Address.
-37	Permission denied.
-56	Empty values are not allowed.
-61	Input not as expected.
-160	CFG_ER_GENERIC (common generic configuration error)
-553	Name conflicts with an interface, VDOM, switch-interface, zone, or interface name used for hardware switch interfaces.
-651	Input value is invalid.

Additional resources

To learn more about FortiGate and FortiOS, and for information about technical issues, refer to the following resources.

Fortinet Document Library

Installation Guides, Administration Guides, Quick Start Guides, and other technical documents are available online at the Fortinet Document Library.

Release notes

Issues that arise after the technical documentation has been published are often listed in the release notes. The release notes are available in the Fortinet Document Library.

Fortinet Video Library

The Fortinet Video Library hosts a collection of videos that provide valuable information about Fortinet products.

Fortinet Community

The Fortinet Community provides a place to collaborate, share insights and experiences, and get answers to questions. It incorporates the Fortinet Knowledge Base and technical discussion forums. You can access the Fortinet Community at https://community.fortinet.com.

Knowledge Base

The Fortinet Knowledge Base provides access to a variety of articles, white papers, and other documentation that provides technical insight into a range of Fortinet products. You can access the Knowledge Base at https://community.fortinet.com/t5/Knowledge-Base/ct-p/knowledgebase.

Fortinet technical discussion forums

The online technical forum allows administrators to contribute to discussions about issues related to their Fortinet products. Searching the forum can help an administrator identify if an issue has been experienced by another user. You can access the support forum at https://community.fortinet.com/t5/Fortinet-Forum/bd-p/fortinet-discussion.

Fortinet Training Institute

The Fortinet Training Institute hosts a collection of tutorials and training materials that you can use to increase your knowledge of Fortinet products. You can access these training resources at https://www.fortinet.com/training.html.

Fortinet Support

You defined your problem, researched a solution, put together a plan to find the solution, and executed that plan. At this point, if the problem has not been solved, contact Fortinet Support for assistance.

www.fortinet.com

Copyright© 2026 Fortinet, Inc. All rights reserved. Fortinet®, FortiGate®, FortiCare® and FortiGuard®, and certain other marks are registered trademarks of Fortinet, Inc., and other Fortinet names herein may also be registered and/or common law trademarks of Fortinet. All other product or company names may be trademarks of their respective owners. Performance and other metrics contained herein were attained in internal lab tests under ideal conditions, and actual performance and other results may vary. Network variables, different network environments and other conditions may affect performance results. Nothing herein represents any binding commitment by Fortinet, and Fortinet disclaims all warranties, whether express or implied, except to the extent Fortinet enters a binding written contract, signed by Fortinet's Chief Legal Officer, with a purchaser that expressly warrants that the identified product will perform according to certain expressly-identified performance metrics and, in such event, only the specific performance metrics expressly identified in such binding written contract shall be binding on Fortinet. For absolute clarity, any such warranty will be limited to performance in the same ideal conditions as in Fortinet's internal lab tests. Fortinet disclaims in full any covenants, representations, and guarantees pursuant hereto, whether express or implied. Fortinet reserves the right to change, modify, transfer, or otherwise revise this publication without notice, and the most current version of the publication shall be applicable.